Leonid Libkin
Ulrich Kohlenbach
Ruy de Queiroz (Eds.)

Logic, Language, Information, and Computation

LNCS 8071

20th International Workshop, WoLLIC 2013
Darmstadt, Germany, August 2013
Proceedings

Springer

Lecture Notes in Computer Science 8071

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

FoLLI Publications on Logic, Language and Information

Subline of Lectures Notes in Computer Science

Leonid Libkin Ulrich Kohlenbach
Ruy de Queiroz (Eds.)

Logic, Language, Information, and Computation

20th International Workshop, WoLLIC 2013
Darmstadt, Germany, August 20-23, 2013
Proceedings

Springer

Volume Editors

Leonid Libkin
University of Edinburgh
School of Informatics
EH8 9AB Edinburgh, UK
E-mail: libkin@inf.ed.ac.uk

Ulrich Kohlenbach
TU Darmstadt
Fachbereich Mathematik
Schlossgartenstrasse 7
64289 Darmstadt, Germany
E-mail: kohlenbach@mathematik.tu-darmstadt.de

Ruy de Queiroz
Universidade Federal de Pernambuco (UFPE)
Centro de Informática
Av. Jornalista Anibal Fernandes, s/n
Cidade Universitária
50.740-560 Recife, PE, Brazil
E-mail: ruy@cin.ufpe.br

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-39991-6 e-ISBN 978-3-642-39992-3
DOI 10.1007/978-3-642-39992-3
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013943913

CR Subject Classification (1998): F.4, I.2.3, F.3, I.1

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the papers of the 20th Workshop on Logic, Language, Information and Computation (WoLLIC 2013), which was held during August 20–23, 2013, at Fachbereich Mathematik, Technische Universität Darmstadt, Germany.

The WoLLIC series of workshops series started in 1994 with the aim of fostering interdisciplinary research in pure and applied logic. The idea is to have a forum which is large enough for dialogues between logic and the sciences relating to information and computation, and yet small enough for effective interactions among participants.

WoLLIC 2013 received 30 submissions, from which 17 were accepted for presentation at the workshop and for inclusion in the proceedings. Each submitted paper was reviewed by at least three members of the Program Committee, who were assisted by 33 external reviewers. We would like to thank the members of the Program Committee and the external reviewers for their review work, as well as Andrei Voronkov for his EasyChair system that made the review process and the preparation of this volume easy and smooth.

In addition to the contributed papers, the WoLLIC program contained invited lectures by Natasha Alechina (University of Nottingham), Steve Awodey (Carnegie Mellon University), Mikołaj Bojanczyk (Warsaw University), Wim Martens (Universität Bayreuth), Catuscia Palamidessi (INRIA Saclay and LIX), and Thomas Schwentick (TU Dortmund).

We would like to thank the entire local Organizing Committee (Martin Otto, Thomas Streicher, and Martin Ziegler from TU Darmstadt and Anjolina G. de Oliveira from Universidade Federal de Pernambuco) for making WoLLIC 2013 a success. Last, but not least, we acknowledge the financial support of Technische Universität Darmstadt (Fachbereich Mathematik), and the Deutsche Vereinigung für Mathematische Logik und für Grundlagenforschung der Exakten Wissenschaften (DVMLG), as well as the scientific sponsorship of the following organizations: Technische Universität Darmstadt (Fachbereich Mathematik), Universidade Federal de Pernambuco, Brazil, Interest Group in Pure and Applied Logics (IGPL), Association for Logic, Language and Information (FoLLI), Association for Symbolic Logic (ASL), European Association for Theoretical Computer Science (EATCS), European Association for Computer Science Logic (EACSL), Sociedade Brasileira de Computação (SBC), and Sociedade Brasileira de Lógica (SBL).

May 2013

Leonid Libkin
Ulrich Kohlenbach
Ruy de Queiroz

Organization

Program Chair

Leonid Libkin	University of Edinburgh, UK

Program Committee

Albert Atserias	UPC Barcelona, Spain
Alexandru Baltag	University of Amsterdam, The Netherlands
Stéphanie Delaune	ENS, CNRS, France
Amy Felty	University of Ottawa, Canada
Santiago Figueira	University of Buenos Aires, Argentina
Amélie Gheerbrant	University of Edinburgh, UK
Radha Jagadeesan	DePaul University, USA
Delia Kesner	University of Paris-Diderot, France
Benoit Larose	Concordia University, Canada
Fenrong Liu	Tsinghua University, China
Jerzy Marcinkowski	Wroclaw University, Poland
Peter O'Hearn	UCL, UK
Joel Ouaknine	Oxford University, UK
Gerald Penn	University of Toronto, Canada
Gabriele Puppis	CNRS/LaBRI - University of Bordeaux, France
R. Ramanujam	The Institute of Mathematical Sciences, India
Peter Selinger	Dalhousie University, Canada
Szymon Toruńczyk	Warsaw University, Poland
Anna Zamansky	University of Haifa, Israel

Steering Committee

Samson Abramsky	Angus Macintyre
Johan van Benthem	Grigori Mints
Anuj Dawar	Luke Ong
Joe Halpern	Hiroakira Ono
Wilfrid Hodges	Ruy de Queiroz
Daniel Leivant	

Additional Reviewers

Arieli, Ofer
Banerjee, Mohua
Baumann, Ringo
Berwanger, Dietmar
Bonelli, Eduardo
Boshuck, William
Brotherston, James
Bucheli, Samuel
Dapoigny, Richard
Dawar, Anuj
Demri, Stéphane
Durand, Arnaud
Fitting, Melvin
Galliani, Pietro
Gorín, Daniel
Hermant, Olivier
Krishnaswami, Neelakantan
Markey, Nicolas
Maubert, Bastien
Oliva, Paulo
Regnier, Laurent
Renne, Bryan
Sack, Joshua
Salvati, Sylvain
Sirangelo, Cristina
Spendier, Lara
Steedman, Mark
Valiron, Benoît
van Den Berg, Benno
Waller, Emmanuel
Wansing, Heinrich
Yu, Junhua
Zawadowski, Marek

Table of Contents

Logic and Agent Programming Languages

Natasha Alechina

School of Computer Science
University of Nottingham
nza@cs.nott.ac.uk

Abstract. Agent programming languages based on the Belief, Desire and Intentions (BDI) framework present some interesting challenges for logicians. While BDI logics have been studied extensively, problems related to belief update and analysis of plans in BDI agent programming languages have received less attention.

1 Introduction

This paper describes work in progress and proposes some possible research questions. It first introduces main ideas of agent programming languages and then describes some interesting open problems in agent programming language design that relate to logic, database theory and reasoning about actions. The paper is targeted mainly at logicians and not at agent programming languages researchers, and presents agent programming languages in a somewhat simplified way, abstracting away from many subtle differences between them. The aim of the paper is to encourage more logicians to investigate open problems in this area. Most of the questions discussed in this paper arose in discussions with agent programming languages researchers: Brian Logan, Koen Hindriks, Mehdi Dastani and Rafael Bordini.

2 Agent Programming Languages

There are many definitions of 'agent' in the literature [1]. Key ideas include:

autonomy: an agent operates without the direct intervention of humans or other agents
situatedness: an agent interacts with its environment (which may contain other agents)
reactivity: an agent responds in a timely fashion to changes in its environment
proactivity: an agent exhibits goal-directed behaviour

Arguably the dominant agent programming paradigm is the Belief, Desire and Intentions (BDI) model [2]. The BDI model is based on the work of Michael Bratman [3], and views an agent as a computational system whose behaviour can be usefully characterised in terms of propositional attitudes such as beliefs and goals, and which is programmed in an agent programming language that makes

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 1–10, 2013.

explicit use of propositional attitudes. *BDI agent programming languages* are designed to facilitate the implementation of BDI agents. A BDI agent programming language has programming constructs corresponding to beliefs, desires and intentions. An agent *architecture* or *interpreter* enforces relationships between the agent's beliefs, desires and intentions, and causes the agent to choose actions to achieve its goals based on its beliefs. One of the first BDI agent programming languages was the Procedural Reasoning System (PRS) [4]. Example applications of PRS include space shuttle fault diagnosis, controlling a mobile robot, air traffic control, business process control etc. PRS was very influential, with many derivatives (e.g., PRS-CL, PRS-Lite, dMARS), and modern BDI agent programming languages such as AgentSpeak(L) [5], CAN [6], SPARK [7], Jason [8], 2APL [9], and Goal [10] are based on similar ideas.

The basic idea of a BDI agent programming language is that the agent's state contains beliefs (about the agent's environment), goals (states the agent desires to bring about), and plans (sequences of actions the agent intends to carry out). *Beliefs* are typically represented as facts and Horn clause rules. For example, an agent may have the following beliefs about the world:

$$Location(home)$$

$$Seaside(x) \leftarrow NextTo(sea, x)$$

Goals or desires are usually conjunctions of literals describing desired states of affairs the agent would like to achieve. For example:

$$Location(x) \wedge Seaside(x)$$

is a goal to be at the seaside (x implicitly existentially quantified; any seaside would do).

Finally, agents have a set of *plans* (intentions) they have adopted and are in the process of executing. In most BDI languages plans look like imperative programs (sequential composition, if tests and while loops) which may contain variables in actions and tests. For example:

$$\text{if } (Location(x) \wedge Seaside(y) \wedge CheapFlight(x, y))\ buyTicket(x, y); go(x, y)$$

This means: find some substitution for x and y which satisfies

$$Location(x) \wedge Seaside(y) \wedge CheapFlight(x, y)$$

and then execute a sequence of actions

$$buyTicket(x, y); go(x, y)$$

for those values of x and y. The sets of beliefs, goals and plans are referred to as the agent's belief base, goal base and plan base.

The agent operates in a cycle:

belief update. The agent checks what is happening in the environment and updates its belief base accordingly.

intention adoption. The agent decides, using its current beliefs and goals, which new plans to adopt (if any), and adds them to its plan base.

intention execution. The agent decides which actions (forming part of current intentions) to execute, and executes them.

3 Briefly: BDI Logics

The development of first agent programming languages was accompanied by the development of BDI logics, which formalise logical relationships between beliefs, desires and intentions. See, for example, [11, 12], and subsequent work, for example, [13, 14].

The logics study questions such as relationships between beliefs and goals, when a rational agent should adopt and drop intentions, should logical consequences of intentions be intended, etc. These are deep questions, and the rational properties of the relationship between, for example, beliefs and goals proposed by different authors are sometimes completely the opposite of each other. For example, Cohen and Levesque [12] gave a standard possible world semantics for BEL and $GOAL$ modalities, requiring that goal-accessible (desirable) worlds are a subset of belief-accessible worlds. This makes sense because of all the worlds the agent considers possible, only some are desirable, and it does not make sense to have desirable states which are not considered possible. This property corresponds to the axiom:

$$BEL\,\phi \rightarrow GOAL\,\phi$$

On the other hand, Rao and Georgeff [11] have a more complex semantics where possible worlds are trees (branching histories) and for each belief-accessible history, there is a goal-accessible sub-history inside it, intutitively representing those courses of events which the agent finds desirable. Achievement goals are expressed by existential temporal formulas (there is a future state satisfying the goal), which are preserved under extensions. This means that for an existential temporal formula ϕ, the following axiom holds:

$$GOAL\,\phi \rightarrow BEL\,\phi$$

Finally, in modern agent programming languages, achievement goals correspond to state properties (describing a desirable state). Since it does not make sense for the agent to intend what it already believes is achieved, the following axiom holds:

$$GOAL\,\phi \rightarrow \neg BEL\,\phi$$

There are clearly problems associated with defining beliefs and goals as standard modalities, similar to logical omniscience in standard epistemic logics, and it is possible to define BDI logics which do not have this problem: see, for example, [15].

There is a lot of interesting work on BDI logics, but this paper focusses on other aspects of agent programming languages where logical reasoning is involved: the belief update phase and the intention adoption phase of the BDI agent execution cycle.

4 Belief Update

In a typical modern BDI agent programming language, the set of beliefs is essentially a deductive database. It consists of ground atoms (facts, EDB) and Horn clause rules, which define a different set of predicates (IDB). However, there are extensions of programming languages such as Jason which include ontologies and other sources of definitions in the belief base [16], and people would really like agents to be able to do some temporal reasoning [17].

4.1 Standard Belief Update

Let us use the term 'standard belief update' to refer to belief update for the case when the agent's belief base is a deductive database, and an update is represented as two sets of ground atoms A^+ and A^- (an add list and a delete list). A^+ are the facts which have become true, and A^- are the facts which have become false. The idea of course is that $A^+ \cap A^- = \emptyset$. Then the result of updating a belief base S with (A^+, A^-) is $S' = (S \cup A^+) \setminus A^-$. There is nothing else to do to ensure the belief base is consistent, since the sets of EDB and IDB predicates are disjoint and hence 'false' facts can not be derived from the update.

Usually, at this stage, those elements of the goal base which have become true (derivable from the belief base) are removed from the goal base. (The reason for this is that in case of declarative achievement goals, there is no point trying to achieve them if they are already true.)

4.2 Standard Belief Update in the Presence of Query Caching

The intention adoption and intention execution stages may involve evaluating queries against the agent's belief base (and possibly goal base). If the results of queries are cached (remembered for further use), the belief update problem becomes less trivial (since it involves updating the cache).

First we discuss the queries which need to be evaluated and why it is a good idea to cache them.

The intention adoption phase requires checking which plans are applicable given the agent's beliefs and possibly given its goals. A typical approach is to have 'plan adoption rules' of the form

$$\phi \leftarrow \psi \mid \pi$$

which mean 'if query ϕ succeeds against the goal base, and, extending the same substitution, query ψ succeeds against the belief base, then adopt the plan π with the resulting substitution'. Queries are usually built from literals (where $\neg$ is interpreted as negation as failure, and any variables in its scope should also appear in the query in a positive literal) using disjunctions and conjunctions.

For example:

$$On(x, y) \leftarrow Block(x) \wedge Block(y) \wedge \neg(x = y) \wedge Clear(x) \wedge Clear(y) \mid stack(x, y)$$

The intention execution phase may require evaluating belief tests, as in the previous example

$$\text{if } (Location(x) \wedge Seaside(y) \wedge CheapFlight(x,y))\ buyTicket(x,y); go(x,y)$$

However, studying existing programs in various agent programming languages suggests that the agents repeatedly evaluate the same queries in the same cycle and across different cycles [18], so it does make sense to cache complex derived formulas.

Assume that in addition to the belief base S, we also have a set of cached query results C. Each element of C is a pair (ϕ, θ) where ϕ is a query and θ is a substitution which currently makes ϕ true. Depending on the agent programming language, a query may return a set of answers (substitutions) or a single substitution.

In the presence of caching, the belief update problem becomes more involved than in the standard case, since we need to decide how to update C given S and (A^+, A^-). (The result of updating S with (A^+, A^-) is defined as before: $S' = (S \cup A^+) \setminus A^-$.) In fact we have two problems:

invalidation of cached beliefs. Given a change of belief base from S to S', which of the cached beliefs should be removed from C?

maintaining a complete cache. Given a change of belief base from S to S', are there are any new elements which need to be added to C? (In other words, when does it make sense to re-evaluate a query to get new answers?) The second problem occurs when we want to cache *all* answers to a query.

Invalidation of Cached Beliefs. The problem of invalidating cached beliefs can be solved using well known AI reason maintenance techniques [19]. We can keep track of which facts in S were used in deriving a particular element of C, and store them together with the element of C whose truth depends on them. In line with AI reason-maintenance terminology, let us call a set of literals used to derive a certain element (ϕ, θ) of C an *environment* for (ϕ, θ), and the set of all environments for (ϕ, θ) a *label* for (ϕ, θ). There may be several environments in one label, for example if S contains

$$P(x) \leftarrow Q_1(x), R(x)$$

$$P(x) \leftarrow Q_2(x), R(x)$$

and $Q_1(a)$, $Q_2(a)$ and $R(a)$ are all in S, then $(P(x), x/a)$ has a label with two environments, $\{Q_1(a), R(a)\}$ and $\{Q_2(a), R(a)\}$. Let us say that an environment is 'destroyed' if one of its elements is no longer in S. For example, if $Q_1(a)$ is no longer in S, then $\{Q_1(a), R(a)\}$ is destroyed. Destroyed environments are removed from the label. If some element of C has an empty label, it should be removed from C, because all old ways of deriving it have disappeared from S.

This works in a straightforward way if ϕ is an atomic query, and it is easy to see how to generalise this to conjunctions and disjunctions of atomic queries.

Negations are more of a problem, since we have negation as failure which is not stored as an explicit element of S. If we cache queries with negations, or if rules can have negative literals in the body, we need to store negative literals in the environments. For example, let S be

$$P(x) \leftarrow Q_1(x), R(x)$$
$$P(x) \leftarrow Q_2(x), R(x)$$
$$Q_1(a), Q_2(a), R(a), R(b)$$

and let ϕ be $R(x) \wedge \neg P(x)$. Given S, we have

$$(R(x) \wedge \neg P(x), x/b) \in C$$

and the only environment in its label is $\{R(b), \neg Q_1(b), \neg Q_2(b)\}$. An environment containing both positive and negative literals is destroyed by an update (A^+, A^-) if either one of its positive literals is in A^-, or for some negative literal $\neg L$, $L \in A^+$. Clearly, this cache update operation is much less computationally expensive than a call to the inference engine. Building and maintaining labels of cache elements in the case of only positive environments not more expensive that a call to an inference agent to derive them in the first place. In the case of environments with negative elements, efficient environment computation appears to be an open problem (it is obviously do-able, but can it be done as efficiently as computing positive environments?).

Maintaining a Complete Cache. The second problem (maintaining a complete cache) makes sense if we want the cache to always contain all answers to a query.

The question then is, given a set of cached queries and an update, which new substitutions should be added? Clearly, this question can be solved in a straightforward way by re-deriving all the queries, but we are looking for an efficient way of at least detecting when a new call to an inference engine is required (and ideally, just reducing this to a pattern matching problem on S, A^+ and A^-). In other words, we need to maintain a relationship between literals and complex queries such that if new literals of a certain form are added or removed, then a new substitution should be added to a query. It is likely that to solve this problem efficiently one would need to be able to compute something like a RETE or TREAT network [20, 21] but not for Horn clause rules but for arbitrary boolean combinations of literals and with negative literals as possible tokens.

Open problems. Here are some open problems related to cache update:

efficient standard update in the presence of a cache: how to compute and maintain environments with negative literals, how to efficiently maintain a complete cache when the agent's beliefs correspond to a deductive database

extensions to the standard case: how to update the cache if an update can contain disjunctions of literals; for which fragments of description logics and temporal logic cache update problems can be solved efficiently

5 Maintaining the Plan Base

In general the agent's plan base will contain several plans. These plans may be executed simultaneously, e.g., in a round robin fashion or an arbitrarily interleaved order. For example, the agent may be stacking blocks on several tables at the same time, using several robotic arms or moving from table to table. In some cases, making progress on several tasks at the same time is a good idea, while in other cases plans may interfere in unfortunate ways. The latter problem is intuitively clear, but how do we define a 'rational plan base'? There are several aspects to this notion, some to do with whether it makes sense to have several plans for the same goal (usually not) and others to do with how much work an agent can rationally commit to (considered at the end of this section). First, let us clarify the notion of 'interfering plans'.

One approach, see for example [22], considers it a bad idea to adopt plans for conflicting (logically inconsistent) goals, for example a goal to be in Amsterdam and a goal to be in Paris. A plan which involves achieving one of them will prevent achievement of another. Whether goals conflict is reasonably straightforward to check.

Another meaning of interfering plans [23–25] is that executing steps of one plan is undoing or making impossible executing steps of another plan, even if the goals of the two plans are consistent. For example, it is consistent to have two goals $Clean(room1)$ and $Clean(room2)$ (where $room1$ and $room2$ are different rooms in different directions from the agent), but most obvious plans for doing this would involve going to room 1 and going in the opposite direction to room 2.

Consider a blocks world environment [26]. Plans which involve manipulating blocks on the same table may interfere in both senses. Two goals may be inconsistent, for example $On(a, b)$ and $On(b, a)$.

An example of a situation where the goals are consistent but the plans may interfere is as follows. Suppose we have blocks a, b, c, d and e, and a and e are red and the rest are green. Suppose the first goal is to have a red block on top of two green blocks where the bottom block is on the table, and the second goal is to have a green block on top of a red block which is on the table. Both goals are achievable separately and even together, for example we can have

$$On(a, b) \wedge On(b, c) \wedge OnTable(c)$$

satisfying the first goal, and

$$On(d, e) \wedge OnTable(e)$$

satisfying the second goal. However suppose the following plans are adopted to achieve them (for example in the state where the agent believes that all blocks are on the table and clear):

$$\text{if } (OnTable(x) \wedge Green(x) \wedge Green(y))\ stack(y, x); \text{if } (Red(z))stack(z, y)$$

for the first goal and

$$\text{if } (OnTable(x) \wedge Red(x) \wedge Green(y))stack(y, x)$$

If both plans generate the same substitution for picking a red block (a) then one of them will become unexecutable, given the obvious pre- and postconditions of the *stack* action ($stack(x, y)$ is executable if x and y do not have anything on top of them, and result in $On(x, y)$ which means that x is on top of y).

Before formulating the problems related to checking for interference of plans, we need some definitions.

A *schedule* (of a set of actions) is an assignment of actions to processors (e.g. robotic arms) together with a linear order of each set of actions assigned to the same processor. For example, a schedule for two robotic arms could be:

$$arm1: \ stack(b, c) \prec stack(a, b)$$

$$arm2: \ stack(b, a)$$

where $\prec$ is the linear order (temporal precedence) relation.

Two actions *interfere* (or one of them clobbers another) if executing one of them makes execution of another impossible; in other words, the effect of the first action makes precondition of the second action false. In the schedule above, $stack(b, a)$ interferes with $stack(a, b)$ since the precondition of $stack(a, b)$ requires that a has nothing on top of it, and the effect of $stack(b, a)$ is that it does.

Open problems. There are a number of problems relating to the interference of plans, including:

plan interference: given a set of plans P and a set of pre- and postconditions of actions in those plans, do these plans interfere?

interference-free schedule: given a set of plans P and a set of pre- and postconditions of actions in those plans, return a schedule (if one exists) of actions of plans in P where action do not interfere

schedule interference: given a set of plans P, a set of pre- and postconditions of actions in those plans, and a schedule of actions in the plans, check whether actions in the schedule interfere

Clearly, these problems relate to planning and scheduling and can be solved using existing methods; however it would be good to find more efficient solutions which are tailored to plans in agent programming languages. One approach would be to investigate special scheduling algorithms which make use of logical checks before the plans are adopted to make scheduling an easier problem. It may also be possible to come up with a weaker notion of an acceptable ('almost free from interference') set of plans, for which the scheduling problem can be solved in polynomial time.

There are other aspects to deciding whether a set of plans the agent is committed to is 'rational', apart from interference of plans. For example, the goals the agent is trying to achieve may have deadlines. Given the time required to execute its plans, the agent may not be able to achieve all its goals, and will waste time and energy executing plans which are certain not to achieve their goal in time. Related problems have been investigated in [27–29]. In [29], we proposed an efficient scheduling algorithm which did not return the optimal schedule (which is computationally expensive) but a schedule satisfying reasonable 'rationality criteria'.

The same algorithm was used in a plan adoption algorithm for an agent which decides whether to commit to a set of obligations (which also have deadlines) in [30]. However, in this area much work remains to be done, trying to find a balance between computationally efficient and ideally rational criteria for plan schedules.

References

1. Wooldridge, M., Jennings, N.R.: Intelligent agents: Theory and practice. Knowledge Engineering Review 10(2), 115–152 (1995)
2. Georgeff, M., Pell, B., Pollack, M., Tambe, M., Wooldridge, M.: The Belief-Desire-Intention model of agency. In: Müller, J.P., Rao, A.S., Singh, M.P. (eds.) ATAL 1998. LNCS (LNAI), vol. 1555, pp. 1–10. Springer, Heidelberg (1999)
3. Bratman, M.: Intention, Plans, and Practical Reason. Harvard University Press (1987)
4. Georgeff, M.P., Lansky, A.L.: Reactive reasoning and planning. In: Proceedings of the Sixth National Conference on Artificial Intelligence, AAAI 1987, pp. 677–682 (1987)
5. Rao, A.S.: Agentspeak(l): Bdi agents speak out in a logical computable language. In: Van de Velde, W., Perram, J. (eds.) MAAMAW 1996. LNCS, vol. 1038, pp. 42–55. Springer, Heidelberg (1996)
6. Winikoff, M., Padgham, L., Harland, J., Thangarajah, J.: Declarative & procedural goals in intelligent agent systems. In: Fensel, D., Giunchiglia, F., McGuinness, D.L., Williams, M.A. (eds.) Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR 2002), Toulouse, France, pp. 470–481. Morgan Kaufmann (April 2002)
7. Morley, D., Myers, K.: The SPARK agent framework. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), pp. 714–721. IEEE Computer Society, Washington, DC (2004)
8. Bordini, R.H., Hubner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak using Jason. Wiley (2007)
9. Dastani, M.: 2APL: A practical agent programming language. Journal of Autonomous Agents and Multi-Agent Systems 16(3), 214–248 (2008)
10. Hindriks, K.V.: Programming rational agents in goal. In: El Fallah Seghrouchni, A., Dix, J., Dastani, M., Bordini, R.H. (eds.) Multi-Agent Programming: Languages, Tools and Applications, pp. 119–157. Springer US (2009)
11. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR 1991), pp. 473–484 (1991)
12. Cohen, P.R., Levesque, H.J.: Intention is choice with committment. Artificial Intelligence 42(2-3), 213–261 (1990)
13. Wooldridge, M.: Reasoning about Rational Agents. MIT Press (2000)
14. Meyer, J.J.C., van der Hoek, W., van Linder, B.: A logical approach to the dynamics of commitments. Artif. Intell. 113(1-2), 1–40 (1999)
15. Alechina, N., Logan, B.: A logic of situated resource-bounded agents. Journal of Logic, Language and Information 18(1), 79–95 (2009)
16. Mascardi, V., Ancona, D., Bordini, R.H., Ricci, A.: Cool-agentspeak: Enhancing agentspeak-dl agents with plan exchange and ontology services. In: Boissier, O., Bradshaw, J., Cao, L., Fischer, K., Hacid, M.S. (eds.) Proceedings of the 2011 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2011, Campus Scientifique de la Doua, Lyon, France, August 22-27, pp. 109–116. IEEE Computer Society (2011)

17. Bulling, N., Hindriks, K.V.: Taming the complexity of linear time BDI logics. In: Sonenberg, L., Stone, P., Tumer, K., Yolum, P. (eds.) 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, vol. 1-3, pp. 275–282. IFAAMAS (2011)
18. Alechina, N., Behrens, T., Dastani, M., Hindriks, K.V., Hubner, J., Logan, B., Nguyen, H., van Zee, M.: Multi-cycle query caching in agent programming. In: Proceedings of the Twenty-Seventh AAAI Conferernce on Artificial Intelligence (AAAI 2013), Bellevue, Washington. AAAI, AAAI Press (to appear, July 2013)
19. Doyle, J.: A truth maintenance system. Artificial Intelligence 12(3), 231–272 (1979)
20. Forgy, C.: Rete: A fast algorithm for the many pattern/many object pattern match problem. Artificial Intelligence 19(1), 17–37 (1982)
21. Miranker, D.P.: TREAT: A better match algorithm for AI production systems. In: Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI 1987), pp. 42–47. AAAI Press (1987)
22. van Riemsdijk, M.B., Dastani, M., Meyer, J.J.C.: Goals in conflict: semantic foundations of goals in agent programming. Autonomous Agents and Multi-Agent Systems 18(3), 471–500 (2009)
23. Thangarajah, J., Padgham, L.: Computationally effective reasoning about goal interactions. J. Autom. Reasoning 47(1), 17–56 (2011)
24. Thangarajah, J., Sardiña, S., Padgham, L.: Measuring plan coverage and overlap for agent reasoning. In: van der Hoek, W., Padgham, L., Conitzer, V., Winikoff, M. (eds.) International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 3 vols., pp. 1049–1056. IFAAMAS (2012)
25. Shapiro, S., Sardiña, S., Thangarajah, J., Cavedon, L., Padgham, L.: Revising conflicting intention sets in BDI agents. In: van der Hoek, W., Padgham, L., Conitzer, V., Winikoff, M. (eds.) International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 3 vols., pp. 1081–1088. IFAAMAS (2012)
26. Russell, S.J., Norvig, P.: Artificial Intelligence - A Modern Approach (3rd internat. edn.). Pearson Education (2010)
27. Bordini, R.H., Bazzan, A.L.C., de Oliveira Jannone, R., Basso, D.M., Vicari, R.M., Lesser, V.R.: AgentSpeak(XL): efficient intention selection in BDI agents via decision-theoretic task scheduling. In: Proceedings of the First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, Bologna, Italy, July 15-19, pp. 1294–1302. ACM (2002)
28. Thangarajah, J., Padgham, L.: An empirical evaluation of reasoning about resource conflicts. In: Proceedings of the Third International Conference on Autonomous Agents and Multiagent Systems, vol. 3, pp. 1298–1299 (2004)
29. Vikhorev, K., Alechina, N., Logan, B.: Agent programming with priorities and deadlines. In: Turner, K., Yolum, P., Sonenberg, L., Stone, P. (eds.) Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, pp. 397–404 (May 2011)
30. Alechina, N., Dastani, M., Logan, B.: Programming norm-aware agents. In: Conitzer, V., Winikoff, M., Padgham, L., van der Hoek, W. (eds.) Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012), Valencia, Spain, vol. 2, pp. 1057–1064. IFAAMAS (June 2012)

Natural Models of Homotopy Type Theory (Abstract)

Steve Awodey

Carnegie Mellon University

Homotopy type theory is an interpretation of constructive Martin-Löf type theory into abstract homotopy theory. It allows type theory to be used as a formal calculus for reasoning about homotopy theory, as well as more general mathematics such as can be formulated in category theory or set theory, under this new homotopical interpretation. Because constructive type theory has been implemented in computational proof assistants like Coq, it also facilitates the use of those tools in homotopy theory, category theory, set theory, and other fields of mathematics. This is the idea behind the new *Univalent Foundations Program*, which has recently been the object of quite intense investigation [4].

One thing missing from homotopy type theory, however, has been a notion of *model* that is both faithful to the precise formalism of type theory and yet general and flexible enough to be a practical tool. Past attempts have relied either on highly structured categories corresponding closely to the syntax of type theory, such as the *categories with families* of Dybjer [3], which are, however, somewhat impractical to work with semantically, or more natural and flexible categorical models based on homotopical algebra, as is done in [1,2], which however must be equipped with unnatural coherence conditions.

In the present work, I will present a new approach which, hopefully, combines the best of each of these two strategies. It is based on the observation that a category with families is the same thing as a representable natural transformation in the sense of Grothendieck. Ideas from Voevodsky [5] and Lumsdaine-Warren are also used.

Acknowledgments. The author would like to thank the Institute for Advanced Study, where this research was conducted. Support was provided by the Air Force Office of Scientific Research through award FA9550-11-1-0143 and by the National Science Foundation through award DMS-1001191. This material is based in part upon work supported by the AFOSR and the NSF under the above awards. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the AFOSR or the NSF.

References

1. Awodey, S., Warren, M.A.: Homotopy theoretic models of identity types. Math. Proc. Camb. Phil. Soc. 146, 45–55 (2009)
2. van den Berg, B., Garner, R.: Topological and Simplicial Models of Identity Types. ACM Transactions on Computational Logic 13(1) (2012)

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 11–12, 2013.

3. Dybjer, P.: Internal Type Theory. In: Berardi, S., Coppo, M. (eds.) TYPES 1995. LNCS, vol. 1158, pp. 120–134. Springer, Heidelberg (1996)
4. Homotopy Type Theory: Univalent Foundations of Mathematics, The Univalent Foundations Program, Institute for Advanced Study (2013), http://homotopytypetheory.org/book
5. Kapulkin, C., LeFanu Lumsdaine, P., Voevodsky, V.: The Simplicial Model of Univalent Foundations (in preparation, 2013)

Modelling Infinite Structures with Atoms

Mikołaj Bojańczyk*

University of Warsaw

Abstract. The topic is a variant of first-order logic defined in the framework of sets with atoms, which allows formulas to use orbit-finite boolean operations. The main contribution is a notion of model for this logic, which admits the compactness theorem.

1 Introduction

This paper studies a variant of first-order logic in sets with atoms (also known as nominal sets, also known as Fraenkel-Mostowski sets, also known as permutation models). The logic was introduced in [3], its intention is to describe properties of objects such as data words, data trees, or data graphs. The focus of [3] was the study of finite models, i.e. the logic made most sense for finite data words, finite data trees, and finite data graphs. When studying infinite objects, such as infinite data words, the approach from [3] runs against difficulties, e.g. the compactness theorem does not hold. The goal of this paper is a more thorough model-theoretic study of that logic, with a focus on the problems with infinite structures. This paper proposes a different semantics of first-order logic than the one in [3], which admits compactness, and has a sound and complete proof system.

Apart from the technical results on the logic, this paper discusses how to model and how not to model objects that talk about infinitely many atoms, while respecting the finite support constraint in the definition of sets with atoms. The running example in this paper is infinite data words: how can they be modelled using sets with atoms, and how logics can be used to express their properties.

2 Sets with Atoms

Sets with atoms were introduced in set theory by Fraenkel in 1922 and rediscovered for the computer science community, by Gabbay and Pitts [5]. Sets with atoms are now widely studied in the semantics community, under the name of nominal sets, see the book [9]. This paper is part of a research programme which studies a notion of finiteness which only makes sense in sets with atoms, called "orbit-finiteness". The research programme is to see what happens to discrete mathematics when sets are replaced by sets with atoms, and finiteness is replaces by orbit-finiteness. Two examples of this research programme include the study of finite automata in [2], and the study of programming languages [4].

* Author supported by ERC Starting Grant "Sosna".

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 13–28, 2013.

What are sets with atoms? Sets with atoms are an extended notion of a set – such sets are allowed to contain "atoms". The existence of atoms is postulated as an axiom. The key role in the theory is played by automorphisms of atoms. For instance, if a, b, c, d are atoms, and the atoms have no structure except for equality, then the sets

$$\{a, \{a, b, c\}, \{a, c\}\} \qquad \{b, \{b, c, d\}, \{b, d\}\}$$

are equal up to automorphisms of atoms. If the atoms are real numbers, equipped with the successor relation $x = y + 1$ and linear order $x < y$, then the sets

$$\{-1, 0, 0.3\} \qquad \{5.2, 6.2, 6.32\}$$

are equal up to automorphism of the atoms, but the sets

$$\{0, 2\} \qquad \{5.3, 8.3\}$$

are not.

The formal definition is parametrized by a notion of atoms, which is given as a relational structure, and induces a notion of automorphism. (One can also consider atoms with function symbols, but we do not do this here.) A *set with atoms* is any set that can contain atoms or other sets with atoms, in a well-founded way[1]. The key notion is the notion of a *legal* set of atoms, defined below. Suppose that X is a set with atoms. If π is an automorphism of atoms, then π can be applied to X, by renaming all atoms that appear in X, and appear in elements of X, and so on. We say that a set S of atoms is a *support* of the set X if X is invariant under every automorphism of atoms which is the identity on S. (For instance, the set of all atoms is supported by the empty set, because every automorphism maps the set to itself.) A set with atoms is called *legal* if it has some finite support, each of its elements has some finite support, and so on recursively.

Orbit-finiteness. Sets with atoms are a good abstraction for some infinite systems because they have a more relaxed notion of finiteness, which requires finitely many elements, but only up to automorphisms of atoms. More precisely, we say that a set with atoms X is *orbit-finite* if it is included in a union of finitely many single-orbit sets. A single orbit set is obtained from an element x (which may itself be a set) by applying all possible atom automorphisms.

Consider for example sets with atoms where the atoms have no structure, and therefore automorphisms are arbitrary permutations. The set of atoms itself is orbit-finite, actually has only one orbit, because every atom can be mapped to every other atom by a permutation. Likewise, the set of pairs of atoms has two elements up to permutation, namely (a, a) and (a, b) for $a \neq b$. The set of triples of atoms has five orbits: all coordinates are equal, all coordinates are distinct, and three orbits where exactly two coordinates are equal.

[1] Formally speaking, sets with atoms are defined by induction on their rank, which is an ordinal number. Sets of a given rank can contain atoms and sets of lower rank.

Homogeneous structure. A structure is called *homogeneous* if every finite partial automorphism (i.e. an isomorphism between two finite substructures) can be extended to a full automorphism. Examples of homogeneous structures include: the natural numbers with only equality (we call this the *equality atoms*), and the ordered rational numbers (we call this the *total order atoms*). We assume in this paper that the atom structure is a countable homogeneous relational structure over a finite relational vocabulary. This guarantees that the notion of orbit-finiteness is relatively well behaved, e.g. orbit-finite sets are closed under finite Cartesian products and finitely-supported subsets.

3 First-Order Logic and Flat Models

The topic of this paper is a study of first-order logic in sets with atoms. In this section, we take a first attempt at defining first-order logic and the models that it will be evaluated in. Later in the paper, we will present more sophisticated versions of both the logic and the models.

We are interested mainly in legal sets with atoms, and therefore we adopt the convention that, unless otherwise stated, all sets with atoms are implicitly assumed to be legal, such as in the following definition. The definition of vocabulary is the same as usual, only sets are required to be sets with atoms. A vocabulary is any set with atoms Σ, whose elements are called *predicates*, together with a finitely supported arity function, which maps predicates of Σ to their arities, which are natural numbers. We will write Σ_n for the set of n-ary predicates, i.e. the inverse image of n under the arity function. In this paper, we only study countable vocabularies.

Running example. [A vocabulary for describing data words] In a running example, we will talk about infinite data words. The idea is to model an ω-word (i.e. positions are the natural numbers) where each position is labelled by an atom. The vocabulary for this will contain one binary predicate $x < y$ (which will be used to order the positions), and one unary predicate $a(x)$ for every atom $a \in \mathbb{A}$ (which will be used to determine the labels of the positions). Observe that this vocabulary is infinite, but orbit-finite. The predicate $x < y$ is a singleton orbit, while all the predicates $a(x)$ are in the same orbit. □

3.1 Flat Models and Stand First-Order Logic

Flat models. A *flat model* is obtained by taking the standard definition of a model and interpreting it in sets with atoms. As we shall later see, this notion of model does not behave as well as the standard notion of model.

A *flat model* $\mathfrak{A}$ over a vocabulary Σ consists of

- A set with atoms A called the *universe* of the model.
- For every predicate $R \in \Sigma$, an interpretation $R^{\mathfrak{A}} \subseteq A^{arity(n)}$.

A flat model is a special case of a set with atoms (it is a pair of a set and a set of interpretations functions). In this special case, legality means: a) the

universe should be finitely supported; b) for every R, the set $R^{\mathfrak{A}}$ should be finitely supported; and c) the function $R \mapsto R^{\mathfrak{A}}$ must also be finitely supported, as a function from the vocabulary to sets of tuples of elements in the universe.

However, legal flat models turn out to be so weak that we will consider possibly illegal flat models in some constructions. That is why, for flat models, we make an exception to the convention that every object is implicitly legal, and explicitly write "legal flat model" or "possibly illegal flat model".

Running example. [Legal flat model describe only data words with finitely many atoms] Consider an infinite word over the alphabet of the atoms

$$w = a_1 a_2 \cdots \in \mathbb{A}^\omega.$$

Such a word is a function from natural numbers to atoms. In this case, legality means that only finitely many different atoms can appear in the word.

Based on the word w, we can define a flat model, denoted by $\underline{w}$, as follows. The universe is the natural numbers, with $<$ is interpreted as the usual order on natural numbers. For every atom $a \in \mathbb{A}$, the interpretation $a^{\underline{w}}$ is the set of positions in w that have label a. It is not difficult to see that $\underline{w}$ is a flat model, which is legal as long as w was legal.

What would happen if we tried to define $\underline{w}$ for a word w with infinitely many different atoms? For instance, what about a word that is an enumeration of all atoms (i.e. every atom appears on exactly one position)? For such a word, every individual predicate interpretation $a^{\underline{w}}$ is a finitely supported set, namely it contains one natural number (the number of the atom a in the enumeration). What is wrong is that the function

$$a \in \mathbb{A} \quad \mapsto \quad a^{\underline{w}} \subseteq \mathbb{N}$$

is not finitely supported. □

First-order logic. To express properties of flat models, we can use standard first-order logic. Formulas can be built out of the predicates in the vocabulary, using standard logical operations $\vee$, $\wedge$, $\neg$, $\exists$ and $\forall$. Semantics in flat models are defined in the standard way, by induction on the structure of formulas. This definition ignores the additional structure of flat models and vocabularies, namely the possibility of acting on both by using atom automorphisms.

Running example. [First-order definable properties of data words] Suppose taht a is an atom, and consider the formula

$$\forall x \exists y \qquad x < y \wedge a(x).$$

When evaluated in a structure of the form $\underline{w}$, as defined in Example 3.1, the formula says that the atom a appears infinitely often. □

A single formula of first-order logic can only use only finitely many predicates from the vocabulary. Therefore, as far as using a single formula of the logic is concerned, it makes little sense to have a flat model specify the interpretations of

all predicates in an infinite vocabulary. The infinite vocabulary *does* make sense when we consider an infinite set of formulas, as we will do in the Section 3.2, or when we consider a variant of first order-logic that admits orbit-finite boolean operations, as we will do starting with Section 4.

3.2 Compactness Fails in Legal Flat Models

We have already identified one problem with legal flat models: they can only model data words with finitely many distinct atoms, which means they are insufficient to model interesting data words. Here we identify another, related, problem: the compactness theorem fails.

Running example. [Compactness Fails for Legal Flat Models.] Let $\varphi_<$ be a formula which says that $<$ is a linear order. Let Γ be the following set of formulas:

$$\{\varphi_<\} \quad \cup \quad \{\exists x\ a(x)\}_{a\in\mathbb{A}} \quad \cup \quad \{\forall x\ a(x) \Rightarrow \neg b(x)\}_{a\neq b\in\mathbb{A}}.$$

When seen as an infinite conjunction of formulas, the set Γ says that every atom appears exactly once as a label of a position. Observe that Γ is infinite, but orbit-finite (three orbits). We will show that every finite subset of Γ is satisfiable in a legal flat model, but Γ itself is not, and therefore compactness fails for legal flat models.

Every finite subset $\Delta \subseteq \Gamma$ is satisfiable in a finite linear order where the positions are labelled by the atoms that appear in Δ.

To prove that the whole set Γ has no legal flat model, imagine for the sake of contradiction that it does have a legal flat model $\mathfrak{A}$. Suppose that S is a finite support of the model. Based on the model, we define a relation $<_{\mathfrak{A}}$ on atoms as follows: $a <_{\mathfrak{A}} b$ if there are elements x, y in the universe of the model $\mathfrak{A}$ such that

$$x \in a^{\mathfrak{A}} \qquad y \in b^{\mathfrak{A}} \qquad x <^{\mathfrak{A}} y.$$

The function $\mathfrak{A} \mapsto <_{\mathfrak{A}}$ is a $\emptyset$-supported function from legal flat models to relations on the atoms, and therefore if $\mathfrak{A}$ is finitely supported then also $<_{\mathfrak{A}}$ is finitely supported. By the properties axiomatised in Γ, the relation $<_{\mathfrak{A}}$ must be a linear order on atoms. But a finitely supported linear order on the (equality) atoms does not exist. Indeed, if S were the support of such an order, then for atoms $a, b \notin S$, the automorphism π which swaps a with b fixes all atoms in S, and witnesses

$$a <_{\mathfrak{A}} b \qquad \text{iff} \qquad b = \pi(a) <_{\mathfrak{A}} a = \pi(b).$$

This proves that Γ has no legal flat model.

Observe that the reasoning above is specific to the equality atoms (because we assumed that there is some automorphism that fixes S and swaps atoms a and b). Indeed, the set Γ does have a legal flat model when the total order atoms are used. The universe is the atoms, ordered by their built-in order, and the interpretation says that each atom is labelled by itself,

$$a^{\mathfrak{A}} = \{a\},$$

which is an $\emptyset$-supported interpretation. The model we have just described is not a data word because the positions are densely ordered. If we extended the formula $\varphi_<$ to also say that every position has a successor, then we would have an example for failure of compactness in the total order atoms, because the total order atoms cannot be ordered in a finitely supported way so that every atom has a successor. □

The failure of compactness is actually related to the weak modelling power of legal flat models. Indeed, as illustrated in the example, the compactness theorem would imply that some legal flat model represents a data word where all positions have different data values.

The statement of the compactness theorem, for which we presented a counterexample above, would be: "if every finite subset of Γ has a legal flat model, then also Γ has a legal flat model." One could imagine a weaker form the theorem, with a stronger assumption: "if every orbit-finite subset of Γ has a legal flat model, then also Γ has a legal flat model." The counterexample above no longer works, because the set Γ, which has no legal flat models, is already orbit-finite itself. The weaker statement is also false.

Running example. [Orbit-finite compactness also fails for legal flat models.] Let φ_{succ} be a formula which says that every element has a $<$-successor:

$$\forall x \exists y \ \big(x < y \wedge \forall z \ (z \leq x \vee z \geq y)\big).$$

For $n \in \mathbb{N}$, define $succ_n(x, y)$ to be the formula which says that y is the n-fold successor of y. For every $n \in \mathbb{N}$, define Γ_n to be the set of formulas, which says that positions at distance n have different labels:

$$\{\forall x \forall y \ succ_n(x, y) \Rightarrow \neg(a(x) \wedge a(y))\}_{a \in \mathbb{A}}.$$

Define Γ to be $\varphi_<$, φ_{succ}, together with the union of all Γ_n. We claim that every orbit-finite subset of Γ has a legal flat model, but Γ itself has no legal flat model.

For an orbit-finite subset $\Delta \subseteq \Gamma$, let n be the biggest number such that

$$\Gamma_n \cap \Delta \neq \emptyset.$$

We define the model for Δ to be the ultimately periodic data word, which uses $n + 1$ atoms $a_1, \ldots, a_{n+1}$ in a periodic arrangement.

To prove that Γ itself has no legal flat model, use a reasoning similar to the one in the previous example. Choose some element x_1 of the universe, and let a_1 be its (unique) label. By following successors, we get infinitely many distinct atoms $a_2, a_3, \ldots$ together with a linear order. □

4 First-Order Logic with Orbit-Finite Boolean Operations

This section, presents another variant of first-order logic, which was introduced in [3]. The motivation behind this logic is two-fold. First, the logic is more expressive. Second, it is a natural variant of first-order logic consistent with the paradigm of replacing "finite" by "orbit-finite". The central question of this paper is finding the right semantics for this richer logic.

Definition of the logic. Since the vocabulary is a set with atoms, it makes sense to apply atom automorphisms to predicates in the vocabulary, and therefore also to formulas built out of these predicates. Therefore it makes sense to say that a set of formulas is orbit-finite. Suppose that we extend first-order logic, as defined previously, by allowing the boolean connectives (disjunction and conjunction) to take range over a (legal) orbit-finite set of formulas. Call the resulting logic *first-order logic with orbit-finite boolean operations*.

Observe that first-order logic with orbit-finite boolean operations can be seen as a fragment of $\mathcal{L}_{\omega_1\omega}$. Recall that $\mathcal{L}_{\omega_1\omega}$ is a variant of first-order logic which allows boolean operations ranging over countable sets of formulas. Since the atoms are assumed to be countable, orbit-finite sets are also countable, and therefore our logic is a special case of $\mathcal{L}_{\omega_1\omega}$. One of the differences is that $\mathcal{L}_{\omega_1\omega}$ has uncountably many formulas, while there are only countably many formulas of first-order logic with orbit-finite boolean operations (under the assumption that the vocabulary is countable).

Example 1. Suppose that the vocabulary has one unary predicate $a(x)$ for every atom a, as in the running example. We write a formula with one free variable x, which holds in positions that satisfy at most one unary predicate of the form $a(x)$.

For an atom a, consider the set of formulas

$$\Gamma_a \stackrel{\text{def}}{=} \{\neg b(x) : b \neq a\}.$$

This set is legal (supported by a), and also orbit-finite (it is included in the single-orbit set $\{a(x) : a \in \mathbb{A}\}$). Therefore, we are allowed to form a new formula by taking the disjunction over Γ_a. We adopt a notational convention where this formula is denoted by

$$\varphi_a \stackrel{\text{def}}{=} \bigwedge_{b \in \mathbb{A} - \{a\}} \neg b(x).$$

Generally speaking, the notational convention says that if I is an orbit-finite set and $i \mapsto \varphi_i$ is a finitely supported function from I to already defined formulas, then $\bigwedge_{i \in I} \varphi_i$ denotes the formula $\bigwedge\{\varphi_i : i \in I\}$.

To the formula φ_a, we can also apply an atom automorphism, by applying the automorphism to every conjunct in parallel. It is not difficult to see that if an atom automorphism maps the atom a to the atom a', then it maps the formula φ_a to the formula $\varphi_{a'}$. It follows that the set

$$\Gamma \stackrel{\text{def}}{=} \{\varphi_a : a \in \mathbb{A}\}$$

is itself orbit-finite, namely it is one orbit under the action of automorphisms. Therefore, we are allowed to form a new formula by taking the disjunction over this set. This formula, according to our notational convention, is denoted by

$$\bigvee_{a \in \mathbb{A}} \bigwedge_{b \in \mathbb{A} - \{a\}} \neg b(x).$$

It says that position x satisfies at most one predicate of the form $a(x)$.

Semantics. As mentioned above, the central question of this paper is about the "right" semantics of first-order logic with orbit-finite boolean operations.

One idea for the semantics is to use flat models. A formula of first-order logic with orbit-finite boolean operations can be evaluated in a flat model, be it legal or not. The definition is standard, and can be seen as a special case of logics with infinitary boolean operations: $\bigvee_{i \in I} \varphi_i$ is true in a flat model (under an appropriate evaluation of the free variables) if and only if φ_i is true for some i; likewise for $\bigwedge$.

Running example. Recall the set of formulas Γ from Example 3.2, which said that the universe is totally ordered, every atom a labels some position, and every position is labelled by at most one atom. In the example, we showed that there is no legal flat model which satisfies all formulas from Γ. Since Γ is orbit-finite, its conjunction $\bigwedge \Gamma$ is a well-formed formula of first-order logic with orbit-finite boolean operations. This formula is not satisfied in any legal flat model. On the other hand, the formula can be satisfied in illegal flat models, e.g. a model of the form $\underline{w}$ where w is an enumeration of all atoms. □

So far, we have two candidates for the semantics: satisfiability in legal flat models, and satisfiability in possibly illegal flat models. The first definition fails the compactness theorem, while the second definition uses notions that are not allowed in sets with atoms (as we shall later see, the second definition does admit the compactness theorem). The rest of this paper presents arguments in favour of the second definition. Section 5, presents a proof system which is sound and complete for the second definition. Section 6 defines stratified models, which are legal objects that satisfy the same formulas as possibly illegal flat models.

Differences with [3]. The syntax of first-order logic with orbit-finite boolean operations and its semantics are the same as in [3], except that [3] made the following restrictions, which are lifted in this paper:

- In [3], all elements in the universe of a relational structure are required to have empty support, which means that if x is an element of the universe and π is an atom automorphism, then $\pi(x) = x$.
- In [3], there is a finite partition $\Sigma = \Sigma_1 \cup \cdots \cup \Sigma_n$ of the vocabulary vocabulary. For every block Σ_i all predicates in Σ_i have the same arity, call it n_i, and a given n_i-tuple can satisfy at most one predicate from Σ_i.

5 A Proof System

Consider sequent calculus, with the only difference being that the rules for Boolean operations are adapted to cover the orbit-finite arity of the operations. The rule for introducing $\vee$ becomes

$$\frac{\Gamma \vdash \varphi_i, \Delta \quad \text{for some } i \in I}{\Gamma \vdash \bigvee_{i \in I} \varphi_i, \Delta} \text{ (introduce } \vee\text{)},$$

while the rule for eliminating $\vee$ requires an orbit-finite set of premises

$$\frac{\Gamma, \varphi_i \vdash \psi, \Delta \quad \text{for all } i \in I}{\Gamma, \bigvee_{i \in I} \vdash \psi, \Delta} \text{ (eliminate } \vee\text{)}.$$

A proof is defined to be a well-founded tree[2], where nodes are labelled by sequents. Every node and its children must be consistent with one of the proof rules in the standard way, with one subtle difference to be described later. The tree together with the labelling by sequents must be finitely supported, i.e. a legal set with atoms. Note that the requirement on the proof tree being finitely supported implies that if a node in the proof tree has an orbit-finite set of children proving different sequents, then the dependence of the sub-proof on the sequent must be finitely supported.

The subtle difference in the definition of proofs concerns the proof rules that require their premise to be true for "some $i \in I$", namely the proof rules "introduce $\vee$" and "eliminate $\wedge$". As opposed to the standard sequent calculus, a proof in our system is allowed to have some redundancy, by proving the premise for more than one $i \in I$. In other words, a more exact wording for the "introduce $\vee$" rule would be

$$\frac{\Gamma \vdash \varphi_i \quad \text{for all } i \text{ in a nonempty subset of } I}{\Gamma \vdash \bigvee_{i \in I} \varphi_i} \text{ (introduce } \vee\text{)}.$$

The need for redundancy is illustrated in the following example.

Example 2. Consider the total order atoms. This example is essentially in propositional logic, since all predicates have arity zero. The vocabulary has one zero-ary predicate $P_{a,b}$ for every pair of distinct atoms $a < b$; the vocabulary has one orbit. We prove the following sequent

$$\bigwedge_{a \in \mathbb{A}} \bigwedge_{b \in \mathbb{A}, b > a} P_{a,b} \quad \vdash \quad \bigwedge_{a \in \mathbb{A}} \bigvee_{b \in \mathbb{A}, b > a} P_{a,b}. \tag{1}$$

We use the rule for introducing $\wedge$, which means that the root of the proof tree for (1) has one child per atom $a \in \mathbb{A}$, containing a proof of the sequent

$$\bigwedge_{a \in \mathbb{A}} \bigwedge_{b \in \mathbb{A}, b > a} P_{a,b} \quad \vdash \quad \bigvee_{b \in \mathbb{A}, b > a} P_{a,b}. \tag{2}$$

Since the proof for (1) must be legal, the proof (2) must depend on a in a finitely supported way. (More precisely, the function that maps a to the proof of (2) must be a finitely supported function from atoms to proofs). This is why we can not choose, for every a, some unique $b \neq a$, because this can not be done in a finitely

[2] Without orbit-finite boolean operations, one can also use proofs which are sequences of sequents, such that each sequent follows from earlier sequents via proof rules. Such proofs will not work with orbit-finite operations, since in some cases they would require imposing a linear order on the atoms.

supported way. (This is why we use total order atoms. In the equality atoms, there is a finitely supported, but not emptily supported, function that maps every atom to some other atom. Namely, choose two atoms b_1, b_2, and then that maps b_1 to b_2 and all other atoms to b_1.) Therefore, the proof of (2) will have one child per $b > a$, containing a proof of

$$\bigwedge_{a \in \mathbb{A}} \bigwedge_{b > a} P_{a,b} \qquad \vdash \qquad P_{a,b}. \tag{3}$$

In particular, the proof for (2) has infinitely many child sub-proofs (although one orbit of them), even though just one would be enough.

The definition of a proof requires the tree to be well-founded; but there might be paths of unbounded length since the tree is not finitely branching (only orbit-finitely branching). One can show that if a sequent has a proof, then it has a proof with an orbit-finite set of nodes, and depth bounded by some natural number.

Completeness and compactness. The following theorem shows that the proof system is sound and complete with respect to semantics in (possibly illegal) flat models.

Theorem 1 (Completeness). *Let Γ be a finitely supported set of sentences of first-order logic with orbit-finite boolean operations, and φ a single sentence. There is a proof of $\Gamma \vdash \varphi$ if and only if φ is satisfied in every (possibly illegal) flat model which satisfies all sentences from Γ.*

The theorem can be proved using the Henkin constant saturation method, to be found in logic textbooks, e.g. [1]. Below we show that the logic also has compactness. Note the difference with $\mathcal{L}_{\omega_1\omega}$, which has completeness (under the assumption that Γ is countable), but does not have compactness (even for countable Γ), see [8].

Corollary 1 (Compactness). *Let Γ be a finitely supported set of sentences of first-order logic with orbit-finite boolean operations. If every orbit-finite subset of Γ is satisfied by a (possibly illegal) flat model, then also Γ is satisfied by a (possibly illegal) flat model.*

Proof (sketch). Toward a contradiction, assume that Γ is not satisfied by any (possibly illegal) flat model. Then, by the right-to-left implication in Theorem 1, there is a proof of $\Gamma \vdash \bot$. As remarked before, this proof can be chosen so that it is orbit-finite, and therefore only uses an orbit-finite number of premises from Γ. By the left-to-right implication in Theorem 1, these premises cannot be satisfied in any (possibly illegal) flat model, a contradiction with the assumption.

Example 3. In Corollary 1, the assumption is that all orbit-finite subsets of Γ are satisfied by (possibly illegal) flat models. The statement is no longer true when this assumption is weakened to finite subsets. Consider a set Γ of formulas which has one formula P_a for every atom, and also the formula $\bigvee_{a \in A} \neg P_a$. Every finite subset of Γ is satisfied by a flat model, even a legal one. The whole set is not satisfied in any (possibly illegal) flat model.

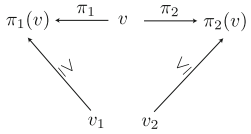
6 Stratified Models

As we have seen in Theorem 1, the "natural" proof system coincides with the semantics in possibly illegal flat models[3]. In this section, we present an alternative semantics, which only uses legal sets with atoms, and which is equivalent to the semantics in possibly illegal flat models, and therefore is equivalent to the proof system[4].

The general idea is to present a possibly illegal flat model by giving a set of finitely supported local versions of the model. A local version does not contain full information about the model, but it does have sufficient information to evaluate formulas with a given support. When one needs to evaluate formulas with a bigger support, then a more precise local view can be taken. The local views need to be consistent with each other; our notion of consistency is modelled using a variant of directed sets, as presented below.

6.1 Ultimately Directed Sets

Suppose that V is a partially ordered legal set with atoms, which means that both the underlying set and the partial order are legal sets with atoms. If S is a set of atoms, then we say that $v \in V$ is *S-greater* than $w \in V$ if $\pi(v) \geq w$ holds for some automorphism π that fixes S. We say that V is *S-directed* if for every two elements of $v_1, v_2 \in V$, some element of $v \in V$ is S-greater than both of them. This definition is illustrated below:



The bigger S becomes, the more difficult it is to be S-directed, because of the requirements on π_1 and π_2. The standard notion of directed set is recovered in the limit, when S is the set of all atoms.

Define $V|_{\geq v}$ to be V restricted to elements that are greater or equal to v. A set V is called *ultimately directed* if for every $v \in V$ and every finite set of atoms S, there is some $w \geq v$ such that $V|_{\geq w}$ is S-directed.

Example 4. Consider the equality atoms. The set $\mathbb{A}^*$ with the prefix ordering is not $\emptyset$-directed, because the words *aab* and *abb* cannot be extended to words in the same orbit. If we restrict $\mathbb{A}^*$ to words where each letter appears exactly once, call this set $\mathbb{A}^{(*)}$, then the set becomes $\emptyset$-directed; it even has a stronger property:

[3] The dual approach would be to find a proof system that coincides with the semantics in legal flat models. It seems that this is the approach taken in [6], although for a different (and intuitionistic) logic, so the exact connection is not clear to the author.

[4] Another idea would be to somehow relax, but not completely lift, the finite support condition in the definition of sets with atoms so that some interesting illegal flat models become legal. Such an approach is pursued in [7].

every two elements are comparable up to atom automorphisms. Furthermore, if $a_1 \cdots a_n$ is a finite word in $\mathbb{A}^{(*)}$, then $\mathbb{A}^{(*)}|_{\geq a_1 \cdots a_n}$ is $\{a_1, \ldots, a_n\}$-directed. It follows that $\mathbb{A}^{(*)}$ is ultimately directed.

Example 5. This example generalises the previous one to arbitrary atom structures, e.g. the total order atoms. Choose some non-repeating enumeration of the atoms $a_1 a_2 \cdots$. We claim that

$$V = \{\pi(a_1 \cdots a_n) : n \in \mathbb{N}, \pi \text{ is an atom automorphism.}\},$$

equipped with the prefix order, is ultimately directed. (Although the enumeration of atoms is not a legal object, the set V that it yields is.) It is not difficult to see that for every $b_1 \cdots b_n \in V$, the set $V|_{\geq b_1 \cdots b_n}$ is S-directed if and only if $S \subseteq \{b_1, \ldots, b_n\}$. This implies that V itself is ultimately directed.

Actually, we did not even use the property that $a_1 a_2 \cdots$ was an enumeration of all atoms, it could even have been an enumeration of some infinite subset of the atoms. However, when the enumeration does use all the atoms, it has the following property: for every $v \in V$ and every finite set of atoms S, there is some node $w \geq v$ such w supports all the atoms in S (i.e. an automorphism of the atoms that fixes w must also fix all atoms in S). This additional property will be useful in some constructions.

A disadvantage of the previous example was that it required some enumeration of the atoms, which is an illegal object. This can be avoided by using a hack called a fake atom structure. A *fake atom structure* is defined to be an isomorphic copy of the atom structure, but where all elements of the universe have empty support. In particular, even infinite subsets of fake atom structure are legal.

Example 6. Consider the equality atoms. Recall the standard set-theoretic encoding of natural numbers, where for instance 2 is encoded by $\{\emptyset, \{\emptyset\}\}$. Each natural number is built without using atoms, and therefore the relational structure $(\mathbb{N}, =)$ is an example of a fake atom structure for the equality atoms: it is constructed inside legal sets with atoms, does not use atoms, but is isomorphic to the atom structure (a countable set of abstract atoms with equality only). Note that any isomorphism of this fake atom structure with the real atom structure would itself be an illegal object, since it would in particular induce an order on the atoms.

Example 7. Choose some fake atom structure, and let V be the set of isomorphisms between finite substructures of the fake atoms, and finite substructures of the atoms, ordered by extension. One can think of a partial isomorphism $f \in V$ as a finite tuple of atoms, indexed by finitely many many elements from the universe of the fake atom structure.

Claim. For every $f \in V$ with range S, $V|_{\geq f}$ is S-directed.

The claim implies that V is ultimately directed. This is because for every finite set of atoms S, every element of V can be extended so that its range includes S.

6.2 Stratified Models

We now propose our definition of models which will turn out to be equivalent to (possibly illegal) flat models. Let S be a finite set of atoms. An *S-stratified model* over a vocabulary Σ consists of

- **The frame.** An partially ordered set with atoms V which is called the *frame*. The frame must be countable, S-directed and ultimately directed, and have a least element called the *root*. Elements of the frame are called nodes.
- **The local models.** A labelling which maps each node v of the frame to a legal flat model $\mathfrak{A}(v)$, called the *local model in* v, whose vocabulary is a subset of Σ. This labelling must satisfy the following conditions:
  1. The labelling function $v \mapsto \mathfrak{A}(v)$ is *weakly S-supported*, which means that it can be extended to some S-supported function.
  2. If $w \geq v$, then the local model in w extends the local model in v in the following sense. The universe of $\mathfrak{A}(v)$ is a subset of the universe of $\mathfrak{A}(w)$, and the interpretation of $\mathfrak{A}(v)$ is obtained from the interpretation of $\mathfrak{A}(w)$ by restricting to the vocabulary and universe of $\mathfrak{A}(v)$.
  3. For every predicate R in Σ there is a node $w \geq v$ such that R is in the vocabulary of the local model in w.

If v is a node in the frame of an S-stratified model $\mathfrak{A}$, then the *submodel* in node v, denoted by $\mathfrak{A}|_{\geq v}$, is defined by restricting the frame to nodes greater or equal to v, and keeping the same labelling by local models. Submodels of an S-stratified model are also S-stratified.

Semantics in stratified models. We now define the semantics of first-order logic with orbit-finite boolean operations in stratified models. Suppose that $\mathfrak{A}$ is an S-stratified model and φ is an S-supported formula. If φ has free variables, then we also need a valuation ν, which maps the free variables of the formula to S-supported elements in the universe of the local model in the root of $\mathfrak{A}$. Given these ingredients, we can define the truth value, which is denoted by a triple turnstile

$$\mathfrak{A}, \nu \models\!\!\!\models \varphi.$$

We underline that the above is only defined when both the formula φ and the valuation ν are supported by the parameter S such that $\mathfrak{A}$ is an S-stratified model. The definition is by induction on the structure of φ.

- For a predicate formula, define

$$\mathfrak{A}, \nu \models\!\!\!\models R(x_1, \ldots, x_n) \tag{4}$$

  to be true if there is some node w in the frame such that the local model in w has R in its vocabulary and $R(x_1, \ldots, x_n)$ holds in that local model, under the valuation ν. Using the assumptions that the frame is S-directed, and that the labelling by local models is weakly S-supported, one can show that this definition does not depend on the choice of v, i.e. if it $R(x_1, \ldots, x_n)$ holds in some local model whose vocabulary contains R, then it holds in all local models whose vocabulary contains R.

- Negation is defined in the standard way.
- For an orbit-finite disjunction, define

$$\mathfrak{A}, \nu \models \bigvee_{i \in I} \varphi_i$$

to be true if there is some finite set of atoms $T \supseteq S$, a submodel $\mathfrak{B}$ of $\mathfrak{A}$ that is T-stratified, and some $i \in I$ such that φ_i is T-supported and

$$\mathfrak{B}, \nu \models \varphi_i.$$

The definition for conjunction is obtained by using De Morgan's law.

- For a existential quantification, we define

$$\mathfrak{A}, \nu \models \exists x.\varphi(x)$$

to be true if there is some finite set of atoms $T \supseteq S$, a submodel $\mathfrak{B}$ of $\mathfrak{A}$ that is T-stratified, and some T-supported element a in the universe of the local model in the root of $\mathfrak{B}$ such that

$$\mathfrak{B}, \nu[x \to a] \models \varphi(x)$$

The definition for $\forall$ is obtained by using De Morgan's law.

Running example.[A stratified model describing a data word] We will describe a $\emptyset$-stratified model, call it $\mathfrak{A}$, that describes a data word (for equality atoms) where all letters are different. The frame is non-repeating sequences of atoms, which was shown in Example 4 to be ultimately directed. In a node $a_0 \cdots a_n$ of the frame, the local model looks as follows. The universe is always the natural numbers, regardless of the node (the universe is not built using atoms, i.e. all of its elements have empty support). The vocabulary contains the order predicate $<$, regardless of the node, which is interpreted in the standard way. Furthermore, the vocabulary contains the predicates $a_0(x), \ldots, a_n(x)$, which are interpreted so that predicate $a_i(x)$ holds exactly in position i, and nowhere else. In particular, each local model has only finitely many positions satisfying the unary predicates.

Recall the property "$<$ is a linear order and every atom appears in exactly one position", which was studied previously in the running example, and which is expressed by the sentence

$$\varphi_< \quad \wedge \quad \bigwedge_{a \in \mathbb{A}} \exists x\ a(x) \quad \wedge \quad \bigwedge_{a \neq b \in \mathbb{A}} \forall x\ a(x) \Rightarrow \neg b(x).$$

This property was false in all legal flat models, but it is true in the stratified model $\mathfrak{A}$ we have just described. To illustrate the semantics, we show that

$$\mathfrak{A} \models \bigwedge_{a \in \mathbb{A}} \exists x\ a(x).$$

(There are no free variables, so there is no valuation.) Recall that $\models$ is not defined for any choice of model and formula: the formula needs to be supported by a set S such that the model is S-stratified. In this case, the formula has empty support and the model is $\emptyset$-stratified, so $\models$ is defined. By unravelling the semantics of $\bigwedge$, we need to show that for every atom $a \in \mathbb{A}$, and every node $a_1 \cdots a_n$ in the model, if the submodel $\mathfrak{A}|_{\geq a_0 \cdots a_n}$ is S-stratified for some $S \ni a$, then that submodel satisfies $\exists x\ a(x)$. It is not difficult to see that the submodel in node $a_0 \cdots a_n$ is S-stratified for some $S \ni a$ if and only if $a \in \{a_0 \cdots a_n\}$. Such submodels satisfy $\exists x\ a(x)$. Observe that even when a is fixed, then the position x that satisfies $a(x)$ will depend on the choice of the node $a_0 \cdots a_n$. For instance, if the node is chosen to be abc then x will be the position zero, and if the node is chosen to be cba then x will be position two. □

Running example.[An "isomorphic" model] Choose some bijection f of the natural numbers. Consider a $\emptyset$-stratified model that is defined the same way as in the previous example, with the difference that in node $a_0 \cdots a_n$, the predicate $a_i(x)$ holds exactly in position $f(i)$. One can show that this model satisfies the same sentences as the previous one. □

Stratified models are equivalent to (possibly illegal) flat models. The following theorem shows stratified models are equivalent to (possibly illegal) flat models that (possibly illegal) flat models, at least when restricted to uniformly supported sets of sentences. A set is called *uniformly supported* if there is some support S that supports all elements of the set. Every finite set of sentences is uniformly supported.

Theorem 2. *Let Γ be a uniformly finitely supported set of sentences of first-order logic with orbit-finite boolean operations. Then Γ has a (possibly illegal) flat model if and only if it has a stratified model.*

The following proposition shows the left-to-right implication in Theorem 2.

Proposition 1. *Let S be a finite support.*

1. *For every (possibly illegal) flat model there is an S-stratified model which satisfies the same S-supported sentences.*
2. *For every S-stratified model there is a (possibly illegal) flat model which satisfies the same S-supported sentences.*

In the proof of the first item in the proposition, we construct a stratified model satisfying the following properties: all local models have the same universe; all elements of this universe have empty support; the frame is obtained from Example 7. Therefore, Theorem 2 would still be true under a definition of stratified model that would require these properties.

Acknowledgement. I would like to thank Nathanaël Fijalkow, Bartosz Klin, Sawomir Lasota, and especially Szymon Toruńczyk for inspiring discussions.

References

1. Adamowicz, Z., Zbierski, P.: Logic of Mathematics: A Modern Course of Classical Logic. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley (2011)
2. Bojanczyk, M., Klin, B., Lasota, S.: Automata with group actions. In: LICS, pp. 355–364 (2011)
3. Bojańczyk, M., Place, T.: Toward model theory with data values. In: Czumaj, A., Mehlhorn, K., Pitts, A., Wattenhofer, R. (eds.) ICALP 2012, Part II. LNCS, vol. 7392, pp. 116–127. Springer, Heidelberg (2012)
4. Bojanczyk, M., Torunczyk, S.: Imperative programming in sets with atoms. In: D'Souza, D., Kavitha, T., Radhakrishnan, J. (eds.) FSTTCS. LIPIcs, vol. 18, pp. 4–15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2012)
5. Gabbay, M.J., Pitts, A.M.: A new approach to abstract syntax with variable binding. Formal Asp. Comput. 13(3-5), 341–363 (2002)
6. Gabbay, M.J.: Fresh logic: proof-theory and semantics for fm and nominal techniques. J. Applied Logic 5(2), 356–387 (2007)
7. Gabbay, M.J.: Finite and infinite support in nominal algebra and logic: nominal completeness theorems for free. J. Symb. Log. 77(3), 828–852 (2012)
8. Lopez-Escobar, E.G.K.: An interpolation theorem for denumerably long formulas. Fundam. Math. 57, 253–272 (1965)
9. Pitts, A.M.: Nominal Sets: Names and Symmetry in Computer Science. Cambridge Tracts in Theoretical Computer Science, vol. 57. Cambridge University Press (2013)

Counting in SPARQL Property Paths: Perspectives from Theory and Practice

Wim Martens

Institute for Computer Science, University of Bayreuth
wim.martens@uni-bayreuth.de

Abstract. RDF and SPARQL are becoming increasingly popular and are bringing many new and interesting research challenges. During the development of these standards, the World Wide Web Consortium (W3C) does not necessarily always have all the cards on the table in order to make perfectly informed design decisions and therefore it partly relies on input from the research community. This is a very interesting situation for researchers since it can give the opportunity to immediately have research results incorporated into practice. In this talk I will discuss some experiences from our interaction with the W3C concerning the semantics of property paths in SPARQL. Property paths are a relatively new feature in SPARQL 1.1 and essentially correspond to regular expressions that should be evaluated over RDF graphs.

The popularity of the Resource Description Framework (RDF) [5] and the SPARQL Protocol and RDF Query Language (SPARQL) [3] is posing new challenges for computer science researchers. In brief, the RDF data model stores data as triples of the form (x, p, y) where, intuitively, x corresponds to a subject, y to an object, and p to a predicate that denotes a relationship between x and y. As such, a collection of RDF triples can be seen as a graph in which the nodes are the subjects and objects of triples in the collection and there is an edge labelled p from node x to node y if and only if there is a triple (x, p, y) in the collection [4].

The SPARQL query language [3] represents a serious effort by the World Wide Web Consortium in its quest for a query language for RDF that bears a good tradeoff between coverage of use cases, ease-of-use, desirable features, expressivity, and complexity. However, at the moment where design decisions need to be made, the designers of SPARQL do not necessarily have all the cards on the table, e.g., regarding tradeoffs between expressiveness and complexity. Therefore, there are many aspects of the SPARQL language for which the World Wide Web Consortium relies on input from the research community in order to make well-informed decisions.

This situation can be win-win scenario for researchers and developers of the language. The World Wide Web Consortium is open for comments and input from the research community, which can in turn influence the further development of the standard. We will discuss this interaction between theory and practice with a focus on our own experience with some aspects of the SPARQL 1.1 recommendation, more precisely, aspects of SPARQL property

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 29–30, 2013.

paths [1,6]. SPARQL property paths essentially correspond to regular expressions that should be evaluated against paths in an RDF graph. As such, there is a very close connection between navigational queries on graph databases (see [2] for a recent overview) and current developments in the SPARQL language.

The word "counting" will be used with two different meanings. The first refers to the multiset semantics that property paths currently have. The second refers to counting operators that can make property paths exponentially more succinct. We discuss recent findings regarding the impact of counting on the complexity of evaluating property paths and how the development of SPARQL reacted to these.

References

1. Arenas, M., Conca, S., Pérez, J.: Counting beyond a yottabyte, or how SPARQL 1.1 property paths will prevent adoption of the standard. In: World Wide Web Conference (WWW), pp. 629–638 (2012)
2. Barcelo, P.: Querying graph databases. In: Symposium on Principles of Database Systems (PODS), Invited tutorial (to appear, 2013)
3. Harris, S., Seaborne, A., Prud'hommeaux, E.: SPARQL 1.1 query language. Technical report, W3C (January 2012), http://www.w3.org/TR/2012/WD-sparql11-query-20120105/
4. Hayes, P., McBride, B.: RDF semantics. Technical report, W3C (February 2004), http://www.w3.org/TR/2004/REC-rdf-mt-20040210/
5. Klyne, G., Carroll, J.J., McBride, B.: RDF 1.1 concepts and abstract syntax. Technical report, W3C (January 2013), http://www.w3.org/TR/2013/WD-rdf11-concepts-20130115/
6. Losemann, K., Martens, W.: The complexity of evaluating path expressions in SPARQL. In: Symposium on Principles of Database Systems (PODS), pp. 101–112 (2012)

Quantitative Approaches to Information Protection

Catuscia Palamidessi

INRIA and Ecole Polytechnique, France

Abstract. Secure information flow is the problem of ensuring that the information made publicly available by a computational system does not leak information that should be kept secret. Since it is practically impossible to avoid leakage entirely, in recent years there has been a growing interest in considering the quantitative aspects of information flow, in order to measure and compare the amount of leakage.

In this talk, we revise the main recent approaches which have been proposed to quantify and reason about leakage, the information-theoretic approaches based on Shannon entropy and on Rényi min-entropy, and a novel one based on decision theory. Furthermore, we study the relation with differential privacy, a notion which has been proposed to cope with the problem of *statistical disclosure control* in the area of databases. Finally, we consider a generalization of differential privacy, which can be naturally applied to protect sensitive information also in domains other than databases.

The talk is based on the papers [1,2,3].

References

1. Alvim, M.S., Andrés, M.E., Chatzikokolakis, K., Palamidessi, C.: On the relation between Differential Privacy and Quantitative Information Flow. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part II. LNCS, vol. 6756, pp. 60–76. Springer, Heidelberg (2011)
2. Alvim, M.S., Chatzikokolakis, K., Palamidessi, C., Smith, G.: Measuring information leakage using generalized gain functions. In: Proceedings of the 25th IEEE Computer Security Foundations Symposium (CSF), pp. 265–279 (2012)
3. Chatzikokolakis, K., Andrés, M.E., Bordenabe, N.E., Palamidessi, C.: Broadening the scope of Differential Privacy using metrics. In: Proc. of PETS. IEEE (to appear, 2013), Technical report available at: `http://hal.inria.fr/hal-00767210`
4. Chatzikokolakis, K., Palamidessi, C., Panangaden, P.: Anonymity protocols as noisy channels. Information and Computation 206(2-4), 378–401 (2008)
5. Clark, D., Hunt, S., Malacaria, P.: Quantified interference for a while language. In: Proceedings of the Second Workshop on Quantitative Aspects of Programming Languages (QAPL 2004). Electronic Notes in Theoretical Computer Science, vol. 112, pp. 149–166. Elsevier Science B.V. (2005)
6. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
7. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: Privacy via distributed noise generation. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 486–503. Springer, Heidelberg (2006)

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 31–32, 2013.

8. Köpf, B., Basin, D.A.: An information-theoretic model for adaptive side-channel attacks. In: Ning, P., De Capitani di Vimercati, S., Syverson, P.F. (eds.) Proceedings of the 2007 ACM Conference on Computer and Communications Security (CCS 2007), pp. 286–296. ACM (2007)
9. Smith, G.: On the foundations of quantitative information flow. In: de Alfaro, L. (ed.) FOSSACS 2009. LNCS, vol. 5504, pp. 288–302. Springer, Heidelberg (2009)

Perspectives of Dynamic Complexity

Thomas Schwentick*

Technische Universität Dortmund

Abstract. Many current data processing scenarios deal with about large collections of permanently changing data. In this context, it is often outright impossible to compute the answer for a query from scratch. Rather some auxiliary data needs to be stored that helps answering queries quickly, but also requires to be maintained incrementally. This incremental maintenance scenario can be studied in various ways, e.g., from the perspective of dynamic algorithms with the goal to reduce (re-) computation time. Other options are to study the scenario from the perspective of low-level parallel computational complexity [3] or parallelizable database queries [1]. As the "lowest" complexity class AC^0 (with a suitable uniformity condition) and the core of the standard database query language SQL both coincide with first-order predicate logic, one naturally arrives at the question which queries can be answered/maintained dynamically with first-order predicate logic (DynFO).

The most intensily studied query in this dynamic setting is the reachability query on graphs, arguably the "simplest recursive" query. It has been shown that it can be maintained in DynFO on undirected [3] or acyclic directed graphs [1]. However, whether it can be maintained on general directed graphs is considered the main open question of the field.

Actually, it turned out that showing that a given query can *not* be maintained in DynFO is a very challenging problem, for which currently no methods are available. Furthermore, even though AC^0 is a small complexity class in the static setting, first-order logic is already quite powerful in the dynamic world. These two observations have recently led to the study of fragments of DynFO, e.g., by restricting or forbidding quantification, with the idea to start developing inexpressibility tools there. A surprising result found along these lines is that on strings, quantifier free predicate logic can *exactly* maintain the regular languages [2]. The talk will give an introduction into dynamic complexity, survey some of its most important results, and report about recent work on fragments of DynFO.

References

1. Dong, G., Su, J.: Incremental and decremental evaluation of transitive closure by first-order queries. Inf. Comput. 120(1), 101–106 (1995)
2. Gelade, W., Marquardt, M., Schwentick, T.: The dynamic complexity of formal languages. ACM Trans. Comput. Log. 13(3), 19 (2012)
3. Patnaik, S., Immerman, N.: Dyn-fo: A parallel, dynamic complexity class. J. Comput. Syst. Sci. 55(2), 199–209 (1997)

* This work was supported by the DFG Grant SCHW678/36-1.

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, p. 33, 2013.

Linear Time Proof Verification on N-Graphs: A Graph Theoretic Approach

Laís Andrade, Ruan Carvalho, Anjolina de Oliveira, and Ruy de Queiroz

Centro de Informática, Universidade Federal de Pernambuco,
50740-560 Recife, Pernambuco, Brazil
{lsa,rvbc,ago,ruy}@cin.ufpe.br

Abstract. This paper presents a linear time algorithm for proof verification on N-Graphs. This system, introduced by de Oliveira, incorporates the geometrical techniques from the theory of proof-nets to present a multiple-conclusion calculus for classical propositional logic. The soundness criterion is based on the one given by Danos and Regnier for Linear Logic. We use a DFS-like search to check the validity of the cycles in a proof graph, and some properties from trees to check the connectivity of every switching (a concept similar to D-R graph). Since the soundness criterion in proof graphs is analogous to Danos-Regnier's procedure, the algorithm can also be extended to check proofs in the multiplicative linear logic without units (MLL^-) with linear time complexity.

Keywords: automatic proof-checking, natural deduction, classical logic, linear logic, MLL^-, N-Graphs, graph theory, dfs.

1 Introduction

The pioneering work of R. Statman in his doctoral thesis Structural Complexity of Proofs [15] showed that extracting structural properties of proofs in natural deduction (ND) using appropriate geometric intuitions offers itself as a very promising approach to the study of formal proofs. However, the lack of symmetry in ND presents a challenge for such a kind of study. The obvious alternative, of course, is to look at multiple-conclusion calculi. One already has in the literature different approaches involving such calculi, such as, for example, Kneale's tables of development [9] (studied in depth by Shoesmith & Smiley [14]) and Ungar's multiple-conclusion ND [16]. More recently, the development of a proof system based on "natural deduction graphs", N-Graphs for short, developed by A. de Oliveira [11,12], has brought an interesting ingredient, namely the combination of the intuitive appeal of Natural Deduction with the structural tools and the built-in symmetry of Sequent Calculus.

We are here concerned with the proof-checking in the N-Graphs system. In particular, we present a linear time algorithm for proof verification on N-Graphs. Unlike regular tree like formalisms such as sequent calculus or ND, N-Graphs, as proof-nets for linear logic, have a notion of correction based on a global criterion

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 34–48, 2013.

expressed in terms of acyclicity and connectedness of a class of subgraphs [6]. For this reason, checking their correctness is not a trivial task.

The proposed algorithm can also be applied to check proofs in MLL^- with linear time complexity. Although linear algorithms already exist for checking MLL^- proof nets [7,10], our approach seems to be simpler than both. We use a Depth-First-Search-like procedure to check the validity of the cycles in a proof graph, and some properties from trees to check the connectivity of every switching (a concept similar to D-R graph).

2 N-Graphs

N-Graphs is a multiple conclusion proof system for the propositional classical calculus where proofs are built in the form of directed graphs (*"digraphs"*). Proposed by de Oliveira [11,12], it is a symmetric natural deduction (ND) calculus with the presence of structural rules, similar to the sequent calculus.

The system incorporates ideas from different works. The derivations are based on symmetric ND systems defined by Kneale [9] and Ungar [16]. The implication connective ("$\rightarrow$") is handled by a special edge that captures the discharge of assumptions, similarly to Statman's approach. Furthermore, it adopts the notion of optical graphs from Carbone and embodies the geometrical techniques coming from the theory of proof-nets [6] to define soundness criteria.

Several studies have been developed on N-Graphs since its first publication in 2001, like Alves' development on the geometric perspective and cycle treatment towards the normalization of the system [1] and Cruz's definition of intuitionistic N-Graphs [5]. A normalization algorithm was presented for classical N-Graphs [3], along with the subformula and separation properties [2]. This work aims at a systematic analysis of the soundness criteria, based on the pure structure of the graph.

2.1 Proof-Graphs

Proofs are represented by digraphs. The vertices are labeled with formula-occurrences and some receive special names [11,12]:

Definition 1. (i) *A* branch point *in a digraph is a vertex with at least three edges attached to it.* (ii) *A* focussing *(*defocussing*)* branch point *is a vertex in a digraph with two edges oriented towards (resp. away from) it.*

The edges in a proof-graph define links, which represent the atomic steps in a derivation. There are three kinds of links, as shown in Fig. 1: focussing, defocussing and simple links.

Definition 2. (i) *A* focussing link *is a set* $\{(u_1, v), (u_2, v)\}$ *in a digraph in which* v *is a focussing branch point, as illustrated by Fig. 1. The vertices* u_1 *and* u_2 *are called* premises *of the link and* v *is the* conclusion. (ii) *A* defocussing link *is a set* $\{(u, v_1), (u, v_2)\}$ *in a digraph in which* u *is a defocussing branch point, as*

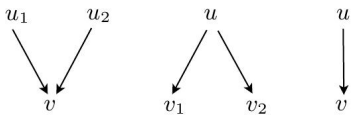


Fig. 1. Links in a proof-graph

illustrated by Fig. 1. The vertices v_1 *and* v_2 *are called* conclusions *of the link and* u *is the* premise. (iii) *A* simple link *is an edge* (u, v) *in a digraph which neither belongs to focussing nor to a defocussing link, as illustrated by Fig. 1. Vertex* u *is called* premise *of the link and* v *is* conclusion.

Definition 3 (proof-graph). *A* proof-graph *is a connected digraph defined as follows:*

1. *each vertex is labeled with a formula-occurrence;*
2. *there are two kinds of edges (*"solid" *and* "meta"*) and the second one are labeled with an* "m" *(*$(u, v)^m$*);*
3. *there are three kinds of links (*simple, focussing *and* defocussing*), divided into* logic *(Fig. 2) and* structural *(Fig. 3) ones;*
4. *each vertex is labeled with a conclusion of a unique link and is a premise of at most one link.*

A *logical link* represents a derivation in ND ($\top - link$ acts as the law of the excluded middle). A *structural link* expresses the application of a structural rule as it is done in sequent calculus: it enables weakening a proof, duplicating premises

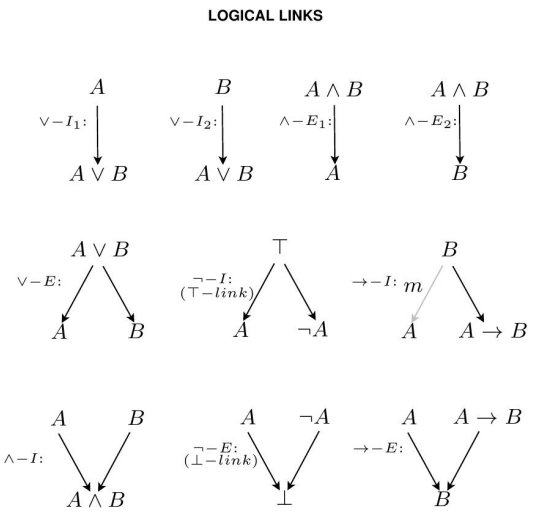


Fig. 2. Logical links

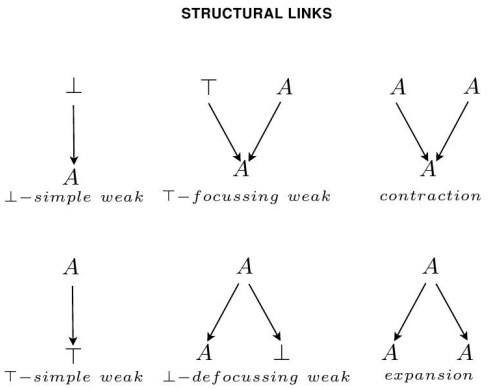


Fig. 3. Structural links

(expansion link) and grouping conclusions in equivalence classes (contraction link).

There is no link to emulate the *interchange* rule because the order of the premises is not important for the application of derivation rules. The axioms are represented by proof graphs with one vertex and no edges.

The focussing and defocussing links may also be classified according to their semantic:

Definition 4. (i) *The links* $\wedge - I$, $\perp - link$, $\rightarrow - E$, $\top - focussing\ weak$ *and* expansion *are called* conjunctive. (ii) *The links* $\vee - E$, $\top - link$, $\perp - defocussing\ weak$ *and* contraction *are called* disjunctive.

Other relevant concepts for a given proof-graph G:

1. the *solid* indegree (outdegree) of a vertex v is the number of solid edges oriented towards (away from) it. The *meta* indegree and outdegree are defined analogously;
2. the set of vertices with indegree equal to zero is the set of premises of G ($PREMISS(G)$);
3. the set of vertices with outdegree equal to zero is the set of conclusions of G ($CONC(G)$);
4. the set of vertices with solid indegree equal to zero and meta indegree equal to one is the set of canceled hypothesis of G ($HYPOT(G)$).

2.2 Inadequacy

In a multiple conclusion calculus we have rules with more than one conclusion ($\vee - elimination$) as well as rules with more than one premise ($\wedge - introduction$). This allows the existence of cycles and makes soundness difficult to prove. In the tables of development, Kneale avoids this problem stating that formulas which are already connected must not be connected again in any way [9]. However,

without the presence of cycles a new problem arises: some tautologies cannot be proved [16], and so the system becomes incomplete.

Shoesmith & Smiley [14] and Ungar [16] solved the inadequacy by defining operations on derivations. With them, the links no longer correspond to a deductive step. Sometimes they refer to a combination of derivations. A simpler approach is the one used by the theory of proof-nets. It allows every link to represent a single deductive step, since it represents a logical relationship between formulas. The characterization of which proof-structures are correct is based on a geometrical concept, and so the cycles are permitted in a controlled way.

As a multiple conclusion calculus, N-Graphs also faced the inadequacy problem. The proposed solution [11,12] adopts some ideas from Shoesmith & Smiley and Ungar for classical logic and the simplicity of Danos & Regnier's solution for MLL^-. The main idea is to distinguish a link that represents a logical derivation from the one that deals with the proof structure. The second kind of links is represented by contraction and expansion links. A proof may contain cycles [1], which shall be controlled by these two structural links.

Expansion and Contraction Links. A cycle in Kneale's tables of development occurs when a conjunction is made on terms generated by the same disjunction (Fig. 4). A valid cycle happens when a disjunction yields the same formula twice and they are contracted to a single occurrence. The latter is similar to the $\vee -$ *elimination* rule in ND.

To handle cycles de Oliveira proposes grouping conclusions into equivalence classes using the contraction link. In the sequent calculus it is done by right contractions. Though this link is focussing, its semantics is disjunctive. In Fig. 4, at the rightmost proof-graph, the cycle involving $A \vee A$ is opened by a disjunctive link and closed by another one (contraction) and therefore is valid.

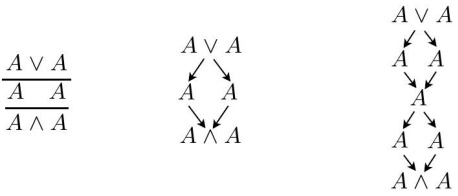


Fig. 4. Proof-graphs for $A \vee A \vdash A \wedge A$: an inadequate on the left and a sound one on the right

In order to group assumptions similarly to single conclusion calculi we use the expansion link. For example, to allow the conjunction of a formula with itself ($A \wedge A$), we need two instances of A, which are joined by an expansion. It corresponds to left contraction in the sequent calculus and allow us to complete the proof of $A \vee A \vdash A \wedge A$. As the contraction link, the semantic of expansion is

[1] Sometimes we talk about "cycles" when in fact "semicycles" are meant (i.e. the direction of edges is not relevant).

contrary to its geometry (conjunctive defocussing). Thus the cycle that includes $A \wedge A$ in Fig. 4 is valid because it is opened by a conjunctive link (expansion) and closed by another one.

Meta-Edge and the Scope of the Hypothesis. Besides expansion and contraction links there is the $\rightarrow -I$ link. In the Kneale's tables of development there is no rule for discarding premises. Both Ungar and Gentzen systems are formulated in such a way that when the $\rightarrow$ connective is introduced, it may eliminate an arbitrary number of premises (including zero). In N-Graphs this introduction is made in a more controlled way, which also complicates the task of identifying inadequate proof-graphs. For example, the first proof in Fig. 5 is not correct, but the second one is.

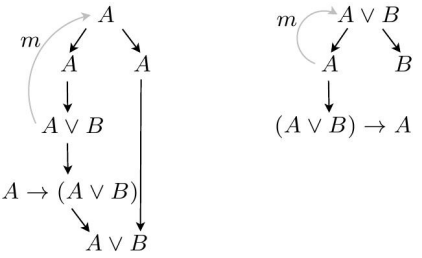


Fig. 5. Meta edge: an invalid application on the left for $\vdash A \rightarrow (A \vee B)$ and a sound one on the right for $\vdash (A \vee B) \rightarrow A, B$

2.3 Soundness Criteria

Similar to Danos-Regnier criterion [6], we define the following subgraphs associated to a proof-graph.

Definition 5. *Given a proof-graph G,* (i) *a* switching graph $S(G)$ *associated with G is a spanning subgraph*[2] *of G in which the following edges are removed: one of the two edges of every expansion link, one of the two edges of every contraction link and all meta edges;* (ii) *a* switching expansion $S_e(G)$ *associated with G is a spanning subgraph of G in which one of the two edges of every expansion link and all meta edges are removed.* (iii) *We say that the* meta-condition *holds for G iff for every meta-edge $(u, v)^m$ of a defocussing link $\rightarrow -I$ $\{(u, w), (u, v)^m\}$ in G, there is a path or semipath from v to u without passing through (u, w) in every switching expansion $S_e(G)$ and the solid indegree of v is equal to zero.*

Definition 6 (N-Graph derivation). *A proof-graph G is a* N-Graph derivation *(or* N-Graph *for short) iff the meta-condition holds for G and every switching graph associated with G is acyclic and connected.*

[2] A spanning subgraph is a subgraph G_1 of G containing all the vertices of G.

The soundness and completeness of the system were proved through a mapping between N-Graphs and LK [11,12]:

Theorem 1 (map to N-Graph). *Given a derivation Π of $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ in the classical sequent calculus, it is possible to build a corresponding N-Graph $NG(\Pi)$ whose elements of $PREMIS(NG(\Pi))$ and $CONC(NG(\Pi))$ are in one-to-one correspondence with the occurrences of formulas $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$, respectively.*

Theorem 2 (sequentialization). *Given a N-Graph derivation G, there is a sequent calculus derivation $SC(G)$ of $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ in the classical sequent calculus whose occurrences of formulas $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ are in one-to-one correspondence with the elements of $PREMIS(G)$ and $CONC(G)$, respectively.*

Analysis of the Fragment $\wedge$, $\vee$ and $\neg$. If a proof-graph does not have any $\rightarrow -I$ link, then meta-condition holds for it. In the next section we define a method to check soundness of derivations in the $\{\vee, \wedge, \neg\}$ fragment of N-Graphs, where the meta-condition is already satisfied.

3 Verification of Proof-Graphs

Given a proof-graph with no meta edge, the goal of the verification algorithm is to check if every switching graph associated with it is acyclic and connected. Danos–Regnier's soundness criterion verifies the same, but on proof-structures instead. The algorithm presented here is based on the types of cycles that may exist in a proof-graph.

We propose a validation mechanism over a given cycle that determines if there is a switching where it appears. If no such cycle is found in a proof-graph, then every associated switching is acyclic. On a second step, we use a relation between the number of edges and vertices in the switching graphs to confirm its connectivity. This relationship was already found for *D-R* graphs [7].

3.1 Cycle Analysis

A valid cycle in a proof-graph is the one that is not present in any switching graph associated with it. The only way a cycle may "disappear" from a graph is by removing one of its edges. The edges which may be not present in a given switching are the contraction, expansion and meta edges. They are the key for the identification of valid cycles and here will be denoted *volatile edges*.

The volatile edges from a single expansion/contraction link have a strong semantic connection while working with switchings. If one of then is present in a given switching graph $S(G)$, the other one is not. Because of this connection, two edges from the same link will be denoted *conjugated edges*:

Definition 7 (conjugated edge). *Let e_1 and e_2 be two edges from an expansion or contraction link in a proof-graph G. The* conjugated edge *of e_1 is $\hat{e}_1 = e_2$, and vice versa, and $e \in S(G) \iff \hat{e} \notin S(G)$ for all switching $S(G)$.*

Let c be a cycle in a proof-graph G. $E_V(c)$ is defined as the set of volatile edges of c. The cycles in a proof-graph may be classified into the following types (also illustrated in Fig. 6):

1. cycles with no volatile edge;
2. cycles c with a non-empty set of volatile edges $E_V(c) = \{e_1, \ldots, e_n\}, n > 0$, where $\forall\ e_i\ \in E_V(c)\ \rightarrow\ \hat{e}_i\ \notin\ E_V(c)$;
3. cycles c with a non-empty set of volatile edges $E_V(c) = \{e_1, \ldots, e_n\}, n > 0$, where $\exists\ e_i\ \in E_V(c)\ \rightarrow\ \hat{e}_i\ \in\ E_V(c)$;

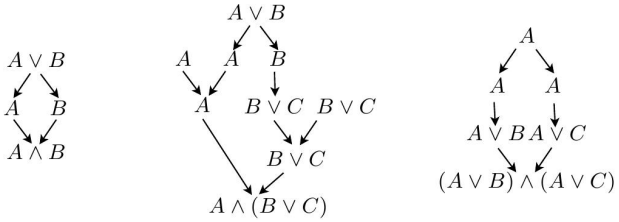


Fig. 6. Cycles in a proof-graph

Analyzing each of these cases separately, it may be noticed that the cycles of type 1 are always invalid ones, since there is no volatile edge within it and so it is present in every switching graph associated with G.

The cycles of type 2 may not be present in some switchings of G. However, when none of the volatile edges within the cycle is a meta edge, there will be at least one switching where $E_V(c) \subseteq S(G)$, which means there will be switchings where c appears. Therefore, the cycles of type 2 are always invalid in proof-graphs with no meta-edge.

The last type of cycles is the most interesting, because it represents only valid cycles. More than that, it defines all valid cycles when the proof-graph has no meta edge. When a cycle c contains at least one pair of conjugated edges, there is no switching where c may be present, because one edge from the pair must be out of a given switching $S(G)$, for every switching of G.

So we have the following lemma:

Lemma 1 (valid cycle). *Let G be a proof-graph, and G' a spanning subgraph of G without all meta edges. A cycle c in G' has at least one expansion or contraction link iff c is not present in any switching $S(G)$ associated to G.*

Proof.

1. *If a cycle c in G' has at least one expansion or contraction link, then c is not present in any switching $S(G)$.*
   Let l be an expansion or contraction link in c. Let $S(G)$ be a switching associated to G. $S(G)$ must eliminate one of the edges from l, wherefore c will no longer be a cycle in $S(G)$.
2. *If c is not present in any switching graph $S(G)$ associated to G, then c has at least one expansion or contraction link.*
   Proof by contrapositive. Suppose there is a cycle c where $\forall e_i \in E_V(c) \rightarrow \hat{e}_i \notin E_V(c)$. Then there is a switching $S(G)$ such that $\forall e_i \in E_V(c) \mid e_i \in S(G)$. Therefore, c is present in $S(G)$. □

Corollary 1 (acyclicity). *Let G be a proof-graph and G' a spanning subgraph of G without meta edges. Every cycle in G' has at least one expansion or contraction link iff every switching graph $S(G)$ associated to G is acyclic.*

3.2 Connectivity Analysis

A *tree* is an acyclic and connected graph. Another way to define the soundness criterion is to say that every switching graph associated to the proof-graph is a tree. By definition, a tree with n vertices has exactly $n - 1$ edges [8]. We may extend this tree's property to define another lemma for proof-graphs, now concerning its connectivity:

Lemma 2 (connectivity). *Let G be a proof-graph such that all switching graph $S(G)$ associated to G is acyclic. Then every switching $S(G)$ is a tree iff the following formula is valid:*

$$|E(G)| - |L_E(G)| - |L_C(G)| - |E_M(G)| = |V(G)| - 1 \,, \tag{1}$$

where $E(G)$ is the set of edges of G, $L_E(G)$ and $L_C(G)$ are the set of expansion and contraction links of G, respectively, $E_M(G)$ is the set of all meta edges of G, and $V(G)$ is the set vertices of G.

Proof. In every switching associated to G, all meta edges are eliminated, as well as one edge from every expansion and contraction link. So every switching has exactly $|E(G)| - |L_E(G)| - |L_C(G)| - |E_M(G)|$ edges. If a switching $S(G)$ is acyclic, then it is a tree iff the number of edges is equal to the number of vertices minus one, i.e., $|E(G)| - |L_E(G)| - |L_C(G)| - |E_M(G)| = |V(G)| - 1$. □

3.3 The Algorithm

The algorithm proposed here uses the lemmas above and is divided in two procedures. The first procedure is a variation of *Depth First Search* (*DFS* for short). The idea is to find a spanning tree of the proof-graph, and whenever a cycle is delimited by a back edge in the search, this cycle must be validated according to Corollary 1. The second procedure is the application of the formula from Lemma 2 to a given proof-graph.

To validate the cycles efficiently, it is necessary to keep track of the volatile edges visited so far by the search. Therewith, it is possible to identify the expansion/contraction links within the cycle. The original DFS algorithm defines three states for the nodes: *not visited* (white), *discovered* (gray) and *finished* (black) [4]. That is useful to prove some properties about the search. In this algorithm the state of the edges are also important, since it is necessary to know which edges belong to a given cycle. So we extend the concept of state to edges: a *discovered* edge is the one traversed by the search in one way; a *finished* edge is the one traversed both ways in the recursion; and a *not visited* edge is the one not traversed yet.

Likewise, the timestamps $d[v]$ and $f[v]$ [4] are also extended to mark the step when an edge is discovered and finished. In order to do that, new stamps $\delta[n]$ and $\zeta[n]$ are defined, where n may be a vertex or an edge of the graph. For n as a vertex, $\delta[n] = d[n]$ and $\zeta[n] = f[n]$. When n is an edge, $\delta[n]$ is set when the edge is traversed for the first time in the recursion, and $\zeta[n]$ when the edge is traversed back while returning from the recursion.

Crossing an edge in the traversal is still an instantaneous event, as it is in the original search. The timestamp set in $\delta[e]$, for a given edge $e = (u, v)$, is given by $max(\delta[u], \delta[v])$, and the timestamp set in $\zeta[e]$ is given by $min(\zeta[u], \zeta[v])$. The following ordering is defined over δ and ζ (the intuition may be seen in Fig. 7):

Definition 8. *Given two elements, vertex or edge, of a graph $G(V, E)$, u and v, the relation $\prec$ is defined as follows:*

1. *If u and v are both vertices or both edges, $\delta[u] \prec \delta[v] \iff \delta[u] < \delta[v]$ and $\zeta[u] \prec \zeta[v] \iff \zeta[u] < \zeta[v]$;*
2. *If u is an edge and v is a vertex, $\delta[u] \prec \delta[v] \iff \delta[u] \leq \delta[v]$ and $\zeta[u] \prec \zeta[v] \iff \zeta[u] \leq \zeta[v]$.*

The property $d[v] < f[v]$ is still valid while using δ and ζ. It may be seen more clearly when noticing that the values of $\delta[e]$ and $\zeta[e]$ for an edge $e = (u, v)$ comes from the last node visited white passing through the edge. The ordering given above ensures that when the edge e is traversed from v towards u, u will be discovered after e, and will be finished before it. Therefore, the parenthesis property [4] is also true for δ and ζ. In other words, $\delta[u] \prec \delta[e] \prec \delta[v] \prec \zeta[v] \prec \zeta[e] \prec \zeta[u]$.

After all this extensions, we present an algorithm which validates a cycle once it is found in the DFS. The algorithm keeps track of the values of δ and ζ for all vertices and edges, and an auxiliary stack of links. A link is activated when one of its volatile edges is traversed, and is inserted into the stack when the second one is traversed while it is still activated. While returning from the recursion, this link is removed from the stack if it was inserted at this step, or deactivated if it was only activated. When a back edge is identified, the cycle is considered invalid only if the stack is empty or if $min(\delta[e_1], \delta[e_2]) < \delta[u]$, where e_1 and e_2 are the two edges from the link at the top of the stack. The pseudocode of this procedure is in Appendix.

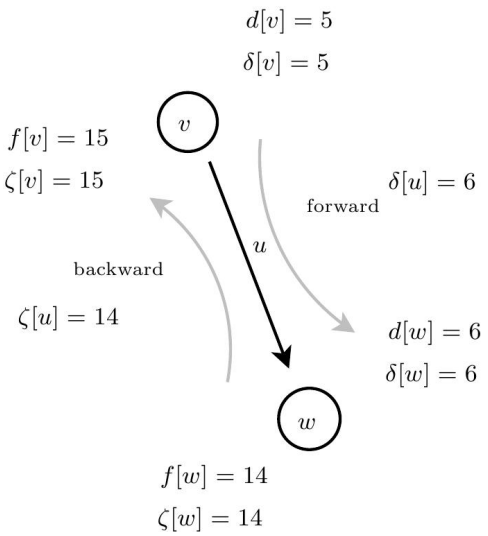


Fig. 7. Ordering induced by δ and ζ. The following is true for an edge $u = (v, w)$: $\delta[v] \prec \delta[u] \prec \delta[w] < \zeta[w] \prec \zeta[u] \prec \zeta[v]$.

Soundness. When a back edge to a *discovered* node v is found by the search, we have the following scenarios for an arbitrary volatile edge e of the proof-graph:

1. e was *not visited*: in this case the expansion/contraction link which contains this edge was not inserted into the stack, and so is not in the cycle;
2. e was *finished*: its expansion/contraction link was deactivated or removed from the stack while this edge was finished. In both cases, the link that contains this edge may not be in the stack, and so is not in the cycle;
3. e was *discovered* and $\hat{e}$ was *not visited*: the link $l = \{e, \hat{e}\}$ is active, but not yet in the stack and consequently not in the cycle;
4. e was *discovered* and $\hat{e}$ was *finished*: by the parenthesis property we know that $\hat{e}$ has activated and deactivated the link before it could be put into the stack. The link is not in the cycle in this case;
5. e was *discovered*, and also was $\hat{e}$: in this case the link $l = \{e, \hat{e}\}$ was added into the stack. This link is not inside the cycle only if $min(\delta[e], \delta[\hat{e}]) \prec \delta[v]$.

The last item defines the case where a link is in the stack at the moment of a cycle verification. The links in the stack are the ones that may be in the tree path between the current node and the node v to where the back edge returns, i.e., in the cycle just found. The top of the stack is the last added link, and so the closest to the current node among those on the stack. If the top one is not in the cycle, by the parenthesis property, no other link in the stack may be in this cycle. To prove the algorithm does such an analysis we have the following lemmas:

Lemma 3. *If an expansion or contraction link $l = \{e, \hat{e}\}$ is in the stack when a node v is visited, then v is a descendant*[3] *of e and $\hat{e}$.*

Proof. If the link l is inside the stack, the second edge was visited while the first one was discovered but not finished. In other words, $\delta[e_1] \prec \delta[e_2] \prec \delta[v]$, where e_1 is the first discovered edge and e_2 the second. With the parenthesis property we have the final configuration $\delta[e_1] \prec \delta[e_2] \prec \delta[v] < \zeta[v] \prec \zeta[e_2] \prec \zeta[e_1]$. □

Lemma 4. *If an expansion or contraction link $l = \{e, \hat{e}\}$ is in the stack when a node u is discovered, then a back edge (u, v) to an already discovered node v where $\delta[v] \prec min(\delta[e], \delta[\hat{e}])$ or $\delta[v] = min(\delta[e], \delta[\hat{e}])$ defines a cycle that contains l.*

Proof. The link l is in the stack when u is discovered. By Lemma 3 u is a descendant of e and $\hat{e}$. When a back edge (u, v) is found, a cycle is defined by the path in the tree from v to u plus the back edge. As there is only one path in the tree between u and v, we have three cases illustrated in Fig. 8:

1. v is a descendant of both edges of l and so $min(\delta[e_1], \delta[e_2]) \prec max(\delta[e_1], \delta[e_2]) \prec \delta[v]$; In this case no edge from l is inside the cycle;
2. e_2 is a descendant of v and e_1 is not. Thus, $min(\delta[e_1], \delta[e_2]) \prec \delta[v] \prec max(\delta[e_1], \delta[e_2])$. In this case the path from v to u in the tree passes through e_2, which is the only descendant of v. The link is again not in the cycle;
3. both edges of l are descendants of v and then $\delta[v] \prec min(\delta[e_1], \delta[e_2]) \prec max(\delta[e_1], \delta[e_2])$. In this case the path in the tree from v to u passes through e_1 and e_2, or only through e_1 and the back edge is e_2. In both cases the cycle contains the link l. □

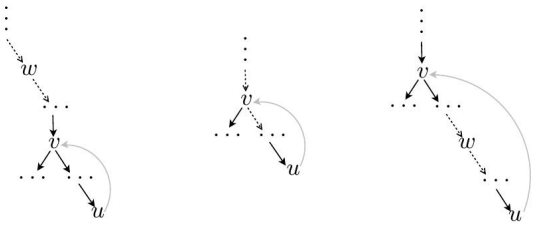


Fig. 8. Illustration of cases where a link l is in the stack when a back edge if found

Since the DFS algorithm finds all cycles in a connected graph, and lemmas 3 and 4 prove that the verification is corresponding to the one given by Lemma 1, the proposed algorithm is correct.

[3] A node v is descendant of an edge e iff $\delta[e] \prec \delta[v] \prec \zeta[v] \prec \zeta[e]$.

Complexity. The core of the algorithm is a depth first search, which has linear time complexity over the number of edges of the graph. The additional cost finding expansion/contraction links from a given volatile edge may be reduced to a constant time operation, if the implementation keeps in each edge a pointer to its link. In the same way, the δ information for edges and vertices may be kept in the edge and vertex itself, reducing the cost of accessing it while validating the cycle to $O(1)$. The stack used to store the active expansion and contraction links is maintained by the recursive search. The reading on the stack is restricted to its top, witch keeps the cycle validation cost to $O(1)$.

The linearity of the algorithm over the number of edges may be extended to the number of vertices, since the graphs we are dealing with are proof-graphs. Each vertex in a proof-graph may be premise and conclusion of at most one link, and all possible links defined in Fig. 1 have no more than two edges on a single vertex. Hence, the number of edges in a proof-graph is limited by $4|V|$.

4 Danos–Regnier and Extension to MLL$^-$

There are only two kinds of links in the MLL$^-$ fragment of proof-nets: times ($\otimes$) and par ($\wp$). The former operator is the *multiplicative and*, and thus conjunctive. The second is the *multiplicative or*, and so disjunctive. Although they have different semantic, their geometry is the same: they are both focussing links with two premises and one conclusion. In this context, a switching graph may be seen as concept similar to *D-R* graph, defined as follows:

Definition 9 (D-R graph). *Given a proof-structure P defined in terms of graphs, a* D-R *associated with P is a spanning subgraph of P in which one of the two edges of every par link is removed.*

Danos–Regnier technique is independent of the logic involved and relies on the structure of the rules [13]. A link l must be *switchable* iff its formulas were already connected before. This occurs when the geometry is contrary to the semantic of the link (i.e. when link is conjunctive defocussing or disjunctive focussing). So the concepts of D-R graph and switching are the same in the context of linear logic.

In Table 1 we see that the *times* rule applies to two distinct proofs. Thus we must not remove $(A, A \otimes B)$ and $(B, A \otimes B)$ edges in a *D-R* graph in order to keep it connected. However, the other three rules are applied to only one previous proof, and keeping the two edges of these links would create a cycle.

Table 1. Times, par, left and right contraction rules

$$\frac{\vdash A, \Gamma \quad \vdash B, \Delta}{\vdash A \otimes B, \Gamma, \Delta} \qquad \frac{\vdash A, B, \Gamma}{\vdash A \wp B, \Gamma} \qquad \frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} \qquad \frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}$$

In a general way, the proposed algorithm verifies if a switchable link is present in all cycles of a graph. It is immediate that it can be applied to MLL^{-}, where *par* is the only switchable link. So what we propose here is another linear algorithm for checking proof-nets. The first one was proposed by Guerrini and uses unification for the verification [7]. Murawski & Ong also use union-find for giving a linear algorithm for verifying Lamarche's essential nets and proof-nets [10]. Our approach, with a DFS-like procedure to validate Lemma 1, seems to be relatively simpler.

References

1. Alves, G.V., de Oliveira, A.G., de Queiroz, R.J.G.B.: Towards normalization for proof-graphs. In: Logic Colloquium, Bulletin of Symbolic Logic, Torino, United States of America, vol. 11, pp. 302–303 (2005)
2. Alves, G.V., de Oliveira, A.G., de Queiroz, R.J.G.B.: Transformations via Geometric Perspective Techniques Augmented with Cycles Normalization. In: Ono, H., Kanazawa, M., de Queiroz, R. (eds.) WoLLIC 2009. LNCS, vol. 5514, pp. 84–98. Springer, Heidelberg (2009)
3. Alves, G.V., de Oliveira, A.G., de Queiroz, R.J.G.B.: Proof-graphs: a thorough cycle treatment, normalization and subformula property. Fundamenta Informaticae 106, 119–147 (2011)
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge (2001)
5. Cruz, M.Q., de Oliveira, A.G., de Queiroz, R.J.G.B., de Paiva, V.: Intuitionistic N-graphs. Logic Journal of the IGPL (Print) (accepted for publication, 2013)
6. Danos, V., Regnier, L.: The Structure of Multiplicatives. Archive for Mathematical Logic 28, 181–203 (1989)
7. Guerrini, S.: A Linear Algorithm for MLL Proof Net Correctness and Sequentialization. Theoretical Computer Science 412(20), 1958–1978 (2011)
8. Harary, F.: Graph Theory. Addison-Wesley Publishing Company (1972)
9. Kneale, W.: The Province of Logic. Contemporary British Philosophy (1958)
10. Murawski, A.S., Ong, C.-H.L.: Dominator Trees and Fast Verification of Proof Nets. In: LICS 2000: Proceedings of the 15th Annual IEEE Symposium on Logic in Computer Science, pp. 181–191. IEEE Computer Society (2000)
11. de Oliveira, A.G.: Proofs from a Geometric Perspective. PhD Thesis, Universidade Federal de Pernambuco (2001)
12. de Oliveira, A.G., de Queiroz, R.J.G.B.: Geometry of Deduction via Graphs of Proof. In: de Queiroz, R. (ed.) Logic for Concurrency and Synchronisation, pp. 3–88. Kluwer (2003)
13. Robinson, E.: Proof Nets for Classical Logic. Journal of Logic and Computation 13, 777–797 (2003)
14. Shoesmith, D.J., Smiley, T.J.: Multiple-Conclusion Logic. Cambridge University Press, London (1978)
15. Statman, R.: Structural Complexity of Proofs. PhD thesis, Stanford (1974)
16. Ungar, A.M.: Normalization, Cut-elimination and the Theory of Proofs. CSLI Lecture Notes, vol. 28. Center for the Study of Language and Information (1992)

Appendix: The Algorithm

Here we present the pseudocode of the algorithm defined in Section 3.

Cycle validation algorithm

```
procedure visite_node (v,timestamp)
  {Assuming v belongs to a proof-graph and the existence of a global flag valid_proof};
   declare
     BOOLEAN activated;
   begin
     v.color := GREY;
     v.delta = timestamp; timestamp := timestamp+1;

     for all edge e=(u,v) incident to v
        if e.color = WHITE then
           e.color = GREY; e.delta := timestamp;
           l := e.link; { retrieving the link that contains this edge };
           activated := FALSE;

           if e.volatile then
              if l.active then { l is active iff its another edge is active }
                 stack_push(l);  activated := TRUE;
              else
                 l.active := TRUE;
           if u.color = WHITE then
              visite_node(u,timestamp);
           else { found a back edge };
              if stack_empty() then
                 valid_proof = FALSE;
              else
                 l := stack_top();
                 if min(l.e1.delta,l.e2.delta) < u.delta then { Scenario 5 }
                    valid_proof := FALSE;
           if e.volatile then
              if activated then
                 l.active := FALSE;
              else
                 stack_pop();
        e.color = BLACK;
     end loop;

     v.color := BLACK;
     timestamp = timestamp+1;
end.
```

First Order Extensions of Residue Classes and Uniform Circuit Complexity

Argimiro Arratia[1,⋆] and Carlos E. Ortiz[2]

[1] LSI, Universitat Politècnica de Catalunya, Barcelona, Spain
arratia@lsi.upc.edu
[2] Arcadia University, Glenside, PA 19038-3295, U.S.A.
ortiz@arcadia.edu

Abstract. The first order logic $\mathcal{R}ing(0,+,*,<)$ for finite residue class rings with order is presented, and extensions of this logic with generalized quantifiers are given. It is shown that this logic and its extensions capture $DLOGTIME$-uniform circuit complexity classes ranging from AC^0 to TC^0. Separability results are obtained for the hierarchy of these logics when order is not present, and for $\mathcal{R}ing(0,+,*,<)$ from the unordered version. These separations are obtained using tools from class field theory, adapting notions as the spectra of polynomials over finite fields to sets of sentences in this logic of finite rings, and studying asymptotic measures of these sets such as their relative densities. This framework of finite rings with order provides new algebraic tools and a novel perspective for descriptive complexity.

1 Introduction

We study the first order logic for finite residue class rings, denoted $\mathcal{R}ing(0,+,*)$, and extensions of this logic with built-in order $<$ (denoted by $\mathcal{R}ing(0,+,*,<)$), and generalized quantifiers (modular quantifiers $\exists^{r,q}$, and majority M, denoted respectively by $\mathcal{R}ing(0,+,*,<)+MOD(q)$ and $\mathcal{R}ing(0,+,*,<)+M$). The structures for these logics are the finite residue classes $\mathbb{Z}_m$ (m positive integer) with modular addition and multiplication. We show that the expressive power of this logic, in the presence of built-in order, coincides with the first order logic for the standard finite models with arithmetic operations as considered in [6,8]. Therefore, from the descriptive complexity perspective, the computational complexity classes that can be described with these algebraic models are the $DLOGTIME$-uniform circuit complexity classes, consisting of circuits of constant depth and polynomial size for which a description can be efficiently computed from the size of their inputs [8].

The perspective of finite residue class rings as instances of problems in these circuit complexity classes allow us to use the extensive algebraic machinery proper of finite rings and fields of integer polynomials, algebraic number theory and, in general, tools from class field theory to study expressibility problems

⋆ Research supported by MICINN project TIN2011-27479-C04-03 (BASMATI), Gen. Cat. 2009-SRG-1428 (LARCA) and MTM2012-36917-C03-03 (SINGACOM)

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 49–63, 2013.

in these logics. More specifically, we use the notion of the prime spectrum of a sentence, defined as the collection of primes p such that the sentence holds in $\mathbb{Z}_p$. Using classical results of Ax and Lagarias [1,2,10] on the nature of these sets of primes for sentences in $\mathcal{R}ing(0,+,*)$, we separate $\mathcal{R}ing(0,+,*)$ from $\mathcal{R}ing(0,+,*) + MOD(q)$ (for every natural $q > 1$). Moreover, we use a much studied asymptotic measure associated with infinite sets of primes, the so-called natural density, together with a result of Friedlander and Iwaniec [7] on the natural density of the set of prime values of polynomials, to separate $\mathcal{R}ing(0,+,*)$ from $\mathcal{R}ing(0,+,*,<)$. The previous results suggest that the notion of density of prime spectra may be useful to separate subclasses of $\mathcal{R}ing(0,+,*,<) + M$. To further that aim, we study the behavior of the density of the spectra of sentences in $\mathcal{R}ing(0,+,*,<) + MOD(q)$ and prove that there exist sentences in this extended ring logic whose prime spectrum has no natural density.

2 Preliminaries

We are interested in the uniform version of the circuit complexity classes ranging from AC^0 to TC^0. Recall that AC^0 is the class of languages accepted by polynomial size, constant depth circuits with NOT gates and unbounded fan-in AND and OR gates. Extending AC^0 circuits with unbounded MOD_q gates, for a fixed integer $q > 1$, one obtains the class $ACC(q)$. For each integer $q > 1$, a MOD_q gate reads its boolean input and returns a 1 if the sum of the input bits is divisible by q (i.e. the sum is equal to 0 mod q); otherwise it returns 0. Joining together all the $ACC(q)$ classes we get the class ACC, that is, $ACC = \bigcup_{q>1} ACC(q)$. On the other hand, extending AC^0 circuits with unbounded MAJ gates one obtains the class TC^0. A MAJ (or *majority*) gate returns a 1 if the sum of n bits given as input is greater or equal to $n/2$; otherwise it returns a 0. The languages decided with the use of MOD_q gates can be decided by using MAJ gates instead. This fact, together with all given definitions of these circuit classes, give us for all $q > 1$ (cf. [4],[8])

$$AC^0 \subseteq ACC(q) \subseteq ACC \subseteq TC^0$$

A circuit family is uniform if a description of each circuit can be computed efficiently from the size of the input; otherwise it is non-uniform. The uniformity condition is crucial to relate time and space complexity with size and depth, the measures for circuit complexity, since it can be shown that in the non uniform setting there are sets with trivial circuit complexity that are not recursive (see [5]). However, many important lower bounds are only known for the non uniform classes (see the survey [4]), and the challenge is to translate these to the uniform setting or find new methods that work for showing lower bounds within uniform circuits.

Our approach to circuit complexity is through finite model theory, and as a consequence we are working with circuit classes that are DLOGTIME-uniform, for as it has been shown in [3], the languages in DLOGTIME-uniform circuit

classes AC^0 to TC^0 are definable in first order logic with built-in arithmetic predicates and some generalized quantifiers. This works as follows. Consider first the basic logic $\mathrm{FO}(\leq, \oplus, \otimes)$, which is first order logic with built-in order relation $\leq$, and two ternary predicates $\oplus$ and $\otimes$. In a finite model for this logic, denoted here as $\mathcal{A}_m$ (m is the cardinality of the model, and its universe is $|\mathcal{A}_m| = \{0, 1, \ldots, m-1\}$), the interpretation of $\leq$ on $\mathcal{A}_m$ is as a total ordering on $|\mathcal{A}_m|$, and the interpretations of $\oplus$ and $\otimes$ are as truncated addition and multiplication (e.g. any pair of elements that add up -or multiplies- to a quantity greater than m is not a defined triple). Consider further the following generalized quantifiers:

($G1$) *Modular* quantifiers, $\exists^{(r,q)}$, which for integers r and q, with $0 \leq r < q$, and first order formula $\phi(x)$, the quantified formula $\exists^{(r,q)} x \phi(x)$ holds in $\mathcal{A}_m$ if and only if the number of values for x that makes $\phi(x)$ true is equal to r modulo q.

($G2$) *Majority* quantifier, M, which for first order formula $\phi(\overline{x}, z)$ with one free variable z, $(Mz)\phi(\overline{a}, z)$ holds in $\mathcal{A}_m$ if and only if $\phi(\overline{a}, z)$ is true for more than half of the possible values for z.

Let $\mathrm{FO}(\leq, \oplus, \otimes) + MOD(q)$, for a fixed integer $q > 1$, be the logic $\mathrm{FO}(\leq, \oplus, \otimes)$ extended with modular quantifiers with moduli q; that is, the set of first order formulas as before plus the quantifiers $\exists^{(r,q)}$ with $0 \leq r < q$.
Let $\mathrm{FO}(\leq, \oplus, \otimes) + MOD = \bigcup_{q>1} (\mathrm{FO}(\leq, \oplus, \otimes) + MOD(q))$, and $\mathrm{FO}(\leq, \oplus, \otimes) + M$ the logic extended with the majority quantifier M. Barrington et al proved

Theorem 1 ([3]). *The languages in DLOGTIME-uniform class $\mathcal{C}$ are exactly those definable in the logic $\mathcal{L}$, where $\mathcal{C}$ is AC^0, $ACC(q)$, ACC or TC^0, and $\mathcal{L}$ is $FO(\leq, \oplus, \otimes)$, $FO(\leq, \oplus, \otimes)+MOD(q)$, $FO(\leq, \oplus, \otimes)+MOD$ or $FO(\leq, \oplus, \otimes)+M$, respectively.* □

3 The Logic of Finite Residue Class Rings and Uniform Circuit Complexity Classes

In this section we define the first order logic for finite residue class rings, and extensions with modular and majority quantifiers. We use $\mathbb{Z}$ to denote the integers, $\mathbb{R}$ for the reals and $\mathbb{P}$ to denote the set of prime numbers. For integers a and b, $b \equiv_d a$ denotes that b is congruent to a modulo d, and (a, b) stands for the greatest common divisor of a, b. For each $m \in \mathbb{Z}$, $m > 0$, we denote by $\mathbb{Z}_m$ the finite residue class ring of m elements. $\mathbb{Z}_m$, as an algebraic structure, consists of a set of elements $\{0, 1, \ldots, m-1\}$, and two binary functions $+$ and $*$ which corresponds to addition and multiplication modulo m, respectively.

Definition 1. *By $\mathcal{R}ing(0, +, *)$ we denote the logic of finite residue class rings. This is the collection of first order formulas over the set of built-in predicates $\{0, +, *\}$, where 0 is a constant symbol, and $+$ and $*$ are binary function symbols. The models of $\mathcal{R}ing$ are the finite residue class rings $\mathbb{Z}_m$, and in each $\mathbb{Z}_m$, the 0*

is always interpreted as the 0*-th residue class (mod* m*), and* $+$ *and* $*$ *are addition and multiplication modulo* m*.*

By $\mathcal{R}ing(0,+,*,<)$ *we denote the logic* $\mathcal{R}ing(0,+,*)$ *further extended with an additional (built-in) order relation* $<$*. In this extension each finite ring* $\mathbb{Z}_m$ *is endowed with an order of its residue classes, given by the natural ordering of the representatives of each class from* $\{0,1,\ldots,m-1\}$*. Also, in this case, the constant* 0 *represents the first element in this order.*

We can further extend $\mathcal{R}ing(0,+,*)$ or $\mathcal{R}ing(0,+,*,<)$ with modular quantifiers and the majority quantifier.

Definition 2. *For every integer* $q>0$*, we denote by* $\mathcal{R}ing(0,+,*)+MOD(q)$ *and* $\mathcal{R}ing(0,+,*,<)+MOD(q)$ *the extensions of the logics* $\mathcal{R}ing(0,+,*)$ *and* $\mathcal{R}ing(0,+,*,<)$ *obtained by the additional requirement that these logics be closed for the quantifiers* $\exists^{(r,q)}x$*, for every* $r=0,1,\ldots,q-1$*. These quantifiers are interpreted in* $\mathbb{Z}_m$ *as in* $(G1)$*. We define* $\mathcal{R}ing(0,+,*)+MOD=\mathcal{R}ing(0,+,*)+\bigcup_{q>0}MOD(q)$ *and* $\mathcal{R}ing(0,+,*,<)+MOD=\mathcal{R}ing(0,+,*,<)+\bigcup_{q>0}MOD(q)$*. Finally, we denote by* $\mathcal{R}ing(0,+,*)+MOD+M$ *and* $\mathcal{R}ing(0,+,*<)+MOD+M$ *the extensions of the logic* $\mathcal{R}ing(0,+,*)+MOD$ *and* $\mathcal{R}ing(0,+,*,<)+MOD$ *obtained by the additional requirement that these logics be closed for the majority quantifier* Mz*, interpreted in* $\mathbb{Z}_m$ *as in* $(G2)$*.*

Our first theorem states that in the presence of a built-in order relation it is logically indistinct to work with the standard finite models $\mathcal{A}_m$ or with the finite residue class rings $\mathbb{Z}_m$. The proof of this fact is long and technically involved; so we state this important result below and postpone the technical details and full proof to the Appendix.

Theorem 2. *For every formula* $\phi(x_1,\ldots,x_k)$ *of* $FO(\leq,\oplus,\otimes)$*, there exists a formula* $\Phi(x_1,\ldots,x_k)$ *of* $\mathcal{R}ing(0,+,*,<)$ *such that for every finite structure* $\mathcal{A}_m$ *and integers* $a_1,\ldots,a_k<m$*,*
$\mathcal{A}_m\models\phi(a_1,\ldots,a_k)$ *if and only if* $\mathbb{Z}_m\models\Phi(a_1,\ldots,a_k)$*.*

Conversely, for every formula $\phi(x_1,\ldots,x_k)$ *of* $\mathcal{R}ing(0,+,*,<)$*, there exists a formula* $\Phi(x_1,\ldots,x_k)$ *of* $FO(\leq,\oplus,\otimes)$ *such that for every finite structure* $\mathbb{Z}_m$ *and integers* $a_1,\ldots,a_k<m$*,*
$\mathbb{Z}_m\models\phi(a_1,\ldots,a_k)$ *if and only if* $\mathcal{A}_m\models\Phi(a_1,\ldots,a_k)$*.* □

The result also applies to the respective extensions of the logics with modular quantifiers and the majority quantifier.

Remark 1. Definability (or expressibility) in the logic of finite rings is given in terms of the finite residue structures $\mathbb{Z}_m$. That is, whenever we say that a property of integers $P(x)$ is definable in $\mathcal{R}ing(0,+,*,<)$, or any fragment $\mathcal{L}$ thereof, we mean that there exists a sentence φ of $\mathcal{L}$ such that for every natural m, $P(m)$ holds in $\mathbb{Z}\iff\mathbb{Z}_m\models\varphi$.

For a given circuit class $\mathcal{C}$, we say that it is definable in the ring logic $\mathcal{L}$ if every property $P(x)$ decidable in $\mathcal{C}$ is definable in $\mathcal{L}$ and, for every sentence φ in $\mathcal{L}$, the set of natural numbers m such that $\mathbb{Z}_m\models\varphi$, is decidable in $\mathcal{C}$. □

As a consequence of the logical equivalence in Theorem 2, any separation result proved for fragments of $\mathcal{R}ing(0,+,*,<)+MOD+M$ can be translated into a corresponding separation result in fragments of $\mathrm{FO}(\leq,\oplus,\otimes)+MOD+M$, with the respective implications to circuit complexity. More specifically, from Theorem 2 and Theorem 1 we have the following definability of uniform circuit classes in ring logics.

Theorem 3. *1) DLOGTIME-uniform AC^0 is definable by $\mathcal{R}ing(0,+,*,<)$.*
2) DLOGTIME-uniform $ACC(q)$ is definable by $\mathcal{R}ing(0,+,*,<)+MOD(q)$, for every natural q.
3) DLOGTIME-uniform ACC is definable by $\mathcal{R}ing(0,+,*,<)+MOD$.
4) DLOGTIME-uniform TC^0 is definable by $\mathcal{R}ing(0,+,*,<)+MOD+M$. □

4 The Prime Spectrum of a Sentence

In this section we separate the expressive power of $\mathcal{R}ing(0,+,*)$ from that of $\mathcal{R}ing(0,+,*)+MOD(d)$ for every natural $d>1$.

Definition 3. *The prime spectrum of a sentence σ of $\mathcal{R}ing(0,+,*,<)+MOD+M$ is defined as the set of primes $Sp(\sigma)=\{p\in\mathbb{P} \;:\; \mathbb{Z}_p\models\sigma\}$.*

The set $Sp(\sigma)$ was introduced by James Ax in connection with his proof of decidability of the theory of finite fields [2]. In particular, Ax proved the following:

Theorem 4 (J. Ax [2]). *The spectrum $Sp(\sigma)$ of any sentence σ of $\mathcal{R}ing(0,+,*)$ is, up to finitely many exceptions, a Boolean combination of sets of the form $Sp(\exists t(f(t)=0))$, where $f(t)\in\mathbb{Z}[t]$ is a polynomial with integer coefficients.* □

Therefore to characterize the spectra of sentences of $\mathcal{R}ing(0,+,*)$ it is sufficient to analyze the spectra of sentences of the form $\exists x(f(x)=0)$ for polynomials $f\in\mathbb{Z}[x]$. Given a polynomial $f(x)\in\mathbb{Z}[x]$, we will indistinctly denote by $Sp(f)$ or $Sp(\exists x(f(x)=0))$ the spectrum of the sentence $\exists x(f(x)=0)$. A basic result of Schur states that every non constant polynomial has an infinite number of prime divisors; that is, $Sp(f)$ is infinite for any $f\in\mathbb{Z}[x]\setminus\mathbb{Z}$ (see [9, Thm. 1] for an elementary proof of this fact). If f is irreducible then the same can be said about the complement of $Sp(f)$, namely,

$$Sp(f)^c := Sp(\forall x(f(x)\neq 0)) = \{p\in\mathbb{P} \;:\; \mathbb{Z}_p\not\models\exists x(f(x)=0)\}$$

Thus, we have the following properties of spectra of the form $Sp(f)$.

Theorem 5. (1) *For any $f\in\mathbb{Z}[x]\setminus\mathbb{Z}$, $Sp(f)$ is infinite.*
(2) *If, additionally, f is irreducible, then $Sp(f)^c$ is infinite.* □

This theorem justifies to consider two spectra as equal if they coincide in all but a finite number of primes, and we denote this by $Sp(f)=^* Sp(g)$.

Let S denote any system of polynomial congruences such as:

$$(S): \qquad f_1(x_1,\ldots,x_n)\equiv_p 0,\ldots,f_m(x_1,\ldots,x_n)\equiv_p 0$$

with each $f_i \in \mathbb{Z}[x_1, \ldots, x_n]$. Let $\Sigma(S)$ be the set consisting of all $p \in \mathbb{P}$ for which S is solvable. Let $\mathcal{B}$ be the Boolean algebra of subsets of $\mathbb{P}$ generated by all the sets $\Sigma(S)$, and let B_k be the Boolean algebra generated by sets $\Sigma(S)$, where the polynomials in S are restricted to have at most k variables, i.e.,

$$f_1(x_1, \ldots, x_k) \equiv_p 0, \ldots, f_m(x_1, \ldots, x_k) \equiv_p 0 \tag{1}$$

The Boolean algebra $\mathcal{B}$ corresponds to the collection of spectra of sentences in $\mathcal{R}ing(0, +, *)$ which, by Theorem 4, collapses to its first level B_1. Lagarias in [10] considered this algebra $\mathcal{B}$ and gave a characterization of the sets of integer congruences $\{p \in \mathbb{P} : p \equiv_d a\}$, for given positive integers d and a, that are in $\mathcal{B}$.

Theorem 6 ([10, Theorem 1.4]). *For any pair of integers a and d, the set $\{p \in \mathbb{P} : p \equiv_d a\}$ is in the Boolean algebra $\mathcal{B}$ if and only if a is of order 1 or 2 in $\mathbb{Z}_d$ (i.e. $a \equiv_d 1$ or $a^2 \equiv_d 1$), or $(a, d) > 1$.* □

Rephrasing this theorem in terms of spectra of sentences we obtain:

Theorem 7. *For any pair of positive integers a and d, with $1 < a < d$, the set $\{p \in \mathbb{P} : p \equiv_d a\}$ is the spectrum of a sentence in $\mathcal{R}ing(0, +, *)$ if and only if $a^2 \equiv_d 1$ or $(a, d) > 1$.* □

We use this theorem to separate $\mathcal{R}ing(0, +, *)$ from $\mathcal{R}ing(0, +, *) + MOD(d)$ for d an arbitrary positive integer.

Remark 2. In $\mathcal{R}ing(0, +, *) + MOD(d)$ we have that $\forall a < d$,

$$Sp\left(\exists^{a,d}(x = x)\right) =^* \{p \in \mathbb{P} : p \equiv_d a\}.$$

Therefore, by Theorem 7, if we can find for every d an $1 < a < d$ that is relatively prime to d, and such that $a^2 \not\equiv_d 1$, then we have a set of primes definable in $\mathcal{R}ing(0, +, *) + MOD(d)$ that is not definable in $\mathcal{R}ing(0, +, *)$. Note first that if there exists $a < p$ with $a^2 \not\equiv_p 1$ then for every α, $a^2 \not\equiv_{p^\alpha} 1$. Also note that for every prime $p > 3$ we have that $2^2 = 4 \not\equiv_p 1$. Hence, for any prime $p > 3$ and any α there exist $a < p$ such that $(a, p) = 1$ and $a^2 \not\equiv_{p^\alpha} 1$. Consider now an arbitrary integer d and its prime decomposition: $d = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_n^{\alpha_n}$. If one of the p_i is greater than 3 then there exists $a < p_i$ such that $(a, p_i) = 1$ and $a^2 \not\equiv_{p_1} 1$. Recall that $\mathbb{Z}_d \cong \mathbb{Z}_{p_1^{\alpha_1}} \times \mathbb{Z}_{p_2^{\alpha_2}} \times \ldots \mathbb{Z}_{p_n^{\alpha_n}}$. Because of this ring isomorphism, we will identify the elements of $\mathbb{Z}_d$ with the corresponding tuples in $\mathbb{Z}_{p_1^{\alpha_1}} \times \mathbb{Z}_{p_2^{\alpha_2}} \times \ldots \mathbb{Z}_{p_n^{\alpha_n}}$. Then the element $(1, \ldots, a, \ldots, 1)$, with a in the i-th coordinate and 1 everywhere else, is relatively prime to d. The only elements that are not relatively prime to d are the ones of the form $(a_1, a_2, \ldots, a_n)$ where for some i, $a_i = 0$ or $a_i = p_i^\beta$ with $1 < \beta < \alpha_i$. Also note that $(1, \ldots, a, \ldots, 1)^2 \not\equiv_d (1, \ldots, 1, \ldots, 1)$. Looking now at powers of 3 and 2, note that $3^2 \equiv_{2^4} 9 \not\equiv_{2^4} 1$, and that $2^2 \equiv_{3^2} 4 \not\equiv_{3^2} 1$ so we can repeat the above argument. Hence, for any d such that there is a prime > 3 that divides d, or 3^2 or 2^4 divides d, we have that there exists $a < d$ such that $(a, d) = 1$ and $a^2 \not\equiv_d 1$. Hence, for such a d, $\{p \in \mathbb{P} : p \equiv_d a\}$ is not expressible in $\mathcal{R}ing(0, +, *)$.

We summarize these remarks in the following propositions.

Proposition 1. *For every natural number $d \neq 2^\alpha 3^\beta$, $0 \leq \alpha \leq 3$, $0 \leq \beta \leq 1$ and $d > 1$, there exists $a < d$ with $(a, d) = 1$ and $a^2 \not\equiv_d 1$.* □

Proposition 2. *For every natural number $d \neq 2, 3, 4, 6, 8, 12, 24$ there exists $a < d$ such that there is no sentence $\theta \in \mathcal{R}ing(0, +, *)$ equivalent to $\exists^{a,d}(x = x)$. Hence, in terms of expressive power, for every $d \neq 2, 3, 4, 6, 8, 12, 24$, $\mathcal{R}ing(0, +, *) \subsetneq \mathcal{R}ing(0, +, *) + MOD(d)$.* □

The problem with $d = 2, 3, 4, 6, 8, 12, 24$ is that for each one of these d, $\forall a \in \mathbb{Z}_d$, either a and d are not relatively prime, or $a^2 \equiv_d 1$. Hence for such integers we can not use the canonical counterexample above to separate $\mathcal{R}ing(0, +, *)$ from $\mathcal{R}ing(0, +, *) + MOD(d)$. However, we have obtained the desired inexpressibility for these integers through direct combinatorial, yet convoluted, arguments. We omit the details due to space restrictions, and just point out the general plan of the proof. For each one of the integer values of d listed above, the key idea is to express in $\mathcal{R}ing(0, +, *) + MOD(d)$ a property of the form $\{p : p \equiv_{d^\gamma} c\}$, for some $c < d^\gamma$ such that $c^2 \not\equiv_{d^\gamma} 1$ for some integer γ. Then, from Theorem 7 we can conclude that this property is not expressible in $\mathcal{R}ing(0, +, *)$.

Theorem 8. *For all integer $d \neq 1$, $\mathcal{R}ing(0, +, *) \subsetneq \mathcal{R}ing(0, +, *) + MOD(d)$.*

5 The Density of the Prime Spectrum of a Sentence

In this section we prove that the expressive power of $\mathcal{R}ing(0, +, *)$ is different from the expressive power of $\mathcal{R}ing(0, +, *, <)$. A way of discerning infinite sets of primes is to compare their relative sizes. Given $S \subset \mathbb{P}$, the *natural density* of S is defined as $\delta(S) = \lim_{t\to\infty} \frac{|\{p \in S : p < t\}|}{|\{p \in \mathbb{P} : p < t\}|}$, whenever this limit exists. Note that if S is finite then $\delta(S) = 0$; and if S and T are two sets of primes such that $S =^* T$ then $\delta(S) = \delta(T)$.

The following observation will be helpful for computing the natural density. By the Prime Number Theorem, the number of primes less than t is asymptotic to $t/\ln t$. Therefore,

$$\delta(S) = \lim_{t\to\infty} \frac{|\{p \in S : p < t\}|}{|\{p \in \mathbb{P} : p < t\}|} = \lim_{t\to\infty} \left(\frac{\ln t}{t}\right) \cdot |\{p \in S : p < t\}| \qquad (2)$$

We will also need the following well known theorem on the density of spectra of irreducible polynomials (cf. [11, §5]).

Theorem 9 (Weak Cěbotarev Theorem). *If $f(x)$ is an irreducible polynomial in $\mathbb{Z}[x]$ of degree n, then $\delta(Sp(f)) = 1/n$.* □

From this theorem it follows that every element of the Boolean algebra B_1 has rational density, and it is 0 if and only if the set is finite. Together with Ax's result (Theorem 4) we obtain:

Theorem 10. *The spectrum of any sentence in $\mathcal{R}ing(0, +, *)$ has rational density, and this density is 0 if and only if the spectrum is finite.* □

We are going to prove that this is not the case for the spectra of sentences in $\mathcal{R}ing(0,+,*,<)$. More specifically, there exists sentences $\sigma \in \mathcal{R}ing(0,+,*,<)$ such that the density of $Sp(\sigma)$ is zero, but the cardinality of $Sp(\sigma)$ is infinite.

An outstanding result by Friedlander and Iwaniec in [7] states that the polynomial $f(x,y) = x^2 + y^4$ has infinitely many prime values. More specifically,

Theorem 11 ([7]). *There are infinitely many primes p of the form $p = a^2 + b^4$, for integers a and b, and the number of these primes $p < t$ is $O(t^{3/4})$.* □

Using this result and (2) we obtain that the density of the set of primes

$$FI := \{p \in \mathbb{P} : p = a^2 + b^4,\ a, b \in \mathbb{Z}\}$$

is of order $\lim_{t\to\infty} \frac{\ln t}{t^{1/4}} = 0$. By Theorem 10 the set FI cannot be the spectrum of a sentence in $\mathcal{R}ing(0,+,*)$. It remains to show that the set FI is definable in $\mathcal{R}ing(0,+,*,<)$. We can in fact show a stronger result.

Theorem 12. *Consider a polynomial in $\mathbb{Z}[x,y]$ of the form $f(x,y) = h(x) + g(y)$, with n the degree of h and d the degree of g. Assume that $n, d \geq 1$ and that the leading coefficients of $h(x)$ and $g(x)$ are positive. Then there exists a sentence $\phi_f \in \mathcal{R}ing(0,+,*,<)$ such that for almost every natural m:*

$\mathbb{Z}_m \models \phi_f$ if and only if "m is prime and there exists naturals $a, c < m$ such that $f(a,c) = m$".

Proof Sketch. The ith derivative of $h(x)$ is a polynomial of degree $n - i$ with positive leading coefficient, hence eventually increasing. Then, for some M, the functions g, h, and the derivatives $h^{(1)}, h^{(2)}, \ldots, h^{(n)}$ are strictly increasing and positive in $[M, +\infty)$, and for every $x > M$, $h(x) > h^{(1)}(x) + \frac{h^{(2)}(x)}{2!} + \ldots + \frac{h^{(n)}(x)}{n!}$. Fix $m > M + 1$. Then, for every pair of integers $b + 1, c \in (M, m)$, if $\mathbb{Z} \models h(b) + g(c) < m \leq h(b+1) + g(c)$ then $h(b+1) + g(c) - m < h(b) + g(c) < m$. It follows that for every $M < c < m$ the smallest value $M < b + 1 < m$ for which $\mathbb{Z} \models h(b+1) + g(c) \geq m$ can be characterized as the smallest value $b > M$ such that $\mathbb{Z}_m \models h(b+1) + g(c) < h(b) + g(c)$. This last statement is expressible in $\mathcal{R}ing(0,+,*,<)$ by a formula $\psi(b,c)$. Putting together the previous observations we obtain that for every $m > M$:

$\mathbb{Z}_m \models \exists b, c(\psi(b,c) \wedge f(b+1,c) = 0)$ iff $\mathbb{Z} \models \exists b, c(b, c < m-1 \wedge f(b+1,c) = m)$.□

As a corollary of this theorem we obtain:

Theorem 13. *$\mathcal{R}ing(0,+,*)$ is properly contained in $\mathcal{R}ing(0,+,*,<)$.* □

6 On the Existence of Spectra without Density

The results of the previous section suggest that the notion of density may be useful for separating fragments of $\mathcal{R}ing(0,+,*,<) + MOD + M$. Therefore we study the behavior of the spectra for sentences in $\mathcal{R}ing(0,+,*,<) + MOD$. We first show that the addition of order to $\mathcal{R}ing(0,+,*)$ enriches so much this logic that we can code the modular semantics of $\mathcal{R}ing(0,+,*,<)$ within itself.

Theorem 14 (Coding Theorem). *For all $\varphi(\overline{x})$ in $\mathcal{R}ing(0,+,*,<)$ there exists a formula $TRAN_\varphi(\overline{x},y)$ in $\mathcal{R}ing(0,+,*,<)$ such that for every natural m, for any $b<m$ and any tuple $\overline{a}<b<m$, $\mathbb{Z}_b \models \varphi(\overline{a}) \iff \mathbb{Z}_m \models TRAN_\varphi(\overline{a},b)$.*

Proof. The proof is by induction in formulas, and is given in the Appendix. □

Remark 3. A similar theorem holds for any fragment of $\mathcal{R}ing(0,+,*,<)+MOD+M$ that contains $\mathcal{R}ing(0,+,*,<)$. □

Remark 4. In the rest of the section, when describing the spectrum of a sentence as $Sp(\sigma)=\{p_n\}_{n\in\omega}$ we assume that $p_1<p_2<\ldots<p_n<\ldots$. □

The second ingredient is to focus on sentences in $\mathcal{R}ing(0,+,*,<)$ that are "thin" in the following sense.

Definition 4. *A sentence θ in $\mathcal{R}ing(0,+,*,<)+MOD+M$ has a thin spectrum if $|Sp(\theta)|=\omega$ and there exists a real number $r\geq 2$ such that when listing the elements of $Sp(\theta)$ as $p_1<p_2<\ldots<p_n<\ldots$, we have that for almost all natural numbers n, $rp_n<p_{n+1}$.*

Essentially the spectrum of a sentence is "thin" if the distance between consecutive primes in the spectrum increases exponentially.

Example 1. For every q prime, let $FIRSTPRIME_q$ be the property that says:

> The cardinality of the structure is a prime number p and, if $q^k<p<q^{k+1}$ for some positive integer k, then there is no prime h such that $q^k<h<p$.

Since $\mathcal{R}ing(0,+,*,<)=FO(\leq,\oplus,\otimes)$ it is easy to see that the property "x is a power of y" (i.e., the usual exponentiation in $\mathbb{Z}$) is definable in $\mathcal{R}ing(0,+,*,<)$. It follows that $FIRSTPRIME_q$ is definable in $\mathcal{R}ing(0,+,*,<)$. On the other hand, one can see that for $q>6$, $FIRSTPRIME_q$ has thin spectrum. □

Definition 5. *For a positive integer x, $\pi(x)=|\{q\in\mathbb{P}:q\leq x\}|$. By extension, for a given sentence σ, we define $\pi_\sigma(x)=|\{q\in\mathbb{P}:q\leq x\wedge\mathbb{Z}_q\models\sigma\}|$.*

Using the fact that the function $F(x)=\frac{x}{\ln(x)}$ is strictly increasing in $\mathbb{R}$, observe that if the spectrum of a sentence θ is thin, with $Sp(\theta)=\{p_n\}_{n\in\omega}$, then for almost all natural numbers n we have that

$$\left(\frac{p_{n+1}}{\ln(p_{n+1})}\right)>\left(\frac{p_n}{\ln(p_n)}\right)\left(\frac{r}{(1+\frac{\ln(r)}{\ln(p_n)})}\right)>2\left(\frac{p_n}{\ln(p_n)}\right).$$

These inequalities, together with the Prime Number Theorem, give us a characterization of thin spectra in terms of the numbers $\pi(p_n)$:

Proposition 3. *For a sentence θ with thin spectrum $Sp(\theta)=\{p_n\}_{n\in\omega}$ we have that, for almost all natural numbers n, $2\pi(p_n)<\pi(p_{n+1})$.* □

Our main interest on sentences with thin spectrum is that, in the modular logics $\mathcal{R}ing(0,+,*,<)+MOD(q)$ (q natural number), they generate sentences without natural density. We begin by looking at the logic $\mathcal{R}ing(0,+,*,<)+MOD(2)$. Consider the sentence $PRIME$ in $\mathcal{R}ing(0,+,*)$ which says that the size of finite ring model is prime (it is enough to say that every element has a multiplicative inverse). By Theorem 14 and Remark 3 for every sentence θ in $\mathcal{R}ing(0,+,*,<)$ $+MOD$, we have that $TRAN_\theta$ is the coding of θ in the residue classes.

Theorem 15. *If θ is a sentence in $\mathcal{R}ing(0,+,*,<)+MOD(2)$ with thin spectrum then the spectrum of the sentence*

$$\psi := PRIME \wedge \exists^{0,2} x\,(TRAN_\theta(x) \wedge TRAN_{PRIME}(x))$$

has no density. Note also that ψ is a sentence in $\mathcal{R}ing(0,+,*,<)+MOD(2)$.

The sentence ψ above essentially says: "The size of the model is a prime q and the number of primes $p < q$ such that $\mathbb{Z}_p \models \theta$ is even". This sentence has no density because the increasing sequence of all primes alternates between intervals of exponential length where any prime in it satisfies ψ, followed by intervals of exponential length where no prime in it satisfies ψ. Thus the lim sup of the density of ψ is strictly greater than $1/2$ but the lim inf is strictly less than $1/2$. The full proof of this theorem is in the Appendix. Since Example 1 gives us a thin sentence in $\mathcal{R}ing(0,+,*,<)$ we then have

Corollary 1. *There exist sentences in $\mathcal{R}ing(0,+,*,<)+MOD(2)$ whose prime spectrum has no density.* □

The previous result can be generalized to $\mathcal{R}ing(0,+,*,<)+MOD(q)$, for every integer $q > 1$; hence showing that in these logics we have sentences whose prime spectrum have no density.

7 Final Comments

We have established the separation of subclasses of $\mathcal{R}ing(0,+,*,<)+MOD$ using results from number theory and the notions of prime spectra of sentences and their natural density. Given the fact that separation of $\mathcal{R}ing(0,+,*,<)$ from $\mathcal{R}ing(0,+,*,<)+MOD(2)$ will yield that

$$DLOGTIME\text{-uniform } AC^0 \neq DLOGTIME\text{-uniform } ACC(2),$$

and that similar separations for fragments of the logic $\mathcal{R}ing(0,+,*,<)+MOD+M$ will yield separations for $DLOGTIME$ uniform classes such as TC^0 and ACC, we believe that it is relevant to understand the prime spectra of sentences in $\mathcal{R}ing(0,+,*,<)+MOD+M$. Of particular interest to us are the following questions:

- Does every spectrum in $\mathcal{R}ing(0,+,*,<)$ has a density? If that is the case, then this would separate this logic from $\mathcal{R}ing(0,+,*,<)+MOD(2)$ because of Theorem 15.

- What can be said of the spectrum of a sentence in $\mathcal{R}ing(0,+,*)+MOD(n)$ for n positive integer? The goal here is to separate $\mathcal{R}ing(0,+,*)+MOD(n)$ from $\mathcal{R}ing(0,+,*)+MOD(m)$, for $m \neq n$ positive integers.
- What can be said of the spectrum of a sentence in $\mathcal{R}ing(0,+,*,<)+M$? We expect these sets to be much more "wild" that the spectra of sentences in $\mathcal{R}ing(0,+,*,<)$ because of the expressive power of the majority quantifier. Thus we believe that a different concept from natural density should be used to study these spectra.

Acknowledgements. The authors gratefully acknowledge the fruitful comments of the three anonymous reviewers that lead to an improvement of the original manuscript.

References

1. Ax, J.: Solving diophantine problems modulo every prime. Ann. of Math. 85(2), 161–183 (1967)
2. Ax, J.: The elementary theory of finite fields. Ann. of Math. 88(2), 239–271 (1968)
3. Barrington, D., Immerman, N., Straubing, H.: On uniformity within NC^1. J. Computer and Syst. Sci. 41, 274–306 (1990)
4. Boppana, R., Sipser, M.: The complexity of finite functions. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science. Algorithms and Complexity, vol. A, pp. 757–804. Elsevier (1990)
5. Borodin, A.: On relating time and space to size and depth. SIAM J. Comput. 6(4), 733–744 (1977)
6. Ebbinghaus, H.D., Flum, J.: Finite Model Theory. Springer (1995)
7. Friedlander, J., Iwaniec, H.: Using a parity-sensitive sieve to count prime values of a polynomial. Proc. Natl. Acad. Sci. USA 94, 1054–1058 (1997)
8. Immerman, N.: Descriptive Complexity. Springer (1998)
9. Gerst, I., Brillhart, J.: On the prime divisors of polynomials. American Math. Monthly 78(3), 250–266 (1971)
10. Lagarias, J.C.: Sets of primes determined by systems of polynomial congruences. Illinois J. Math. 27(2), 224–239 (1983)
11. Wyman, B.F.: What is a Reciprocity Law? American Math. Monthly 79(6), 571–586 (1972)

Appendix

We now proceed to prove that the expressive power of $\mathrm{FO}(\leq,\oplus,\otimes)$ is contained in the expressive power of $\mathcal{R}ing(0,+,*,<)$. Recall that the built-in predicates of $\mathrm{FO}(\leq,\oplus,\otimes)$ are an ordering relation and two ternary relations: $\oplus(x,y,z)$ and $\otimes(x,y,z)$, interpreted in finite models $\mathcal{A}_m$ as truncated natural addition and multiplication. We have to see how to interpret these predicates in the arithmetic of residue classes.

Lemma 1. *There exists first order formulas $ADD(x,y,z)$ and $PROD(x,y,z)$ of $\mathcal{R}ing(0,+,*,<)$ such that for every structure $\mathcal{A}_m$ of $FO(\leq,\oplus,\otimes)$, for every $a,b,c<m$, $\mathcal{A}_m \models \oplus(a,b,c)$ if and only if $\mathbb{Z}_m \models ADD(a,b,c)$, and $\mathcal{A}_m \models \otimes(a,b,c)$ if and only if $\mathbb{Z}_m \models PROD(a,b,c)$.*

Proof. Note that in any model $\mathcal{A}_m$, for any three elements $a, b, c \in \mathcal{A}_m$, $\mathcal{A}_m \models \oplus(a, b, c)$ if and only if $\mathbb{Z}_m \models a + b = c$ and for every $0 < y \leq b$, $\mathbb{Z}_m \models a + y \neq 0$.

Note now that for any two elements $d \leq e < m$, if $\mathbb{Z} \models e + d \geq m$ then the reminder of dividing $e + d$ by m is actually less than e, i.e $\mathbb{Z}_m \models e + d < e$. Thanks to this observation we have that for any three elements $a, b, c \in \mathcal{A}_m$ with $a, b \neq 0$, $\mathcal{A}_m \models \otimes(a, b, c)$ if and only if $\mathbb{Z}_m \models a * b = c$ and $\mathbb{Z}_m \models a < a * 2 < a * 3 < \ldots < a * b$. It follows that the desired formulas are the following:

$$\begin{aligned} ADD(x, y, z) &:= (x + y = z \wedge \forall r < x(y + r \neq 0)) \\ PROD(x, y, z) &:= (x = y = z = 0) \vee ((x \neq 0 \wedge y \neq 0) \wedge \\ &\quad (x * y = z) \wedge \forall r < y(x * r \leq x * r + r)) \end{aligned}$$ □

The rest of this section is devoted to prove that the expressive power of $\mathcal{R}ing(0, +, *, <)$ is contained in the expressive power of $\mathrm{FO}(\leq, \oplus, \otimes)$.

We begin by showing how to simulate the modular addition in $\mathrm{FO}(\leq, \oplus, \otimes)$. Suppose we are in $\mathbb{Z}_m$, and let $a, b < m$. If $a + b = c < m$ then the sum (mod m) of a and b coincides with the natural addition, that is, $\mathcal{A}_m \models \oplus(a, b, c)$. However, if $a + b \geq m$ then $a + b = m + c$, with $c < m$, and $a + b \equiv_m c$. In this case note that $a + w = m$ and $b = w + c$, for some $w < m$. Thus, we can describe in $\mathrm{FO}(\leq, \oplus, \otimes)$ the statement $\mathbb{Z}_m \models a + b = c$, whenever $a + b \geq m$, by a first order expression that says that for some w, $a + w = m$ (the cardinality of the model), and $b = w + c$, the latter being a natural sum (i.e. in $\mathcal{A}_m$). We formalise this idea below.

Lemma 2. *There exists a formula $SUM(x, y, z)$ of $FO(\leq, \oplus, \otimes)$ such that for every model $\mathbb{Z}_m$ of $\mathcal{R}ing(0, +, *, <)$, for every a, b, $c < m$,*

$$\mathbb{Z}_m \models a + b = c \iff \mathcal{A}_m \models SUM(a, b, c)$$

Proof. Consider the following formula in $\mathrm{FO}(\leq, \oplus, \otimes)$ of free variables x and y

$$COMP(x, y) := (\forall u < y \exists w(\oplus(x, u, w))) \wedge \forall w(\neg \oplus (x, y, w))$$

that says that y is the first element of the model such that $\forall z \neg \oplus (x, y, z)$. Then, for $a, b < m$, $\mathcal{A}_m \models COMP(a, b) \iff a + b \equiv_m 0$, and we have
$SUM(x, y, z) := \oplus(x, y, z) \vee \exists w(COMP(x, w) \wedge \oplus(w, z, y))$. □

Next we show how to simulate the modular multiplication in $\mathrm{FO}(\leq, \oplus, \otimes)$. Let us first explain the intuition behind. Our goal is to show that the modular operation $ab \equiv_m c$ can be described by formulas of $\mathrm{FO}(\leq, \oplus, \otimes)$. To achieve this goal we shall show that we can decompose a and b in the following form: pick the least $u < m$ such that $u^2 > m$, and write $a = k_1 u + t_1$ and $b = k_2 u + t_2$, for some k_1, k_2, t_1, $t_2 < u$. Note that u does not depend on a or b, but the decomposition above does depend on u. Then ab can be decomposed as

$$ab \equiv_m (k_1 u + t_1)(k_2 u + t_2) \equiv_m (k_1 k_2)u^2 + (k_1 t_2)u + (k_2 t_1)u + t_1 t_2$$

Since $k_1 k_2$, $k_1 t_2$, $k_2 t_1$, $k_2 t_1 < m$ (by the way we chose u), these products are all describable in $\mathrm{FO}(\leq, \oplus, \otimes)$. Finally, we show that for any $h < m$, the products hu

and hu^2 are definable in FO$(\leq, \oplus, \otimes)$. The key in this last step is to decompose u as $(u-1)+1$, and decompose $u^2 \equiv_m su+t$, for $s=0,1$ and $t<u$. This last expression follows by the characteristics of u. Then use these decompositions of u and u^2 to decompose further $(k_1k_2)u^2+(k_1t_2)u+(k_2t_1)u+t_1t_2$ into summands so that each will be expressible in FO$(\leq, \oplus, \otimes)$, and then invoking Lemma 2, we get that this sum, and hence the modular product $ab \equiv_m c$, is expressible in FO$(\leq, \oplus, \otimes)$. We now present the formal details.

Lemma 3. *Let m be a positive integer. Let u be the smallest positive integer such that $u < m$ and $u^2 > m$. Then:*

1. *u is definable in FO$(\leq, \oplus, \otimes)$.*
2. *There exists $s \in \{0,1\}$ and $w < u$ such that $u^2 \equiv_m su+w$. Furthermore the statement $u^2 \equiv_m su+w$ is definable in FO$(\leq, \oplus, \otimes)$.*
3. *For every $a, b < m$, there exists k_1, k_2, t_1, $t_2 < u$, such that $a \equiv_m k_1u+t_1$ and $b \equiv_m k_2u+t_2$. Furthermore, the statements $a \equiv_m k_1u+t_1$ and $b \equiv_m k_2u+t_2$ are definable in FO$(\leq, \oplus, \otimes)$.*

Proof. 1. Consider the following formula of FO$(\leq, \oplus, \otimes)$ in the variable u:

$$ROOT(u) := \forall t < u \exists w(\otimes(t,t,w)) \wedge \forall w(\neg \otimes (u,u,w)) \tag{3}$$

Then, for an element u in $\mathcal{A}_m$, $\mathcal{A}_m \models ROOT(u)$ if and only if u is the smallest element of $\mathcal{A}_m$ such that $u^2 > m$.

2. Since $(u-1)^2 < m < u^2$, then if we divide u^2 by m we get $u^2 = km+r$, for some $k < m$ and some $r < u^2-(u-1)^2 = 2u-1 < 2u$. Hence, $r = su+w$ with $s \in \{0,1\}$ and $w < u$. Note finally that $u^2 \equiv_m su+w$ with $s \in \{0,1\}$ and $w < u$ if and only if $(u-1)u+(1-s)u \equiv_m w$. Since the result of each one of these products is less that m, and we proved that addition modulo m is definable in FO$(\leq, \oplus, \otimes)$, we can conclude that the statement $u^2 \equiv_m su+w$ with $s \in \{0,1\}$ and $w < u$ is definable in FO$(\leq, \oplus, \otimes)$.
3. If $a, b < u$ then $a = 0u+a$ and $b = 0u+b$. If a or $b \geq u$ then the expressions follow by simple division by u (which is definable in FO$(\leq, \oplus, \otimes)$) and by using the fact after division k_1, k_2, t_1, t_2 are less than u so that that the products k_1u, k_2u are directly definable using $\otimes$. The expresibility of these formulas in FO$(\leq, \oplus, \otimes)$ is left to the reader. □

Lemma 4. *Let m and u be as in the previous lemma. For every $a, c < m$, the modular relations $au \equiv_m c$ and $au^2 \equiv_m c$ are expressible in FO$(\leq, \oplus, \otimes)$.*

Proof. We have that $au \equiv_m c$ if and only if
$\exists u \exists k_1, t_1 \, (k_1 < u \wedge t_1 < u \wedge ROOT(u) \wedge a \equiv_m k_1u+t_1 \wedge c \equiv_m (k_1u+t_1)u)$ iff
$\exists u \exists k_1, t_1 \, \big(k_1 < u \wedge t_1 < u \wedge ROOT(u) \wedge a \equiv_m k_1u+t_1 \wedge c \equiv_m k_1u^2+t_1u\big)$ iff
$\exists u \exists k_1, t_1 (k_1 < u \wedge t_1 < u \wedge ROOT(u) \wedge a \equiv_m k_1u+t_1 \wedge$
$\exists s, u, w(s = 0 \vee s = 1 \wedge w < u \wedge u^2 \equiv_m su+w \wedge c \equiv_m k_1(su+w)+t_1u))$ iff
$\exists u \exists k_1, t_1 (k_1 < u \wedge t_1 < u \wedge ROOT(u) \wedge a \equiv_m k_1u+t_1 \wedge$
$\exists s, u, w(s = 0 \vee s = 1 \wedge w < u \wedge u^2 \equiv_m su+w \wedge c \equiv_m k_1su+k_1w+t_1u))$

From the above equivalence and from Lemmas 2 and 3 we can conclude that $au \equiv_m c$ is expressible in FO$(\leq, \oplus, \otimes)$. Analogously, note that

$$au^2 \equiv_m c \text{ iff } \exists k_1, t_1 (k_1 < u \wedge t_1 < u \wedge ROOT(u) \wedge a \equiv_m k_1 u + t_1 \wedge (k_1 u + t_1)u^2 \equiv_m c).$$

As in the previous case, note also that

$$c \equiv_m (k_1 u + t_1)u^2 \text{ iff } \exists s, u, w (s = 0 \vee s = 1 \wedge w < u \wedge u^2 \equiv_m su + w \wedge c \equiv_m (k_1 u + t_1)(su + w))$$

and this last statement is equivalent to

$$\exists s, u, w (s = 0 \vee s = 1 \wedge w < u \wedge u^2 \equiv_m su + w \wedge c \equiv_m k_1 s u^2 + t_1 su + k_1 wu + t_1 w)$$

which in turn, replacing u^2, is equivalent to

$$\exists s, u, w (s = 0 \vee s = 1 \wedge w < u \wedge u^2 \equiv_m su + w \wedge c \equiv_m k_1 s(su + w) + t_1 su + k_1 wu + t_1 w)$$

From Lemmas 2 and 3 we conclude from this expression that the relation $au^2 \equiv_m c$ is also definable in FO$(\leq, \oplus, \otimes)$. □

We can now show that the modular product is definable in FO$(\leq, \oplus, \otimes)$.

Theorem 16. *For every $a, b, c < m$, the modular product $ab \equiv_m c$ is definable in FO$(\leq, \oplus, \otimes)$, i.e., there exists a formula $PROD(x, y, z)$ of FO$(\leq, \oplus, \otimes)$ such that for every model $\mathbb{Z}_m$ of $\mathcal{R}ing(0, +, *, <)$, for every a, b, $c < m$,*

$$\mathbb{Z}_m \models a * b = c \iff \mathcal{A}_m \models PROD(a, b, c)$$

Proof. $ab \equiv_m c$ if, and only if, for the smallest $u < m$, for which $u^2 > m$, there exists k_1, t_1, k_2, $t_2 < u$ such that $a \equiv_m k_1 u + t_1$ and $b \equiv_m k_2 u + t_2$, and

$$c \equiv_m (k_1 u + t_1)(k_2 u + t_2) \equiv_m (k_1 k_2)u^2 + (k_1 t_2)u + (t_1 k_2)u + t_1 t_2$$

By Lemma 4, the relations $(k_1 k_2)u^2 \equiv_m x$, $(k_1 t_2)u \equiv_m y$ and $(t_1 k_2)u \equiv_m z$ are all definable in FO$(\leq, \oplus, \otimes)$; u is definable and also the relations $a \equiv_m k_1 u + t_1$ and $b \equiv_m k_2 u + t_2$ are definable in FO$(\leq, \oplus, \otimes)$. Using Lemma 2 we can obtain a definition for the relation $ab \equiv_m c$ in FO$(\leq, \oplus, \otimes)$ with a, b, c as parameters. □

Proof of the Atomic Case for the Coding Theorem (Theorem 14).
Case: Modular congruence. Fix an arbitrary natural m and three elements x, y, $z < m$. We need to code the statement $x \equiv_y z$ into $\mathbb{Z}_m$. First consider the formula in $\mathcal{R}ing(0, +, *, <)$: $DIV(x, y, z, w) := (\otimes(y, z, t) \wedge \oplus(t, w, x) \wedge 0 \leq w \wedge w < z$, which expresses the Euclidean division of x by y with quotient z and remainder w (Recall that FO$(\leq, \oplus, \otimes) = \mathcal{R}ing(0, +, *, <)$, hence the relations $\oplus$ and $\otimes$ are definable in $\mathcal{R}ing(0, +, *, <)$). Then the formula

$$MOD(x, y, z) := \exists s, w, r (DIV(x, y, s, r) \wedge DIV(z, y, w, r)).$$

that says that the remainders of dividing x by y, and z by y are the same, is such that for every natural m, for every $a, d < c < m$,

$$\mathbb{Z}_c \models a = d \iff \mathbb{Z}_m \models MOD(a, c, d).$$

Case: Modular addition and multiplication. Fix an arbitrary natural m and three elements $x, y, z, w < m$. We need to code the statements $x + w \equiv_y z$ and $xy \equiv_y z$ into $\mathbb{Z}_m$. This is accomplished by the formulas: $MOD(x + w, y, z)$ and $MOD(x * w, y, z)$. More precisely, we have that for every $a, b, d < c < m$,

$$\mathbb{Z}_c \models a + b = d \iff \mathbb{Z}_m \models MOD(a + b, c, d)$$
$$\text{and } \mathbb{Z}_c \models a * b = d \iff \mathbb{Z}_m \models MOD(a * b, c, d).$$

Case: Linear order. Fix an arbitrary natural number m, and three integers $x, y, z < m$. We need to code the statement: $\mathbb{Z}_y \models x < z$. This is accomplished by the following formula:

$$ORD(x, y, z) := \exists a, b(a < b < y \wedge MOD(a, y, x) \wedge MOD(b, y, z))$$

which says that $a \equiv_y x$ and $b \equiv_y z$ and $a < b < y$. More precisely, we have that for every natural m, for every naturals $a, c < b < m$,

$$\mathbb{Z}_b \models a < c \iff \mathbb{Z}_m \models ORD(a, b, c).$$

This completes the proof of the atomic case for the Coding Theorem. Next the conjunction, negation and existential expressions follow directly from the appropriate constructions. Assuming the $TRAN$ process has been defined for formulas $\varphi(\overline{x}), \psi(\overline{x}), \theta(\overline{x}, z)$ we define:

- $TRAN_{\psi \wedge \varphi}(\overline{x}, y) := TRAN_{\psi}(\overline{x}, y) \wedge TRAN_{\varphi}(\overline{x}, y)$.
- $TRAN_{\neg\psi}(\overline{x}, y) := \neg TRAN_{\psi}(\overline{x}, y)$.
- $TRAN_{\exists z\theta}(\overline{x}, y) := \exists z < y TRAN_{\theta}(\overline{x}, z, y)$. □

Proof of Theorem 15

A ring model $\mathbb{Z}_m$ satisfy ψ if and only if m is prime and the number of primes $q < m$ such that $\mathbb{Z}_q \models \theta$ is even. Let $p_1 < p_2 < \ldots < p_n < \ldots$ be the primes in $Sp(\theta)$, then $\mathbb{Z}_m \models \psi$ if and only $m = p$, for some prime p, and there exists an integer j such that $p_{2j} < p \leq p_{2j+1}$. Likewise, $\mathbb{Z}_m \models \neg\psi$ if and only if m is not a prime, or if it is a prime, say p, then there exists an integer j such that $p_{2j-1} < p \leq p_{2j}$. Observe that

$$\pi_{\psi}(p_{2k}) = \pi_{\psi}(p_{2k-1}) + |\{p \in \mathbb{P} : p_{2k-1} < p \leq p_{2k} \wedge \mathbb{Z}_p \models \psi\}|$$

and since no prime p, such that $p_{2k-1} < p \leq p_{2k}$, can generate a model $\mathbb{Z}_p$ of ψ, we have that $|\{p \in \mathbb{P} : p_{2k-1} < p \leq p_{2k} \wedge \mathbb{Z}_p \models \psi\}| = 0$, and in consequence, $\pi_{\psi}(p_{2k}) = \pi_{\psi}(p_{2k-1})$. Using that $Sp(\theta)$ is thin we have that for all integer k,

$$\frac{\pi_{\psi}(p_{2k})}{\pi(p_{2k})} = \left(\frac{\pi_{\psi}(p_{2k-1})}{\pi(p_{2k})}\right) \xrightarrow{\text{for almost all } k} < \frac{1}{2}$$

$$\frac{\pi_{\psi}(p_{2k+1})}{\pi(p_{2k+1})} \geq \frac{\pi(p_{2k+1}) - \pi_{\psi}(p_{2k})}{\pi(p_{2k+1})} = \left(1 - \frac{\pi_{\psi}(p_{2k})}{\pi(p_{2k+1})}\right) \xrightarrow{\text{for almost all } k} > \frac{1}{2}$$

We then have that $\lim_{n\to\infty} \frac{\pi_{\psi}(n)}{\pi(n)}$ does not exists. □

Quantum Probabilistic Dyadic Second-Order Logic*

Alexandru Baltag, Jort M. Bergfeld, Kohei Kishida, Joshua Sack, Sonja J.L. Smets, and Shengyang Zhong

Institute for Logic, Language and Computation, Universiteit van Amsterdam
Science Park 107, 1098XG Amsterdam, The Netherlands

Abstract. We propose an expressive but decidable logic for reasoning about quantum systems. The logic is endowed with tensor operators to capture properties of composite systems, and with probabilistic predication formulas $P^{\geq r}(s)$, saying that a quantum system in state s will yield the answer 'yes' (i.e. it will collapse to a state satisfying property P) with a probability at least r whenever a binary measurement of property P is performed. Besides first-order quantifiers ranging over quantum states, we have two second-order quantifiers, one ranging over quantum-testable properties, the other over quantum "actions". We use this formalism to express the correctness of some quantum programs. We prove decidability, via translation into the first-order logic of real numbers.

1 Introduction

This paper introduces a powerful new logic for reasoning about quantum computation. Our *Quantum Probabilistic Dyadic Second-Order Logic* (**QPDSOL**) is *expressive enough* to capture superpositions, entanglements, measurements, quantum-logical gates and probabilistic features; it can express the correctness of a wide range of complex quantum protocols and algorithms; but at the same time it is logically tractable, in the sense of being *decidable*.

It is well-known that "classical" First-Order Logic is undecidable, and moreover that "classical" Second-Order Logic, as well as its monadic and dyadic fragments[1] are not even axiomatizable. By moving to the quantum world, it is natural to *extend* the range of first-order quantifiers to *quantum "states"* (i.e. superpositions of classical states), while at the same time it is natural to *restrict* the range of monadic second-order quantifiers to *quantum-testable properties* (closed linear subspaces of the state space), and to similarly restrict the range of dyadic second-order quantifiers to *quantum "actions"* (linear maps between

* The research of J. Bergfeld, K. Kishida and J. Sack has been funded by VIDI grant 639.072.904 of the NWO. The research of S. Smets is funded by the VIDI grant 639.072.904 of the NWO and by the FP7/2007-2013/ERC Grant agreement no. 283963. The research of S. Zhong has been funded by China Scholarship Council.

[1] *Monadic* Second-Order Logic is the fragment allowing quantification only over *unary* predicates, while the Dyadic fragment allows quantification only over *unary and binary* predicates.

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 64–80, 2013.

state spaces). Indeed, it is widely accepted in the literature on Quantum Logic and on Foundations of Quantum Mechanics that quantum-testable properties are *the only* experimentally meaningful properties of a quantum system: any other (non-testable, non-linear) properties have no physical/experimental meaning in a quantum setting. Similarly, it is widely accepted in quantum computation that all meaningful quantum programs are obtainable by composing quantum gates (unitary maps) and quantum tests (measurements), and thus are quantum "actions" in the above sense.[2] So restricting the interpretations of the unary and binary predicates as above is a natural thing to do in a quantum setting: it only restricts the second-order quantifiers to properties/actions that are *physically meaningful*. The resulting logic *is indeed the natural "quantum analogue"* of classical (dyadic) second-order logic!

Surprisingly, this quantum analogue turns out to be much more tractable than its classical counterpart: the above well-justified and natural restrictions of range are enough to restore full decidability, even after the addition of "exotic" features such as probabilistic predication and tensors!

In a sense, this is not as surprising as it may first appear. Our semantics for second-order logic is "non-standard": not all sets of states (whose existence is guaranteed by the standard axioms of Set Theory) are accepted as "predicates". The second-order quantifiers are thus restricted to a limited range of predicates. Such non-standard variations of second-order logic have been studied before. Henkin's weak semantics for second-order logic [11] involves a restriction on the range of the second-order quantifiers (to some model-dependent class of admissible predicates), that restores the axiomatizability of the logic. Some variants of monadic second-order logic (for very restricted models) are even decidable [14].

But these classical results are conceptually very different from ours: none of these weaker logics can be considered to be a genuine and natural variant of second-order logic. In particular, Henkin's semantics (restricting second-order quantifiers to some arbitrary collections of subsets of the state space) is not a independently-justifiable restriction. It does not even provide a unique, canonical way to restrict the quantifiers (but a model-dependent one). In contrast, our restriction of quantifiers to quantum-testable properties (and quantum-performable operations) is natural, canonical (providing a unique collection for each dimension) and amply justified on independent grounds by a whole body of literature in Quantum Logic, Foundations of Quantum Mechanics and Quantum Computation.

[2] The converse is not obvious, and may even fail in practice. But from a theoretical perspective, one can argue that the converse is true in a sense: for any quantum action (linear map) f between systems $\mathcal{H}$ and $\mathcal{H}'$ there exists an entangled state s_f in $\mathcal{H} \otimes \mathcal{H}'$ with the property that, if a local measurement performed on the $\mathcal{H}$-subsystem of (a system in state) s_f yields state x, then after that a local measurement on the $\mathcal{H}'$-subsystem will yield the result $f(x)$. In this way, any such action f can be physically computed, in principle: first, prepare a large number of entangled states s_f; then perform local measurements on the $\mathcal{H}$-subsystem until one of them yields the desired input value x; and then perform a measurement on the $\mathcal{H}'$-subsystem, yielding the output-value $f(x)$.

Indeed, seen from the perspective of the quantum world, our "non-standard" semantics *looks like the "true" semantics* of second-order logic: it only eliminates the predicates that are "physically meaningless". Moreover, while in a sense being a restriction of the classical (standard) semantics, in a different sense this can be thought of as *an extension of the classical semantics*! Indeed, one can argue that, if we restrict ourselves to *classical states* (i.e. n-long tuples of bits $|0\rangle$ or $|1\rangle$, for any dimension n) then *all the standard predicates of such classical states are realized as quantum-testable predicates* (and hence fall within the range of our second-order quantifiers): for *every* set $A \subseteq \{|0\rangle, |1\rangle\}^n$, there exists a unique quantum-testable predicate (linear subspace[3]) $P_A \subseteq \mathcal{H}_2^{\otimes n}$ such that a classical n-state $s \in \{|0\rangle, |1\rangle\}^n$ satisfies P_A iff it belongs to the set A. So, insofar as *classical* states are concerned, our range restriction for second-order quantifiers *is not a restriction at all*: their range really includes (quantum counterparts of) *every set* of classical states. It is only when we look at non-classical (superposed) states that we see that the quantifier range is restricted (though in a natural way).

In conclusion, regardless of whether one considers it as a natural restriction of the classical semantics for (predicates of) quantum states, or as a huge extension of the classical semantics for (predicates of) classical states, we can still safely claim that *our logic really is the correct quantum (and probabilistic) counterpart of the classical (dyadic) second-order logic.*

As a consequence, we think that our decidability result is a significant contribution to the logical understanding of quantum mechanics: it shows in essence that, whereas the natural formulation of (dyadic) second-order logic in the *classical* world is undecidable, *the natural formulation of (dyadic) second-order logic for the quantum world is decidable.*

The fundamental reason for this tractability is the one severe constraint put by quantum mechanics on the "meaningful" properties and actions: *linearity.*[4] Once again, this does not really restrict the predicates/actions as far as classical states are concerned (since any two classical states of the same space are orthogonal to each other, a classical state cannot be written as a linear combination of other classical states). But linearity *does* constrain the behavior of "meaningful" predicates/actions on *superposed* states. And, in the end, linearity allows the reduction of all the "meaningful" second-order objects (predicates/actions) to their underlying linear expressions: matrices of (complex) numbers.

So this natural (and physically-imposed) linearity constraint reduces thus our quantum version of second-order logic to the *first-order theory* of complex numbers. And now, a classical result comes to our help: while first-order logic is in general undecidable (and the first-order theories of many useful structures, such as the ring of natural numbers, are not even axiomatizable), *the first-order theory of complex numbers is decidable.* This was pointed out by Alfred Tarski [17] as a corollary to the analogue result for the field of real numbers (proved in the same paper by quantifier-elimination).

[3] In fact, this is the linear subspace P_A generated by A.

[4] For unary predicates: having a linear subspace (not an arbitrary subset) as their extension; for actions: being induced by a linear map.

Our decidability proof makes essential use of Tarski's decidability result, as well as of the finite dimensionality; it translates effectively the probabilistic dyadic second-order logic of finite-dimensional quantum systems into the decidable first-order theory of reals. This proof method is inspired by the one given by Dunn et al. in [10], where the traditional (propositional) quantum logic of any finite-dimensional Hilbert space was proved to be decidable. However, the result in [10] required that we first fix a particular Hilbert space (model of a quantum system) of a finite dimension, so as to translate the logic of the space into the finitary language of reals, thus limiting the scope of application by fixing a finite dimension (and hence the number of *quantum bits* or *qubits*) throughout the discourse. In contrast, our logic overcomes this limitation by using types and tensors in the language, thus accommodating *an unbounded number of qubits*, while preserving the logical tractability.

Our results in this paper can be seen as part of a wider on-going effort towards bridging the gap between traditional quantum logic and the theory of quantum computation. On the one hand, traditional quantum logic (as originated in [7]) has focused on axiomatics and logical properties of the lattice of closed linear subspaces of an *infinite-dimensional* Hilbert space, with the goal being "to discover the logical structure one may hope to find in physical theories which, like QM, do not conform to classical logic" [7]. Quantum computation, on the other hand, concerns encoding and describing computations on the basis of quantum systems, and involves quantum ingredients such as superposition and entanglement, in order to perform certain tasks much faster than classical computers. The underlying theoretical framework for quantum computation is given by *finite-dimensional* Hilbert spaces. Among the few treatments of such finite-dimensional quantum logics and their decidability are the work of [8,10].

Another contrast between quantum logic and quantum computation lies in the treatment of "quantum entanglement". In traditional quantum logic, entanglement has been viewed as a problem-child, posing difficulties to the lattice-theoretic setting [2,15] (though naturally treatable in a category-theoretical setting [1,16]). In quantum computing, however, entanglement is viewed as a *computational resource*, that allows us to go beyond the world of classical computing. Among the papers that address this part of the gap between quantum logic and quantum computation are [3,8], and [9, Chapter 17]. Our work strengthens the connection further. The logic we propose in the following sections—dyadic second-order quantum logic—is fit to deal with multi-partite systems that exhibit quantum entanglement. Equipped with an explicitly typed language, with types for states, predicates, and actions, with tensor operators connecting them, as well as with probabilistic predication, our logic allows us to capture all the essential computational properties of composite quantum systems, and in particular it can encode the correctness of a wide range of quantum algorithms.

The design of dyadic second-order quantum logic in this paper builds further on the earlier work of Baltag and Smets on propositional dynamic quantum logics [5,6]. It is well known that standard Propositional Dynamic Logic (PDL), as well as its fragment called the Hoare Logic, plays an important role in classical

computing and programming. In particular, PDL and Hoare Logic are among the main logical formalisms used for classical program verification. The quantum version of PDL extends the area of applicability to the verification of quantum programs and quantum protocols. In [6], a quantum dynamic logic was designed that was expressive enough to prove the correctness of basic non-probabilistic quantum protocols such as teleportation and quantum secret sharing. The work of [4] used the tools of [10] to prove the decidability of such a propositional dynamic quantum logical system. While these results are important, note that the logic in [4] was unable yet to capture the correctness of any probabilistic quantum protocols. In this paper, we overcome this limitation and equip our logic with a *probabilistic predication operator*, indicating that a state of a quantum system will collapse to a state having property P with probability at least r whenever a measurement of property P is performed. This operator allows us to express the correctness of those quantum algorithms (such as quantum search) that make essential use of quantum probabilities.

A remark is in order regarding the fact that each given program in our syntax, and so each given sentence, uses only a given number of qubits (and thus it refers to a Hilbert space with a given finite number of dimensions). We would like to stress that our result is much more significant than, say, the decidability of checking the correctness of a classical circuit of a given size applied to a problem of given input size. This is because *we do not fix the size of the input, but only the dimension*. This point is important, since for a given fixed dimension (greater than 1) there are *infinitely* (in fact *uncountably*) many non-equivalent quantum states of that dimension (while typically there are only finitely many inputs of a given size). Hence, the algorithm for deciding satisfiability (on states of a space of given dimension) *cannot* simply proceed by exhaustive search over a finite domain (as in the case of models of bounded size). The correctness statements presented in this paper really capture the correctness of a program for uncountably many non-equivalent inputs![5]

2 Preliminaries

According to quantum theory (see e.g. [12]), any quantum system can be described by a Hilbert space $\mathcal{H}$ of appropriate dimension. Similar to the tradition of [13], we identify *(pure) states* of the system with the "rays" in $\mathcal{H}$ (i.e. the one-dimensional linear subspaces of $\mathcal{H}$) and the "impossible state" (zero-dimensional subspace, which we include as it allows us to discuss only total functions without loss of generality). Given a vector $|\psi\rangle \in \mathcal{H}$, we will write $\widehat{|\psi\rangle}$ for the state generated by $|\psi\rangle$. Given a state space $\mathcal{H}$ of some quantum system, we write $\Sigma_{\mathcal{H}}$

[5] Moreover, these correctness statements, even when translated back into the arithmetic of real numbers, do *not* boil down to simple equations involving addition and multiplication of *specific* real numbers and/or matrices. Instead, they reduce to complex first-order statements in the theory of real numbers, that involve in an essential way quantification over uncountably many objects. It just happens that (due to Tarski's theorem) this theory is still decidable!

for the set of all states, i.e. the set of all one-dimensional linear subspaces of $\mathcal{H}$ and $\widehat{\mathbf{0}_{\mathcal{H}}}$ (where $\mathbf{0}_{\mathcal{H}}$ is the zero vector).

Any change of the state of a quantum system can be described by a linear map on $\mathcal{H}$. There are two important kinds of linear maps: unitary operators and projectors. A *unitary operator* U is such that both $U^\dagger U$ and $UU^\dagger$ are the identity operator, where $(\cdot)^\dagger$ is the adjoint operation on linear maps. In quantum computation, unitary operators are the counterpart of logical gates in classical computation. An operator A is a *projector*, if it is bounded, idempotent, i.e. $AA = A$, and self-adjoint, i.e $A^\dagger = A$. Projectors are essential in describing quantum measurements, which are the only way we extract information from a quantum system. In this paper, our level of abstraction allows us to consider not only linear maps on a Hilbert space but also those between different Hilbert spaces. Every linear map $A : \mathcal{H} \to \mathcal{H}'$ from a quantum system $\mathcal{H}$ to a possibly different system $\mathcal{H}'$ naturally induces a unique function (also denoted by A) from the set of states $\Sigma_{\mathcal{H}}$ to the set of set of states $\Sigma_{\mathcal{H}'}$, given by $A(\widehat{|\psi\rangle}) := \widehat{A(|\psi\rangle)}$ for every $|\psi\rangle \in \mathcal{H}$. An *action* is any such function $A : \mathcal{H} \to \mathcal{H}'$ induced on state spaces by some linear map $A : \Sigma_{\mathcal{H}} \to \Sigma_{\mathcal{H}'}$. We can also define composition, tensor product and adjoint of actions in a natural way via composition, tensor product and adjoint of linear maps which induce the actions[6]. We will use the same symbols for operations on actions as those for linear maps.

In this paper, a *property* of a quantum system with state space $\mathcal{H}$ is just a subset of $\Sigma_{\mathcal{H}}$. However, according to quantum theory, not any subset of $\Sigma_{\mathcal{H}}$ represents a property of the system that can be tested. A property is *testable* iff it corresponds to a closed linear subspace W of $\mathcal{H}$ in such a way that the states in the property are exactly those generated by vectors in W. Since this correspondence is one-to-one and natural, we will always use the same symbol to denote a testable property and its corresponding closed linear subspace. Moreover, according to linear algebra, closed linear subspaces lie in one-to-one correspondence with projectors in the following sense:

1. For every projector A on $\mathcal{H}$, $\mathrm{ran}(A)$ (the range of A) is a closed linear subspace of $\mathcal{H}$, and for every vector $|\psi\rangle \in \mathcal{H}$, $|\psi\rangle \in \mathrm{ran}(A)$ iff $A(|\psi\rangle) = |\psi\rangle$.
2. For every closed linear subspace W of $\mathcal{H}$, there is a *unique* projector on $\mathcal{H}$, called *the projector onto* W and denoted by $?^{\mathcal{H}}(W)$, such that for every vector $|\psi\rangle \in \mathcal{H}$, $|\psi\rangle \in W$ iff $?^{\mathcal{H}}(W)(|\psi\rangle) = |\psi\rangle$.

The state space of a qubit, the unit of quantum information, is of dimension 2. Given a fixed orthonormal basis $\{|0\rangle, |1\rangle\}$ of the state space of a qubit, the two states generated by $|0\rangle$ and $|1\rangle$, respectively, correspond to the values 0 and 1 of a classical bit. Given several qubits indexed by elements in a finite set I, they form a compound quantum system, and the state space for I is the tensor product $\bigotimes_{i\in I} \mathcal{H}_i$ of the state space $\mathcal{H}_i$ for each qubit $i \in I$. A standard way of obtaining an orthonormal basis of this state space is to take tensor products of vectors in the fixed orthonormal bases of each $\mathcal{H}_i$. It is easy to see that there

[6] Note that different linear maps could induce the same action, but the operations on actions are still well-defined according to linear algebra.

are $2^{|I|}$ vectors in this basis, and we will index them by elements in ${}^{I}\mathbf{2}$, the set of all functions from I to $\mathbf{2} = \{0, 1\}$, in such a way that $|f\rangle = \otimes_{i \in I} |f(i)\rangle_i$, for each $f \in {}^{I}\mathbf{2}$. We call a state of a compound system *classical* if it is generated by a vector in this basis. Moreover, we write $|0\rangle_I$ for $\otimes_{i \in I} |0\rangle_i$.

It is well known that an n-dimensional Hilbert space is isomorphic to $\mathbb{C}^n$. In this case, every linear subspace is closed and every operator is bounded. Moreover, every state can be represented by n complex numbers if we pick a vector in the state as its representative. Every property, identified with its corresponding projector, can be represented by an $n \times n$ matrix of complex numbers. Every linear map from an n-dimensional Hilbert space to an m-dimensional one can be represented by an $m \times n$ matrix of complex numbers.

In this paper, for generality, we assume that we are supplied with countably infinitely many qubits indexed by elements in ω, the set of all natural numbers, which we take to be non-negative integers. We further assume that an orthonormal basis has been fixed for each qubit, and we obtain an orthonormal basis for compound systems consisting of a finite number of qubits by applying the tensor product in the way just described. Finally, we use $\mathcal{P}_{\text{fin}}(\omega)$ to denote the set of all finite, *non-empty* subsets of ω. For each $\tau \in \mathcal{P}_{\text{fin}}(\omega)$, by *$\tau$-system* we mean the quantum system consisting of qubits indexed by elements of τ. Whenever $\mathcal{H}_\tau$, the state space of the τ-system, appears as a superscript or subscript in a symbol, we simply write τ; for example, we write simply Σ_τ for $\Sigma_{\mathcal{H}_\tau}$.

Moreover, for each $\tau, \rho \in \mathcal{P}_{\text{fin}}(\omega)$ s.t. $\tau \subseteq \rho$, we know from linear algebra that $\mathcal{H}_\tau$ can be canonically embedded into $\mathcal{H}_\rho$, by "padding" all the vectors with $|0\rangle$'s for all the extra dimensions. Hence in this paper we write $\Theta_{\tau \to \rho} : \mathcal{H}_\tau \to \mathcal{H}_\rho$ for this canonical embedding

$$\Theta_{\tau \to \rho} = \textstyle\sum_{f \in {}^{\tau}\mathbf{2}} (|f\rangle \otimes |0\rangle_{\rho \setminus \tau}) \langle f|.$$

We also write $\Theta_{\rho \to \tau} : \mathcal{H}_\rho \to \mathcal{H}_\tau$ for the canonical projection that reverses the above embedding:

$$\Theta_{\rho \to \tau} = \textstyle\sum_{f \in {}^{\tau}\mathbf{2}} |f\rangle \, (\langle f| \otimes \langle 0|_{\rho \setminus \tau}).$$

Using the canonical embeddings and projections, one can *generalize projectors to arbitrary dimensions*: For every space $\mathcal{H}_\tau$ and every closed linear subspace W_ρ of some other space $\mathcal{H}_\rho$, we can define *the generalized projector of $\mathcal{H}_\tau$ onto W_ρ*, denoted by $?^\tau(W_\rho)$, by putting:

$$?^\tau(W_\rho) = \Theta_{\rho \cup \tau \to \rho} \circ \left(?^\rho(W_\rho) \otimes |0\rangle_{\tau \setminus \rho} \langle 0|_{\tau \setminus \rho} \right) \circ \Theta_{\tau \to \rho \cup \tau}$$

This is a linear map that takes a vector in $\mathcal{H}_\tau$ and "projects" it onto W_ρ. Physically, this action corresponds to *a successful measurement of a ρ-property performed on a τ-system.*

We introduce some notation. Given a binary relation R and a set $A \subseteq \text{dom}(R) = \{x \mid \exists y. (x, y) \in R\}$, let $R[A] \stackrel{\text{def}}{=} \{b \mid \exists a \in A. (a, b) \in R\}$ be the direct image of A under R. Given a set $B \subseteq \text{ran}(R) = \{y \mid \exists x. (x, y) \in R\}$, we let $[R]B \stackrel{\text{def}}{=} \{a \mid \forall b. (a, b) \in R \Rightarrow b \in B\}$ be the so-called weakest precondition of

B under R. Note that when R is a function instead of a relation in general, $[R]B$ is sometimes called the inverse image of B under R. In general, given two sets A and B, we write ${}^AB$ for the set of functions from A to B. Given a positive number N, let $\mathbf{N} = \{0, 1, \ldots, N-1\}$. Given a linear map T, let $\mathbf{T}$ be its matrix representation under the fixed bases.

3 Quantum Probabilistic Dyadic Second-Order Logic

Syntax of QPDSOL. Our language consists of terms (for quantum states), predicates symbols (for quantum testable properties), and function symbols (for actions). The language is *typed*: each of these symbols comes with a type, which is an element of $\mathcal{P}_{\text{fin}}(\omega)$, indicating the underlying set of qubits involved in that state, property or action. E.g. terms of type τ refer to the possible (pure) states of the τ-system; predicate symbols of type τ are unary predicates referring to *quantum-testable properties* of the τ-system; function symbols of type $\tau \to \rho$ are dyadic predicates (restricted to functions) referring to *actions*. As the types range over all of $\mathcal{P}_{\text{fin}}(\omega)$, the entire domain of discourse involves infinitely many qubits; but each formula involves only finitely many types, each involving only finitely many qubits, so that a formula can only talk about finitely many qubits.

For each pair of elements $\tau, \rho \in \mathcal{P}_{\text{fin}}(\omega)$, we include in the language a countable set of *state variables* x_τ of type τ, a countable set of *state constants* c_τ of type τ, a countable set of *predicate variables* p_τ of type τ, a countable set of *predicate constants* T_τ of type τ, a countable set of *action variables* $a_{\tau\to\rho}$ of type $\tau \to \rho$, and a countable set of *action constants* $C_{\tau\to\rho}$ of type $\tau \to \rho$. It is assumed that these sets are pairwise disjoint, and that each of them is indexed by elements in ω without repetition.

Definition 3.1. *For any $\tau, \rho \in \mathcal{P}_{fin}(\omega)$, we define by (triple) mutual recursion the following sets of syntactic expressions: the set $Term_\tau$ of terms of type τ*

$$t_\tau ::= \ x_\tau \mid c_\tau \mid t_{\tau_1} \otimes t_{\tau_2} \mid \alpha_{\rho\to\tau}(t_\rho)$$

(where $\tau_1, \tau_2 \in \mathcal{P}_{fin}(\omega)$ are such that $\tau_1 \cup \tau_2 = \tau$, $\tau_1 \cap \tau_2 = \emptyset$), the set $\mathcal{P}_\tau$ of (unary) predicate symbols of type τ

$$P_\tau ::= p_\tau \mid T_\tau \mid t_\tau \mid {\sim}P_\tau \mid P_\tau \cap P_\tau \mid P_{\tau_1} \otimes P_{\tau_2} \mid \alpha_{\rho\to\tau}[P_\rho] \mid [\alpha_{\tau\to\rho}]P_\rho$$

(where $\tau_1, \tau_2 \in \mathcal{P}_{fin}(\omega)$ are such that $\tau_1 \cup \tau_2 = \tau$, $\tau_1 \cap \tau_2 = \emptyset$), and the set $\mathcal{A}_{\tau\to\rho}$ of function symbols of type $\tau \to \rho$

$$\alpha_{\tau\to\rho} ::= a_{\tau\to\rho} \mid C_{\tau\to\rho} \mid ?^\tau P_\rho \mid \alpha^\dagger_{\rho\to\tau} \mid \alpha_{\tau\to\mu}; \alpha_{\mu\to\rho} \mid \alpha_{\tau_1\to\rho_1} \otimes \alpha_{\tau_2\to\rho_2}$$

(where $\mu, \tau_1, \rho_1, \tau_2, \rho_2 \in \mathcal{P}_{fin}(\omega)$ are such that $\tau_1 \cup \tau_2 = \tau$, $\rho_1 \cup \rho_2 = \rho$ and $\tau_1 \cap \tau_2 = \rho_1 \cap \rho_2 = \emptyset$).

We write *Term* for the set $\bigcup_{\tau\in\mathcal{P}_{\text{fin}}(\omega)} Term_\tau$ of all terms, $\mathcal{P}$ for the set $\bigcup_{\tau\in\mathcal{P}_{\text{fin}}(\omega)} \mathcal{P}_\tau$ of all predicate symbols, and $\mathcal{A}$ for the set $\bigcup_{\tau,\rho\in\mathcal{P}_{\text{fin}}(\omega)} \mathcal{A}_{\tau\to\rho}$ of all function symbols. When $\tau = \rho$, we simply write $P_\rho?$ for the function symbol $?^\tau P_\rho$.

Definition 3.2. *We now define by induction the set $\mathcal{L}$ of formulas of our logic:*

$$\varphi ::= P_\tau^{\geq r}(t_\tau) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x_\tau \varphi \mid \forall p_\tau \varphi \mid \forall a_{\rho\to\tau}\varphi$$

where $\tau \in \mathcal{P}_{fin}(\omega)$, $t_\tau \in Term_\tau$, $P_\tau \in \mathcal{P}_\tau$ and $r \in [0,1]$ is a definable real number (described below before Definition 3.3).

The intended meaning of our basic formula $P_\tau^{\geq r}(t_\tau)$ is that a *quantum system in state t_τ will yield the answer 'yes'* (i.e. it will collapse to a state satisfying property P_τ) *with a probability at least r whenever a binary measurement of property P_τ is performed*. The rest of our logical formulas are built from such basic formulas using standard Boolean connectives, as well as three types of quantifiers: first-order quantifiers $\forall x_\tau$ ranging over quantum states, second-order quantifiers $\forall p_\tau$ over quantum (testable) predicates, and second-order quantifiers $\forall a_{\tau\to\rho}$ ranging over quantum actions.

The notions of free variables, bound variables, etc. are defined in the standard way. As usual, a formula $\varphi \in \mathcal{L}$ is called *closed* if it has no free (state, predicate or action) variables. A *pure* formula is a closed formula containing no (state, predicate or action) constants.

Semantics of QPDSOL. Following standard practice, we introduce the notion of *frame* (also known as *structure* in the semantics of first-order logic), by which we mean a structure that fixes the (state, predicate and action) constants. Then, given a frame, we define a *model* on it (also known as an *interpretation* in the semantics of first-order logic), which can determine the denotation of each remaining term, predicate symbol and function symbol. Finally, we define the *satisfaction relation*.

Recall that we say that a real number r is *definable* if there is a formula $\phi(x)$ in the first-order language of $(\mathbb{R}, +, \cdot, 0, 1)$ such that $(\mathbb{R}, +, \cdot, 0, 1) \models \phi[r] \wedge \forall x(\phi(x) \to x = r)$. We also say that a complex number z is *simple* if $z = a + bi$ for definable real numbers a and b. Extending the terminology, we say that a state of the τ-system, a testable property of the τ-system and an action from the τ-system to ρ-system are *definable* if they can be represented under the fixed basis respectively by a $2^{|\tau|}$-tuple (with the state identified with the representative of it), a $2^{|\tau|} \times 2^{|\tau|}$-matrix (with the closed linear subspace identified with the corresponding projector), and a $2^{|\rho|} \times 2^{|\tau|}$-matrix (with the action identified with a linear map that induces it) of simple complex numbers.

Definition 3.3. *An* $\mathcal{H}$-valuation *is a function V defined on a subset of $\mathcal{P} \cup \mathcal{A} \cup Term$ and satisfying the following conditions:*

- *$V(t_\tau) \in \Sigma_\tau$ if $t_\tau \in Term_\tau$;*
- *$V(P_\tau)$ is a testable property of τ-system, if $P_\tau \in \mathcal{P}_\tau$;*
- *$V(\alpha_{\tau\to\rho})$ is an action from Σ_τ to Σ_ρ if $\alpha_{\tau\to\rho} \in \mathcal{A}_{\tau\to\rho}$.*

Definition 3.4. *A* frame $\mathfrak{F}$ *is an $\mathcal{H}$-valuation whose domain is the set of all (state, predicate and action) constants and whose values are all definable.*

Actually, for the decidability result to hold, a frame must be a computable function in some sense. We neglect this technicality here.

Definition 3.5. *A* model $\mathfrak{M}$ *on a frame* $\mathfrak{F}$ *is an* $\mathcal{H}$*-valuation whose domain is* $\mathcal{P} \cup \mathcal{A} \cup Term$*, that extends* $\mathfrak{F}$ *and that satisfies the following, for any terms* $t_\tau, t_{\tau_1}, t_{\tau_2}$*, function symbols* $\alpha_{\tau\to\rho}, \beta_{\rho\to\mu}, \alpha_{\tau_1\to\rho_1}, \alpha_{\tau_2\to\rho_2}$*, and predicate symbols* $P_\tau, Q_\tau, P_\rho, P_{\tau_1}, Q_{\tau_2}$ *such that* $\tau_1 \cap \tau_2 = \emptyset$ *and* $\rho_1 \cap \rho_2 = \emptyset$*:*

$\mathfrak{M}(t_{\tau_1} \otimes t_{\tau_2})$	$= \mathfrak{M}(t_{\tau_1}) \otimes \mathfrak{M}(t_{\tau_2})$
$\mathfrak{M}(\alpha_{\tau\to\rho}(t_\tau))$	$= \mathfrak{M}(\alpha_{\tau\to\rho})(\mathfrak{M}(t_\tau))$
$\mathfrak{M}(\alpha_{\tau\to\rho}; \beta_{\rho\to\mu})$	$= \mathfrak{M}(\beta_{\rho\to\mu}) \circ \mathfrak{M}(\alpha_{\tau\to\rho})$
$\mathfrak{M}(\alpha^\dagger_{\tau\to\rho})$	$= (\mathfrak{M}(\alpha_{\tau\to\rho}))^\dagger$
$\mathfrak{M}(\alpha_{\tau_1\to\rho_1} \otimes \alpha_{\tau_2\to\rho_2})$	$= \mathfrak{M}(\alpha_{\tau_1\to\rho_1}) \otimes \mathfrak{M}(\alpha_{\tau_2\to\rho_2})$
$\mathfrak{M}(?^\tau P_\rho)$	$= ?^\tau(\mathfrak{M}(P_\rho))$
$\mathfrak{M}(\sim P_\tau)$	$= \sim\mathfrak{M}(P_\tau)$
$\mathfrak{M}(P_\tau \cap Q_\tau)$	$= \mathfrak{M}(P_\tau) \cap \mathfrak{M}(Q_\tau)$
$\mathfrak{M}(P_{\tau_1} \otimes Q_{\tau_2})$	$= \mathfrak{M}(P_{\tau_1}) \otimes \mathfrak{M}(Q_{\tau_2})$
$\mathfrak{M}(\alpha_{\tau\to\rho}[P_\tau])$	$= \mathfrak{M}(\alpha_{\tau\to\rho})[\mathfrak{M}(P_\tau)]$
$\mathfrak{M}([\alpha_{\tau\to\rho}]P_\rho)$	$= [\mathfrak{M}(\alpha_{\tau\to\rho})]\mathfrak{M}(P_\rho)$

To interpret quantifiers, for each (state, predicate, or action) variable v we introduce an equivalence relation $\sim_v$ among models on the same frame such that $\mathfrak{M} \sim_v \mathfrak{M}'$ iff $\mathfrak{M}(v') = \mathfrak{M}'(v')$ for all variables v' except possibly v.

Definition 3.6. *The satisfaction relation between a model* $\mathfrak{M}$ *and a formula is defined recursively, where* v *is any (state, predicate, or action) variable,*

$$
\begin{array}{lll}
\mathfrak{M} \models P^{\geq r}_\tau(t_\tau) & \iff & |\langle\psi|\, ?^\tau(\mathfrak{M}(P_\tau))\,|\psi\rangle|^2 \geq r \| |\psi\rangle\|^2 \| ?^\tau(\mathfrak{M}(P_\tau))\,|\psi\rangle\|^2, \\
 & & \textit{for any vector } |\psi\rangle \in \mathfrak{M}(t_\tau) \\
\mathfrak{M} \models \neg\varphi & \iff & \mathfrak{M} \not\models \varphi, \\
\mathfrak{M} \models \varphi \wedge \psi & \iff & \mathfrak{M} \models \varphi \textit{ and } \mathfrak{M} \models \psi, \\
\mathfrak{M} \models \forall v\varphi & \iff & \mathfrak{M}' \models \varphi, \textit{ for all } \mathfrak{M}' \sim_v \mathfrak{M}.
\end{array}
$$

Obviously, other Boolean connectives such as $\vee$, $\to$ and $\leftrightarrow$ can be defined in the usual manner. Existential quantifiers over states, predicates and actions can also be defined in the usual manner. Moreover, this logic is at least as expressive as the first-order language of the lattice $L(\mathbb{C}^{2^n})$, which is discussed in [10].

Now we introduce some useful abbreviations:

$$
\begin{array}{ll}
P^{\leq r}_\tau(t_\tau) \stackrel{\text{def}}{=} (\sim P)^{\geq(1-r)}_\tau(t_\tau) & P^{=r}_\tau(t_\tau) \stackrel{\text{def}}{=} P^{\geq r}_\tau(t_\tau) \wedge P^{\leq r}_\tau(t_\tau) \\
P^{<r}_\tau(t_\tau) \stackrel{\text{def}}{=} \neg P^{\geq r}_\tau(t_\tau) & P^{>r}_\tau(t_\tau) \stackrel{\text{def}}{=} \neg P^{\leq r}_\tau(t_\tau) \\
s_\tau \perp t_\tau \stackrel{\text{def}}{=} s^{\leq 0}_\tau(t_\tau) &
\end{array}
$$

$$
s_\tau \doteq t_\tau \stackrel{\text{def}}{=} \big[s^{=1}_\tau(t_\tau) \wedge \neg(s_\tau \perp t_\tau)\big] \vee \big[(s_\tau \perp s_\tau) \wedge (t_\tau \perp t_\tau)\big]
$$

Essentially, the meaning of $P^{\leq r}_\tau(t_\tau)$ (or respectively $P^{=r}_\tau(t_\tau)$, $P^{<r}_\tau(t_\tau)$, $P^{>r}_\tau(t_\tau)$) is that a quantum system in state t_τ will yield the answer 'yes' (i.e. it will collapse to a state satisfying property P_τ) with a probability $\leq r$ (or respectively $= r$, $< r$, $> r$) whenever a binary measurement of property P_τ is performed. Moreover, $\mathfrak{M} \models s_\tau \perp t_\tau$ iff s_τ and t_τ denote two orthogonal states. (Note that

the impossible state $\widehat{\mathbf{0}_\tau}$ is the only state that is orthogonal to itself.) Finally, we have $\mathfrak{M} \models s_\tau \doteq t_\tau$ iff s_τ and t_τ refer to *the same state*: the first disjunct ensures that s_τ and t_τ are equal but neither denotes $\widehat{\mathbf{0}_\tau}$ (note that $s_\tau^{=1}(t_\tau)$ and $s_\tau \perp t_\tau$ are together satisfiable where either s_τ or t_τ is interpreted by $\widehat{\mathbf{0}_\tau}$), while the second disjunct ensures that both s_τ and t_τ denote $\widehat{\mathbf{0}_\tau}$.

We now define the notion of validity.

Definition 3.7. *A formula φ of $\mathcal{L}$ is said to be valid in a frame $\mathfrak{F}$, written $\mathfrak{F} \models \varphi$, if $\mathfrak{M} \models \varphi$ for every model $\mathfrak{M}$ on $\mathfrak{F}$. A formula φ of $\mathcal{L}$ is said to be valid, written $\models \varphi$, if $\mathfrak{F} \models \varphi$ for every frame $\mathfrak{F}$.*

As in classical predicate logic, we have

Lemma 3.8. *For every* closed *formula φ in $\mathcal{L}$ and every frame $\mathfrak{F}$, $\mathfrak{F} \models \varphi$ iff there is a model $\mathfrak{M}$ on $\mathfrak{F}$ such that $\mathfrak{M} \models \varphi$. For every* pure *formula φ in $\mathcal{L}$, $\models \varphi$ iff there is a frame $\mathfrak{F}$ such that $\mathfrak{F} \models \varphi$.*

4 Examples

Here we show how our language can be used to express many properties of quantum algorithms. We start with introducing some notation that will be commonly used in the following examples.

First, for each qubit i, we introduce state constants 0_i and 1_i to denote the state generated by $|0\rangle_i$ and $|1\rangle_i$, respectively.

We furthermore have the following action constants for a single qubit i, and for some, we provide the matrix representation (in the fixed bases) of linear maps which are usually used to induce the actions interpreting these constants:

- I_i interpreted as the identity action,
- H_i the action induced by the Hadamard gate with matrix $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
- X_i the action induced by the Pauli X gate $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
- Z_i the action induced by the Pauli Z gate $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

We furthermore have an action symbol $CNOT_{ij}$ $(i \neq j)$ for the *controlled-NOT* action with control qubit i and target qubit j usually induced by a linear map with the matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

For any distinct i and j, we also define an abbreviation for an action that interchanges the states of qubits i and j:

$$FP_{ij} \stackrel{\text{def}}{=} CNOT_{ij}; CNOT_{ji}; CNOT_{ij}$$

We introduce an abbreviation $CS_\tau(t_\tau)$ for the formula saying that a state t_τ is a classical state:

$$CS_\tau(t_\tau) \stackrel{\text{def}}{=} \exists\{x_i \,|\, i \in \tau\}\Big(t_\tau \doteq \otimes_{i\in\tau} x_i \wedge \bigwedge_{i\in\tau}(x_i \doteq 0_i \vee x_i \doteq 1_i)\Big),$$

where $\exists\{x_i \,|\, i \in \tau\}$ means a sequence of existential quantifiers on state variables of type $i \in \tau$. Similarly, we introduce an abbreviation $Unit(\alpha_{\tau\to\tau})$ for the formula saying that the variable $\alpha_{\tau\to\tau}$ denotes (an action induced by) a unitary operator on a τ-system:

$$Unit(\alpha_{\tau\to\tau}) \stackrel{\text{def}}{=} \forall x_\tau(\alpha_{\tau\to\tau}; \alpha^\dagger_{\tau\to\tau}(x_\tau) \doteq x_\tau).$$

Next, we write $H^{\otimes\tau}$ for $\otimes_{i\in\tau} H_i$ and $I^{\otimes\tau}$ for $\otimes_{i\in\tau} I_i$. Finally, we recursively introduce an abbreviation $\alpha^n_{\tau\to\tau}$ for the action obtained by iterating the action $\alpha_{\tau\to\tau}$ for n times:

$$\begin{aligned} \alpha^0_{\tau\to\tau} &= I^{\otimes\tau} \text{ (the identity map on } \tau\text{-system)} \\ \alpha^{n+1}_{\tau\to\tau} &= \alpha^n_{\tau\to\tau}; \alpha_{\tau\to\tau} \text{ (for } n \geq 1) \end{aligned}$$

4.1 Quantum Teleportation

In quantum teleportation, Alice and Bob, who are separated by a long distance, share a pair of qubits in Bell state $\frac{1}{\sqrt{2}}(|0\rangle_2 |0\rangle_3 + |1\rangle_2 |1\rangle_3)$ (qubit 2 being with Alice, and 3 being with Bob). Alice would like to let Bob have a qubit whose state is the same as the state q of her qubit 1 (which we represent as a state variable of type $\{1\}$). She first interacts the qubit with her end of the Bell state. Define

$$PRE(q) \stackrel{\text{def}}{=} (CNOT_{12}; (H_1 \otimes I_2)) \otimes I_3\Big(q \otimes \big(CNOT_{23}; (H_2 \otimes I_3)(0_2 \otimes 0_3)\big)\Big).$$

She then measures her qubits 1 and 2, and depending on the result sends Bob instructions as to any further operation that must be performed on his qubit 3.

The *standard frame for Teleportation* is the frame that interprets as intended all the constants occurring in the Teleportation protocol: the constants 0_i and 1_i for each $i \in \{1, 2, 3\}$ as well as I_2, I_3, H_1, H_2 $CNOT_{12}$, $CNOT_{23}$ and FP_{13}.

The correctness of the Teleportation protocol is equivalent to the validity in its standard frame of the formula

$$\begin{aligned} &\forall q\big[(q \otimes 0_2 \otimes 0_3) \doteq (0_1? \otimes 0_2? \otimes I_3); (FP_{13} \otimes I_2)(PRE(q)) \\ &\quad \wedge (q \otimes 1_2 \otimes 0_3) \doteq (0_1? \otimes 1_2? \otimes I_3); (I_1 \otimes I_2 \otimes X_3); (FP_{13} \otimes I_2)(PRE(q)) \\ &\quad \wedge (q \otimes 0_2 \otimes 1_3) \doteq (1_1? \otimes 0_2? \otimes I_3); (I_1 \otimes I_2 \otimes Z_3); (FP_{13} \otimes I_2)(PRE(q)) \\ &\quad \wedge (q \otimes 1_2 \otimes 1_3) \doteq (1_1? \otimes 1_2? \otimes I_3); (I_1 \otimes I_2 \otimes (X_3; Z_3)); (FP_{13} \otimes I_2)(PRE(q))\big] \end{aligned}$$

4.2 Quantum Search Algorithm

In the search problem, we are given a unitary operator O, which is usually called an *oracle*, acting on $N+1$ qubits (we assume them to be indexed by elements in $\mathbf{N+1}$), such that there is a classical state $|f_0\rangle$ with the property that, for each classical state $|f\rangle$ and $b \in \mathbf{2}$,

$$O(|f\rangle \otimes |b\rangle_N) = \begin{cases} |f\rangle \otimes |1-b\rangle_N, & \text{if } f = f_0, \\ |f\rangle \otimes |b\rangle_N, & \text{if } f \in {}^{\mathbf{N}}\mathbf{2} \setminus \{f_0\} \end{cases} \quad (1)$$

The aim of the algorithm is to find out the classical state $|f_0\rangle$.

To formalize the correctness of this algorithm, we use an action variable O of type $\mathbf{N+1} \to \mathbf{N+1}$ to denote the oracle. Moreover, we assume that we have an action constant $PS_{\mathbf{N}}$ of type $\mathbf{N} \to \mathbf{N}$ for the action induced by the conditional phase shift gate on the first N qubits, whose matrix under the fixed basis is the following:

$$\begin{bmatrix} \mathbf{Z} & \mathbf{O}_{2\times(N-2)} \\ \mathbf{O}_{(N-2)\times 2} & -\mathbf{I}_{(N-2)\times(N-2)} \end{bmatrix}$$

Here $\mathbf{O}_{2\times(N-2)}$ is the 2 by $N-2$ matrix of only 0 entries, and similarly for $\mathbf{O}_{(N-2)\times 2}$, and $\mathbf{I}_{(N-2)\times(N-2)}$ is the $N-2$ by $N-2$ identity matrix.

As before, the *standard frame* for the $(N+1)$-qubit quantum search algorithm is the one that interprets as intended all the above constants, as well as all the constants 0_i and 1_i. For convenience, we make the following abbreviation

$$\begin{aligned} &Oracle_{\mathbf{N+1}}(O) \stackrel{\text{def}}{=} Unit(O) \wedge \exists x_{\mathbf{N}} \Big[CS_{\mathbf{N}}(x_{\mathbf{N}}) \wedge \forall y_{\mathbf{N}} \Big(CS_{\mathbf{N}}(y_{\mathbf{N}}) \\ &\to (x_{\mathbf{N}} \doteq y_{\mathbf{N}} \to O(y_{\mathbf{N}} \otimes 0_{N+1}) \doteq y_{\mathbf{N}} \otimes 1_{N+1} \wedge O(y_{\mathbf{N}} \otimes 1_{N+1}) \doteq y_{\mathbf{N}} \otimes 0_{N+1}) \\ &\wedge (x_{\mathbf{N}} \perp y_{\mathbf{N}} \to O(y_{\mathbf{N}} \otimes 0_{N+1}) \doteq y_{\mathbf{N}} \otimes 0_{N+1} \wedge O(y_{\mathbf{N}} \otimes 1_{N+1}) \doteq y_{\mathbf{N}} \otimes 1_{N+1}) \Big) \Big] \end{aligned}$$

for the formula saying that O is an action induced by an oracle acting on the $(\mathbf{N+1})$-system satisfying Eq.(1).

The correctness of $(N+1)$-qubit Quantum Search Algorithm (with $N > 2$) is equivalent to the validity in its standard frame of the following formula:

$$\begin{aligned} &\forall O \forall x_{\mathbf{N}} \Big\{ Oracle_{\mathbf{N+1}}(O) \wedge CS_{\mathbf{N}}(x_{\mathbf{N}}) \\ &\wedge O(x_{\mathbf{N}} \otimes 0_N) \doteq x_{\mathbf{N}} \otimes 1_N \wedge O(x_{\mathbf{N}} \otimes 1_N) \doteq x_{\mathbf{N}} \otimes 0_N \to (x_{\mathbf{N}} \otimes H_N(1_N))^{>0.5} \\ &\Big(H^{\otimes(\mathbf{N+1})}; (O; ((H^{\otimes \mathbf{N}}; PS_{\mathbf{N}}; H^{\otimes \mathbf{N}}) \otimes I_N))^K (0_{\mathbf{N}} \otimes 1_N) \Big) \Big\}, \end{aligned}$$

where K is the largest natural number less than $\frac{\pi}{4}\sqrt{2^N}$.

4.3 Deutsch-Josza Algorithm

In the Deutsch-Josza problem, we are given a unitary operator O (usually called an oracle) acting on $N+1$ qubits (we assume them to be indexed by elements in $\mathbf{N+1}$), which is known to satisfy one of the following properties:

(i) The oracle is *constant* (having the same value for all inputs): there is $i \in \{0,1\}$ s.t. $O(|f\rangle \otimes |b\rangle_N) = |f\rangle \otimes |b \oplus i\rangle_N$ for all $b \in \mathbf{2}$ and classical state $|f\rangle$, with $f \in {}^{\mathbf{N}}\mathbf{2}$;
(ii) The oracle is *balanced* (equal to 1 for exactly half of all the possible inputs, and 0 for the other half): there is $X \subseteq {}^{\mathbf{N}}\mathbf{2}$ s.t. $|X| = 2^{N-1}$ and $O(|f\rangle \otimes |b\rangle_N)$ is $|f\rangle \otimes |1-b\rangle_N$ if $f \in X$, and is $|f\rangle \otimes |b\rangle_N$, otherwise, for all $b \in \mathbf{2}$.

The goal of the algorithm is to determine which of the two properties holds for O.

To formalize the correctness of this algorithm, we use an action variable O of type $\mathbf{N}+\mathbf{1} \to \mathbf{N}+\mathbf{1}$ to denote the oracle. For convenience, we introduce some abbreviations: first, let us denote by $ConOra(O)$ the formula

$$\begin{aligned}
&Unit(O) \wedge \Big[\\
&\forall x_{\mathbf{N}} \Big(CS_{\mathbf{N}}(x_{\mathbf{N}}) \to O(x_{\mathbf{N}} \otimes 0_{N+1}) \doteq x_{\mathbf{N}} \otimes 0_{N+1} \wedge O(x_{\mathbf{N}} \otimes 1_{N+1}) \doteq x_{\mathbf{N}} \otimes 1_{N+1} \Big) \\
&\vee \forall x_{\mathbf{N}} \Big(CS_{\mathbf{N}}(x_{\mathbf{N}}) \to O(x_{\mathbf{N}} \otimes 0_{N+1}) \doteq x_{\mathbf{N}} \otimes 1_{N+1} \wedge O(x_{\mathbf{N}} \otimes 1_{N+1}) \doteq x_{\mathbf{N}} \otimes 0_{N+1} \Big) \Big]
\end{aligned}$$

saying that O is an action induced by a constant oracle; second, we denote by $BalOra(O)$ the formula (where $k = 2^{N-1}$)

$$\begin{aligned}
&Unit(O) \wedge \exists x^1_{\mathbf{N}} ... \exists x^k_{\mathbf{N}} \Big[\Big(\bigwedge_{i=1}^{k} CS_{\mathbf{N}}(x^i_{\mathbf{N}}) \Big) \wedge \Big(\bigwedge_{1 \le i < j \le k} x^i_{\mathbf{N}} \perp x^j_{\mathbf{N}} \Big) \wedge \forall y_{\mathbf{N}} \Big(CS_{\mathbf{N}}(y_{\mathbf{N}}) \to \\
&\Big(\bigvee_{i=1}^{k} y_{\mathbf{N}} \doteq x^i_{\mathbf{N}} \to O(y_{\mathbf{N}} \otimes 0_{N+1}) \doteq y_{\mathbf{N}} \otimes 1_{N+1} \wedge O(y_{\mathbf{N}} \otimes 1_{N+1}) \doteq y_{\mathbf{N}} \otimes 0_{N+1} \Big) \\
&\wedge \Big(\bigwedge_{i=1}^{k} y_{\mathbf{N}} \perp x^i_{\mathbf{N}} \to O(y_{\mathbf{N}} \otimes 0_{N+1}) \doteq y_{\mathbf{N}} \otimes 0_{N+1} \wedge O(y_{\mathbf{N}} \otimes 1_{N+1}) \doteq y_{\mathbf{N}} \otimes 1_{N+1} \Big) \Big) \Big) \Big]
\end{aligned}$$

saying that O is an action induced by a balanced oracle.

Finally, the *correctness of the $(N+1)$-qubit Deutsch-Jozsa algorithm* (for any natural number N) is equivalent to the assertion that the following formula is valid in its standard frame:

$$\begin{aligned}
&\forall O \Big\{ ConOra(O) \vee BalOra(O) \to \\
&\quad \Big[\Big(ConOra(O) \leftrightarrow H^{\otimes(\mathbf{N}+\mathbf{1})}; O; H^{\otimes(\mathbf{N}+\mathbf{1})}(0_{\mathbf{N}} \otimes 1_N) \doteq 0_{\mathbf{N}} \otimes 1_N \Big) \\
&\quad \wedge \Big(BalOra(O) \leftrightarrow H^{\otimes(\mathbf{N}+\mathbf{1})}; O; H^{\otimes(\mathbf{N}+\mathbf{1})}(0_{\mathbf{N}} \otimes 1_N) \perp 0_{\mathbf{N}} \otimes 1_N \Big) \Big] \Big\}
\end{aligned}$$

5 Decidability

The set of validities of **QPDSOL** on any *given frame* is decidable. Using the same proof strategy, the validity problem for *pure* formulas over (the class of) *all frames* is also decidable. In this section, we sketch the proofs of these results.

The basic technique for proving these decidability results is a generalization and extension of the method used in [10]: We express validity of formulas of $\mathcal{L}$ without free variables in a given frame $\mathfrak{F}$ via truth of first-order sentences of $(\mathbb{R}, +, \cdot, 0, 1)$; then the decidability of our logic follows from Tarski's theorem in [17] which states that the first-order theory of $(\mathbb{R}, +, \cdot, 0, 1)$ is decidable. This idea is unfolded into several technical steps.

In the first step, we need to deal with intersection of testable properties. For a function symbol of the form $(P_\tau \cap Q_\tau)?$, it is well known that calculating the matrix of the corresponding projector typically involves a process of taking limits and hence can not be expressed in the first-order theory of $(\mathbb{R}, +, \cdot, 0, 1)$. The key to solving this is the observation that complex predicate symbols, i.e. those built with $\cap$, $\otimes$, $\sim$ and other operations, can be recursively eliminated from our language with the help of quantifiers (over states). Let $\mathcal{L}^*$ be the result of this translation. Its formulas consist of those built as follows (where constraints on the types are those given in Definition 3.1 and 3.2, but with the additional requirement that for each singleton $\tau = \{i\}$, there exists a constant 0_τ that denotes $\widehat{|0\rangle}_i$, so as to facilitate the translation of generalized projectors):

$$\begin{aligned} t_\tau &::= x_\tau \mid c_\tau \mid x_{\tau_1} \otimes x_{\tau_2} \mid \alpha_{\rho\to\tau}(x_\rho) \\ P_\tau &::= p_\tau \mid T_\tau \\ \alpha_{\tau\to\rho} &::= a_{\tau\to\rho} \mid C_{\tau\to\rho} \mid a^\dagger_{\rho\to\tau} \mid a_{\tau_1\to\rho_1} \otimes a'_{\tau_2\to\rho_2} \mid P_\tau? \\ \varphi &::= x^{<r}_\tau(t_\tau) \mid x^{=r}_\tau(t_\tau) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x_\tau \varphi \mid \forall p_\tau \varphi \mid \forall a_{\rho\to\tau}\varphi \end{aligned}$$

With the possible exception of the constants 0_τ, we have that $\mathcal{L}^* \subseteq \mathcal{L}$, and the semantics of $\mathcal{L}^*$ is the same as for $\mathcal{L}$. One can define a function $\nabla : \mathcal{L} \to \mathcal{L}^*$ by recursion (and hence it is computable) s.t. $\mathfrak{M} \models \varphi \Leftrightarrow \mathfrak{M} \models \nabla(\varphi)$ for every model $\mathfrak{M}$. To illustrate why this is the case and how it helps to solve the problem, we exhibit one case in its definition:

$$\begin{aligned} \nabla(x^{=r}_\tau((P_\tau \cap Q_\tau)?(t_\tau))) = \exists y_\tau \exists z_\tau \big[\ & \nabla(t_\tau \doteq y_\tau \oplus z_\tau) \wedge y^{=0}_\tau(z_\tau) \wedge x^{=r}_\tau(y_\tau) \\ & \wedge \forall u_\tau \big(\nabla(P^{=1}_\tau(u_\tau)) \wedge \nabla(Q^{=1}_\tau(u_\tau)) \to z^{=0}_\tau(u_\tau)\big)\ \big] \end{aligned}$$

where x_τ is a state variable, t_τ is a term and $t_\tau \doteq y_\tau \oplus z_\tau$ is defined to be $\forall v_\tau(v^{=0}_\tau(y_\tau) \wedge v^{=0}_\tau(z_\tau) \to v^{=0}_\tau(t_\tau))$, which means that t_τ "lies on the plane generated by" y_τ and z_τ.

In the second step, we define for each frame $\mathfrak{F}$, a function $TR_\mathfrak{F} : \mathcal{L}^* \to \mathcal{L}_\mathbb{C}$, where $\mathcal{L}_\mathbb{C}$ is the first-order language of $(\mathbb{C}, +, \cdot, \bar{\cdot}, \prec, \mathbb{C})$, where $\bar{\cdot}$ is the conjugate operator, $\prec$ is a binary relation between complex numbers such that $a+bi \prec c+di$ iff $a < c$, and the last component $\mathbb{C}$ is the set of numbers named by a constant. Towards this aim, we first formalize in $\mathcal{L}_\mathbb{C}$ the matrix representation of the interpretation in $\mathfrak{F}$ of terms, predicate symbols and function symbols. This is possible because every term, predicate symbol and function symbol involves only finitely many qubits indicated by its type. In fact, one can define by recursion a computable function $\mathfrak{F}^\natural$ from the set of terms, predicate symbols and function symbols that can occur in formulas in $\mathcal{L}^*$ to the set of finite sets of terms in

$\mathcal{L}_{\mathbb{C}}$. For the base case, we define $\mathfrak{F}^\sharp(x_\tau)$, $\mathfrak{F}^\sharp(p_\tau)$ and $\mathfrak{F}^\sharp(a_{\tau\to\rho})$ to be the sets of variables indexed by $^\tau\mathbf{2}$, $^\tau\mathbf{2} \times^\tau \mathbf{2}$ and $^\rho\mathbf{2} \times^\tau \mathbf{2}$ in such a way that different state, predicate or action variables are mapped to disjoint sets of variables. Moreover, $\mathfrak{F}^\sharp(c_\tau)$, $\mathfrak{F}^\sharp(T_\tau)$ and $\mathfrak{F}^\sharp(C_{\tau\to\rho})$ are indexed in a similar way but they are sets of constants. Care must be taken to ensure that the constants are defined according to the interpretation in $\mathfrak{F}$. For complex symbols built with operations, we can mimic the manipulation of vectors and matrices. For example, assume that we have defined $\mathfrak{F}^\natural(x_\tau)$ to be the set of variables $\{\mathsf{x}[f] \mid f \in^\tau \mathbf{2}\}$ and $\mathfrak{F}^\natural(y_\rho)$ to be $\{\mathsf{y}[g] \mid g \in^\rho \mathbf{2}\}$ respectively, then we can mimic the Kronecker product of matrices and define $\mathfrak{F}^\natural(x_\tau \otimes y_\rho)$ to be the set of terms $\{\mathsf{x} \otimes \mathsf{y}[h] \mid h \in^{\tau\cup\rho} \mathbf{2}\}$ s.t. $\mathsf{x} \otimes \mathsf{y}[h] = \mathsf{x}[h\restriction\tau] \cdot_{\mathbb{C}} \mathsf{y}[h\restriction\rho]$, where $\cdot_{\mathbb{C}}$ is the symbol for multiplication in $\mathcal{L}_{\mathbb{C}}$. Using the function $\mathfrak{F}^\natural$, we proceed to define $TR_{\mathfrak{F}}$ in such a way that given a model $\mathfrak{M}$ on the frame $\mathfrak{F}$, $\mathfrak{M} \models \varphi$ iff $(\mathbb{C}, +, \cdot, \bar{\cdot}, \prec, \mathbb{C}) \models_{\mathfrak{M}} TR_{\mathfrak{F}}(\varphi)$, for every $\varphi \in \mathcal{L}^*$. Here the subscript in "$\models_{\mathfrak{M}}$" is an interpretation (added to the structure $(\mathbb{C}, +, \cdot, \bar{\cdot}, \prec, \mathbb{C})$) of the free variables in $TR_{\mathfrak{F}}(\varphi)$ according to the model $\mathfrak{M}$. In defining $TR_{\mathfrak{F}}$ as such, care is taken in order to verify that quantification over (finitely many) variables in $\mathfrak{F}^\sharp(x_\tau)$, $\mathfrak{F}^\sharp(p_\tau)$ or $\mathfrak{F}^\sharp(a_{\tau\to\rho})$ in the input formula really corresponds to quantification of x_τ, p_τ or $a_{\tau\to\rho}$ in the translated formula.

In the third step, we focus on the behaviour of $TR_{\mathfrak{F}}$ on the set of closed formulas. Since the definition of frames ensures that the matrix representation of the interpretation of constant symbols only has simple complex numbers as entries, the translation $TR_{\mathfrak{F}}(\varphi)$ of a closed formula φ of $\mathcal{L}^*$ in a given frame $\mathfrak{F}$ is actually a first-order sentence of $(\mathbb{C}, +, \cdot, \bar{\cdot}, \prec, \mathcal{S})$, where $\mathcal{S}$ is the set of simple complex numbers (see page 72). A consequence of this is that pure formulas of $\mathcal{L}$ are translated via $TR_{\mathfrak{F}}$ into first-order sentences of $(\mathbb{C}, +, \cdot, \bar{\cdot}, \prec)$, because there are no constants in a pure formula. Therefore, by Lemma 3.8 and the property of $TR_{\mathfrak{F}}$ by definition, we know that on a given frame $\mathfrak{F}$, $\mathfrak{F} \models \varphi$ iff $(\mathbb{C}, +, \cdot, \bar{\cdot}, \prec, \mathcal{S}) \models TR_{\mathfrak{F}} \circ \nabla(\varphi)$, for every closed formula $\varphi \in \mathcal{L}$.

The final step is to reduce the first-order theory of $(\mathbb{C}, +, \cdot, \bar{\cdot}, \prec, \mathcal{S})$ to the first-order theory of the reals. This is a simple translation, where each simple complex number is mapped to a pair of definable real numbers, and addition and multiplication are mapped according to complex arithmetic. Thus the decidability of our logic follows from these reductions and Tarksi's theorem. To summarize, we have the following decidability result.

Theorem 5.1. *The set $\{\varphi \in \mathcal{L} \mid \varphi$ is closed and $\mathfrak{F} \models \varphi\}$ is decidable, for any given frame $\mathfrak{F}$. Moreover, the set $\{\varphi \in \mathcal{L} \mid \varphi$ is pure and $\models \varphi\}$ is decidable.*

6 Conclusions

This paper extends decidability results from [10] and [4] to a language that is much more versatile in its ability to express quantum algorithms and their correctness. Our techniques can be applied to a wider range of quantum logics, giving a general recipe for showing decidability as long as definability of the sentences and operators can be done along the lines presented in this paper.

In addition we have described how to express the correctness of Quantum Teleportation, the Quantum Search algorithm and the Deutsch-Josza algorithm; however this is not an exhaustive list of algorithms whose correctness can be expressed in our language. The Fourier transform can easily be expressed in our language and this may lead to a wealth of further examples, notably those involving the hidden subgroup problem, such as order-finding and factoring; however we leave these for future work. Other future tasks involve finding a complete axiomatization and determining the complexity of the decision procedure.

References

1. Abramsky, S., Coecke, B.: A categorical semantics of quantum protocols. In: Proceedings of the 19th IEEE Conference on Logic in Computer Science (LiCS 2004), pp. 415–425. IEEE Press (2004)
2. Aerts, D.: Description of compound physical systems and logical interaction of physical systems. In: Beltrametti, E., van Fraassen, B. (eds.) Current Issues on Quantum Logic, pp. 381–405. Kluwer Academic (1981)
3. Baltag, A., Bergfeld, J., Kishida, K., Sack, J., Smets, S., Zhong, S.: PLQP & company: Decidable logics for quantum algorithms. Submitted to the International Journal of Theoretical Physics (2013)
4. Baltag, A., Bergfeld, J., Kishida, K., Sack, J., Smets, S., Zhong, S.: A Decidable Dynamic Logic for Quantum Reasoning. In: EPTCS (2012) (in print)
5. Baltag, A., Smets, S.: Complete Axiomatizations for Quantum Actions. International Journal of Theoretical Physics 44(12), 2267–2282 (2005)
6. Baltag, A., Smets, S.: LQP: The Dynamic Logic of Quantum Information. Mathematical Structures in Computer Science 16(3), 491–525 (2006)
7. Birkhoff, G., von Neumann, J.: The Logic of Quantum Mechanics. The Annals of Mathematics 37, 823–843 (1936)
8. Chadha, R., Mateus, P., Sernadas, A., Sernadas, C.: Extending classical logic for reasoning about quantum systems. In: Engesser, K., Gabbay, D.M., Lehmann, D. (eds.) Handbook of Quantum Logic and Quantum Structures: Quantum Logic, pp. 325–371. Elsevier (2009)
9. Dalla Chiara, M.L., Giuntini, R., Greechie, R.: Reasoning in quantum theory: sharp and unsharp quantum logics. Trends in logic, vol. 22. Kluwer Acadamic Press, Dordrecht (2004)
10. Dunn, J.M., Hagge, T.J., Moss, L.S., Wang, Z.: Quantum Logic as Motivated by Quantum Computing. The Journal of Symbolic Logic 70(2), 353–359 (2005)
11. Henkin, L.: Completeness in the Theory of Types. The Journal of Symbolic Logic 15, 81–91 (1950)
12. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
13. Piron, C.: Foundations of Quantum Physics. W.A. Benjamin Inc. (1976)
14. Rabin, M.: Decidability of second order theories and automata on infinite trees. Transactions of the American Mathematical Society, 1–35 (1969)
15. Randall, C., Foulis, D.: Tensor products of quantum logics do not exist. Notices Amer. Math. Soc. 26(6) (1979)
16. Selinger, P.: Towards a quantum programming language. Mathematical Structures in Computer Science 14, 527–586 (2004)
17. Tarski, A.: A Decision Method for Elementary Algebra and Geometry. RAND Corporation, Santa Monica (1948)

Structural Extensions of Display Calculi: A General Recipe

Agata Ciabattoni and Revantha Ramanayake

Vienna University of Technology
{agata,revantha}@logic.at

Abstract. We present a systematic procedure for constructing cut-free display calculi for axiomatic extensions of a logic via structural rule extensions. The sufficient conditions for the procedure are given in terms of (purely syntactic) abstract properties of the display calculus and thus the method applies to large classes of calculi and logics. As a case study, we present cut-free calculi for extensions of well-known logics including Bi-intuitionistic and tense logic.

1 Introduction

Driven by the rising demand of researchers and practitioners, the last decades have witnessed a tremendous growth in research on logics different from classical logic and also the definition of many new logics. The usefulness of these logics and the key to their application often lies in the existence of *analytic calculi*, that is, calculi in which proofs proceed by stepwise decomposition of the formulae to be proved. Indeed, analytic calculi have been widely applied to establish fundamental properties of the logics and are themselves the focus of much research.

Since its introduction by Gentzen [8], the sequent calculus has been the favourite framework for defining analytic calculi. However, this framework is not powerful enough to formalise all interesting logics. For this reason a huge range of extensions of the sequent calculus have been introduced, in many cases for the sole purpose of obtaining analytic calculi for particular logics. General and well-known formalisms include hypersequents [1], bunched calculi [13], labelled deductive systems [7,16] and the calculus of structures [12]. Since the construction of an analytic calculus is often tailored to the specific logic under consideration, a large numbers of papers in the literature deal with this topic and yet many interesting logics still lack an analytic calculus. The display calculus [2] is a powerful and semantic-independent formalism that can be used to capture a variety of different logics ranging from resource-oriented logics [4] to substructural [9] and temporal logics [14]. The beauty of the display calculus lies in a general cut-elimination theorem for all calculi obeying eight easily verifiable syntactic conditions [2,17]; this makes the display calculus a good candidate for capturing large classes of logics in a unified way, irrespective of their semantics or connectives.

Nonclassical logics are often introduced by adding properties — expressed as Hilbert axioms — to known systems. Systematic procedures to automate the construction of new analytic calculi from such Hilbert axioms are highly desirable. In this direction e.g. [5,6,16,14,15,11] introduce methods to extract rules out of suitable Hilbert axioms.

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 81–95, 2013.

More precisely [5,6] generate sequent and hypersequent rules, [11] nested sequent rules, [15] sequent rules for certain modal axioms, and [16] labelled rules; finally [14] transforms suitable modal and tense axioms (called primitive tense axioms) into structural rules for the display calculus. [14] also provides a characterisation as it is shown that each such rule added to the base system is equivalent to the extension of the logic by primitive tense axioms.

All the above results start with a *specific logic* and introduce calculi for (some of) its axiomatic extensions, e.g., Full Lambek calculus with exchange FLe for [5,6], or the tense logic Kt for [14,11]. This paper proposes instead a recipe that utilises the more common Hilbert axioms to construct analytic calculi[1]. In particular, given a suitable base calculus for a logic, we identify a hierarchy of axiom classes — computed as a function of the invertible (logical) introduction rules of the base calculus — and show how to translate axioms from suitable classes into equivalent structural rules. More invertible rules in the base calculus lead to larger sets of axioms in each suitable class, and then to the construction of cut-free calculi for more logics. In the case of intermediate logics, for example, we capture more logics than the hypersequent calculi in [5].

The emphasis is not to define such calculi for specific families of logics but to provide a *methodology* to construct them in a uniform and systematic way starting from a display calculus satisfying general conditions. Since the conditions are given in terms of (purely syntactic) abstract properties of the display calculus, the method applies to large classes of calculi and logics. As a case study, we present analytic calculi for axiomatic extensions of propositional (Bi-)intuitionistic logic, bunched, modal and tense logics. This allows for the automated introduction of (infinitely many) analytic display calculi for logics.

2 Display Calculi in a Nutshell

Given a language $\mathcal{L}$, we write $\mathbf{For}\mathcal{L}$ to denote the formulae of $\mathcal{L}$. We identify a *logic* with the set of theorems in its Hilbert calculus.

Belnap's Display Calculus [2] — introduced under the name Display Logic — generalises Gentzen's sequent calculus by supplementing the structural connective (comma) with new structural connectives. A (display) sequent $X \vdash Y$ is a tuple (X, Y) where X and Y are *structures* which are built from formulae and structure constants using the structural connectives of the calculus. Structure X (resp. Y) is called the antecedent (succedent) of the sequent. A display calculus consists of initial sequents and rules and includes the cut-rule. The rules of the calculus are usually presented as rule schemata. Concrete instances of a rule are obtained by substitution of a formula (resp. structure) for each schematic formula (structure) variable. Following standard practice, we do not explicitly distinguish between a rule and a rule schema. A derivation in the display calculus is defined in the usual way. In this paper we use $A, B, C, D, \ldots$ (possibly with subscripts) to denote formulae and $X, Y, U, V, \ldots$ to denote structures.

A *structural* rule in the display calculus is constructed from structure variables using structural connectives and structure constants. The *logical* rules usually introduce

[1] In the direction of general results, [10] shows how to extract display calculi starting from the algebraic Gaggle-theoretic semantics of a logic.

exactly one logical connective, as the primary connective in a formula that is the whole of the antecedent or succedent of the conclusion. The cut-rule has the following form, where X, Y are structures and A is a formula:

$$\frac{X \vdash A \qquad A \vdash Y}{X \vdash Y}\ cut$$

The calculus obtained by the addition of structural rules is a *structural rule extension*. A rule is *admissible* in $\mathcal{C}$ if the conclusion is derivable when the premises are derivable. A rule is *invertible* in $\mathcal{C}$ if the premises are derivable when the conclusion is derivable.

Definition 1 (equivalent rules). *Let $\mathcal{R}_0$ and $\mathcal{R}_1$ be sets of rules. We say that $\mathcal{R}_0$ and $\mathcal{R}_1$ are* equivalent *wrt $\mathcal{C}$ if each rule in $\mathcal{R}_i$ is admissible in $\mathcal{C} + \mathcal{R}_{1-i}$ for $i = 0, 1$.*

By viewing a sequent $X \vdash Y$ as the zero-premise rule with conclusion $X \vdash Y$, we can define in the obvious way what it means for two sequents to be equivalent, and for a sequent to be equivalent to a rule.

Let Z be a structure. Any structure that occurs in Z is called a *substructure* of Z. Trivially, Z is a substructure of itself. The defining feature of a display calculus is that it satisfies the display property.

Definition 2 (display property; a-part, s-part). *Let Z be an occurrence of a substructure occurring in a sequent $X \vdash Y$. Using the invertible structural rules (the 'display rules') a sequent of the form $Z \vdash U$ or $U \vdash Z$ can be derived for suitable U. In the former (resp. latter) case, the occurrence Z is said to be* displayed *as an* a-part *(*s-part*) structure.*

Since a formula is itself a structure, the display property applies to a formula occurring in a sequent but not to its proper subformulae.

A calculus is said to be *cut-eliminable* if it is possible to eliminate all occurrences of the cut-rule from a given derivation in order to obtain a *cut-free* derivation of the same sequent. A display calculus has the *subformula property* if every formula that occurs in a cut-free derivation appears as a subformula of the final sequent. An important feature of the display calculus are Belnap's conditions C1–C8 on the rules of the calculus.

- (C1) Each (schematic) formula variable occurring in a premise of a rule $\rho \neq cut$ is a subformula of some formula in the conclusion of ρ.
- (C2) *Congruent parameters* is a relation between parameters of the identical structure variable occurring in the premise and conclusion sequents of a rule.
- (C3) Each parameter is congruent to at most one structure variable in the conclusion. Ie. no two structure variables in the conclusion are congruent to each other.
- (C4) Congruent parameters are all either a-part or s-part structures.
- (C5) A formula variable in the conclusion of a rule ρ is either the entire antecedent or the entire succedent. This formula is called a *principal formula* of ρ.
- (C6/7) Each rule is closed under simultaneous substitution of arbitrary structures for congruent parameters.
- (C8) If there are rules ρ and σ with respective conclusions $X \vdash A$ and $A \vdash Y$ with formula A principal in both inferences (in the sense of C5) and if cut is applied to yield $X \vdash Y$, then either $X \vdash Y$ is identical to either $X \vdash A$ or $A \vdash Y$; or it is possible to pass from the premises of ρ and σ to $X \vdash Y$ by means of inferences falling under cut where the cut-formula always is a proper subformula of A.

Belnap's general cut-elimination theorem states that C2–C8 constitute sufficient conditions for a calculus to be cut-eliminable and C1 is the subformula property. Only condition C8 is non-trivial to check. However, C8 is not relevant for structural rules. This further motivates the interest in structural rule extensions of the display calculus.

Definition 3. *Let $\mathcal{C}$ be a display calculus and let L be a logic in the language $\mathcal{L}$. We say that $\mathcal{C}$* is a calculus for *L to mean that for every $A \in \mathbf{For}\mathcal{L}$: $\mathcal{C}$ derives A iff $A \in L$.*

Given a display calculus $\mathcal{C}$, we denote by $\mathcal{L}_\mathcal{C}$ the language determined by the connectives introduced by its logical rules. We do not exclude the possibility that a display calculus $\mathcal{C}$ for a logic in the language $\mathcal{L}$ derives B for some $B \notin \mathbf{For}\mathcal{L}$. This can occur only when the subset relation $\mathcal{L} \subset \mathcal{L}_\mathcal{C}$ is strict.

3 The Recipe

Suppose that $\mathcal{C}$ is a display calculus for a logic L in the language $\mathcal{L}$ satisfying C1–C8. We show how to define structural rules $r_1, \ldots, r_m$ so that $\mathcal{C} + \{r_1, \ldots, r_m\}$ is a cut-eliminable calculus for the axiomatic extension $L + \{A_1, \ldots, A_n\}$ ($A_i \in \mathbf{For}\mathcal{L}$). Our method is constructive and works whenever the base calculus $\mathcal{C}$ is 'expressive enough' (i.e., it is *amenable*), and the axioms A_i have a certain syntactic form.

Definition 4 (amenable calculus). *Let $\mathcal{C}$ be a display calculus satisfying C1–C8. Assume that we have two functions l and r mapping structures into $\mathbf{For}\mathcal{L}_\mathcal{C}$ such that $l(A) = r(A) = A$ when $A \in \mathbf{For}\mathcal{L}_\mathcal{C}$, and for an arbitrary structure X*

(i) $X \vdash l(X)$ *and* $r(X) \vdash X$ *are derivable in* $\mathcal{C}$.
(ii) $X \vdash Y$ *derivable implies* $l(X) \vdash r(Y)$ *is derivable* C.

Let there be a structure constant $\mathbf{I}$*, and let the following rules be admissible in $\mathcal{C}$ for arbitrary structures X, Y such that the premise and conclusion are well-defined in $\mathcal{C}$.*

$$\frac{\mathbf{I} \vdash X}{Y \vdash X}\, l\mathbf{I} \qquad \frac{X \vdash \mathbf{I}}{X \vdash Y}\, r\mathbf{I}$$

Let there be binary logical connectives $\vee, \wedge \in \mathcal{L}_\mathcal{C}$ such that $\cdot \in \{\vee, \wedge\}$ is associative *in $\mathcal{C}$ — $A \cdot (B \cdot C) \vdash (A \cdot B) \cdot C$ and $(A \cdot B) \cdot C \vdash A \cdot (B \cdot C)$ are derivable — and* commutative *in $\mathcal{C}$ — $A \cdot B \vdash B \cdot A$ is derivable. Also, for $A, B \in \mathbf{For}\mathcal{L}_\mathcal{C}$:*

$(a)_\vee$ $A \vdash X$ *and* $B \vdash X$ *implies* $\vee(A, B) \vdash X$
$(b)_\vee$ $X \vdash A$ *implies* $X \vdash \vee(A, B)$ *for any formula* B.
$(a)_\wedge$ $X \vdash A$ *and* $X \vdash B$ *implies* $X \vdash \wedge(A, B)$
$(b)_\wedge$ $A \vdash X$ *implies* $\wedge(A, B) \vdash X$ *for any formula* B.

A display calculus satisfying the above conditions is said to be amenable.

Requiring that $l\mathbf{I}$ and $r\mathbf{I}$ are admissible in $\mathcal{C}$ is weaker than requiring that $\mathcal{C}$ contains weakening rules. Indeed, the rules $l\mathbf{I}$ and $r\mathbf{I}$ are admissible in the bi-Lambek calculus [9]. The function l (resp. r) 'interprets' the structural connectives in the antecedent

(resp. succedent). Above, we use the notation $\wedge$ and $\vee$ to reflect that in a classical calculus, the connectives conjunction and disjunction satisfy the respective properties.

Our recipe abstracts and reformulates for display calculi the procedure in [5,6], defined for (hyper)sequent calculi and substructural logics. To transform axioms into structural rules we use: (1) the invertible logical rules of $\mathcal{C}$ and (2) the display calculus formulation, below, of the so-called Ackermann's lemma that allows a formula in a rule to switch sides of the sequent moving from conclusion to premises.

Lemma 1. *The following rules are pairwise equivalent in an amenable calculus where $A \in \mathbf{For}\mathcal{L}$, $\mathcal{S}$ is a set of sequents and $Z(\neq X)$ is a structure variable not in $\mathcal{S}$.*

$$\boxed{\dfrac{\mathcal{S}}{X \vdash A}\rho_1 \quad \dfrac{\mathcal{S} \quad A \vdash Z}{X \vdash Z}\rho_2} \qquad \boxed{\dfrac{\mathcal{S}}{A \vdash X}\delta_1 \quad \dfrac{\mathcal{S} \quad Z \vdash A}{Z \vdash X}\delta_2}$$

Proof. Suppose that we have concrete derivations of the premises $\mathcal{S} \cup \{A \vdash Z\}$ of ρ_2. Applying ρ_1 to $\mathcal{S}$ we get $X \vdash A$. Applying cut with $A \vdash Z$ we get $X \vdash Z$ and thus it follows that ρ_2 is admissible in a calculus containing ρ_1.

Now suppose we have concrete derivations of the premises $\mathcal{S}$ of ρ_1. Observe that $r(A) \vdash A$ is derivable. Applying ρ_2 to $\mathcal{S} \cup \{r(A) \vdash A\}$ we get $X \vdash A$ as required. The proof that δ_1 and δ_2 are equivalent is analogous.

We now give an abstract description of the axioms that we can handle. The description is based on the invertible rules of the chosen display calculus $\mathcal{C}$ and is inspired by the classification in [5] of formulae of FLe. We identify three classes of formulae in the language of $\mathcal{L}$ from which the logical connectives can be removed using the invertible rules of $\mathcal{C}$ at various levels. The class $\mathcal{I}_0$ consists of formulae with no logical connective (so there is no need for the invertible rules). Logical connectives in formulae in $\mathcal{I}_1$ can be eliminated by repeatedly applying the invertible rules starting with sequents (thus obtaining sets of sequents). Logical connectives in formulae in $\mathcal{I}_2$ can be eliminated by repeatedly applying the invertible rules to formulae, sequents and to the premises of rules obtained via Lemma 1 (thus obtaining sets of structural rules).

Definition 5 (inv). *The function inv takes a sequent $X \vdash Y$ and applies all the invertible logical rules in $\mathcal{C}$ that are possible and returns the (necessarily finite) set $\{X_i \vdash Y_i\}_{i\in\Omega}$ of sequents for some index set Ω.*

Definition 6 (soluble). *A formula $A \in \mathbf{For}\mathcal{L}$ is* a-soluble *(resp.* s-soluble*) if the sequents in inv($A \vdash$) (resp. inv($\vdash A$)) do not contain any logical connectives.*

Definition 7. *Let $\mathcal{C}$ be an amenable calculus for L. The classes $\mathcal{I}_0, \mathcal{I}_1, \mathcal{I}_2$ of formulae of $\mathbf{For}\mathcal{L}$ are defined in the following way: $A \in \mathbf{For}\mathcal{L}$ with inv($\vdash A$) $= \{U_i \vdash V_i\}_{i\in\Omega}$ for some finite Ω belongs to*

- $\mathcal{I}_0$ *if A contains no logical connectives*
- $\mathcal{I}_1$ *if each a-part formula in $U_i \vdash V_i$ is a-soluble and each s-part formula in $U_i \vdash V_i$ is s-soluble*
- $\mathcal{I}_2$ *if each a-part formula in $U_i \vdash V_i$ is s-soluble and each s-part formula in $U_i \vdash V_i$ is a-soluble*

A propositional variable is both a-soluble and s-soluble so $\mathcal{I}_0 \subseteq \mathcal{I}_1$ and $\mathcal{I}_0 \subseteq \mathcal{I}_2$. Note that every a-part (resp. s-part) formula B occurring in a sequent in inv($\vdash A$) that is a-soluble (s-soluble) must be a propositional variable and thus B is s-soluble (a-soluble) in that sequent. It follows that $\mathcal{I}_1 \subseteq \mathcal{I}_2$.

Remark 1. The above classes are a function of the invertible rules of the base calculus. In particular, these coincide with the classes in the hierarchy of [5] that can be handled by structural sequent rules, when the base calculus has the same invertible rules. More invertible rules lead to larger classes of formulae in $\mathcal{I}_1$ and $\mathcal{I}_2$ (see Section 4.2).

Henceforth a rule whose conclusion is constructed from structure variables and structure constants using structural connectives, and whose premises might additionally contain propostional variables will be called a *semi-structural* rule.

Proposition 1. *Let $\mathcal{C}$ be an amenable calculus for L. Suppose $A \in \mathbf{For}\mathcal{L}$ with inv($\vdash A$) $= \{U_i \vdash V_i\}_{i\in\Omega}$. If $A \in \mathcal{I}_2$ then there are equivalent semi-structural rules $\{\rho_i\}_{i\in\Omega}$ so that $\mathcal{C} + \{\rho_i\}_{i\in\Omega}$ is a cut-eliminable calculus for $L + A$.*

Proof. Clearly $\vdash A$ is equivalent to $\{U_i \vdash V_i\}_{i\in\Omega}$ in $\mathcal{C}$. We show how to construct a semi-structural rule equivalent to each $U_i \vdash V_i$. Suppose that $U_i \vdash V_i$ consists of a-part formulae $C_1, \ldots, C_n$ and s-part formulae $D_1, \ldots, D_m$. Starting with $U_i \vdash V_i$, display each C_i (as $C_i \vdash W_i$ for suitable W_i) and apply Lemma 1 in turn, to obtain an equivalent rule of the form below left. Start with this rule and display in the conclusion each D_i (as $W_{n+i} \vdash D_i$ for suitable W_{n+i}) and apply Lemma 1 in turn, to obtain an equivalent rule of the form below right:

$$\frac{Z_1 \vdash C_1 \quad \ldots \quad Z_n \vdash C_n}{Z_n \vdash W_n} \qquad \frac{Z_1 \vdash C_1 \ \ldots \ Z_n \vdash C_n \quad D_1 \vdash Z_{n+1} \ \ldots \ D_m \vdash Z_{n+m}}{W_{n+m} \vdash Z_{n+m}}$$

Observe that $W_{n+m} \vdash Z_{n+m}$ is constructed only from structure variables and structure constants using structural connectives. Since $A \in \mathcal{I}_2$, every C_i (resp. D_i) formula is s-soluble (a-soluble) and so the following is a semi-structural rule equivalent to $U_i \vdash V_i$:

$$\frac{\text{inv}(Z_1 \vdash C_1) \ \ldots \ \text{inv}(Z_n \vdash C_n) \quad \text{inv}(D_1 \vdash Z_{n+1}) \ \ldots \ \text{inv}(D_m \vdash Z_{n+m})}{W_{n+m} \vdash Z_{n+m}}\ \rho_i$$

By inspection it may be verified that ρ_i satisfies Belnap's conditions with the possible exception of C1 and C4 since the same propositional variable might appear in the premises as an a-part and s-part formula. However, since no propositional variable occurs in the conclusion of ρ_i, cut-elimination for $\mathcal{C} + \{\rho_i\}_{i\in\Omega}$ proceeds without difficulty.

Notice that the calculus $\mathcal{C} + \{\rho_i\}_{i\in\Omega}$ in the above result has cut-elimination but not in general the subformula property. If we restrict our attention to a subclass of $\mathcal{I}_2$ satisfying the additional condition of *acyclicity* then the propositional variables appearing in each ρ_i can be suitably removed. In this way we obtain structural rules satisfying C1–C8 so the resulting calculus is cut-eliminable and has the subformula property.

Definition 8 (proper structural rules; extensions). *A proper structural rule (extension) is a structural rule (extension) that satisfies C1–C8.*

Our transformation of semi-structural rules into proper structural rules mirrors the 'completion' procedure in [6] and amounts to applying the cut-rule in all possible ways to the premises of the former rules. Below we present formally the transformation.

A (possibly empty) set $\mathcal{S}$ of sequents is said to *respect multiplicities wrt* p for some propositional variable p if it can be written in one of the forms below:

$$\{p \vdash U \mid p \notin U\} \cup \{V \vdash p \mid \text{every } p \text{ in } V \vdash p \text{ is s-part}\} \cup \{S \mid p \notin S\} \tag{1}$$

$$\{U \vdash p \mid p \notin U\} \cup \{p \vdash V \mid \text{every } p \text{ in } p \vdash V \text{ is a-part}\} \cup \{S \mid p \notin S\} \tag{2}$$

An alternative definition is that (i) no $S \in \mathcal{S}$ contains both an a-part and s-part occurrence of p — eg. $p \vdash p$ cannot be in $\mathcal{S}$, and (ii) there do not exist $S_1, S_2 \in \mathcal{S}$ such that S_1 contains multiple a-part occurrences of p and S_2 contains multiple s-part occurrences of p. Eg. if both occurrences of p in $p \otimes p \vdash X$ (resp. $Y \vdash p \otimes p$) are a-part (s-part) for a structural connective $\otimes$, then it cannot be that $p \otimes p \vdash X \in \mathcal{S}$ and $Y \vdash p \otimes p \in \mathcal{S}$.

Let $\mathcal{S}$ be a set of sequents respecting multiplicities wrt p. If it is not the case that $p \vdash U \in \mathcal{S}$ *and* $V \vdash p \in \mathcal{S}$ (upto display equivalence) then define $\mathcal{S}_p$ as $\{S \in \mathcal{S} \mid p \notin S\}$. Otherwise, depending on the form of $\mathcal{S}$ as (1) or (2), define respectively $\mathcal{S}_p$ as follows:

$$\{S \mid S \text{ is a subst. instance of } V \vdash p \in \mathcal{S} \text{ s.t. each occ. } p \mapsto U \text{ for some } p \vdash U \in \mathcal{S}\} \cup \{S \mid p \notin S\}$$

$$\{S \mid S \text{ is a subst. instance of } p \vdash V \in \mathcal{S} \text{ s.t. each occ. } p \mapsto U \text{ for some } U \vdash p \in \mathcal{S}\} \cup \{S \mid p \notin S\}$$

In the above, notice that distinct occurrences of p in $V \vdash p$ (resp. $p \vdash V$) may be substituted for distinct U_i so long as $p \vdash U_i$ ($U_i \vdash p$) is in $\mathcal{S}$. Also observe that the substitution instance S contains no occurrence of p since $p \notin U$.

Intuitively, if $\mathcal{S}$ contains sequents of the form $p \vdash U$ *and* $V \vdash p$ then $\mathcal{S}_p$ is obtained by (i) applying the cut-rule in all possible ways on p (using the sequents in $\mathcal{S}$ as the premises) and then keeping only those conclusion sequents of the cut-rule that do not contain p, and (ii) retaining $\{S \in \mathcal{S} \mid p \notin S\}$.

Lemma 2. *If $\mathcal{S}$ respects multiplicities wrt p, then p does not occur in $\mathcal{S}_p$.*

Proof. Follows immediately from the form of $\mathcal{S}$ and the definition of $\mathcal{S}_p$.

Let $\mathbb{V}(\mathcal{S})$ be the set of propositional variables occurring in a set $\mathcal{S}$ of sequents.

Definition 9. *A finite set $\mathcal{S}$ of sequents is* acyclic *if $\mathbb{V}(\mathcal{S}) = \emptyset$ or for every $p \in \mathbb{V}(\mathcal{S})$: (i) $\mathcal{S}$ respects multiplicities wrt p, and (ii) $\mathcal{S}_p$ is acyclic.*

Definition 10. *Suppose that $A \in \mathcal{I}_2$ and let $\{\rho_i\}_{i \in \Omega}$ be the equivalent semi-structural rules obtained using Prop. 1. We say that A is* acyclic *if the set of premises of each rule in $\{\rho_i\}_{i \in \Omega}$ is acyclic.*

Remark 2. Every axiom in $\mathcal{I}_1$ is acyclic. This follows from the observation that for every $A \in \mathcal{I}_1$, the premise of each semi-structural rule obtained using Prop. 1 has the form $p \vdash L$ or $L \vdash p$ where L is a schematic structure variable.

Lemma 3. *Let $\mathcal{S}$ be an acyclic set of sequents and $p \in \mathbb{V}(\mathcal{S})$. Then the semi-structural rule ρ with premises $\mathcal{S}$ and the semi-structural rule ρ_p with premises $\mathcal{S}_p$ are equivalent w.r.t. an amenable calculus $\mathcal{C}$.*

Proof. Let $\mathcal{S}$ be an acyclic set of sequents. Suppose that $\mathcal{S}$ does not contain sequents of the form $p \vdash U$ *and* $V \vdash p$. Then $\mathcal{S}$ has one of the following forms

$$\{V_1 \vdash p, \ldots, V_{n+1} \vdash p\} \cup \{S \,|\, p \notin S\} \quad \{p \vdash V_1, \ldots, p \vdash V_{n+1}\} \cup \{S \,|\, p \notin S\}$$

and $\mathcal{S}_p$ is $\{S \in \mathcal{S} \,|\, p \notin S\}$. Suppose the case above left (the other case is similar). One direction is immediate, and to show that ρ_p is admissible in $\mathcal{C} + \rho$ it is enough to apply ρ using the derivable sequents $\{r(\mathbf{I}) \vdash V_i[p \mapsto r(\mathbf{I})]\}_{1 \leq i \leq n+1}$ — obtained from the derivation of $r(\mathbf{I}) \vdash \mathbf{I}$ using the rule $r\mathbf{I}$ — for the missing premises.

Now suppose that $\mathcal{S}$ contains sequents of the form $p \vdash U$ *and* $V \vdash p$. Clearly ρ is admissible in $\mathcal{C} + \rho_p$ — it suffices to apply the cut-rule to concrete premises of ρ and then apply ρ_p. For the other direction, assume, to fix ideas that the premises $\mathcal{S}$ of ρ have the form (1) (the other case is similar, use $(a)_\vee$ and $(b)_\vee$ instead of $(a)_\wedge$ and $(b)_\wedge$), i.e.,

$$\{p \vdash U_i \,|\, p \notin U_i; 1 \leq i \leq n\} \cup \{V \vdash p \,|\, \text{every } p \text{ in } V \vdash p \text{ is s-part}\} \cup \{S \,|\, p \notin S\}$$

Then the premises $\mathcal{S}_p$ of ρ_p have the following form:

$$\{S \,|\, S \text{ is a subst. instance of } V \vdash p \in \mathcal{S} \text{ s.t. each occ. } p \mapsto U_i \text{ for some } 1 \leq i \leq n) \,\} \cup \{S \,|\, p \notin S\}$$

Suppose that we are given concrete instances of the premises of ρ_p. Repeatedly using $(a)_\wedge$ and the display rules, obtain the set $\mathcal{S}_p^*$:

$$\{S \,|\, S \text{ is a subst. instance of } V \vdash p \in \mathcal{S} \text{ s.t. each occ. } p \mapsto \wedge_{1 \leq i \leq n} \mathrm{r}(U_i) \,\} \cup \{S \,|\, p \notin S\}$$

Making use of $(b)_\wedge$, derive the set $\{\wedge_{1 \leq i \leq n} \mathrm{r}(V_i) \vdash V_i\}$ of sequents. By inspection, this set together with $\mathcal{S}_p^*$ yield concrete instances of the premises of ρ (in particular, p has been instantiated with $\wedge_{1 \leq i \leq n} \mathrm{r}(V_i)$). Applying ρ to these and noting that ρ and ρ_p have the same conclusion, we have that ρ_p is admissible in $\mathcal{C} + \rho$.

Theorem 2. *Let $\mathcal{C}$ be an amenable calculus for L and suppose that $A \in \mathcal{I}_2$ is acyclic. Then there is a proper structural rule extension for $\mathrm{L} + A$.*

Proof. Let $\{\rho_i\}_{i \in \Omega}$ be the semi-structural rules computed from A in Prop. 1. Notice that each ρ_i might violate (only) Belnap's conditions C1 and C4 due to the presence of propositional variables in the set $\mathcal{S}$ of sequents that are its premises. Let $\mathbb{V}(\mathcal{S}) = \{p_1, p_2, \ldots, p_n\}$ be such variables and ρ_i' be the rule with premises $((\ldots(\mathcal{S}_{p_1})_{p_2}\ldots)_{p_{n-1}})_{p_n}$. By inspection of the construction of ρ_i' from ρ follows that ρ_i' is a proper structural rule (in particular, observe that any structure variable that appears only as an a-part (resp. s-part) structure in every sequent in $\mathcal{S}$ has the same property in $((\ldots(\mathcal{S}_{p_1})_{p_2}\ldots)_{p_{n-1}})_{p_n})$. Since A is acyclic, so is $\mathcal{S}$ and hence, by (repeteadly applying) Lemma 2 it follows that ρ_i' is equivalent to ρ_i.

By repeating this process to all $\{\rho_i\}_{i \in \Omega}$ we obtain a new set of rules $\{\rho_i'\}_{i \in \Omega}$ such that $\mathcal{C} + \{\rho_i'\}_{i \in \Omega}$ is a proper structural rule extension of $\mathrm{L} + A$.

4 Case Studies

We apply the recipe in Section 3 to obtain many existing results *uniformly*, and to show that new calculi can be defined in an automated way. When dealing with a concrete

base calculus we can provide an explicit description (grammar-like) of the class $\mathcal{I}_2$ of axioms that can be transformed into equivalent structural rules to obtain cut-eliminable display calculi. We present this grammar for the case of intermediate logics to compare our results with those in [5].

4.1 Bi-intuitionistic Logic

Bi-intuitionistic logic (also known as Heyting-Brouwer logic) is the logic which results when the dual $\rightarrow_d$ of implication (alias coimplication) is added to the language of intuitionistic logic. Here we show how to construct cut-free display calculi for infinitely many axiomatic extensions of this logic in a uniform way.

The language $\mathcal{L}_{\text{HB}}$ of Heyting-Brouwer logic HB is obtained from the language $\mathcal{L}_{\text{Ip}}$ of intuitionistic propositional logic Ip by the addition of $\rightarrow_d$. To simplify the language, we abbreviate $\neg p := p \rightarrow \bot$ and $\neg_d p := \top \rightarrow_d p$. Wansing [18] give a proper display calculus δHB for HB. Our presentation here differs in that we use the invertible forms for $\wedge r$ and $\vee l$. Equivalence with the original rules can be shown using the structural rules of contraction and weakening in δHB.

The set of structures $\mathfrak{Str}(\mathcal{L}_{\text{HB}})$ generated from $\mathcal{L}_{\text{HB}}$ has the following grammar:

$$X ::= A \in \mathbf{For}\mathcal{L}_{\text{HB}} \mid \mathbf{I} \mid (X \circ X) \mid (X \bullet X)$$

The initial sequents of δHB are of the form $p \vdash p$ for any propositional variable p, and $\mathbf{I} \vdash \top$ and $\bot \vdash \mathbf{I}$. Now we present the structural rules. In the first row, below, we use a double line to separate the premises from the conclusion to indicate that a rule is invertible and also 'combine' two rules into a single one for the sake of brevity. The first two columns (counting from the left) in the first row are the *display rules* of δHB.

$$\dfrac{\dfrac{Y \vdash X \circ Z}{X \circ Y \vdash Z}}{X \vdash Y \circ Z} \qquad \dfrac{\dfrac{X \bullet Y \vdash Z}{X \vdash Y \bullet Z}}{X \bullet Z \vdash Y} \qquad \dfrac{\dfrac{\mathbf{I} \circ X \vdash Y}{X \vdash Y}}{X \circ \mathbf{I} \vdash Y} \qquad \dfrac{\dfrac{X \vdash Y \bullet \mathbf{I}}{X \vdash Y}}{X \vdash \mathbf{I} \bullet Y}$$

$$\dfrac{X \vdash Y}{X \vdash Y \bullet Z} \qquad \dfrac{X \vdash Y}{X \circ Z \vdash Y} \qquad \dfrac{X \vdash Y \bullet Z}{X \vdash Z \bullet Y} \qquad \dfrac{X \circ Z \vdash Y}{Z \circ X \vdash Y}$$

$$\dfrac{X \vdash Y \bullet Y}{X \vdash Y} \qquad \dfrac{X \circ X \vdash Y}{X \vdash Y} \qquad \dfrac{X \vdash (Y \bullet Z) \bullet U}{X \vdash Y \bullet (Z \bullet U)} \qquad \dfrac{(X \circ Y) \circ Z \vdash U}{X \circ (Y \circ Z) \vdash U}$$

The logical rules of δHB are given below:

$$\dfrac{\mathbf{I} \vdash X}{\top \vdash X}\,\top l \qquad \dfrac{X \vdash \mathbf{I}}{X \vdash \bot}\,\bot r \qquad \dfrac{A \circ B \vdash X}{A \wedge B \vdash X}\,\wedge l$$

$$\dfrac{X \vdash A \qquad X \vdash B}{X \vdash A \wedge B}\,\wedge r \qquad \dfrac{A \vdash X \qquad B \vdash X}{A \vee B \vdash X}\,\vee l \qquad \dfrac{X \vdash A \bullet B}{X \vdash A \vee B}\,\vee r$$

$$\dfrac{X \vdash A \qquad Y \vdash B}{A \rightarrow B \vdash X \circ Y}\,\rightarrow l \qquad \dfrac{X \vdash A \circ B}{X \vdash A \rightarrow B}\,\rightarrow r \qquad \dfrac{B \bullet A \vdash X}{B \rightarrow_d A \vdash X}\,\rightarrow_d l$$

$$\dfrac{X \vdash B \qquad Y \vdash A}{X \bullet Y \vdash B \rightarrow_d A}\,\rightarrow_d r$$

Define the functions l and r from $\mathfrak{Stt}(\mathcal{L}_{\mathrm{HB}})$ into $\mathbf{For}\mathcal{L}_{\mathrm{HB}}$:

$$\begin{aligned} l(A) &= A & r(A) &= A \\ l(\mathbf{I}) &= \top & r(\mathbf{I}) &= \bot \\ l(X \circ Y) &= l(X) \wedge l(Y) & r(X \circ Y) &= l(X) \to r(Y) \\ l(X \bullet Y) &= l(X) \to_d l(Y) & r(X \bullet Y) &= r(X) \vee r(Y) \end{aligned}$$

It is easy to check that δHB is amenable. (Note that $\wedge$ and $\vee$ are associative and commutative connectives in δHB). We have the following result.

Proposition 3. *Every logical rule except $\to l$ and $\to_d r$ is invertible.*

Theorem 4. *Let A be any (acyclic) axiom within $\mathcal{I}_2$. Then there is a (proper) structural rule extension of δHB for HB $+ A$.*

The following examples contain analytic display calculi for two axiomatic extensions of HB introduced in [19].

Example 1. Let A_1 be the axiom $(p \to q) \vee (q \to p)$. Then inv($\vdash A_1$) is the sequent $\vdash (p \circ q) \bullet (q \circ p)$. Since each formula in that sequent is a propositional variable, it is a,s-soluble. Thus $A_1 \in \mathcal{I}_1$. From Prop. 1 we obtain the equivalent semi-structural rule (below left). The set $\mathcal{S}$ of premises of this rule can be written $\{X \vdash p\} \cup \{p \vdash V\} \cup \{Z \vdash q, q \vdash Y\}$. Then $\mathcal{S}_p = \{X \vdash V, Z \vdash q, q \vdash Y\}$. Hence $\mathcal{S}_{pq} = \{X \vdash V, Z \vdash Y\}$. So $\mathcal{S}$ is acyclic and is equivalent to the proper structural rule below right:

$$\frac{X \vdash p \quad q \vdash Y \quad Z \vdash q \quad p \vdash V}{\mathbf{I} \vdash (X \circ Y) \bullet (Z \circ V)}\,\rho_1 \qquad \frac{X \vdash V \qquad Z \vdash Y}{\mathbf{I} \vdash (X \circ Y) \bullet (Z \circ V)}\,\rho_1'$$

Thus δHB $+ \rho_1'$ is a cut-eliminable display calculus for HB $+ A_1$ with subformula property. In practice, ρ_1' can be obtained from ρ_1 on sight, by applying the cut-rule to the premises in 'all possible ways'.

Example 2. Let A_2 be $\neg((p \to_d q) \wedge (q \to_d p))$. $A_2 \in \mathcal{I}_1$. Applying our recipe we get the equivalent rule ρ_2 such that δHB $+ \rho_2$ is a proper display calculus for HB $+ A_2$.

$$\frac{X \vdash Z \qquad U \vdash Y}{(X \bullet Y) \circ (U \bullet Z) \vdash \mathbf{I}}\,\rho_2$$

4.2 Intuitionistic Logic

We discuss intermediate logics and compare our algorithm for display logic with the algorithm in [5] that works for hypersequent calculus – a simple generalization of Gentzen calculus [1] whose basic objects are multisets of sequents.

The calculus $\delta\mathrm{HB}^-$ obtained by deleting the logical rules for $\to_d$ is a display calculus for Ip — soundness of HB^- relies on the fact that Ip is a conservative extension of HB, and completeness follows from cut-elimination for HB. Observe that in $\delta\mathrm{HB}^-$ — unlike in Gentzen's calculus LJ — the $\vee r$ rule is also invertible. Following the idea of

the classification in [5], which is sketched below for the connectives of Ip (= FLe with weakening and contraction), we can define $\mathcal{I}_2$ axioms for Ip and $\delta\mathrm{HB}^-$ as follows

$$\mathcal{I}_0 ::= \text{prop. variables} \quad \mathcal{I}_{n+1} ::= \bot \mid \top \mid \mathcal{I}_n \to \mathcal{I}_{n+1} \mid \mathcal{I}_{n+1} \wedge \mathcal{I}_{n+1} \mid \mathcal{I}_{n+1} \vee \mathcal{I}_{n+1}$$

The class $\mathcal{I}_2$ is larger than the class of axioms that can be captured by structural hypersequent rules over LJ (see [5]). The latter consists of all axioms within the class $\mathcal{P}_3$ defined by the following grammar: $\mathcal{N}_0, \mathcal{P}_0$ contains the set of atomic formulae, and

$$\mathcal{P}_{n+1} ::= \bot \mid \top \mid \mathcal{N}_n \mid \mathcal{P}_{n+1} \wedge \mathcal{P}_{n+1} \mid \mathcal{P}_{n+1} \vee \mathcal{P}_{n+1}$$
$$\mathcal{N}_{n+1} ::= \bot \mid \top \mid P_n \mid \mathcal{P}_{n+1} \to \mathcal{N}_{n+1} \mid \mathcal{N}_{n+1} \wedge \mathcal{N}_{n+1}$$

(the classes $\mathcal{P}_n$ and $\mathcal{N}_n$ stand for axioms with leading positive and negative connective, i.e. having left (resp. right) logical rule invertible). It is easy to see that $\mathcal{P}_3 \subseteq I_2$.

By applying our recipe and making use of the weakening, commutativity and contraction rules in $\delta\mathrm{HB}^-$ we can show the following:

Proposition 5. *There is a proper structural rule extension of δHB^- for Ip + $\mathcal{A}$, for any set $\mathcal{A}$ of axioms in $\mathcal{P}_3$.*

Proof. By [5, Lemma 3.4] any $\mathcal{P}_3$ formula can be written as a conjunction of formulae $(*)$ $\vee_{1\leq i\leq N} (\alpha^i_1 \wedge \ldots \wedge \alpha^i_{n_i} \to \beta^i)$ where each β^i has the form $\overline{q}^i_1 \vee \ldots \vee \overline{q}^i_{m_i}$ ($\overline{q}^i_j$ is a conjunction of propositional variables or $\bot$). Hence inv($\mathbf{I} \vdash A$) consists of sequents of the following form, where q^i_j is some propositional variable occurring in $\overline{q}^i_j$.

$$\mathbf{I} \vdash \big((\alpha^1_1 \circ \ldots \circ \alpha^1_{n_1}) \circ (q^1_1 \bullet \ldots \bullet q^1_{m_1})\big) \bullet \ldots \bullet \big((\alpha^N_1 \circ \ldots \circ \alpha^N_{n_N}) \circ (q^N_1 \bullet \ldots \bullet q^N_{m_N})\big)$$

Note that each α^i_j is an a-part formula and each q^i_j is an s-part formula. Now, following Prop. 1 we apply Lemma 1 and obtain that A is equivalent to the following semi-structural rule, where L^i_j is a structure variable (corresponding to α^i_j) and Q^i_j is a structure variable (corresponding to q^i_j).

$$\frac{\{\text{inv}(L^i_j \vdash \alpha^i_j)\}_{1\leq i\leq N; 1\leq j\leq n_i} \qquad \{q^i_j \vdash Q^i_j\}_{1\leq i\leq N; 1\leq j\leq m_i}}{\mathbf{I} \vdash \big((L^1_1 \circ \ldots \circ L^1_{n_1}) \circ (Q^1_1 \bullet \ldots \bullet Q^1_{m_1})\big) \bullet \ldots \bullet \Big((L^N_1 \circ \ldots \circ L^N_{n_N}) \circ (Q^N_1 \bullet \ldots \bullet Q^N_{m_N})\Big)}\ \rho'$$

where the structure variables $L^1_1, \ldots, L^N_{n_N}, Q^1_1, \ldots, Q^N_{m_N}$ are distinct. By [5, Lemma 3.4], each α^i_j in $(*)$ has the form $\wedge_{1\leq k\leq a_j}(U^i_{jk} \to v^i_{jk})$ where U^i_{jk} is $\top$ or a conjunction of propositional variables. Hence each set inv($L^i_j \vdash \alpha^i_j$) consists of sequents of the form

$$L^i_j \vdash (u^i_{j1} \circ \ldots \circ u^i_{ja^i_j}) \circ v^i_j$$

Observe that each propositional variable u^i_{jk} is an a-part formula and each propositional variable v^i_j is an s-part formula. Since the calculus has contraction on $\circ$ in the antecedent, we may assume without loss of generality that the u^i_{jk} are distinct for fixed i. If some $u^i_{jk} = v^i_j$ then the sequent is derivable as follows by repeated use of the weakening, commutativity and display rules for $\circ$:

$$\dfrac{\dfrac{\dfrac{v_j^i \vdash v_j^i}{(u_{j1}^i \circ \ldots \circ u_{ja_{ij}}^i) \vdash v_j^i}}{L_j^i \circ (u_{j1}^i \circ \ldots \circ u_{ja_{ij}}^i) \vdash v_j^i}}{L_j^i \vdash (u_{j1}^i \circ \ldots \circ u_{ja_{ij}}^i) \circ v_j^i}$$

Thus we can delete those premises of ρ' such that $u_{jk}^i = v_j^i$ to obtain an equivalent rule ρ. The premises $\mathcal{S}$ of ρ have the following form:

$$\{U \vdash p \,|\, p \notin U\} \cup \{p \vdash V \,|\, p \notin V\} \cup \{S \,|\, p \notin S\}$$

Let $\mathcal{S}'_p$ be the set $\{U \vdash V \,|\, p \notin U, p \notin V\} \cup \{S \,|\, p \notin S\}$. Arguing as in Lemma 3 we can show that ρ is equivalent to the rule ρ'_p obtained by replacing the premises $\mathcal{S}$ with $\mathcal{S}'_p$. While $\mathcal{S}'_p$ does not contain p, it may contain a sequent with (i) multiple a-part occurrences of some propositional variable or, (ii) an a-part and s-part occurrence of the same propositional variable. Obtain the rule ρ_p from ρ'_p by contracting multiplicities and deleting sequents witnessing (ii). Denote the premises of ρ_p by $\mathcal{S}_p$. Repeat for all propositional variables in $\mathcal{S}$ to obtain ultimately an equivalent proper structural rule.

Hence we can get proper structural rule extensions of δHB$^-$ for all intermediate logics that can be formalized by hypersequent calculi using the algorithm in [5]. But we can do more. Consider the axioms (Bd_k) $(k \geq 1)$, defining intermediate logics semantically characterized by Kripke models of depth $\leq k$, belong to the classes $\mathcal{P}_{2k}$ in the classification in [5]; these axioms are recursively defined as follows:

$$(Bd_1) \quad p_1 \vee \neg p_1 \qquad (Bd_{i+1}) \quad p_{i+1} \vee (p_{i+1} \to (Bd_i))$$

For $k \geq 2$, no axiom within $\mathcal{P}_3$ is known to be equivalent, yet these all belong to $\mathcal{I}_1$.

Example 3. The proper structural rule equivalent to the axiom (Bd_2) is

$$\dfrac{Y \vdash X \qquad V \vdash U}{\mathbf{I} \vdash X \bullet (Y \circ (U \bullet (V \circ \mathbf{I})))}\ \rho$$

In contrast *no* equivalent hypersequent structural rule is known.

Although our algorithm is inspired by that in [5], the key point is that the expressive power of the display calculus permits a base calculus for Ip in which the $\vee r$ rule is also invertible, leading to cut-eliminable structural rule extensions for more logics (see Remark 1). This justifies the use of the more complex machinery of the display calculus.

Example 4. δHB$^- + \rho'_1$ (cf. Example 1) is a cut-free calculus for Ip$+A_1$ (= Gödel logic) with subformula property. Classical propositional logic Cp is obtained as Ip $+ p \vee \neg p$. Since $p \vee \neg p \in I_1$ we can define a proper structural rule extension of δHB$^-$ for Cp.

4.3 Bunched Logics

Bunched logics [13] provide a powerful framework to reason about resources. They are obtained by combining an additive propositional logic with a multiplicative linear

logic [4]. The combination led to the definition of four systems: BI, BBI (Boolean BI), dMBI (de Morgan BI) and CBI (classical BI). Brotherston [4] obtains display calculi for these logics by freely combining a calculus DL_{IL} (resp. DL_{CL}) for intutionistic (classical) propositional logic with a calculus DL_{LM} (resp. DL_{dMM}) for multiplicative intuitionistic linear logic (multiplicative classical linear logic).

Using the calculus $\delta\mathrm{HB}^-$ for intuitionistic logic instead of DL_{IL}, new calculi for BI and BBI can be obtained. These calculi can be extended with the structural rule for classical logic (see Example 4) to obtain new calculi for dMBI and CBI that are structural extensions of the calculi for BI and BBI.

More generally, our algorithm yields proper structural rule extensions over $\delta\mathrm{HB}^-$ for a large class of intermediate logics. Taking the free combination of such calculi with $\{DL_{LM}, DL_{dMM}\}$ yield cut-eliminable calculi for new bunched logics (intermediate between BI and CBI) which may express interesting properties on resources.

4.4 Modal and Tense Logics

The modal language $\mathcal{L}_K$ is obtained from the propositional classical language by the addition of the modal operators $\Diamond$ and $\Box$. The tense language $\mathcal{L}_{Kt}$ is obtained from $\mathcal{L}_K$ by the addition of the tense operators $\blacklozenge$ and $\blacksquare$. The normal basic modal logic K and tense logic Kt are conservative extensions of classical propositional logic Cp, obtained by the addition of the usual axioms (see [3]).

The display calculus δKt [14] for Kt is well-known. Here we use the invertible form of the rules for $\wedge r$, $\vee l$ and $\rightarrow l$. The set of structures $\mathfrak{Str}(\mathcal{L}_{Kt})$ generated from $\mathcal{L}_{Kt}$ has the following grammar:

$$X ::= A \in \mathbf{For}\mathcal{L}_{Kt} \mid \mathbf{I} \mid (X \circ X) \mid \bullet X \mid *X$$

The initial sequents of δKt are of the form $p \vdash p$ for any propositional variable p, and $\mathbf{I} \vdash \top$ and $\bot \vdash \mathbf{I}$. In the following we use a double line to separate the premises from the conclusion to indicate that a rule is invertible. The *display rules* of δKt are:

$$\frac{X \circ Y \vdash Z}{X \vdash Z \circ *Y} \qquad \frac{X \circ Y \vdash Z}{Y \vdash *X \circ Z} \qquad \frac{X \vdash Y \circ Z}{X \circ *Z \vdash Y}$$

$$\frac{X \vdash Y \circ Z}{*Y \circ X \vdash Z} \qquad \frac{*X \vdash Y}{*Y \vdash X} \qquad \frac{X \vdash *Y}{Y \vdash *X}$$

$$\frac{** X \vdash Y}{X \vdash Y} \qquad \frac{X \vdash ** Y}{X \vdash Y} \qquad \frac{X \vdash \bullet Y}{\bullet X \vdash Y}$$

The remaining structural rules of δKt are given below.

$$\frac{X \vdash Z}{\mathbf{I} \circ X \vdash Z} \qquad \frac{X \vdash Z}{X \vdash \mathbf{I} \circ Z} \qquad \frac{\mathbf{I} \vdash Y}{*\mathbf{I} \vdash Y} \qquad \frac{X \vdash \mathbf{I}}{X \vdash *\mathbf{I}}$$

$$\frac{X \vdash Z}{Y \circ X \vdash Z} \qquad \frac{X \vdash Z}{X \circ Y \vdash Z} \qquad \frac{\mathbf{I} \vdash Y}{\bullet\mathbf{I} \vdash Y} \qquad \frac{X \vdash \mathbf{I}}{X \vdash \bullet\mathbf{I}}$$

$$\frac{X \circ Y \vdash Z}{Y \circ X \vdash Z} \qquad \frac{Z \vdash X \circ Y}{Z \vdash Y \circ X} \qquad \frac{X \circ X \vdash Z}{X \vdash Z} \qquad \frac{Z \vdash X \circ X}{Z \vdash X}$$

$$\frac{X_1 \circ (X_2 \circ X_3) \vdash Z}{(X_1 \circ X_2) \circ X_3 \vdash Z} \qquad \frac{Z \vdash X_1 \circ (X_2 \circ X_3)}{Z \vdash (X_1 \circ X_2) \circ X_3}$$

Name	Axiom	Rule	Name	Axiom	Rule
D	$\Box A \to \Diamond A$	$(* \bullet *)\bullet X \vdash Y / X \vdash Y$	B	$A \to \Box\Diamond A$	$* \bullet * X \vdash Y / \bullet X \vdash Y$
	$\Diamond\Box A \to \Box\Diamond A$	$\bullet X \vdash * \bullet * Y / * \bullet * X \vdash \bullet Y$	4	$\Box A \to \Box\Box A$	$\bullet X \vdash Y / \bullet \bullet X \vdash Y$
5	$\Diamond A \to \Box\Diamond A$	$* \bullet * X \vdash Y / * \bullet * X \vdash \bullet Y$	T	$\Box A \to A$	$\bullet X \vdash Y / X \vdash Y$

Fig. 1. Some $\mathcal{I}_2$ axioms and corresponding proper structural rules

The logical rules of δKt are given below.

$$\frac{\mathbf{I} \vdash X}{\top \vdash X}\,\top l \qquad \frac{X \vdash \mathbf{I}}{X \vdash \bot}\,\bot r \qquad \frac{* A \vdash X}{\neg A \vdash X}\,\neg l$$

$$\frac{X \vdash * A}{X \vdash \neg A}\,\neg r \qquad \frac{A \circ B \vdash X}{A \wedge B \vdash X}\,\wedge l \qquad \frac{X \vdash A \qquad X \vdash B}{X \vdash A \wedge B}\,\wedge r$$

$$\frac{A \vdash X \qquad B \vdash X}{A \vee B \vdash X}\,\vee l \qquad \frac{X \vdash A \circ B}{X \vdash A \vee B}\,\vee r \qquad \frac{X \circ * Y \vdash A \qquad B \vdash * X \circ Y}{A \to B \vdash * X \circ Y}\,\to l$$

$$\frac{X \circ A \vdash B}{X \vdash A \to B}\,\to r \qquad \frac{A \vdash X}{\Box A \vdash \bullet X}\,\Box l \qquad \frac{X \vdash \bullet A}{X \vdash \Box A}\,\Box r$$

$$\frac{* \bullet * A \vdash X}{\Diamond A \vdash X}\,\Diamond l \qquad \frac{X \vdash A}{* \bullet * X \vdash \Diamond A}\,\Diamond r \qquad \frac{\bullet A \vdash X}{\blacklozenge A \vdash X}\,\blacklozenge l$$

$$\frac{X \vdash A}{\bullet X \vdash \blacklozenge A}\,\blacklozenge r \qquad \frac{A \vdash X}{\blacksquare A \vdash * \bullet * X}\,\blacksquare l \qquad \frac{X \vdash * \bullet * A}{X \vdash \blacksquare A}\,\blacksquare r$$

Define the functions l and r from $\mathfrak{Stt}(\mathcal{L}_{Kt})$ into $\mathbf{For}\mathcal{L}_{Kt}$.

$$\begin{aligned} l(A) &= A & r(A) &= A \\ l(\mathbf{I}) &= \top & r(\mathbf{I}) &= \bot \\ l(*X) &= \neg r(X) & r(*X) &= \neg l(X) \\ l(X \circ Y) &= l(X) \wedge l(Y) & r(X \circ Y) &= r(X) \vee r(Y) \\ l(\bullet X) &= \blacklozenge l(X) & r(\bullet X) &= \Box r(X) \end{aligned}$$

It is easy to check that δKt is amenable.

Proposition 6. *Every logical rule with the exception of $\Box l$, $\Diamond r$, $\blacklozenge r$ and $\blacksquare l$ is invertible.*

Theorem 7. *There is a proper structural rule extension of δKt for axiomatic extension of Kt with acyclic $\mathcal{I}_2$ axioms.*

A procedure to define proper structural display logic rules for primitive axiomatic extensions of K and Kt was introduced by Kracht's [14]. A primitive tense axiom has the form $A \to B$ where both A and B are constructed from propositional variables and $\top$ using $\{\wedge, \vee, \Diamond, \blacklozenge\}$ and A contains each propositional variable at most once.

Kracht's method to extract structural rules is very different from our method, and relies on being able to transform the axiom into a primitive tense formula. Eg. the axiom $\Box A \to A$ must be rewritten as the primitive tense formula $A \to \Diamond A$. [17] rewrites the familiar B axiom $A \to \Box\Diamond A$ in the primitive tense form as $(A \wedge \Diamond B) \to \Diamond(B \wedge \Diamond A)$.

Example 5. Fig. 1 contains some examples of $\mathcal{I}_2$ axioms (see Table IV in [17]) and corresponding rules generated using our procedure. Contrast our strucutural rule for the B axiom with the rule generated by Kracht's method (see [17]):

$$\frac{* \bullet *(X \circ * \bullet *Y) \vdash Z}{Y \circ * \bullet *X \vdash Z}$$

In contrast with out method, Kracht's result provides a characterisation (a necessary and sufficient condition). Indeed

Theorem 8 (Kracht). *Let L be a tense logic. Then L is an axiomatic extension of Kt by primitive tense axioms iff there is a proper structural rule extension of δKt for L.*

It follows that every acyclic $\mathcal{I}_2$ axiom is equivalent to a primitive tense axiom.

References

1. Avron, A.: A Constructive Analysis of RM. J. of Symbolic Logic 52(4), 939–951 (1987)
2. Belnap, N.D.: Display Logic. Journal of Philosophical Logic 11(4), 375–417 (1982)
3. Blackburn, P., de Rijke, M., Venema, Y.: Modal logic. Cambridge Tracts in Theoretical Computer Science (2001)
4. Brotherston, J.: Bunched Logics Displayed. Studia Logica 100(6), 1223–1254 (2012)
5. Ciabattoni, A., Galatos, N., Terui, K.: From axioms to analytic rules in nonclassical logics. In: Proceedings of LICS 2008, pp. 229–240 (2008)
6. Ciabattoni, A., Straßburger, L., Terui, K.: Expanding the realm of systematic proof theory. In: Grädel, E., Kahle, R. (eds.) CSL 2009. LNCS, vol. 5771, pp. 163–178. Springer, Heidelberg (2009)
7. Fitting, M.: Proof Methods for Modal and Intuitionistic Logics. Dordrecht, Holland (1983)
8. Gentzen, G.: The collected papers of Gerhard Gentzen. Studies in Logic and the Foundations of Mathematics, Amsterdam (1969); Szabo, M.E. (ed.)
9. Goré, R.: Substructural Logics on Display. Logic Journal of the IGPL 6(3), 451–504 (1998)
10. Goré, R.: Gaggles, Gentzen and Galois: how to display your favourite substructural logic. Logic Journal of the IGPL 6(5), 669–694 (1998)
11. Goré, R., Postniece, L., Tiu, A.: On the Correspondence between Display Postulates and Deep Inference in Nested Sequent Calculi for Tense Logics. Logical Methods in Computer Science 7(2) (2011)
12. Guglielmi, A.: A system of interaction and structure. ACM Transaction on Computational Logic 8(1), 1–64 (2007)
13. O'Hearn, P., Pym, D.: The Logic of Bunched Implications. Bulletin of Symbolic Logic 5(2), 215–244 (1999)
14. Kracht, M.: Power and weakness of the modal display calculus. In: Proof Theory of Modal Logic, pp. 93–121. Kluwer (1996)
15. Lellmann, B., Pattinson, D.: Constructing Cut Free Sequent Systems With Context Restrictions Based on Classical or Intuitionistic Logic. In: Lodaya, K. (ed.) ICLA 2013. LNCS, vol. 7750, pp. 148–160. Springer, Heidelberg (2013)
16. Negri, S.: Proof analysis in non-classical logics. In: Logic Colloquium 2005, pp. 107–128 (2007)
17. Wansing, H.: Displaying modal logic. Kluwer Academic Publishers (1998)
18. Wansing, H.: Constructive negation, implication, and co-implication. Journal of Applied Non-Classical Logics 18, 341–364 (2008)
19. Wolter, F.: On logics with coimplication. Journal of Philosophical Logic 27(4), 353–387 (1998)

The Same, Similar, or Just Completely Different? Equivalence for Argumentation in Light of Logic

Sjur Kristoffer Dyrkolbotn*

Durham Law School, Durham University, UK
s.k.dyrkolbotn@durham.ac.uk

Abstract. In recent years, argumentation theory and logic have moved closer to each other, a development due in large part to Dung's mathematically precise definition of an abstract argumentation framework as a digraph and the intuitively plausible semantics for argumentation that can be formulated using this structure. This work raises some questions, however, regarding the relationship between an abstract argumentation framework – a directed graph – and the underlying argumentative structure that it is taken to represent. One such question, which we study in this paper, is the question of when two arguments should be considered *the same*, a question which has been surprisingly controversial, and which also, as we will demonstrate, gives rise to interesting technical results and future challenges.

1 Introduction

Abstract argumentation in the style of Dung [11] has gained much popularity in the AI-community, we point to [10] for an overview. Its appeal seems due in large part to the mathematically simple and precise basis which Dung uses to introduce his semantic notions. His *argumentation frameworks* are nothing but directed graphs, and semantics for abstract argumentation are defined in terms of the underlying graph-structure. This suggests *logical* modeling, a challenge that has been taken on in recent work, e.g., in [8,16,15]. Whereas these contributions rely on modal logic, we will follow [13,14,20] and use Łukasiewicz logic to study argumentation. The formalization we propose is very simple, in some sense naive, but our purpose is not to provide a new form of representation for the semantic structures themselves, which are already nicely represented in terms of digraphs. What we want is a logic for reasoning about them, to allow us to state and explore their properties in a succinct way. In the following, we take this route to shed new light on the question of equivalence.

* I am very thankful for helpful comments from Truls Pedersen, Piotr Kaźmierczak and Michał Walicki. I also thank BNC@ECAI 2012 for accepting a preliminary version of this paper for a presentation at their conference, and the anonymous reviewers, for making helpful suggestions for improvements.

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 96–110, 2013.

In Section 2, we briefly present abstract argumentation and Łukasiewicz logic, as well as the connection between them. In Section 3, we exploit this connection to describe notions of equivalence arising from well-known semantics for argumentation. Then, in Section 4, we address the problem of characterizing logical equivalence, concluding with a result which employs a new notion of bisimulation to give a sufficient condition for when two argumentation frameworks behave the same way, and have the same logical properties.

2 Background

We fix a countably infinite set of atoms Π, which we think of as abstract arguments. Given some set $\mathcal{A} \subseteq \Pi$, an *argumentation framework* over $\mathcal{A}$, a framework for short, is a digraph, $\mathsf{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ with $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ a set of directed edges, called the *attack relation*. If $(a, b) \in \mathcal{R}$, we say that *a attacks* the argument *b*. We use the notation $\mathcal{R}^-(x) = \{y \mid (y, x) \in \mathcal{R}\}$ and $\mathcal{R}^+(x) = \{y \mid (x, y) \in \mathcal{R}\}$, extended pointwise to sets, such that, for instance, $\mathcal{R}^+(X) = \bigcup_{x \in X} \mathcal{R}^+(x)$. For general relations $\alpha \subseteq X \times Y$, we drop + as a superscript and use $\alpha(x) = \{y \mid (x, y) \in \alpha\}$ and $\alpha^-(y) = \{x \mid (x, y) \in \alpha\}$. This also extends pointwise to sets. For two frameworks F and F_2, we say they are isomorphic if there is a (graph-)isomorphism between them, i.e., a bijective function $i : \mathcal{A} \to \mathcal{A}_2$ such that $\mathcal{R}_2^+(i(a)) = i(\mathcal{R}^+(a))$

A framework $\mathsf{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ is a *subframework* of a framework $\mathsf{F}_2 = \langle \mathcal{A}_2, \mathcal{R}_2 \rangle$ iff $\mathcal{A} \subseteq \mathcal{A}_2$ and $\mathcal{R} \subseteq \mathcal{R}_2$. A subset of arguments $X \subseteq \mathcal{A}$ gives rise to the *induced subframework* $\mathsf{X} = \langle X, \mathcal{R}_X \rangle$ with $\mathcal{R}_X = \{(x, y) \in \mathcal{R} \mid x, y \in X\}$. $\mathsf{F} \setminus X$ denotes the subframework of F induced by $\mathcal{A} \setminus X$. A *backwards infinite walk* is a sequence $\lambda = x_1 x_2 x_3 \ldots$ such that $x_{i+1} \in \mathcal{R}^-(x_i)$ for all $i \geq 1$. Notice that in finite argumentation frameworks, there can be backwards infinite walks, but they must involve one or more arguments twice, i.e., they involve cycles.

Some of the most well-known semantics for argumentation, first introduced in [11] and [19,9] (semi-stable semantics), are given in the following definition.

Definition 2.1. *Given any argumentation framework* $\mathsf{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ *and a subset* $A \subseteq \mathcal{A}$*, we define* $\mathcal{D}(A) = \{x \in \mathcal{A} \mid \mathcal{R}^-(x) \subseteq \mathcal{R}^+(A)\}$*, the set of vertices* defended *by A. We say that*

- *A is* conflict-free *if* $\mathcal{R}^+(A) \subseteq \mathcal{A} \setminus A$*, i.e., if there are no two arguments in A that attack each other.*
- *A is* admissible *if it is conflict free and* $A \subseteq \mathcal{D}(A)$*. The set of all admissible sets in* F *is denoted* $a(\mathsf{F})$*.*
- *A is* complete *if it is conflict free and* $A = \mathcal{D}(A)$*. The set of all complete sets in* F *is denoted* $c(\mathsf{F})$*.*
- *A is* preferred *if it is admissible and not strictly contained in any admissible set. The set of all preferred sets in* F *is denoted* $p(\mathsf{F})$*.*
- *A is* stable *if* $\mathcal{R}^+(A) = \mathcal{A} \setminus A$*. The set of all stable sets in* F *is denoted* $s(\mathsf{F})$
- *A is* semi-stable *if it is admissible and there is no admissible set B such that* $A \cup \mathcal{R}^+(A) \subset B \cup \mathcal{R}^+(B)$*. The set of all semi-stable sets in* F *is denoted by* $ss(\mathsf{F})$*.*

We also mention the grounded semantics, which can be seen as an application of the complete semantics. It collects those arguments that are present in every complete set, i.e., it can be defined as $g(\mathsf{F}) = \bigcap c(\mathsf{F})$.[1]

For any $\epsilon \in \mathcal{S} = \{a, c, p, s, ss\}$, we also say that $S \in \epsilon(\mathsf{F})$ is an *extension* (of the type prescribed by ϵ). Given $S \in \epsilon(\mathsf{F}), x \in \mathcal{A}$ we say that x is *accepted* (by S) if $x \in S$ and that it is *defeated* if $\mathcal{R}^-(x) \cap S \neq \emptyset$. If neither of these obtain, x is said to be *rejected*. Moreover, such a status holds *skeptically* if it obtains with respect to *all* $S \in \epsilon(\mathsf{F})$, and *credulously* if it obtains for *some* $S \in \epsilon(\mathsf{F})$.

Clearly, these notions have *logical* structure. Viewing x as a formula, we might as well say that it is skeptically accepted just in case it follows as a logical consequence from the framework and credulously accepted if it is mutually consistent with it. Moreover, if x is defeated, we should be able to say, taking the logical view further, that $\neg x$ is true. Notice that the notion of rejection does not coincide with defeat, the obvious example being the self-attacking argument, which is skeptically rejected under any semantics, yet cannot be defeated. It follows that the logic we need for expressing properties of arguments is not classical, but three-valued.[2]

We will use the following (infinitary) propositional language $\mathcal{L}$:

$$\phi ::= a \;\mid\; \bigwedge \Phi \;\mid\; \neg\phi \;\mid\; \phi \to \phi$$

where $a \in \mathcal{A}$ and $\Phi \subseteq \mathcal{L}$ is a (possibly infinite) set of formulas. We remark that if we restrict attention to finite frameworks, the finitary language will always suffice. We define $\phi \wedge \psi, \phi \leftrightarrow \psi$ and $\bot$ (contradiction) standardly, and we will also use the following abbreviations.

$$\bigvee \Phi ::= \neg \bigwedge \{\neg\phi \mid \phi \in \Phi\}, \quad \mathbf{!^w}\phi ::= \neg\phi \to \phi, \quad \mathbf{!}\phi ::= \neg\mathbf{!^w}\neg\phi$$

The appropriate semantics is Łukasiewicz's three-valued one, defined as follows, see for instance [5].

Definition 2.2. *Given an assignment $\pi : \Pi \to \{0, 1/2, 1\}$ its extension, $\bar{\pi}$, is defined inductively on the complexity of formulas.*

[1] This said, the grounded set has attracted quite some interest in more application-oriented works on argumentation. It is useful, in particular, because it can be computed in linear time [11].

[2] The logical view allows us to offer further qualification for our claim that it might be more appropriate to view the grounded set as an application of the complete set. Indeed, a similar perspective can be taken on all so-called *unique status* semantics, of which the grounded is an example (like ideal and eager semantics [12,6]). They start from existing semantics and impose some more or less sensible rule for "picking the winners", arriving at a unique extension. It follows – in logical terms – that anything satisfiable (credulously acceptable) is tautologically true (skeptically accepted), and this holds already at the level of individual atoms/arguments, effectively rendering any logic based on the semantics, like we propose here, void of content. It seems to us, however, that the difficulty on agreeing on the correct rule for picking winners suggests that a more flexible, and structurally rich, approach, by way logic, is often more appropriate than committing oneself to a particular unique-status approach.

- $\bar{\pi}(a) = \pi(a)$ *for all* $a \in \Pi$
- $\bar{\pi}(\neg\phi) = 1 - \bar{\pi}(\phi)$
- $\bar{\pi}(\phi \to \psi) = min\{1, (1 - \bar{\pi}(\phi)) + \bar{\pi}(\phi)\}$
- $\bar{\pi}(\bigwedge \Phi) = min\{\bar{\pi}(\phi) \mid \phi \in \Phi\}$

The consequence relation of Łukasiewicz logic is $\models_{\text{Ł}} \subseteq 2^{\mathcal{L}} \times \mathcal{L}$*, defined such that* $\Phi \models_{\text{Ł}} \phi$ *iff for all* $\pi : \Pi \to \{0, 1/2, 1\}$*, we have that* $\bar{\pi}(\phi) = 1$ *whenever* $\bar{\pi}(\psi) = 1$ *for all* $\psi \in \Phi$*.*

The meaning of the abbreviations $\mathbf{!}\phi$ and $\mathbf{!}^{\mathbf{w}}\phi$, due to Tarski, can now be explained: they say of ϕ that it is true and not false respectively, and they do so in a definitive way; they cannot themselves obtain the status of being undetermined (evaluate to $\frac{1}{2}$). For $\pi : \Pi \to \{0, 1/2, 1\}$ we define $\pi^1 = \{x \mid \pi(x) = 1\}$ and similarly for π^0 and $\pi^{\frac{1}{2}}$.

Using Łukasiewicz logic, we can encode extensions of a framework as logical formulas. We define, in particular, for all $A \subseteq \mathcal{A}$, the (possibly infinitary) conjunction $\phi_{id}(A)$:

$$\phi_{id}(A) = \bigwedge_{x \in A} x \wedge \bigwedge_{y \in \mathcal{R}^+(A)} \neg y \wedge \bigwedge_{z \in \mathcal{A} \setminus (A \cup \mathcal{R}^+(A))} z \leftrightarrow \neg z$$

$\phi_{id}(A)$ characterizes A logically by being made true by exactly those assignments that agree with A on $\mathcal{A}$. That is, such that for all $\pi : \Pi \to \{0, 1/2, 1\}$, we have $\pi \models \phi_{id}(A)$ if and only if $\pi^1 \cap \mathcal{A} = A$ and $\pi^0 \cap \mathcal{A} = \mathcal{R}^+(A)$. We can now write every extension as the following formula, defined for all $\epsilon \in \mathcal{S}$:

$$\phi_\epsilon(\mathsf{F}) = \mathbf{!} \bigvee \{\phi_{id}(A) \mid A \in \epsilon(\mathsf{F})\}$$

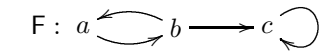
Clearly, this representation, being constructed by brute force by taking the disjunction of all extensions that the framework admits under ϵ, provides a faithful view. We have, in particular, the following simple fact, the proof of which is trivial and omitted.

Fact 2.3. *For any* $\mathsf{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ *and any* $\epsilon \in \mathcal{S}$*, we have* $\epsilon(\mathsf{F}) = \{\pi^1 \cap \mathcal{A} \mid \pi \models_{\text{Ł}} \phi_\epsilon(\mathsf{F}))\}$

The representation provides what we want, namely a logic for reasoning about *consequences* of a given semantics.[3] As we observed earlier, credulous acceptance should amount to mutual consistency, and it does; we have that $a \in \mathcal{A}$ is credulously accepted just in case $\phi_\epsilon(\mathsf{F}) \wedge a \not\models_{\text{Ł}} \bot$. Since $\phi_\epsilon(\mathsf{F})$ can never evaluate to $\frac{1}{2}$ (because we put $\mathbf{!}$ in front of the conjunction), we also have the relevant instance

[3] We mention that Łukasiewicz logic offer a less trivial characterization of $\phi_c(\mathsf{F})$. In [14], it is established that $\phi_c(\mathsf{F})$ is in fact equivalent to the formula $\{x \leftrightarrow \bigwedge\{\neg y \mid y \in \mathcal{R}^-(x)\} \mid x \in \mathcal{A}\}$, which is linear in the size of the framework, and thus provides a representation of the complete semantics which is more useful, computationally speaking. See also [1], which builds on a series of similar observations and develops them within a framework of quantified boolean logic.

of the deduction theorem, i.e., the above is equivalent to $\not\models_L \phi_\epsilon(\mathsf{F}) \wedge a \rightarrow \bot$. Skeptical acceptance, on the other hand, becomes logical consequence, such that a is skeptically accepted with respect to ϵ just in case $\models_L \phi_\epsilon(\mathsf{F}) \rightarrow a$. For a more complex example of what we can express, consider the following scenario, with a framework for which it is a logical consequence that if we defeat b, then we can neither accept nor defeat c. This, in particular, is what the formula on the right says in logical language.



$$\phi_a(\mathsf{F}) \models_L \neg b \rightarrow c \leftrightarrow \neg c$$

Although our logical representation is simple, examples such as these suggests its usefulness. Since we have a logical language, we can now express arbitrarily complex properties and interactions, as long as they arise in a truth-functional way from the basic notions of acceptance, defeat and rejection.

3 Notions of Equivalence for Argumentation

In the literature, if equivalence is considered, it seems typical to consider a very simple notion whereby two frameworks are regarded as equivalent if they have *the same* sets of extensions. In terms of logic, this equivalence arises at the level of the theories describing the extensions; it obtains just in case the theories have the same set of satisfying assignments, i.e., just in case they are logically equivalent. Another notion of equivalence, which is more subtle, and also seems more interesting from the point of view of applications, is the notion that follows as a *logical consequence* of the extensions. Rather than saying that the frameworks are the same, this is the notion which says that two arguments, in the same framework, *behave the same way*. Such a notion of equivalence has, as far as we are aware, only been studied in one previous paper [17]. In this work, however, the focus is on the grounded semantics, and on a notion of semantics which introduces *new distinctions* between arguments, distinctions that do *not* arise from the grounded semantics itself, but from new notions that the authors find appealing. While introducing new notions might be appropriate, it is not the same as investigating equivalences that arise from semantic notions already established, which is our aim here.

Given the simple logical footing we have provided above, it is simple to account for the behavioral equivalences we have in mind, and to do so in a way that makes it clear that they arise from existing semantics, and do not represent new proposals. It seems, in particular, that the notions of equivalence we capture in the following definition do not require, and, indeed, can not possibly be given any other justification than that which one might find occasion to provide in support of ϵ itself.

Definition 3.1. *Structural equivalence: For any two frameworks* F *and* F_2*, we say that they are* structurally equivalent *under* $\epsilon \in \mathcal{S}$ *if* $\models_{\mathrm{L}} \phi_\epsilon(\mathsf{F}) \leftrightarrow \phi_\epsilon(\mathsf{F}_2)$.

Behavioral equivalence: For any framework F *and* $a, b \in \mathcal{A}$*, we say that* a *and* b *are* behaviorally equivalent *under* $\epsilon \in \mathcal{S}$ *if* $\models_{\mathrm{L}} \phi_\epsilon(\mathsf{F}) \rightarrow (a \leftrightarrow b)$.

The following trivial fact explicates the equivalent semantic characteristics, and both points follows trivially from Fact 2.3.

Fact 3.2. *For any two frameworks* F, F_2 *and arguments* $a, b \in \mathcal{A}$*:*

(1) F *and* F_2 *are structurally equivalent under* $\epsilon \in \mathcal{S}$ *iff* $\epsilon(\mathsf{F}) = \epsilon(\mathsf{F}_2)$
(2) a *and* b *are behaviorally equivalent under* $\epsilon \in \mathcal{S}$ *iff* $\forall A \in \epsilon(\mathsf{F}) : a \in A \Leftrightarrow b \in A$

As we mentioned, the notion of structural equivalence has already received some attention and recognition in the argumentation community. Moreover, nice results have been obtained in [18,3], where the authors investigate conditions under which structural equivalence is preserved under various ways in which an argumentation framework can be *extended*. This is particularly interesting because extending a framework is not a logically monotone operation; adding arguments and attacks to a framework does not, for any of the semantics we consider, correspond to adding formulas to the theories characterizing its extensions. Rather, it introduces new notions of *logical revision* that have yet to be defined and explored from a (monotonic) logical point of view. We think doing this is an interesting challenge for future work, but we do not pursue it further in this paper. Moreover, we note that the results obtained in [18,3] are limiting in the sense that they demonstrate that structural equivalence tends to be preserved under expansion/revision only in case of frameworks that are *isomorphic*. This seems to indicate that the notions of expansion needs to be restricted, or else that structural equivalence itself might be of limited relevance.[4] It also suggests the appropriateness of considering other notions of equivalence, and in the following we will offer a contribution in this regard, by *lifting* the notion of behavioral equivalence such it can also be applied to relate different, but behaviorally similar, frameworks.

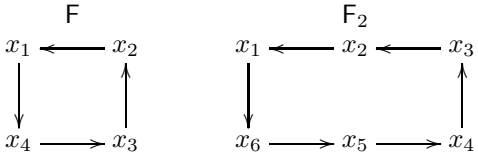
First, let us first consider a class of frameworks that serves to motivate what is to follow, namely those that consist of a single cycle of attack. Under all semantics we consider here, these fall into one of two categories: those that have even length and those that have odd length.[5]

Now, it is a basic fact of argumentation that no odd cycle has a non-empty admissible set, and hence no argument is acceptable with respect to any semantics from Definition 2.1, in any odd cycle. In particular, all odd cycles are structurally equivalent in the sense of Definition 3.1. So far so good, but for the even cycles, structural equivalence fails: no two even cycles of different length

[4] The only straightforward notion we know of which does not give rise to severe collapses, is the notion of *weak expansion*. But as noted in [3], it appears to be of limited interest, since it disallows introducing any new attacks on arguments already present in the framework. We also mention [4], which introduces another notion of equivalence based on considering the *minimal change* needed to enforce a set of arguments. This notion (or related/competing ones) might be interesting with respect to specific heuristics for instantiation, as explored in [7], but will not be addressed further here.

[5] Indeed, some have claimed that there is only one category; that even and odd-length cycles should be treated the same way since they are both ungrounded chains of arguments. This requires new semantic notions however, as presented in [2].

are structurally equivalent. In our opinion, this is unsatisfying. To illustrate further, consider two arbitrary even length cycles, F and F_2 depicted below.



How do we reason about F and F_2? Well, suppose that the argument x_1 from F has some proponent. Then this proponent should recognize that his argument is attacked by the argument x_2 and then become a proponent of argument x_3 as well, since this argument attacks x_2 and therefore defends x_1. This is when the reasoning stops in F, the proponent notices at this point that although x_4 attacks x_3, it is in turn attacked by x_1, so his argument, while not conclusive, is at least admissible. In F_2, the story, in all essential aspects, is the same; a proponent of x_1 realizes he should also support x_3, but now, since the cycle is longer, he also comes to support x_5 for the *same reason*.

The observation we want to make is that even when the length of cycles differ, if they have the same parity, they are still – logically speaking – the same. But on the level of the frameworks, we have no notion of equivalence that allows us to recognize this sense of sameness. On the level of arguments, however, the notion of behavioral equivalence *does* detect it. For it follows, for any even cycle, that every other vertex is behaviorally equivalent. In F_2, for instance, we have $\phi_s(\mathsf{F}_2) \models_L (x_1 \leftrightarrow x_3 \leftrightarrow x_5) \wedge (x_2 \leftrightarrow x_4 \leftrightarrow x_6)$. Now, in our opinion, this example serves to illustrate two things. First, that the notion of behavioral equivalence is often more informative and useful than that of structural equivalence, and, second, that behavioral equivalence should indeed be lifted to frameworks.

Formally, it seems natural to do so by first defining the *behavioral contraction* of a framework, that is, the framework resulting from collapsing all arguments that are behaviorally equivalent. Given a framework F, such a contraction is an obvious candidate for a canonical framework, since it captures, in a minimal way, how F behaves under ϵ. Towards formalization, we will, given $\epsilon \in \mathcal{S}$, start by partitioning $\mathcal{A}$ into equivalence classes $\bigcup_{a \in \mathcal{A}} \{[a]_\epsilon = \{b \in \mathcal{A} \mid \phi_\epsilon(\mathsf{F}) \models_L a \leftrightarrow b\}\}$. Then we can provide the following definition.

Definition 3.3. *Given a framework $\mathsf{F} = \langle \mathcal{A}, \mathcal{R} \rangle$, its behavioral contraction under $\epsilon \in \mathcal{S}$ is the framework $\mathcal{C}_\epsilon(\mathsf{F}) = \langle \mathcal{C}_\epsilon(\mathcal{A}), \mathcal{C}_\epsilon(\mathcal{R}) \rangle$ where*

$$\mathcal{C}_\epsilon(\mathcal{A}) = \{[a]_\epsilon \mid a \in \mathcal{A}\}$$

and

$$([a]_\epsilon, [b]_\epsilon) \in \mathcal{C}_\epsilon(\mathcal{R}) \Leftrightarrow \exists a' \in [a]_\epsilon, b' \in [b]_\epsilon : (a', b') \in \mathcal{R}$$

Given two frameworks F and F_2, we say that they are behaviorally equivalent *if there are frameworks $i(\mathcal{C}_\epsilon(\mathsf{F}))$ and $i_2(\mathcal{C}_\epsilon(\mathsf{F}_2))$, isomorphic to $\mathcal{C}_\epsilon(\mathsf{F})$ and $\mathcal{C}_\epsilon(\mathsf{F}_2)$ respectively, such that $\models_{\mathrm{L}} \phi_\epsilon(i(\mathcal{C}_\epsilon(\mathsf{F}))) \leftrightarrow \phi_\epsilon(i_2(\mathcal{C}_\epsilon(\mathsf{F}_2)))$*

That is, we regard two frameworks as being behaviorally the same if their behavioral contractions are structurally equivalent under renaming of arguments.

In fact, it seems to us, coming from logic, that this definition is fairly standard, following, for instance, the notions of equivalence arising from the use of various kinds of bisimulations in modal logic. So, is it adequate? Does it indeed capture a notion of equivalence arising logically from the underlying semantics? It is not hard to answer this in the affirmative. Formally, for $\phi \in \mathcal{L}$, we let $\Pi(\phi)$ denote the atoms appearing in ϕ. Then, for any function $f : \Pi(\phi) \to \Pi$, we denote by $f(\phi)$ the formula resulting from replacing x in ϕ by $f(x)$ for each $x \in \Pi(\phi)$. Now, given frameworks F and F_2 which have behavioral contractions that admit isomorphisms $i(\mathcal{C}_\epsilon(\mathsf{F}))$ and $i_2(\mathcal{C}_\epsilon(\mathsf{F}_2))$ such that $\models_L \phi_\epsilon(i(\mathcal{C}_\epsilon(\mathsf{F}))) \leftrightarrow \phi_\epsilon(i_2(\mathcal{C}_\epsilon(\mathsf{F}_2)))$, we let $f : \mathcal{A} \to \mathcal{C}_\epsilon(\mathcal{A})$ and $f_2 : \mathcal{A}_2 \to \mathcal{C}_\epsilon(\mathcal{A}_2)$ be defined by $f(a) = [a]_\epsilon$, $f_2(a_2) = [a_2]_\epsilon$ for all $a \in \mathcal{A}, a_2 \in \mathcal{A}_2$. Then it is easy to verify that our definition is *logically adequate*, in the sense that logical consequences of F and F_2 (under $\epsilon \in \mathcal{S}$) are preserved under behavioral equivalence, in the following sense:

$$\begin{array}{l}(1) : \phi_\epsilon(\mathsf{F}) \models_L \phi \Leftrightarrow i(f(\phi_\epsilon(\mathsf{F}))) \models_L i(f(\phi)) \Leftrightarrow i_2(f_2(\phi_\epsilon(\mathsf{F}_2))) \models_L i_2(f_2(\phi)) \\ \quad \text{for all } \phi \text{ such that } \Pi(\phi) \subseteq \mathcal{A} \\ \\ (2) : \phi_\epsilon(\mathsf{F}_2) \models_L \phi \Leftrightarrow i_2(f_2(\phi_\epsilon(\mathsf{F}_2))) \models_L i_2(f_2(\phi) \Leftrightarrow i(f(\phi_\epsilon(\mathsf{F})))) \models_L i(f(\phi)) \\ \quad \text{for all } \phi \text{ such that } \Pi(\phi) \subseteq \mathcal{A}_2\end{array} \tag{3.4}$$

The usefulness of this result depends on how difficult it is to establish behavioral equivalences, either between arguments, or between frameworks, and in the following we will study this problem. However, its appropriateness seems beyond doubt. If we can establish behavioral equivalence, we can move freely between frameworks, analyzing the argumentation scenario at hand in the structure for which it is easiest to do so. It is a logical truth, in particular, that *no information* is lost by doing so. If the feeling is to the contrary, if it appears that the notion gives rise to unacceptable collapses or over-simplifications, this is due to the semantic notions themselves, not the notion of equivalence they give rise to.

4 Characterizing Behavioral Equivalence

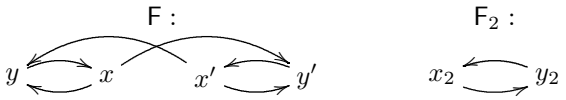
Having demonstrated how a notion of behavioral equivalence arises from argumentation semantics, and having provided a logical basis for it, we now turn to the question of obtaining results that structurally characterizes when two arguments and/or frameworks are behaviorally equivalent. First, we investigate the link with relations that happen to both preserve and reflect all extensions of a given semantics.

Definition 4.1. *Given two frameworks* F *and* F_2, $\epsilon \in \mathcal{S}$ *and a relation* $\alpha \subseteq \mathcal{A} \times \mathcal{A}$, *we say that* α *is an* ϵ*-relation, if the following two conditions hold:*

(1) For all $A \in \epsilon(\mathsf{F}), \alpha(A) \in \epsilon(\mathsf{F}_2)$
(2) For all $A_2 \in \epsilon(\mathsf{F}_2), \alpha^-(A_2) \in \epsilon(\mathsf{F})$

If there is an ϵ*-relation between* F *and* F_2, *we write* $\mathsf{F} \equiv^\epsilon \mathsf{F}_2$

Clearly, if $\mathsf{F} \equiv^{\epsilon} \mathsf{F}_2$, there is a close relationship between the arguments, and how they interact, in these two frameworks. After all, any ϵ-extension in one gives rise to an ϵ-extension in the other. Does it follow that two argumentation frameworks are behaviorally equivalent if and only if there is an ϵ-relation between them? It turns out that the answer is yes for the preferred, semi-stable and stable semantics, and no for the complete and admissible semantics. To show the latter claim first, consider the following two argumentation frameworks.



It is not hard to verify that under admissible and complete semantics, F and F_2 are both their own behavioral contractions – no two arguments are equivalent in either framework – and then, since they are not structurally equivalent (under any renaming), it follows that they are not behaviorally equivalent. Still, the relation $\alpha = \{(x, x_2), (x', x_2), (y, y_2), (y', y_2)\}$ both preserves and reflects admissible and complete sets. It follows that existence of an ϵ-relation does not imply behavioral equivalence for these semantics. Next, let us show that existence of an ϵ-relation *does* imply behavioral equivalence with respect to preferred, semi-stable and stable semantics.

Proposition 4.2. *For any two frameworks* F *and* F_2 *and any* $\epsilon \in \{p, ss, s\}$*, we have* $\mathsf{F} \equiv^{\epsilon} \mathsf{F}_2$ *if and only if* F *and* F_2 *are behaviorally equivalent.*

Proof. ($\Leftarrow$) Define $f : \mathcal{A} \to \mathcal{C}_{\epsilon}(\mathsf{F}), f_2 : \mathcal{A}_2 \to \mathcal{C}_{\epsilon}(\mathsf{F}_2)$ by $f(a) = [a]_{\epsilon}, f_2(a_2) = [a_2]_{\epsilon}$ for all $a \in \mathcal{A}, a_2 \in \mathcal{A}_2$. Clearly, we have that f and f_2 are ϵ-relations between F, F_2 and $\mathcal{C}_{\epsilon}(\mathsf{F}), \mathcal{C}_{\epsilon}(\mathsf{F}_2)$ respectively. Since F and F_2 are behaviorally equivalent, there are isomorphisms i, i_2 such that $\models_L \phi_{\epsilon}(i(\mathcal{C}_{\epsilon}(\mathsf{F}))) \leftrightarrow \phi_{\epsilon}(i_2(\mathcal{C}_{\epsilon}(\mathsf{F}_2)))$, and it follows that $f_2^- \circ i_2^- \circ i \circ f$ is an ϵ-relation between F and F_2. ($\Rightarrow$) The orbit of α partitions $\mathcal{A}$ and $\mathcal{A}_2$ into equivalence classes $[a]_{\alpha}$ and $[a_2]_{\alpha}$, defined for all $a \in \mathcal{A}$, $a_2 \in \mathcal{A}_2$ as the least set A (resp. A_2) such that $a \in A$ (resp. $a_2 \in A_2$) and $\alpha^-(\alpha(A)) = A$ (resp. $\alpha(\alpha^-(A_2)) = A_2$). To complete the proof, we will show that the partitioning of $\mathcal{A}$ and $\mathcal{A}_2$ into behavioral equivalence classes is a refinement of the partitioning induced by α. By definition of orbit and the fact that equivalence is transitive, it suffices to show (1): for all $a, b \in \mathcal{A}$, if $a_2 \in \alpha(a) \cap \alpha(b)$ then $\phi_{\epsilon}(\mathsf{F}) \models_L a \leftrightarrow b$ and (2): for all $a_2, b_2 \in \mathcal{A}$, if $a \in \alpha^-(a_2) \cap \alpha^-(b_2)$ then $\phi_{\epsilon}(\mathsf{F}_2) \models_L a_2 \leftrightarrow b_2$. We show (1) for $\epsilon = p$, the other cases are similar. Assume towards contradiction that $a_2 \in \alpha(a) \cap \alpha(b)$ for some preferred set P such that $a \in P, b \notin P$. Since α preserves and reflects extensions, we know that $P' = \alpha^-(\alpha(P))$ is a preferred set. Clearly, we have $P \subseteq P'$ and, moreover, we have $b \in P'$ since $a_2 \in \alpha(b) \cap \alpha(a) \subseteq \alpha(P)$. This contradicts maximality of P, required by definition of preferred sets. □

Note that a-relations are c-relations, and are also ϵ-relations for all other semantics. So existence of an a-relation (and c-relation) also establishes behavioral equivalence with respect to the other semantics. Moreover, note that the failure of a-relations to establish a-behavioral equivalence is down to some arguments

not obtaining any definite status. It holds, in particular, for every $\epsilon \in \mathcal{S}$ and ϵ-relation α that if $a_2 \in \alpha(a) \cap \alpha(b)$, then there can not be any admissible A such that $a \in A$ and $b \in \mathcal{R}^+(A)$. The reason is that this would imply existence of $c \in \mathcal{R}^-(b) \cap A$, which in turn would imply $a, b, c \in A' = \alpha^-(\alpha(A)$, meaning A' is not independent, hence not admissible, contradicting that α is an a-relation.

Moreover, the argument used in the proof of Proposition 4.2 also shows that for any two $a, b \in \mathcal{A}$, if they are in the same equivalence class induced by the orbit of some a-relation, it means that they are equivalent in the looser sense of satisfying $\forall A \in a(\mathsf{F}) : a \in A \Rightarrow \exists A' \supseteq A : A' \in a(\mathsf{F}) \wedge b \in A'$.

Having found an alternative characterization of behavioral equivalence, using ϵ-relations, we can study behavioral equivalence by proxy, working with the more straightforward and manageable Definition 4.1 in place of Definition 3.1. We do so in the following, when we investigate bisimulations.[6]

Definition 4.3. *Given argumentation frameworks* F *and* F_2*, a relation* $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ *is said to be a bisimulation if we have:*
forth*: For every* $x \in \mathcal{A}$*,* $y \in \mathcal{R}^-(x)$*, for all* $x_2 \in \beta(x)$ *there is* $y_2 \in \mathcal{R}_2^-(x_2) \cap \beta(y)$
back*: For every* $x_2 \in \mathcal{A}_2, y_2 \in \mathcal{R}_2^-(x_2)$*, for all* $x \in \beta^-(x_2)$ *there is* $y \in \mathcal{R}^-(x) \cap \beta^-(y_2)$

Notice that the definition asks for mutual simulation of *incoming* attacks. For $\epsilon \in \{a, c, p, s, ss\}$, it is not hard to see that existence of bisimulations are neither necessary nor sufficient for existence of ϵ-relations. The problem is that a bisimulation does not ensure that attacks are absent when they need to be in order to ensure conflict-freeness. It is easy to see, for instance, that an even cycle is bisimilar to a single self-attacking argument.

However, as we show in the Appendix, bisimulations behave well with respect to defense, and they only fail to preserve conflict-freeness in specific circumstances. To illustrate, assume that you have two arguments a, b in some framework F such that a and b are not in any conflict, and that you then relate them by a bisimulation β to some a_2, b_2 in F_2 with $b_2 \in \mathcal{R}^-(a_2)$. It then follows by β being a bisimulation ("back"), that there must be some $c \in \beta^-(b_2)$ such that $c \in \mathcal{R}^-(a)$. So an attacker of a, namely c, was merged with a non-attacker of a, namely b. Such a collapse *has to* occur when bisimulations fail to preserve and reflect extensions. Let us attempt to limit it, by offering the following definition.

Definition 4.4. *Given two frameworks* F *and* F_2*, a bisimulation* $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ *is* finitely collapsing *(called an fc-bisimulation) if the following holds:*

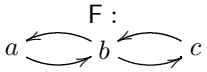
global forth: *For every backwards infinite walk* $\lambda = x_1 x_2 x_3 \ldots$ *in* F_2*, there exists some* $i \in \mathbb{N}$ *such that* $|\beta^-(x_i)| = 1$
global back: *For every backwards infinite walk* $\lambda = x_1 x_2 x_3 \ldots$ *in* F*, there exists some* $i \in \mathbb{N}$ *such that* $|\beta(x_i)| = 1$

[6] We should mention that bisimulations also feature in the work of Grossi [16,15], but there with respect to the authors interpretation of argumentation frameworks as Kripke models, and hence in the standard modal logical sense.

The main result, established in the Appendix (Theorem 5.3), is that fc-bisimulations are ϵ-relations with respect to all semantics we consider. From this, we obtain immediately the following corollary regarding behavioral equivalence.

Theorem 4.5. *Given a framework* F *and a reflexive relation* $\alpha \subseteq \mathcal{A} \times \mathcal{A}$*. If* α *is a bisimulation and every backwards infinite path contains at least one argument which is only related to itself, then all arguments related by* α *are behaviorally equivalent under all* $\epsilon \in \{p, ss, s\}$*.*

As an example of an instance of equivalence witnessed by this result, consider the following framework.



Here, we have that a and c are behaviorally equivalent, and this is witnessed by the reflexive bisimulation which relates a and c and maps b only to itself.

5 Conclusion

We presented a simple representation of argumentation semantics in terms of Łukasiewicz logic, allowing us to express complex claims concerning arguments, their semantic status, and their interactions. We showed its merit by studying equivalence, arriving at notions that are all *logically derived* from existing semantic notions. We argued for the importance of the notion of behavioral equivalence between arguments, and we lifted it to the level of frameworks, taking quotients, and without making any new claims about the nature of argumentation, beyond those already present in the semantics we considered. We followed up on this by addressing the problem of finding nice structural characterizations of behavioral equivalence, providing a result linking the logically defined notion with the concept of an arbitrary relation between frameworks that preserve and reflect solutions. Moreover, we provided a sufficient condition for behavioral equivalence in terms of bisimulations.

We believe our work shows both the appropriateness and usefulness of the simple logical representation in Łukasiewicz logic, and, moreover, that it suggests the merit of investigating further the notions of equivalence that this representation allows us to recognize.

Appendix

First, we show that bisimulations behave nicely when it comes to defense.

Fact 5.1. *Consider arbitrary frameworks* F, F_2 *and some bisimulation* $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$*. Then we have*

(1) For all $A \subseteq \mathcal{A}$*,* $\beta(\mathcal{D}(A)) = \mathcal{D}(\beta(A))$ *– β preserves defended arguments and*

(2) For all $A_2 \subseteq \mathcal{A}_2$, $\beta^-(\mathcal{D}(A_2)) = \mathcal{D}(\beta^-(A_2))$ – β reflects defended arguments.

Proof. (1) Given $A \in \mathcal{A}$, we show both inclusions. ($\subseteq$) Consider arbitrary $y \in \mathcal{D}(A)$, $y_2 \in \beta(y)$ and $z_2 \in \mathcal{R}_2^-(y_2)$. Then by β being a bisimulation ("back"), it follows that there is $z \in \beta^-(z_2)$ such that $z \in \mathcal{R}^-(y)$. Since $y \in \mathcal{D}(A)$ it follows that there is $x \in A$ such that $x \in \mathcal{R}^-(z)$. Then by β being a bisimulation ("forth") it follows that there is $x_2 \in \beta(x)$ such that $x_2 \in \mathcal{R}_2^-(z_2)$, meaning $z_2 \in \mathcal{R}_2^-(\beta(A))$. We conclude $y_2 \in \mathcal{D}(\beta(A))$ as desired. ($\supseteq$) Consider arbitrary $y_2 \in \mathcal{D}(\beta(A))$, $y \in \beta^-(y_2)$ and $z \in \mathcal{R}^-(y)$. Then by β being a bisimulation ("forth"), it follows that there is $z_2 \in \beta(z)$ such that $z_2 \in \mathcal{R}^-(y_2)$. Since $y_2 \in \mathcal{D}(\beta(A))$, it follows that there is $x_2 \in \beta(A)$ such that $x_2 \in \mathcal{R}^-(z_2)$. From β being a bisimulation ("back"), it follows that there is $x \in \beta^-(x_2)$ such that $x \in \mathcal{R}^-(z)$. It follows that $y \in \mathcal{D}(A)$, meaning that $y_2 \in \beta(\mathcal{D}(A))$ as desired.
(2) The argument is symmetric to that used to show (1). □

The next result concerns the relationship between various semantics.

Theorem 5.2. *Given frameworks F and F_2, if $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ is a bisimulation, then if β preserves and reflects admissible sets, it also preserves and reflects preferred, semi-stable and stable sets.*

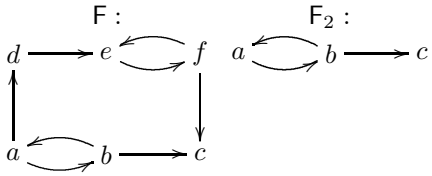
Proof. For all semantics, we only show preservation. Reflection can be shown symmetrically. **Stable**: Assume that $S \subseteq \mathcal{A}$ is stable. We know $\beta(S)$ is conflict-free and must show $\mathcal{A}_2 \setminus \beta(S) = \mathcal{R}_2^+(\beta(S))$. Consider arbitrary $x_2 \in \mathcal{A}_2 \setminus \beta(S)$. Then $\beta^-(x_2) \subseteq \mathcal{A} \setminus S$, so there is $y \in S$ such that $y \in \mathcal{R}^-(\beta^-(x_2))$. By β being a bisimulation ("forth"), we have $x_2 \in \mathcal{R}_2^+(\beta(S))$ as desired. **Preferred**: Assume that $S \subseteq \mathcal{A}$ is preferred. Then $\beta(S)$ is admissible. Assume towards contradiction that there is $A_2 \supset \beta(S)$ which is admissible in F_2. Then $\beta^-(A_2)$ is admissible in F and since $\beta(\beta^-(A_2)) \supseteq A_2 \supset \beta(S)$, we have $\beta^-(A_2) \supset S$, contradiction. **Semi-stable**: Assume that $S \subseteq \mathcal{A}$ is semi-stable, i.e. that S is admissible, and that there is no admissible $A \subseteq \mathcal{A}$ such that $S \cup \mathcal{R}^+(S) \subset A \cup \mathcal{R}^+(A)$. Assume towards contradiction that $\beta(S)$ is not semi-stable. Then there is $S_2 \subseteq \mathcal{A}_2$ such that a) $S_2 \cup \mathcal{R}_2^+(S_2) \supset \beta(S) \cup \mathcal{R}_2^+(\beta(S))$. By β being a bisimulation ("forth"), we have b) $\beta(\mathcal{R}^+(S)) \subseteq \mathcal{R}_2^+(\beta(S))$ and also ("back") that c) $\beta^-(\mathcal{R}_2^+(S_2)) \subseteq \mathcal{R}^+(\beta^-(S_2))$. We will show that $\beta^-(S_2 \cup \mathcal{R}_2^+(S_2)) = \beta^-(S_2) \cup \beta^-(\mathcal{R}_2^+(S_2)) \supset S \cup \mathcal{R}^+(S)$, which is a contradiction since it allows us to conclude, by applying c), that $\beta^-(S_2) \cup \mathcal{R}^+(\beta^-(S_2)) \supset S \cup \mathcal{R}^+(S)$. We show inclusion first.

$$\begin{aligned}
\beta^-(S_2 \cup \mathcal{R}_2^+(S_2)) &\overbrace{\supseteq}^{a)} \beta^-(\beta(S) \cup \mathcal{R}_2^+(\beta(S))) \\
&= \beta^-(\beta(S)) \cup \beta^-(\mathcal{R}_2^+(\beta(S))) \\
&\overbrace{\supseteq}^{b)} \beta^-(\beta(S)) \cup \beta^-(\beta(\mathcal{R}^+(S))) \\
&\supseteq S \cup \mathcal{R}^+(S)
\end{aligned}$$

To show that the inclusion is strict, consider $x_2 \in (S_2 \cup \mathcal{R}_2^+(S_2)) \setminus (\beta(S) \cup \mathcal{R}_2^+(\beta(S))$. For arbitrary $x \in \beta^-(x_2)$, observe first that since $x_2 \notin \beta(S)$, we have

$x \notin S$. We also have $x_2 \notin \mathcal{R}_2^+(\beta(S))$ and from b) it follows that $x_2 \notin \beta(\mathcal{R}^+(S))$. Then we conclude that $x \notin \mathcal{R}^+(S)$. □

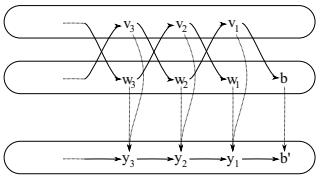
Interestingly, a bisimulation that preserves and reflects admissible sets might not preserve complete sets, as shown by the frameworks F and F_2 below. Here, we have the bisimulation $\beta = \{(a,a),(e,a),(b,b),(d,b),(f,b),(c,c)\}$ which also preserves and reflects with respect to the admissible semantics. We notice, however, that $\{a\}$ is a complete set in F while $\beta(a) = \{a\}$ is not complete in F_2 since d is defended by $\{a\}$.



The main theorem now follows, stating that fc-bisimulations are ϵ-relations with respect to all $\epsilon \in \mathcal{S}$. We remark that it is sufficient to show that fc-bisimulations preserve and reflect admissible and complete sets, from which it follows by Theorem 5.2 that they also preserve and reflect preferred, stable and semi-stable sets.

Theorem 5.3. *Given frameworks* F *and* F_2*, if there is an fc-bisimulation* $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$*, then* $\mathsf{F} \equiv^{\epsilon} \mathsf{F}_2$ *for all* $\epsilon \in \{s, a, p, ss, c\}$

Proof. **Admissible**: Let $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ be an arbitrary fc-bisimulation. We show that β preserves admissible sets. Then, by symmetry, β also reflects them, since the inverse of β, $\beta^- \subseteq \mathcal{A}_2 \times \mathcal{A}$ is clearly also an fc-bisimulation. Let $E \subseteq \mathcal{A}$ be an admissible set in F and consider $E_2 = \beta(E)$. If $x_2 \in \mathcal{R}_2^-(y_2)$ for $y_2 \in E_2$, then there is $y \in E$ such that $y_2 \in \beta(y)$, and by β being a bisimulation ("back"), there is some $x \in \mathcal{R}^-(y)$ such that $x_2 \in \beta(x)$. Since E defends itself, it follows that there is $z \in \mathcal{R}^-(x) \cap E$. Then, by β being a bisimulation ("forth"), it follows that there is some $z_2 \in \mathcal{R}_2^-(x_2)$ such that $z_2 \in \beta(z)$, meaning $z_2 \in E_2$. This shows that $E_2 \subseteq \mathcal{D}(E_2)$. To show that E_2 is conflict free, assume towards contradiction that there is $x_2, b' \in E_2$ with $x_2 \in \mathcal{R}_2^-(b')$. Then, by definition of E_2, there is $x, b \in E$ with $x_2 \in \beta(x)$ and $b' \in \beta(b)$. Also, we know that $x \notin \mathcal{R}^-(b)$ since E is conflict-free. But by β being a bisimulation ("back"), there must be $z \in \mathcal{R}^-(b)$ such that $x_2 \in \beta(z)$. Since E is conflict-free, we know that $z \in \mathcal{R}^-(E) \subseteq \mathcal{A} \setminus E$ Now we have $x_2 \in E_2 \cap \beta(x) \cap \beta(z)$ such that z attacks E, and this is the first step towards showing that there exists an infinite backwards walk $\lambda = y_1 y_2 y_3 \ldots$ in $\mathcal{A}_2$ such that for all $i \geq 1$, we have $|\beta^-(y_i)| \geq 2$. This will contradict the assumption that β is an fc-bisimulation ("global forth"). We take $y_1 = x_2$ and let $w_1 = x$, $v_1 = z$. Then for all $i \geq 2$, we define y_i, w_i, v_i inductively, assuming that $y_{i-1}, w_{i-1}, v_{i-1}$ have been defined such that $w_{i-1} \in E, v_{i-1} \in \mathcal{R}^-(E) \subseteq \mathcal{A} \setminus E$ and $y_{i-1} \in \beta(w_{i-1}) \cap \beta(v_{i-1})$. The construction is visualized below.



Since E defends itself against all attacks, we can find $w_i \in E \cap \mathcal{R}^-(v_{i-1})$. Since we have $y_{i-1} \in \beta(v_{i-1})$ it follows by β being a bisimulation ("forth") that we can find $y_i \in \beta(w_i) \cap \mathcal{R}^-(y_{i-1})$. But we also have $y_{i-1} \in \beta(w_{i-1})$, so by β being a bisimulation ("back"), we find $v_i \in \beta^-(y_i) \cap \mathcal{R}^-(w_{i-1})$. Since $w_{i-1} \in E$ and E is conflict-free, it follows that $v_i \in \mathcal{R}^-(E) \subseteq \mathcal{A} \setminus E$. So y_i, w_i, v_i can be found for all $i \in \mathbb{N}$, proving existence of λ that contradicts "global forth".
Complete: We know that β preserves and reflects admissible sets, and now we assume that $S \subseteq \mathcal{A}$ is complete. Consider arbitrary $x_2 \in \mathcal{A}_2 \setminus (\beta(S) \cup \mathcal{R}_2^+(\beta(S)))$. By β being a bisimulation ("forth"), we get $\beta^-(x_2) \cap \mathcal{R}^+(S) = \emptyset$, which implies $\beta^-(x_2) \subseteq \mathcal{A} \setminus (S \cup \mathcal{R}^+(S))$. Then, since S is complete, there is $y \in \mathcal{A} \setminus (S \cup \mathcal{R}^+(S))$ such that $y \in \mathcal{R}^-(\beta^-(x_2))$. Then, since β is a bisimulation ("forth"), it follows that there is $y_2 \in \beta(y) \cap \mathcal{R}_2^-(x_2)$. Since $x_2 \notin \mathcal{R}_2^+(\beta(S))$ it follows that $y_2 \notin \beta(S)$. Assume towards contradiction that $y_2 \in \mathcal{R}_2^+(z_2)$ for some $z_2 \in \beta(S)$. Then there is $z \in S \cap \beta^-(z_2)$ and also, since β is a bisimulation ("back"), there is $z' \in \mathcal{R}^-(y) \cap \beta^-(z_2)$. Since $y \notin \mathcal{R}^+(S)$, $z' \notin S$. Since β is a bisimulation ("forth") and $z_2 \in \beta(S)$ and $\beta(S)$ is conflict-free, $z' \notin \mathcal{R}^+(S)$. It follows that $z' \in \mathcal{A} \setminus (S \cup \mathcal{R}^+(S))$. To contradict global forth, we prove existence of a backwards infinite walk $\lambda = x_1 x_2 x_3 \ldots$ in F_2 such that for all $i \geq 1$ we have $|\beta^-(x_i)| \geq 2$. We take $x_1 = z_2$, $v_1 = z', w_1 = z$ and for all $i \geq 2$, we assume that we have $x_{i-1}, v_{i-1}, w_{i-1}$ with $x_{i-1} \in \beta(S) \cup \mathcal{R}_2^+(\beta(S))$ and $w_{i-1} \in (S \cup \mathcal{R}^+(S)) \cap \beta^-(x_{i-1})$, $v_{i-1} \in (\mathcal{A} \setminus (S \cup \mathcal{R}^+(S))) \cap \beta^-(x_{i-1})$. There are two cases. I) $x_{i-1} \in \beta(S)$. Then since $\beta(S)$ is admissible and $w_{i-1} \in \beta^-(x_{i-1})$, we have $w_{i-1} \notin \mathcal{R}^+(S)$ by β being a bisimulation ("forth"). Since S is complete, we find $v_i \in \mathcal{R}^-(v_{i-1}) \cap (\mathcal{A} \setminus (S \cup \mathcal{R}^+(S)))$. Since β is a bisimulation ("forth"), we find $x_i \in \mathcal{R}_2^-(x_{i-1}) \cap \beta(v_i)$, and since $\beta(S)$ is admissible, $x_i \in \mathcal{R}_2^+(\beta(S))$. Then, going back, we find $w_i \in \beta^-(x_i) \cap \mathcal{R}^-(w_{i-1})$, and since $w_{i-1} \in S$ and S is admissible, $w_i \in \mathcal{R}^+(S)$. II) $x_{i-1} \in \mathcal{R}_2^+(\beta(S))$. Since $w_{i-1} \in \beta^-(x_{i-1}) \cap (S \cup \mathcal{R}^+(S))$ and $\beta(S)$ is admissible, we have $w_{i-1} \in \mathcal{R}^+(S)$. We choose $w_i \in S \cap \mathcal{R}^-(w_{i-1})$. By β being a bisimulation ("forth"), we find $x_i \in \beta(w_i) \cap \mathcal{R}_2^-(x_{i-1})$ and ("back") $v_i \in \beta^-(x_i) \cap \mathcal{R}^-(v_{i-1})$. Since $v_{i-1} \notin \mathcal{R}^+(S)$, $v_i \notin S$. Also, by β being a bisimulation ("forth") and $x_i \in \beta(v_i) \cap \beta(S)$ and $\beta(S)$ being conflict-free, we have $v_i \notin \mathcal{R}^+(S)$. Having established the claim for $\epsilon \in \{a, c\}$, the claim follows by Theorem 5.2 for all $\epsilon \in \{a, c, p, ss, s\}$. □

References

1. Arieli, O., Caminada, M.W.A.: A QBF-based formalization of abstract argumentation semantics. Journal of Applied Logic 11(2), 229–252 (2013)
2. Baroni, P., Giacomin, M.: Solving semantic problems with odd-length cycles in argumentation. In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 440–451. Springer, Heidelberg (2003)
3. Baumann, R.: Normal and strong expansion equivalence for argumentation frameworks. Artif. Intell. 193, 18–44 (2012)
4. Baumann, R.: What does it take to enforce an argument? Minimal change in abstract argumentation. In: De Raedt, L., Bessière, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F., Lucas, P.J.F. (eds.) ECAI. Frontiers in Artificial Intelligence and Applications, vol. 242, pp. 127–132. IOS Press (2012)
5. Béziau, J.-Y.: A sequent calculus for Łukasiewicz's three-valued logic based on Suszko's bivalent semantics. Bulletin of the Section of Logic 28(2), 89–97 (1998)
6. Caminada, M.W.A.: Comparing two unique extension semantics for formal argumentation: Ideal and eager. In: BNAIC 2007, pp. 81–87 (2007)
7. Caminada, M.W.A., Amgoud, L.: On the evaluation of argumentation formalisms. Artificial Intelligence 171(56), 286–310 (2007)
8. Caminada, M.W.A., Gabbay, D.M.: A logical account of formal argumentation. Studia Logica 93(2-3), 109–145 (2009)
9. Caminada, M.W.A.: Semi-stable semantics. In: Proceedings of the 2006 Conference on Computational Models of Argument: Proceedings of COMMA 2006, pp. 121–130. IOS Press, Amsterdam (2006)
10. Bench Capon, T.J.M., Dunne, P.E.: Argumentation in artificial intelligence. Artif. Intell. 171(10-15), 619–641 (2007)
11. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77, 321–357 (1995)
12. Dung, P.M., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. Artificial Intelligence 171(1015), 642–674 (2007)
13. Dyrkolbotn, S.: Doing argumentation using theories in graph normal form. In: Rendsvig, R.K. (ed.) ESSLLI 2012 Student Session Proceedings (2012)
14. Dyrkolbotn, S., Walicki, M.: Propositional discourse logic. Synthese (to appear)
15. Grossi, D.: Argumentation in the view of modal logic. In: McBurney, P., Rahwan, I., Parsons, S. (eds.) ArgMAS 2010. LNCS, vol. 6614, pp. 190–208. Springer, Heidelberg (2011)
16. Grossi, D.: On the logic of argumentation theory. In: van der Hoek, W., Kaminka, G.A., Lespérance, Y., Luck, M., Sen, S. (eds.) AAMAS, pp. 409–416. IFAAMAS (2010)
17. Grossi, D., Gabbay, D.: When are two arguments the same? Invariance in abstract argumentation. Technical Report 4, University of Liverpool (2012)
18. Oikarinen, E., Woltran, S.: Characterizing strong equivalence for argumentation frameworks. Artificial Intelligence 175(14-15), 1985–2009 (2011)
19. Verheij, B.: Two approaches to dialectical argumentation: Admissible sets and argumentation stages. In: Proceedings of the Biannual International Conference on Formal and Applied Practical Reasoning (FAPR) Workshop, pp. 357–368, Universiteit (1996)
20. Wu, Y., Caminada, M.W.A., Gabbay, D.M.: Complete extensions in argumentation coincide with 3-valued stable models in logic programming. Studia Logica 93(2-3), 383–403 (2009)

Boolean Dependence Logic and Partially-Ordered Connectives

Johannes Ebbing[1], Lauri Hella[2], Peter Lohmann[1], and Jonni Virtema[2]

[1] Leibniz University Hannover, Theoretical Computer Science, Germany
{ebbing,lohmann}@thi.uni-hannover.de
[2] University of Tampere, Mathematics, School of Information Sciences, Finland
{lauri.hella,jonni.virtema}@uta.fi

Abstract. We introduce a new variant of dependence logic ($\mathcal{D}$) called Boolean dependence logic ($\mathcal{BD}$). In $\mathcal{BD}$ dependence atoms are of the type $=(x_1, \ldots, x_n, \alpha)$, where α is a Boolean variable. Intuitively, with Boolean dependence atoms one can express quantification of relations, while standard dependence atoms express quantification over functions.

We compare the expressive power of $\mathcal{BD}$ to $\mathcal{D}$ and first-order logic enriched by partially-ordered connectives, $\mathcal{FO}(\mathcal{POC})$. We show that the expressive power of $\mathcal{BD}$ and $\mathcal{D}$ coincide. We define natural syntactic fragments of $\mathcal{BD}$ and show that they coincide with the corresponding fragments of $\mathcal{FO}(\mathcal{POC})$ with respect to expressive power. We then show that the fragments form a strict hierarchy.

Keywords: Dependence Logic, Partially-Ordered Connectives, Expressivity, Existential Second-Order Logic.

1 Introduction

Dependence is an important concept in many scientific disciplines. A multitude of logical formalims have been designed to model dependences, for example, in database theory, social choice theory and quantum mechanics.

Dependences between variables in formulae is the most direct way to model dependences in logical systems. In first-order logic the order in which quantifiers are written determines dependence relations between variables. For example, when using game theoretic semantics to evaluate the formula

$$\forall x_0 \exists x_1 \forall x_2 \exists x_3 \, \varphi,$$

the choice for x_1 depends on the value for x_0, and the choice for x_3 depends on the value of both universally quantified variables x_0 and x_2. The first to consider more complex dependences between variables was Henkin [8] with his partially-ordered quantifiers (1).

$$\begin{pmatrix} \forall x_0 & \exists x_1 \\ \forall x_2 & \exists x_3 \end{pmatrix} \varphi \quad (1) \qquad \begin{pmatrix} \forall \boldsymbol{x} & \exists \alpha \\ \forall \boldsymbol{y} & \exists \beta \end{pmatrix} \varphi \quad (2) \qquad \begin{pmatrix} \forall \boldsymbol{x} & \bigvee_{b_1 \in \{0,1\}} \\ \forall \boldsymbol{y} & \bigvee_{b_2 \in \{0,1\}} \end{pmatrix} \gamma \quad (3)$$

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 111–125, 2013.

In (1) x_1 depends only on x_0 and x_3 depends only on x_2. Enderton [6] and Walkoe [17] observed that exactly the properties definable in existential second-order logic ($\mathcal{ESO}$) can be expressed with partially-ordered quantifiers. Building on the ideas of Henkin, Blass and Gurevich introduced in [3] the narrow Henkin quantifier (2). In (2) the value of the Boolean variable α_1 depends on value of the first-order and Boolean variables in the tuple $\boldsymbol{x}$, while the value of α_2 depends on the value of the tuple $\boldsymbol{y}$. The idea of Blass and Gurevich was further developed by Sandu and Väänänen in [13] where they introduced partially-ordered connectives (3); here γ is a tuple $(\gamma_{00}, \gamma_{01}, \gamma_{10}, \gamma_{11})$ of formulae. In this paper we adopt the approach based on Boolean variables from [3] for our definition of partially-ordered connectives. Hence in this paper, a partially ordered connective is as in (2) with the exception that $\boldsymbol{x}$ and $\boldsymbol{y}$ are not allowed to contain Boolean variables. However the expressive power of partially-ordered connectives defined here and in [13] coincide. For recent work on partially-ordered connectives see e.g. [7,15].

The first to linearize the idea behind the syntax of partially-ordered quantifiers were Hintikka and Sandu [9,10], who introduced independence-friendly logic ($\mathcal{IF}$). $\mathcal{IF}$-logic extends $\mathcal{FO}$ in terms of so-called slashed quantifiers. Dependence logic ($\mathcal{D}$), introduced by Väänänen [16], was inspired by $\mathcal{IF}$-logic, but the approach of Väänänen provided a fresh perspective on quantifier dependence. In dependence logic the dependence relations between variables are written in terms of novel atomic dependence formulae. For example, the partially-ordered quantifier (1) can be expressed in dependence logic as follows

$$\forall x_0 \exists x_1 \forall x_2 \exists x_3 (=(x_2, x_3) \wedge \varphi).$$

The atomic formula $=(x_2, x_3)$ has the explicit meaning that x_3 is completely determined by x_2 and nothing else.

In recent years, research related to dependence logic has been intense. A variety of closely related logics have been defined and various applications suggested, see e.g. [1,4,5,11,12,14].

In this paper we introduce a new variant of dependence logic called Boolean dependence logic ($\mathcal{BD}$). In $\mathcal{BD}$ dependence atoms are of the type $=(x_1, \ldots, x_n, \alpha)$, where α is a Boolean variable. Boolean dependence atoms provide a direct way to express partially-ordered connectives in the same way as dependence atoms express partially-ordered quantifiers. The partially-ordered connective (2) can be expressed in Boolean dependence logic as follows

$$\forall \boldsymbol{x} \exists \alpha \forall \boldsymbol{y} \exists \beta (=(\boldsymbol{y}, \beta) \wedge \varphi).$$

Intuitively, with Boolean dependence atoms one can express quantification of relations, while standard dependence atoms $=(x_1, \ldots, x_n, y)$ express quantification over functions. Since in second-order logic it is clear that functions and relations are interdefinable, the question arises whether there is any significant difference between $\mathcal{D}$ and $\mathcal{BD}$. We show that in terms of expressive power there is no difference. On the other hand, in $\mathcal{BD}$ one can define directly natural fragments that correspond to logics with partially-ordered connectives.

Our results can be seen as a contribution to the analysis of fragments of $\mathcal{ESO}$. In particular we are able to separate natural fragments of $\mathcal{ESO}$. We also give new

insight concerning interdefinability of functions and relations in the framework of dependence logic.

The structure of the paper is as follows. In Section 2 we define Boolean dependence logic, $\mathcal{BD}$, and its fragments $\mathcal{BBD}$, $\mathcal{RBD}$ and $\forall$-$\mathcal{BD}$. In addition we define the extension of first-order logic with all partially-ordered connectives, $\mathcal{FO}(\mathcal{POC})$, and its fragments $\mathcal{FO}(\mathcal{POC}^+)$, $\mathcal{POC}[\mathcal{FO}]$ and $\mathcal{POC}[\mathcal{QF}]$. In Section 3 we prove a normal form for $\mathcal{BBD}$ and using this normal form show that the expressive power of $\mathcal{BBD}$, $\mathcal{RBD}$ and $\forall$-$\mathcal{BD}$ coincide with $\mathcal{FO}(\mathcal{POC}^+)$, $\mathcal{POC}[\mathcal{FO}]$ and $\mathcal{POC}[\mathcal{QF}]$, respectively. In Section 4 we show that $\mathcal{BD}$, $\mathcal{BBD}$, $\mathcal{RBD}$ and $\forall$-$\mathcal{BD}$ form a strict hierarchy with respect to expressive power.

2 Preliminaries

2.1 Boolean Dependence Logic

Boolean dependence logic $\mathcal{BD}$ is a variant of dependence logic where the consequents of dependence atoms are Boolean variables instead of first-order variables. We use $\alpha, \beta, \ldots$ to denote Boolean variables and $x, y, \ldots$ to denote first-order variables. Tuples of variables are denoted by $\boldsymbol{x}, \boldsymbol{y}$ and $\boldsymbol{\alpha}, \boldsymbol{\beta}$, respectively.

Definition 1. *Let τ be a relational vocabulary. The logics $\mathcal{D}(\tau)$ and $\mathcal{BD}(\tau)$ are defined by the following grammars. The rules common to both $\mathcal{D}$ and $\mathcal{BD}$ are*

$$\varphi ::= x_1 = x_2 \mid \neg\, x_1 = x_2 \mid R(\boldsymbol{x}) \mid \neg R(\boldsymbol{x}) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid \forall x\varphi \mid \exists x\varphi.$$

The additional grammar rule for $\mathcal{D}$ is $\varphi ::= =(x_1, \ldots, x_n)$, and the additional grammar rules for $\mathcal{BD}$ are $\varphi ::= \alpha \mid \neg\alpha \mid =(x_1, \ldots, x_n, \alpha) \mid \exists\alpha\varphi$.

In order to define the semantics of $\mathcal{D}$ and $\mathcal{BD}$ we first need to define the concept of a *team*. Let $\mathfrak{A}$ be a model with the domain A. *Assignments* over $\mathfrak{A}$ are finite functions that map first-order variables to elements of A and Boolean variables to elements of $\{\bot, \top\}$. If s is an assignment, χ a Boolean or first-order variable, and $a \in \{\bot, \top\}$ or $a \in A$, respectively, then $s(a/\chi)$ denotes the assignment (with domain $\mathrm{dom}(s) \cup \{\chi\}$) which agrees with s everywhere except that $s(a/\chi)(\chi) = a$.

Let A be a set and $\{x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_m\}$ a finite (possibly empty) set of variables. A *team* X of A with the domain $\mathrm{dom}(X) = \{x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_m\}$ is any set of assignments from the variables $\{x_1, \ldots, x_k, \alpha_1, \ldots, \alpha_m\}$ into the set $A \cup \{\bot, \top\}$. If X is a team of A, $F\colon X \to A$ and $G\colon X \to \{\bot, \top\}$, we use $X(F/x)$ to denote the team $\{s(F(s)/x) \mid s \in X\}$, $X(G/\alpha)$ to denote the team $\{s(G(s)/\alpha) \mid s \in X\}$ and $X(A/x)$ for the team $\{s(a/x) \mid s \in X \text{ and } a \in A\}$. For a set $W \subseteq \mathrm{dom}(X)$ we call the function F *W-determined* iff for all $s, s' \in X$, with $s(\chi) = s'(\chi)$ for all $\chi \in W$, we have that $F(s) = F(s')$.

Definition 2. *Let $\mathfrak{A}$ be a model and X a team. The satisfaction relation $\mathfrak{A} \models_X \varphi$ is defined as follows.*

$$\begin{aligned}
\mathfrak{A} \models_X \ell \quad &\Leftrightarrow \quad \forall s \in X : \mathfrak{A}, s \models \ell, \text{ when } \ell \text{ is a first-order literal} \\
\mathfrak{A} \models_X \alpha \quad &\Leftrightarrow \quad \forall s \in X : s(\alpha) = \top \\
\mathfrak{A} \models_X \neg\alpha \quad &\Leftrightarrow \quad \forall s \in X : s(\alpha) = \bot \\
\mathfrak{A} \models_X \varphi \wedge \psi \quad &\Leftrightarrow \quad \mathfrak{A} \models_X \varphi \text{ and } \mathfrak{A} \models_X \psi
\end{aligned}$$

$$\begin{array}{rcl}
\mathfrak{A} \models_X \varphi \vee \psi & \Leftrightarrow & \mathfrak{A} \models_Y \varphi \text{ and } \mathfrak{A} \models_Z \psi \text{ for some } Y \cup Z = X \\
\mathfrak{A} \models_X =(\boldsymbol{x}, \alpha) & \Leftrightarrow & \forall s, s' \in X : s(\boldsymbol{x}) = s'(\boldsymbol{x}) \text{ implies } s(\alpha) = s'(\alpha) \\
\mathfrak{A} \models_X =(\boldsymbol{x}, y) & \Leftrightarrow & \forall s, s' \in X : s(\boldsymbol{x}) = s'(\boldsymbol{x}) \text{ implies } s(y) = s'(y) \\
\mathfrak{A} \models_X \exists x \psi & \Leftrightarrow & \mathfrak{A} \models_{X(F/x)} \psi \text{ for some } F \colon X \to A \\
\mathfrak{A} \models_X \exists \alpha \psi & \Leftrightarrow & \mathfrak{A} \models_{X(F/\alpha)} \psi \text{ for some } F \colon X \to \{\bot, \top\} \\
\mathfrak{A} \models_X \forall x \psi & \Leftrightarrow & \mathfrak{A} \models_{X(A/x)} \psi
\end{array}$$

Definition 3. *Let $\varphi \in \mathcal{BD}$. The set $\mathrm{Fr}(\varphi)$ of free variables of a formula φ is defined as for $\mathcal{FO}$, except that we have the new cases*

$$\mathrm{Fr}(\alpha) = \mathrm{Fr}(\neg\alpha) = \{\alpha\} \quad \mathrm{Fr}(=(\boldsymbol{x}, \alpha)) = \{\boldsymbol{x}, \alpha\} \quad \mathrm{Fr}(\exists\alpha\varphi) = \mathrm{Fr}(\varphi)\backslash\{\alpha\}$$

If $\mathrm{Fr}(\varphi) = \varnothing$, we call φ a sentence. We say that the sentence φ is true in the model $\mathfrak{A}$ and write $\mathfrak{A} \models \varphi$, if $\mathfrak{A} \models_{\{\varnothing\}} \varphi$ holds.

Definition 4. *Let $V = \{x_{i_1}, \ldots, x_{i_n}\}$ where $i_j \leqslant i_{j+1}$ for all $j < n$. By $=(V, \alpha)$ and $=(V, y)$ we denote $=(x_{i_1}, \ldots, x_{i_n}, \alpha)$ and $=(x_{i_1}, \ldots, x_{i_n}, y)$, respectively.*

2.2 Partially-Ordered Connectives

According to the definition of Sandu and Väänänen [13], a partially-ordered connective is an expression of the form

$$D = \begin{pmatrix} \forall x_{11} & \ldots & \forall x_{1n} & \bigvee_{b_1 \in \{0,1\}} \\ \vdots & & \vdots & \vdots \\ \forall x_{m1} & \ldots & \forall x_{mn} & \bigvee_{b_m \in \{0,1\}} \end{pmatrix}$$

that binds a tuple $\gamma = (\varphi_{\boldsymbol{b}})_{\boldsymbol{b} \in \{0,1\}^m}$ of formulae. Here it is assumed that all the variables x_{ij} are distinct. We denote the set of all such partially ordered connectives D by D. By $\mathcal{FO}(\mathsf{D})$ we denote the extension of first-order logic by all partially-ordered connectives $D \in \mathsf{D}$. Furthermore, by $\mathsf{D}[\mathcal{FO}]$ and $\mathsf{D}[\mathcal{QF}]$ we denote the logics consisting of formulae of the form $D\,(\varphi_{\boldsymbol{b}})_{\boldsymbol{b} \in \{0,1\}^m}$, where $D \in \mathsf{D}$ and $\varphi_{\boldsymbol{b}}$ are first-order formulae or quantifier free formulae, respectively.

We will deviate from the definitions of [13] in two ways: First, we will replace the disjunctions $\bigvee_{b_i \in \{0,1\}}$ by existentially quantified Boolean variables $\exists\alpha_i$; this makes it easier to relate logics with partially-ordered connectives to fragments of $\mathcal{BD}$. Second, in order to simplify the proofs in Section 3, we will relax the restriction that the variables x_{ij} should be distinct, and that the tuples $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{in})$ should be of the same length.

Definition 5. *Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{in_i})$, $1 \leqslant i \leqslant m$, be tuples of first-order variables, and let α_i, $1 \leqslant i \leqslant m$, be distinct Boolean variables. Then*

$$C = \begin{pmatrix} \forall \boldsymbol{x}_1 & \exists \alpha_1 \\ \vdots & \vdots \\ \forall \boldsymbol{x}_m & \exists \alpha_m \end{pmatrix}$$

is a partially-ordered connective. *The* pattern *of C is $\pi = (n_1, \ldots, n_m, E)$, where E describes the identities between the variables in the tuples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$: $E = \{(i, j, k, l) \mid 1 \leqslant i, k \leqslant m, 1 \leqslant j \leqslant n_i, 1 \leqslant l \leqslant n_k, x_{ij} = x_{kl}\}$. If C is a partially-ordered connective with pattern π, we denote C by $N_\pi\, \boldsymbol{x}_1 \alpha_1 \ldots \boldsymbol{x}_m \alpha_m$.*

Definition 6. *The logic $\mathcal{FO}(\mathcal{POC})$ is defined by the following grammar:*

$$\varphi ::= \alpha \mid \neg\alpha \mid x_1 = x_2 \mid \neg x_1 = x_2 \mid R\boldsymbol{x} \mid \neg R\boldsymbol{x} \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid \forall x\varphi \mid \exists x\varphi \mid N_\pi \boldsymbol{x}_1\alpha_1 \dots \boldsymbol{x}_m\alpha_m\, \varphi \mid \neg N_\pi \boldsymbol{x}_1\alpha_1 \dots \boldsymbol{x}_m\alpha_m\, \varphi.$$

$\mathcal{FO}(\mathcal{POC}^+)$ is the fragment of $\mathcal{FO}(\mathcal{POC})$ that allows only positive occurrences of partially-ordered quantifiers. In other words, $\mathcal{FO}(\mathcal{POC}^+)$ is defined by the grammar above without the last rule $\neg N_\pi \boldsymbol{x}_1\alpha_1 \dots \boldsymbol{x}_m\alpha_m\, \varphi$. The logic $\mathcal{POC}[\mathcal{FO}]$ ($\mathcal{POC}[\mathcal{QF}]$) consists of all formulae of the form $N_\pi \boldsymbol{x}_1\alpha_1 \dots \boldsymbol{x}_m\alpha_m\, \varphi$, where $\varphi \in \mathcal{FO}$ (φ is a quantifier free formula of $\mathcal{FO}$).

The semantics of these logics is defined in terms of models and assignments in the usual way. The clause for a formula starting with a partially-ordered connective $C = N_\pi \boldsymbol{x}_1\alpha_1 \dots \boldsymbol{x}_m\alpha_m$ with pattern $\pi = (n_1, \dots, n_m, E)$ is the following:

$$\begin{aligned} \mathfrak{A}, s \models C\,\varphi \quad \Leftrightarrow \quad & \text{there exist functions } f_i : A^{n_i} \to \{\bot, \top\}, 1 \leqslant i \leqslant m, \\ & \text{such that for all tuples } \boldsymbol{a}_i \in A^{n_i}, 1 \leqslant i \leqslant m : \\ & \text{if } \boldsymbol{a}_1 \dots \boldsymbol{a}_m \text{ is of pattern } \pi, \text{ then } \mathfrak{A}, s' \models \varphi, \text{where} \\ & s' = s(\boldsymbol{a}_1/\boldsymbol{x}_1, \dots, \boldsymbol{a}_m/\boldsymbol{x}_m, f_1(\boldsymbol{a}_1)/\alpha_1, \dots, f_m(\boldsymbol{a}_m)/\alpha_m). \end{aligned}$$

Here "$\boldsymbol{a}_1 \dots \boldsymbol{a}_m$ is of pattern π" means that $a_{ij} = a_{kl}$ whenever $(i, j, k, l) \in E$. Note that s' is well-defined for all tuples $\boldsymbol{a}_1 \dots \boldsymbol{a}_m$ that are of pattern π.

It is straightforward to prove that the partially-ordered connectives N_π defined here have the same expressive power as those defined in [13].

A logic $\mathcal{L}'$ is *at least as expressive* as $\mathcal{L}$ ($\mathcal{L} \leqslant \mathcal{L}'$) iff for all sentences $\varphi \in \mathcal{L}$ there is a sentence $\varphi' \in \mathcal{L}'$ such that for all structures $\mathfrak{A}$ it holds that $\mathfrak{A} \models \varphi$ iff $\mathfrak{A} \models \varphi'$. We say $\mathcal{L}$ and $\mathcal{L}'$ are *equivalent* ($\mathcal{L} \equiv \mathcal{L}'$) iff $\mathcal{L} \leqslant \mathcal{L}'$ and $\mathcal{L}' \leqslant \mathcal{L}$.

Proposition 1. *$\mathcal{FO}(\mathcal{POC}) \equiv \mathcal{FO}(\mathsf{D})$. Moreover, $\mathcal{POC}[\mathcal{FO}] \equiv \mathsf{D}[\mathcal{FO}]$ and $\mathcal{POC}[\mathcal{QF}] \equiv \mathsf{D}[\mathcal{QF}]$.*

2.3 Fragments of $\mathcal{BD}$

In this section we define fragments of $\mathcal{BD}$ that correspond to, with respect to expressive power, the fragments $\mathcal{FO}(\mathcal{POC}^+)$, $\mathcal{POC}[\mathcal{FO}]$ and $\mathcal{POC}[\mathcal{QF}]$ of $\mathcal{FO}(\mathcal{POC})$.

Let φ be a $\mathcal{BD}$-formula or a $\mathcal{FO}(\mathcal{POC})$-formula, and and let ψ be a subformula of φ. We define $V_\varphi(\psi)$ to be the set of all first-order variables χ such that there exists a formula ϑ such that for some quantifier Q, $Q\chi\theta$ is a subformula of φ and ψ is a subformula of ϑ. In other words, $V_\varphi(\psi)$ is the set of first-order variables χ that are quantified in φ and bound ψ.

Definition 7. *We define the following sublogics of $\mathcal{BD}$.*

1. *$\mathcal{BBD}$ is the restriction of $\mathcal{BD}$ to formulae φ such that for every subformula $\exists x\psi$ of φ and each occurrence of a dependence atom $=(x_1, \dots, x_n, \alpha)$ in ψ it holds that $V_\varphi(\psi) \subseteq \{x_1, \dots, x_n\}$.*
2. *$\mathcal{RBD}$ is the restriction of $\mathcal{BD}$ to formulae where no dependence atoms occur inside the scope of an existential first-order quantifier.*
3. *$\forall$-$\mathcal{BD}$ is the restriction of $\mathcal{BD}$ to formulae without existential quantification of first-order variables.*

It is easy to see, that every $\forall$-$\mathcal{BD}$ formula is an $\mathcal{RBD}$ formula, every $\mathcal{RBD}$ formula is a $\mathcal{BBD}$ formula and every $\mathcal{BBD}$ formula is a $\mathcal{BD}$ formula.

Proposition 2. $\forall$-$\mathcal{BD} \leqslant \mathcal{RBD} \leqslant \mathcal{BBD} \leqslant \mathcal{BD} \leqslant \mathcal{D}$

Proof. The first three inclusions follow by the observation made above. For the last inclusion we show that for every $\mathcal{BD}$-sentence φ there exists a $\mathcal{D}$-sentence φ' such that for all structures $\mathfrak{A}$

$$\mathfrak{A} \models \varphi \text{ iff } \mathfrak{A} \models \varphi'. \tag{4}$$

W.l.o.g. we can restrict our attension to models with at least two elements. Let $\varphi' := \exists x_\bot \exists x_\top (x_\bot \neq x_\top \wedge \varphi^*)$, where φ^* is the sentence obtained from φ by replacing every subformula of type $\exists \alpha_i \psi$ by $\exists x_{\alpha_i}\big((x_{\alpha_i} = x_\top \vee x_{\alpha_i} = x_\bot) \wedge \psi\big)$, every atomic not negated Boolean variable α_i by $x_{\alpha_i} = x_\top$, every atomic negated Boolean variable $\neg\alpha_i$ by $x_{\alpha_i} = x_\bot$, and every $\mathcal{BD}$ dependence atom $=(\boldsymbol{x}, \alpha_i)$ by the first-order dependence atom $=(\boldsymbol{x}, x_{\alpha_i})$. Clearly (4) holds for any model $\mathfrak{A}$ of cardinality at least two.

3 Equivalences

In this section we prove a normal form for $\mathcal{BBD}$. Using this normal form we show that $\mathcal{BBD} \equiv \mathcal{FO}(\mathcal{POC}^+)$, $\mathcal{RBD} \equiv \mathcal{POC}[\mathcal{FO}]$ and $\forall$-$\mathcal{BD} \equiv \mathcal{POC}[\mathcal{QF}]$. In addition we show that $\mathcal{BD} \equiv \mathcal{D}$.

Definition 8. *A formula $\varphi \in \mathcal{BD}$ is in* variable normal form *if no variable in* $\mathrm{Fr}(\varphi)$ *is quantified in φ, and if each variable is quantified at most once in φ.*

Definition 9. *A sentence $\varphi \in \mathcal{BBD}$ is in* Q-normal form *if φ is in variable normal form and there exists a formula $\vartheta \in \mathcal{BBD}$ such that the following holds.*

1. *$\varphi = \forall \boldsymbol{x} \exists \boldsymbol{\alpha} \vartheta$, for some (possibly empty) block of universal quantifiers $\forall \boldsymbol{x}$ followed by a (possibly empty) block of existential Boolean quantifiers $\exists \boldsymbol{\alpha}$.*
2. *Each quantifier in ϑ occurs in some block of quantifiers $\exists \boldsymbol{x} \forall \boldsymbol{y} \exists \boldsymbol{\alpha}$, where at least $\boldsymbol{x}$ is non-empty.*

In other words, a sentence φ is in Q-normal form if each quantifier in φ is pulled to the nearest existential first-order quantifier and then each universal first-order quantifier is pulled past Boolean quantifiers.

Definition 10. *A sentence $\varphi \in \mathcal{BBD}$ is in* dependence normal form *if φ is in Q-normal form, and for every maximal non-empty block of Boolean existential quantifiers $\exists \boldsymbol{\alpha}$ in φ there exists a subformula $\exists \boldsymbol{\alpha}\big(\big(\bigwedge_{i \in I} =(\boldsymbol{x_i}, \alpha_i)\big) \wedge \psi\big)$ of φ such that the Boolean variables α_i, $i \in I$, are exactly the variables quantified in $\exists \boldsymbol{\alpha}$ and every dependence atom in ψ is in scope of some quantifier in ψ.*

More informally, a sentence in Q-normal form is in dependence normal form if there is one-to-one correspondence between Boolean existential quantifiers and Boolean dependence atoms such that each quantifier $\exists \alpha$ is immediately followed by the corresponding dependence atom $=(\boldsymbol{x}, \alpha)$, and conversely each dependence atom is directly preceded by the corresponding Boolean quantifier.

Proposition 3 (Lemma A5 and Propositions A2 and A3 in the Appendix). *Every sentence in $\mathcal{BBD}$ has an equivalent sentence in dependence normal form.*

Let $\varphi \in \mathcal{BBD}$ be a formula in dependence normal form. We say that a subformula ψ of φ is *dependence maximal* (with respect to φ) if it is either a maximal subformula containing no dependence atoms, or it is of the form $\exists \boldsymbol{y} \forall \boldsymbol{x} \exists \boldsymbol{\alpha}\, \vartheta$, where $\exists \boldsymbol{y} \forall \boldsymbol{x} \exists \boldsymbol{\alpha}$ is a maximal and non-empty block of quantifiers.

Theorem 1. $\mathcal{BBD} \equiv \mathcal{FO}(\mathcal{POC}^+)$

Proof. We will first prove that $\mathcal{BBD} \leqslant \mathcal{FO}(\mathcal{POC}^+)$. Let φ be a $\mathcal{BBD}$ sentence and $\mathfrak{A}$ a model. By Proposition 3 we may assume that φ is in dependence normal form. We translate φ into an equivalent $\mathcal{FO}(\mathcal{POC}^+)$ sentence φ^* by substituting each maximal block of quantifiers along with the corresponding dependence atoms in φ by a partially-ordered connective.

More precisely, we define a translation $\psi \mapsto \psi^*$ for all subformulas ψ of φ that are dependence maximal or Boolean combinations of dependence maximal subformulas. The translation is defined recursively as follows:

(i) If ψ is a maximal subformula without dependence atoms, then let $\psi^* := \psi$.

(ii) If $\psi = \exists \boldsymbol{y} \forall \boldsymbol{x} \exists \boldsymbol{\alpha} \Big(\big(\bigwedge_{1 \leqslant i \leqslant m} {=}(V_\varphi(\psi) \cup \{\boldsymbol{y}\boldsymbol{x}_i\}, \alpha_i) \big) \wedge \vartheta \Big)$ is dependence maximal, we define

$$\psi^* := \exists \boldsymbol{y} N_\pi\, \boldsymbol{x}_0 \alpha_0 \boldsymbol{x}_1 \alpha_1 \ldots \boldsymbol{x}_m \alpha_m\, \vartheta^*, \tag{5}$$

where $\boldsymbol{x}_0$ are exactly those variables in $\boldsymbol{x}$ that are not in any $\boldsymbol{x}_i$, $1 \leqslant i \leqslant m$, and α_0 is a fresh Boolean variable not occurring in ϑ^*.

(iii) If $\psi = \vartheta \otimes \eta$ for $\otimes \in \{\wedge, \vee\}$, then $\psi^* = \vartheta^* \otimes \eta^*$.

Note that since φ is in dependence normal form, φ^* is defined, and clearly $\varphi^* \in \mathcal{FO}(\mathcal{POC}^+)$. Thus, it suffices to prove that ψ and ψ^* are equivalent for all dependence maximal subformulas ψ of φ and their Boolean combinations: if $\mathfrak{A}$ is a model, and X is a team on $\mathfrak{A}$ such that $\mathrm{dom}(X) = V_\varphi(\psi)$, then

$$\mathfrak{A} \models_X \psi \;\Leftrightarrow\; \mathfrak{A}, s \models \psi^* \text{ for all } s \in X.$$

The proof is done by induction on the definition of the translation.

(i) If ψ is without dependence atoms, the claim holds by [16, Cor. 3.32].

(ii) Assume that $\psi = \exists \boldsymbol{y}\, \eta$ is dependence maximal, where

$$\eta = \forall \boldsymbol{x} \exists \boldsymbol{\alpha} \Big(\big(\bigwedge_{1 \leqslant i \leqslant m} {=}(V_\varphi(\eta) \cup \{\boldsymbol{x}_i\}, \alpha_i) \big) \wedge \vartheta \Big).$$

Then $\psi^* = \exists \boldsymbol{y}\, N_\pi\, \boldsymbol{x}_0 \alpha_0 \boldsymbol{x}_1 \alpha_1 \ldots \boldsymbol{x}_m \alpha_m\, \vartheta^*$. We will show that for every team Y

$$\mathfrak{A} \models_Y \eta \;\Leftrightarrow\; \mathfrak{A}, s \models N_\pi\, \boldsymbol{x}_0 \alpha_0 \boldsymbol{x}_1 \alpha_1 \ldots \boldsymbol{x}_m \alpha_m\, \vartheta^* \text{ for all } s \in Y.$$

It is easy to see that the claim for ψ follows from this.

Let $\mathfrak{A}$ be a model, and Y a team on $\mathfrak{A}$ with $\mathrm{dom}(Y) = V_\varphi(\eta)$. Now $\mathfrak{A} \models_Y \eta$ if and only if there are functions $F_1, \ldots, F_m : Y(A^n/\boldsymbol{x}) \to \{\bot, \top\}$, for $n = |\boldsymbol{x}|$, such that

$$\mathfrak{A} \models_Z \Big(\bigwedge_{1 \leqslant i \leqslant m} {=}(V_\varphi(\eta) \cup \{\boldsymbol{x}_i\}, \alpha_i) \Big) \wedge \vartheta, \text{ where } Z = Y(A^n/\boldsymbol{x}, \boldsymbol{F}/\boldsymbol{\alpha}). \tag{6}$$

Assume that (6) is true. For each $s \in Y$, let $f_i^s : A^{n_i} \to \{\bot, \top\}$, $1 \leqslant i \leqslant m$, be the functions such that $f_i^s(\boldsymbol{a}_i) = F_i(s(\boldsymbol{a}/\boldsymbol{x}))$, whenever $\boldsymbol{a}_i \in A^{n_i}$ is the restriction of $\boldsymbol{a}$ to the variables in $\boldsymbol{x}_i$. Note that f_i^s is well-defined, since by the first conjunct in (6), F_i is $V_\varphi(\eta) \cup \{\boldsymbol{x}_i\}$-determined. Let $f_0^s(\boldsymbol{a}_0) := \top$ for all $\boldsymbol{a}_0 \in A^{n_0}$.

If $\boldsymbol{a}_i \in A^{n_i}$ are tuples such that $\boldsymbol{a}_0\boldsymbol{a}_1 \dots \boldsymbol{a}_m$ is of pattern π, then we see that the assignment $s' := s(\boldsymbol{a}_0/\boldsymbol{x}_0, \dots, \boldsymbol{a}_m/\boldsymbol{x}_m, f_1^s(\boldsymbol{a}_1)/\alpha_1, \dots, f_m^s(\boldsymbol{a}_m)/\alpha_m)$ is in Z. Since $\mathfrak{A} \models_Z \vartheta$, by induction hypothesis we have $\mathfrak{A}, s' \models \vartheta^*$. Since α_0 does not occur in ϑ^*, $\mathfrak{A}, s'' \models \vartheta^*$, where $s'' = s'(\top/\alpha_0)$. Hence, the functions f_i^s, $i \leqslant m$, are as required in the truth condition of N_π, and we conclude that $\mathfrak{A}, s \models N_\pi \boldsymbol{x}_0\alpha_0\boldsymbol{x}_1\alpha_1 \dots \boldsymbol{x}_m\alpha_m \vartheta^*$ for all $s \in Y$.

Assume on the other hand that, for each $s \in Y$, $f_i^s : A^{n_i} \to \{\bot, \top\}$, $i \leqslant m$, are functions witnessing that $\mathfrak{A}, s \models N_\pi \boldsymbol{x}_0\alpha_0\boldsymbol{x}_1\alpha_1 \dots \boldsymbol{x}_m\alpha_m \vartheta^*$. Then we define $F_1, \dots, F_m : Y(A^n/\boldsymbol{x}) \to \{\bot, \top\}$ by setting $F_i(s(\boldsymbol{a}/\boldsymbol{x})) := f_i^s(\boldsymbol{a}_i)$, where $\boldsymbol{a}_i \in A^{n_i}$ is the restriction of $\boldsymbol{a}$ to the variables in $\boldsymbol{x}_i$. The functions F_i are then obviously $V_\varphi(\eta) \cup \{\boldsymbol{x}_i\}$-determined, whence $\mathfrak{A} \models_Z \bigwedge_{1 \leqslant i \leqslant m} {=}(V_\varphi(\eta) \cup \{\boldsymbol{x}_i\}, \alpha_i)$ for $Z := Y(A^n/\boldsymbol{x}, \boldsymbol{F}/\boldsymbol{\alpha})$. Furthermore, if $s' \in Z$, then it is of the form

$$s' = s(\boldsymbol{a}_0/\boldsymbol{x}_0, \dots, \boldsymbol{a}_m/\boldsymbol{x}_m, f_1^s(\boldsymbol{a}_1)/\alpha_1, \dots, f_m^s(\boldsymbol{a}_m)/\alpha_m),$$

whence $\mathfrak{A}, s' \models \vartheta^*$. Thus, by induction hypothesis, $\mathfrak{A} \models_Z \vartheta$, and we conclude that $\mathfrak{A} \models_Y \eta$.

(iii) The case $\psi = \vartheta \otimes \eta$ for $\otimes \in \{\wedge, \vee\}$, is trivial.

For the direction $\mathcal{FO}(\mathcal{POC}^+) \leqslant \mathcal{BBD}$, assume that $\varphi \in \mathcal{FO}(\mathcal{POC}^+)$ is a sentence; assume w.l.o.g that φ is in variable normal form. We define recursively a translation $\psi \mapsto \psi^+$ for all subformulas ψ of φ as follows. If ψ is a literal, we let $\psi^+ := \psi$. Furthermore, we define $(\vartheta \otimes \eta)^+ := \vartheta^+ \otimes \eta^+$ for $\otimes \in \{\wedge, \vee\}$, $(\exists x\vartheta)^+ := \exists x\vartheta^+$ and $(\forall x\vartheta)^+ := \forall x\vartheta^+$. Finally, if $\psi = N_\pi \boldsymbol{x}_1\alpha_1 \dots \boldsymbol{x}_m\alpha_m \vartheta$, then we define

$$\psi^+ := \forall \boldsymbol{x} \exists \alpha_1 \dots \exists \alpha_m \Big(\Big(\bigwedge_{1 \leqslant i \leqslant m} {=}(V_\varphi(\psi) \cup \{\boldsymbol{x}_i\}, \alpha_i) \Big) \wedge \vartheta^+ \Big),$$

where $\boldsymbol{x}$ lists all variables in the tuples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$.

It is now easy to prove by induction that for every subformula ψ of φ

$$\mathfrak{A} \models_X \psi^+ \;\Leftrightarrow\; \mathfrak{A}, s \models \psi \text{ for all } s \in X$$

holds for all models $\mathfrak{A}$ and teams X on $\mathfrak{A}$ such that $\mathrm{dom}(X) = V_\varphi(\psi)$.

Corollary 1. *$\mathcal{RBD} \equiv \mathcal{POC}[\mathcal{FO}]$ and $\forall$-$\mathcal{BD} \equiv \mathcal{POC}[\mathcal{QF}]$.*

Proof. The equivalences follow immediatelly from the proof of Theorem 1.

Theorem 2 (Theorem A1 in the Appendix). *$\mathcal{BD} \equiv \mathcal{D}$*

4 Separations

Lemma 1. *Let $\mathfrak{A}$ be a model, $\mathfrak{B}$ a submodel of $\mathfrak{A}$ and φ a $\forall$-$\mathcal{BD}$-sentence. If $\mathfrak{A} \models \varphi$ then $\mathfrak{B} \models \varphi$.*

Proof. By Corollary 1, $\forall$-$\mathcal{BD} \equiv \mathcal{POC}[\mathcal{QF}]$, and by [7, Lemma 6] $\mathcal{POC}[\mathcal{QF}]$ is equivalent with strict Σ_1^1. For strict Σ_1^1 the claim follows from [2, Lemma 1.2].

Proposition 4. $\forall\text{-}\mathcal{BD} < \mathcal{RBD}$

Proof. By Proposition 2, $\forall\text{-}\mathcal{BD} \leqslant \mathcal{RBD}$. Since by Lemma 1 the truth of a $\forall\text{-}\mathcal{BD}$-sentence is preserved under submodels, it is enough to give an $\mathcal{RBD}$-sentence for which this does not hold. Clearly $\exists x \exists y\, x \neq y$ is such a sentence.

Definition 11. *Let $\mathcal{L}$ be a logic (or a fragment of a logic), τ a vocabulary and $\mathfrak{A}$, $\mathfrak{B}$ first-order structures over τ. Then we write $\mathfrak{A} \Rrightarrow_{\mathcal{L}} \mathfrak{B}$ if and only if the implication $\mathfrak{A} \models \varphi \Rightarrow \mathfrak{B} \models \varphi$ holds for all sentences $\varphi \in \mathcal{L}$.*

Let $N_\pi[\mathcal{FO}_r]$ denote the set of all sentences in $\mathcal{POC}[\mathcal{FO}]$ which are of the form $N_\pi \boldsymbol{x}_1 \alpha_1 \dots \boldsymbol{x}_m \alpha_m \varphi$, where φ is a first-order formula with quantifier rank at most r. We will next define an Ehrenfeucht-Fraïssé game that captures the truth preservation relation $\mathfrak{A} \Rrightarrow_{N_\pi[\mathcal{FO}_r]} \mathfrak{B}$. This game is a straightforward modification of the corresponding game for $\mathsf{D}[\mathcal{FO}]$ by Sevenster and Tulenheimo [15], which in turn is based on the game for $\mathcal{FO}(\mathsf{D})$ by Sandu and Väänänen [13].

Definition 12. *Let $\mathfrak{A}$ and $\mathfrak{B}$ be first-order structures over a vocabulary τ and $r \geqslant 0$. The $N_\pi[\mathcal{FO}_r]$-EF game $N_\pi \mathrm{EF}_r(\mathfrak{A}, \mathfrak{B})$ on $\mathfrak{A}$ and $\mathfrak{B}$ is played by two players, Spoiler (S) and Duplicator (D), and it has two phases.*

Phase 1:

- *S chooses functions $f_i : A^{n_i} \to \{\bot, \top\}$, $1 \leqslant i \leqslant m$, where $n_i = |\boldsymbol{x}_i|$.*
- *D answers by choosing functions $g_i : B^{n_i} \to \{\bot, \top\}$, $1 \leqslant i \leqslant m$.*
- *S chooses tuples $\boldsymbol{b}_i \in B^{n_i}$, $1 \leqslant i \leqslant m$ such that $\boldsymbol{b}_1 \dots \boldsymbol{b}_m$ is of pattern π.*
- *D answers by choosing tuples $\boldsymbol{a}_i \in A^{n_i}$, $1 \leqslant i \leqslant m$ such that $\boldsymbol{a}_1 \dots \boldsymbol{a}_m$ is of pattern π, and $f_i(\boldsymbol{a}_i) = g_i(\boldsymbol{b}_i)$ for each $1 \leqslant i \leqslant m$. If there are no such tuples $\boldsymbol{a}_1, \dots, \boldsymbol{a}_m$, then D loses the play of the game.*

Phase 2:

- *S and D play the usual first-order EF-game of r rounds on the structures $(\mathfrak{A}, \boldsymbol{a}_1 \dots \boldsymbol{a}_m)$ and $(\mathfrak{B}, \boldsymbol{b}_1 \dots \boldsymbol{b}_m)$: In each round $1 \leqslant j \leqslant r$, S picks an element $c_j \in A$ (or $d_j \in B$), and D answer by choosing an element $d_j \in B$ (or $c_j \in A$, respectively).*

D wins the play of the game if and only if the mapping $\boldsymbol{a}_1 \dots \boldsymbol{a}_m c_1 \dots c_r \mapsto \boldsymbol{b}_1 \dots \boldsymbol{b}_m d_1 \dots d_r$ is a partial isomorphism $\mathfrak{A} \to \mathfrak{B}$.

We show next that the game $N_\pi \mathrm{EF}_r$ can be used for studying the truth preservation relation $\Rrightarrow_{N_\pi[\mathcal{FO}_r]}$.

Lemma 2 (Lemma A6 in the Appendix). *Let $\mathfrak{A}$ and $\mathfrak{B}$ be first-order structures over a vocabulary τ, π a pattern and $r \geqslant 0$. If D has a winning strategy in the game $N_\pi \mathrm{EF}_r(\mathfrak{A}, \mathfrak{B})$, then $\mathfrak{A} \Rrightarrow_{N_\pi[\mathcal{FO}_r]} \mathfrak{B}$.*

Theorem 3. $\mathcal{RBD} < \mathcal{BBD}$

Proof. We show that non-connectivity of graphs is definable in $\mathcal{BBD}$, but not in $\mathcal{RBD}$. Note first that a graph $\mathfrak{A} = (A, E^{\mathfrak{A}})$ is not connected if and only if there

is a subset $U \subseteq A$ such that U and $A \backslash U$ are non-empty, and there are no edges $(a, b) \in E^{\mathfrak{A}}$ between U and $A \backslash U$. This can be expressed by the $\mathcal{BBD}$-sentence

$$\exists u \exists v \forall x \forall y \exists \alpha \exists \beta \big(=(x, \alpha) \wedge =(y, \beta) \wedge (x = y \to (\alpha \leftrightarrow \beta)) \\ \wedge (x = u \to \alpha) \wedge (x = v \to \neg\alpha) \wedge (\alpha \wedge \neg\beta \to \neg Exy) \big).$$

We use Lemma 2 to prove that non-connectivity is not definable in $\mathcal{POC}[\mathcal{FO}]$. By Corollary 1, it then follows that non-connectivity is not definable in $\mathcal{RBD}$. Let us fix the number of rounds $r \geqslant 0$, and the pattern π, and consider the game $N_\pi \mathrm{EF}_r$. Let $\mathfrak{A} = (A, E^{\mathfrak{A}})$ and $\mathfrak{B} = (B, E^{\mathfrak{B}})$ be the graphs such that

- $B = \{u_1, \ldots, u_k\}$, and $A = B \cup \{v_1, \ldots, v_k\}$,
- $E^{\mathfrak{B}} = \{(u_i, u_j) \mid |i - j| = 1\} \cup \{(u_1, u_k), (u_k, u_1)\}$,
- $E^{\mathfrak{A}} = E^{\mathfrak{B}} \cup \{(v_i, v_j) \mid |i - j| = 1\} \cup \{(v_1, v_k), (v_k, v_1)\}$.

Thus, $\mathfrak{B}$ is a cycle of length k, and $\mathfrak{A}$ is the disjoint union of two cycles of length k. In particular, $\mathfrak{A}$ is not connected, but $\mathfrak{B}$ is connected. We will now show that if k is large enough, then D has a winning strategy in the game $N_\pi \mathrm{EF}_r(\mathfrak{A}, \mathfrak{B})$.

Let $f_i : A^{n_i} \to \{\bot, \top\}$, $1 \leqslant i \leqslant m$, be the first move of S in the game. The answer of D is then $g_i : B^{n_i} \to \{\bot, \top\}$, $1 \leqslant i \leqslant m$, where $g_i := f_i \upharpoonright B$ for each $1 \leqslant i \leqslant m$. In the next move S picks tuples $\boldsymbol{b}_i \in B^{n_i}$, $1 \leqslant i \leqslant m$, such that $\boldsymbol{b}_1 \ldots \boldsymbol{b}_m$ is of pattern π. Now D can simply answer by choosing the same tuples: let $\boldsymbol{a}_i = \boldsymbol{b}_i$ for each $1 \leqslant i \leqslant m$. Clearly the requirement $f_i(\boldsymbol{a}_i) = g_i(\boldsymbol{b}_i)$ is then satisfied. The game continues after this as the first-order EF-game with r rounds on the structures $(\mathfrak{A}, \boldsymbol{a}_1 \ldots \boldsymbol{a}_m)$ and $(\mathfrak{B}, \boldsymbol{b}_1 \ldots \boldsymbol{b}_m)$. Since the mapping $\boldsymbol{a}_1 \ldots \boldsymbol{a}_m \mapsto \boldsymbol{b}_1 \ldots \boldsymbol{b}_m$ respects distances between nodes in the graphs $\mathfrak{A}$ and $\mathfrak{B}$, a standard argument shows that Duplicator has a winning strategy in the rest of the game, provided that k is big enough.

Proposition 5. *$\mathcal{BBD} < \mathcal{BD}$, and moreover $\mathcal{BD} \not\leqslant \mathcal{FO}(\mathcal{POC})$.*

Proof. In [7] is was proven that $\mathcal{FO}(\mathsf{D})$ has 0-1 law, thus by Proposition 1 and Theorem 1, neither $\mathcal{BBD}$ nor $\mathcal{FO}(\mathcal{POC})$ can express even cardinality of the domain. But by Theorem 2 and [16, Ch. 4.1] this property is definable in $\mathcal{BD}$.

5 Conclusion

In this paper we define a new variant of dependence logic called Boolean dependence logic. Boolean dependence logic is an extension of first-order logic with dependence atoms of the form $=(\boldsymbol{x}, \alpha)$, where $\boldsymbol{x}$ is a tuple of first-order variables and α is a Boolean variable. We also introduce a notational variant of partially-ordered connectives based on the narrow Henkin quantifiers of [3]. We show that the expressive power of Boolean dependence logic and dependence logic coincide. We define natural syntactic fragments of Boolean dependence logic and prove that the expressive power of these fragments coincide with corresponding logics based on partially-ordered connectives. Finally we prove that the fragments of Boolean dependence logic form a strict hierarchy in terms of expressive power.

References

1. Abramsky, S., Väänänen, J.: From IF to BI. Synthese 167(2) (2009)
2. Barwise, J.: Applications of Strict Π^1_1 Predicates to Infinitary Logic. J. Symb. Log. 34(3) (1969)
3. Blass, A., Gurevich, Y.: Henkin quantifiers and complete problems. Annals of Pure and Applied Logic 32 (1986)
4. Durand, A., Kontinen, J.: Hierarchies in Dependence Logic. ACM Transactions on Computational Logic 13(4) (2012)
5. Grädel, E., Väänänen, J.: Dependence and independence. Studia Logica (2013)
6. Enderton, H.B.: Finite partially-ordered quantifiers. Z. Math. Logik Grundlagen Math. 16, 393–397 (1970)
7. Hella, L., Sevenster, M., Tulenheimo, T.: Partially Ordered Connectives and Monadic Monotone Strict NP. J. of Logic, Lang. and Inf. 17(3) (2008)
8. Henkin, L.: Some remarks on infinitely long formulas. In: Infinitistic Methods (Proc. Sympos. Foundations of Math., Warsaw, 1959). Pergamon, Oxford (1961)
9. Hintikka, J.: The principles of mathematics revisited. Cambridge Univ. Press (1996)
10. Hintikka, J., Sandu, G.: Informational independence as a semantical phenomenon. In: Logic, Methodology and Philosophy of Science, VIII (Moscow, 1987). Stud. Logic Found. Math., vol. 126, pp. 571–589 (1989)
11. Kontinen, J., Kuusisto, A., Lohmann, P., Virtema, J.: Complexity of two-variable Dependence Logic and IF-Logic. In: Proceedings of LICS 2011, pp. 289–298 (2011)
12. Lohmann, P., Vollmer, H.: Complexity results for modal dependence logic. In: Dawar, A., Veith, H. (eds.) CSL 2010. LNCS, vol. 6247, pp. 411–425. Springer, Heidelberg (2010)
13. Sandu, G., Väänänen, J.: Partially ordered connectives. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 38 (1992)
14. Sevenster, M.: Model-theoretic and computational properties of modal dependence logic. J. Log. Comput. 19(6), 1157–1173 (2009)
15. Sevenster, M., Tulenheimo, T.: Partially Ordered Connectives and $\sum^1_1$ on Finite Models. In: Beckmann, A., Berger, U., Löwe, B., Tucker, J.V. (eds.) CiE 2006. LNCS, vol. 3988, pp. 516–525. Springer, Heidelberg (2006)
16. Väänänen, J.: Dependence logic: A new approach to independence friendly logic. London Math. Soc. Stud. Texts, vol. 70 (2007)
17. Walkoe Jr., W.J.: Finite partially-ordered quantification. J. Symbolic Logic 35, 535–555 (1970)

A Appendix

Definition A1. *Let φ and ψ be $\mathcal{BD}$-formulae of vocabulary τ. We say that the formulae φ and ψ are* equivalent *and write $\varphi \equiv \psi$ if for all τ-models $\mathfrak{A}$ and teams X of $\mathfrak{A}$ it holds that $\mathfrak{A}, X \models \varphi$ iff $\mathfrak{A}, X \models \psi$.*

Proposition A1 (Analogous to [16, Lemma 3.27]). *Let φ be a $\mathcal{BD}$-formula of vocabulary τ, $\mathfrak{A}$ a τ-model and X a team of $\mathfrak{A}$. If $V \supseteq \mathrm{Fr}(\varphi)$, then $\mathfrak{A}, X \models \varphi$ if and only if $\mathfrak{A}, (X \upharpoonright V) \models \varphi$.*

Lemma A1 (Analogous to [16, Lemma 3.25]). *Suppose that θ, $\varphi_0(\boldsymbol{x}, \boldsymbol{\alpha})$ and $\varphi_1(\boldsymbol{x}, \boldsymbol{\alpha})$ are $\mathcal{BD}$-formulae such that $\varphi_0(\boldsymbol{x}, \boldsymbol{\alpha}) \equiv \varphi_1(\boldsymbol{x}, \boldsymbol{\alpha})$. Then $\theta \equiv \theta(\varphi_1/\varphi_0)$.*

Lemma A2. *Let φ be a $\mathcal{BD}$-formula and let $\boldsymbol{x}, \boldsymbol{\alpha}$ be the free variables of φ. Let $\boldsymbol{y}, \boldsymbol{\beta}$, $|\boldsymbol{y}| = |\boldsymbol{x}|$, $|\boldsymbol{\beta}| = |\boldsymbol{\alpha}|$, be tuples of variables that do not occur in φ. Let φ' be the formula obtained from φ by replacing each free occurence of the variables $\boldsymbol{x}, \boldsymbol{\alpha}$ by $\boldsymbol{y}, \boldsymbol{\beta}$. Let $\mathfrak{A}$ be a model and X a team of $\mathfrak{A}$ where $\mathrm{dom}(X) = \{\boldsymbol{x}, \boldsymbol{\alpha}\}$. Let X' be the team consisting of the assignments $s'(\boldsymbol{y}, \boldsymbol{\beta}) = s(\boldsymbol{x}, \boldsymbol{\alpha})$, $s \in X$.*

Now $\mathfrak{A}, X \models \varphi \Leftrightarrow \mathfrak{A}, X' \models \varphi'$.

Lemma A3. *Let $\varphi, \vartheta \in \mathcal{BD}$ such that $x, \alpha \notin \mathrm{Fr}(\vartheta)$. The following hold.*

1. $(\forall x\varphi) \vee \vartheta \equiv \forall x(\varphi \vee \vartheta)$.
2. $(\forall x\varphi) \wedge \vartheta \equiv \forall x(\varphi \wedge \vartheta)$.
3. $(\exists \alpha\varphi) \vee \vartheta \equiv \exists \alpha(\varphi \vee \vartheta)$.
4. $(\exists \alpha\varphi) \wedge \vartheta \equiv \exists \alpha(\varphi \wedge \vartheta)$.

Lemma A4. *Let φ be a $\mathcal{BD}$ formula and $\psi = \exists\alpha\forall\boldsymbol{x}\exists\boldsymbol{\beta}\vartheta$ a subformula of φ. Then $\varphi \equiv \varphi\Big(\forall\boldsymbol{x}\exists\alpha\exists\boldsymbol{\beta}\big(=\!(V_\varphi(\psi), \alpha) \wedge \vartheta\big)/\psi\Big)$.*

Lemma A5 (Follows from Lemmas A1 and A2.). *Let φ be a $\mathcal{BD}$ ($\mathcal{BBD}$) formula of vocabulary τ. There exists a $\mathcal{BD}$ ($\mathcal{BBD}$) formula φ^* in variable normal form such that $\varphi \equiv \varphi^*$.*

Proposition A2. *Every sentence in $\mathcal{BBD}$ has an equivalent sentence in Q-normal form.*

Proof. Let $\varphi \in \mathcal{BBD}$. By Lemma A5 we can assume that φ is in variable normal form. We will give an algorithm that transforms φ to an equivalent $\mathcal{BBD}$ formula in Q-normal form.

We will first transform φ to an equivalent $\mathcal{BBD}$ formula φ^* such that

$$\varphi^* = \boldsymbol{Q\xi}\psi \tag{7}$$

where $\boldsymbol{Q\xi}$ is a (possibly empty) vector of universal first-order and existential Boolean quantifiers. Furthermore in ψ every universal first-order or existential Boolean quantifier $Q\chi$ occurs in a subformula θ of ψ such that $\theta = Q'\eta Q\chi\gamma$ where $Q'\eta$ is a quantifier and γ is a $\mathcal{BBD}$ formula. In order to obtain φ^* from φ we use the equivalences from Lemma A3 repetitively substituting subformulae with equivalent subformulae. More precisely there exists a natural number $n \in \mathbb{N}$ and a tuple $(\varphi_i)_{i \leqslant n}$ of $\mathcal{BBD}$ formulae such that $\varphi_0 = \varphi$ and $\varphi_n = \varphi^*$. Furthermore

1. for each $i < n$ there exist subformulae θ, ψ_1 and ψ_2 of φ_i such that $\theta = (Q\chi\psi_1) \otimes \psi_2$ (or $\theta = \psi_1 \otimes (Q\chi\psi_2)$), where $Q\chi \in \{\forall x, \exists\alpha\}$ and $\otimes \in \{\vee, \wedge\}$, and φ_{i+1} is obtained from φ_i by substituting θ by $Q\chi(\psi_1 \otimes \psi_2)$.

It is easy to see that for each $\mathcal{BBD}$ formula the substitution procedure described in 1. terminates, i.e., there exists some $n \in \mathbb{N}$ such that there are no subformulae of φ_n that can be substituted as described in 1. Clearly φ_n is in the form described in (7). By induction it is easy to show, that since φ_0 is in variable normal form it follows that φ_i is in variable normal form for all $i \geqslant 0$. Hence the assumptions on free variables needed for Lemma A3 hold for each φ_i. By Lemmas A3 and A1 we conclude that for each $i < n$ the formulae φ_i and φ_{i+1} are equivalent. Hence the formulae φ_0 and φ_n are equivalent.

We still need to transform the formula φ^* into an equivalent formula φ' in Q-normal form. In order to obtain φ' from φ^* we use the equivalence from Lemma A4 repetitively. More precisely there exists a natural number $m \in \mathbb{N}$ and a tuple

$(\varphi_i^*)_{i\leqslant m}$ of $\mathcal{BBD}$ formulae such that $\varphi_0^* = \varphi^*$ and $\varphi_m^* = \varphi'$. Furthermore

2. for each $i < m$ there exists subformulae θ and ψ of φ_i^* and a quantifier $\exists\alpha$ such that $\theta = \exists\alpha\forall\boldsymbol{x}\exists\boldsymbol{\beta}\psi$ where $\forall\boldsymbol{x}$ is a nonempty vector of universal first-order quantifiers and ψ does not start with a quantifier, and φ_{i+1}^* is obtained from φ_i^* by substituting θ by $\forall\boldsymbol{x}\exists\alpha\exists\boldsymbol{\beta}\big(=(V_{\varphi_i^*}(\theta),\alpha)\wedge\psi\big)$.

By Lemmas A4 and A1 we conclude that for each i the formulae φ_i^* and φ_{i+1}^* are equivalent. It is easy to see that for each $\mathcal{BBD}$ formula the substitution procedure described above terminates, i.e., there exists some $m \in \mathbb{N}$ such that there are no subformulae of φ_m^* that can be substituted as descibed in 2. Now clearly φ_m^* is in Q-normal form.

Proposition A3. *Every sentence in $\mathcal{BBD}$ has an equivalent sentence in dependence normal form.*

Proof. Let $\varphi \in \mathcal{BBD}$. By Proposition A2 we can assume that φ is in Q-normal form. We will give an algorithm that transforms φ to an equivalent $\mathcal{BBD}$ formula φ^* in dependence normal form. We show that there exists a tuple $(\varphi_i)_{i\leqslant n}$, $n \in \mathbb{N}$, of equivalent $\mathcal{BBD}$ formulae in Q-normal form such that $\varphi_0 = \varphi$, $\varphi_n = \varphi^*$.

Assume φ_i is not in dependence normal form. Assume first that this is due to the fact that there exists a dependence atom $=(\boldsymbol{x},\alpha)$ and subformulae ψ and θ of φ_i such that $\theta = \exists\boldsymbol{y}\forall\boldsymbol{z}\exists\boldsymbol{\beta}\psi$, where $=(\boldsymbol{x},\alpha)$ is a subformula of ψ not bound by any quantifier in ψ that violates the condition of Definition 10. Let U denote the set of variables that are in the vector $\boldsymbol{x}$ but not in the set $V_{\varphi_i}(\forall\boldsymbol{z}\exists\boldsymbol{\beta}\psi)$, and let $\boldsymbol{u}$ be the variables from U in some order. Since φ_i is in $\mathcal{BBD}$ the formula $=(V_{\varphi_i}(\forall\boldsymbol{z}\exists\boldsymbol{\beta}\psi)\cup U,\alpha)$ and $=(\boldsymbol{x},\alpha)$ are equivalent. Therefore due to Lemma A1 we may assume that $=(\boldsymbol{x},\alpha)$ is $=(V_{\varphi_i}(\forall\boldsymbol{z}\exists\boldsymbol{\beta}\psi)\cup U,\alpha)$.

Let $\theta' := \exists\boldsymbol{y}\forall\boldsymbol{z}\forall\boldsymbol{w}\exists\boldsymbol{\beta}\exists\beta'\big(=(V_{\varphi_i}(\forall\boldsymbol{z}\exists\boldsymbol{\beta}\psi)\cup W,\beta')\wedge\psi'\big)$, where $\boldsymbol{w}$ is a tuple of fresh distinct first-order variables of the same length as $\boldsymbol{u}$, W is the set of variables in the tuple $\boldsymbol{w}$, β' is a fresh Boolean variable and $\psi' = \psi((\boldsymbol{u} = \boldsymbol{w} \to \alpha = \beta')/=(\boldsymbol{x},\alpha))$. Notice that the formulae θ and θ' are equivalent. To see this, first observe that since the variables $\boldsymbol{w},\beta'$ do not occur in ψ it is easy to conclude by Proposition A1 that θ is equivalent to the formula $\gamma := \exists\boldsymbol{y}\forall\boldsymbol{z}\forall\boldsymbol{w}\exists\boldsymbol{\beta}\exists\beta'\big(=(V_{\varphi_i}(\forall\boldsymbol{z}\exists\boldsymbol{\beta}\psi)\cup W,\beta')\wedge\psi\big)$.

We still need to show that γ is equivalent to θ'. The idea here is that $\boldsymbol{w}$ and β' are used to encode a partial function needed for the satisfaction of the dependence atom $=(\boldsymbol{x},\alpha)$ in ψ. First notice that for each team X of $\mathfrak{A}$ the following two conditions are equivalent to $\mathfrak{A}, X \models \gamma$.

1. $\mathfrak{A}, Y \models =(V_{\varphi_i}(\forall\boldsymbol{z}\exists\boldsymbol{\beta}\psi)\cup W,\beta')\wedge\psi$, for some team Y obtained from X by evaluating the quantifier prefix of γ.
2. $\mathfrak{A}, Y' \models =(V_{\varphi_i}(\forall\boldsymbol{z}\exists\boldsymbol{\beta}\psi)\cup W,\beta')\wedge\psi$, for every team Y' that can be obtained from Y (Y as above) by changing the values of β' in any such way that Y' still satisfies $=(V_{\varphi_i}(\forall\boldsymbol{z}\exists\boldsymbol{\beta}\psi)\cup W,\beta')$.

Assume then that 1 holds. While there might be many strategies for $\mathfrak{A}, Y$ to satisfy ψ, i.e., there might be many different ways to split the team on disjunctions, for each instance e of evaluating ψ on $\mathfrak{A}, Y$ there is a team Y_e such

that $\mathfrak{A}, Y_e \models =(V_{\varphi_i}(\forall \boldsymbol{z}\, \exists \boldsymbol{\beta}\, \psi) \cup U, \alpha)$ is required to hold. Therefore there exists a partial function $f_e : A^{|V_{\varphi_i}(\forall \boldsymbol{z}\, \exists \boldsymbol{\beta}\, \psi) \cup U|} \to \{0,1\}$ that maps the values of the variables from $V_{\varphi_i}(\forall \boldsymbol{z}\, \exists \boldsymbol{\beta}\, \psi) \cup U$ to the value of α in the team Y_e. Let $g_e : A^{|V_{\varphi_i}(\forall \boldsymbol{z}\, \exists \boldsymbol{\beta}\, \psi) \cup U|} \to \{0,1\}$ be a function such that $f_e \subseteq g_e$ and let Y^{g_e} be the variant of Y where the values for β' have been picked using the function g_e. It is quite easy to see that $\mathfrak{A}, Y_e^g \models \boldsymbol{w} = \boldsymbol{u} \to \alpha = \beta'$, where $Y_e^{g_e}$ is the team obtained from Y^{g_e} analogously to Y_e from Y. Therefore due to the equivalence of 1 and 2, we conclude that $\mathfrak{A}, X \models \theta'$.

Assume then that $\mathfrak{A}, X \models \theta'$. Hence $\mathfrak{A}, Y \models =(V_{\varphi_i}(\forall \boldsymbol{z}\, \exists \boldsymbol{\beta}\, \psi) \cup W, \beta') \wedge \psi'$, for some team Y obtained from X by evaluating the quantifier prefix of θ'. Similarly as above, for each strategy e to evaluate ψ' in $\mathfrak{A}, Y$ there is a team Y_e such that $\mathfrak{A}, Y_e \models \boldsymbol{w} = \boldsymbol{u} \to \alpha = \beta'$ is required to hold. Since the variables $\boldsymbol{w}, \beta'$ do not occur in other subformulae of ψ' than $\boldsymbol{w} = \boldsymbol{u} \to \alpha = \beta'$, we may assume w.l.o.g that for each assignment $s \in Y_e$ and $\boldsymbol{a} \in A^{|\boldsymbol{w}|}$ the modified assignment $s' \in Y$ of s that maps $\boldsymbol{w}$ to $\boldsymbol{a}$ and β' to 0 or 1 is also in Y_e. Now since $\mathfrak{A}, Y \models =(V_{\varphi_i}(\forall \boldsymbol{z}\, \exists \boldsymbol{\beta}\, \psi) \cup W, \beta')$ and $Y_e \subseteq Y$ we conclude that $\mathfrak{A}, Y_e \models =(V_{\varphi_i}(\forall \boldsymbol{z}\, \exists \boldsymbol{\beta}\, \psi) \cup U, \alpha)$ and furthermore that $\mathfrak{A}, X \models \gamma$. Hence θ and θ' are equivalent.

Let φ_{i+1} be the formula obtained from φ_i by substituting θ with θ'. By the above observation and Lemma A1 we conclude that φ_i and φ_{i+1} are equivalent. Notice if φ_i is in Q-normal form, then φ_{i+1} is also in Q-normal form. It is easy to see that for large enough k the formula φ_k does not have dependence atoms that violate the condition in Definition 10. Hence if φ_k is not in dependence normal form there exists a subformula $\exists \beta\, \exists \boldsymbol{\alpha}\, \psi$ of φ_k such that there is no dependence atom $=(\boldsymbol{x}, \beta)$ in ψ that is not bound by any quantifier in ψ for any $\boldsymbol{x}$. Let φ_{k+1} denote the formula obtained from φ_k by substituting ψ by $\big(=(V_{\varphi_k}(\psi), \beta) \wedge \psi\big)$. Clearly φ_k and φ_{k+1} are equivalent. It is easy to see that the procedure described here terminates and finally produces an equivalent formula in dependence normal form.

Theorem A1. $\mathcal{BD} \equiv \mathcal{D}$

Proof. $\mathcal{BD} \leqslant \mathcal{D}$ holds by Proposition 2. For the other direction we will give a translation from $\mathcal{D}$ to $\mathcal{BD}$. Let φ be an arbitrary $\mathcal{D}$-sentence in the normal form for $\mathcal{D}$ from [16, p. 98], i.e., $\varphi := \forall \boldsymbol{x}\, \exists \boldsymbol{y}\, \big(\bigwedge_{i \in I} =(\boldsymbol{x}_i, y_i) \ \wedge \ \psi \big)$, where ψ is a quantifier-free first-order formula and I a finite index set, and $\boldsymbol{x}_i$ is a vector of variables from $\boldsymbol{x}$ and y_i a variable from $\boldsymbol{y}$, for every $i \in I$. The translation φ^* of φ is the $\mathcal{BD}$-sentence

$$\forall \boldsymbol{x}\, \exists \boldsymbol{y} \big(\psi \wedge \forall \boldsymbol{z}\, \exists \boldsymbol{\alpha} \bigwedge_{i \in I} \big(=(\boldsymbol{x}_i, z_i, \alpha_i) \wedge (z_i = y_i \leftrightarrow \alpha_i) \big) \big),$$

where $\boldsymbol{z}$ and $\boldsymbol{\alpha}$ are tuples of fresh variables of length $|I|$ and z_i and α_i, $i \in I$, are variables from the corresponding tuple. We will show that for every model $\mathfrak{A}$ it holds that $\mathfrak{A} \models_{\{\varnothing\}} \varphi$ iff $\mathfrak{A} \models_{\{\varnothing\}} \varphi^*$.

Assume first that $\mathfrak{A} \models_{\{\varnothing\}} \varphi$. Hence $\mathfrak{A} \models_X \bigwedge_{i \in I} =(\boldsymbol{x}_i, y_i) \ \wedge \ \psi$ for some team X obtained from $\{\varnothing\}$ by evaluating the quantifier prefix of φ. Now since $\mathfrak{A} \models_X \bigwedge_{i \in I} =(\boldsymbol{x}_i, y_i)$ there exists, for every $i \in I$, a function $F_i : A^{|\boldsymbol{x}_i|} \to A$ that

maps the values of the variables $\boldsymbol{x}_i$ to the value of the variable y_i in the team X. Let Y denote the team $X(A^{|\boldsymbol{z}|}/\boldsymbol{z}, (H_i/\alpha_i)_{i\in I})$, where $H_i : X(A^{|\boldsymbol{z}|}/\boldsymbol{z}) \to \{\bot, \top\}$ is obtained from F_i in the obvious way

$$H_i(s) = \begin{cases} \top & \text{iff } F_i(s(\boldsymbol{x}_i)) = z_i \\ \bot & \text{iff } F_i(s(\boldsymbol{x}_i)) \neq z_i. \end{cases}$$

Notice that $\mathfrak{A} \models_Y \bigwedge_{i\in I} \big(=(\boldsymbol{x}_i, z_i, \alpha_i) \wedge (z_i = y_i \leftrightarrow \alpha_i)\big)$. Hence we have that $\mathfrak{A} \models_X \forall \boldsymbol{z}\, \exists \boldsymbol{\alpha} \bigwedge_{i\in I} \big(=(\boldsymbol{x}_i, z_i, \alpha_i) \wedge (z_i = y_i \leftrightarrow \alpha_i)\big)$. Therefore since $\mathfrak{A} \models_X \psi$, and since X was obtained from $\{\varnothing\}$ by evaluating the quantifier prefix $\forall \boldsymbol{x} \exists \boldsymbol{y}$, we conclude that $\mathfrak{A} \models_{\{\varnothing\}} \varphi^*$.

Assume then that $\mathfrak{A} \models_{\{\varnothing\}} \varphi^*$. Hence $\mathfrak{A} \models_X \psi \wedge \forall \boldsymbol{z}\, \exists \boldsymbol{\alpha} \bigwedge_{i\in I} \big(=(\boldsymbol{x}_i, z_i, \alpha_i) \wedge (z_i = y_i \leftrightarrow \alpha_i)\big)$, for some team X that can be obtained from $\{\varnothing\}$ by evaluating the quantifier prefix of φ^*. Furthermore $\mathfrak{A} \models_Y \bigwedge_{i\in I} \big(=(\boldsymbol{x}_i, z_i, \alpha_i) \wedge (z_i = y_i \leftrightarrow \alpha_i)\big)$, for some team Y obtained from X by evaluating the quantifiers $\forall \boldsymbol{z}\, \exists \boldsymbol{\alpha}$. We will show that for each $i \in I$ it holds that $\mathfrak{A} \models_X =(\boldsymbol{x}_i, y_i)$. This together with the fact that $\mathfrak{A} \models_X \psi$ is enought to prove that $\mathfrak{A} \models_{\{\varnothing\}} \varphi$.

Let $s, t \in X$ be any two assignments such that $s(\boldsymbol{x}_i) = t(\boldsymbol{x}_i)$. Clearly there exist assignments $s', t' \in Y$ such that $s = s' \upharpoonright \mathrm{dom}(X)$, $t = t' \upharpoonright \mathrm{dom}(X)$ and $s'(z_i) = s(y_i) = t'(z_i)$. Now since $\mathfrak{A} \models_Y z_i = y_i \leftrightarrow \alpha_i$ we have that $s'(\alpha_i) = 1$. Furthermore since $\mathfrak{A} \models_Y =(\boldsymbol{x}_i, z_i, \alpha_i)$ we have that $s'(\alpha) = t'(\alpha)$. Hence $t'(\alpha) = 1$ and furthermore $t'(y_i) = t'(z_i)$. Since $t(y_i) = t'(y_i)$ and $s'(z_i) = s(y_i)$ and we conclude that $t(y_i) = s(y_i)$. Hence $\mathfrak{A} \models_X =(\boldsymbol{x}_i, y_i)$.

Lemma A6. *Let $\mathfrak{A}$ and $\mathfrak{B}$ be τ-structures, π a pattern, and $r \geqslant 0$. If D has a winning strategy in the game $N_\pi \mathrm{EF}_r(\mathfrak{A}, \mathfrak{B})$, then $\mathfrak{A} \Rrightarrow_{N_\pi[\mathcal{FO}_r]} \mathfrak{B}$.*

Proof. Assume that D has a winning strategy in the game $N_\pi \mathrm{EF}_r(\mathfrak{A}, \mathfrak{B})$. To prove $\mathfrak{A} \Rrightarrow_{N_\pi[\mathcal{FO}_r]} \mathfrak{B}$, assume that $\psi = N_\pi \boldsymbol{x}_1 \alpha_1 \dots \boldsymbol{x}_m \alpha_m \vartheta$ is a sentence of $N_\pi[\mathcal{FO}_r]$ such that $\mathfrak{A} \models \psi$. Thus, there are functions $f_i : A^{n_i} \to \{\bot, \top\}$, $1 \leqslant i \leqslant m$, such that for all $\boldsymbol{a}_i \in A^{n_i}$, if $\boldsymbol{a}_1 \dots \boldsymbol{a}_m$ is of pattern π, then $\mathfrak{A} \models_{s[\boldsymbol{a},\boldsymbol{f}]} \psi$, where $s[\boldsymbol{a}, \boldsymbol{f}]$ denotes the assignment that maps $\boldsymbol{x}_i$ to $\boldsymbol{a}_i$ and α_i to $f_i(\boldsymbol{a}_i)$, for $1 \leqslant i \leqslant m$.

Let $g_i : B^{n_i} \to \{\bot, \top\}$, $1 \leqslant i \leqslant m$, be the answer given by the winning strategy of D in the play where $f_1, \dots, f_m$ is the first move of S. It suffices to show that $\mathfrak{B} \models_{s[\boldsymbol{b},\boldsymbol{g}]} \psi$ for all tuples $\boldsymbol{b}_1 \dots \boldsymbol{b}_m$ of pattern π such that $\boldsymbol{b}_i \in B^{n_i}$ for each i. Thus, let $\boldsymbol{b}_i \in B^{n_i}$, $1 \leqslant i \leqslant m$, be arbitrary tuples such that $\boldsymbol{b}_1 \dots \boldsymbol{b}_m$ is of pattern π. Let $\boldsymbol{a}_1, \dots, \boldsymbol{a}_m \in A^n$ be the answer given by the winning strategy of D when the second move of S is $\boldsymbol{b}_1, \dots, \boldsymbol{b}_m$. By the definition of the game $N_\pi \mathrm{EF}_r$, D has then a winning strategy in the first-order EF game with r rounds between the structures $(\mathfrak{A}, \boldsymbol{a}_1, \dots, \boldsymbol{a}_m)$ and $(\mathfrak{B}, \boldsymbol{b}_1, \dots, \boldsymbol{b}_m)$. By the standard EF theorem, it follows that $(\mathfrak{A}, \boldsymbol{a}_1, \dots, \boldsymbol{a}_m)$ and $(\mathfrak{B}, \boldsymbol{b}_1, \dots, \boldsymbol{b}_m)$ satisfy the same $\mathcal{FO}_r$-sentences. Note further that $f_1(\boldsymbol{a}_1) = g_1(\boldsymbol{b}_1) \dots, f_m(\boldsymbol{a}_m) = g_m(\boldsymbol{b}_m)$ by the condition governing the choice of $\boldsymbol{b}_1, \dots, \boldsymbol{b}_m$, whence $(\mathfrak{A}, \boldsymbol{a}_1, \dots, \boldsymbol{a}_m, f_1(\boldsymbol{a}_1), \dots, f_m(\boldsymbol{a}_m))$ and $(\mathfrak{B}, \boldsymbol{b}_1, \dots, \boldsymbol{b}_m, g_1(\boldsymbol{b}_1), \dots, g_m(\boldsymbol{b}_m))$ are also equivalent with respect to all sentences of $\mathcal{FO}_r$ extended with Boolean variables. In particular, since $\mathfrak{A} \models_{s[\boldsymbol{a},\boldsymbol{f}]} \psi$, we have $\mathfrak{B} \models_{s[\boldsymbol{b},\boldsymbol{g}]} \psi$, as desired.

Extended Modal Dependence Logic $\mathcal{EMDL}^{\star}$

Johannes Ebbing[1], Lauri Hella[2], Arne Meier[1], Julian-Steffen Müller[1], Jonni Virtema[2], and Heribert Vollmer[1]

[1] Institut für Theoretische Informatik, Leibniz Universität Hannover, Appelstr. 4, 30167 Hannover, Germany
{ebbing, meier,mueller,vollmer}@thi.uni-hannover.de
[2] School of Information Sciences, University of Tampere, Kanslerinrinne 1 B, 33014 University of Tampere, Finland
{lauri.hella,jonni.virtema}@uta.fi

Abstract. In this paper we extend modal dependence logic $\mathcal{MDL}$ by allowing dependence atoms of the form $\mathrm{dep}(\varphi_1, \ldots, \varphi_n)$ where φ_i, $1 \leq i \leq n$, are modal formulas (in $\mathcal{MDL}$, only propositional variables are allowed in dependence atoms). The reasoning behind this extension is that it introduces a temporal component into modal dependence logic. E.g., it allows us to express that truth of propositions in some world of a Kripke structure depends only on a certain part of its past. We show that $\mathcal{EMDL}$ strictly extends $\mathcal{MDL}$, i.e., there exist $\mathcal{EMDL}$-formulas which are not expressible in $\mathcal{MDL}$. However, from an algorithmic point of view we do not have to pay for this since we prove that the complexity of satisfiability and model checking of $\mathcal{EMDL}$ and $\mathcal{MDL}$ coincide. In addition we show that $\mathcal{EMDL}$ is equivalent to $\mathcal{ML}$ extended by a certain propositional connective.

1 Introduction

Dependences between values of variables occur in many scientific disciplines. For example in physics there are dependences in experimental data, in computer science dependences are occurring in the execution of discrete systems and in social science they can occur between voting extrapolations. With the aim to express such dependences, Väänänen [11] extended first-order logic by adding a new kind of atomic formulas called *dependence atoms*. A dependence atom, denoted by $\mathrm{dep}(x_1, \ldots, x_n)$, intuitively states that the value of x_n is solely determined by the values of the variables $x_1, \ldots, x_{n-1}$, i.e., x_n depends functionally on $x_1, \ldots, x_{n-1}$. The extension of first-order logic with dependence atoms is called dependence logic. The idea behind the compositional semantics used for dependence logic, defined using the concept of teams, i.e., sets of assignments was introduced by Hodges [5]; for more details on dependence logic, see the monograph [11].

The study of *modal depencence logic* $\mathcal{MDL}$ was initiated by Väänänen in [12]. Modal dependence logic extends standard modal logic by modal dependence atoms, $\mathrm{dep}(p_1, \ldots, p_n)$. Teams in this context are sets of worlds of a Kripke

⋆ This paper was supported by a grant from DAAD within the PPP programme under project ID 50740539 and grant 138163 of the Academy of Finland.

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 126–137, 2013.

structure, and the meaning of the formula $\mathrm{dep}(p_1, \ldots, p_n)$ is that, in the worlds of the current team, the value of the proposition p_n is functionally determined by the values of the propositions $p_1, \ldots, p_{n-1}$.

Temporal logics have been studied widely in the field of automatic system verification. Thus, the computational complexity of temporal logics, e.g., satisfiability and model checking is important. Following [2], the *basic temporal logic* $\mathcal{TL}$ is a multi modal logic which contains two dually defined accessibility relations, R_F looking into the future and R_P looking into the past. Only Kripke models are considered in which R_F is the converse of R_P; these models are called *bidirectional models*. In this logic we can express, e.g, that in one time step in the past ($\langle R_P \rangle \varphi$) or in the next future time step ($\langle R_F \rangle \varphi$) some property φ holds (other notations used for the modalities are $\langle P \rangle, \langle F \rangle$ or $\Diamond_P, \Diamond_F$; sometimes the above formulas are succinctly written as $P\varphi$ and $F\varphi$, see [2, p. 12].) The most used temporal logics in the field of verification (computational tree logic $\mathcal{CTL}$ introduced by Emerson and Clarke [3], linear temporal logics $\mathcal{LTL}$ introduced by Sistla and Clarke [10], etc.) are extensions of the basic temporal logic $\mathcal{TL}$.

In this way, $\mathcal{MDL}$ (generalized to multi-modal frames) can bee seen as a first step towards introducing functional dependences into the framework of temporal logic. Sevenster [9] showed that the satisfiability problem for $\mathcal{MDL}$ is NEXPTIME-complete. A more detailed classification over sublogics of $\mathcal{MDL}$ was given by Lohmann and Vollmer [7]. Ebbing and Lohmann [4] studied the model checking problem for $\mathcal{MDL}$ and showed that this problem is less complex than the satisfiability problem but is still NP-complete.

Though $\mathcal{MDL}$ can be seen as a first step towards combining functional dependences and temporal logic, it is clearly not sufficient for modelling temporal aspects of dependence, since the only dependences that are allowed to occur are dependences between propositional statements within the same world. In this paper we introduce *extended modal dependence logic*, $\mathcal{EMDL}$, to overcome this defect. $\mathcal{EMDL}$ is obtained from $\mathcal{MDL}$ by extending the scope of dependence atoms to arbitrary modal formulas, i.e., dependence atoms in $\mathcal{EMDL}$ are of the form $\mathrm{dep}(\varphi_1, \ldots \varphi_n)$, where $\varphi_1, \ldots, \varphi_n$ are $\mathcal{ML}$ formulas. In contrast to $\mathcal{MDL}$, in $\mathcal{EMDL}$ we can directly express dependences between events in time, thus $\mathcal{EMDL}$ can be seen as the next logical step to combine functional dependences with temporal reasoning. As an example, the formula

$$\mathrm{dep}\big(\langle P \rangle p, \langle P \rangle^2 p, \ldots, \langle P \rangle^n p, p\big)$$

expresses that the truth of p at this moment only depends on the truth of p in the previous n time steps.

We will show that $\mathcal{EMDL}$ strictly extends $\mathcal{MDL}$ by giving an explicit $\mathcal{EMDL}$-formula that is not expressible in $\mathcal{MDL}$ (see Theorem 1). However we will prove that $\mathcal{ML}$ extended with the classical disjunction $\veebar$ can express all $\mathcal{EMDL}$-formulas (Theorem 2). In fact, we will see that the expressive power of $\mathcal{EMDL}$ coincides with that of $\mathcal{ML}$ extended by a restricted use of $\veebar$ (Corollary 1). We will also show that, although $\mathcal{EMDL}$ is strictly more expressive than $\mathcal{MDL}$, the computational complexity of the satisfiability and the model checking problems for $\mathcal{MDL}$ and $\mathcal{EMDL}$ coincide (Theorems 4 and 5).

2 Preliminaries

Complexity Theory. In this paper, we make use of the complexity classes NEXPTIME and NP. Informally speaking NEXPTIME is the class of problems that can be solved by a nondeterministic turing machine in exponential time. Similar to this, NP is a class of problems that can be computed by such a turing machine in *polynomial* time. For further information we refer to [8].

Modal Dependence Logics. *Extended modal dependence logic*, $\mathcal{EMDL}$, is obtained from usual modal logic $\mathcal{ML}$ by adding dependence atoms $\mathrm{dep}(\varphi_1, \ldots, \varphi_n)$. As in $\mathcal{MDL}$, the dependence atom $\mathrm{dep}(\varphi_1, \ldots, \varphi_n)$ intuitively states, that the truth value of φ_n is solely determined by the truth values of $\varphi_1, \ldots, \varphi_{n-1}$. The difference between $\mathcal{EMDL}$ and $\mathcal{MDL}$ is that in $\mathcal{MDL}$ the formulas $\varphi_1, \ldots, \varphi_n$ in dependence atoms are required to be atomic propositions, while $\mathcal{EMDL}$ allows arbitrary $\mathcal{ML}$-formulas in dependence atoms.

Before defining $\mathcal{EMDL}$, let us first introduce the syntax of usual modal logic $\mathcal{ML}$ and its extension by classical disjunction $\mathbin{⩒}$.

Definition 1 (Syntax of $\mathcal{ML}$, $\mathcal{ML}(⩒)$ and $\mathcal{ML}(⩒_{\mathcal{ML}})$). *Let Φ be a set of atomic propositions, and let I be an index set of modal operators. The syntax of $\mathcal{ML}$ is defined by the following grammar*

$$\varphi ::= p \mid \neg p \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \Diamond_j \varphi \mid \Box_j \varphi,$$

where $p \in \Phi$ and $j \in I$.

The syntax of $\mathcal{ML}(⩒)$ is obtained by extending the syntax of $\mathcal{ML}$ by the grammar rule

$$\varphi ::= (\varphi \mathbin{⩒} \varphi).$$

We denote by $\mathcal{ML}(⩒_{\mathcal{ML}})$ the fragment of $\mathcal{ML}(⩒)$ where the rule $(\varphi \mathbin{⩒} \varphi)$ is allowed only when the φ are $\mathcal{ML}$-formulas.

Next we define $\mathcal{MDL}$ which is the syntactical extension of $\mathcal{ML}$ by dependence atoms of the type $\mathrm{dep}(p_1, \ldots, p_n)$.

Definition 2 (Syntax of $\mathcal{MDL}$). *The syntax for $\mathcal{MDL}$ is obtained by extending the syntax of $\mathcal{ML}$ by dependence atoms*

$$\varphi ::= \mathrm{dep}(p_1, \ldots, p_n)\,,$$

where $p_1, \ldots, p_n \in \Phi$. As in Definition 1, Φ is a set of atomic propositions.

Now, we introduce $\mathcal{EMDL}$, which is the syntactical extension of $\mathcal{MDL}$ that allows arbitrary $\mathcal{ML}$-formulas inside dependence atoms instead of only atomic propositions.

Definition 3 (Syntax of $\mathcal{EMDL}$). *The syntax for $\mathcal{EMDL}$ is obtained by extending the syntax of $\mathcal{ML}$ by extended dependence atoms*

$$\varphi ::= \mathrm{dep}(\varphi_1, \ldots, \varphi_n)\,,$$

where $\varphi_1, \ldots, \varphi_n$ are $\mathcal{ML}$ formulas.

If φ is a formula in some logic, then $\mathrm{SF}(\varphi)$ is the set of all of its subformulas.

The syntactic inclusion structure of the logics defined here is described in Figure 1 (a).

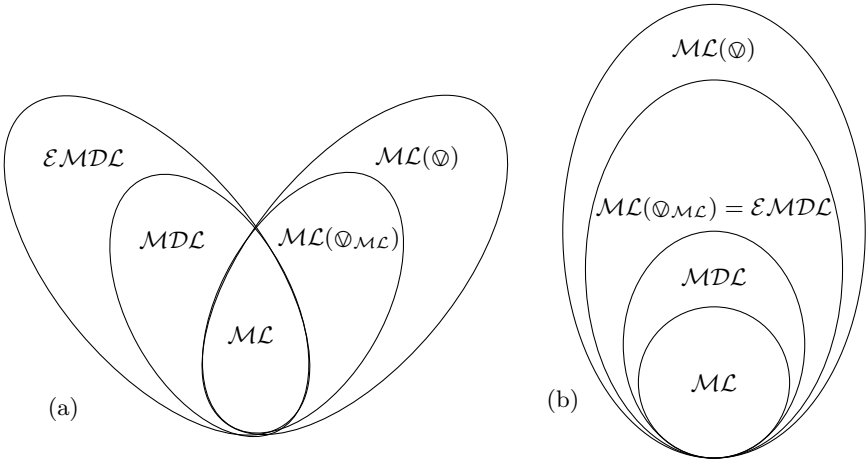


Fig. 1. (a) Syntactical inclusion structure, (b) Expressiveness inclusion structure

Similar as for $\mathcal{ML}$ formulas we evaluate $\mathcal{EMDL}$ formulas on Kripke structures. We will use multimodal Kripke structure here, i.e., Kripke structures with (possibly) several different accessibility relations. As explained in the introduction, this is useful in modelling temporal behavior of systems; see also Example 2.

Definition 4 (Kripke structure). *Let Φ be a set of atomic propositions. A Kripke structure (or frame) K over Φ is a tuple $K = (W, \mathbf{R}, V)$, where W is a set of* worlds*, $\mathbf{R} = \{R_1, \ldots, R_\ell\}$ is a set of different* accessibility relations*, and V is the* labeling function *$V\colon W \to \mathcal{P}(\Phi)$.*

Note that in contrast to usual modal logic, $\mathcal{EMDL}$ formulas are interpreted over a set of worlds (or a *team*) $T \subseteq W$. We write $K, T \models \varphi$ if the formula φ is satisfied by the team T in the Kripke structure K. We also make use of the shorthand notation $K, s \models \varphi$ for $K, \{s\} \models \varphi$.

Definition 5 (Semantics of $\mathcal{ML}$, $\mathcal{MDL}$, and $\mathcal{EMDL}$). *Let $K = (W, \mathbf{R}, V)$ be a Kripke model with $\mathbf{R} = \{R_1, \ldots, R_\ell\}$, $T \subseteq W$ be a team, and $\varphi, \psi \in \mathcal{EMDL}$, $\psi_1, \ldots, \psi_n$ be $\mathcal{ML}$ formulas. Then*

$K, T \models p$	*iff for all $w \in T$: $p \in V(w)$,*
$K, T \models \neg p$	*iff for all $w \in T$: $p \notin V(w)$,*
$K, T \models \varphi \vee \psi$	*iff $K, T_1 \models \varphi$ and $K, T_2 \models \psi$ for some $T_1, T_2 \subseteq T$ such that $T_1 \cup T_2 = T$,*
$K, T \models \varphi \mathbin{\mathbb{0}} \psi$	*iff $K, T \models \varphi$ or $K, T \models \psi$,*
$K, T \models \varphi \wedge \psi$	*iff $K, T \models \varphi$ and $K, T \models \psi$,*

$K, T \models \Diamond_i \varphi$ *iff* $K, T' \models \varphi$ *for some* $T' \subseteq W$ *s.t. for every* $w \in T$ *there exists a* $w' \in T'$ *with* wR_iw',
$K, T \models \Box_i \varphi$ *iff* $K, T' \models \varphi$ *for* $T' := \{w' \mid w \in T \wedge wR_iw'\}$,
$K, T \models \mathrm{dep}(\psi_1, \ldots, \psi_n)$ *iff* $\forall w_1, w_2 \in T$*:* $\bigwedge_{i=1}^{n-1}(K, \{w_1\} \models \psi_i \Leftrightarrow K, \{w_2\} \models \psi_i)$ *implies* $(K, \{w_1\} \models \psi_n \Leftrightarrow K, \{w_2\} \models \psi_n)$

For convenience reasons we will omit the index i at $\Diamond$ *or* $\Box$ *if it is clear in the context or if we only use one single type of transition relation.*

Note that $\mathrm{dep}(\psi)$ is a special case of dependence atom. By the last clause of Definition 5, $K, T \models \mathrm{dep}(\psi)$ iff the truth value of ψ is constant in the team T, i.e., either $K, \{w\} \models \psi$ for every $w \in T$, or $K, \{w\} \not\models \psi$ for every $w \in T$.

Satisfiability and Model Checking. We investigate two different computational problems: Satisfiability and model checking. In the satisfiability problem, we are given an arbitrary $\mathcal{EMDL}$ formula. The problem is then to decide, whether there exist a Kripke structure and a team satisfying the formula. For the model checking problem the input is an $\mathcal{EMDL}$ formula, a Kripke structure and a team on which the formula is evaluated. The problem is to decide, whether the given structure satisfies the formula w.r.t. the given team.

Definition 6 ($\mathcal{EMDL}$-SAT). *The satisfiability problem for* $\mathcal{EMDL}$ *is the decision problem over the set*

$$\mathcal{EMDL}\text{-SAT} := \left\{ \langle\varphi\rangle \;\middle|\; \begin{array}{l} \varphi \in \mathcal{EMDL} \text{ and there exists a Kripke structure } K = (W, \mathbf{R}, V), \text{ and a team } T \subseteq W, \text{ s.t.} \\ K, T \models \varphi \end{array} \right\}.$$

Definition 7 ($\mathcal{EMDL}$-MC). *The model checking problem for* $\mathcal{EMDL}$ *is the decision problem over the set*

$$\mathcal{EMDL}\text{-MC} := \left\{ \langle K, T, \varphi\rangle \;\middle|\; \begin{array}{l} \varphi \in \mathcal{EMDL},\ K = (W, \mathbf{R}, V) \text{ is a finite Kripke} \\ \text{structure, and } T \subseteq W, \text{ s.t. } K, T \models \varphi \end{array} \right\}.$$

Besides determining the computational complexity of the above two decision problems, the second contribution of our paper is a study the expressivity of $\mathcal{EMDL}$. To make this precise, we give the following definition.

Definition 8. *Let* $\mathcal{L}$ *be one of the above logics. We say that a formula* φ *is* expressible in $\mathcal{L}$ *if there exists an* $\mathcal{L}$*-formula* ψ *such that* ψ *is logically equivalent to* φ*. For logics* $\mathcal{L}$ *and* $\mathcal{L}'$*, we write* $\mathcal{L} \leq \mathcal{L}'$ *if every* $\mathcal{L}$*-formula* φ *is expressible in* $\mathcal{L}'$*. Furthermore, we write* $\mathcal{L} \equiv \mathcal{L}'$ *if* $\mathcal{L} \leq \mathcal{L}'$ *and* $\mathcal{L}' \leq \mathcal{L}$*.*

Example 1. (extending the motivating example given in the introduction) The following formula expresses that some property ψ of a computation depends on some outcome p when measuring two signals A and B on k measure points in the past:

$$\mathrm{dep}\big(\Diamond_{\mathrm{measA}} p, \Diamond_{\mathrm{measB}} p, \Diamond^2_{\mathrm{measA}} p, \Diamond^2_{\mathrm{measB}} p, \ldots, \Diamond^k_{\mathrm{measA}} p, \Diamond^k_{\mathrm{measB}} p, \Box_{\mathrm{comp}} \psi\big)\,.$$

Here $\Diamond_{\mathrm{measA}}, \Diamond_{\mathrm{measB}}$ refers to the (past oriented) measure relations for signals A and B, and $\Box_{\mathrm{comp}}$ refers to the (future oriented) computation relation.

Example 2. Cellular automata are a model of computation that has a wide variety of applications ranging from computability theory to physics and theoretical biology; see e.g. [1], [6]. Any two-dimensional cellular automaton with cell colors drawn from a set Φ can be presented as a Kripke structure

$$K = (W, P, L, R, U, D, V)$$

over atomic propositions Φ, where

- W is a set of triples (t, i, j) of integers (t represents a time step, (i, j) a cell);
- P is the (past oriented) successor relation on the first coordinate of W;
- L (R, U, D, resp.) are the the neighbor relations on the grid, i.e., the successor relation to the left (right, up, down, resp.) on the pair given by the second and third coordinate of W;
- $V((t, i, j)) \subseteq \Phi$ is the color of the cell (i, j) at time step t of the computation.

On the other hand, a Kripke structure of this form describes a cellular automaton only if the color of a cell (i, j) at time $t+1$ is completely determined by its own color and the color of its neighbors at time t. In case where $\Phi = \{q\}$ and only the immediate successors and predecessors are considered as neighbors, this can be expressed by the following formula φ_{CA} of $\mathcal{EMDL}$:

$$\mathrm{dep}(\langle P\rangle\langle L\rangle q, \langle P\rangle\langle R\rangle q, \langle P\rangle\langle U\rangle q, \langle P\rangle\langle D\rangle q, \langle P\rangle q, q)\,.$$

It is straightforward to modify the formula φ_{CA} for larger sets Φ of colors and different definitions of neighborhoods.

3 Expressivity of $\mathcal{EMDL}$

In this section we prove the strict inclusion $\mathcal{MDL} < \mathcal{EMDL}$. We will also show that $\mathcal{EMDL} \leq \mathcal{ML}(\mathbin{⫽})$. In fact, we will see that the expressive power of $\mathcal{EMDL}$ and the fragment $\mathcal{ML}(\mathbin{⫽}_{\mathcal{ML}})$ of $\mathcal{ML}(\mathbin{⫽})$ coincide. These results suggest that indeed $\mathcal{EMDL}$ is a natural extension of $\mathcal{MDL}$.

Proposition 1. *Let K be a Kripke model, T be a team of K, and φ be an $\mathcal{ML}$-formula.*

$$K, T \models \varphi \text{ iff } K, \{w\} \models \varphi \text{ for all } w \in T.$$

Proof. Follows from [9, Theorem 1].

Theorem 1. $\mathrm{dep}(\Diamond p)$ *is not expressible in $\mathcal{MDL}$.*

Proof. Let $\Phi = \{p\}$ and $K = (W, R, V)$ be a Kripke model over Φ such that $W = \{a, b\}$, $R = \{(b, b)\}$ and $V(a) = V(b) = \{p\}$. We will define a translation $\varphi \mapsto \varphi^*$ from $\mathcal{MDL}$-formulas to $\mathcal{ML}$-formulas and show that on this model each formula $\varphi \in \mathcal{MDL}$ is equivalent to the $\mathcal{ML}$ formula φ^*. We define that

$$(\psi \wedge \theta)^* := (\psi^* \wedge \theta^*),\ (\psi \vee \theta)^* := (\psi^* \vee \theta^*),\ (\Box\psi)^* := \Box\psi^*,$$
$$(\Diamond\psi)^* := \Diamond\psi^*,\ p^* := p,\ (\neg p)^* := \neg p,\ \mathrm{dep}(p, \ldots, p)^* := p.$$

Note that since $\Phi = \{p\}$ each dependence atom in φ is of type $\mathrm{dep}(p, \ldots, p)$, and

$$K, T \models \mathrm{dep}(p, \ldots, p)\,,$$

for all $T \subseteq W$. Hence it is a trivial inductive proof, that for all $T \subseteq W$ and $\varphi \in \mathcal{MDL}$

$$K, T \models \varphi \text{ iff } K, T \models \varphi^*. \tag{1}$$

We will now show that the $\mathcal{EMDL}$-formula $\mathrm{dep}(\Diamond p)$ is not expressible in $\mathcal{MDL}$. For contradiction, assume that there exists a $\mathcal{MDL}$-formula ψ that is equivalent to $\mathrm{dep}(\Diamond p)$. Clearly $K, \{a\} \models \mathrm{dep}(\Diamond p)$ and $K, \{b\} \models \mathrm{dep}(\Diamond p)$. Hence $K, \{a\} \models \psi$ and $K, \{b\} \models \psi$. Therefore, by (1) $K, \{a\} \models \psi^*$ and $K, \{b\} \models \psi^*$. Now since ψ^* is an $\mathcal{ML}$-formula by Proposition 1 we have that $K, \{a, b\} \models \psi^*$. Hence by (1) we have that $K, \{a, b\} \models \psi$ and finally that $K, \{a, b\} \models \mathrm{dep}(\Diamond p)$. This is a contradiction since clearly $K, \{a, b\} \not\models \mathrm{dep}(\Diamond p)$. □

We will next examine the expressivity of $\mathcal{EMDL}$ and show that it coincides with that of $\mathcal{ML}(\otimes_{\mathcal{ML}})$. We split the proof into two steps.

Theorem 2. $\mathcal{EMDL} \leq \mathcal{ML}(\otimes_{\mathcal{ML}})$.

Proof. We start by the observation of Väänänen [12, Section 8] that the basic dependence atoms $\mathrm{dep}(p_1, \ldots, p_{n+1})$ are definable by using the classical disjunction $\otimes$. The definition given in [12] works for our extended dependence atoms $\mathrm{dep}(\varphi_1, \ldots, \varphi_n, \vartheta)$ as well. Let $\delta := \delta(\varphi_1, \ldots, \varphi_n, \vartheta)$ be the formula

$$\bigvee_{\boldsymbol{b} \in \{\bot, \top\}^n} \Big(\varphi_1^{b_1} \wedge \ldots \wedge \varphi_n^{b_n} \wedge (\vartheta \otimes \vartheta^{\bot}) \Big), \tag{2}$$

where $\boldsymbol{b} = (b_1, \ldots, b_n)$, and $\varphi^{\top} := \varphi$ and $\varphi^{\bot}$ is the formula obtained from $\neg\varphi$ by pushing negations in front of atomic formulas. Now $K, T \models \delta$ if and only if T can be divided into subteams $T_{\boldsymbol{b}}$, $\boldsymbol{b} \in \{\bot, \top\}^n$, such that $T_{\boldsymbol{b}}$ contains exactly those states t that satisfy the formulas φ^{b_i} for $1 \leq i \leq n$. Moreover, since $K, T_{\boldsymbol{b}} \models \vartheta \otimes \vartheta^{\bot}$, the truth value of ϑ is constant in the team $T_{\boldsymbol{b}}$ for each $\boldsymbol{b} \in \{\bot, \top\}^n$. Clearly, this is equivalent with $K, T \models \mathrm{dep}(\varphi_1, \ldots, \varphi_n, \vartheta)$.

Using the equivalence above as a starting point, we will now define a translation $\varphi \mapsto \varphi^*$ from $\mathcal{EMDL}$ to $\mathcal{ML}(\otimes_{\mathcal{ML}})$ as follows:

$$\begin{aligned}
&\mathrm{dep}(\varphi_1, \ldots, \varphi_n, \vartheta)^* := \delta(\varphi_1^*, \ldots, \varphi_n^*, \vartheta^*)\\
&p^* := p,\ (\neg p)^* := \neg p,\ (\psi \wedge \theta)^* := (\psi^* \wedge \theta^*)\\
&(\psi \vee \theta)^* := (\psi^* \vee \theta^*),\ (\Box\psi)^* := \Box\psi^*,\ (\Diamond\psi)^* := \Diamond\psi^*.
\end{aligned}$$

Note that $\varphi^* = \varphi$ for every formula not containing dependence atoms. Thus, $\delta(\varphi_1^*, \ldots, \varphi_n^*, \vartheta^*)$ is in $\mathcal{ML}(\otimes_{\mathcal{ML}})$ for all $\mathcal{ML}$-formulas $\varphi_1, \ldots, \varphi_n, \vartheta$, and hence φ^* is in $\mathcal{ML}(\otimes_{\mathcal{ML}})$ for every $\mathcal{EMDL}$-formula φ.

It is straightforward to prove by induction that $K, T \models \varphi$ iff $K, T \models \varphi^*$ holds for every Kripke model K and every team T of K. □

Note that to represent the dependence atoms in $\mathcal{ML}(\oslash_{\mathcal{ML}})$ we needed an exponential size formula. In fact, the translation from $\mathcal{EMDL}$ into $\mathcal{ML}(\oslash_{\mathcal{ML}})$ cannot be made polynomial, unless the complexity classes PSPACE and NEXPTIME collapse. This follows from the fact that satisfiability for $\mathcal{EMDL}$ is NEXPTIME-complete (see Theorem 4 below), while satisfiability for $\mathcal{ML}(\oslash_{\mathcal{ML}})$ is PSPACE-complete, see [7, Corollary 3.3a].

Theorem 3. $\mathcal{ML}(\oslash_{\mathcal{ML}}) \leq \mathcal{EMDL}$

Proof. We will show that for every $\mathcal{ML}(\oslash_{\mathcal{ML}})$ formula φ there exists an $\mathcal{EMDL}$-formula φ^* such that for every Kripke model K and a team T of K it holds that

$$K, T \models \varphi \text{ iff } K, T \models \varphi^*. \tag{3}$$

We definite the translation $\varphi \mapsto \varphi^*$ recursively in the following manner:

1. $(\psi \wedge \theta)^* := (\psi^* \wedge \theta^*)$, $(\psi \vee \theta)^* := (\psi^* \vee \theta^*)$, $(\Box\psi)^* := \Box\psi^*$, $(\Diamond\psi)^* := \Diamond\psi^*$, $p^* := p$, $(\neg p)^* := \neg p$, and
2. $(\varphi \oslash \psi)^* := (\varphi^* \vee \psi^*) \wedge \mathrm{dep}((\varphi^* \wedge \neg\psi^*) \vee (\neg\varphi^* \wedge \psi^*), \psi^*)$.

The proof is done by induction on the structure of the formulas. The only interesting case is the case for $\oslash$. Assume first that $K, T \models (\varphi \oslash \psi)$, and that φ and ψ are $\mathcal{ML}$-formulas for which (3) holds. Hence $K, T \models \varphi$ or $K, T \models \psi$. Therefore $K, T \models \varphi^*$ or $K, T \models \psi^*$ and hence $K, T \models \varphi^* \vee \psi^*$. We still need to show that

$$K, T \models \mathrm{dep}((\varphi^* \wedge \neg\psi^*) \vee (\neg\varphi^* \wedge \psi^*), \psi^*)\,. \tag{4}$$

Since $K, T \models \varphi^*$ or $K, T \models \psi^*$ by Proposition 1 we have that $K, \{a\} \models \varphi^*$ for all $a \in T$ or $K, \{a\} \models \psi^*$ for all $a \in T$. If the latter holds then clearly (4) holds, hence we assume that the former holds. If

$$K, \{a\} \models (\varphi^* \wedge \neg\psi^*) \vee (\neg\varphi^* \wedge \psi^*)$$

for $a \in T$, then since $K, \{a\} \models \varphi^*$ we have that $K, \{a\} \models \neg\psi^*$. If on the other hand it holds that

$$K, \{a\} \not\models (\varphi^* \wedge \neg\psi^*) \vee (\neg\varphi^* \wedge \psi^*)$$

for some $a \in T$, then since $K, \{a\} \models \varphi^*$ we have that $K, \{a\} \models \psi^*$. Hence (4) holds and we have that

$$K, T \models (\varphi \oslash \psi)^*.$$

Assume then that $K, T \models (\varphi \oslash \psi)^*$. Hence

$$K, T \models (\varphi^* \vee \psi^*) \text{ and} \tag{5}$$

$$K, T \models \mathrm{dep}((\varphi^* \wedge \neg\psi^*) \vee (\neg\varphi^* \wedge \psi^*), \psi^*)\,. \tag{6}$$

Notice that by Proposition 1 it follows from (5) that $K, \{a\} \models \varphi^*$ or $K, \{a\} \models \psi^*$ for all $a \in T$. Hence if for $a \in T$ it holds that

$$K, \{a\} \not\models (\varphi^* \wedge \neg\psi^*) \vee (\neg\varphi^* \wedge \psi^*)$$

then $K, \{a\} \models (\varphi^* \wedge \psi^*)$ and hence $K, \{a\} \models (\varphi \wedge \psi)$. Now by Proposition 1, if we can show that for all $a \in T$ such that

$$K, \{a\} \models (\varphi^* \wedge \neg\psi^*) \vee (\neg\varphi^* \wedge \psi^*) \tag{7}$$

$K, \{a\} \models \varphi$ (or $K, \{a\} \models \psi$), we are done. Let $A \subseteq T$ be the set of points a from T such that (7) holds. Now since (6) we have that $K, \{a\} \models \psi^*$ for all $a \in A$ or $K, \{a\} \not\models \psi^*$ for all $a \in A$. And now since we had that $K, \{a\} \models \varphi^*$ or $K, \{a\} \models \psi^*$ for all $a \in T$, we conclude that $K, \{a\} \models \psi^*$ for all $a \in A$ or $K, \{a\} \models \varphi^*$ for all $a \in A$. □

The three results given in this section can now be summarized as follows:

Corollary 1. $\mathcal{ML} < \mathcal{MDL} < \mathcal{EMDL} \equiv \mathcal{ML}(@_{\mathcal{ML}}) \leq \mathcal{ML}(@)$

We leave it as an open question whether $\mathcal{EMDL} \equiv \mathcal{ML}(@)$.

4 Complexity of $\mathcal{EMDL}$

In this section we show that we can decide the satisfiability problem of $\mathcal{EMDL}$ in NEXPTIME and the model checking problem in NP. Hardness for these problems follows directly since these problems for $\mathcal{MDL}$ are NEXPTIME-complete and NP-complete, respectively [4, 7, 9].

Theorem 4. $\mathcal{EMDL}$-SAT *is* NEXPTIME*-complete.*

Proof. Let φ be an $\mathcal{EMDL}$ formula. Then each dep(·)-subformula of the form $\mathrm{dep}(\delta_1, \ldots, \delta_n)$ is substituted by

$$\mathrm{dep}(p_{\delta_1}, \ldots, p_{\delta_n}) \wedge \bigwedge_{1 \leq i \leq n} p_{\delta_i} \leftrightarrow \delta_i,$$

where $p_{\delta_1}, \ldots, p_{\delta_n}$ are fresh variables.

Claim. For every $\varphi \in \mathcal{EMDL}$ it holds that φ is equisatisfiable to φ', where φ' is the transformed formula from above.

Proof (of claim). Let $\varphi \in \mathcal{EMDL}$-SAT via the satisfying model $K = (W, R, V)$. Construct a new Kripke model $K' = (W, R, V')$ from K by labelling the newly added propositions. For all $\mathrm{dep}(\delta_1, \ldots, \delta_n) \in \mathrm{SF}(\varphi)$ we label with p_{δ_i} those worlds satisfying δ_i; more formally: for all $w \in W$

$$V'(w) = V(w) \cup \{p_{\delta_i} \mid K, w \models \delta_i \text{ and } \mathrm{dep}(\delta_1, \ldots, \delta_n) \in \mathrm{SF}(\varphi)\,, i \leq n\}.$$

Let $\mathrm{dep}(\delta_1, \ldots, \delta_n) \in \mathrm{SF}(\varphi)$ and $T \subseteq W$. By the above definition it holds for all $w, w' \in T$ that

$$\left(\begin{array}{c} K, w \models \delta_1 \Leftrightarrow K, w' \models \delta_1 \text{ and} \\ \vdots \\ K, w \models \delta_{n-1} \Leftrightarrow K, w' \models \delta_{n-1} \end{array}\right) \text{implies } K, w \models \delta_n \Leftrightarrow K, w' \models \delta_n$$

if and only if

$$\left(\begin{array}{c} K', w \models p_{\delta_1} \Leftrightarrow K', w' \models p_{\delta_1} \text{ and} \\ \vdots \\ K', w \models p_{\delta_{n-1}} \Leftrightarrow K', w' \models p_{\delta_{n-1}} \end{array}\right) \text{implies } K', w \models p_{\delta_n} \Leftrightarrow K', w' \models p_{\delta_n}.$$

Hence $K, T \models \mathrm{dep}(\delta_1, \ldots, \delta_n)$ if and only if $K', T \models \mathrm{dep}(p_{\delta_1}, \ldots, p_{\delta_n})$. Furthermore, since all δ_i are $\mathcal{ML}$-formulas we conclude by Proposition 1 that $K', T \models \bigwedge_{i=1}^{n}(\delta_i \leftrightarrow p_{\delta_i})$. Hence it is an easy inductive proof that K' is a satisfying model for φ'. For the other direction, it is easy to prove by a similar argument that any satisfying model for φ' is also a satisfying model for φ.

Observe that the above transformation is polynomial. This means that we can solve satisfiability for $\mathcal{EMDL}$ with the satisfiability algorithm for $\mathcal{MDL}$-SAT, giving us the upper bound NEXPTIME [7]. Since $\mathcal{MDL}$ is a sublogic of $\mathcal{EMDL}$, NEXPTIME-hardness for $\mathcal{EMDL}$-SAT follows from that of $\mathcal{MDL}$-SAT [7]. □

Theorem 5. $\mathcal{EMDL}$-MC *is* NP*-complete.*

Proof. The membership result is shown with an algorithm completely analogous to the one for $\mathcal{MDL}$-MC in [4]. In the following the check algorithm for the dependence atom is given, which will be the only part changed compared to the original $\mathcal{MDL}$-MC algorithm.

Algorithm 1. Algorithm deciding $\mathcal{EMDL}$-MC

```
Input: K, T, φ
if φ = dep(φ_1, ..., φ_n) then
    forall the w_1 ∈ T do
        forall the w_2 ∈ T do
            t_1 = (K, {w_1} ⊨ φ_1, ..., K, {w_1} ⊨ φ_{n-1});
            t_2 = (K, {w_2} ⊨ φ_1, ..., K, {w_2} ⊨ φ_{n-1});
            if t_1 = t_2 and K, {w_1} ⊨ φ_n ⇔ K, {w_2} ⊭ φ_n then reject
        end
    end
end
// Other cases like in the usual MDL-MC algorithm [4];
accept
```

Let $\mathrm{dep}(\varphi_1, \ldots, \varphi_n)$ be the dependence atom that has to be checked. Since all φ_i are $\mathcal{ML}$ formulas we can check $K, \{w\} \models \varphi_i$ in polynomial time, which together with $\mathcal{MDL}$-MC being in NP [4] shows the upper bound.

The lower bound follows from the NP-completeness of $\mathcal{MDL}$-MC shown in [4] and the fact that $\mathcal{MDL}$ is a subset of $\mathcal{EMDL}$. □

5 Conclusion

We introduced $\mathcal{EMDL}$ as a first approach towards a temporal dependence logic. $\mathcal{EMDL}$ is a multi-modal extension of $\mathcal{MDL}$ that allows to express dependences between arbitrary modal formulas. We showed that there are properties definable in $\mathcal{EMDL}$ that cannot be defined in $\mathcal{MDL}$, but on the other hand $\mathcal{EMDL}$ can be embedded into $\mathcal{ML}$ with classical disjunction $\veebar$. The translation, however, is exponential and cannot be made polynomial unless PSPACE = NEXPTIME. Unfortunately, we were only able to prove a partial converse, obtaining that the expressive power of $\mathcal{EMDL}$ coincides with that of $\mathcal{ML}$ with only restricted use of the classical disjunction operator $\veebar$, see Corollary 1. Thus, we leave the question, whether $\mathcal{EMDL} \equiv \mathcal{ML}(\veebar)$, as an open problem.

Somewhat surprising, given the strict chain of Corollary 1, is the second achievement of this paper: We showed that the computational complexity of satisfiability and model checking of $\mathcal{EMDL}$ is not higher than that of $\mathcal{MDL}$.

In a similar fashion as the basic temporal logic $\mathcal{TL}$ has been extended to $\mathcal{CTL}$ and $\mathcal{LTL}$, one might introduce temporal operators like "Until" and "Globally" into $\mathcal{EMDL}$. Further work has to be done here to reasonably represent paths of a Kripke model in modal dependence logic.

References

1. Berto, F., Tagliabue, J.: Cellular automata. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Summer 2012 edn. (2012)
2. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logics, Cambridge Tracts in Theoretical Computer Science, vol. 53. Cambridge University Press, Cambridge (2001)
3. Clarke, E.M., Emerson, E.A.: Desing and synthesis of synchronisation skeletons using branching time temporal logic. In: Kozen, D. (ed.) Logic of Programs 1981. LNCS, vol. 131, pp. 52–71. Springer, Heidelberg (1982)
4. Ebbing, J., Lohmann, P.: Complexity of model checking for modal dependence logic. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) SOFSEM 2012. LNCS, vol. 7147, pp. 226–237. Springer, Heidelberg (2012)
5. Hodges, W.: Some strange quantifiers. In: Mycielski, J., Rozenberg, G., Salomaa, A. (eds.) Structures in Logic and Computer Science. LNCS, vol. 1261, pp. 51–65. Springer, Heidelberg (1997)
6. Ilachinski, A.: Cellular Automata: A Discrete Universe. World Scientific, Singapore (2001)
7. Lohmann, P., Vollmer, H.: Complexity results for modal dependence logic. In: Dawar, A., Veith, H. (eds.) CSL 2010. LNCS, vol. 6247, pp. 411–425. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-15205-4_32
8. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley (1994)
9. Sevenster, M.: Model-theoretic and computational properties of modal dependence logic. Journal of Logic and Computation 19(6), 1157–1173 (2009), http://logcom.oxfordjournals.org/cgi/content/abstract/exn102v1

10. Sistla, A.P., Clarke, E.M.: The complexity of propositional linear temporal logics. J. ACM 32(3), 733–749 (1985)
11. Väänänen, J.: Dependence logic: A new approach to independence friendly logic. London Mathematical Society student texts, vol. 70. Cambridge University Press (2007)
12. Väänänen, J.: Modal dependence logic. In: Apt, K.R., van Rooij, R. (eds.) New Perspectives on Games and Interaction, Texts in Logic and Games, vol. 4, pp. 237–254. Amsterdam University Press (2008)

Dependence Logic with Generalized Quantifiers: Axiomatizations

Fredrik Engström[1], Juha Kontinen[2], and Jouko Väänänen[2,3]

[1] Department of Philosophy, Linguistics and Theory of Science, University of Gothenburg
[2] Department of Mathematics and Statistics, University of Helsinki, Finland
[3] Institute for Logic, Language and Computation, University of Amsterdam, The Netherlands

Abstract. We prove two completeness results, one for the extension of dependence logic by a monotone generalized quantifier Q with weak interpretation, weak in the sense that the interpretation of Q varies with the structures. The second result considers the extension of dependence logic where Q is interpreted as "there exist uncountably many." Both of the axiomatizations are shown to be sound and complete for $\mathrm{FO}(Q)$ consequences.

1 Introduction

Generalized quantifiers constitute a well-studied method of extending the expressive power of first order logic. A more recent extension of first order logic is obtained by adding dependence atoms, permitting the expression of partially ordered quantification. In this paper we study the combination of the two methods, adding to first order logic both generalized quantifiers Q and dependence atoms as defined in [1]. It was shown in [2] that the resulting logic, $\mathrm{D}(Q)$, properly extends both $\mathrm{FO}(Q)$ and dependence logic. We analyse further the expressive power and give natural axioms for the new logic. There are theoretical limits to the extent that the axioms can be complete but we give partial completeness results in the sense that completeness is shown with respect to $\mathrm{FO}(Q)$ consequences.

Generalized quantifiers were introduced by Mostowski [3]. The most important of them is perhaps the quantifier

$$\mathbb{M} \vDash Q_1 x \phi(x, \bar{b}) \iff \mathbb{M} \vDash \phi(a, \bar{b}) \text{ for uncountably many } a \in M$$

owing to the beautiful axiomatization of it by Keisler [4]. On the other hand, generalized quantifiers have made an entrance to both linguistics [5] and computer science [6]. In natural language we can use generalized quantifiers to analyse constructs such as

Most boys run,

where we think of "most" as a generalized quantifier. Other natural language quantifiers are "two thirds", "quite a few", "many", etc. There are various so-called polyadic lifts of such quantifiers, for example,

Ramsey lift:

At least two thirds of the boys in your class like each other. [7]

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 138–152, 2013.

Branching lift:

A few boys in my class and a few girls in your class have dated each other. [8]

Resumption lift:

Most neighbours like each other. [7]

In computer science, or more exactly finite model theory, we have the *counting quantifiers*

$$\mathbb{M} \vDash \exists_{\geq k} x \phi(x, \bar{b}) \iff \mathbb{M} \vDash \phi(a, \bar{b}) \text{ for at least } k \text{ elements } a \in M$$

which, although first order definable, have turn out relevant, but also the non-first order

$$\mathbb{M} \vDash \exists_{\text{even}} x \phi(x, \bar{b}) \iff \mathbb{M} \vDash \phi(a, \bar{b}) \text{ for an even number of } a \in M$$

and other similar ones. The lifts, also called vectorizations, are important in finite model theory, too. For example, the resumption lift sequence of the transitive closure quantifiers characterises in ordered models NLOGSPACE [9], and the resumption lift sequence of the so-called alternating transitive closure quantifier characterises, even in unordered models, least fixpoint logic [10].

Dependence logic was introduced in [11]. It gives compositional semantics to the partially ordered quantifiers of [12]:

$$\begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} \phi(x, y, u, v, \bar{z}) \iff \exists f \exists g \forall x \forall u \phi(x, f(x), u, g(u), \bar{z}).$$

The compositional analysis is

$$\begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} \phi(x, y, u, v, \bar{z}) \iff \forall x \exists y \forall u \exists v (=(\bar{z}, u, v) \wedge \phi(x, y, u, v, \bar{z})),$$

where $=(\bar{z}, u, v)$ is a so-called *dependence atom*, see section 2.1 for its semantics. Dependence logic has the same expressive power as existential second order logic [13]. Thus dependence logic alone cannot express, for example, uncountability, in fact not even finiteness.

The idea of combining partially ordered quantifiers and generalized quantifiers was first suggested by Barwise [8]. He used this combination to analyse lifts such as the Ramsey lift and the branching lift. It was proved in [7] that the polyadic lifts of monotone (unbounded) generalized quantifiers lead to a strong hierarchy, giving immediately the result that there is no finite number of generalized quantifiers, including partially ordered quantifiers, which would be able to express all Ramsey lifts of a given monotone (unbounded) quantifier. The same is true of branching lifts, and to a lesser extent of resumption lifts [7].

The situation is quite different with the extension of *dependence* logic (rather than *first order* logic) by a monotone generalized quantifier. All the mentioned polyadic lifts (and vectorizations) can be readily defined (in all arities). Let us see how this is done for the Ramsey lift of a monotone quantifier Q.

$$\exists A{\in}Q\ \forall x{\in}A\ \forall y{\in}A\ \phi(x,y,\bar{z})$$

$$\Longleftrightarrow$$

$$\begin{aligned}\exists w(=(\bar{z},w) \wedge Qx\exists y(\ & y=w \wedge \\ & \forall u\exists v(=(\bar{z},u,v) \wedge (x=u \to v=w) \wedge \\ & \forall u'\exists v'(=(\bar{z},u',v') \wedge (u=u' \to v'=w) \wedge \\ & ((v=w \wedge v'=w) \to \phi(u,u',\bar{z}))))))\end{aligned}$$

Resumption and branching can be handled similarly.

Thus putting generalized quantifiers and dependence atoms together results in a powerful combination extending far beyond either generalized quantifiers alone or dependence atoms alone.

This paper is organised as follows. In Section 2 we review the basics on dependence logic and generalized quantifiers in the dependence logic context. In Section 3 we present a system of natural deduction for the extension $\mathrm{D}(Q,\check{Q})$ of dependence logic by a monotone generalized quantifier Q and its dual $\check{Q}$, and show that these rules are sound. Finally in Section 4 two completeness results for $\mathrm{FO}(Q)$ consequences are shown for $\mathrm{D}(Q,\check{Q})$. In the first result Q has the so-called weak interpretation, and in the second Q is interpreted as Q_1, that is, "there exist uncountably many."

2 Preliminaries

2.1 Dependence Logic

In this section we give a brief introduction to dependence logic. For a detailed account see [11].

The syntax of dependence logic extends the syntax of first order logic with new atomic formulas, the dependence atoms. There is one dependence atom for each arity. We write the atom expressing that the term t_n is uniquely determined by the values of the terms $t_1,\ldots,t_{n-1}$ as $=(t_1,\ldots,t_n)$. We consider formulas where negation can only appear in front of formulas without dependence atoms. For a vocabulary τ, $\mathrm{D}[\tau]$ denotes the set of τ-formulas of dependence logic. The set $\mathrm{FV}(\phi)$ of free variables of a formula ϕ is defined as in first order logic except that

$$\mathrm{FV}(=(t_1,\ldots,t_n)) = \mathrm{FV}(t_1) \cup \ldots \cup \mathrm{FV}(t_n).$$

To define a compositional semantics for dependence logic we use *sets of assignments*, called *teams*, instead of single assignments as in first order logic. An assignment is a function $s : V \to M$ where V is a finite set of variables and M is the universe under consideration. A team on M is a set of assignments for some fixed finite set of variables V. If $V=\emptyset$ there is only one assignment, the empty function $\emptyset$. Observe that the team of the empty assignment $\{\emptyset\}$ is different from the empty team $\emptyset$.

– Given an assignment $s : V \to M$ and $a \in M$ let $s[a/x] : V \cup \{x\} \to M$ be the assignment:

$$s[a/x] : y \mapsto \begin{cases} s(y) & \text{if } y \in V \setminus \{x\}, \text{ and} \\ a & \text{if } x = y. \end{cases}$$

– Let $X[M/y]$ be the team

$$\{ s[a/y] \mid s \in X, a \in M \},$$

– and whenever $f : X \to M$, let $X[f/y]$ denote the team

$$\{ s[f(s)/y] \mid s \in X \}.$$

The domain of a non-empty team X, denoted $\mathrm{dom}(X)$, is the set of variables V. The interpretation of the term t in the model $\mathbb{M}$ under the assignment s is denoted by $t^{\mathbb{M},s}$. We write $s(\bar{x})$ for the tuple obtained by pointwise application of s to the finite sequence $\bar{x}$ of variables.

The satisfaction relation for dependence logic $\mathbb{M}, X \models \phi$ is defined as follows. Below, the notation $\mathbb{M}, s \models \phi$ refers to the ordinary satisfaction relation of first order logic. We also assume that $\mathrm{FV}(\phi) \subseteq \mathrm{dom}(X)$.

1. For formulas ψ without dependence atoms: $\mathbb{M}, X \models \psi$ iff $\forall s \in X : \mathbb{M}, s \models \psi$.
2. $\mathbb{M}, X \models =(t_1, \ldots, t_{n+1})$ iff $\forall s, s' \in X : \bigwedge_{1 \leq i \leq n} t_i^{\mathbb{M},s} = t_i^{\mathbb{M},s'} \to t_{n+1}^{\mathbb{M},s} = t_{n+1}^{\mathbb{M},s'}$
3. $\mathbb{M}, X \models \phi \wedge \psi$ iff $\mathbb{M}, X \models \phi$ and $\mathbb{M}, X \models \psi$
4. $\mathbb{M}, X \models \phi \vee \psi$ iff there are Y and Z such that $X = Y \cup Z$, and $\mathbb{M}, Y \models \phi$ and $\mathbb{M}, Z \models \psi$
5. $\mathbb{M}, X \models \exists y \phi$ iff there is $f : X \to M$, such that $\mathbb{M}, X[f/y] \models \phi$
6. $\mathbb{M}, X \models \forall y \phi$ iff $\mathbb{M}, X[M/y] \models \phi$.

We define $\mathbb{M} \models \sigma$ for a sentence σ to hold if $\mathbb{M}, \{ \emptyset \} \models \sigma$. Let us make some easy remarks.

– Every formula is satisfied by the empty team.
– The satisfaction relation is downwards closed: If $\mathbb{M}, X \models \phi$ and $Y \subseteq X$ then $\mathbb{M}, Y \models \phi$.
– The satisfaction relation is local: $\mathbb{M}, X \models \phi$ iff $\mathbb{M}, Y \models \phi$ where

$$Y = \{ s \upharpoonright \mathrm{FV}(\phi) \mid s \in X \}.$$

The expressive power for sentences of dependence logic is the same as that of existential second order logic.

2.2 D(Q)

The notion of a generalized quantifier goes back to Mostowski [3] and Lindström [14]. In [1] semantics for generalized quantifiers in the framework of dependence logic was introduced. We will review the definitions below.

Let Q be a quantifier of type $\langle k \rangle$, meaning that Q is a class of τ-structures, where the signature τ has a single k-ary relation symbol. Also, assume that Q is monotone increasing, i.e., for every M and every $A \subseteq B \subseteq M^k$, if $A \in Q_M$ then also $B \in Q_M$, where $Q_M = \{ R \subseteq M^k \mid (M, R) \in Q \}$.

The formulas of dependence logic extended with a quantifier Q, $\mathrm{D}(Q)$, is built up from $\mathrm{FO}(Q)$-formulas and dependence atoms using the connectives $\wedge$ and $\vee$, and the

quantifier expressions $\exists x$, $\forall x$ and Qx in the usual way. We write $\phi \to \psi$ as a shorthand for $\neg\phi \vee \psi$, where ϕ is a formula without dependence atoms.

An assignment s satisfies a formula $Q\bar{x}\,\phi$ in a structure $\mathbb{M}$,

$$\mathbb{M}, s \vDash Q\bar{x}\,\phi, \text{ if the set } \{\, \bar{a} \in M^k \mid \mathbb{M}, s[\bar{a}/\bar{x}] \vDash \phi \,\} \text{ is in } Q_M.$$

In the context of teams we say that a team X satisfies a formula $Q\bar{x}\,\phi$,

$$\mathbb{M}, X \vDash Q\bar{x}\,\phi, \text{ if there exists } F : X \to Q_M \text{ such that } \mathbb{M}, X[F/\bar{x}] \vDash \phi, \qquad (1)$$

where $X[F/\bar{x}] = \{\, s[\bar{a}/\bar{x}] \mid \bar{a} \in F(s), s \in X \,\}$. This definition works well only with monotone (increasing) quantifiers, see [1] for details.

The following easy proposition suggests that we indeed have the right truth condition for monotone quantifiers:

Proposition 1 ([1])

(i) $\mathrm{D}(Q)$ *is downwards closed.*
(ii) $\mathrm{D}(Q)$ *is local, in the sense that* $\mathbb{M}, X \vDash \phi$ *iff* $\mathbb{M}, (X \upharpoonright FV(\phi)) \vDash \phi$.
(iii) Viewing $\exists$ *and* $\forall$ *as generalized quantifiers of type* $\langle 1 \rangle$*, the truth conditions in* (1) *are equivalent to the truth conditions of dependence logic.*
(iv) For every $\mathrm{D}(Q)$ *formula* ϕ *we have* $\mathbb{M}, \emptyset \vDash \phi$.

As proved in [2], the expressive power of $\mathrm{D}(Q)$ sentences corresponds to that of a certain natural extension of existential second order logic by Q.

In order to get a prenex normal form for all formulas we will focus on the logics $\mathrm{D}(Q, \check{Q})$, where $\check{Q}$ is the dual of Q, i.e,

$$\check{Q} = \{\, (M, M^k \setminus R) \mid R \subseteq M^k, (M, R) \notin Q \,\},$$

instead of $\mathrm{D}(Q)$. Note that, according to our definition of $\mathrm{D}(Q)$, a formula $Qx\phi$ may be negated only if ϕ is a $\mathrm{FO}(Q)$ formula.

We will consider monotone increasing quantifiers Q satisfying two non-triviality assumptions: $(M, \emptyset) \notin Q$ and $(M, M^k) \in Q$ for all M. In [2] the following normal form for sentences of $\mathrm{D}(Q)$ was shown for such non-trivial quantifiers. In the following we will use the notations $\bar{x}^i$ and $\bar{x}_i$ interchangeably.

Theorem 1. *Every* $\mathrm{D}(Q)$ *sentence in negation normal form, where* Q *is non-trivial, can be written as*

$$\mathcal{H}^1 \bar{x}_1 \ldots \mathcal{H}^m \bar{x}_m \exists y_1 \ldots \exists y_n \Big(\bigwedge_{1 \le i \le n} {=}(\bar{z}^i, y_i) \wedge \theta \Big),$$

where $\mathcal{H}^i$ *is either* Q *or* $\forall$ *and* θ *is a quantifer-free* FO*-formula.*

In the present paper a similar normal form for all $\mathrm{D}(Q, \check{Q})$ formulas is obtained in Proposition 3.

A *weak semantics* can be given for $\mathrm{D}(Q)$ (and $\mathrm{FO}(Q)$, etc) by regarding Q as an interpreted symbol rather than a logical constant in the following way (see [4] and [15]

for more on this). A weak model is a structure together with an interpretation of Q, often denoted by q. We define $T \models_w \sigma$ to hold if every weak model (M, q) of T satisfies σ. In this paper we require the interpretation q of Q to be monotone increasing and non-trivial (In essence this is the *monotone logic* of [16]). In the weak semantics for $\mathrm{D}(Q, \check{Q})$ we require that the interpretation $\check{q}$ of $\check{Q}$ is the dual of the interpretation q of Q. Thus, if $T \cup \{\, \sigma \,\}$ consists of $\mathrm{D}(Q, \check{Q})$ sentences, then $T \models_w \sigma$ if every model $(M, q, \check{q})$ of T satisfies σ.

3 Natural Deduction for $\mathrm{D}(Q, \check{Q})$

In this section we present a set of natural deduction rules for the logic $\mathrm{D}(Q, \check{Q})$, where Q is monotone and satisfies the non-triviality conditions: $(M, \emptyset) \notin Q$ and $(M, M^k) \in Q$ for all M. Observe that then also the dual quantifier $\check{Q}$ satisfies these conditions. To simplify notation, we will restrict attention to type $\langle 1 \rangle$ quantifiers.

We use an abbreviation $\bar{x} = \bar{y}$ for the formula $\bigwedge_{1 \le i \le \mathrm{len}(\bar{x})} x_i = y_i$, assuming of course that $\bar{x}$ and $\bar{y}$ are tuples of the same length $\mathrm{len}(\bar{x})$. The substitution of a term t to the free occurrences of x in ψ is denoted by $\psi[t/x]$. Analogously to first order logic, no variable of t may become bound in such a substitution. For tuples $\bar{t} = (t_1, \ldots, t_n)$ and $\bar{x} = (x_1, \ldots, x_n)$ we write $\psi[\bar{t}/\bar{x}]$ to denote the simultaneous substitution of x_i by t_i for $1 \le i \le n$.

We use the standard first order introduction and elimination rules for conjunction, existential quantifier, and universal quantifier. We also adopt rules for commutativity and associativity of disjunction, and the usual identity axioms for $\mathrm{FO}(Q, \check{Q})$ formulas. The rest of the rules are listed below:

1. Disjunction:

$$\frac{\phi}{\phi \vee \psi}\ \vee\mathrm{I} \qquad \frac{\psi}{\phi \vee \psi}\ \vee\mathrm{I} \qquad \frac{\phi \vee \psi \quad \begin{array}{c}[\phi]\\ \vdots\\ \gamma\end{array} \quad \begin{array}{c}[\psi]\\ \vdots\\ \gamma\end{array}}{\gamma}\ \vee\mathrm{E}$$

   where γ is a $\mathrm{FO}(Q, \check{Q})$ formula.
2. Negation and duality:

$$\frac{\begin{array}{c}[\phi]\\ \vdots\\ \bot\end{array}}{\neg\phi}\ \neg\mathrm{I} \qquad \frac{\begin{array}{c}[\neg\phi]\\ \vdots\\ \bot\end{array}}{\phi}\ \mathrm{RAA} \qquad \frac{\phi \quad \neg\phi}{\bot}\ \bot\mathrm{I} \qquad \frac{\check{Q}x\phi}{\neg Qx\neg\phi}$$

   where ϕ is a $\mathrm{FO}(Q, \check{Q})$ formula.
3. Disjunction substitution:

$$\frac{\phi \vee \psi \quad \begin{array}{c}[\psi]\\ \vdots\\ \gamma\end{array}}{\phi \vee \gamma}$$

4. Extending scope:
$$\frac{\mathcal{H}x\phi \vee \psi}{\mathcal{H}x(\phi \vee \psi)} \qquad \frac{Qx\phi \wedge \psi}{Qx(\phi \wedge \psi)}$$
where $\mathcal{H} \in \{Q, \check{Q}, \exists, \forall\}$, and the prerequisite for applying these rules is that x does not appear free in ψ. The rule on the right is also assumed for $\check{Q}$.
5. Unnesting:
$$\frac{=(t_1, ..., t_n)}{\exists z(=(t_1, ..., z, ..., t_n) \wedge z = t_i)}$$
where z is a new variable.
6. Dependence distribution: let
$$\phi = \exists y_1 \dots \exists y_n(\bigwedge_{1 \le j \le n} =(\bar{z}^j, y_j) \wedge \phi_0),$$
$$\psi = \exists y_{n+1} \dots \exists y_m(\bigwedge_{n+1 \le j \le m} =(\bar{z}^j, y_j) \wedge \psi_0).$$
where ϕ_0 and ψ_0 are quantifier-free formulas without dependence atoms, and y_i, for $1 \le i \le n$, does not appear in ψ and y_i, for $n+1 \le i \le m$, does not appear in ϕ. Then,
$$\frac{\phi \vee \psi}{\exists y_1 \dots \exists y_m(\bigwedge_{1 \le j \le m} =(\bar{z}^j, y_j) \wedge (\phi_0 \vee \psi_0))}$$
7. Dependence introduction:
$$\frac{\exists x \forall y \phi}{\forall y \exists x(=(\bar{z}, x) \wedge \phi)} \qquad \frac{\exists x Q y \phi}{Q y \exists x(=(\bar{z}, x) \wedge \phi)}$$
where $\bar{z}$ lists the variables in $\mathrm{FV}(\phi) - \{x, y\}$. Similar for $\check{Q}$.
8. Monotonicity of Q and $\check{Q}$:
$$\frac{Qx\phi \qquad \begin{array}{c}[\phi]\\ \vdots \\ \psi\end{array}}{Qx\psi}$$
where the prerequisite for applying this rule is that the variable x cannot appear free in any non-discharged assumption used in the derivation of ψ, except for ϕ. Similar for $\check{Q}$.
9. Bound variables:
$$\frac{Qx\phi}{Qy\phi[y/x]},$$
where y does not appear in ϕ. Similar for $\check{Q}$.

Observe that $\mathrm{FO}(Q, \check{Q}) \equiv \mathrm{FO}(Q)$, but syntactically $\mathrm{FO}(Q, \check{Q})$ includes more formulas.

3.1 Soundness of the Rules

In this section we show the soundness of the rules introduced in the previous section under any monotone and non-trivial interpretation of Q. Clearly this is the same as soundness in the weak semantics for Q.

The following lemmas will be needed in the proof.

Lemma 1. *Let $\phi(x)$ be a $\mathrm{D}(Q, \check{Q})$ formula, and t a term such that in the substitution $\phi(t/x)$ no variable of t becomes bound. Then for all $\mathbb{M}$ and teams X, where $(FV(\phi) - \{x\}) \cup \mathrm{Var}(t) \subseteq \mathrm{dom}(X)$*

$$\mathbb{M}, X \vDash \phi(t/x) \Leftrightarrow \mathbb{M}, X(F/x) \vDash \phi(x),$$

where $F\colon X \to A$ is defined by $F(s) = t^{\mathbb{M}}\langle s \rangle$.

Proof. Analogous to Lemma 8 in [17].

It is easy to verify that Lemma 1 gives the following familiar property concerning changing free variables.

Lemma 2 (Change of free variables). *Let the free variables of $\phi \in \mathrm{D}(Q, \check{Q})$ be $x_1, \ldots, x_n$ and let $y_1, \ldots, y_n$ be distinct variables. Then for all structures $\mathbb{M}$ and teams X with domain $\{x_1, \ldots, x_n\}$ it holds that*

$$\mathbb{M}, X \vDash \phi \Leftrightarrow \mathbb{M}, X' \vDash \phi(\bar{y}/\bar{x}),$$

where X' is the team with domain $\{y_1, \ldots, y_n\}$ containing the assignments $s'\colon y_i \mapsto s(x_i)$ for $s \in X$.

Proposition 2. *Assume that Q is monotone and non-trivial. Let $T \cup \{\phi\}$ be a set of sentences of $\mathrm{D}(Q, \check{Q})$. If $T \vdash \phi$, then $T \vDash \phi$.*

Proof. We prove the statement that if $T \vdash \phi$, where $T \cup \{\,\phi\,\}$ is a set of *formulas*, then for any $\mathbb{M}$ and X where $\mathrm{dom}(X) \supseteq \mathrm{FV}(T) \cup \mathrm{FV}(\phi)$, if $\mathbb{M}, X \vDash T$ then $\mathbb{M}, X \vDash \phi$. This is done by using induction on the length of derivation.

It suffices to consider the rules 2 (duality), 4, 7, 8, and 9 since the soundness of the other rules can be proved analogously to [17] using the fact that $\mathrm{D}(Q, \check{Q})$ is local and has downwards closure (see (ii) and (i) of Proposition 1). In particular, Lemma 1 is used in the soundness proofs of the rules $\exists$ I and $\forall$ E.

2. Assume $\mathbb{M}, X \vDash \check{Q}x\phi$ then, since $\check{Q}x\phi$ is a $\mathrm{FO}(Q, \check{Q})$ formula we have $\mathbb{M}, s \vDash \check{Q}x\phi$ for all $s \in X$. This clearly implies that $\mathbb{M}, s \vDash \neg Qx\neg\phi$ for all $s \in X$, which is equivalent to $\mathbb{M}, X \vDash \neg Qx\neg\phi$.
4. These two rules preserve logical equivalence analogously to Lemma 3.2 in [2].
7. The soundness of this rule follows from the logical equivalence

$$Qy\exists x(=(\bar{z}, x) \wedge \phi) \equiv \exists x Qy\phi$$

the proof of which is analogous to the case where Q is replaced by $\forall$ (see [17]).

8. Assume that we have a natural deduction proof of $Qx\psi$ from the assumptions

$$\{\gamma_1, \ldots, \gamma_k\}$$

with the last rule 13. Let $\mathbb{M}$ and X be such that $\mathbb{M}, X \models \phi_i$, for $1 \leq i \leq k$. By the assumption, we have a shorter deduction of $Qx\phi$ from the assumptions $\{\gamma_{n_1}, \ldots, \gamma_{n_l}\}$ and a deduction of ψ from the assumptions $\{\phi, \gamma_{n_{l+1}}, \ldots, \gamma_{n_m}\}$. Hence by the induction assumption it holds that $\mathbb{M}, X \models Qx\phi$. Therefore, there is $F\colon X \to Q_M$ such that $\mathbb{M}, X[F/x] \models \phi$. Since the variable x cannot appear free in the formulas $\gamma_{n_{l+1}}, \ldots, \gamma_{n_m}$ it follows that $\mathbb{M}, X[F/x] \models \gamma_i$, for $i \in \{n_{l+1}, \ldots, n_m\}$. Now by the induction assumption we get that $\mathbb{M}, X[F/x] \models \psi$ and $\mathbb{M}, X \models Qx\psi$.
9. This rule preserves logical equivalence by Lemma 2. □

Note that since Proposition 2 holds for every monotone non-trivial quantifier Q we get also soundness for weak semantics: If $T \vdash \phi$ then $T \models_w \phi$.

4 Completeness Results for $\mathrm{FO}(Q, \check{Q})$ Consequences

4.1 Deriving a Normal Form for $\mathrm{D}(Q, \check{Q})$

In this section we show that from each formula $\phi \in \mathrm{D}(Q, \check{Q})$ we can derive a logically equivalent formula in the following normal form:

$$\mathcal{H}^1 x_1 \ldots \mathcal{H}^m x_m \exists y_1 \ldots \exists y_n \big(\bigwedge_{1 \leq i \leq n} {=}(\bar{x}^i, y_i) \wedge \theta \big), \tag{2}$$

where $\mathcal{H}^i$ is either Q, $\check{Q}$ or $\forall$, and θ is a quantifier-free FO-formula.

Proposition 3. *Let ϕ be a formula of $\mathrm{D}(Q, \check{Q})$. Then $\phi \vdash \phi'$, where ϕ' is of the form* (2)*, and ϕ' is logically equivalent to ϕ.*

Proof. The proof of this Proposition is analogous to the proof of the corresponding result for dependence logic formulas in [17]. See the full version of this article [18]. □

4.2 Completeness for $\mathrm{D}(Q, \check{Q})$

In this section we prove a completeness result for $\mathrm{D}(Q, \check{Q})$ with respect to $\mathrm{FO}(Q, \check{Q})$ consequences of $\mathrm{D}(Q, \check{Q})$-sentences, with weak semantics. Analogously to [17], we approximate $\mathrm{D}(Q, \check{Q})$-sentences in the normal form (2) by an infinite set of $\mathrm{FO}(Q, \check{Q})$ sentences. We use an extra predicate R to encode a team witnessing the satisfiability of the quantifier prefix $\mathcal{H}^1 x_1 \ldots \mathcal{H}^m x_m$.

Let σ be

$$\mathcal{H}^1 x_1 \ldots \mathcal{H}^m x_m \exists y_1 \ldots \exists y_n \big(\bigwedge_{1 \leq i \leq n} {=}(\bar{x}^i, y_i) \wedge \theta(x_1, \ldots, x_m, y_1, \ldots, y_n) \big),$$

where each $\mathcal{H}^i$ is either Q, $\check{Q}$ or $\forall$.

We define finite approximations $A^i\sigma$ of σ as follows. The first approximation, $A^1\sigma$, is

$$\forall\bar{x}\exists\bar{y}\big(R(\bar{x}) \to \theta(\bar{x},\bar{y})\big).$$

The second approximation $A^2\sigma$ is

$$\forall\bar{x}_1\exists\bar{y}_1\forall\bar{x}_2\exists y_2\big(R(\bar{x}_1)\wedge R(\bar{x}_2)\to\theta(\bar{x}_1,\bar{y}_1)\wedge\theta(\bar{x}_2,\bar{y}_2)\wedge \bigwedge_{1<i<n}(\bar{x}_1^i=\bar{x}_2^i\to y_{i,1}=y_{i,2})\big)$$

With the notational convention that $(x_{i_1},\dots,x_{i_k})_j$ is the sequence $(x_{i_1,j},\dots,x_{i_k,j})$. By generalizing this construction we get the k-th approximation:

$$\forall\bar{x}_1\exists\bar{y}_1\dots\forall\bar{x}_k\exists\bar{y}_k\big(\bigwedge_{1\le j\le k}R(\bar{x}_j)\to\bigwedge_{1\le j\le k}\theta(\bar{x}_j,\bar{y}_j)\wedge \bigwedge_{\substack{1\le i\le n\\ 1\le j,j'\le k}}(\bar{x}_j^i=\bar{x}_{j'}^i\to y_{i,j}=y_{i,j'})\big)$$

Also we need a sentence saying that R is of the right kind, witnessing the quantifier prefix: Let $B\sigma$ be

$$\mathcal{H}^1x_1\dots\mathcal{H}^mx_mR(x_1,\dots,x_m).$$

We will adopt the following rule in our deduction system:

(10) Approximation rule:

$$\frac{\begin{array}{cc}[B\sigma] & [A^n\sigma]\\ \ddots & \iddots\\ \sigma & \psi\end{array}}{\psi}\ \text{(Approx)}$$

where σ is a sentence in normal form, and R does not appear in ψ nor in any uncancelled assumptions in the derivation of ψ, except for $B\sigma$ and $A^n\sigma$.

Proposition 4. *Adding the approximation rule to the inference system results in a sound system for* $\mathrm{D}(Q,\check{Q})$ *with regard to weak semantics.*

Proof. We plug in the following induction step to the proof of Proposition 2:

Assume that there is a derivation of ψ from Γ ending with the approximation rule. Then there are shorter derivations from Γ of σ and from $\Gamma', B\sigma, A^n\sigma$ of ψ, where $\Gamma'\subseteq\Gamma$ is such that R does not occur in Γ'. By the induction hypothesis we get $\Gamma\vDash\sigma$ and $\Gamma', B\sigma, A^n\sigma\vDash\psi$. We will prove that $\Gamma\vDash\psi$, by assuming $\mathbb{M},X\vDash\Gamma$ for some non-empty X and proving that $\mathbb{M},X\vDash\psi$.

Assume σ is of the form

$$\mathcal{H}^1x_1\dots\mathcal{H}^mx_m\exists y_1\dots\exists y_n\big(\bigwedge_{1\le i\le n}{=}(\bar{x}^i,y_i)\wedge\theta\big).$$

where θ is a quantifier free first order formula.

From the fact that $\mathbb{M}, X \models \sigma$ we get $\mathbb{M} \models \sigma$ and thus there is a (non-empty) team Y such that

$$\mathbb{M}, Y \models \exists y_1 \dots \exists y_n \big(\bigwedge_{1 \le i \le n} {=}(\bar{x}^i, y_i) \wedge \theta \big).$$

Let $r \subseteq M^m$ be the relation $Y(\bar{x}) = \{ s(\bar{x}) \mid s \in X \}$ corresponding to Y. Then $(\mathbb{M}, r) \models B\sigma$ and, it should also be clear that $(\mathbb{M}, r) \models A^n\sigma$. Since R does not occur in Γ' we have $(\mathbb{M}, r), X \models \Gamma', B\sigma, A^n\sigma$. By the induction hypothesis $(\mathbb{M}, r), X \models \psi$, and since R does not occur in ψ we have $\mathbb{M}, X \models \psi$. □

Lemma 3. *If T is a set of $\mathrm{FO}(Q, \check{Q})$-sentences consistent in the deduction system described above then there are a countable recursively saturated model $\mathbb{M}$ and an interpretation q of Q such that $(\mathbb{M}, q, \check{q}) \models T$.*

Proof. First translate T to $T^\neg$ in which each $\check{Q}x\phi$ is replaced by $\neg Qx\neg\phi$. By using the same argument as in [4,15] we may reduce $\mathrm{FO}(Q)$ to FO by replacing subformulas of the form $Qx\phi$ with new relation symbols $R_\phi(\bar{y})$, $\bar{y}$ being the free variables of $Qx\phi$. This will reduce the set $T^\neg$ to a set T^*. Let T' be T^* together with the translations of the universal closures of

- $(\phi \to \psi) \to (Qx\phi \to Qx\psi)$, for all ϕ and ψ; and
- $Qx\phi \to Qy(\phi[y/x])$, for all ϕ such that the substitution is legal.

Now T' is consistent by the same argument as in [15]. Let $\mathbb{M}^*$ be a countable recursively saturated model of T', and M its reduct to the original signature. Now we may define q to be

$$\{ A \subseteq M \mid A \supseteq \{ a \in M \mid \mathbb{M}^*, s[a/x] \models \phi^* \} \text{ for some } \phi \text{ s.t. } M^*, s \models Qx\phi^* \}.$$

Proposition 2.3.4 in [15] shows that $(\mathbb{M}, q) \models T^\neg$, and thus $(\mathbb{M}, q, \check{q}) \models T$. □

The main result of this section can now be stated as follows.

Theorem 2. *Let T be a set of sentences of $\mathrm{D}(Q, \check{Q})$ and $\phi \in \mathrm{FO}(Q, \check{Q})$ a sentence. Then the following are equivalent:*

(I) $T \models_w \phi$
(II) $T \vdash \phi$

Lemma 4. *In a countable recursively saturated weak model $(\mathbb{M}, q, \check{q})$ in which $B\sigma$ and $A^n\sigma$ holds for all n, σ holds.*

Proof. Suppose σ is

$$\mathcal{H}^1 x_1 \dots \mathcal{H}^m x_m \exists y_1 \dots \exists y_n \big(\bigwedge_{1 \le i \le n} {=}(\bar{x}^i, y_i) \wedge \theta \big).$$

Note that the sentences $A^n\sigma$ can be viewed as the finite approximations as defined in [17] (and see also [19]) of the D sentence σ':

$$\forall \bar{x} \exists \bar{y} \big(\bigwedge_{1 \le i \le n} {=}(\bar{x}^i, y_i) \wedge (R(\bar{x}) \to \theta) \big).$$

Thus by Theorem 2.4 in [19] (see also [17]), we know that $\mathbb{M} \vDash \sigma'$.

Let X be the team $\{ s : \{ x_1, \ldots, x_k \} \to M \mid (s(x_1), \ldots, s(x_m)) \in R^{\mathbb{M}} \}$. To prove that $(\mathbb{M}, q, \check{q}) \vDash \sigma$ we find $F_1, \ldots, F_m$ so that $\{ \emptyset \}[F_1/x_1] \ldots [F_m/x_m] = X$.

$$F_1(\emptyset) = X \upharpoonright \{ x_1 \},$$
$$F_{i+1} : \{ \emptyset \}[F_1/x_1] \ldots [F_i/x_i] \to M$$
$$F_{i+1}(s) = \{ a \in M \mid \exists s' \in X : (s'(x_1), \ldots, s'(x_{i+1})) = (s(x_1), \ldots, s(x_i), a) \}.$$

By the assumption $(\mathbb{M}, q, \check{q}) \vDash B\sigma$ it follows that $F_i(s) \in \mathcal{H}^i_M$. Furthermore, since $\mathbb{M} \vDash \sigma'$ we get that

$$\mathbb{M}, X \vDash \exists y_1 \ldots \exists y_n \big(\bigwedge_{1 \le i \le n} {=}(\bar{x}^i, y_i) \wedge \theta \big).$$

Therefore

$$(\mathbb{M}, q, \check{q}) \vDash \mathcal{H}^1 x_1 \ldots \mathcal{H}^m x_m \exists y_1 \ldots \exists y_n \big(\bigwedge_{1 \le i \le n} {=}(\bar{x}^i, y_i) \wedge \theta \big)$$

as wanted. □

Proof (Proof of Theorem 2.). (I) ⇒ (II): This is just a special case (for sentences) of soundness.

(II) ⇒ (I): Suppose $T \nvdash \phi$, where ϕ is a $\mathrm{FO}(Q, \check{Q})$-sentence. We will construct a weak model of $T \cup \{ \neg\phi \}$ showing that $T \nvDash_w \phi$. Replacing T with the set $T' = \{ B\sigma, A^n\sigma \mid \sigma \in T, n \in \mathbb{N} \}$ we can conclude that $T' \cup \{ \neg\phi \} \nvdash \bot$. By applying Lemma 3 we get a weak countable recursively saturated model $(M, q, \check{q})$ of T'. Lemma 4 implies that $(M, q, \check{q}) \vDash T$. Now since $(M, q, \check{q}) \nvDash \phi$, we get $T \nvDash_w \phi$ as wanted. □

4.3 Completeness for $\mathrm{D}(Q_1, \check{Q}_1)$

We will now prove a completeness result similar to Theorem 2 for the logic $\mathrm{D}(Q, \check{Q})$ where Q is interpretated as Q_1, the quantifier "there exist uncountably many." In this section we consider only structures over uncountable universes.

We add the following two rules from [4] to the system presented in Section 3. Note that the approximation rule of section 4.2 is not included.

$$\frac{}{\neg Qx(x = y \vee x = z)}$$

$$\frac{Qx\exists y\phi}{\exists yQx\phi \vee Qy\exists x\phi}$$

The intuitive meaning of the second rule is that a countable union of countable sets is countable. The first is needed to avoid Q being interpreted as the quantifier "the exists at least two."

For each $\mathrm{D}(Q, \check{Q})$ sentence σ

$$\mathcal{H}^1 x_1 \ldots \mathcal{H}^m x_m \exists y_1 \ldots \exists y_n \big(\bigwedge_{1 \le i \le n} {=}(\bar{x}^i, y_i) \wedge \theta \big)$$

in normal form we define the Skolem translation $S\sigma$ of σ to be:

$$\mathcal{H}^1 x_1 \dots \mathcal{H}^m x_m \theta(f_i(\bar{x}^i)/y_i),$$

where the f_i's are new function symbols of the right arity. If σ is a sentence in the signature τ then $S\sigma$ will be in the extended signature $\tau \cup \{ f_1, \dots, f_n \}$.

The last rule of the deduction system is the following:

$$\frac{\begin{array}{c}[S\sigma] \\ \vdots \\ \sigma \qquad \psi\end{array}}{\psi}\ \text{(Skolem)}$$

Here σ is a $\mathrm{D}(Q, \check{Q})$ sentence in normal form, and the function symbols $f_1, \dots, f_n$ do not occur in ψ nor in any uncancelled assumption of the derivation of ψ, except for $S\sigma$.

Proposition 5. *If $T \vdash \phi$ in the deduction system for $\mathrm{D}(Q_1, \check{Q}_1)$ then $T \models \phi$.*

Proof. We extend the proof of Proposition 2 to also cover the three new rules:

(1) The soundness of the first rule is easily seen by observing that the formula $Q_1 x(x = y \vee x = z)$ is a $\mathrm{FO}(Q_1)$ formula and thus a team satisfies it iff every assignment in the team satisfies the formula.

(2) For the second rule we need to prove that if $\mathbb{M}, X \models \Gamma, Q_1 x \exists y \phi$ then $\mathbb{M}, X \models \Gamma, \exists y Q_1 x \phi \vee Q_1 y \exists x \phi$. By the assumption we get functions $F : X \to Q_M$ and $f : X[F/x] \to M$ such that $\mathbb{M}, X[F/x][f/y] \models \phi$. Thus, for each $s \in X$ there is a binary relation $R_s = \{ (a, f(s[a/y])) \mid a \in F(s) \}$ such that $(M, R_s) \models Q_1 x \exists y R(x, y)$. Let

$$Y = \{ s \in X \mid (M, R_s) \models \exists y Q_1 x R(x, y) \}$$

and

$$Z = \{ s \in X \mid (M, R_s) \models Q_1 y \exists x R(x, y) \}.$$

By the validity of the rule for $\mathrm{FO}(Q_1)$ we see that $X = Y \cup Z$.

It should be clear that $Y \models \exists y Q_1 x \phi$ since by letting $g(s)$ be such that $(M, R_s) \models Q_1 x R(x, g(s))$ and

$$G(s[g(s)/y]) = \{ a \in M \mid (M, R_s) \models R(a, g(s)) \},$$

we have that $Y[g/y][G/x] \subseteq X[F/x][f/y]$ and thus by downward closure

$$\mathbb{M}, Y[g/y][G/x] \models \phi.$$

Similarly we can prove that $\mathbb{M}, Z \models Q_1 y \exists x \phi$, and thus that $\mathbb{M}, X \models \Gamma, \exists y Q_1 x \phi \vee Q_1 y \exists x \phi$.

(3) For the Skolem rule assume that there is a derivation of ψ from Γ ending with the Skolem rule. Then there are shorter derivations from Γ of σ and from $\Gamma, S\sigma$ of ψ. By the induction hypothesis we get $\Gamma \models \sigma$ and $\Gamma, S\sigma \models \psi$. We will prove that $\Gamma \models \psi$, by assuming $\mathbb{M}, X \models \Gamma$ for some non-empty X and proving that $\mathbb{M}, X \models \psi$.

From the proof of Theorem 3.5 in [2] we see that $\mathbb{M} \models \sigma$ iff $\mathbb{M} \models \exists f_1 \ldots \exists f_k S\sigma$. From $\mathbb{M}, X \models \Gamma$ and $\Gamma \models \sigma$ we get that $\mathbb{M} \models \sigma$ and thus there are $f_1, \ldots, f_k$ such that $(M, f_1, \ldots, f_k) \models S\sigma$. Now since the f_i's do not occur in formulas $\Gamma' \subseteq \Gamma$ used in the derivation of ψ, $\mathbb{M}, X \models \Gamma$ implies that $(M, f_1, \ldots, f_k), X \models \Gamma'$. By locality, we also have $(M, f_1, \ldots, f_k), X \models S\sigma$. Therefore, by the induction hypothesis, we get that $(M, f_1, \ldots, f_k), X \models \psi$, and, since the f_i's do not occur in ψ, $\mathbb{M}, X \models \psi$ follows. □

Theorem 3. *If T is a set of $\mathrm{D}(Q_1, \check{Q}_1)$ sentences and ϕ is a $\mathrm{FO}(Q_1, \check{Q}_1)$ sentence then $T \vdash \phi$ iff $T \models \phi$.*

Proof. Assume $T \nvdash \phi$. We build a model of $T' = T \cup \{ \neg\phi \} \nvdash \bot$ by translating sentences σ of T into normal form σ_{nf} and considering the $\mathrm{FO}(Q, \check{Q})$ theory $T_S = \{ S\sigma_{\mathrm{nf}} \mid \sigma \in T \} \cup \{ \neg\phi \}$. This theory is consistent, since otherwise the Skolem rule and Proposition 3 would allow us to derive a contradiction from T'.

Since the deduction system for $\mathrm{D}(Q_1, \check{Q}_1)$ contains Keisler's system [4] we may apply the completeness theorem for $\mathrm{FO}(Q_1)$ and get a model $\mathbb{M}$ of $T_S \cup \{ \neg\phi \}$. By the remark made in the proof of Proposition 5 and Proposition 3 $\mathbb{M}$ is also a model of $T \cup \{ \neg\phi \}$. Thus $T \not\models \phi$. □

5 Conclusion

In this article we have presented inference rules and axioms for extensions of dependence logic by monotone generalized quantifiers. We also proved two completeness results for $\mathrm{FO}(Q)$ consequences in the cases where Q either has a weak interpretation or Q is interpreted as "there exist uncountably many." In the first completeness theorem, an important feature of the proof is the approximation of a $\mathrm{D}(Q_1, \check{Q}_1)$ sentence by an infinite set of $\mathrm{FO}(Q)$ sentences. In the second completeness theorem the approximations were replaced by the Skolem rule which however is slightly unsatisfactory due to the extra function symbols f_i used in its formulation. In future work our plan is to further analyze the completeness theorem of $\mathrm{D}(Q_1, \check{Q}_1)$, and replace the Skolem rule with rules that do not rely on the explicit use of the Skolem functions f_i.

References

1. Engström, F.: Generalized quantifiers in dependence logic. Journal of Logic, Language and Information 21, 299–324 (2012)
2. Engström, F., Kontinen, J.: Characterizing quantifier extensions of dependence logic. Journal of Symbolic Logic 78(1), 307–316 (2013)
3. Mostowski, A.: On a generalization of quantifiers. Fund. Math. 44, 12–36 (1957)
4. Keisler, H.: Logic with the quantifier "there exist uncountably many". Annals of Mathematical Logic 1(1), 1–93 (1970)
5. Peters, S., Westerståhl, D.: Quantifiers in Language and Logic. Clarendon Press (2006)
6. Kolaitis, P.G., Väänänen, J.A.: Generalized quantifiers and pebble games on finite structures. Ann. Pure Appl. Logic 74(1), 23–75 (1995)
7. Hella, L., Väänänen, J., Westerståhl, D.: Definability of polyadic lifts of generalized quantifiers. J. Logic Lang. Inform. 6(3), 305–335 (1997)

8. Barwise, J.: On branching quantifiers in English. J. Philos. Logic 8(1), 47–80 (1979)
9. Immerman, N.: Languages that capture complexity classes. SIAM J. Comput. 16(4), 760–778 (1987)
10. Dahlhaus, E.: Skolem normal forms concerning the least fixpoint. In: Börger, E. (ed.) Computation Theory and Logic. LNCS, vol. 270, pp. 101–106. Springer, Heidelberg (1987)
11. Väänänen, J.: Dependence Logic - A New Approach to Independence Friendly Logic. London Mathematical Society Student Texts, vol. 70. Cambridge University Press, Cambridge (2007)
12. Henkin, L.: Some remarks on infinitely long formulas. In: Infinitistic Methods (Proc. Sympos. Foundations of Math., Warsaw, 1959, pp. 167–183. Pergamon, Oxford (1961)
13. Kontinen, J., Väänänen, J.A.: On definability in dependence logic. Journal of Logic, Language and Information 18(3), 317–332 (2009)
14. Lindström, P.: First order predicate logic with generalized quantifiers. Theoria 32, 186–195 (1966)
15. Kaufmann, M.: The quantifier "there exist uncountably many", and some of its relatives. In: Barwise, J., Feferman, S. (eds.) Perspectives in Mathematical Logic. Model Theoretic Logics, pp. 123–176. Springer (1985)
16. Makowsky, J., Tulipani, S.: Some model theory for monotone quantifiers. Archive for Mathematical Logic 18(1), 115–134 (1977)
17. Kontinen, J., Väänänen, J.: Axiomatizing first order consequences in dependence logic. Annals of Pure and Applied Logic (June 6, 2013)
18. Engström, F., Kontinen, J., Väänänen, J.: Dependence logic with generalized quantifiers: Axiomatizations. arxiv:1304.0611 (2013)
19. Barwise, J.: Some applications of henkin quantifiers. Israel Journal of Mathematics 25(1), 47–63 (1976)

Continuous Truth II: Reflections

Michael P. Fourman

School of Informatics, The University of Edinbugh, Scotland, UK
Michael.Fourman@ed.ac.uk

Abstract. In the late 1960s, Dana Scott first showed how the Stone-Tarski topological interpretation of Heyting's calculus could be extended to model intuitionistic analysis; in particular Brouwer's continuity principle. In the early '80s we and others outlined a general treatment of non-constructive objects, using *sheaf models*—constructions from topos theory—to model not only Brouwer's non-classical conclusions, but also his creation of "new mathematical entities". These categorical models are intimately related to, but more general than Scott's topological model.

The primary goal of this paper is to consider the question of iterated extensions. Can we derive new insights by repeating the second act?

In *Continuous Truth I*, presented at Logic Colloquium '82 in Florence, we showed that general principles of continuity, local choice and local compactness hold in the gros topos of sheaves over the category of separable locales equipped with the open cover topology.

We touched on the question of iteration. Here we develop a more general analysis of iterated categorical extensions, that leads to a reflection schema for statements of predicative analysis.

We also take the opportunity to revisit some aspects of both Continuous Truth I and Formal Spaces (Fourman & Grayson 1982), and correct two long-standing errors therein.

Keywords: sheaf model, logic, intuitionism, predicative, analysis, topos.

1 Introduction

Brouwer, in his Cambridge lectures [8], distinguishes two "acts of intuitionism". The first (p. 4) is to reject some "principles of classical logic, blindly formulated." In particular, Brouwer rejects the *principium tertii exclusi:* "the principle of the excluded third, ... cannot in general serve as a principle for discovering mathematical truths." This first act is formally enshrined in Heyting's predicate calculus, which intuitionism shares with various flavours of constructive mathematics.

Brouwer's SECOND ACT OF INTUITIONISM is more subtle.

> *Admitting two ways of creating new mathematical entities: firstly in the shape of more or less freely proceeding infinite sequences of mathematical entities previously acquired; secondly in the shape of mathematical species, ...* (*op cit.* p.8)

Brouwer uses such non-constructive creations to derive strongly non-classical results, such as his celebrated continuity principle:

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 153–167, 2013.

Each full function of the unity continuum is uniformly continuous. (p.80)

To model this 'second act', we base ourselves in a constructive setting $\mathbb{B}$, and model the addition of new mathematical entities by the passage to an extension $\mathbb{E} = \mathbb{B}[D]$, that includes a new entity, D. Working within the extension, we model Brouwer's arguments, and his non-classical conclusions—such as the contnuity principle.

The Lawvere-Tierney notion of an *elementary topos* $\mathbb{E}$ provides a paradigmatic example of a constructive setting, and their construction of a classifying topos, extending a base topos by adding a generic model, D, of some geometric theory, has now been widely used to model the introduction of new mathematical entities (see e.g. [9] for a recent example).

For the simplest infinitary extensions — adding a generic infinite sequence by taking sheaves over formal Baire space or formal Cantor space [1, 2] — it is easy to see that the construction is reflexive. This was part of the folklore thirty years ago, but appears to be still unrecorded in the literature. We first review these examples, and then consider models such as those introduced in [3–6] and used extensively by, e.g., [7, 9].

In [6], we considered the interpretation of logic in the gros topos of sheaves over the category of separable locales equipped with the open cover topology. We showed that general principles of continuity, local choice and local compactness hold for these models. In §5 we touched on the question of iteration. Our analysis there focussed on low-level detail. We failed to see the wood for the trees. Plans, announced there, to develop a high-level account in collaboration with Max Kelly never materialised.

Here we provide a quite general category-theoretic account of iteration — the construction of a model within the model — a preliminary report of ongoing work. This allows us to show that in some models, $\mathfrak{M}$, a reflection principle that states that, *a statement ϕ of predicative analysis is true iff it is true in the model,* is valid:

$$\text{Reflection for } \phi: \qquad \mathfrak{M} \vDash \phi \quad \text{iff} \quad \mathfrak{M} \vDash \ulcorner \mathfrak{M} \vDash \phi \urcorner$$

We also present the basic facts we need relating open locales to open maps, correcting an error in [10]. These facts are no longer new — for an elephantine account see [11] () — but our presentation maybe more accessible, from a logical perspective, than the definitive treatment in [12].

2 Preliminaries

We compose morphisms in diagram order: for $a \xrightarrow{f} b \xrightarrow{g} c$ we have $a \xrightarrow{fg} c$. Otherwise, our notations and definitions generally follow those of Mac Lane & Moerdijk [13] (M&M) or Johnstone  , except where some constructive finesse is required.

Context. Our arguments are intended always to be formalisable in Higher-order Heyting Arithmetic (HAH), a simple impredicative type theory also known as the logic of topoi. Much, maybe all, of what we say will be formalisable within a weaker predicative setting [9, 14], but we have neither space nor time to attempt that here. A set (Kuratowski) *finite* iff it can be enumerated by some natural number, and *countable* iff it can be enumerated by $\mathbb{N}$; in each case, repetitions are allowed. Any countable X is *inhabited* — which means that there is some x such that $x \in X$.

We use the locutions of dependent types, for example when we discuss coverings and sheaves, but these can be interpreted within a simple type theory using a standard categorical trick, due originally to Grothendieck. An indexed type $A_i \mid i \in I$ is given by a morphism $A \longrightarrow I$. This representation means that operations are defined uniformly across the family. Jean Bénabou and his school showed how it can be used to develop category theory in an essentially predicative setting [15].

2.1 Frames and Locales

We recap some facts, which should be well-known [10, 12, 14, 16–19].

Definition 1. *A* frame, $\mathcal{F}$, *is a complete* $\wedge\bigvee$*-distributive lattice; finite meets distribute over arbitrary joins:* $a \wedge \bigvee_{i\in I} b_i = \bigvee_{i\in I}(a \wedge b_i)$. *A frame* morphism *preserves* $\top \wedge \bigvee$. *A* basis $\mathcal{G} \subseteq \mathcal{F}$ *is a subset such that* $u = \bigvee\{v \in \mathcal{G} \mid v \leq u\}$, *for every* $u \in \mathcal{F}$. *The category* $\mathcal{L}$ *of* locales *is defined to be the opposite of the category of frames. Following [12] we often refer to its objects as spaces.*

Given $f : \mathcal{X} \longrightarrow \mathcal{Y}$, a morphism in $\mathcal{L}$, the corresponding frame morphism is the inverse image morphism $f^* : \mathcal{O}(\mathcal{Y}) \longrightarrow \mathcal{O}(\mathcal{X})$. $\mathcal{L}$ can be viewed as a category of generalised spaces [19]. Any set X, can be viewed as a discrete space corresponding to the frame $\mathcal{P}(X)$. The one-point space $\mathbb{1} = \{*\}$ corresponds to the frame $\mathcal{P}(\mathbb{1})$. A *point* of $\mathcal{X}$ is a morphism $x : \mathbb{1} \longrightarrow \mathcal{X}$. Classically, $\mathcal{P}(\mathbb{1})$ appears trivial; constructively it encapsulates the ambient propositional logic,

For $\mathcal{U} \subseteq \mathbb{1}$ we have $\mathcal{U} = \bigvee\{\top \mid * \in \mathcal{U}\}$ ($\{\mathbb{1}\}$ is a basis). Since this join must be preserved, for any locale, $\mathcal{X}$, there is a unique frame morphism,

$$\hat{} : \mathcal{P}(\mathbb{1}) \longrightarrow \mathcal{O}(\mathcal{X}), \text{ given by } \hat{\mathcal{U}} = \bigvee\{\top \mid * \in \mathcal{U}\}, \tag{1}$$

and thus a unique locale morphism $\mathcal{X} \longrightarrow \mathbb{1}$ to the one-point space.

Open Maps. Any frame provides a model of Heyting's propositional calculus. Heyting's implication is given by, $p \Rightarrow q = \bigvee\{r \mid r \wedge p \leq q\}$. Heyting's implication $\Rightarrow$ is not, in general preserved by a frame morphism. Frame morphisms that do preserve $\Rightarrow$ correspond to open maps of locales. They also have a simple logical characterisation [12].

Definition 2. *A locale morphism $f : \mathcal{A} \longrightarrow \mathcal{B}$ is defined to be:* surjective *iff f^* is 1-1;* injective *iff f^* is onto;* open [12, 20][1] *iff f^* preserves both $\bigwedge$ and $\Rightarrow$.*

A locale $\mathcal{A}$ is said to be open (and surjective) *iff the locale morphism $\mathcal{A} \longrightarrow *$ is open (and surjective).*

Remark 1. ([12] V.1; A Lemma 1.5.8) Since a frame morphism f^* preserves $\bigvee$, it has a right adjoint, f_*, given by $f_*A = \bigvee\{B \mid f^*B \leq A\}$. Dually, f^* preserves $\bigwedge$, iff it has a left adjoint, $f_!$, given by $f_!A = \bigwedge\{B \mid A \leq f^*B\}$, in which case f^* preserves $\Rightarrow$ iff $f_!$ satisfies the *Frobenius Condition:* $f_!(A \wedge f^*(B)) = f_!A \wedge B$.

Lemma 1. *The locale morphism $\mathcal{A} \longrightarrow *$ is open iff it preserves $\bigwedge$.*

Proof. (c.f. [12] Chapter V 3.1) For $\mathcal{U}, \mathcal{V} \subseteq \mathbb{1}$

$$\mathcal{U} \leq \mathcal{V} \quad \text{iff} \quad * \in \mathcal{U} \to * \in \mathcal{V} \quad \text{iff} \quad \mathcal{U} = \mathbb{1} \to \mathcal{V} = \mathbb{1} \tag{2}$$

Assuming we have a left adjoint $! \dashv \hat{\cdot}$, so that $!U \leq p$ iff $U \leq \hat{p}$, obviously $\widehat{p \Rightarrow q} \leq \hat{p} \Rightarrow \hat{q}$. It remains to show $\hat{p} \Rightarrow \hat{q} \leq \widehat{p \Rightarrow q}$. Equivalently it suffices to show, assuming $U \wedge \hat{p} \leq \hat{q}$ that $U \leq \widehat{p \Rightarrow q}$. Now the following are equivalent:

$$U \leq \widehat{p \Rightarrow q} \quad \text{iff} \quad !U \leq p \Rightarrow q \quad \text{iff} \quad p \wedge !U \leq q\,.$$

Assuming $U \wedge \hat{p} \leq \hat{q}$, we apply (2) to show the last of these. If $p \wedge !U = \mathbb{1}$ then $p = \mathbb{1}$, so $\hat{p} = \top$. Substituting $\top$ for $\hat{p}$ in our assumption tells us that $U \leq \hat{q}$, so $!U \leq q$; but we also know that $!U = \mathbb{1}$, so $q = \mathbb{1}$. □

As an exercise in this form of constructive argument, we give a direct proof of the Frobenius condition.

Lemma 2. *(p. 618) If the inverse image of locale map $\mathcal{A} \longrightarrow *$ has a left adjoint, $! \dashv \hat{\cdot}$ then it satisfies the* Frobenius condition $!(U \wedge \hat{p}) = !U \wedge p$.

Proof. By adjointness, $!(U \wedge \hat{p}) \leq !U \wedge p$. To show equality, assume $!U \wedge p = \mathbb{1}$ then $!U = \mathbb{1}$ and $p = \mathbb{1}$ so $\hat{p} = \top$ and $!(U \wedge \hat{p}) = !U = \mathbb{1}$. □

Any frame, $\mathcal{O}(\mathcal{X})$, can be used to provide an $\mathcal{O}(\mathcal{X})$-valued interpretation, as in [16], of the impredicative higher-order logic (HAH). This is the interpretation of HAH in the localic topos, $\mathrm{Sh}(\mathcal{X})$, of sheaves on $\mathcal{X}$.

Example 1. Given a locale morphism $\pi_{\mathcal{A}} : \mathcal{A} \longrightarrow \mathcal{X}$ we define an $\mathcal{O}(\mathcal{X})$-valued poset $\mathcal{O}(\mathcal{A}_{/\mathcal{X}})$ with underlying set $\mathcal{O}(\mathcal{A})$. For $U, V \in \mathcal{O}(\mathcal{A})$ we define

$$[\![U = V]\!] = \bigvee\{p \in \mathcal{O}(\mathcal{X}) \mid U \wedge {\pi_{\mathcal{A}}}^*(p) = V \wedge {\pi_{\mathcal{A}}}^*(p)\} \tag{3}$$

$$[\![U \leq V]\!] = \bigvee\{p \in \mathcal{O}(\mathcal{X}) \mid U \wedge {\pi_{\mathcal{A}}}^*(p) \leq V \wedge {\pi_{\mathcal{A}}}^*(p)\} \tag{4}$$

[1] Our earlier paper on Formal Spaces [10] betrayed an unfortunate confusion: our Definition 2.9 of *open map* omitted the Frobenius condition. We are grateful to the eagle-eyed Peter Johnstone for pointing this out in his review, MR0717242 (85c:03023). The statement of Theorem 2 (below) appears already as Lemma 2.12 of [10], but in the context of this weaker definition of 'open'—thus making a weaker claim. Lemma 1 provides the necessary buttress to our earlier proof.

This $\mathcal{O}(\mathcal{X})$-valued poset can be viewed a frame within the $\mathcal{O}(\mathcal{X})$-valued interpretation. In fact, every internal frame in a localic topos arises in this way [10]. Given $\pi_{\mathcal{B}} : \mathcal{B} \longrightarrow \mathcal{X}$, a map $f^* : \mathcal{O}(\mathcal{A}) \longrightarrow \mathcal{O}(\mathcal{B})$ represents an internal map $f^* : \mathcal{O}(\mathcal{A}_{/\mathcal{X}}) \longrightarrow \mathcal{O}(\mathcal{B}_{/\mathcal{X}})$ iff it is *extensional* in the sense that,

$$\text{for all } U, V \in \mathcal{O}(\mathcal{A}), \text{ we have } [\![U = V]\!] \leq [\![f^*U = f^*V]\!]\,. \tag{5}$$

Extensional maps correspond to commuting triangles $\pi^*_{\mathcal{A}} = \pi^*_{\mathcal{B}} f^*$.

Lemma 3. *For any extensional map*

$$f^* : \mathcal{O}(\mathcal{A}_{/\mathcal{X}}) \longrightarrow \mathcal{O}(\mathcal{X}_{/\mathcal{X}}), \text{ we have } f^*(V) \wedge p \leq f^*(V \wedge {\pi_{\mathcal{A}}}^*(p))\,. \tag{6}$$

Proof. It follows from (3) that, $p \leq [\![U = V]\!]$ iff $U \wedge {\pi_{\mathcal{A}}}^*(p) = V \wedge {\pi_{\mathcal{A}}}^*(p)$. Since $p \leq [\![V = V \wedge {\pi_{\mathcal{A}}}^*(p)]\!]$, we have $p \leq [\![f^*(V) = f^*(V \wedge {\pi_{\mathcal{A}}}^*(p))]\!]$, and thus, $f^*(V) \wedge p \leq f^*(V \wedge {\pi_{\mathcal{A}}}^*(p))$. This is the semantic counterpart to Lemma 2. □

Proposition 1. *[12] The locale morphism $\pi_{\mathcal{A}} : \mathcal{A} \longrightarrow \mathcal{X}$ is open (and surjective) iff the the $\mathcal{O}(\mathcal{X})$-valued poset $\mathcal{O}(\mathcal{A}_{/\mathcal{X}})$ it represents is internally open (and surjective).*

Proposition 2. *An element $U \in \mathcal{O}(\mathcal{X})$ is said to be* positive (Pos(U)) *iff every cover of U is inhabited. A locale, $\mathcal{X}$, is* surjective *iff* Pos($\top$)*, and* open *iff* $\{U \mid \text{Pos}(U)\}$ *is a basis for $\mathcal{O}(\mathcal{X})$.* [2]

Formal Spaces. are locales presented as spaces of models for some, possibly infinitary, geometric propositional theory. If $x : \mathbb{1} \longrightarrow \mathcal{X}$ is a point of $\mathcal{X}$, then for each $\mathcal{U} \in \mathcal{O}(\mathcal{X})$, we write $x \in \mathcal{U}$ to mean that $* \in x^*(\mathcal{U})$, so $x^*(\mathcal{U}) = [\![x \in \mathcal{U}]\!]$. We can use the same notation, $[\![\alpha \in \mathcal{U}]\!] = \alpha^*(\mathcal{U})$, for a *generalised point*, α, which is just a morphism $\alpha : \mathcal{A} \longrightarrow \mathcal{X}$.

Consider a language $\mathcal{L}$ with a set of basic propositions $p \in \mathbb{P} \subseteq \mathcal{O}(\mathcal{X})$. An $\mathcal{O}(\mathcal{A})$-valued model for $\mathcal{L}$ is given by a morphism $\alpha : \mathcal{A} \longrightarrow \mathcal{X}$. We give each basic proposition $p \in \mathbb{P}$ the truth value $[\![p]\!]_\alpha = \alpha^*(p) = [\![\alpha \in p]\!]$. We say a *sequent*, $p \vdash C$, where $p \in \mathbb{P}$ and $C \subseteq \mathbb{P}$, is *valid* for α iff $[\![p]\!]_\alpha \leq \bigvee \{[\![q]\!]_\alpha \mid q \in C\}$.

Definition 3. *[10] A* geometric presentation *of a formal space* $(\mathbb{P}, \mathcal{A})$ *consists of a structure* $\mathbb{P}$ *of* basic propositions *and a collections* $\mathcal{A}$ *of* axioms:

$\mathbb{P}$ *is a preordered set with conditional finite meets: if a finite set has a lower bound then it has a greatest lower bound. In particular, if* $\mathbb{P}$ *is inhabited, then it has a top element* $\top$*. We write* $p{\downarrow}$ *for* $\{q \mid q \leq p\}$*. A* crible *of p is a set $K \subseteq p{\downarrow}$, such that $\forall q \in K.\, q{\downarrow} \subseteq K$. For K a crible of p and $q \leq p$, observe that, $K \restriction q = K \cap q{\downarrow}$ is a crible of q.*

$\mathcal{A}$ *is a* covering relation*, that is, a set of* sequents*, $p \vdash C$, "C is a basic cover of p", where $p \in \mathbb{P}$ and $C \subseteq p{\downarrow}$, which is* stable *in the sense that, if $p \vdash C$, K is a crible of p with $C \subseteq K$, and $q \leq p$, then there is some basic cover $q \vdash D$ of q such that $D \subseteq K \restriction q$.*

[2] These appeared in [10] but are due to Joyal ([12] Chapter V; Lemma C3.1.7).

A crible, K of $\top$ is closed *iff for all basic covers $p \vdash C$, if $C \subseteq K$ then $p \in K$. The closed cribles are the* formal opens $\mathcal{O}(\mathbb{P}, \mathcal{A})$ *of the formal space.*

We say the formal space is separable *if $\mathbb{P}$ is countable and has decidable equality.*

$\mathcal{O}(\mathbb{P}, \mathcal{A})$ is a frame. The corresponding locale is the formal space $(\mathbb{P}, \mathcal{A})$ of models of the presentation. Geometrically a sequent, $p \vdash C$, is a cover; logically we read it as an entailment, where the right-hand side is an implicit disjunction. When presenting a formal space we often write a cover as a formal disjunction—this can be viewed as simply a suggestive notation for a set of basic propositions.

Proposition 3. *The formal space $(\mathbb{P}, \mathcal{A})$ is* open *if for every cover $p \vdash C \in \mathcal{A}$, C is inhabited. In this case, if $\mathbb{P}$ is inhabited the space is surjective.*

Definition 4. *A $\mathcal{O}(\mathcal{X})$-valued model of $(\mathbb{P}, \mathcal{A})$ is an assignment of a* truth value *$[\![p]\!] \in \mathcal{O}(\mathcal{X})$ to each basic proposition such that:*

$$p \leq q \to [\![p]\!] \leq [\![q]\!] \qquad [\![\top]\!] = \top \qquad [\![p]\!] \wedge [\![q]\!] \leq \bigvee \{[\![p \wedge q]\!]\} \tag{7}$$

$$[\![p]\!] \leq \bigvee \{[\![q]\!] \mid q \in C\} \text{ for each axiom } p \vdash C \in \mathcal{A}\,. \tag{8}$$

Morphisms, $\alpha^* \colon \mathcal{O}(\mathbb{P}, \mathcal{A}) \longrightarrow \mathcal{O}(\mathcal{X})$, from a locale $\mathcal{X}$ to a formal space $(\mathbb{P}, \mathcal{A})$ correspond to $\mathcal{O}(\mathcal{X})$-valued models. Since each $p \in \mathbb{P}$ is a basic open of the formal space, we write $[\![\alpha \in p]\!]$ for $\alpha^*(p)$.

Examples. For each example below, the basic opens are well-known from mathematical practice, and we adjust our notation accordingly — for example, to axiomatise a real number, we write $p < r < q$ in place of $r \in (p, q)$.

For any set X, the *discrete formal space*, $\mathcal{X}$, is given by $\mathbb{X} = X^\top$, the poset obtained by adjoining a (new) top element, $\top$, with $\forall x \in X.\, x < \top$, together with a single axiom:

$$\mathcal{X} \qquad \top \vdash \bigvee_{x \in X} \alpha = x \tag{9}$$

Each basic open is a singleton; we write $\alpha = x$ for the basic proposition x. The corresponding frame is the power set, $\mathcal{O}(\mathbb{X}, \top \vdash X) = \mathcal{P}(X)$. A $\mathcal{P}(\mathbb{1})$-valued model, corresponds to a point of the formal space of models of $(\mathbb{P}, \mathcal{A})$. A $\mathcal{P}(X)$-valued model corresponds to a function: an X-indexed family of models.

If $(\mathbb{P}, \mathcal{A})$ and $(\mathbb{Q}, \mathcal{B})$ are formal spaces, their *product* is given by $(\mathbb{P} \times \mathbb{Q}, \mathcal{A} + \mathcal{B})$ where $\mathbb{P} \times \mathbb{Q}$ has the product (pointwise) preorder and $\mathcal{A} + \mathcal{B}$ includes both $\{(p, q) \vdash C \times \{q\} \mid p \vdash_{\mathcal{A}} C\}$ and $\{(p, q) \vdash \{p\} \times C \mid q \vdash_{\mathcal{B}} C\}$.

If $\mathcal{X} = (\mathbb{P}, \mathcal{A})$ is a formal space, and X is a set then the formal product space is $\mathcal{X}^X = (X \otimes \mathbb{P}, X \otimes \mathcal{A})$. We introduce a formal $\alpha : X \longrightarrow \mathcal{X}$. The basic proposition (x, p) should be read as $x \in \alpha^*(p)$.

Here, $X \otimes \mathbb{P}$ consists of those finite subsets $F \subseteq X \times \mathbb{P}$ satisfying the *compatibility condition:* $(x, p) \in F \wedge (x, q) \in F \to (x, p \wedge q) \in F$, ordered by,

$$F \leq G \text{ iff } \forall (x, p) \in G.\, p = \top \vee \exists q \leq p.\, (x, q) \in F\,. \tag{10}$$

We say (x, p) is *compatible with* $F \in X \otimes \mathbb{P}$ iff for every q such that $(x, q) \in F$ the meet $p \wedge q$ is defined. In this case, we write

$$F \oplus (x, p) \text{ for } F \cup \{(x, p)\} \cup \{(x, p \wedge q) \mid (x, q) \in F\}\,.$$

$X \otimes \mathcal{A}$ includes a family of covers for each cover $p \vdash C \in \mathcal{A}$. For each (x, p) compatible with F we have a cover $F \oplus \{(x, p)\} \vdash \{\{F \oplus \{q\} \mid q \in C\}$. This constructive presentation of a product of locales is a (very) special case of Hyland's construction of exponents [21] Proposition 3: a discrete space is locally compact!

Formal *Baire Space*, $\mathcal{B}$, is the formal space of models of the theory of a function $\alpha : \mathbb{N} \longrightarrow \mathbb{N}$; for brevity we call it simply the formal space of functions $\alpha \in \mathbb{N}^{\mathbb{N}}$. Similarly, formal *Cantor Space*, $\mathcal{C}$, is the formal space of functions $\alpha \in \mathbf{2}^{\mathbb{N}}$. The basic opens correspond to finite initial segments of an infinite sequence: $a \prec \alpha$ is the proposition represented by $a \in \mathbb{T} = \mathbb{X}^{<\mathbb{N}}$, the tree of finite sequences, where $\mathbb{X}$ is $\mathbb{N}$ or $\mathbf{2}$, respectively. In any model, $\top \vdash \langle\rangle \prec \alpha$, since $\langle\rangle = \top$. We also require:

$$\mathcal{B} \qquad a \prec \alpha \vdash \bigvee_{n \in \mathbb{N}} a\hat{}n \prec \alpha \qquad \text{for each } a \in \mathbb{T} = \mathbb{N}^{<\mathbb{N}}, \tag{11}$$

$$\mathcal{C} \qquad a \prec \alpha \vdash a\hat{}0 \prec \alpha \vee a\hat{}1 \prec \alpha \qquad \text{for each } a \in \mathbb{T} = \mathbf{2}^{<\mathbb{N}}, \tag{12}$$

where, $a\hat{}n$ is the extension of a by n. So, a model corresponds to an infinite path through the tree. We could, of course, construct these as exponents of discrete spaces.

The formal *Dedekind Reals*, $\mathcal{R}$, axiomatise an open cut in the rationals. Our basic propositions are proper, rational open intervals, (p, q) with $p < q$, where $p, q \in \mathbb{Q}^* = \mathbb{Q} \cup \{-\infty, \infty\}$. These intervals are ordered by inclusion. We write $p < r < q$, for $r \in (p, q)$. The covering axioms are:

$$\mathcal{R} \qquad \begin{array}{l} p < r < q \vdash \bigvee \{p' < r < q' \mid p < p' < q' < q\} \\ p < r < q \vdash p < r < q' \vee p' < r < q \text{ where, } p < p' < q' < q \end{array} \tag{13}$$

Definition 5. *[17] A locale $\mathcal{A}$ is T_1 iff for every locale $\mathcal{X}$, the specialisation ordering on the set of $\mathcal{X}$-valued points $[\mathcal{X}, \mathcal{A}]$ is trivial: $x \leq y \to x = y$, or, equivalently, if every localic topos, $Sh(\mathcal{X})$ satisfies*

$$\forall x, y \in Pt(\mathcal{A}).\, \forall U \in \mathcal{O}(\mathcal{A}).\, (x \in U \to y \in U) \to x = y\,.$$

Lemma 4. *Each of $\mathcal{X}, \mathcal{B}, \mathcal{C}, \mathcal{R}$ is T_1. if $\mathcal{A}$ is T_1 then so is $\mathcal{A}^X$ for any set X.*

Proof. For Baire space and Cantor space this is straightforward, since the values $[\![\alpha(n) = m]\!]$, where $m, n \in \mathbb{N}$, determine α, which is single-valued. Lemma 1.2.17 tells us that since $\mathcal{R}$ is regular, it is T_1 (there called T_U), but a direct constructive proof "in the internal logic" is instructive.

Suppose $s \leq r$, are generalised points of $\mathcal{R}$: that is, for any $\mathcal{U} \in \mathcal{O}(\mathcal{R})$, if $s \in \mathcal{U}$ then $r \in \mathcal{U}$ (it suffices to assume this for every proper rational interval

$\mathcal{U}$). For $p < q < q' < p'$, let $\mathcal{P} = (p, p')$ and $\mathcal{Q} = (q, q')$. It suffices to show that, if $r \in \mathcal{Q}$ then $s \in \mathcal{P}$, since such proper subintervals cover $\mathcal{P}$. Let $\mathcal{W}$ be such that $\mathcal{W} \wedge \mathcal{Q} = \bot$ and $\mathcal{W} \vee \mathcal{P} = \top$ in $\mathcal{O}(\mathcal{R})$. (if such an $\mathcal{W}$ exists we say $\mathcal{Q} \lhd \mathcal{P}$; in this case, it can be chosen as the join of two basic intervals.) Certainly $s \in \mathcal{P}$ or $s \in \mathcal{W}$, since $\mathcal{W} \vee \mathcal{P} = \top$. So $s \in \mathcal{P}$ or $r \in \mathcal{W}$. Now suppose $r \in \mathcal{Q}$; this is incompatible with $r \in \mathcal{W}$, since $\mathcal{W} \wedge \mathcal{Q} = \bot$, so we conclude that $s \in \mathcal{P}$.

Tracing the interpretation of this argument would give an algebraic proof: a sequence of inequalities starting from the assumption that $[\![s \in \mathcal{U}]\!] \leq [\![r \in \mathcal{U}]\!]$ and showing for the chosen basic opens $\mathcal{Q} \lhd \mathcal{P}$ that $[\![r \in \mathcal{Q}]\!] \leq [\![s \in \mathcal{P}]\!]$. □

Lemma 5. *If $\mathcal{A} = (\mathbb{P}, \mathcal{A})$ is a formal space, and we lift the presentation to $Sh(\mathcal{X})$ then the corresponding internal locale — the interpretation of the (lifting of the) presentation in $Sh(\mathcal{X})$ — is represented by the projection $\mathcal{X} \times \mathcal{A} \longrightarrow \mathcal{X}$, where $\mathcal{X} \times \mathcal{A}$ is the product locale.*

Proposition 4. *If $\mathcal{A} \longrightarrow \mathcal{X}$ is a morphism of locales, then the corresponding geometric morphism $Sh(\mathcal{A}) \longrightarrow \mathbb{E} = Sh(\mathcal{X})$ is equivalent to the extension $\underset{\mathbb{E}}{Sh}(\mathcal{A}_{/\mathcal{X}}) \longrightarrow \mathbb{E} = Sh(\mathcal{X})$.*

Adjoint Retracts. Now consider two formal spaces, $\mathcal{P} = (\mathbb{P}, \mathcal{A})$ and $\mathcal{Q} = (\mathbb{Q}, \mathcal{A})$, where $\mathbb{P} \subseteq \mathbb{Q}$ is a subset, with the inherited preorder, closed under conditional finite meets: if a finite subset of $\mathbb{P}$ has a meet in $\mathbb{Q}$ then its meet is in $\mathbb{P}$. These presentations have possibly different posets of basic propositions, but the same axioms, which must mention only propositions in $\mathbb{P}$. Clearly the map $i^* : \mathcal{O}(\mathbb{Q}, \mathcal{A}) \longrightarrow \mathcal{O}(\mathbb{P}, \mathcal{A})$, given by $V \longmapsto V \cap \mathbb{P}$, is a frame morphism, that also preserves $\bigwedge$. So it has both right (i_*) and left ($i_!$) adjoints:

$$U \overset{i_*}{\longmapsto} \{q \mid \forall p \leq q.\, p \in U\} \qquad U \overset{i_!}{\longmapsto} \{q \mid \exists p \in U.\, q \leq p\}$$

$$\mathcal{O}(\mathbb{P}, \mathcal{A}) \underset{i_! = r^*}{\overset{i_*}{\underset{i^* = r_*}{\leftarrow}}} \mathcal{O}(\mathbb{Q}, \mathcal{A})$$

Lemma 6. *[18] In the situation just described, $i_!$ preserves $\wedge$, so we have an adjoint retraction $\mathcal{P} \subsetneq \mathcal{Q}$ of locales. For any T_1 locale, $\mathcal{X}$, we have an equivalence $\mathcal{L}[\mathcal{P}, \mathcal{X}] \cong \mathcal{L}[\mathcal{Q}, \mathcal{X}]$.*

3 Reflections

A reflection principle in set theory asserts that some property of the class of all sets is *reflected* already in some set, and thus serves to extend the universe of discourse and reduce incompleteness. A proto-example might be the introduction of an infinite set by reflection on the closure of the class of all sets under the successor operation $x \longmapsto x \cup \{x\}$. (See e.g. [22] for more elevated examples.)

Brouwer's introspection serves a similar philosophical purpose. It is natural to ask whether iterating Brouwer's second act leads to further insights. We say

that an extension is reflexive if truth in the iterated model is reflected to the model, as described in the Introduction.

3.1 Topological Models

Joyal first pointed out that topological models are best viewed as localic models that introduce a *generic point* of a formal space. From this perspective, Scott's topological model [1, 2] is an extension constructed by adding a generic point of $\mathbb{N}^{\mathbb{N}}$. From the classical perspective adopted in Scott's two papers there is no difference between the open sets of the space of points of $\mathbb{N}^{\mathbb{N}}$, equipped with the product topology, and the formal Baire space $\mathcal{B}$ [10]. Here we start from a non-classical base. We take the formal space as the primary object of study.

Classically, the theories $\mathcal{R}, \mathcal{B}, \mathcal{C}$ are complete — which means, in each case, that the formal space has enough points (to distinguish the formal opens), or equivalently that the topological opens and formal opens coincide. Constructively, this is not provable in HAH—completeness is equivalent (in HAH), for $\mathcal{R}$, to the Heine-Borel theorem ($\mathbb{R}$ is locally compact), and for $\mathcal{B}, \mathcal{C}$ to Brouwer's *Principle of Bar Induction*, and *Fan Theorem*, respectively [10].

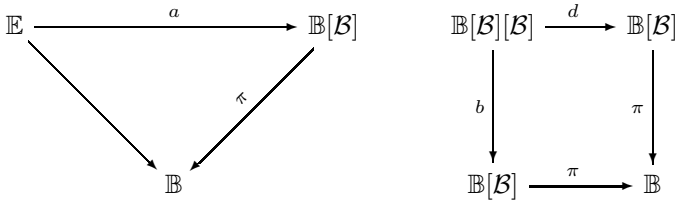
Theorem 1. *The $\mathcal{O}(\mathcal{X})$-valued model includes sufficient points to distinguish the formal opens of $\mathcal{X}$, where $\mathcal{X}$ may be $\mathcal{B}, \mathcal{C}$, or $\mathcal{R}$.*

Proof. Our proof is constructive, and does not presume a metatheory in which $\mathcal{X}$ has enough points. Let $\mathcal{O}(\mathcal{X}) = \mathcal{O}(\mathbb{P}, \mathcal{A})$. In the $\mathcal{O}(\mathcal{X})$-valued model, $\mathcal{X}$ is represented by a projection $\pi : \mathcal{X} \times \mathcal{X} \longrightarrow \mathcal{X}$. Elements of $\mathbb{P}$ play two rôles: in the first dimension (onto which we project), $p \in \mathbb{P}$ is a basic truth value. In the second dimension $q \in \mathbb{P}$ is a basic proposition of the internal presentation of $\mathcal{X}$. An internal formal open $K \in \mathcal{O}(\mathcal{X})$ is represented by a formal open of $\mathcal{X} \times \mathcal{X}$, determined by the values $[\![q \in K]\!]$, for $q \in \mathbb{P}$, which, in turn, are determined by the sets $\{p \in \mathbb{P} \mid p \leq [\![q \in K]\!]\}$.

Internal points are functions $\alpha : \mathcal{X} \longrightarrow \mathcal{X}$, or, equivalently, sections of the projection [16]. The identity function on $\mathcal{X}$ (the diagonal section) gives a *generic point*, γ. By definition, $[\![\gamma \in q]\!] = q{\downarrow}$, for any $q \in \mathbb{P}$.

To show that $\mathcal{X}$ has enough points it suffices to exhibit points $\alpha_{p,q}$, where if $q = \top$ then $p = \top$, such that $[\![\alpha_{p,q} \in \bar{q}]\!] = p{\downarrow}$; that is, frame morphisms $\alpha^*_{a,b} : \mathcal{X} \longrightarrow \mathcal{X}$ such that $\alpha^*_{p,q}(q{\downarrow}) = p{\downarrow}$. In the case of the reals, for example, there is a unique rational linear function that maps one rational open interval to another; so these functions suffice to distinguish formal opens. We leave $\mathcal{B}$ and $\mathcal{C}$ as an exercise for the reader. □

Whether the base from which we start is classical or constructive, the localic model using the formal opens produces an extension $\pi : \mathbb{B}[\mathcal{B}] \longrightarrow \mathbb{B}$ that includes a generic point, corresponding to the identity morphism $\gamma : \mathcal{B} \longrightarrow \mathcal{B}$. Geometrically, *generic* means that the points α of $\mathcal{B}$ in any topos $\mathbb{E} \longrightarrow \mathbb{B}$ correspond to geometric morphisms, $\mathbb{E} \xrightarrow{a} \mathbb{B}[\mathcal{B}]$, with $\alpha = a^*(\gamma)$, making the triangle commute.



Iterating this construction gives us a topos $\mathbb{B}[\mathcal{B}][\mathcal{B}] \xrightarrow{b} \mathbb{B}[\mathcal{B}]$. Like any topos over $\mathbb{B}[\mathcal{B}]$ it includes a point $\beta = b^*\gamma \in \mathcal{B}$; it also includes another point $\delta \in \mathcal{B}$ which is generic for points of $\mathcal{B}$ in toposes over $\mathbb{B}[\mathcal{B}]$. Since any topos over $\mathbb{B}[\mathcal{B}]$ is also a topos over $\mathbb{B}$, we see that δ corresponds to a morphism $\mathbb{B}[\mathcal{B}][\mathcal{B}] \xrightarrow{d} \mathbb{B}[\mathcal{B}]$ making the square commute. Furthermore, the square is a pullback, by the universal property of our second extension. Logically, $\mathbb{B}[\mathcal{B}][\mathcal{B}] \longrightarrow \mathbb{B}$ classifies pairs of models of the formal space $\mathcal{B}$. Geometrically, it is given by the formal space $\mathcal{B} \times \mathcal{B}$ whose points are pairs of points of $\mathcal{B}$.

Classically, it is well-known that $\mathcal{B} \times \mathcal{B} \cong \mathcal{B}$. The classical proof exhibits a homeomorphism $\mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \cong \mathbb{N}^{\mathbb{N}}$, for example, a `zip` function that interleaves two sequences, whose inverse takes α to the pair $(\mathtt{even}(\alpha), \mathtt{odd}(\alpha))$. So the double extension is equivalent to the single extension, and has the same logic. Working constructively, the same is true, but we must work directly with the formal opens. The map $a\downarrow \longmapsto (\mathtt{even}(a), \mathtt{odd}(a))\downarrow$ gives a homeomorphism of formal spaces.

Entirely analogous remarks hold for formal Cantor space, $\mathcal{C}$, *mutatis mutandis*, with $\mathbb{T} = \mathbf{2}^{\mathbb{N}}$. So we have full reflection for [5] (2.1) Open Data.

Proposition 5. *If $\mathbb{B}$ is an elementary topos, $\mathcal{B}$ the formal Baire space, and $\mathcal{C}$ the formal Cantor space in $\mathbb{B}$, then $\mathbb{B}[\mathcal{B}][\mathcal{B}] \equiv \mathbb{B}[\mathcal{B}]$ and $\mathbb{B}[\mathcal{C}][\mathcal{C}] \equiv \mathbb{B}[\mathcal{C}]$ as toposes over $\mathbb{B}$. So, $\mathbb{B}[\mathcal{B}] \vDash \phi$ iff $\mathbb{B}[\mathcal{B}] \vDash \ulcorner \mathbb{B}[\mathcal{B}] \vDash \phi \urcorner$, for any formula ϕ of HAH.*

We have no such straightforward reflection theorem for $\mathcal{R}$, since $\mathcal{R} \not\equiv \mathcal{R} \times \mathcal{R}$.

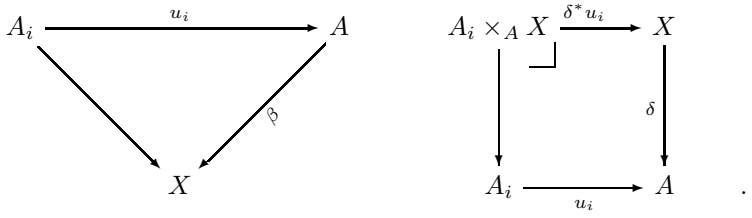
3.2 Extensions over Sites

Definition 6. *[12] A* site $(\mathbb{C}, \mathcal{J})$ *is a category $\mathbb{C}$ equipped with a covering system, $\mathcal{J}$. That is, a collection, $\mathcal{J}(A)$, of* covers $R = \{A_i \longrightarrow A\}_{i \in I}$, *for each object A in $\mathbb{C}$, such that: if $\alpha : A' \longrightarrow A$ and $R \in \mathcal{J}(A)$, then, for some $R' \in \mathcal{J}(A')$ we have $R' \subseteq \{\beta : A'' \longrightarrow A' \mid \beta\alpha \in \hat{R}\}$.*

The Fundamental Fibration. [15] Let $\mathbb{T} = (\mathbb{C}, \mathcal{J})$ be a site, where $\mathbb{C}$ has pullbacks. The topos of sheaves can be viewed as an extension $\mathbb{B}[\mathbb{C}, \mathcal{J}] = \mathrm{Sh}(\mathbb{C}, \mathcal{J})$: the Yoneda embedding provides a universal model of $(\mathbb{C}, \mathcal{J})$ in a topos over our base, $\mathbb{B}$. Grothendieck showed how the Yoneda embedding $Y : \mathbb{C} \longrightarrow \mathbb{B}[\mathbb{C}, \mathcal{J}]$ can be viewed as an internal site. We view the codomain projection $\mathbb{C}^{\mathbf{2}} \longrightarrow \mathbb{C}$, from the category of arrows $\mathbb{C}^{\mathbf{2}}$ to $\mathbb{C}$ as an internal category, $\mathbb{C}^{\mathbf{2}}_{/\mathbb{C}}$, whose fibre over a representable X is the slice category $\mathbb{C}/X$. For $\alpha : Y \longrightarrow X$, the restriction map $\alpha^* : \mathbb{C}/X \longrightarrow \mathbb{C}/Y$ is given by pullbacks along α.

We have described the internal category $\mathbb{C}^{\mathbf{2}}_{/\mathbb{C}}$ corresponding to the fibration ∂_1. The topology $\mathcal{J}$ also lifts to an internal topology $\mathcal{J}_{/\mathbb{C}}$ on $\mathbb{C}^{\mathbf{2}}_{/\mathbb{C}}$: if $R = \{u_i : A_i \longrightarrow A\}_{i\in I}$ is a covering family in $\mathbb{C}$, and $\beta : A \longrightarrow X \in \mathbb{C}/X$, then $R\beta = \{u_i : u_i\beta \longrightarrow \beta\}_{i\in I}$ covers in $\mathbb{C}/X$.

To give an external representation of this extension, let $\mathcal{J}^{\mathbf{2}}$ be the topology on $\mathbb{C}^{\mathbf{2}}$ with covering families as follows: If $R = \{u_i : A_i \longrightarrow A\}_{i\in I}$ is a covering family in $\mathbb{C}$, and $\beta : A \longrightarrow X \in \mathbb{C}/X$, then $R\beta = \{u_i : u_i\beta \longrightarrow \beta\}_{i\in I}$ covers $\beta : A \longrightarrow X$. If, furthermore, $\delta : X \longrightarrow A$ then the pullbacks $\delta^* u_i : A_i \times_A X \longrightarrow X$ below cover $\delta : X \longrightarrow A$:



The codomain morphism $\partial_1 : \mathbb{C}^{\mathbf{2}} \longrightarrow \mathbb{C}$ gives a geometric morphism,

$$\partial_1 : \mathrm{Sh}(\mathbb{C}^{\mathbf{2}}, \mathcal{J}^{\mathbf{2}}) \longrightarrow \mathrm{Sh}(\mathbb{C}, \mathcal{J}),$$

whose inverse image is composition with ∂_1 followed by sheafification; and whose direct image is given by composition with the inclusion $\Delta : \mathbb{C} \longrightarrow \mathbb{C}^{\mathbf{2}}$, which takes each object to its identity morphism. The inverse image just constructs internal constant sheaves, and the direct image takes global sections. Since Δ preserves covers and $\partial_1 \dashv \Delta$, this is a case of Theorem 4 of M&M §VII.10.

Proposition 6

The geometric morphism $\partial_1 : Sh(\mathbb{C}^{\mathbf{2}}, \mathcal{J}^{\mathbf{2}}) \longrightarrow \mathbb{E} = Sh(\mathbb{C}, \mathcal{J})$
is equivalent to the extension $\underset{\mathbb{E}}{Sh}(\mathbb{C}^{\mathbf{2}}_{/\mathbb{C}}, \mathcal{J}^*) \longrightarrow \mathbb{E} = Sh(\mathbb{C}, \mathcal{J})$.

We now investigate the logical properties of the iterated extension. Just as in the localic case, it suffices to find some functor comparing $\widetilde{\mathbb{C}^{\mathbf{2}}}$ and $\widetilde{\mathbb{C}}$. We have three functors $\partial_1 \dashv \Delta \dashv \partial_0$. These all preserve covers, so ∂_1 and Δ have the cover lifting property (M&M §VII.10 Lemma 3).

Lemma 7. *∂_0 also has the cover lifting property.*

Proof. If $\mathbf{R} = \{\mathbf{X_i} \longrightarrow \mathbf{X}\}_{i\in I}$ is such that both $\partial_0(\mathbf{R}) = \{\partial_0(\mathbf{X_i}) \longrightarrow \partial_0(\mathbf{X})\}_{i\in I}$ and $\partial_1(\mathbf{R})$, defined similarly, are covers in $\mathbb{C}$, then $\mathbf{R}$ is a cover in $\mathbb{C}^{\mathbf{2}}$.

Suppose $\mathbf{X} = \pi : X \longrightarrow Y$ and $R = \{u_i : X_i \longrightarrow X\}_{i\in I}$ is a cover in $\mathbb{C}$ then the morphisms, $(u_i, \mathbb{1}_Y)$, from the objects $R/Y = \{u_i\pi : X_i \longrightarrow Y\}_{i\in I}$ in $\mathbb{C}^{\mathbf{2}}$, to $\mathbf{X}$, form a cover in $\mathcal{J}^{\mathbf{2}}$.

We are now in the situation where each of the adjoint functors $\partial_1 \dashv \Delta \dashv \partial_0$ preserves covers and has the cover lifting property. Therefore (M&M §VII.10), we

have three geometric morphisms whose inverse images are given by composition (e.g. $\partial_0^*(A) = \partial_0 A$), followed by sheafification. We write $\widetilde{\mathbb{C}}$ for $\mathrm{Sh}(\mathbb{C})$.

$$\widetilde{\mathbb{C}^{\mathbf{2}}} \underset{\partial_1}{\overset{\partial_0}{\rightleftarrows}} \widetilde{\mathbb{C}} \qquad \widetilde{\mathbb{C}^{\mathbf{2}}} \xleftarrow{\partial_1^*} \widetilde{\mathbb{C}} \underset{\partial_{1*} = \partial_{0!}}{\overset{\Delta^*}{\leftarrow}} \widetilde{\mathbb{C}^{\mathbf{2}}} \underset{\Delta_*}{\overset{\partial_0^*}{\leftarrow}} \widetilde{\mathbb{C}} \tag{14}$$

Both Δ^* and ${\partial_0}^*$ preserve sheaves, so, for them, sheafification is unnecessary. Since $\Delta\partial_0 = \mathbb{1}_{\mathbb{C}}$ we have an adjoint retraction $\widetilde{\mathbb{C}} \underset{\delta_0}{\overset{\Delta}{\rightleftarrows}} \widetilde{\mathbb{C}^{\mathbf{2}}}$, and ∂_0 is a surjection. We have $\partial_{0!} = \Delta^* \dashv \partial_0^*$ and, since these functors preserve sheaves,

$$\Delta^*(B \times_{\widetilde{\mathbb{C}^{\mathbf{2}}}} \partial_0^*(A)) = (\Delta B \times_{\widetilde{\mathbb{C}}} \Delta\partial_0 A) = \Delta B \times_{\widetilde{\mathbb{C}}} A = \Delta^*(B) \times_{\widetilde{\mathbb{C}}} A\,; \tag{15}$$

the Frobenius condition holds. This means that ∂_0 is locally connected, hence open; it preserves exponentials and first-order logic. In fact, we are in the situation described by Moerdijk and Reyes [23] Theorem 2.2: ∂_0 is a left-exact functor which preserves covers and has the covering lifting property; $\Delta \dashv \partial_0$ is a left-adjoint "right inverse", $\Delta\partial_0 = \mathbb{1}_{\mathbb{C}}$. This gives a principle of *predicative reflection.*

Proposition 7. *[23] ∂_0^* preserves and reflects first-order logic, preserves exponentials, and preserves the sheaf of points of any T_1 formal space.*

So, $Sh(\mathbb{C},\mathcal{J}) \vDash \phi$ iff $Sh(\mathbb{C},\mathcal{J}) \vDash \ulcorner Sh(\mathbb{C}^{\mathbf{2}}_{/\mathcal{X}}, \mathcal{J}^*) \vDash \phi \urcorner$, for ϕ a formula in a language for predicative analysis — a language with finite types over $\mathbb{N}, \mathcal{R}$, possibly with constants for relations and functions in $Sh(\mathbb{C},\mathcal{J})$.

We might hope for an impredicative reflection, but this seems unlikely for extensions over sites. Extensions that preserve powersets are quite special.

Lemma 8. *Let $\mathbb{C},\mathcal{J})$ be a site. If $\Omega(B)$, the frame of closed cribles of some $B \in \mathbb{C}_0$, is isomorphic to a powerset $\mathcal{P}(X)$, then X is a singleton, and every inhabited sieve contains $\mathbb{1}_B$.* [3]

Proof. Let ϕ be the composite $X \longrightarrow \mathcal{P}(X) \longrightarrow \Omega(B)$ be the composite of the singleton map with the isomorphism. Then, since $X = \bigcup\{\{x\} \mid x \in X\}$, we have $\mathbb{1}_B \in \bigcup\{\phi(x) \mid x \in X\}$. Thus, for some $x \in X$ we have $\mathbb{1}_B \in \phi(x)$, so, $\phi(x) = \top$. Furthermore, given such an x, for all $y \in X$ we have $\phi(y) \leq \phi(x)$, whence $\{y\} \subseteq \{x\}$; *ergo,* $y = x$. □

Theorem 2. *If $\mathbb{E}^{\mathbb{C}^{\mathrm{op}}}$ is an atomic topos then $\mathbb{C}$ is a groupoid in $\mathbb{E}$.*[4]

Proof. This is a direct consequence of the lemma since Barr and Diaconescu show (*op. cit.*) that for each object A in an atomic topos $\Gamma(\Omega^A)$ is isomoprphic to some $P(X)$. □

[3] A classical proof of this fact appears in [24] (§7 Example 2). Here we give a constructive proof that can be interpreted in any elementary topos.

[4] This is implicit in [12] VII.4. Barr and Diaconescu use their classical version of the lemma to prove this for a Boolean base topos, $\mathbb{E}$. (Corollary C3.5.2).

3.3 Continuous Truth

Definition 7. *A* topological site *is a category of open locales and continuous maps, including enough open inclusions (a basis for each locale), closed under finite limits, and equipped with the open cover topology.*

This definition differs from those of [18, 23]; Moerdijk and Reyes use topological spaces to construct their topological sites; we use locales. They, therefore, have to appeal to principles such as Bar Induction, or Fan Theorem, in the metatheory in order to show they hold in the topos of sheaves.

In [6] §4, we claimed, with proofs formalisable in HAH, that general principles of continuity, local choice and local compactness hold for these models. The proof of the key result, Proposition 4.1, presumes that a projection $\pi : \mathcal{W} \times \mathcal{U} \longrightarrow \mathcal{U}$ is a cover for the open cover topology. This is true if $\mathcal{W}$ is an open surjective locale, but obviously not in general—consider the empty space. The remedy is to require that the site $(\mathbb{C}, \mathcal{J})$ introduced in the opening sentence of §4 should be a topological site, whose objects are *open* locales. The results claimed in §4 are then valid if we take any elementary topos with natural number object as a base. We restate and prove Proposition 4.1.

Theorem 3. *Let $(\mathbb{C}, \mathcal{J})$ be a topological site. For any $\mathcal{X} \in \mathbb{C}$, the internal locale, $\mathfrak{X}$ represented by, $\mathcal{O}(\mathfrak{X})(\mathcal{U}) = \mathcal{O}(\mathcal{U} \times \mathcal{X})$, has enough points.*

Proof. We must show that, if $\mathcal{U} \Vdash \mathcal{K}$ covers $\mathrm{Pt}(\mathcal{W})$ and $\mathcal{U} \Vdash \mathcal{K}$ is closed, then $\mathcal{U} \Vdash \mathcal{W} \in \mathcal{K}$, for $\mathcal{U} \in \mathbb{C}$, and $\mathcal{W} \in \mathcal{O}(\mathfrak{X})$. We assume the hypotheses, and let

$$\mathbb{K} = \{\mathcal{U}_i \times \mathcal{W}_i \mid \mathcal{U}_i \Vdash \mathcal{W}_i \in \mathcal{K}{\restriction}\mathcal{U}_i\} \tag{16}$$

Clearly, $\mathbb{K}$ is a closed crible of $\mathcal{O}(\mathcal{U}) \times \mathcal{O}(\mathcal{X})$, that is, an open in $\mathcal{O}(\mathcal{U} \times \mathcal{X})$. We will show that $\mathbb{K}$ covers $\mathcal{U} \times \mathcal{W}$, so $\mathcal{U} \times \mathcal{W} \in \mathbb{K}$, which means that $\mathcal{U} \Vdash \mathcal{W} \in \mathcal{K}$. We pull back along the projection $\pi_2 : \mathcal{W} \times \mathcal{U} \longrightarrow \mathcal{U}$. This introduces a generic point of $\mathcal{W}$ given by $\pi_1 : \mathcal{W} \times \mathcal{U} \longrightarrow \mathcal{W}$. We have, $\mathcal{W} \times \mathcal{U} \Vdash \mathcal{K}{\restriction}\pi_2$ covers $\mathrm{Pt}(\mathcal{W})$, by persistence. In particular, the generic point is covered:

$$\mathcal{W} \times \mathcal{U} \Vdash K{\restriction}\pi_2 \text{ covers } \pi_1 \quad \text{that is, } \mathcal{W} \times \mathcal{U} \Vdash \exists V \in \mathcal{K}{\restriction}\pi_2.\ \pi_1 \in V\,.$$

V ranges over basic opens of $\mathcal{X}$. By unpacking the forcing definition, we see that

$$\mathbb{K}^* = \{\mathcal{W}_i \times \mathcal{U}_i \mid \text{for some } V \leq \mathcal{W},\ \mathcal{W}_i \times \mathcal{U}_i \Vdash V \in \mathcal{K}{\restriction}\pi_2 \wedge \pi_1 \in V\} \tag{17}$$

covers $\mathcal{W} \times \mathcal{U}$. We now show that every basic open $W \times U \in \mathbb{K}^*$ is in $\mathbb{K}$. Given U, V, W such that $W \times U \Vdash V \in \mathcal{K}{\restriction}\pi_2 \wedge \pi_1 \in V$ we must show $U \Vdash W \in \mathcal{K}$. First, $W \times U \Vdash \pi_1 \in V$ iff $W \leq V$, so, by monotonicity, $W \times U \Vdash W \in \mathcal{K}{\restriction}\pi_2$. Second, W is open surjective, so π_2 is an open surjection; viewed as a basic open in $\mathcal{O}(\mathfrak{X})$, W is constant, so $U \Vdash W \in \mathcal{K}$. □

We observe that this proof does not require that every object of $\mathbb{C}$ be surjective, indeed a subcategory of $\mathcal{L}$ with only open surjective objects will seldom be closed under limits. However, if W is a positive open of any open space, like the W in

the final steps of the proof just given, then W, as a subspace in its own right, is open surjective. Any open space is surjective in so far as its positive basis is inhabited. Nor do we require a full subcategory of $\mathcal{L}$, so the result should apply, for example, to a suitable localic version of the Euclidean topos, of sheaves over the category of closed subspaces of $\mathbb{R}^n$ with C^∞ functions, defined by Moerdijk and Reyes [23], and other smooth topoi.

Proposition 8. *If $(\mathbb{C}, \mathcal{J})$ is a topological site and $\mathcal{X} \in \mathbb{C}$, then the inclusion functor $i : \mathcal{O}(\mathcal{X}) \longrightarrow \mathbb{C}/\mathcal{X}$ has a right adjoint π which induces an adjoint retract pair of geometric morphisms $i : Sh(\mathcal{X}) \underset{\pi}{\overset{i}{\rightleftarrows}} Sh(\mathbb{C}/\mathcal{X})$. So, π^* preserves and reflects first-order logic, preserves exponentials, and preserves the sheaf of points of any T_1 formal space.*

Proof. The right adjoint π is given by $f \longmapsto \bigvee\{U \mid U \text{ factors through } f\}$, which satisfies the conditions of [23] Theorem 2.2 . So Proposition 7 applies. (The spatial counterpart of this proposition appears in M&M (§VII.10 Theorem 5 ff.).

To show a predicative reflection principle, we must choose a suitable topological site $(\mathbb{C}, \mathcal{J})$ with extension $\mathbb{E} = \mathrm{Sh}_{\mathbb{B}}(\mathbb{C}, \mathcal{J})$; then provide a representation of the iterated extension $\mathrm{Sh}_{\mathbb{E}}(\mathbb{C}, \mathcal{J})$ as $\mathrm{Sh}_{\mathbb{B}}(\mathbb{C}^\dagger, \mathcal{J}^\dagger)$, together with a left-exact functor $\mathbb{C}^\dagger \longrightarrow \mathbb{C}$ which preserves covers, and has the covering lifting property and a left-adjoint "right inverse".

Conjecture 1. Let $(\mathbb{C}, \mathcal{J})$ be the topological site of open subspaces of separable open locales with open maps, with the open cover topology, then $(\mathbb{C}^{\mathbf{2}}, \mathcal{J}^{\mathbf{2}})$ provides a representation of the corresponding internal site.

This general setting should provide reflection principles for several of the extensions introduced in [5]: (2.2) Independent Open Data , (2.3) Lawless Data, and (2.4) Spread Data. One key point is that the category of open spaces with open maps is closed under finite limits.

To extend such an account to more general examples, such as (2.5) Continuous Data, will require further analysis of the constructive theory of the category of open locales and continuous maps.

Acknowledgements. I am grateful to Bénabou for hosting my extended visit to *Séminaire Bénabou* in 1975, my introduction to fibrations; to André Joyal for formal spaces; to Thomas Streicher & Peter Johnstone for writing things down; and to Martin Hyland, Alex Simpson, John Longley, and Martín Escardo for more recent discussions.

References

1. Scott, D.S.: Extending the topological interpretation to intuistionistic analysis. Compos. Math. 20, 194–210 (1968)
2. Scott, D.S.: Extending the topological interpretation to intuitionistic analysis, II. In: Kino, A., Myhill, J., Vesley, R.E. (eds.) Intuitionism and Proof Theory: Proc. Buffalo, N.Y., 1968. Stud. Logic Found. Math., vol. 60, pp. 235–255. Elsevier B.V. (1970) ISBN: 978-0-7204-2257-3
3. Fourman, M.P.: Continuous truth. Abstr. Amer. Math. Soc. (1981) (abstract) 81-T-03-135
4. Fourman, M.P.: A model for the theory of choice sequences (CS). Abstr. Amer. Math. Soc., 183 (1982) (abstract) 82-T-03-80
5. Fourman, M.P.: Notions of choice sequence. In: [25], pp. 91–105
6. Fourman, M.P.: Continuous truth I, non-constructive objects. In: Lolli, G., Longo, G., Marcja, A. (eds.) Logic Colloquium: Florence, 1982. Stud. Logic Found. Math., pp. 161–180. North-Holland (1984)
7. van der Hoeven, G., Moerdijk, I.: On choice sequences determined by spreads. JSL 49, 908–916 (1984)
8. van Dalen, D. (ed.): Brouwer's Cambridge lectures on intuitionism. CUP (1981) ISBN: 0521234417
9. Xu, C., Escardó, M.: A constructive model of uniform continuity. In: Hasegawa, M. (ed.) TLCA 2013. LNCS, vol. 7941, pp. 236–249. Springer, Heidelberg (2013)
10. Fourman, M.P., Grayson, R.J.: Formal spaces. In: [25], pp. 107–122
11. Johnstone, P.T.: Sketches of an Elephant: a topos theory compendium. Oxford Logic Guides, vol. 43-44. OUP (2002) ISBN: 9780198524960
12. Joyal, A., Tierney, M.: An extension of the Galois theory of Grothendieck. Mem. Amer. Math. Soc., vol. 309. AMS (1984) ISBN: 0821823124
13. Mac Lane, S., Moerdijk, I.: Sheaves in Geometry and Logic; A First Introduction to Topos Theory. Springer (1992)
14. Grayson, R.J.: Forcing in intuitionistic systems without power-set. JSL 48, 670–682 (1983), `http://www.jstor.org/stable/2273459`
15. Streicher, T.: Fibred categories à la Jean Bénabou. PDF document (1999-2012), `http://www.mathematik.tu-darmstadt.de/~streicher/FIBR/FibLec.pdf`
16. Fourman, M.P., Scott, D.S.: Sheaves and logic. In: Fourman, M.P., Mulvey, C.J., Scott, D.S. (eds.) Applications of Sheaves. Lect. Notes Math., vol. 753, pp. 302–401. Springer (1979) ISBN: 0-387-09564-0
17. Grayson, R.J.: Concepts of general topology in constructive mathematics and sheaves. Ann. Math. Logic 20, 1–41 (1981)
18. Fourman, M.P.: T_1 spaces over topological sites. JPAA 27, 223–224 (1983)
19. Johnstone, P.T.: Stone Spaces. Cam. St. Adv. Math., vol. 3. CUP (1986) ISBN: 978-0-5213-3779-3
20. Johnstone, P.T.: Open maps of toposes. Manuscripta Math. 31, 217–247 (1980)
21. Hyland, J.: Function spaces in the category of locales. In: Continuous Lattices (Proc. Bremen workshop, 1979). Lect. Notes Math., pp. 264–281. Springer (1981)
22. Koellner, P.: On reflection principles. Ann. P. Appl. Logic 157, 206–219 (2009)
23. Moerdijk, I., Reyes, G.E.: Smooth spaces versus continuous spaces in models for synthetic differential geometry. JPAA 32, 143–176 (1984)
24. Barr, M., Diaconescu, R.: Atomic toposes. JPAA 17, 1–24 (1980)
25. Troelstra, A.S., van Dalen, D. (eds.): The L.E.J. Brouwer Centenary Symposium. Stud. Logic Found. Math., vol. 110. North-Holland (1982) ISBN: 0-444-86494-6

A Simple Separation Logic

Andreas Herzig

University of Toulouse, CNRS
Institut de recherche en informatique de Toulouse (IRIT)
Toulouse, France
www.irit.fr/~Andreas.Herzig

Abstract. The kinds of models that are usually considered in separation logic are structures such as words, trees, and more generally pointer structures (heaps). In this paper we introduce the separation logic of much simpler structures, viz. sets. The models of our set separation logic are nothing but valuations of classical propositional logic. Separating a valuation V consists in splitting it up into two partial valuations v_1 and v_2. Truth of a formula $\varphi_1 * \varphi_2$ in a valuation V can then be defined in two different ways: first, as truth of φ_1 in *all* total extensions of v_1 and truth of φ_2 in *all* total extensions of v_2; and second, as truth of φ_1 in *some* total extension of v_1 and truth of φ_2 in *some* total extension of v_2. The first is an operator of separation of resources: the update of $\varphi_1 * \varphi_2$ by ψ is the conjunction of the update of φ_1 by ψ and the update of φ_2 by ψ; in other words, $\varphi_1 * \varphi_2$ can be updated independently. The second is an operator of separation of processes: updates by $\psi_1 * \psi_2$ can be performed independently. We show that the satisfiability problem of our logic is decidable in polynomial space (PSPACE). We do so by embedding it into dynamic logic of propositional assignments (which is PSPACE complete). We moreover investigate its applicability to belief update and belief revision, where the separation operators allow to formulate natural requirements on independent pieces of information.

1 Introduction

Separation logics [7, 13, 17] have a modal operator $*$ which allows to talk about the separation of resources. Basically, the formula $\varphi_1 * \varphi_2$ is true in the model M if M can be split into two parts M_1 and M_2 such that φ_1 is true in M_1 and φ_2 is true in M_2. The kinds of models that are usually considered in separation logic are structures such as words, trees, and more generally pointer structures (heaps). The separation logics of such structures are often undecidable. In this paper we investigate the separation logic of much simpler structures, viz. sets. We call our logic *set separation logic*, abbreviated SSL. The models of SSL are nothing but valuations of classical propositional logic. Separating a valuation V consists in splitting it up into two partial valuations v_1 and v_2. Then separability of φ_1 and φ_2 in a valuation V can be defined in two different ways: first, as truth of φ_1 in *all* total extensions of v_1 and truth of φ_2 in *all* total extensions of v_2; and second, as truth of φ_1 in *some* total extension of v_1 and truth of φ_2 in *some* total extension of v_2. We respectively denote these two separation operators by $\dot{\wedge}$ and $\dot{\|}$. We chose the symbol $\dot{\wedge}$ due to its analogy with the symbol of disjoint union $\dot{\cup}$, and we chose the symbol $\dot{\|}$ because $\|$ denotes parallel execution.

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 168–178, 2013.

We show that the satisfiability problem of set separation logic is decidable in polynomial space (PSPACE). We do so by embedding SSL into dynamic logic of propositional assignments DL-PA [2], whose star-free fragment is PSPACE complete. This contrasts with separation logics having the implicational connective $-\!\!*$, which are often undecidable even in the proopositional language [4, 11].

Our initial motivation to investigate separation operators was that they can be given an interesting interpretation in the context of the revision and update of propositional belief bases: we consider that when φ_1 and φ_2 are separable then they are independent pieces of information. This naturally leads to the following requirements.

- We suppose that $\dot{\wedge}$ expresses independence of resources: the update of $\varphi_1 \dot{\wedge} \varphi_2$ by ψ is the conjunction of the update of φ_1 by ψ and the update of φ_2 by ψ; in other words, $\varphi_1 \dot{\wedge} \varphi_2$ can be updated independently.
- We suppose that $\dot{\|}$ expresses independence of processes: the update of φ by $\psi_1 \dot{\|} \psi_2$ is the parallel update of φ by ψ_1 and by ψ_2; in other worlds, updates by $\psi_1 \dot{\|} \psi_2$ can be performed independently.

This extends previous approaches by Parikh, Makinson and others that are based on splitting languages [3, 10, 14]. We investigate the compatibility of existing belief change operations with the above two requirements.

The paper is organised as follows. In Section 2 we introduce set separation logic SSL. In Section 3 we provide a PSPACE upper bound for both its model checking and its satisfiability problem. In Section 4 we discuss the relation between SSL and language splitting-based belief change. Section 5 concludes.

2 Set Separation Logic SSL

Throughout the paper we use the following conventions.

$\mathbb{P} = \{p, q, \ldots\}$ is a countable set of propositional variables. The set $\{P_1, P_2\}$ is a partition of $\mathbb{P}$ iff $P_1 \cup P_2 = \mathbb{P}$ and $P_1 \cap P_2 = \emptyset$.

A *valuation* is a total function from $\mathbb{P}$ to $\{0, 1\}$. We use V, V_1, ... for valuations. Two valuations V and V' *agree on the set of variables* $P \subseteq \mathbb{P}$, if both give the same truth value to each of the variables in P: $V \sim_P V'$ iff $V(p) = V'(p)$ for every $p \in P$.

A *partial valuation* is a partial function from $\mathbb{P}$ to $\{0, 1\}$. For a valuation $V : \mathbb{P} \longrightarrow \{0, 1\}$ and a set of propositional variables $P \subseteq \mathbb{P}$, the *restriction of* V *to* P is the partial function whose domain is P, noted $V|_P$. We use v, v_1, ... for partial valuations. The total valuation V is an *extension* of the partial valuation v if $V(p) = v(p)$ for every $p \in \mathsf{dom}(v)$.[1]

The *language of* SSL is defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \dot{\wedge} \varphi \mid \varphi \dot{\|} \varphi$$

where p ranges over the set of propositional variables $\mathbb{P}$. The formula $\varphi \dot{\wedge} \psi$ may be read "φ and ψ are statically separable" and $\varphi \dot{\|} \psi$ may be read "φ and ψ are dynamically

[1] We might as well define valuations to be sets of propositional variables. However, it would have been less elegant to account for partial valuations under such a presentation.

separable". Our intuition is the following: when $\varphi \mathbin{\dot\wedge} \psi$ is true then the conjunction of φ and ψ can be updated separately; and when $\varphi \mathbin{\dot\|} \psi$ is true then updating by the conjunction of φ and ψ can be performed in parallel.

We abbreviate the logical connectives $\wedge$, $\rightarrow$ and $\leftrightarrow$ in the usual way.

The truth conditions are as follows:

$$
\begin{aligned}
V \models p \ &\text{iff}\ V(p) = 1;\\
V \models \neg\varphi \ &\text{iff}\ V \not\models \varphi;\\
V \models \varphi_1 \wedge \varphi_2 \ &\text{iff}\ V \models \varphi_1 \text{ and } V \models \varphi_2;\\
V \models \varphi_1 \mathbin{\dot\wedge} \varphi_2 \ &\text{iff}\ \text{there is a partition } \{P_1, P_2\} \text{ of } \mathbb{P} \text{ such that}\\
&\qquad V_1 \models \varphi_1 \text{ for every valuation } V_1 \text{ agreeing with } V \text{ on } P_1 \text{ and}\\
&\qquad V_2 \models \varphi_2 \text{ for every valuation } V_2 \text{ agreeing with } V \text{ on } P_2;\\
V \models \varphi_1 \mathbin{\dot\|} \varphi_2 \ &\text{iff}\ \text{there is a partition } \{P_1, P_2\} \text{ of } \mathbb{P} \text{ such that}\\
&\qquad V_1 \models \varphi_1 \text{ for some valuation } V_1 \text{ agreeing with } V \text{ on } P_1 \text{ and}\\
&\qquad V_2 \models \varphi_2 \text{ for some valuation } V_2 \text{ agreeing with } V \text{ on } P_2.
\end{aligned}
$$

The conditions for the two separation operators can be reformulated in terms of partial valuations as follows:

$$
\begin{aligned}
V \models \varphi_1 \mathbin{\dot\wedge} \varphi_2 \ &\text{iff}\ \text{there is a partition } \{P_1, P_2\} \text{ of } \mathbb{P} \text{ such that}\\
&\qquad V_1 \models \varphi_1 \text{ for every extension } V_1 \text{ of } V|_{P_1} \text{ and}\\
&\qquad V_2 \models \varphi_2 \text{ for every extension } V_2 \text{ of } V|_{P_2};\\
V \models \varphi_1 \mathbin{\dot\|} \varphi_2 \ &\text{iff}\ \text{there is a partition } \{P_1, P_2\} \text{ of } \mathbb{P} \text{ such that}\\
&\qquad V_1 \models \varphi_1 \text{ for some extension } V_1 \text{ of } V|_{P_1} \text{ and}\\
&\qquad V_2 \models \varphi_2 \text{ for some extension } V_2 \text{ of } V|_{P_2}.
\end{aligned}
$$

Some observations:

- In the truth condition for $\mathbin{\dot\wedge}$, the exhaustiveness condition $P_1 \cup P_2 = \mathbb{P}$ can be dropped. If we dropped the disjointness condition $P_1 \cap P_2 = \emptyset$ then $\varphi \mathbin{\dot\wedge} \psi$ trivialises to the conjunction $\varphi \wedge \psi$.
- In the truth condition for $\mathbin{\dot\|}$, if we drop the exhaustiveness condition $P_1 \cup P_2 = \mathbb{P}$ then $\varphi \mathbin{\dot\|} \psi$ trivialises to the consistency of both φ and ψ.

Here are some examples. Let V_{pq} be a valuation such that $V_{pq}(p) = V_{pq}(q) = 1$ and let $V_{p\bar q}$ be a valuation such that $V_{p\bar q}(p) = 1$ and $V_{p\bar q}(q) = 0$. Then we have:

$$
\begin{array}{lll}
V_{pq} \models p \mathbin{\dot\wedge} q & V_{pq} \models p \mathbin{\dot\|} q & V_{pq} \models (\neg p) \mathbin{\dot\|} (\neg q)\\
V_{pq} \models p \mathbin{\dot\wedge} (p \vee q) & V_{pq} \models p \mathbin{\dot\|} (\neg p \wedge \neg q) & V_{pq} \not\models \neg p \mathbin{\dot\|} (\neg p \wedge \neg q)\\
V_{p\bar q} \not\models p \mathbin{\dot\wedge} (p \vee q) & V_{p\bar q} \models p \mathbin{\dot\|} (p \vee q) & V_{p\bar q} \models \neg p \mathbin{\dot\|} (p \vee q)\\
V_{p\bar q} \models p \mathbin{\dot\wedge} (p \vee \neg q) & V_{p\bar q} \models p \mathbin{\dot\|} (p \vee \neg q) & V_{p\bar q} \not\models \neg p \mathbin{\dot\|} \neg(p \vee q)
\end{array}
$$

Satisfiability and validity are defined as usual. The following formula schemas are valid:

$$\varphi_1 \mathbin{\dot\wedge} \varphi_2 \leftrightarrow \varphi_2 \mathbin{\dot\wedge} \varphi_1 \qquad \varphi_1 \mathbin{\dot\|} \varphi_2 \leftrightarrow \varphi_2 \mathbin{\dot\|} \varphi_1$$

$$\varphi_1 \mathbin{\dot\wedge} \varphi_2 \rightarrow \varphi_2 \wedge \varphi_1 \qquad \varphi_1 \wedge \varphi_2 \rightarrow \varphi_2 \mathbin{\dot\|} \varphi_1$$

$$\top \mathbin{\dot\wedge} \varphi \leftrightarrow \varphi \qquad \top \mathbin{\dot\|} \varphi \leftrightarrow \begin{cases} \top & \text{if } \varphi \text{ is satisfiable} \\ \bot & \text{otherwise} \end{cases}$$

As the last line shows, consistency of a formula φ can be expressed in the language of SSL by the formula $\top \mathbin{\dot\|} \varphi$. Here are two inference rules preserving validity:

$$\frac{\varphi \rightarrow \psi}{(\varphi \mathbin{\dot\wedge} \chi) \rightarrow (\psi \mathbin{\dot\wedge} \chi)} \qquad \frac{\varphi \rightarrow \psi}{(\varphi \mathbin{\dot\|} \chi) \rightarrow (\psi \mathbin{\dot\|} \chi)}$$

The following equivalences are valid, where the propositional variables p and q are supposed to be different:

$$p \mathbin{\dot\wedge} p \leftrightarrow \bot \qquad p \mathbin{\dot\|} p \leftrightarrow p$$

$$p \mathbin{\dot\wedge} \neg p \leftrightarrow \bot \qquad p \mathbin{\dot\|} \neg p \leftrightarrow \top$$

$$p \mathbin{\dot\wedge} q \leftrightarrow p \wedge q \qquad p \mathbin{\dot\|} q \leftrightarrow \top$$

$$p \mathbin{\dot\wedge} (p \vee q) \leftrightarrow p \wedge q \qquad p \mathbin{\dot\|} (p \vee q) \leftrightarrow \top$$

$$(p \vee q) \mathbin{\dot\wedge} (p \vee q) \leftrightarrow p \wedge q \qquad (p \vee q) \mathbin{\dot\|} (p \vee q) \leftrightarrow \top$$

3 Complexity

In this section we establish an upper bound for the complexity of both model checking and satisfiability checking of set separation logic. We prove this by showing that both $\varphi_1 \mathbin{\dot\wedge} \varphi_2$ and $\varphi_1 \mathbin{\dot\|} \varphi_2$ can be expressed in dynamic logic of propositional assignments DL-PA (that we have recently proposed with Philippe Balbiani and Nicolas Troquard in [2]) by equivalent formulas whose length is polynomial in the length of $\varphi_1 \mathbin{\dot\wedge} \varphi_2$ and $\varphi_1 \mathbin{\dot\|} \varphi_2$, respectively.

3.1 DL-PA: Dynamic Logic of Propositional Assignments

The language of DL-PA is defined by the following grammar:

$$\pi ::= p{\leftarrow}\top \mid p{\leftarrow}\bot \mid \varphi? \mid \pi;\pi \mid \pi \cup \pi \mid \pi^*$$
$$\varphi ::= p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\pi\rangle\varphi$$

where p ranges over $\mathbb{P}$. So an *atomic program* of the language of DL-PA is a program of the form $p{\leftarrow}\varphi$. The operators of sequential composition (";"), nondeterministic composition ("$\cup$"), unbounded iteration ("*", the Kleene star), and test ("?") are familiar from PDL.

We define $\mathbb{P}_\varphi$ to be the set of variables from $\mathbb{P}$ occurring in formula φ, and we define $\mathbb{P}_\pi$ to be the set of variables from $\mathbb{P}$ occurring in program π. For example, $\mathbb{P}_{p\leftarrow q \cup p\leftarrow\neg q} = \{p, q\} = \mathbb{P}_{\langle p\leftarrow\bot\rangle q}$.

We abbreviate the logical connectives $\wedge$, $\rightarrow$ and $\leftrightarrow$ in the usual way. Moreover, $[\pi]\varphi$ abbreviates $\neg\langle\pi\rangle\neg\varphi$. The program `skip` abbreviates $\top?$ ("nothing happens") and the program $p{\leftarrow}q$ abbreviates $(q?; p{\leftarrow}\top) \cup (\neg q?; p{\leftarrow}\bot)$ ("p gets the truth value of q").

DL-PA programs are interpreted by means of a (unique) *relation between valuations*: atomic programs $p{\leftarrow}\top$ and $p{\leftarrow}\bot$ update valuations in the obvious way, and complex programs are interpreted just as in PDL by mutual recursion. Table 1 gives the interpretation of the DL-PA connectives.

Table 1. Interpretation of the DL-PA connectives

$$
\begin{aligned}
||p{\leftarrow}\top|| &= \{(V, V') \ : \ V'(p) = 1 \text{ and } V' \text{ agrees with } V \text{ on } \mathbb{P} \setminus \{p\}\} \\
||p{\leftarrow}\bot|| &= \{(V, V') \ : \ V'(p) = 0 \text{ and } V' \text{ agrees with } V \text{ on } \mathbb{P} \setminus \{p\}\} \\
||\pi; \pi'|| &= ||\pi|| \circ ||\pi'|| \\
||\pi \cup \pi'|| &= ||\pi|| \cup ||\pi'|| \\
||\pi^*|| &= \bigcup_{k \in \mathbb{N}_0} (||\pi||)^k \\
||\varphi?|| &= \{(V, V) \ : \ V \in ||\varphi||\} \\
||p|| &= \{V \ : \ V(p) = 1\} \\
||\top|| &= 2^{\mathbb{P}} \\
||\bot|| &= \emptyset \\
||\neg\varphi|| &= 2^{\mathbb{P}} \setminus ||\varphi|| \\
||\varphi \vee \psi|| &= ||\varphi|| \cup ||\psi|| \\
||\langle\pi\rangle\varphi|| &= \{V \ : \ \text{there is } V' \text{ s.t. } (V, V') \in ||\pi|| \text{ and } V' \in ||\varphi||\}
\end{aligned}
$$

A formula φ is DL-PA *valid* if $||\varphi|| = 2^{\mathbb{P}}$, and φ is DL-PA *satisfiable* if $||\varphi|| \neq \emptyset$. For example, the formulas $\langle p{\leftarrow}\top\rangle\top$ and $\langle p{\leftarrow}\top\rangle\varphi \leftrightarrow \neg\langle p{\leftarrow}\top\rangle\neg\varphi$ are DL-PA valid. Other examples of DL-PA validities are $\langle p{\leftarrow}\top\rangle p$ and $\langle p{\leftarrow}\bot\rangle\neg p$. Observe that if p does not occur in φ then both $\varphi \rightarrow \langle p{\leftarrow}\top\rangle\varphi$ and $\varphi \rightarrow \langle p{\leftarrow}\bot\rangle\varphi$ are valid. This is due to the following semantical property.

Proposition 1. *Suppose $p \notin \mathbb{P}_\varphi$, i.e., p does not occur in φ. Then $\varphi \in ||V \cup \{p\}||$ iff $\varphi \in ||V \setminus \{p\}||$.*

Theorem 1 ([2]). *For the full language, both the* DL-PA *satisfiability problem and the* DL-PA *model checking problem are EXPTIME complete.*

For the star-free fragment, both the DL-PA *satisfiability problem and the* DL-PA *model checking problem are PSPACE complete.*

3.2 Embedding Set Separation Logic into DL-PA

We now give a polynomial transformation mapping set separation logic formulas φ_0 into DL-PA formulas.

Let P' be the set of variables p' such that p is in P and p' is fresh: p' does not occur in the formula φ_0 under consideration. The following abbreviations will be useful:

$$\begin{aligned}
\pm p &= p{\leftarrow}\top \cup p{\leftarrow}\bot \\
\mathsf{changeSome}(\{p_1,\cdots,p_n\}) &= \pm p_1;\cdots;\pm p_n \\
\mathsf{store}(\{p_1,\cdots,p_n\}) &= p'_1{\leftarrow}p_1;\cdots;p'_n{\leftarrow}p_n \\
\mathsf{retrieve}(\{p_1,\cdots,p_n\}) &= p_1{\leftarrow}p'_1;\cdots;p_n{\leftarrow}p'_n \\
\mathsf{changeSomeMarked}(\{p_1,\cdots,p_n\}) &= \neg(p_1 \leftrightarrow p'_1)?\cup(p_1 \leftrightarrow p'_1?;\pm p_1); \\
&\quad\cdots \\
&\quad \neg(p_n \leftrightarrow p'_n)?\cup(p_n \leftrightarrow p'_n?;\pm p_n) \\
\mathsf{changeSomeUnmarked}(\{p_1,\cdots,p_n\}) &= p_1 \leftrightarrow p'_1?\cup(\neg(p_1 \leftrightarrow p'_1)?;\pm p_1); \\
&\quad\cdots \\
&\quad p_n \leftrightarrow p'_n?\cup(\neg(p_n \leftrightarrow p'_n)?;\pm p_n) \\
\mathsf{changeRestAndRestore}(\{p_1,\cdots,p_n\}) &= ((p_1 \leftrightarrow p'_1?;\pm p_1)\cup(\neg(p_1 \leftrightarrow p'_1)?;p_1{\leftarrow}p'_1)); \\
&\quad\cdots \\
&\quad ((p_n \leftrightarrow p'_n?;\pm p_n)\cup(\neg(p_n \leftrightarrow p'_n)?;p_n{\leftarrow}p'_n))
\end{aligned}$$

The program $\mathsf{changeSome}(P)$ nondeterministically changes the truth value of some variables in P. The program $\mathsf{store}(P)$ stores the truth value of each variable p by means of a fresh variable p', and $\mathsf{retrieve}(P)$ reestablishes that 'old' value. When p and p' have different truth values then we say that p is marked; else we say that p is unmarked. The program $\mathsf{changeSomeMarked}(P)$ arbitrarily changes only the unmarked variables. The other way round, the program $\mathsf{changeSomeUnmarked}(P)$ leaves every unmarked $p \in P$ unchanged and arbitrarily changes the marked p's.

Observe that each of the above programs has length linear in the cardinality n of the set of propositional variables $\{p_1,\ldots,p_n\}$.

The next two propositions provide an embedding of set separation logic into DL-PA.

Proposition 2. *Let φ_1 and φ_2 be two propositional formulas. Let $P = \mathbb{P}_{\varphi_1} \cap \mathbb{P}_{\varphi_2}$. Let P' be the set of variables p' such that p is in P and p' is fresh: p' does not occur in the formula under consideration. Then the formula $\varphi_1 \,\dot{\|}\, \varphi_2$ is equivalent to the* DL-PA *formula*

$$\begin{aligned}
\langle \mathit{store}(P);\mathit{changeSome}(P)\rangle\,(\,&\langle \mathit{changeSome}(\mathbb{P}_{\varphi_1}\setminus P)\rangle\varphi_1 \wedge \\
&\langle \mathit{changeRestAndRestore}(P)\rangle\langle \mathit{changeSome}(\mathbb{P}_{\varphi_2}\setminus P)\rangle\varphi_2\,)
\end{aligned}$$

Proposition 3. *Let φ_1 and φ_2 be two propositional formulas. Let $P = \mathbb{P}_{\varphi_1} \cap \mathbb{P}_{\varphi_2}$. Let P' be the set of variables p' such that p is in P and p' is fresh: p' does not occur in the formula under consideration. Then the formula $\varphi_1 \,\dot{\wedge}\, \varphi_2$ is equivalent to the following* DL-PA *formula:*

$$\begin{aligned}
\langle \mathit{store}(P);\mathit{changeSome}(P')\rangle\,(\,&[\mathit{changeSomeMarked}(P)]\varphi_1 \wedge \\
&[\mathit{changeSomeUnmarked}(P)]\varphi_2\,)
\end{aligned}$$

Intuitively, after the program store(P) has stored the value of each of the elements of P, the program changeSome(P') allows to (nondeterministically) identify a subset of P: those p whose value differs from its copy p'. We consider that these 'marked' variables are those of the partial valuation for φ_1, while the complementary, unmarked variables make up the partial valuation for φ_2.

This can be turned more formally into a transformation from the language of set separation logic into the language of DL-PA. The transformation is clearly linear in the size of the original formula φ_0.

The codomain of the transformation is the star-free fragment of DL-PA. As both model checking and satisfiability checking in DL-PA are PSPACE complete, it follows that model checking and satisfiability checking in set separation logic are in PSPACE. It remains to investigate the lower bounds.

4 Separability in the Context of Belief Change Operations

As we have mentioned in the introduction, one can use the SSL operators to formulate new postulates for belief change operations such as AGM belief revision operators [1,6] and KM update operators [8, 9]. We investigate this now in more depth.

4.1 The Basic Belief Change Postulates

Let $\circ$ be a belief change operator and let β and ψ be boolean formulas. (We use β for the base and ψ for the input.) $\beta \circ \psi$ is the result of incorporating the input ψ into the base β. Both revision and update operations were mainly studied from a semantical perspective: $\beta \circ \psi$ is viewed as a set of valuations.

Katsuno and Mendelzon promoted the distinction between belief update and belief revision [9]. Their idea is that update keeps track of changes in the world while revision corrects errors about an unchanged world. This can be illustrated by the revised and updated edition of a dictionary: we say that it has been revised because past errors have been corrected, and we say that it has been updated because new usages of existing words have been added to it and outdated usages have been dropped. Traditionally, $\beta \diamond \psi$ denotes the update of the base β by the input ψ and $\beta * \psi$ denotes the revision of the base β by the input ψ.

Alchourrón, Gärdenfors and Makinson designed a set of postulates for belief revision operations (the so-called AGM postulates), and Katsuno and Mendelzon designed a set of postulates for belief update operations (the so-called KM-postulates). The following postulates are common to both kinds of operations:

(RE) if $||\beta_1|| = ||\beta_2||$ and $||\psi_1|| = ||\psi_2||$ then $\beta_1 \circ \psi_1 = \beta_2 \circ \psi_2$
(SUCCESS) $\beta \circ \psi \subseteq ||\psi||$
(PRES_w) if $||\beta|| \subseteq ||\psi||$ then $\beta \circ \psi = ||\beta||$

where $||\varphi||$ is the set of valuations where φ is true (just as in Section 3.1). We call the above the *basic belief change postulates*.

RE is a postulate of insensitivity to syntax. SUCCESS says that belief change is successful: the input has priority. PRES_w is a weak preservation postulate: if the input is

already in the base then the base should not change. AGM revision operations moreover satisfy a strengthening of PRES_w:

$$\text{(PRES) if } ||\beta|| \cap ||\psi|| \neq \emptyset \text{ then } \beta \circ \psi = ||\beta \wedge \psi||$$

4.2 Belief Change Operations and Language Splitting

It has been observed by many that the drastic update operation defined as

$$\beta \circ \psi = \begin{cases} ||\beta|| & \text{if } ||\beta|| \subseteq ||\psi|| \\ ||\psi|| & \text{otherwise} \end{cases}$$

satisfies the KM postulates. Similarly, the following drastic revision operation

$$\beta \circ \psi = \begin{cases} ||\beta \wedge \psi|| & \text{if } ||\beta|| \cap ||\psi|| \neq \emptyset \\ ||\psi|| & \text{otherwise} \end{cases}$$

satisfies the AGM postulates. In order to exclude such operations, Parikh, Makinson and others argued for a further postulate of relevance [3, 10, 14]. Its formulation refers to the syntax of the base and the input.

$$\text{(REL) } (\beta_1 \wedge \beta_2) \circ \psi = (\beta_1 \circ \psi) \cap (\beta_2 \circ \psi) \qquad \text{if } \mathbb{P}_{\beta_1} \cap \mathbb{P}_{\beta_2} = \emptyset$$

Just as in Section 2, $\mathbb{P}_\varphi$ denotes the set of propositional variables occurring in the boolean formula φ. Therefore $\mathbb{P}_{\beta_1} \cap \mathbb{P}_{\beta_2} = \emptyset$ means that the signatures of β_1 and β_2 are disjoint: the languages of β_1 and β_2 can be split. Each of the above drastic operations violates the postulate REL.

4.3 Separation-Based Belief Change Operations

In the same spirit and as already stated informally in the introduction, the SSL operators enable us to go beyond such syntax-based postulates and strengthen the above relevance postulate REL. The strengthening comes in a static version and in a dynamic version:

$$\begin{aligned} (\text{REL}_s)\ & (\beta_1 \dot{\wedge} \beta_2) \circ \psi = (\beta_1 \circ \psi) \cap (\beta_2 \circ \psi) \\ (\text{REL}_d)\ & \beta \circ (\psi_1 \dot{||} \psi_2) = (\beta \circ \psi_1) \circ \psi_2 \\ & \qquad\qquad = (\beta \circ \psi_2) \circ \psi_1 \end{aligned}$$

where $\circ$ is any belief change operation, be it update or revision.[2] The static relevance postulate REL_s says that when the bases β_1 and β_2 are statically separable then they can be updated separately. Its dynamic counterpart REL_d says that when the inputs ψ_1 and ψ_2 are dynamically separable then the update can be performed in parallel (or rather, in an interleaving fashion).

It turns out that both postulates are violated by any AGM revision operation and and any KM update operation.

[2] Strictly speaking, REL_d requires to build a formula representing the updates $\beta \circ \psi_1$ and $\beta \circ \psi_2$, as usually done in the KM framework.

Proposition 4. *There is no operation $\circ$ satisfying both the basic belief change postulates and REL$_s$.*

Proof. Suppose $\circ$ satisfies the basic belief change postulates and REL$_s$. Consider the base $\beta = (p \vee q) \dot{\wedge} (p \vee q)$ and the input $\psi = p \vee q$. We have seen above that β is equivalent to $p \wedge q$, and we therefore have:

$$\begin{array}{rll} \beta \circ \psi & = (p \vee q) \dot{\wedge} (p \vee q) \circ p \vee q & \\ & = p \wedge q \circ p \vee q & \text{(by RE)} \\ & = \|p \wedge q\| & \text{(by PRES}_w\text{)} \end{array}$$

This is incompatible with what postulate REL$_s$ gives us:

$$\begin{array}{rll} \beta \circ \psi & = (p \vee q) \dot{\wedge} (p \vee q) \circ p \vee q & \\ & = p \vee q \circ p \vee q \cap p \vee q \circ p \vee q & \text{(by REL}_s\text{)} \\ & = \|p \vee q\| \cap \|p \vee q\| & \text{(by PRES}_w\text{)} \\ & = \|p \vee q\| & \end{array}$$

□

Proposition 5. *There is no operation $\circ$ satisfying both the basic belief change postulates and REL$_d$.*

Proof. Suppose $\circ$ satisfies the KM postulates and REL$_d$. Consider the base $\beta = \neg p$ and the input $\psi = \neg p \,\dot{\|}\, p$. We have seen above that ψ is equivalent to $\top$, and we therefore have:

$$\begin{array}{rll} \beta \circ \psi & = \neg p \circ \neg p \,\dot{\|}\, p & \\ & = \neg p \circ \top & \text{(by RE)} \\ & = \|\neg p\| & \text{(by PRES}_w\text{)} \end{array}$$

This is incompatible with what postulate REL$_d$ gives us:[3]

$$\begin{array}{rll} \beta \circ \psi & = \neg p \circ \neg p \,\dot{\|}\, p & \\ & = (\neg p \circ \neg p) \diamond p & \text{(by REL}_d\text{)} \\ & = \neg p \circ p & \text{(by PRES}_w\text{)} \\ & \subseteq \|p\| & \text{(by SUCCESS)} \end{array}$$

Incompatibility is the case because the set of valuations where $\neg p$ is true is non empty. □

5 Discussion and Conclusion

We have introduced a simple version of separation logic working on sets (alias propositional valuations) that we have called set separation logic, SSL. Our logic has two separation operators: $\dot{\wedge}$ allows to separate resources, and $\dot{\|}$ allows to separate updates. We have shown that in our logic, both model checking and satisfiability checking can be done in polynomial space. We conjecture that the PSPACE upper bound that we have established coincides with the lower bound, but this remains to be proved. We would also like to provide an axiomatisation.

[3] We recall that the second line of the proof is formulated sloppily: instead of the set of valuations $\neg p \circ \neg p$ there should be a formula representing that valuation.

In the last part of the paper we have investigated the relation between SSL and belief change operations. We have formulated two postulates that appear to be natural and have shown that they are nevertheless incompatible with both AGM belief revision operations and KM belief update operations.

The problem of belief change respecting separation that we have studied in the last section is related to the frame problem in artificial intelligence [12]. Reiter's solution to that problem [15, 16] is by now widely accepted for actions without ramifications, i.e., without side effects. In joint work with Hans van Ditmarsch and Tiago de Lima [5] we have recently shown that Reiter's solution can be mapped to dynamic logics with propositional assignments DL-PA. Given that set separation logic can be embedded into DL-PA, it is immediate to extend it by propositional assignments.

It would be interesting to add the implicational connective $-\!\!*$ of separation logic to SSL (which should lead to undecidability given the results of [4, 11]). it is however not clear how the semantics of $-\!\!*$ can be defined in the framework of valuations.

Acknowledgements. The work in this paper was done in the framework of the ANR project DynRes (Dynamic Resources and Separation and Update Logics, project no. ANR-11-BS02-011). I would like to thank the members of the project for their comments of a (very preliminary) presentation of the ideas that are worked out in more detail here. Thanks are also due to the reviewers of WoLLIC 2013 whose comments I took into account as far as I could.

References

[1] Alchourrón, C., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. J. of Symbolic Logic 50, 510–530 (1985)

[2] Balbiani, P., Herzig, A., Troquard, N.: Dynamic logic of propositional assignments: a well-behaved variant of PDL. In: Kupferman, O. (ed.) Logic in Computer Science (LICS), New Orleans, June 25-28. IEEE (2013), http://www.ieee.org/

[3] Bienvenu, M., Herzig, A., Qi, G.: Prime implicate-based belief revision operators. In: Ghallab, M., Spyropoulos, C.D., Fakotakis, N., Avouris, N. (eds.) European Conference on Artificial Intelligence (ECAI), Patras, Greece, pp. 741–742. IOS Press (July 2008)

[4] Brotherston, J., Kanovich, M.I.: Undecidability of propositional separation logic and its neighbours. In: Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, Edinburgh, United Kingdom, July 11-14, pp. 130–139. IEEE Computer Society (2010)

[5] van Ditmarsch, H., Herzig, A., de Lima, T.: From Situation Calculus to Dynamic Logic. Journal of Logic and Computation 21(2), 179–204 (2011), http://logcom.oxfordjournals.org/content/21/2/179.abstract?etoc

[6] Gärdenfors, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. MIT Press (1988)

[7] Ishtiaq, S.S., O'Hearn, P.W.: BI as an assertion language for mutable data structures. In: Hankin, C., Schmidt, D. (eds.) POPL, pp. 14–26. ACM (2001)

[8] Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. Artificial Intelligence 52, 263–294 (1991)

[9] Katsuno, H., Mendelzon, A.O.: On the difference between updating a knowledge base and revising it. In: Gärdenfors, P. (ed.) Belief Revision, pp. 183–203. Cambridge University Press (1992); Preliminary version in Allen, J.A., Fikes, R., Sandewall, E. (eds.) Principles of Knowledge Representation and Reasoning: Proc. 2nd Int. Conf., pp. 387–394. Morgan Kaufmann Publishers (1991)
[10] Kourousias, G., Makinson, D.: Parallel interpolation, splitting, and relevance in belief change. Journal of Symbolic Logic 72(3), 994–1002 (2007)
[11] Larchey-Wendling, D., Galmiche, D.: The undecidability of boolean bi through phase semantics. In: Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, Edinburgh, United Kingdom, July 11-14, pp. 140–149. IEEE Computer Society (2010)
[12] McCarthy, J., Hayes, P.J.: Some philosophical problems from the standpoint of artificial intelligence. In: Meltzer, B., Mitchie, D. (eds.) Machine Intelligence, vol. 4, pp. 463–502. Edinburgh University Press (1969)
[13] O'Hearn, P.W., Reynolds, J.C., Yang, H.: Local reasoning about programs that alter data structures. In: Fribourg, L. (ed.) CSL 2001. LNCS, vol. 2142, pp. 1–19. Springer, Heidelberg (2001)
[14] Parikh, R.: Beliefs, belief revision, and splitting languages. In: Moss, L.S., Ginzburg, J., de Rijke, M. (eds.) Logic, Language, and Computation, vol. 2, pp. 266–278. CSLI Publications (1999)
[15] Reiter, R.: The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In: Lifschitz, V. (ed.) Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy, pp. 359–380. Academic Press, San Diego (1991)
[16] Reiter, R.: Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. The MIT Press (2001)
[17] Reynolds, J.C.: Separation logic: A logic for shared mutable data structures. In: Proceedings of the 17th IEEE Symposium on Logic in Computer Science (LICS 2002), Copenhagen, Denmark, July 22-25, pp. 55–74. IEEE Computer Society (2002)

Independence in Database Relations

Juha Kontinen[1], Sebastian Link[2], and Jouko Väänänen[1,3]

[1] Department of Mathematics and Statistics, University of Helsinki, Finland
[2] Department of Computer Science, The University of Auckland, New Zealand
[3] Institute for Logic, Language and Computation, University of Amsterdam, The Netherlands

Abstract. We investigate the implication problem for independence atoms $X \perp Y$ of disjoint attribute sets X and Y on database schemata. A relation satisfies $X \perp Y$ if for every X-value and every Y-value that occurs in the relation there is some tuple in the relation in which the X-value occurs together with the Y-value. We establish an axiomatization by a finite set of Horn rules, and derive an algorithm for deciding the implication problem in low-degree polynomial time in the input. We show how to construct Armstrong relations which satisfy an arbitrarily given set of independence atoms and violate every independence atom not implied by the given set. Our results establish independence atoms as an efficient subclass of embedded multivalued data dependencies which are not axiomatizable by a finite set of Horn rules, and whose implication problem is undecidable.

1 Introduction

Independence and conditional independence are fundamental concepts in areas as diverse as artificial intelligence, probability theory, social choice theory, and statistics [2,9,17]. Recently, independence logic has been introduced as an extension of classical first-order logic by independence atoms [8]. In databases, conditional independence is better known as the class of embedded multivalued data dependencies. Their associated implication problem is known to be not axiomatizable by a finite set of Horn rules, and undecidable [11,16]. Multivalued data dependencies [3] form an efficient subclass of embedded multivalued data dependencies whose implication problem has been axiomatized by a finite set of Horn rules [1] and can be decided in almost linear time [5]. They form the basis for Fagin's fourth normal form proposal to avoid data redundancy in database relations and guarantee the absence of processing difficulties [3].

In this paper we investigate an efficient subclass of embedded multivalued data dependencies which we call—in accordance with [8]— independence atoms. Intuitively, a relation r satisfies the independence atom $X \perp Y$ between two disjoint sets X and Y of attributes, if for all tuples $t_1, t_2 \in r$ there is some tuple $t \in r$ which matches the values of t_1 on all attributes in X and matches the values of t_2 on all attributes in Y. In other words, in relations that satisfy $X \perp Y$, the occurrence of X-values is independent of the occurrence of Y-values.

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 179–193, 2013.

Example 1. Consider a simple database schema that stores information about the enrolment of students into a fixed course. In fact, the schema records for each enrolled student, the year in which they completed a prerequisite course. More formally, we have the schema ENROL={*S(tudent)*, *P(rerequisite)*, *Y(ear)*}. Intuitively, every student must have completed every prerequisite in some year. For this reason, for any value in the *Student* column and every value in the *Prerequisite* column there is some year for when this student has completed that prerequisite. That is, the values in the *Student* column are independent of the values in the *Prerequisite* column. A snapshot relation r over ENROL may be:

Student	*Prerequisite*	*Year*
Turing	Math201	1932
Gödel	Math201	1925
Turing	Phys220	1932
Gödel	Phys220	1925

illustrating the independence of *Student* from *Prerequisite*.

Primarily, we propose the use of independence atoms to restrict the set of possible relations to those considered semantically meaningful for the given application domain. In this sense, updates would only be allowed if they result in a relation that satisfies all the independence atoms declared on the schema. For efficient updates it is therefore important to eliminate redundant independence atoms from those that need to be validated whenever updates occur. Naturally, this leads us to the implication problem: given a set $\Sigma \cup \{\varphi\}$ of independence atoms, does every relation that satisfies all elements in Σ also satisfy φ? If that is true, then φ is redundant since validating that the updated relation satisfies all elements in Σ guarantees that it also satisfies φ. If it is false, then it must also be validated that φ is satisfied after the update. In our example, $S \perp P$ implies $P \perp S$. Hence, we do not need to validate $P \perp S$ explicitly, since it is already validated implicitly by validating $S \perp P$ explicitly. Efficient solutions to the implication problem for classes of embedded data dependencies are particularly important for efficient updates. Indeed, validating independence atoms is rather costly due to the high amount of redundancy they cause relations to exhibit. However, in contrast to full dependencies, such as multivalued dependencies, it is yet unknown how to effectively avoid data redundancy caused by embedded dependencies. For multivalued dependencies, larger database schemata can be decomposed into smaller schemata that satisfy Fagin's Fourth Normal Form condition, which characterizes the absence of data redundancy under such dependencies [3]. For now, therefore, the primary practical impact of solving the implication problem efficiently is an effective way of avoiding redundant independence atoms.

Besides increased consistency and integrity, independence atoms can be exploited for other important data processing tasks. In query optimization, for example, they can be used to avoid expensive database operations to return query answers more efficiently. For instance, knowing that the independence atom $S \perp P$ is implied by the atoms that relations are validated against, the query on the right can be used instead of the query on the left - both returning

all combinations of students and prerequisites that occur in the relation. The right query does not use an expensive join unlike the left query.

```
SELECT E.Student, E'.Prerequisite
FROM Enrol AS E, Enrol AS E'
```

```
SELECT E.Student, E.Prerequiste
FROM Enrol AS E
```

Contributions. Motivated by these benefits we investigate the implication problem of independence atoms in database relations. Our work is inspired by and based on [7] where, however, the basic setup is one of random variables. In particular, we show that the results of [7] can be transferred from the context of random variables to the context of databases. Our first contribution is an axiomatization of the implication problem by a finite set of Horn rules. In particular, for each atom φ which cannot be inferred by our inference rules from the atoms in Σ, we construct a finite relation r_φ that satisfies Σ and violates φ. This also shows that finite and unrestricted implication problem coincide for the class of independence atoms. Exploiting our axiomatization we establish an algorithm that decides the implication problem in $\mathcal{O}(|\Sigma| \cdot ||\varphi||^2 + |\Sigma| \cdot ||\Sigma \cup \{\varphi\}||)$ time, where $|\Sigma|$ denotes the number of atoms in Σ, and $||\Sigma||$ denotes the number of attributes in Σ. Finally, we show that the implication problem of independence atoms can be reduced to the model checking problem on a single relation. For that purpose, we show how to construct for an arbitrarily given set Σ of independence atoms a relation that satisfies all the elements of Σ and violates every independence atom not implied by Σ. In the literature such relations are known as Armstrong relations [4]. Hence, checking whether φ is implied by Σ amounts to checking whether an Armstrong relation for Σ satisfies φ. Inspecting Armstrong relations is likely to increase the number of data dependencies that business analysts discover to be meaningful for a given application domain [12].

Organization. We summarize related work in Section 2, providing further motivation for the study of independence atoms and relating them to existing work in probability theory and artificial intelligence. We introduce independence atoms and their associated implication problem in Section 3. Axiomatic and algorithmic characterizations of the implication problem are established in Sections 4 and 5, respectively. The construction of Armstrong relations is shown in Section 6. We conclude in Section 7 where we also comment on future work.

2 Related Work

Approximately 100 different classes of relational data dependencies have been studied in the research literature [22]. The expressivity of embedded multivalued dependencies results in the non-axiomatizability of its implication problem by a finite set of Horn rules [16] and its undecidability [11]. Multivalued dependencies [3] form an efficient sub-class of embedded multivalued dependencies, whose implication problem has been characterized by a finite axiomatization of Horn rules [1], by an almost linear time algorithm [5], and by a fragment of Boolean propositional logic [19]. These results have recently been generalized to multivalued

dependencies over SQL databases [10]. Multivalued dependencies are very special embedded dependencies, called full dependencies, as their attributes cover the full set of attributes of the underlying relation schema. Our results show that independence atoms form another efficient sub-class of embedded multivalued dependencies. In contrast to multivalued dependencies, independence atoms are not full dependencies. Given the vast amount of literature on data dependencies, given that independence is a natural concept in many areas, and given the outlined benefits of independence atoms, it is surprising that their investigation in the database literature is rather limited [15,20][1].

In [20] Sagiv and Walecka introduce the class of *subset dependencies* which are generalizations of embedded multivalued dependencies. A subset dependency $Z(X) \subseteq Z(Y)$ for attribute sets X, Y, Z of R, where both X and Y are disjoint from Z, is satisfied by some relation r over R, if for all tuples $t_1, t_2 \in r$ that agree on all attributes in X there is some tuple $t_3 \in r$ that agrees with t_1 on all attributes in Y and that agrees with t_2 on all attributes in Z. In particular, the independence atom $Y \perp Z$ is satisfied by r if and only if the subset dependency $Z(\emptyset) \subseteq Z(Y)$ is satisfied by r. The authors establish a finite axiomatization for the class of Z-subset dependencies, which, for all relation schemata R and some fixed set $Z \subseteq R$, consists of the subset dependencies $Z(X) \subseteq Z(Y)$. It follows from the definitions that Z-subset dependencies and independence atoms are different classes of embedded multivalued dependencies.

In [15], Paredaens investigates, among three other classes of data dependencies, so called *crosses* $X \times Y$ which are equivalent to independence atoms $X \perp Y$ where both X and Y are non-empty. Paredaens establishes a finite axiomatization for crosses, including the symmetry and decomposition rules, and a somewhat convoluted version of the exchange rule. If empty attribute sets are excluded, then our axiomatization is equivalent to that for crosses, as one would expect. The motivation for our axiomatization comes from its strong analogy to the Geiger-Paz-Pearl axioms for independence atoms over probability distributions [7], in particular the simplicity of the exchange rule. Indeed, Paradaens' version of the exchange rule can be derived from the symmetry, decomposition and our exchange rule. Based on our motivation from the introduction we were also interested in algorithmic solutions to the implication problem, and Armstrong relations, which Paradaens did not aim to address.

As indicated, our study is further motivated by the existing studies of conditional independence in artificial intelligence and statistics. Here, the implication problem of conditional independence atoms is known to be not axiomatizable by a finite set of Horn rules, and to be different from that of embedded multivalued dependencies [21]. The implication problem of saturated conditional independence atoms is axiomatizable by a finite set of Horn rules, equivalent to the implication problem of multivalued dependencies, and thus equivalent to that of a fragment in Boolean propositional logic and decidable in almost linear time [2,6]. In contrast to databases, independence atoms have been investigated in

[1] We would like to thank the anonymous reviewers who pointed us to the paper by Sagiv and Walecka [20], which pointed us to the paper by Paredaens [15].

probability theory. Indeed, their implication problem over discrete probability measures has been axiomatized by a finite set of Horn rules, and can be decided in low-degree polynomial time [7]. Furthermore, probability distributions can be constructed that satisfy a given set of probabilistic independence atoms and violate all those probabilistic independence atoms not implied by the given set [7]. Therefore, our paper establishes results for independence atoms over database relations that correspond to those known for independence atoms over probability distributions. They further show that reasoning about probabilistic independence atoms does not require probabilities at all.

3 Independence Atoms

In this section we first summarize basic concepts from the relational model of data, and then introduce the syntax and semantics of independence atoms, as well as their associated implication problem.

Let $\mathfrak{A} = \{A_1, A_2, \ldots\}$ be a (countably) infinite set of symbols, called *attributes*. A *relation schema* is a finite set $R = \{A_1, \ldots, A_n\}$ of attributes from $\mathfrak{A}$. Each attribute A of a relation schema is associated with a domain $dom(A)$ which represents the set of possible values that can occur in the column named A. A *tuple* over R is a function $t : R \to \bigcup_{A \in R} dom(A)$ with $t(A) \in dom(A)$ for all $A \in R$. For $X \subseteq R$ let $t(X)$ denote the restriction of the tuple t over R on X, and $dom(X) = \prod_{A \in X} dom(A)$ the Cartesian product of the domains of attributes in X. A *relation* r over R is a finite set of tuples over R. Let $r(X) = \{t(X) \mid t \in r\}$ denote the *projection* of the relation r over R on $X \subseteq R$. For attribute sets X and Y we often write XY for their set union $X \cup Y$. For disjoint subsets $X, Y \subseteq R$, $r_1 \subseteq dom(X)$ and $r_2 \subseteq dom(Y)$ let $r_1 \times r_2 = \{t \in dom(XY) \mid \exists t_1 \in r_1, t_2 \in r_2(t(X) = t_1(X) \wedge t(Y) = t_2(Y))\}$ denote the *Cartesian product* of r_1 and r_2.

3.1 Syntax and Semantics

Intuitively, an attribute set X is independent of a disjoint attribute set Y, if X-values occur independently of Y-values. That is, the independence holds in a relation, if every X-value that occurs in the relation occurs together with every Y-value that occurs in the relation. Therefore, we arrive at the following concept, in analogy with the similar concept in so-called team semantics [8]:

Definition 1. *An* independence atom *over relation schema R is an expression $X \perp Y$ where X and Y are two disjoint subsets of R. A relation r over R is said to* satisfy *the independence atom $X \perp Y$ over R if and only if for all $t_1, t_2 \in r$ there is some $t \in r$ such that $t(X) = t_1(X)$ and $t(Y) = t_2(Y)$. If r does not satisfy $X \perp Y$, then we also say that r* violates *$X \perp Y$.*

The semantics of independence atoms can be stated explicitly as that of an embedded dependency. In the context of the attribute set XY, the concept represented by X is independent of the concept represented by Y.

Proposition 1. *Let r be a relation, and $X \perp Y$ an independence atom over relation schema R. Then r satisfies $X \perp Y$ if and only if $r(XY) = r(X) \times r(Y)$.* □

Proposition 1 captures the equivalence of independence atoms to crosses [15]. Rissanen shows in [18] that a relation r over relation schema R which satisfies a functional dependency $X \to Y$ is the lossless join of its projections $r(XY)$ and $r(X(R - XY))$. Proposition 1 shows that for a relation r which satisfies the independence atom $X \perp Y$, the projection $r(XY)$ is the lossless Cartesian product of the projections $r(X)$ and $r(Y)$.

We illustrate the semantics of independence atoms on our running example.

Example 2. The projection of relation r from Example 1 on *Student* and *Prerequisite* is the Cartesian product of the projection on *Student* and the projection on *Prerequisite*. Hence, r satisfies *Student* $\perp$ *Prerequisite*. However, the projection of r on *Student* and *Year* is not the Cartesian product of the projection on *Student* and the projection on *Year*. Thus, r violates *Student* $\perp$ *Year*.

3.2 The Implication Problem

Data dependencies are usually defined as semantic constraints that restrict the possible relations of a schema to those considered meaningful for a given application domain. Relations that satisfy all those data dependencies that express "business rules" of the domain are considered meaningful, relations that violate some business rule are considered meaningless. For efficient data processing it is therefore important to minimize the time that it takes to validate whether a relation satisfies the given set of data dependencies. Indeed, relations that satisfy a given set of data dependencies do not need to be tested whether they satisfy any data dependency that is implied by the given set. Therefore, it is essential to efficiently decide the implication problem of data dependencies. We will now define this problem for independence atoms.

For a set $\Sigma \cup \{\varphi\}$ of independence atoms we say that Σ *implies* φ, or that φ *is implied* by Σ, written $\Sigma \models \varphi$, if every relation that satisfies every element in Σ also satisfies φ. For a set Σ of independence atoms over some fixed relation schema R, we let $\Sigma^* = \{\varphi \mid \Sigma \models \varphi\}$ be the *semantic closure* of Σ, i.e., the set of all independence atoms implied by Σ. In order to determine the implied independence atoms we use a syntactic approach by applying inference rules. These inference rules have the form $\frac{\text{premise}}{\text{conclusion}}$ and inference rules without any premise are called axioms. An inference rule is called *sound*, if the independence atoms in the premise of the rule imply the independence atom in the conclusion of the rule. We let $\Sigma \vdash_{\mathfrak{R}} \varphi$ denote the *inference* of φ from Σ by the set $\mathfrak{R}$ of inference rules. That is, there is some sequence $\gamma = [\sigma_1, \ldots, \sigma_n]$ of independence atoms such that $\sigma_n = \varphi$ and every σ_i is an element of Σ or results from an application of an inference rule in $\mathfrak{R}$ to some elements in $\{\sigma_1, \ldots, \sigma_{i-1}\}$. For Σ, let $\Sigma^+_{\mathfrak{R}} = \{\varphi \mid \Sigma \vdash_{\mathfrak{R}} \varphi\}$ be its *syntactic closure* under inferences by $\mathfrak{R}$. A set $\mathfrak{R}$ of inference rules is said to be *sound* (*complete*) for the implication of independence atoms, if for every R and for every set Σ of independence atoms

over R we have $\Sigma^+_{\mathfrak{R}} \subseteq \Sigma^*$ ($\Sigma^* \subseteq \Sigma^+_{\mathfrak{R}}$). The (finite) set $\mathfrak{R}$ is said to be a (finite) *axiomatization* for the implication of independence atoms if $\mathfrak{R}$ is both sound and complete. The implication problem of independence atoms is defined as follows.

PROBLEM:	Implication problem for independence atoms
INPUT:	Relation schema R,
	Set $\Sigma \cup \{\varphi\}$ of independence atoms over R
OUTPUT:	Yes, if $\Sigma \models \varphi$; No, otherwise

We illustrate the implication problem on our running example.

Example 3. Continuing Example 1, the set Σ of independence atoms consisted of *Student* $\perp$ *Prerequisite*. If φ denotes the independence atom *Student* $\perp$ *Year*, then the relation r from Example 1 shows that $\Sigma \not\models \varphi$. An example of an independence atom that is implied by Σ is *Prerequisite* $\perp$ *Student*.

4 Axiomatic Characterization

In this section we establish an axiomatic characterization of the implication problem for independence atoms over relations. In fact, we show that the set $\mathfrak{I}$ of inference rules from Table 1 is sound and complete. The set $\mathfrak{I}$ of inference rules is the same set used in [7] to axiomatize implication among independence atoms in the framework of random variables. It is remarkable that the same axioms have found their way also to the study of concurrency [13] and secrecy [14]. If the attribute sets X, Y and Z are interpreted as sets of vectors in a vector space, the rules $\mathfrak{I}$ would govern the concept of linear (as well as algebraic) independence as was noted already in [23,24].

Table 1. Axiomatization $\mathfrak{I}$ of Independence in Database Relations

$\dfrac{}{X \perp \emptyset}$ (trivial independence, $\mathcal{T}$)	$\dfrac{X \perp Y}{Y \perp X}$ (symmetry, $\mathcal{S}$)
$\dfrac{X \perp YZ}{X \perp Y}$ (decomposition, $\mathcal{D}$)	$\dfrac{X \perp Y \quad XY \perp Z}{X \perp YZ}$ (exchange, $\mathcal{E}$)

Using Definition 1 it is not difficult to show the soundness of the inference rules in $\mathfrak{I}$ for the implication of independence atoms. As an example, we prove the soundness of the exchange rule $\mathcal{E}$. Let r be a relation that satisfies the independence atoms $X \perp Y$ and $XY \perp Z$. Let $t_1, t_2 \in r$. Then there is some tuple $\bar{t} \in r$ such that $\bar{t}(X) = t_1(X)$ and $\bar{t}(Y) = t_2(Y)$, since r satisfies $X \perp Y$. Since r satisfies $XY \perp Z$, for $\bar{t}, t_2 \in r$ there must be some $t \in r$ such that $t(XY) = \bar{t}(XY)$ and $t(Z) = t_2(Z)$. In particular, $t(X) = \bar{t}(X) = t_1(X)$, $t(Y) = \bar{t}(Y) = t_2(Y)$,

and $t(Z) = t_2(Z)$. Hence, there is some $t \in r$ such that $t(X) = t_1(X)$ and $t(YZ) = t_2(YZ)$. That is, r also satisfies the independence atom $X \perp YZ$.

The soundness of the rules in $\mathfrak{I}$ allows us to mechanically infer several implied independence atoms.

Example 4. Recall that $\Sigma = \{Student \perp Prerequisite\}$ in our running example. A single application of the symmetry rule $\mathcal{S}$ to $Student \perp Prerequisite$ gives us the independence atom $Prerequisite \perp Student \in \Sigma^+_{\mathfrak{I}}$. Consequently, $Prerequisite \perp Student \in \Sigma^*$ due to the soundness of the symmetry rule.

The inference rules in $\mathfrak{I}$ are also complete. That is, every implied independence atom can be inferred by applications of the inference rules in $\mathfrak{I}$. The following theorem is like Theorem 3 of [7]:

Theorem 1. *The set $\mathfrak{I}$ of Horn rules forms a finite axiomatization for the class of independence atoms.*

Proof. We proceed as in [7, Theorem 3], but working with relations instead of random variables. Let R be some relation schema and Σ a set of independence atoms over R. Let $\varphi = X \perp Y \notin \Sigma^+_{\mathfrak{I}}$. Without loss of generality we assume that for all non-empty sets $X' \subseteq X$ and $Y' \subseteq Y$ with $X'Y' \neq XY$, $X' \perp Y' \in \Sigma^+_{\mathfrak{I}}$ holds. An independence atom φ with these properties is called *minimal*. Indeed, if $\varphi = X \perp Y$ is not minimal, then we can remove attributes from X or from Y to obtain a minimal atom $\varphi' = X' \perp Y' \notin \Sigma^+_{\mathfrak{I}}$. Note that, if X' and Y' are both singletons, then $X' \perp Y'$ is a minimal atom due to the trivial independence axiom $\mathcal{T}$. For each minimal atom φ' we construct a relation $r_{\varphi'}$ that satisfies Σ and violates φ'. Due to the decomposition rule $\mathcal{D}$, $r_{\varphi'}$ also violates φ and, hence, φ is not implied by Σ.

Let $\varphi = X \perp Y \notin \Sigma^+_{\mathfrak{I}}$ be a minimal atom. For all $A \in R$ assume that $dom(A) = \{0, 1\}$, and let $Z = R - XY$. Let $A_0 \in X$. Define $r_\varphi \subseteq dom(R)$ as follows: for all $t \in dom(R)$ we have,

$$t \in r_\varphi \text{ if and only if } t(A_0) = \textstyle\sum_{A \in (X - A_0)Y} t(A) \mod 2 .$$

Clearly, $r = r(XY) \times \prod_{A \in Z} dom(A)$.

We show first that r_φ violates the independence atom $X \perp Y$. Let t be a tuple where $t(A_0) = 1$ and $t(A) = 0$ for all $A \in XY - A_0$. Then $t \in r(X) \times r(Y)$, but $t \notin r(XY)$.

It remains to show that r_φ satisfies every independence atom $V \perp W \in \Sigma$.

Case 1. Assume that $V \subseteq Z$ or $W \subseteq Z$. Say, for example, that $V \subseteq Z$. By construction, for every tuple $t_1 \in r(Z)$ and every tuple in $t_2 \in r(W)$ there is some tuple $t \in r(VW)$ such that $t(V) = t_1(V)$ and $t(W) = t_2(W)$. The case where $W \subseteq Z$ holds is similar. Hence, $r(VW) = r(V) \times r(W)$.

Case 2. Assume that $V \cap XY \neq \emptyset$ and $W \cap XY \neq \emptyset$.

Case 2.1. Suppose $XY \not\subseteq VW$. For $U \subseteq XY$ with $U \neq XY$ we have $r(U) = \prod_{A \in U} r(A)$. Hence, $r(VW) = \prod_{A \in VW} r(A)$. In particular, $r(VW) = r(V) \times r(W)$.

Case 2.2. Suppose $XY \subseteq VW$. Then let $V = X'Y'Z'$, $W = X''Y''Z''$ where $X = X'X''$, $Y = Y'Y''$, and $Z'Z'' \subseteq Z$ holds. Assume that $V \perp W \in \Sigma_{\mathfrak{J}}^+$. We show, under this assumption, the contradiction that $X \perp Y \in \Sigma_{\mathfrak{J}}^+$ holds. Consequently, $V \perp W \notin \Sigma_{\mathfrak{J}}^+$ and this case cannot occur.

Since $X \perp Y$ is a minimal independence atom, $X' \perp Y', X'' \perp Y \in \Sigma_{\mathfrak{J}}^+$. The inference

$$\dfrac{\dfrac{\dfrac{X'' \perp Y}{\mathcal{D}:\ Y \perp X''} \quad \dfrac{X' \perp Y' \quad \dfrac{X'Y'Z' \perp X''Y''Z''}{\mathcal{D}:\ X'Y' \perp X''Y''}}{\mathcal{E}:\ X' \perp X''Y}}{\mathcal{S}:\ X''Y \perp X'}}{\mathcal{E}:\ Y \perp X}}{\mathcal{S}:\ X \perp Y}$$

gives the anticipated contradiction that $X \perp Y \in \Sigma_{\mathfrak{J}}^+$ under the assumption that $V \perp W \in \Sigma_{\mathfrak{J}}^+$ when $XY \subseteq VW$. Note that the inference of $X \perp Y \in \Sigma_{\mathfrak{J}}^+$ remains valid even if some of the X', X'', Y', Y'' are empty, as long as $X = X'X''$ and $Y = Y'Y''$ hold. □

We illustrate the completeness argument on our running example.

Example 5. Let $\Sigma = \{Student \perp Prerequisite\}$ be a set of independence atoms and $\varphi = Prerequiste \perp Year$ be an independence atom over ENROL. The construction from the completeness proof of Theorem 1 may result in the relation on the left, which may result in the relation on the right by suitable substitutions.

Student	*Prerequiste*	*Year*
S1	P1	Y1
S1	P2	Y2
S2	P1	Y1
S2	P2	Y2

Student	*Prerequiste*	*Year*
Hilbert	Phil101	1900
Hilbert	Phys110	1905
Ackermann	Phil101	1900
Ackermann	Phys110	1905

Both relations satisfy Σ and violate φ.

Instead of the exchange rule $\mathcal{E}$, Paredaens used the following rule on the left

$$\dfrac{X' \perp Z \quad X \perp Y}{X \cap X' \perp Y \cup (X \cap Z)}\ X \cap X' \neq \emptyset \qquad \dfrac{UV \perp WW' \quad UWW'' \perp Y}{U \perp YW}$$

for crosses, defined for non-empty attribute sets. This rule produces an independence atom that cannot already be inferred by the decomposition and symmetry rules alone, if $X \cap X' \neq \emptyset$ and $X \cap Z \neq \emptyset$ hold. In that case, the rule can be rewritten as the rule on the right above. This rule can be inferred as follows

$$\dfrac{\dfrac{\dfrac{\dfrac{UV \perp WW'}{\mathcal{D}:\ UV \perp W}}{\mathcal{S}:\ W \perp UV}}{\mathcal{D}:\ W \perp U} \quad \dfrac{UWW'' \perp Y}{\mathcal{D}:\ UW \perp Y}}{\mathcal{E}:\ U \perp YW}$$

from the decomposition, symmetry, and exchange rule.

5 Algorithmic Characterization

We establish an algorithmic characterization of the implication problem. In practice, one may simply want to check if a single independence atom φ is implied by a given set Σ of independence atoms. One could compute $\Sigma^* = \Sigma_{\mathfrak{I}}^+$ and check if $\varphi \in \Sigma^*$. However, this algorithm is hardly efficient. Instead, we exploit the extra knowledge about φ to decide more efficiently if φ is implied by Σ.

The divide-and-conquer algorithm is presented as Algorithm 1. On input $(\Sigma, X \perp Y)$ it reduces Σ to $\Sigma' = \Sigma[XY] = \{(V \cap XY) \perp (W \cap XY) \mid V \perp W \in \Sigma\}$ (line 3). If $X \perp Y$ is a trivial independence atom, i.e., if one of its sets is empty, or if the atom or its symmetric atom is included in Σ', then the algorithm returns `true` (line 4-5). If there is no non-trivial atom $U \perp V \in \Sigma'$ where $UV = XY$, then the algorithm returns `false` (line 7-8). Otherwise, there is some non-trivial atom $U \perp V \in \Sigma'$ where $U = QR$, $V = ST$, and $X = QS$, $Y = RT$. In this case, the Algorithm returns `true` if and only if it returns `true` on both inputs $(\Sigma', Q \perp R)$ and $(\Sigma', S \perp T)$ (line 12).

Algorithm 1. Implication

```
 1: procedure IMPLIED(Σ, φ)
 2:     φ ← X ⊥ Y;
 3:     Σ' ← Σ[XY];
 4:     if X = ∅ or Y = ∅ or X ⊥ Y ∈ Σ' or Y ⊥ X ∈ Σ' then
 5:         IMPLIED(Σ, φ) ← true;
 6:     end if;
 7:     if for all U ⊥ V ∈ Σ' with U ≠ ∅ and V ≠ ∅, UV ≠ XY then
 8:         IMPLIED(Σ, φ) ← false;
 9:     else          ▷ ∃U ⊥ V ∈ Σ' with ∅ ≠ U = QR, ∅ ≠ V = ST, X = QS, Y = RT
10:         φ1 ← Q ⊥ R;
11:         φ2 ← S ⊥ T;
12:         IMPLIED(Σ, φ) ← IMPLIED(Σ', φ1) ∧ IMPLIED(Σ', φ2);
13:     end if;
14: end procedure
```

Algorithm 1 works correctly in low-degree polynomial time. This yields the following result, reminiscent of Theorems 8 and 9 of [7]:

Theorem 2. *Algorithm 1 terminates, and* IMPLIED(Σ, φ) = `true` *if and only if* $\Sigma \models \varphi$. *The time-complexity of Algorithm 1, on input* (Σ, φ), *is in* $\mathcal{O}(|\Sigma| \cdot ||\varphi||^2 + |\Sigma| \cdot ||\Sigma \cup \{\varphi\}||)$.

Proof (Sketch). Let $\varphi = X \perp Y$. Firstly, it follows from an inspection of the inference rules in $\mathfrak{I}$ that $\Sigma \vdash_{\mathfrak{I}} \varphi$ holds if and only if $\Sigma' \vdash_{\mathfrak{I}} \varphi$ holds.

Secondly, for any non-trivial φ, $\Sigma' \vdash_{\mathfrak{I}} \varphi$ holds only if there is some atom $U \perp V \in \Sigma'$ such that $UV = XY$. This follows from the observation that no inference rule in $\mathfrak{I}$ introduces an attribute to its conclusion that does not already occur in one of its premises.

These observations justify lines 2-7 of Algorithm 1. We will now justify lines 9-12. Let $X = QS$, $Y = RT$, $U = QR$, and $V = ST$. Then we show that, if $U \perp V \in \Sigma_{\mathfrak{J}}^+$, then $X \perp Y \in \Sigma_{\mathfrak{J}}^+$ if and only if $Q \perp R \in \Sigma[QR]_{\mathfrak{J}}^+$ and $S \perp T \in \Sigma[ST]_{\mathfrak{J}}^+$.

Assume first that $QR \perp ST, Q \perp R, S \perp T \in \Sigma_{\mathfrak{J}}^+$. Then the following inference shows that $QS \perp RT \in \Sigma_{\mathfrak{J}}^+$, too.

$$\dfrac{\dfrac{\dfrac{S \perp T \qquad \dfrac{QR \perp ST}{\mathcal{S}:\ ST \perp QR}}{\mathcal{E}:\ S \perp QRT}}{\mathcal{D}:\ S \perp RT}}{\mathcal{S}:\ RT \perp S} \qquad \dfrac{\dfrac{Q \perp R \qquad QR \perp ST}{\mathcal{E}:\ Q \perp RST}}{\mathcal{S}:\ SRT \perp Q}$$

$$\dfrac{\dfrac{\mathcal{E}:\ RT \perp QS}{\mathcal{S}:\ QS \perp RT}}{}$$

If $QS \perp RT \in \Sigma_{\mathfrak{J}}^+$, then there is an inference $U_1 \perp V_1, \ldots, U_k \perp V_k = QS \perp RT$ from Σ. Consequently, $(U_1 \cap QR) \perp (V_1 \cap QR), \ldots, (U_k \cap QR) \perp (V_k \cap QR) = Q \perp R$ is an inference of $Q \perp R$ from $\Sigma[QR]$. Similarly, an inference of $S \perp T$ from $\Sigma[ST]$ can be constructed from an inference of $QS \perp RT$ from Σ.

Note that this shows, in particular, that a selection of $U \perp V$ in line 9 can be made arbitrarily since any selection provides a necessary and sufficient means to check whether $X \perp Y \in \Sigma_{\mathfrak{J}}^+$.

Algorithm 1 terminates since the size of the independence atoms strictly decreases in line 12. If the algorithm did not terminate before, it will terminate when the number of attributes in the two atoms have reached 2 (line 4 or line 7). The first statement of Theorem 2 follows from a simple induction on the number of attributes in φ.

We will now analyze the time complexity of Algorithm 1. The complexity is measured in terms of two types of basic operations: the comparison of two independence atoms and the projection of independence atoms. Both operations are bounded by the number $||\Sigma \cup \{\varphi\}||$ of distinct attributes in $\Sigma \cup \{\varphi\}$. Let $c(\varphi)$ denote the number of basic operations required to solve a problem for an independence atom φ, and assume for now that the distinct attributes in Σ are those in φ. By line 12, $c(\varphi)$ must satisfy the equation $c(\varphi) \leq c(\varphi_1) + c(\varphi_2) + |\Sigma|$, where $|\Sigma|$ denotes the number of atoms in Σ, and where $||\varphi|| = ||\varphi_1|| + ||\varphi_2||$. The solution to this equation is $\mathcal{O}(|\Sigma| \cdot ||\varphi||)$ measured in basic operations. Adding the cost of projecting Σ to the attributes in φ is in $\mathcal{O}(|\Sigma| \cdot ||\Sigma \cup \{\varphi\}||)$. □

Example 6. Let $\Sigma = \{Student \perp Prerequisite\}$ be a set of independence atoms and $\varphi = Prerequiste \perp Year$ be an independence atom over relation schema Enrol. On input (Σ, φ), Algorithm 1 computes $\Sigma' = \emptyset$ in Step 3, and returns Implied(Σ, φ) = `false` in Step 8, since the condition in Step 7 is trivially satisfied. Hence, by Theorem 2, φ is not implied by Σ.

6 Armstrong Relations

We show that independence atoms enjoy Armstrong relations. That is, for every relation schema R and every set Σ of independence atoms, there is a relation over R that satisfies Σ and violates every independence atom not implied by Σ. The property of enjoying Armstrong relations has been characterized by Fagin in a very general framework [4]. We will exploit this characterization to show how to construct Armstrong relations for independence atoms.

Theorem 3. *[4] Let $\mathcal{S}$ denote a set of sentences. The following properties of $\mathcal{S}$ are equivalent:*

1. Existence of a faithful operator. *There exists an operator $\otimes$ that maps non-empty families of models into models, such that if σ is a sentence in $\mathcal{S}$ and $\langle P_i : i \in I\rangle$ is a non-empty family of models, then σ holds for $\otimes\langle P_i : i \in I\rangle$ if and only if σ holds for each P_i.*
2. Existence of Armstrong models. *Whenever Σ is a consistent subset of $\mathcal{S}$ and Σ^* is the set of sentences in $\mathcal{S}$ that are logical consequences of Σ, then there exists a model (an "Armstrong" model) that obeys Σ^* and no other sentence in $\mathcal{S}$.*
3. Splitting of disjunctions. *Whenever Σ is a subset of $\mathcal{S}$ and $\{\sigma_i : i \in I\}$ is a non-empty subset of $\mathcal{S}$, then $\Sigma \models \bigvee\{\sigma_i : i \in I\}$ if and only if there exists some $i \in I$ such that $\Sigma \models \sigma_i$.* □

Indeed, there is a faithful operator for independence atoms. While Fagin's theorem holds for any cardinality of I [4], we use it only for finite non-empty I. An analog of the below theorem was proved in [7, Theorem 11] for distributions.

Theorem 4. *Let $\{r_i : i = 1, \ldots, n\}$ be a finite set of relations. There exists an operation $\otimes$ that maps finite sets of relations to relations such that for each independence atom σ, the relation $\otimes\{r_i : i = 1, \ldots, n\}$ satisfies σ if and only if for $i = 1, \ldots, n$, r_i satisfies σ.*

Proof. We construct the operation $\otimes$ by using a binary operation $\otimes_b$ such that for every independence atom σ, the relation $r_1 \otimes_b r_2$ satisfies σ if and only if both relations r_1 and r_2 satisfy σ. The operation $\otimes$ is then defined recursively by $\otimes\{r_i : i = 1, \ldots, n\} := (\cdots((r_1 \otimes_b r_2) \otimes_b r_3) \cdots \otimes_b r_n)$. Let r_1, r_2 be relations over relation schema R. Then $r_1 \otimes_b r_2$ is defined by

$$((a_1, a_1'), \ldots, (a_n, a_n')) \in r_1 \otimes_b r_2 \text{ iff } (a_1, \ldots, a_n) \in r_1 \text{ and } (a_1', \ldots, a_n') \in r_2.$$

We now show that for an independence atom $X \perp Y$ over R we have, $r_1 \otimes_b r_2$ satisfies $X \perp Y$ if and only if r_1 satisfies $X \perp Y$ and r_2 satisfies $X \perp Y$.

We show first that if r_1 satisfies $X \perp Y$ and r_2 satisfies $X \perp Y$, then $r_1 \otimes_b r_2$ satisfies $X \perp Y$. Let $t_1 = ((a_1, a_1'), \ldots, (a_n, a_n')), t_2 = ((b_1, b_1'), \ldots, (b_n, b_n')) \in r_1 \otimes_b r_2$. Then, $t_1^1 = (a_1, \ldots, a_n), t_2^1 = (b_1, \ldots, b_n) \in r_1$, and $t_1^2 = (a_1', \ldots, a_n'), t_2^2 = (b_1', \ldots, b_n') \in r_2$. Since r_1 satisfies $X \perp Y$ there is some $\bar{t} = (c_1, \ldots, c_n) \in r_1$ such that $\bar{t} = t_1^1(X)$ and $\bar{t} = t_2^1(Y)$. Since r_2 satisfies $X \perp Y$ there is some

$t' = (c'_1, \ldots, c'_n) \in r_2$ such that $t'(X) = t^2_1(X)$ and $t' = t^2_2(Y)$. Let $t := ((c_1, c'_1), \ldots, (c_n, c'_n)) \in r_1 \otimes_b r_2$. It follows that $t(X) = t_1(X)$ and $t(Y) = t_2(Y)$. Hence, $r_1 \otimes_b r_2$ satisfies $X \perp Y$.

It remains to show that if $r_1 \otimes_b r_2$ satisfies $X \perp Y$, then r_1 satisfies $X \perp Y$ and r_2 satisfies $X \perp Y$. Let $t^1_1 = (a_1, \ldots, a_n), t^1_2 = (b_1, \ldots, b_n) \in r_1$ and $t^2_1 = (a'_1, \ldots, a'_n), t^2_2 = (b'_1, \ldots, b'_n) \in r_2$. Then, $t_1 = ((a_1, a'_1), \ldots, (a_n, a'_n)), t_2 = ((b_1, b'_1), \ldots, (b_n, b'_n)) \in r_1 \otimes_b r_2$. Since $r_1 \otimes_b r_2$ satisfies $X \perp Y$ there is some $t = ((c_1, c'_1), \ldots, (c_n, c'_n)) \in r_1 \otimes r_2$ where $t(X) = t_1(X)$ and $t(Y) = t_2(Y)$. Then $t^{r_1} := (c_1, \ldots, c_n)$ satisfies $t^{r_1}(X) = t^1_1(X)$ and $t^{r_1}(Y) = t^1_2(Y)$. Thus, r_1 satisfies $X \perp Y$. Similarly, $t^{r_2} := (c'_1, \ldots, c'_n)$ satisfies $t^{r_2}(X) = t^2_1(X)$ and $t^{r_2}(Y) = t^2_2(Y)$. Hence, r_2 satisfies $X \perp Y$. □

We illustrate the construction of Armstrong relations on our example.

Example 7. Let $\Sigma = \{Student \perp Prerequisite\}$ be a set of independence atoms over ENROL. From previous examples we have seen the relation r_1 on the left that satisfies Σ and $P \perp Y$, but violates $S \perp Y$, and the relation r_2 on the right that satisfies Σ and $S \perp Y$, but violates $P \perp Y$.

Student	*Prerequiste*	*Year*
S1	P1	Y1
S2	P1	Y2
S1	P2	Y1
S2	P2	Y2

Student	*Prerequiste*	*Year*
S3	P3	Y3
S3	P4	Y4
S4	P3	Y3
S4	P4	Y4

The Armstrong construction results in the relation $r_1 \otimes_b r_2$, defined by $((a_1, a'_1), \ldots, (a_n, a'_n)) \in r_1 \otimes_b r_2$ iff $(a_1, \ldots, a_n) \in r_1$ and $(a'_1, \ldots, a'_n) \in r_2$, on the left, and suitable substitutions yield the relation on the right.

Student	*Prerequiste*	*Year*
(S1,S3)	(P1,P3)	(Y1,Y3)
(S1,S3)	(P1,P4)	(Y1,Y4)
(S1,S4)	(P1,P3)	(Y1,Y3)
(S1,S4)	(P1,P4)	(Y1,Y4)
(S2,S3)	(P1,P3)	(Y2,Y3)
(S2,S3)	(P1,P4)	(Y2,Y4)
(S2,S4)	(P1,P3)	(Y2,Y3)
(S2,S4)	(P1,P4)	(Y2,Y4)
(S1,S3)	(P2,P3)	(Y1,Y3)
(S1,S3)	(P2,P4)	(Y1,Y4)
(S1,S4)	(P2,P3)	(Y1,Y3)
(S1,S4)	(P2,P4)	(Y1,Y4)
(S2,S3)	(P2,P3)	(Y2,Y3)
(S2,S3)	(P2,P4)	(Y2,Y4)
(S2,S4)	(P2,P3)	(Y2,Y3)
(S2,S4)	(P2,P4)	(Y2,Y4)

Student	*Prerequiste*	*Year*
Sheldon	Ethi101	2010
Sheldon	Logi120	2011
Leonard	Ethi101	2010
Leonard	Logi120	2011
Howard	Ethi101	2012
Howard	Logi120	2013
Raj	Ethi101	2012
Raj	Logi120	2013
Sheldon	Chem110	2010
Sheldon	Biol105	2011
Leonard	Chem110	2010
Leonard	Biol105	2011
Howard	Chem110	2012
Howard	Biol105	2013
Raj	Chem110	2012
Raj	Biol105	2013

Indeed the latter two relations are Armstrong relations for Σ. That is, they satisfy Σ and violate $S \perp Y$ and $P \perp Y$, and thereby also $S \perp PY$, $SY \perp P$, $SP \perp Y$, $S \perp YP$, and their symmetric independence atoms.

It can now be shown how an arbitrary set of independence atoms can be visualized as a single Armstrong relation. In practice, Armstrong relations can be used by database designers and business analysts as a communication tool to acquire and discuss the meaningfulness of business rules with domain experts [12]. Just as [7, Theorem 11] obtains for distributions, we obtain:

Theorem 5. *The class of independence atoms enjoys Armstrong relations.*

Proof. Let R be an arbitrary relation schema and Σ a set of independence atoms over R. By Theorem 1, for each $\varphi \notin \Sigma^+_{\mathfrak{I}}$ there is some relation r_φ that satisfies Σ and violates φ. Let $r := \otimes\{r_\varphi \mid \varphi \notin \Sigma^+_{\mathfrak{I}}\}$. The relation is well-defined since the set of all independence atoms over a relation schema is finite. According to Theorem 4, r satisfies all independence atoms in Σ and violates every independence atom not implied by Σ. □

It also follows from our results and Theorem 3 that the set $\mathfrak{I}$ of inference rules is powerful enough to infer all disjunctions of independence atoms that are logically implied by a set of independence atoms, and not merely single independence atoms.

7 Conclusion and Future Work

We investigated independence atoms, introduced in [8], as a new class of relational data dependencies. Our results show that independence atoms form an efficient sub-class of embedded multivalued dependencies whose implication problem is not finitely axiomatizable and undecidable. Our efficient solutions to the implication problem can result in enormous cost savings in data processing, for example when validating the consistency of update operations on relations, or when querying relations. Independence atoms form the database counterpart of probabilistic independence atoms known from probability theory.

In future work we plan to implement our algorithms as a tool, and analyze how the inspection of Armstrong relations can help database designers or business analysts with the task of identifying independence atoms that are semantically meaningful for a given application domain. It is interesting to investigate the minimum number of tuples required in Armstrong relations. It is also a challenging problem to identify means to reduce data redundancy caused by embedded dependencies. For the field of (in)dependence logic, it would be interesting to axiomatize the combined class of independence and dependence atoms.

Acknowledgement. This research is supported by the Marsden Fund Council from Government funding, administered by the Royal Society of New Zealand, and grants 264917 and 251557 of the Academy of Finland.

References

1. Beeri, C., Fagin, R., Howard, J.H.: A complete axiomatization for functional and multivalued dependencies in database relations. In: SIGMOD Conference, pp. 47–61. ACM (1977)

2. Dawid, A.P.: Conditional independence in statistical theory. Journal of the Royal Statistical Society. Series B (Methodological) 41(1), 1–31 (1979)
3. Fagin, R.: Multivalued dependencies and a new normal form for relational databases. ACM Trans. Database Syst. 2(3), 262–278 (1977)
4. Fagin, R.: Horn clauses and database dependencies. J. ACM 29(4), 952–985 (1982)
5. Galil, Z.: An almost linear-time algorithm for computing a dependency basis in a relational database. J. ACM 29(1), 96–102 (1982)
6. Geiger, D., Pearl, J.: Logical and algorithmic properties of conditional independence and graphical models. The Annals of Statistics 21(4), 2001–2021 (1993)
7. Geiger, D., Paz, A., Pearl, J.: Axioms and algorithms for inferences involving probabilistic independence. Inf. Comput. 91(1), 128–141 (1991)
8. Grädel, E., Väänänen, J.A.: Dependence and independence. Studia Logica 101(2), 399–410 (2013)
9. Halpern, J.: Reasoning about uncertainty. MIT Press (2005)
10. Hartmann, S., Link, S.: The implication problem of data dependencies over SQL table definitions. ACM Trans. Datab. Syst. 37(2), 13.1–13.52 (2012)
11. Herrmann, C.: On the undecidability of implications between embedded multivalued database dependencies. Inf. Comput. 204(12), 1847–1851 (2006)
12. Langeveldt, W., Link, S.: Empirical evidence for the usefulness of Armstrong relations on the acquisition of meaningful FDs. Inf. Syst. 35(3), 352–374 (2010)
13. More, S.M., Naumov, P., Sapp, B.: Concurrency Semantics for the Geiger-Paz-Pearl Axioms of Independence. In: CSL, vol. 12, pp. 443–457 (2011)
14. Naumov, P.: Independence in information spaces. Studia Logica 100(5), 953–973 (2012)
15. Paredaens, J.: The interaction of integrity constraints in an information system. J. Comput. Syst. Sci. 20(3), 310–329 (1980)
16. Parker Jr., D., Parsaye-Ghomi, K.: Inferences involving embedded multivalued dependencies and transitive dependencies. In: SIGMOD Conference, pp. 52–57 (1980)
17. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)
18. Rissanen, J.: Independent components of relations. ACM Trans. Database Syst. 2(4), 317–325 (1977)
19. Sagiv, Y., Delobel, C., Parker Jr., D., Fagin, R.: An equivalence between relational database dependencies and a fragment of propositional logic. J. ACM 28(3), 435–453 (1981)
20. 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)
21. Studený, M.: Conditional independence relations have no finite complete characterization. In: Transactions of the 11th Prague Conference on Information Theory, pp. 377–396. Kluwer (1992)
22. Thalheim, B.: Dependencies in relational databases. Teubner (1991)
23. van der Waerden, B.L.: Moderne Algebra. J. Springer, Berlin (1940)
24. Whitney, H.: On the Abstract Properties of Linear Dependence. Amer. J. Math. 57(3), 509–533 (1935)

Substructural Logic of Proofs

Hidenori Kurokawa[1] and Hirohiko Kushida[2]

[1] Kobe University
hidenori.kurokawa@gmail.com
[2] The City University of New York, Graduate Center
hkushida@gc.cuny.edu

Abstract. In this paper, we introduce substructural variants of Artemov's logic of proofs. We show a few things here. First, we introduce a bimodal logic that has both the exponential operator in linear logic and an S4 modal operator which does not bring in any structural feature. Both Girard's embedding and Gödel's modal embedding (not the double negation translation) are used to directly connect intuitionistic substructural logics and substructural logics with the involutive negation. Second, we formulate substructural logic of proofs, which is an explicit counterpart of the foregoing bimodal substructural logics, and show that the substructural logic of proofs can realize the bimodal substructural logic, following the idea of [1]. Third, adopting the idea of Yu [10], we also show that the contraction-free, multiplicative S4-fragment of the bimodal substructural logic is realizable without appealing to a so-called "self-referential constant specification."

1 Introduction

The logic of proofs (LP) has been introduced in order to analyze a combinatorial structure of "proofs" underlying S4 compliant (or other normal) modal logics. S4, into which intuitionistic logic (IL) is embeddable ([5]), is usually taken to codify the notion of "informal provability." The structure of proofs underlying the S4 modality is made explicit via proof-terms in LP in such a way that (1) LP, an explicit counterpart of S4, is given an intended arithmetic interpretation (sound and complete w.r.t. PA), whereas S4 itself is not sound under arithmetic interpretation; however, (2) the precise connection with S4 is given via its realization theorem, which roughly states that for any theorem of S4, we can assign an appropriate proof term to each $\Box$.

On the other hand, in linear logic (LL), it has been shown that intuitionistic logic can be embedded in intuitionistic linear logic (ILL) by its linear implication and the exponential operator, and it is embedded into classical linear logic via double negation translation ([4]). We have yet another tradition of modal substructural logics (introduced by Došen [3]), whose modalities have been motivated by a reason different from Girard's exponential. Those modalities are introduced to formulate the modal embeddings from constructive substructural logics into modal substructural logics with the involutive negation. We call this modality "non-structural modality."

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 194–210, 2013.

Although classical LL is usually related to intuitionistic LL only via double negation translation, it may be natural to consider a purely modal framework into which intuitionistic LL can be embedded. Such a system must be a bimodal logic, since we have both Girard's exponential and modality used for Gödel embedding. In this paper, we introduce such a bimodal logic called LLS4 (and its subsystem that has only non-structural modality called BCS4) and we prove cut-elimination for LLS4.

In addition, we introduce a two-sorted substructural logic of proofs (LLP) that is an explicit counterpart of the bimodal substructural logic. This logic has two different kinds of proof terms, one of which allows us to apply structural rules to the formulas with the proof terms (analogous to Girard's exponential) and the other does not. We call the former "structural proof terms" and the latter "non-structural proof terms." We also introduce a subsystem of the two-sorted substructural logic of proofs (MALLP) which is an explicit counterpart of the substructural logic with non-structural modality. We prove the realization theorems for the modal substructural logics via the substructural logics of proofs.

An analysis of combinatorial structure of proof terms under substructural constraints is not the whole point of this project. In LP, in order to prove its proof internalization (if $\overline{s} : \Gamma, \Delta \vdash_{LP} \varphi$, then $\overline{s} : \Gamma, \overline{y} : \Delta \vdash_{LP} t(\overline{s}, \overline{y}) : \varphi$ for some proof term t),[1] we use a machinery called "constant specification" (roughly, this is an assignment of an axiom to a proof term called "proof constant"), by which we can have a rule called an "axiom necessitation" of the form $c : A$ where A is one of the axioms of LP. It is already known ([6]) that in order to realize S4 (and its fragment that is in the range of Gödel embedding [11]), we need a so-called "self-referential constant specification," which yields an axiom necessitation of the form $c : A(c)$ in a schematic form (namely, the same proof constant as used for the whole axiom occurs in an instance of the axiom). The full significance of this self-referentiality is yet to be investigated. However, applying a recent proof-theoretic analysis of self-referentiality by Yu ([10], [12]), we show that the multiplicative fragment of the weakening-free and contraction-free subsystem of S4, i.e. BCS4,[2] is realizable without using self-referential constant specifications. The result suggests that there is an essential relationship between the self-referentiality at issue here and the structural rules, since the multiplicative fragment is the one where the effect of being contraction-free becomes particularly clear, since there is no merge of formulas in a proof tree in this fragment. (See section 6.)

2 Bimodal Substructural Logic

Let us present a Gentzen-style sequent calculus for the bimodal substructural logic LLS4. Here we follow the notation of [7]. Hence, we use $\otimes$, $\oplus$, 0, 1 for the multiplicative constants, $\wedge$, $\vee$, $\top$, $\bot$ for the additive constants, and $\rightarrow$, $\neg$ for the

[1] $\overline{s}$ stands for $s_1, \ldots, s_n$.

[2] Although the name of a logic BC stands for a subsystem of intuitionistic logic, by this nomenclature, we invariantly talk about the logic with involutive negation.

linear implication and the involutive negation, respectively. $\boxdot$ is our notation for the exponential ! in linear logic, and $\Box$ is the additional modal operator that has no structural capacity. (We assume that these modal operators are more strongly associated with other formulas than the other logical constants, so we omit parentheses for these.) The language is officially specified as follows.

$$A ::= 1|\bot|\top|0|P|A \wedge B|A \otimes B|A \vee B|A \oplus B|A \to B|\neg A|\Box A|\boxdot A.$$

We now present a Gentzen-style sequent calculus for LLS4.

Axioms: $A \Rightarrow A \qquad 0 \Rightarrow \qquad \Rightarrow 1 \qquad \Gamma \Rightarrow \top, \Delta \qquad \bot, \Gamma \Rightarrow \Delta$

Inference rules for additive connectives:

$$\frac{A, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta} \wedge : l \qquad \frac{B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta} \wedge : l$$

$$\frac{\Gamma \Rightarrow \Delta, A \quad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B} \wedge : r \qquad \frac{A, \Gamma \Rightarrow \Delta \quad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta} \vee : l$$

$$\frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, A \vee B} \vee : r \qquad \frac{\Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \vee B} \vee : r$$

Inference rules for multiplicative connectives:

$$\frac{A, B, \Gamma \Rightarrow \Delta}{A \otimes B, \Gamma \Rightarrow \Delta} \otimes : l \qquad \frac{\Gamma \Rightarrow \Delta, A \quad \Pi \Rightarrow \Theta, B}{\Gamma, \Pi \Rightarrow \Delta, \Theta, A \otimes B} \otimes : r$$

$$\frac{A, \Gamma \Rightarrow \Delta \quad B, \Pi \Rightarrow \Theta}{A \oplus B, \Gamma, \Pi \Rightarrow \Delta, \Theta} \oplus : l \qquad \frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \oplus B} \oplus : r$$

$$\frac{\Gamma \Rightarrow \Delta, A \quad B, \Pi \Rightarrow \Theta}{A \to B, \Gamma, \Pi \Rightarrow \Delta, \Theta} \to : l \qquad \frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \to B} \to : r$$

$$\frac{A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg A} \neg : r \qquad \frac{\Gamma \Rightarrow \Delta, A}{\neg A, \Gamma \Rightarrow \Delta} \neg : l$$

Inference rules for modalities:

$$\frac{A, \Gamma \Rightarrow \Delta}{\Box A, \Gamma \Rightarrow \Delta} \Box : l \qquad \frac{\boxdot \Gamma, \Box \Delta \Rightarrow B}{\boxdot \Gamma, \Box \Delta \Rightarrow \Box B} \Box : r$$

$$\frac{A, \Gamma \Rightarrow \Delta}{\boxdot A, \Gamma \Rightarrow \Delta} \boxdot : l \qquad \frac{\boxdot \Gamma \Rightarrow B}{\boxdot \Gamma \Rightarrow \boxdot B} \boxdot : r$$

$$\frac{\boxdot A, \boxdot A, \Gamma \Rightarrow \Delta}{\boxdot A, \Gamma \Rightarrow \Delta}\ \textit{contraction} \qquad \frac{\Gamma \Rightarrow \Delta}{\boxdot A, \Gamma \Rightarrow \Delta}\ \textit{weakening}$$

Inference rules for constants and cut:

$$\frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, 0}\ 0 \qquad \frac{\Gamma \Rightarrow \Delta}{1, \Gamma \Rightarrow \Delta}\ 1 \qquad \frac{\Gamma \Rightarrow \Delta, A \quad A, \Pi \Rightarrow \Theta}{\Gamma, \Pi \Rightarrow \Delta, \Theta}\ \mathit{cut}$$

Here, each capital greek letter ($\Gamma, \Delta, \ldots$) denotes a multiset of formulas; when Γ is $(A_1, \ldots, A_n)$, $\Box\Gamma$ denotes a multiset of the form $(\Box A_1, \ldots, \Box A_n)$; $\boxdot\Gamma$ denotes a multiset of the form $(\boxdot A_1, \ldots, \boxdot A_n)$. We write $\vdash_{\mathsf{LLS4}} S$ or $\vdash^{cf}_{\mathsf{LLS4}} S$ to express that the sequent S is provable in LLS4 or cut-free provable in LLS4, respectively.

Note: 1. It is clear that $\vdash_{\mathsf{LLS4}} \boxdot\varphi \to \Box\varphi$. This is motivated by the following consideration. Both $\boxdot$ and $\Box$ are S4 modalities, but the former has an additional structural feature. Hence, the former can be taken as a special case of the latter, although in our embedding they play different roles.

2. Rules for LL ([4]) can be obtained from LLS4 by omitting $\Box : l$ and $\Box : r$ and by restricting the language to that of LL. ILL can be obtained from LL by omitting $\neg : l$ and $\neg : l$ and omitting $\neg$ from the language of LL.

3. Rules for MALL ([4]) can be obtained from LL by omitting $\boxdot : l$ and $\boxdot : r$, contraction, weakening and by restricting the language to that of MALL.

4. Rules for a subsystem of LLS4 called "BCS4" can be obtained by adding $\Box$ to the language of MALL and $\Box : l$ and a restricted $\Box : r$ (with $\boxdot\Gamma = \emptyset$).

5. Although we do not have any space to present a proof, let us give the definitions of the two embeddings.

Definition 1. *1. The embedding of IL into ILL:* $(A \to B)^\circ = \boxdot A^\circ \to B^\circ$. *(Other cases are quite trivial. See [4]) 2. The embedding of ILL into LLS4* $*$*: Put* $\Box$ *in front of every subformula except inside the scope of* $\boxdot$.

The composition of $\circ$ and $*$ gives an embedding from ILL into LLS4.

Theorem 1. *(Cut-Elimination for LLS4) Let* S *be any sequent of LLS4. If* $\vdash_{\mathit{LLS4}} S$, *then* $\vdash^{cf}_{\mathit{LLS4}} S$.

Proof. We can apply the usual proof as in Takeuti [9] for LLS4. We introduce the following form of cut for the cut formula $\boxdot A$.

$$\frac{\Gamma \Rightarrow \Delta, \boxdot A \quad \overbrace{\boxdot A, \ldots, \boxdot A}^{n\ many}, \Pi \Rightarrow \Theta}{\Gamma, \Pi \Rightarrow \Delta, \Theta}\ \mathit{cut}$$

Let P be a proof of S in LLS4 where the cut rule occurs only in the last step. We proceed by the double induction on the *degree* and the *rank* of P, denoted by $d(P)$ and $r(P)$, respectively. $d(P)$ is the number of the logical symbols of the cut formula, and $r(P)$ is the number of the related occurrences of the cut formula in P. (We refer to [9] for a precise definition for *rank*.) See the appendix 2 for the details of the proof.

3 Substructural Logic of Proofs

We move on to substructural logic of proofs. We present two systems MALLP and LLP. Let us first give a specification of the language of LLP.

$s ::= x|c|!s|s \cdot t|\mathbf{s} \cdot t|s \cdot \mathbf{t}|s + t|\mathbf{s} + t|s + \mathbf{t}|s\sharp t|\mathbf{s}\sharp t|s\sharp\mathbf{t}.$
$\mathbf{s} ::= \mathbf{x}|\mathbf{c}|!\mathbf{s}|\mathbf{s} \cdot \mathbf{t}|\mathbf{s} + \mathbf{t}|\mathbf{s}\sharp\mathbf{t}|\mathbf{s}\flat\mathbf{t}.$

$A ::= 1|\bot|\top|0|P|A \wedge B|A \otimes B|A \vee B|A \oplus B|A \to B|\neg A|s : A|\mathbf{s} : A.$

Proof terms consisting only of boldface terms are "structural terms," which have the structural capacity. All other terms are "non-structural terms, " which have no structural capacity. Also, the language of MALLP is a sublanguage of that of LLP obtained by removing all the boldface terms from that of LLP.

We present Hilbert-style axiomatizations of MALLP and LLP in the following.

MALLP: MALLP is obtained from MALL*, which is the auxiliary system for MALL obtained by adding derivations from assumptions as primitive via "internal consequence relation" in [2], by adding the following axiom schemata and inference rules in the language of MALLP. MALL* is introduced to prove the Lifting Lemma smoothly. (We give a Hilbert style system for MALL and MALL* in Appendix 1.)

A1. $s : A \to A$
A2. $s : (A \to B) \to (t : A \to (s \cdot t) : B)$
A3. $s : A \to !s : (s : A)$
A4. $s : A \to (s + t) : A$; $t : A \to (s + t) : A$

R3. $$\frac{\overline{u} : \Gamma \vdash t : A \qquad \overline{u} : \Gamma \vdash s : B}{\overline{u} : \Gamma \vdash t\sharp s : A \wedge B}$$

R4. $\vdash c : A$, where A is an axiom of MALLP and c is a constant.

Note: The new operation $\sharp$ is essentially the same as the pairing operation, which is indispensable to prove the Lifting Lemma for the substructural logic of proofs without the axiom corresponding to weakening, for then we would not have any proof operation corresponding to the rule of adjunction.

LLP: LLP is obtained from MALLP by extending the language to that of LLP and by adding the following axiom schemata and inference rules in the language of LLP.

A5. $(\mathbf{s} : A \to (\mathbf{s} : A \to B)) \to (\mathbf{s} : A \to B)$
A6. $B \to (\mathbf{s} : A \to B)$
A7. $\mathbf{s} : u : A \to (\mathbf{t} : v : B \to (\mathbf{s}\flat\mathbf{t}) : (u\sharp v) : (A \wedge B))$

R5. $\vdash \mathbf{c} : d : A$, where A is an axiom of LLP, $\mathbf{c}$ is a structural constant, and d is a non-structural constant.

The intuitive meaning of the operation $\flat$ in A7 is to 'witness' the internalization of an application of adjunction by non-structural term. Note that the proof-operation $\flat$ is applicable only to structural terms. Also, the meaning of R5 is to 'witness' the axiom necessitation by a structural constant. The axioms used in R4 in LLP are extended to those of LLP. Terms in A1-4 and R3, 4 can be structural or non-structural in LLP. The axiom A7 and the rule R5 are introduced in order to prove the realization theorem smoothly.[3]

Now we show that BCS4 and LLS4 are realizable in MALLP and LLP, respectively. A *realization* of a formula φ of BCS4 or LLS4 is defined to be a replacement of each occurrence of structural modality $\boxdot$ in φ by a structural term and each occurrence of non-structural modality $\Box$ in φ by a non-structural term, preserving derivability. By φ^r we denote the result of a realization r of φ.

Theorem 2. *(Realization of BCS4 and LLS4)*
1. $\vdash_{BCS4} \varphi$ *if and only if, for some realization* r, $\vdash_{MALLP} \varphi^r$.
2. $\vdash_{LLS4} \varphi$ *if and only if, for some realization* r, $\vdash_{LLP} \varphi^r$.

We prove the theorem in a way similar to the case of S4 in [1]. The parts 'if' in both 1 and 2 are shown by the 'forgetful' projection. We can show by an easy inductive argument that from a given proof of φ^r in MALLP or LLP, we can obtain a proof φ in MALLP or LLP, respectively, by replacing all the occurrences of non-structural terms and structural terms by $\Box$ and $\boxdot$, respectively.

For the part 'only if' of 1, we can apply the method of [1] straightforwardly. However, for 'only if' of 2, we need to make sure that both structural modality and non-structural modality are appropriately realized. For this purpose, we needed to introduce an additional proof operation $\flat$, a new axiom, and a new rule.

We prove Theorem 2. Let us first observe the following facts.

($\natural$1) For any structural term $\mathbf{s}$, there is a non-structural term t such that $\vdash_{\mathsf{LLP}} \mathbf{s} : A \to t(\mathbf{s}) : \mathbf{s} : A$.

($\natural$2) For any structural term $\mathbf{s}$, there are a structural term $\mathbf{t}$ and a non-structural term u such that $\vdash_{\mathsf{LLP}} \mathbf{s} : A \to \mathbf{t}(\mathbf{s}) : u(\mathbf{s}) : A$.

For ($\natural$1). Take a non-structural constant c so that $\vdash_{\mathsf{LLP}} c : (\mathbf{s} : A \to \mathbf{s} : A)$, by $R4$. As $\mathbf{s} : A \to !\mathbf{s} : \mathbf{s} : A$ is an axiom, by using A2, we have $\vdash_{\mathsf{LLP}} \mathbf{s} : A \to (c{\cdot}!\mathbf{s}) : \mathbf{s} : A$.

For ($\natural$2). Take constants $c, \mathbf{d}, \mathbf{e}$ so that $\vdash_{\mathsf{LLP}} \mathbf{d} : c : (\mathbf{s} : A \to A)$ and $\vdash_{\mathsf{LLP}} \mathbf{e} : (\mathbf{s} : A \to !\mathbf{s} : \mathbf{s} : A)$, by R4, 5. Then, for some $\mathbf{w}$, $\vdash_{\mathsf{LLP}} \mathbf{w}(\mathbf{e}, \mathbf{d}) : (\mathbf{s} : A \to (c{\cdot}!\mathbf{s}) : A)$.[4] Finally, using A3 and A2, $\vdash_{\mathsf{LLP}} \mathbf{s} : A \to (\mathbf{w}(\mathbf{e}, \mathbf{d}){\cdot}!\mathbf{s}) : (c{\cdot}!\mathbf{s}) : A$.

Lemma 1. *(Lifting Lemma for MALLP, LLP) Let* L *denote MALLP or LLP.*
1. If $\overline{s} : \Gamma \vdash_L A$, *then* $\overline{s} : \Gamma \vdash_L t(\overline{s}) : A$, *for some non-structural term* t.

[3] It turns out to be possible to realize LLS4 in a system without the axiom A7. However, the proof of the realization in this case would become much more complicated, so we omit it here due to lack of space.

[4] This term $\mathbf{w}(\mathbf{e}, \mathbf{d})$ has the form $(\mathbf{g} \cdot \mathbf{e}) \cdot (\mathbf{d} \cdot \mathbf{f})$, where $\mathbf{f}$: [c: ($\mathbf{s}$:A→A) → (!$\mathbf{s}$: $\mathbf{s}$:A) →c·!$\mathbf{s}$:A)] and $\mathbf{g}$: [($\mathbf{s}$:A→!$\mathbf{s}$: $\mathbf{s}$:A) → ((!$\mathbf{s}$: $\mathbf{s}$:A→c·!$\mathbf{s}$:A) → ($\mathbf{s}$:A→c·!$\mathbf{s}$:A))].

2. If $\overline{\mathbf{s}} : \Gamma, \overline{t} : \Delta \vdash_{LLP} A$, *then* $\overline{\mathbf{s}} : \Gamma, \overline{t} : \Delta \vdash_{LLP} u(\overline{\mathbf{s}}, \overline{t}) : A$, *for some non-structural term* u.

3. If $\overline{\mathbf{s}} : \Gamma \vdash_{LLP} A$, *then* $\overline{\mathbf{s}} : \Gamma \vdash_{LLP} \mathbf{t}(\overline{\mathbf{s}}) : A$, *for some structural term* $\mathbf{t}$.

Proof. We treat 2 and 3. The proof for 1 is similar.

(For 2) By induction on the length of a proof of $\overline{\mathbf{s}} : \Gamma, \overline{t} : \Delta \vdash_{\mathsf{L}} A$ in LLP.

When A is an axiom, we take a non-structural constant c such that $c : A$ is provable by R4. When A is some $\mathbf{s} : B \in \overline{\mathbf{s}} : \Gamma$, we have a non-structural term $t(\mathbf{s})$ such that $t(\mathbf{s}) : \mathbf{s} : A$ is provable by ($\natural 1$). When A is some $t : B \in \overline{t} : \Delta$, by using an axiom $t : B \to !t : t : B$, we can derive $!t : t : B$. In the induction step, we treat the rules R2, R3. The proofs for R1, R4 are similar. For R2. By the induction hypothesis, we can derive $s : A$ and $t : B$ for some non-structural s, t. We obtain $(s \sharp t) : (A \wedge B)$ by R3. For R3. When $(s \sharp t) : (A \wedge B)$ is provable with non-structural $s \sharp t$, we can derive $!(s \sharp t) : (s \sharp t) : (A \wedge B)$ by A3. When $(\mathbf{s} \sharp \mathbf{t}) : (A \wedge B)$ is provable with structural $\mathbf{s} \sharp \mathbf{t}$, we have a non-structural term $u(\mathbf{s} \sharp \mathbf{t})$ such that $u(\mathbf{s} \sharp \mathbf{t}) : (\mathbf{s} \sharp \mathbf{t}) : (A \wedge B)$ is provable, by ($\natural 1$).

(For 3) By induction on the length of a proof of $\overline{\mathbf{s}} : \Gamma \vdash_{\mathsf{LLP}} A$ in LLP. We treat only the case when R3 or R4 is applied. The other cases can be proved similarly to the above cases. For R3. When $(\mathbf{s} \sharp \mathbf{t}) : (A \wedge B)$ is provable with structural $(\mathbf{s} \sharp \mathbf{t})$, we can derive $!(\mathbf{s} \sharp \mathbf{t}) : (\mathbf{s} \sharp \mathbf{t}) : (A \wedge B)$ by A3. When $(s \sharp t) : (A \wedge B)$ is provable with non-structural $s \sharp t$ from $s : A$ and $t : B$, by the induction hypotheses, we have $\mathbf{u} : s : A$ and $\mathbf{v} : t : B$ provable with some structural $\mathbf{u}, \mathbf{v}$. By using A7, we can derive $(\mathbf{u} \flat \mathbf{v}) : (s \sharp t) : (A \wedge B)$. For R4. When $c : A$ is provable by R4, A is an axiom, and so, we have structural $\mathbf{d}$ such that $\mathbf{d} : c : A$ is provable. ■

Corollary 1. *(Constructive necessitation for MALLP, LLP)*

Let L *denote* MALLP *or* LLP.

1. If $\vdash_{\mathsf{L}} A$, *then* $\vdash_{\mathsf{L}} s : A$, *for some non-structural term* s.

2. If $\vdash_{LLP} A$, *then* $\vdash_{LLP} \mathbf{s} : A$, *for some structural term* $\mathbf{s}$.

Proof. (Theorem 2) Let us now prove the theorem, but we first introduce some technical terms. We use the standard terminology of the polarity of an occurrence of a formula. Note that in a cut-free proof, the polarity is always preserved. For a sequent $S \equiv \Gamma \Rightarrow \Delta$, let S^r denote $\Gamma^r \Rightarrow \Delta^r$. A realization is said to be *normal* if each negative occurrence of $\Box$ and $\boxdot$ is replaced by a structural proof-variable and a non-structural proof-variable, respectively. A realization for S is normal if that for $\otimes\Gamma \to \oplus\Delta$ is normal. We say that S^r is provable in a system if $\otimes\Gamma^r \to \oplus\Delta^r$ is provable in it. We say that an occurrence of $\Box$ in a premise and a corresponding occurrence of $\Box$ in a conclusion "related." Taking the reflexive transitive closure of this relation on occurrences of $\Box$, we consider an equivalence class of "related" $\Box$es, and we call them a *family* of occurrences of $\Box$. We call a family "essential" if it contains at least one occurrence of $\Box$ introduced by $\Box : r$. Since the relation is an equivalence relation, we can consider a partition of occurrences of $\Box$ in a proof tree.

Suppose that a sequent S is provable in M (where M = BCS4 or LLS4). By Theorem 1, we may assume that there is a cut-free proof P of S in M. We shall construct a proof of S^r in L with a normal r. The construction goes as follows.

(Step 1) For each negative family of $\Box$ and each nonessential family of $\Box$, substitute a non-structural variable, for all occurrences of $\Box$ of the family. Also, for each negative family of $\boxdot$ and each nonessential family of $\boxdot$, substitute a structural variable for all occurrences of $\boxdot$ of the family.

(Step 2) For each essential family of $\Box$, substitute a non-structural term $x_1+\cdots+x_n$ for all occurrences of $\Box$ of the family. Each x_i is called a provisional variable. n is the number of applications of the rule $\Box : r$ for the family. Also, for each essential family of $\boxdot$, substitute a structural term $\mathbf{y_1}+\cdots+\mathbf{y_m}$ for all occurrences of $\boxdot$ of the family. m is the number of applications of the $\boxdot : r$ for the family.

(Step 3) Proceed from top to bottom in the proof figure. For a $\Box : r$, we have the realization of the upper and lower sequents: $\overline{\mathbf{s}} : \Gamma, \overline{t} : \Delta \vdash_{\mathsf{L}} A$ and $\overline{\mathbf{s}} : \Gamma, \overline{t} : \Delta \vdash_{\mathsf{L}} (x_1+\cdots+x_i+\cdots+x_n) : A$ in (Step 1, 2). By applying the Lifting Lemma, construct a non-structural term u such that $\overline{\mathbf{s}} : \Gamma, \overline{t} : \Delta \vdash_{\mathsf{L}} u : A$. (When L is MALLP, Γ is empty.) When the $\Box : r$ is the ith one among all $\Box : r$ in the family, substitute u_i for all the occurrences of the same provisional variable x_i. Then we have, e.g., $\overline{\mathbf{s}} : \Gamma, \overline{t} : \Delta \vdash_{\mathsf{L}} (x_1+\cdots+u_i+\cdots+x_n) : A$. Repeat this procedure until we substitute all of u_i for all of the provisional variables x_i. Then, e.g., we obtain $\overline{\mathbf{s}} : \Gamma, \overline{t} : \Delta \vdash_{\mathsf{L}} (u_1+\cdots+u_i+\cdots+u_n) : A$.

Similarly, for each $\boxdot : r$, we have the realization of the upper and lower sequents: $\overline{\mathbf{s}} : \Gamma \vdash_{\mathsf{LLP}} A$ and $\overline{\mathbf{s}} : \Gamma \vdash_{\mathsf{LLP}} (\mathbf{x_1}+\cdots+\mathbf{x_j}+\cdots+\mathbf{x_m}) : A$ in (Step 1, 2). By the Lifting Lemma, construct a structural term $\mathbf{v}$ such that $\overline{\mathbf{s}} : \Gamma \vdash_{\mathsf{LLP}} \mathbf{v} : A$. When the $\boxdot : r$ is the jth one among all $\boxdot : r$ in the family, substitute $\mathbf{v_j}$ for all the occurrences of the same provisional variable $\mathbf{x_j}$. Then we have, e.g., $\overline{\mathbf{s}} : \Gamma \vdash_{\mathsf{LLP}} (\mathbf{x_1}+\cdots+\mathbf{v_j}+\cdots+\mathbf{x_m}) : A$. Repeat this procedure until we substitute all of $\mathbf{v_j}$ for all of the provisional variables $\mathbf{x_j}$. Then, we obtain, e.g., $\overline{\mathbf{s}} : \Gamma \vdash_{\mathsf{LLP}} (\mathbf{v_1}+\cdots+\mathbf{v_j}+\cdots+\mathbf{v_m}) : A$.

In this procedure, each formula in P is replaced by a formula in the language of MALLP or LLP and it is easy to check that each sequent consisting of these formulas is either an axiom or derivable from the upper sequent. Thus, for a given sequent S provable in M, we obtain a proof of S^r with a normal realization r in L. This completes the proof of Theorem 2 ■

4 Prehistoric-Loop-Free Proofs

Yu ([10], [12]) defines a notion of prehistoric loop. Intuitively, this means that an introduction of an occurrence of a modal operator essentially requires that it be related to another occurrence of the modal operator in the same "family" which precedes it. Yu ([10], [12]) showed that such a loop is a necessary condition that any realization of the modal formula essentially requires a self-referential constant specification. Applying this idea, we identify a natural subsystem of $\mathsf{S4}$ in the hierarchy of modal substructural logics in Došen [3]. We start from giving necessary definitions.

The phenomenon that two occurrences of symbols in premise(s) share the same corresponding occurrence in the conclusion is called a *unification*. Contraction is a structural rule that makes this happen, and also side formulas in the binary rules for additive logical constants typically make this happen.

A Gentzen-style proof (tree) is denoted by $\mathcal{T}$, with each of its node (a sequent) denoted s_1, s_2, $\cdots$, and its conclusion-premise relation denoted by R. The root of a proof tree is the conclusion sequent of the proof. For any proof $\mathcal{T}$ and any sequent s in $\mathcal{T}$. $\mathcal{T} \upharpoonright s$ means the subproof of s in $\mathcal{T}$. Following the definition of a family of occurrences of $\Box$ in section 3, we use a notation $\Box_i$ where $i = 1, 2, 3. \ldots$. By this, we mean that the occurrence of $\Box$ indexed by i belongs to a family i of occurrences of $\Box$. A family of occurrences of $\Box$ has its own polarity (negative or positive). We denote a positive family by $\Box_i^+$ and a negative family by $\Box_i^-$ In the following, we say a family of $\Box_i^+$ or $\Box_i^-$ occurs in φ if there is an occurrence of $\Box_i$ in φ (we write this as $\varphi(\Box_i)$).

Definition 2 (Prehistoric graph [12]).
In a BCS4 proof $\mathcal{T}$, the prehistoric graph $\mathcal{P}(\mathcal{T})$ is defined as the directed graph $\mathcal{P}(\mathcal{T}) := (F, \prec_L, \prec_R, \prec)$, where F is the set of positive families in $\mathcal{T}$, all of $\prec_L$, $\prec_R$, $\succ$ are binary relations on F, $\prec := \prec_L \cup \prec_R$, and

$$\prec_L := \{(i,j) |\ \frac{\Box^- \Theta(\Box_i^+) \Rightarrow \varphi}{\Box^- \Theta(\Box_i^+) \Rightarrow \Box_j^+ \varphi}\ (\Box : r) \text{ is in } \mathcal{T}\ \}$$

$$\prec_R := \{(i,j) |\ \frac{\Box^- \Theta \Rightarrow \varphi(\Box_i^+)}{\Box^- \Theta \Rightarrow \Box_j^+ \varphi(\Box_i^+)}\ (\Box : r) \text{ is in } \mathcal{T}\ \}$$

Lemma 2. *In a proof $\mathcal{T}$, each family has a unique occurrence in the root.*

Proof. In any rule in those systems, each occurrence in a premise has exactly one corresponding occurrence in the conclusion. ■

Due to the limitation of space, we omit proofs of some statements below up to Proposition 1. (The details of these can be found in [12].)

Lemma 3. *In $\mathcal{P}(\mathcal{T})$ of a proof $\mathcal{T}$, for any $i \in F$, $(i,i) \notin \prec_R$.*

Lemma 4. *In $\mathcal{P}(\mathcal{T})$ of a proof $\mathcal{T}$, for any i, j, $k \in F$, (1) if $k \prec_R j \prec_L i$ and (2) $k \prec_R j \prec_R$ then $k \prec_R i$*

Definition 3 ([12]).
A (left, right) prehistoric graph has a loop *w.r.t. $\prec$ ($\prec_L$, $\prec_R$) if there is a finite sequence of $i_1 \prec i_2 \prec \cdots \prec i_n \prec i_1$.*

Theorem 3 ([12]).
In $\mathcal{P}(\mathcal{T})$ of a proof of $\mathcal{T}$, (1) $(F, \prec_R)$ has no loop, and (2) $(F, \prec)$ has a loop if and only if $(F, \prec_L)$ has a loop.

Obviously, if we have no introduction of a new edge and no unification of occurrences of modal operator by applying a rule, it introduces no loop.

Proposition 1. *1. $\otimes : r$, $\otimes : l$, $\oplus : r$, $\oplus : l$, $\rightarrow : r$, and $\rightarrow : l$ do not introduce any loop. 2. $\Box : r$ and $\Box : l$ introduce no loop, either.*

Yu [10] observes that in his formulation of S4 in a G3-type sequent calculus only his versions of $\rightarrow$: l(context sharing), $\Box : l$ (contraction built-in[5]) rules can possibly bring in loop. It is obvious that if the form of $\Box : l$ is not the one in which contraction is built-in, then $\Box : l$ involves no unification. Also, note that $\Box : r$ introduces a new edge, but clearly this introduces no unification.

Lemma 5. *In no sequent in a BCS4 proof, more than one occurrence of $\Box^+$ of the same family can occur in it.*

Proof. Note that there is exactly one occurrence of $\Box^+$ from one family in the root, due to the definition of a family. To show this lemma, it is sufficient to observe that no rule in BCS4 can make two occurrences of a formula into one. ■

In addition to Yu's observation that there is no direct right loop, we can also observe that in BCS4, there is no direct left loop.

Lemma 6. *In $\mathcal{P}(\mathcal{T})$ of a proof $\mathcal{T}$, for any $i \in F$, $(i, i) \notin \prec_L$.*

Proof. Officially, it is shown by the induction of the length of proofs. But it suffices to observe that there is no rule in these systems that relates two occurrences of $\Box^+$ in the premises can come in the same occurrence of $\Box^+$. ■

Since BCS4 enjoys cut-elimination and the subformula property, we can state the following.

Proposition 2. *In BCS4, any $\Box$ formula in each positive essential family of $\Box_i^+$ is introduced as a principal formula of $\Box : r$ only in one place in each branch s.*

Proof. In BCS4, only binary additive rules can introduce a unification, but such a case occurs only at a node of a proof tree where two branches meet. Hence, for each branch, any $\Box$ formula in each positive family of $\Box_i^+$ is introduced only once via $\Box : r$. ■

Yu's theorem (theorem 3) concerning equivalence between having a loop and having a left loop still holds here. Hence, we can conclude that there is no loop in s of $\mathcal{T}$ at all.

Theorem 4. *Let $\mathcal{T}$ be a BCS4 proof. No branch s in $\mathcal{T}$ can have a left-prehistoric loop. Hence, no branch s in $\mathcal{T}$ can have a prehistoric loop.*

This theorem on the loop-free nature of BCS4 proofs is stated only for a branch s. However, to prove an interesting property of a proof, we focus on the multiplicative fragment of BCS4. We call it mBCS4. But then the following hold for mBCS4. since (1) no rules in these allow us to introduce a prehistoric loop in one branch and (2) there is no additive (context-sharing) binary rule that makes a unification possible.

Lemma 7. *In an mBCS4 derivation, each positive family of $\Box_i^+$ can be introduced only once in the entire proof tree.*

Theorem 5. *An mBCS4 proof cannot have a left prehistoric loop.*

[5] This rule has the form $\dfrac{\theta, \Box\theta, \Gamma \Rightarrow \Delta}{\Box\theta, \Gamma \Rightarrow \Delta}$.

5 Non-self-Referential Constant Specifications

Now let us show that there is a realization of a prehistoric loop-free proof that does not require a self-referential constant specification. Let us first give the definition of self-referential constant specification.

Definition 4 (Self-referentiality in constant specifications [6])
A constant specification $\mathcal{CS}$ is self-referential *if $\mathcal{CS}$ has at least one subset of the form $\{c_1 : A_1(c_2), \ldots, c_{n-1} : A_{n-1}(c_n), c_n : A_n(c_1)\}$. A constant specification $\mathcal{CS}$ is* non-self-referential*, otherwise.*

Theorem 6. *There is a realization of r such that $\mathsf{mBCS4} \vdash \varphi$ if and only if $\mathsf{L} \vdash \varphi^r$ which satisfies the following two conditions.*
1. L is the multiplicative fragment of MALLP (called "MLLP.")
2. The constant specification contains no self-referential constant specification.

Proof. ($\Longrightarrow$) (Proof sketch) Let us first prove the harder part.

(1) We prove that there exists a desired realization, following a refined version of the realization algorithm. We refine the original realization algorithm given in [1] via the order of families of modal operators. The order of families can be given by keeping track of an introduction of a positive occurrence of a $\Box^+$ by using a function called "ϵ-function."

Let us give a few necessary definitions about this function here. Using Yu's notation ([12]), we write $\epsilon(i, j)$. This means a function that assigns a natural number to the j-th application of $\Box : r$ in the i-th family of positive occurrences of the modal operator. The j-th application of $\Box : r$ in the family i is denoted by $(\Box : r)_{ij}$. The premise and the conclusion of $(\Box : r)_{ij}$ are denoted by I_{ij} and O_{ij}, respectively. Since we have only one case of principal application of $\Box : r$ in one family, it suffices to have $\epsilon(i)$. Let R^+ be a transitive closure of a relation expressing the relationship between the premise and the conclusion of $\Box : r$. Here clearly it follows by definition that $O_{i_1} R^+ O_{i_2}$ if and only if $\epsilon(i_1) < \epsilon(i_2)$. Thus, a new constant c_i is associated with the unique $(\Box : r)_i$. $\epsilon(i)$ is the number associated with this constant.

Claim. $h_1 \prec h_2$ implies $\epsilon(h_1) < \epsilon(h_2)$

Proof. (proof sketch of the claim) There must be a i_1 such that for all h, $h \not\prec i_1$ (it is like an innermost family). Otherwise, there is a loop in a proof. Then, excluding i_1 from the set of all relevant families, we can pick i_2 such that $\epsilon(i_1) < \epsilon(i_2)$, and so on. Since the set of families is at most finite, this can always be done. Hence, the families can be linearly ordered according to the order of applications of $\Box : r$ in them.

(2) To prove that our realization does not rely on any self-referential constant specification, we show that our refined realization which ensures that the substitution for a provisional variable can be done strictly according to the order

of the applications of $\Box : r$ works. Note that, in the general case of realization algorithm used in section 3, it is not possible to exclude a case in which an occurrence of a provisional variable can be substituted only when a relevant application of the Lifting Lemma occurs below the provisional variable in a proof tree. Let us define now the following notion.

Definition 5 ([12]). *An application of Axiom Necessitation* $\dfrac{}{c_i : A_i}$ *is ϵ-allowed if (1) A_i contains no provisional variables and (2) A_i contains no $c_{i'}$ s.t. $\epsilon(i) \leq \epsilon(i')$.*

(3) The description of the realization algorithm can be given as follows. 1) Take a cut-free proof of mBCS4. 2) Replace all the negative occurrences of modal operators (and positive occurrences of modal operators belonging to an non-essential family introduced by weakening) with proof variables by using the same variable for the occurrences related to a $\Box$ at the root node of a proof tree. Replace all positive occurrences of modal operators in an essential family by the same provisional variable. (Note that since we have only one possible application of $\Box : r$ for one family, we do not need to introduce $+$ part of the realization at all.) 3) At every application of $\Box : r$, we substitute a relevant provisional variable by an appropriate proof term by applying the Lifting Lemma.

(4) We show that the realization algorithm produces a proof in the corresponding substructural logic of proofs (i.e. MLLP). We use an induction on $\epsilon(i)$ and a subinduction on the structure of a subtree of a proof tree to ensure that both the substitution steps and the subderivations between substitution steps work as desired.

(4.1) We carry out substitutions of an appropriate proof terms with ϵ-allowed c.s. into provisional variable based on induction over $\epsilon(i)$.

By using IH for $\epsilon(i_1) - 1$, we have a provisional-variable-free derivation of $I_i^{\epsilon(i)-1}$ with an ϵ-allowed c.s. Then the step from $I_i^{\epsilon(i)-1}$ to $O_i^{\epsilon(i)-1}$ can be treated as follows. 1) Apply the Lifting Lemma to $I_i^{\epsilon(i)-1}$ (the c.s. used here is ϵ-allowed, since all the constants in the c.s. used in Lifting are fresh, according to (1)).[6] 2) By carrying out the substitution, we can replace the provisional variable by a provisional-variable-free term. Then we get $\mathcal{T}^i$ from what is obtained from $\mathcal{T}^{i-1}$ possibly by some other steps in the derivation (here $\mathcal{T}^i$ stands for a subtree of $\mathcal{T}$ in which the substitution for $\epsilon(i)$ is done). This completes the i-th step of induction in the substitution procedure in the realization algorithm.

(4.2) Between the previous substitution for obtaining $\mathcal{T}^{i-1}$ and the substitution for obtaining $\mathcal{T}^i$, we show that all the steps of the derivation here can be mimicked in the Hilbert-style system of the corresponding substructural logic proofs (MLLP). The proof of this part is tedious but a routine and essentially the same as in [12].

Hence, we have completed both the description of the algorithm and the proof that the algorithm produces a proof in MLLP with ϵ-allowed c.s in such a way that all the substitutions are done following the order of $\epsilon(i)$.

[6] Our procedure is simpler than Yu's since we do not need $+$ in our realization and, unlike in G3-style systems, our modal rules do not incorporate weakening.

(5) We now show that our constructed proof in MLLP is non-self-referential, i.e., the ϵ-allowed c.s. that we have used (we call it $\mathcal{CS}''$ is non-self-referential. To prove this, suppose $\mathcal{CS}''$ is self-referential. Namely, assume the following holds.

$$\{c_{i_1,k_1}{:}A_{i_1,k_1}(c_{i_2,k_2}), \ldots, c_{i_{z-1},k_{z-1}}{:}A_{i_{z-1},k_{z-1}}(c_{i_z,k_z}), c_{i_z,k_z}{:}A_{i_z,k_z}(c_{i_1,k_1})\} \subseteq \mathcal{CS}''.$$

Since $\mathcal{CS}''$ is ϵ-allowed, we have $\epsilon(i_z) < \epsilon(i_{z-1}) < \cdots < \epsilon(i_1) < \epsilon(i_z)$. But this is impossible. Hence, $\mathcal{CS}''$ is non-self-referential.

($\Longleftarrow$) Given any derivation in the multiplicative fragment of MALLP, we can apply the forgetful projection. It can be shown by an easy inductive argument that the forgetful projection gives a proof in mBCS4 ■

6 Discussions

1. We acknowledge that there is another work independently done concerning the issue of non-self-referentiality of contraction-less fragment of S4 ([8]). However, the work only considers a fragment of G3-style formulation of S4. The author does not systematically consider how this fragment can be located in the hierarchy of modal substructural logics. As a result, it is not clear what kind of logic we obtain if we take a forgetful projection of the fragment of G3-style formulation of S4. On the other hand, our result presented here is a consequence of a more systematic consideration about how realization works in the hierarchy of modal substructural logics. Hence, it is obvious that the resulting logic by applying the forgetful projection to MLL is exactly mBCS4. Also, it is worth noting that the realization does not require + operation, either.
2. Although we formulate only BCS4 and MLLP in this paper mainly due to the limitation of space, it is quite obvious that the same method works for the multiplicative fragment of BCKS4. In addition, our proof method also suggests a natural conjecture that multiplicative and additive BCS4 (and BCKS4) can also be realized without using self-referential constant specification.
3. Not only has it been observed that contraction-free logics have an important connection to polynomial time computation, but it was also observed by Gödel already in 1930's that the intended interpretation (Brouwer-Heyting-Kolmogorov interpretation) of intuitionistic implication contains some sort of "impredicativity." Self-referentiality in realization of S4 seems to have an important connection with this impredicativity of intuitionistic logic through contraction.[7]

[7] Yu [11] has recently proved that self-referentiality also occurs in the range of Gödel embedding from intuitionistic propositional logic into S4.

Appendix 1

(1) A Hilbert-style axiomatic system for MALL (a notational variant from [2]).

Axioms schemata:
1. $A \to A$
2. $(A \to B) \to ((B \to C) \to (A \to C))$
3. $(A \to (B \to C)) \to (B \to (A \to C))$
4. $\neg\neg A \to A$
5. $(A \to \neg B) \to (B \to \neg A)$
6. $(A \to (B \to A \otimes B))$
7. $(A \to (B \to C)) \to ((A \otimes B) \to C)$
8. 1
9. $1 \to (A \to A)$
10. $A \to (\neg A \to 0)$
11. $\neg 0$
12. $(A \oplus B) \to (\neg A \to B)$
13. $(\neg A \to B) \to (A \oplus B)$
14. $(A \wedge B) \to A$
15. $(A \wedge B) \to B$
16. $((A \to B) \wedge (A \to C)) \to (A \to (B \wedge C))$
17. $A \to (A \vee B)$
18. $B \to (A \vee B)$
19. $((A \to C) \wedge (B \to C)) \to ((A \vee B) \to C))$
20. $A \to \top$
21. $\bot \to A$

Inference rules: R1. $\dfrac{A \qquad A \to B}{B}$ R2. $\dfrac{A \qquad B}{\vdash A \wedge B}$

(2) The internal consequence relation $\vdash$ for LL, which makes it possible to talk about a derivation from assumption naturally, can be introduced as follows.

Internal consequence : $A_1, \ldots, A_n \vdash B$ if and only if $\vdash A_1 \to (\cdots (A_n \to B) \cdots)$.

(3) To facilitate the proof of the Lifting Lemma, we introduce an auxiliary axiomatization of MALL* which takes $\vdash$ as primitive. The axioms are the same as those of MALL, but the inferences rules are formulated as follows.

Inference rules: R1*. $\dfrac{\Gamma_1 \vdash A \qquad \Gamma_2 \vdash A \to B}{\Gamma_1, \Gamma_2 \vdash B}$ R2*. $\dfrac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$

Given the foregoing definition of $\vdash$, the equivalence of MALL and MALL* is obvious (it suffices to show the derivability of R1* and R2* by MALL and the definition of $\vdash$).

Appendix 2

In this appendix, we show the details of our proof of cut-elimination for LLS4. We examine only the cases when the cut formula is a modal formula. First, we handle the cases when the cut formula is $\boxdot A$ and $r(P) = 2$. We have a few cases. Below, we illustrate the proof-transformation to be carried out for each case.

(1)
$$\dfrac{\dfrac{\dfrac{\boxdot\Gamma \Rightarrow A}{\boxdot\Gamma \Rightarrow \boxdot A}\,\boxdot : r \quad \dfrac{A, \Pi \Rightarrow \Theta}{\boxdot A, \Pi \Rightarrow \Theta}\,\boxdot : l}{\boxdot\Gamma, \Pi \Rightarrow \Theta}\,cut}{}$$

$$\bigtriangledown$$

$$\dfrac{\boxdot\Gamma \Rightarrow A \quad A, \Pi \Rightarrow \Theta}{\boxdot\Gamma, \Pi \Rightarrow \Theta}\,cut$$

(2)
$$\dfrac{\dfrac{\dfrac{\boxdot\Gamma \Rightarrow A}{\boxdot\Gamma \Rightarrow \boxdot A}\,\boxdot : r \quad \dfrac{\Pi \Rightarrow \Theta}{\boxdot A, \Pi \Rightarrow \Theta}\,weak.}{\boxdot\Gamma, \Pi \Rightarrow \Theta}\,cut}{}$$

$$\bigtriangledown$$

$$\dfrac{\Pi \Rightarrow \Theta}{\boxdot\Gamma, \Pi \Rightarrow \Theta}\,weak.$$

$d(P)$ is reduced by one in the transformation of (1). Thus, we can eliminate the new cut inference in (1). Next, we examine the cases when the cut formula is $\boxdot A$ and $r(P) \geq 3$. When the left upper inference is not $\boxdot : r$. We demonstrate the proof-transformation only for one case.

(3)
$$\dfrac{\dfrac{\dfrac{\Gamma \Rightarrow \Delta, B \quad C, \Pi \Rightarrow \Theta, \boxdot A}{B \to C, \Gamma, \Pi \Rightarrow \Delta, \Theta, \boxdot A}\,\to : l \quad (\boxdot A)^n, \Sigma \Rightarrow \Phi}{B \to C, \Gamma, \Pi, \Sigma \Rightarrow \Delta, \Theta, \Phi}\,cut}{}$$

$$\bigtriangledown$$

$$\dfrac{\Gamma \Rightarrow \Delta, B \quad \dfrac{C, \Pi \Rightarrow \Theta, \boxdot A \quad (\boxdot A)^n, \Sigma \Rightarrow \Phi}{C, \Pi, \Sigma \Rightarrow \Theta, \Phi}\,cut}{B \to C, \Gamma, \Pi, \Sigma \Rightarrow \Delta, \Theta, \Phi}\,\to : l$$

When the left upper inference is $\boxdot : r$ and the right upper one is not concerned with the cut formula, the transformation is similar to (3). Otherwise, there are the following three cases.

(4)
$$\dfrac{\dfrac{\dfrac{\boxdot\Gamma \Rightarrow A}{\boxdot\Gamma \Rightarrow \boxdot A}\,\boxdot : r \quad \dfrac{\boxdot A, (\boxdot A)^n, \Pi \Rightarrow \Theta}{(\boxdot A)^n, \Pi \Rightarrow \Theta}\,contr.}{\boxdot\Gamma, \Pi \Rightarrow \Theta}\,cut}{}$$

$$\bigtriangledown$$

$$\dfrac{\dfrac{\boxdot\Gamma \Rightarrow A}{\boxdot\Gamma \Rightarrow \boxdot A}\,\boxdot : r \quad \boxdot A, (\boxdot A)^n, \Pi \Rightarrow \Theta}{\boxdot\Gamma, \Pi \Rightarrow \Theta}\,cut$$

(5)
$$\dfrac{\dfrac{\dfrac{\boxdot\Gamma \Rightarrow A}{\boxdot\Gamma \Rightarrow \boxdot A}\ \boxdot : r \quad \dfrac{(\boxdot A)^n, \Pi \Rightarrow \Theta}{\boxdot A, (\boxdot A)^n, \Pi \Rightarrow \Theta}\ \textit{weak.}}{\boxdot\Gamma, \Pi \Rightarrow \Theta}\ \textit{cut}$$

$$\triangledown$$

$$\dfrac{\dfrac{\boxdot\Gamma \Rightarrow A}{\boxdot\Gamma \Rightarrow \boxdot A}\ \boxdot : r \quad (\boxdot A)^n, \Pi \Rightarrow \Theta}{\boxdot\Gamma, \Pi \Rightarrow \Theta}\ \textit{cut}$$

(6)
$$\dfrac{\dfrac{\dfrac{\boxdot\Gamma \Rightarrow A}{\boxdot\Gamma \Rightarrow \boxdot A}\ \boxdot : r \quad \dfrac{A, (\boxdot A)^n, \Pi \Rightarrow \Theta}{\boxdot A, (\boxdot A)^n, \Pi \Rightarrow \Theta}\ \boxdot : l}{\boxdot\Gamma, \Pi \Rightarrow \Theta}\ \textit{cut}$$

$$\triangledown$$

$$\dfrac{\dfrac{\boxdot\Gamma \Rightarrow A}{\boxdot\Gamma \Rightarrow \boxdot A}\ \boxdot : r \quad A, (\boxdot A)^n, \Pi \Rightarrow \Theta}{\boxdot\Gamma, A, \Pi \Rightarrow \Theta}\ \textit{cut}$$

In all cases (3-6), $r(P)$ is reduced by one, and, hence, we can eliminate the new cut inferences. As to (6), we have not yet reached the desired sequent, which is obtained by further applying the cut in the following way.

$$\dfrac{\dfrac{\boxdot\Gamma \Rightarrow A \quad \boxdot\Gamma, A, \Pi \Rightarrow \Theta}{\boxdot\Gamma, \boxdot\Gamma, \Pi \Rightarrow \Theta}\ \textit{cut}}{\boxdot\Gamma, \Pi \Rightarrow \Theta}\ \textit{contraction}$$

This cut can be eliminated as the $d(P)$ is reduced by one.

Next, we examine the cases when the cut formula is $\Box A$. When the left upper inference is not $\Box : r$, we can handle the cases in a similar way to the case of (4) above. When the left upper inference is $\Box : r$, we apply the following transformation.

(7)
$$\dfrac{\dfrac{\dfrac{\boxdot\Gamma, \Box\Delta \Rightarrow A}{\boxdot\Gamma, \Box\Delta \Rightarrow \Box A}\ \Box : r \quad \dfrac{A, \Pi \Rightarrow \Theta}{\Box A, \Pi \Rightarrow \Theta}\ \Box : l}{\boxdot\Gamma, \Box\Delta, \Pi \Rightarrow \Theta}\ \textit{cut}$$

$$\triangledown$$

$$\dfrac{\boxdot\Gamma, \Box\Delta \Rightarrow A \quad A, \Pi \Rightarrow \Theta}{\boxdot\Gamma, \Box\Delta, \Pi \Rightarrow \Theta}\ \textit{cut}$$

$d(P)$ is reduced by one in (7). So, we can eliminate the new cut inferences in (7). ■

References

1. Artemov, S.N.: Explicit provability and constructive semantics. The Bulletin of Symbolic Logic 7(1), 1–36 (2001)
2. Avron, A.: The semantics and proof theory of linear logic. Theoretical Computer Science 57, 161–184 (1988)
3. Došen, K.: Modal translations in substructural logics. Journal of Philosophical Logic 21(3), 283–336 (1992)
4. Girard, J.: Linear logic. Theoretical Computer Science 50, 1–102 (1987)
5. Gödel, K.: Zum intuitionistischen Aussagenkalkül. Anzeiger Akademie der Wissenschaften Wien, Mathematisch-naturwiss. Klasse 32, 65–66 (1932)
6. Kuznets, R.: Self-referential justifications in epistemic logic. Theory of Computing Systems 46(4), 636–661 (2010)
7. Paoli, F.: Substructural Logics: A Primer. Trends in Logic, vol. 13. Kluwer Academic Pub. (2002)
8. Pulver, C.: Self-referentiality in contraction-free fragments of modal logic S4. MS. Thesis (2010)
9. Takeuti, G.: Proof Theory, 2nd edn. North Holland (1987)
10. Yu, J.: Prehistoric phenomena and self-referentiality. In: Ablayev, F., Mayr, E.W. (eds.) CSR 2010. LNCS, vol. 6072, pp. 384–396. Springer, Heidelberg (2010)
11. Yu, J.: Self-referentiality in the brouwer–heyting–kolmogorov semantics of intuitionistic logic. In: Artemov, S., Nerode, A. (eds.) LFCS 2013. LNCS, vol. 7734, pp. 401–414. Springer, Heidelberg (2013)
12. Yu, J.: Prehistoric graph of modal sequent proof and non-self-referential realization (unpublished manuscript)

Full Lambek Hyperdoctrine: Categorical Semantics for First-Order Substructural Logics

Yoshihiro Maruyama*

Quantum Group, Dept. of Computer Science, University of Oxford
http://researchmap.jp/ymaruyama

Abstract. We pursue the idea that predicate logic is a "fibred algebra" while propositional logic is a single algebra; in the context of intuitionism, this algebraic understanding of predicate logic goes back to Lawvere, in particular his concept of hyperdoctrine. Here, we aim at demonstrating that the notion of monad-relativised hyperdoctrines, which are what we call fibred algebras, yields algebraisations of a wide variety of predicate logics. More specifically, we discuss a typed, first-order version of the non-commutative Full Lambek calculus, which has extensively been studied in the past few decades, functioning as a unifying language for different sorts of logical systems (classical, intuitionistic, linear, fuzzy, relevant, etc.). Through the concept of Full Lambek hyperdoctrines, we establish both generic and set-theoretical completeness results for any extension of the base system; the latter arises from a dual adjunction, and is relevant to the tripos-to-topos construction and quantale-valued sets. Furthermore, we give a hyperdoctrinal account of Girard's and Gödel's translation.

1 Introduction

Categorical logic deconstructs the traditional dichotomy between proof theory and model theory, in the sense that both of them can be represented in certain syntactic and set-theoretical categories (or hyperdoctrines in this paper) respectively. We may thus say that categorical semantics does encompass both proof-theoretic and model-theoretic semantics in terms of philosophy of logic.

Categorical semantics divides into two sub-disciplines: semantics of provability (e.g., semantics via toposes or logoses) and semantics of proofs (e.g., semantics via CCC or monoidal CC). Our focus shall be upon the former wrt. logic and the latter wrt. type theory because we aim at developing categorical semantics for a broad range of logics over type theories, including classical, intuitionistic, linear, and fuzzy logics. Type theories have inherent identities of proofs (or terms), and fully admit semantics of proofs, however, logics in general do not allow semantics of proofs, due to collapsing phenomena on their identities of proofs (for the case of classical logic, refer to the Joyal lemma, e.g., in Lambek-Scott [11]).

Thus, the Curry-Howard paradigm does not make so much sense in this general context of logics over type theories, for the logics of the latter (types) may

* I am indebted to Samson Abramsky and Bob Coecke for both useful discussion and kind encouragement. This work was supported by the Nakajima Foundation.

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 211–225, 2013.

differ from the former original logics (propositions), just as Abramsky-Coecke's type theory of quantum mechanics is distinct from Birkhoff-von Neumann's logic of it. In general, we thus need to treat logic and type theory separately, and the concept of fibred universal algebras does the job, as elucidated below. Aczel's idea of logic-enriched type theory is along a similar line. Fibred algebras to represent logics over monoidal type theories even allow us to reconcile Birkhoff-von Neumann's cartesian logic of quantum propositions and Abramsky-Coecke's monoidal logic (or type theory) of quantum systems; this is future work, however.

Substructural logics over the Full Lambek calculus (FL for short), which encompass a wide variety of logical systems (classical, intuitionistic, linear, fuzzy, relevant, etc.), have extensively been investigated in the past few decades, especially by algebraic logicians in relation to residuated lattices (a major reference is Galatos-Jipsen-Kowalski-Ono [3]). Although some efforts have been made towards the algebraic treatment of logics over quantified FL (see, e.g., Ono [14,15] and references therein), however, it seems that there has so far been no adequate concept of algebraic models of them. Note that complete residuated lattices can only give complete semantics for those classes of substructural predicate logics for which completions (such as Dedekind-MacNeille's or Crawley's) of Lindenbaum-Tarski algebras work adequately (see, e.g., Ono [14,15]); for this reason, complete residuated lattices (or quantales) cannot serve the purpose.

In the context as articulated above, we propose fibred algebras as algebraic models of predicate logic, especially substructural logics over quantified FL. Fibred algebras expand Lawvere's concept of hyperdoctrine [12]. According to Pitts' formulation [16], a hyperdoctrine is a functor (presheaf) $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{HA}$ where $\mathbf{HA}$ is the category of heyting algebras; there are additional conditions on P (and $\mathbf{C}$) to express quantifiers and other logical concepts (for a fibrational formulation of hyperdoctrine, see Jacobs [8]; the two formulations are equivalent via the Grothendieck construction).[1] We may see a hyperdoctrine as a fibred heyting algebra $(P(C))_{C \in \mathbf{C}}$, a bunch of algebras indexed by $\mathbf{C}$.

Now, a fibred algebra is a universal algebra indexed by a category: categorically, it is a functor (presheaf) $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{Alg}(T)$ (apart from logical conditions to express quantifiers and others) where T is a monad on $\mathbf{Set}$, and $\mathbf{Alg}(T)$ is its Eilenberg-Moore algebras; note that monads on $\mathbf{Set}$ are equivalent to (possibly infinitary) varieties in terms of universal algebra (see, e.g., Adámek et al. [1]). The intuitive meaning of the base category $\mathbf{C}$ is the category of types (aka. sorts) or domains of discourse, and then $P(C)$ is the algebra of predicates on a type C. If a propositional logic L is sound and complete wrt. a variety $\mathbf{Alg}(T)$, then the corresponding fibred algebras $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{Alg}(T)$ yield sound and complete semantics for the predicate logic that extends L. This may be called the thesis of completeness lifting: the completeness of propositional logic wrt. $\mathbf{Alg}(T)$ lifts to the completeness of predicate logic wrt. $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{Alg}(T)$.

The present paper is meant to demonstrate the thesis in the fairly general context of substructural logics over FL, hopefully bridging between algebraic

[1] Toposes amount to higher-order hyperdoctrines via the two functors of taking subobject hyperdoctrines and of the tripos-to-topos construction (see, e.g., Frey [4]).

logic, in which logics over FL have been studied, and categorical logic, in which hyperdoctrines have been pursued. Although the two disciplines are currently separated to the author's eyes, nevertheless, Lawvere's original ideas on categorical logic are of algebraic nature (especially, his functorial semantics directly targets universal algebra), and it would be fruitful to restore lost interactions between them. Along this line of thought, in subsequent work, the author plans to develop "Categorical Universal Logic" qua theory of fibred universal algebras or monad-relativised hyperdoctrines. The paper takes a first step towards it.

Main technical contributions are three-fold as follows: (i) generic completeness wrt. Full Lambek hyperdoctrines, which are hyperdoctrines for logics over FL; (ii) Tarskian completeness wrt. **Set**-based models, which arise from a dual adjunction, with more specialised completeness results; (iii) hyperdoctrinal formulations of Girard's translation and Gödel's translation. The generic and set-theoretical completeness results are established for any axiomatic extension of FL, therefore covering a great majority of standard logical systems (classical, intuitionistic, linear, fuzzy, relevant, and so on). In passing, we briefly discuss the tripos-to-topos construction in the present context, which is originally due to Hyland-Johnstone-Pitts [7]. Our uniform categorical semantics for various logical systems enables us to compare different categorical logics on the one setting, and (iii) indeed embodies such a comparison in terms of logical translation. The paper proceeds in the same order as above, after an introduction to typed FL.

2 Typed Full Lambek Calculus

In this section, we define a typed (or many-sorted) version of quantified FL as in Ono [15], which shall be called TFL^q ("T" means "typed"; "q" means "quantified"). In particular, TFL^q follows the typing style of Pitts [16].

Standard categorical logic discusses a typed version of intuitionistic (or coherent or regular) logic, as observed in Pitts [16], Lambek-Scott [11], Jacobs [8], and Johnstone [10]. Typed logic is more natural than single-sorted one from a categorical point of view, and is more expressive in general, since it can encompass various type constructors. If one prefers single-sorted logic to typed logic, the latter can be reduced to the former by allowing for one type (or sort) only.

To put it differently, typed logic is the combination of logic and type theory, and has not only a logic structure but also a type structure, and the latter itself has a rich structure as well as the former. For this reason, syntactic hyperdoctrines constructed from typed systems of logic (which are discussed in relation to completeness in the next section) are amalgamations of syntactic categories obtained from their type theories on the one hand, and Lindenbaum-Tarski algebras obtained from their logic parts on the other; in a nutshell, syntactic hyperdoctrines are type-fibred Lindenbaum-Tarski algebras.

Another merit of typed logic is that the problem of empty domains is resolved because it allows us to have explicit control on type contexts. This was discovered by Joyal, and shall be touched upon later, in more detail.

TFL^q has the following logical connectives:

$$\otimes, \wedge, \vee, \backslash, /, 1, 0, \top, \bot, \forall, \exists.$$

Note that there are two kinds of implication connectives $\backslash$ and $/$, owing to the non-commutative nature of TFL^q.

In TFL^q, every variable x comes with its type σ. That is, TFL^q has basic types, which are denoted by letters like σ, τ, and $x : \sigma$ is a formal expression meaning that a variable x is of type σ. Then, a (type) context is a finite list of type declarations on variables: $x_1 : \sigma_1, ..., x_n : \sigma_n$. A context is often denoted Γ.

Accordingly, TFL^q has typed predicate symbols (aka. predicates in context) and typed function symbols (aka. function symbols in context): $R(x_1, ..., x_n)$ $[x_1 : \sigma_1, ..., x_n : \sigma_n]$ is a formal expression meaning that R is a predicate with n variables $x_1, ..., x_n$ of types $\sigma_1, ..., \sigma_n$ respectively; likewise, $f : \tau$ $[x_1 : \sigma_1, ..., x_n : \sigma_n]$ is a formal expression meaning that f is a function symbol with n variables $x_1, ..., x_n$ of types $\sigma_1, ..., \sigma_n$ and with its values in τ. Then, formulae-in-context φ $[\Gamma]$ and terms-in-context $t : \tau$ $[\Gamma]$ are defined in the usual, inductive way. Our terminology is basically following Pitts [16].

In the present paper, we do not consider any specific type constructor. Higher-Order Full Lambek Calculus shall be discussed in a subsequent paper, and have products and function spaces as type constructors. In this paper, however, we shall focus upon plainly typed predicate logic with no complicated type structure; still, products (not as types but as categorical structures) shall be used in categorical semantics in the next section, to the end of interpreting predicate and function symbols (of arity greater than one).

TFL^q thus has both a type structure and a logic structure, dealing with sequents-in-contexts: $\Phi \vdash \varphi$ $[\Gamma]$ where Γ is a type context, and Φ is a finite list of formulae: $\varphi_1, ..., \varphi_n$. Although it is common to write $\Gamma \mid \Phi \vdash \varphi$ rather than $\Phi \vdash \varphi$ $[\Gamma]$, we employ the latter notation in this paper, following Pitts [16], since TFL^q is an adaptation of Pitts' typed system for intuitionistic logic to the system of the Full Lambek calculus.

The syntax of type contexts Γ in TFL^q is the same as that of typed intuitionistic logic in Pitts [16]; due to space limitations, we do not repeat it here, referring to Pitts [16] for details. Yet we note it is allowed to add a fresh $x : \sigma$ to a context Γ: e.g., $\Phi \vdash \varphi$ $[\Gamma, x : \sigma]$ whenever $\Phi \vdash \varphi$ $[\Gamma]$. On the other hand, it is not permitted to delete redundant variables; the reason becomes clear in later discussion on empty domains. It is allowed to change the order of contexts (e.g., $[\Gamma, \Gamma']$ into $[\Gamma', \Gamma]$). In the below, we focus upon logical rules of inference, which are most relevant part of TFL^q in the paper, being of central importance for us.

TFL^q has no structural rule other than the following cut rule

$$\frac{\Phi_1 \vdash \varphi \ [\Gamma] \quad \Phi_2, \varphi, \Phi_3 \vdash \psi \ [\Gamma]}{\Phi_2, \Phi_1, \Phi_3 \vdash \psi \ [\Gamma]} \ (cut)$$

where ψ may be empty; this is allowed in the following L (left) rules as well. As usual, we have the rule of identity

$$\frac{}{\varphi \vdash \varphi \ [\Gamma]} \ (id)$$

In the following, we list the rules of inference for the logical connectives of TFL^q. There are two kinds of conjunction in TFL^q: multiplicative or monoidal $\otimes$ and additive or cartesian $\wedge$:

$$\frac{\Phi, \varphi, \psi, \Psi \vdash \chi \ [\Gamma]}{\Phi, \varphi \otimes \psi, \Psi \vdash \chi \ [\Gamma]} \ (\otimes L) \qquad \frac{\Phi \vdash \varphi \ [\Gamma] \quad \Psi \vdash \psi \ [\Gamma]}{\Phi, \Psi \vdash \varphi \otimes \psi \ [\Gamma]} \ (\otimes R)$$

$$\frac{\Phi, \varphi, \Psi \vdash \chi \ [\Gamma]}{\Phi, \varphi \wedge \psi, \Psi \vdash \chi \ [\Gamma]} \ (\wedge L_1) \qquad \frac{\Phi, \varphi, \Psi \vdash \chi \ [\Gamma]}{\Phi, \psi \wedge \varphi, \Psi \vdash \chi \ [\Gamma]} \ (\wedge L_2)$$

$$\frac{\Phi \vdash \varphi \ [\Gamma] \quad \Phi \vdash \psi \ [\Gamma]}{\Phi \vdash \varphi \wedge \psi \ [\Gamma]} \ (\wedge R)$$

There is only one disjunction in TFL^q, which is additive, since TFL^q is intuitionistic in the sense that only one formula is allowed to appear on the right-hand side of sequents. Nevertheles, we can treat classical logic as an axiomatic extension of TFL^q, by adding to TFL^q exchange, weakening, contraction, and the excluded middle; note that structural rules can be expressed as axioms.

$$\frac{\Phi, \varphi, \Psi \vdash \chi \ [\Gamma] \quad \Phi, \psi, \Psi \vdash \chi \ [\Gamma]}{\Phi, \varphi \vee \psi, \Psi \vdash \chi \ [\Gamma]} \ (\vee L)$$

$$\frac{\Phi \vdash \varphi \ [\Gamma]}{\Phi \vdash \varphi \vee \psi \ [\Gamma]} \ (\vee R_1) \qquad \frac{\Phi \vdash \varphi \ [\Gamma]}{\Phi \vdash \psi \vee \varphi \ [\Gamma]} \ (\vee R_2)$$

Due to non-commutativity, there are two kinds of implication in TFL^q, $\backslash$ and $/$, which are a right adjoint of $\varphi \otimes$ (-) and a right adjoint of (-) $\otimes \psi$ respectively.

$$\frac{\Phi \vdash \varphi \ [\Gamma] \quad \Psi_1, \psi, \Psi_2 \vdash \chi \ [\Gamma]}{\Psi_1, \Phi, \varphi \backslash \psi, \Psi_2 \vdash \chi \ [\Gamma]} \ (\backslash L) \qquad \frac{\varphi, \Phi \vdash \psi \ [\Gamma]}{\Phi \vdash \varphi \backslash \psi \ [\Gamma]} \ (\backslash R)$$

$$\frac{\Phi \vdash \varphi \ [\Gamma] \quad \Psi_1, \psi, \Psi_2 \vdash \chi \ [\Gamma]}{\Psi_1, \psi / \varphi, \Phi, \Psi_2 \vdash \chi \ [\Gamma]} \ (/ L) \qquad \frac{\Phi, \varphi \vdash \psi \ [\Gamma]}{\Phi \vdash \psi / \varphi \ [\Gamma]} \ (/ R)$$

There are two kinds of truth and falsity constants, monoidal and cartesian ones.

$$\frac{\Psi_1, \Psi_2 \vdash \varphi \ [\Gamma]}{\Psi_1, 1, \Psi_2 \vdash \varphi \ [\Gamma]} \ (1L) \qquad \frac{}{\vdash 1 \ [\Gamma]} \ (1R)$$

$$\frac{}{0 \vdash \ [\Gamma]} \ (0L) \qquad \frac{\Phi \vdash \ [\Gamma]}{\Phi \vdash 0 \ [\Gamma]} \ (0R)$$

$$\frac{}{\Phi \vdash \top \ [\Gamma]} \ (\top R) \qquad \frac{}{\Phi_1, \bot, \Phi_2 \vdash \varphi \ [\Gamma]} \ (\bot L)$$

Finally, we have the following rules for quantifiers $\forall$ and $\exists$, in which type contexts explicitly change; notice that type contexts do not change in the rest of the rules presented above.

$$\frac{\Phi_1, \varphi, \Phi_2 \vdash \psi \ [x : \sigma, \Gamma]}{\Phi_1, \forall x \varphi, \Phi_2 \vdash \psi \ [x : \sigma, \Gamma]} \ (\forall L) \qquad \frac{\Phi \vdash \varphi \ [x : \sigma, \Gamma]}{\Phi \vdash \forall x \varphi \ [\Gamma]} \ (\forall R)$$

$$\frac{\Phi_1, \varphi, \Phi_2 \vdash \psi \ [x : \sigma, \Gamma]}{\Phi_1, \exists x\varphi, \Phi_2 \vdash \psi \ [\Gamma]} \ (\exists L) \qquad \frac{\Phi \vdash \varphi \ [x : \sigma, \Gamma]}{\Phi \vdash \exists x\varphi \ [x : \sigma, \Gamma]} \ (\exists R)$$

As usual, there are eigenvariable conditions on the rules above: x does not appear as a free variable in the bottom sequent of Rule $\forall R$; likewise, x does not appear as a free variable in the bottom sequent of Rule $\exists L$. The other two rules do not have eigenvariable conditions, and this is why contexts do not change in them.

The deducibility of sequents-in-context in TFL^q is defined in the usual way. In this paper, we denote by FL the propositional (and hence no contextual) part of TFL^q. Note that what is called FL in the literature often lacks $\bot$ and $\top$.

As is well known, the following propositional (resp. predicate) logics can be represented as axiomatic (to be precise, axiom-schematic) extensions of FL (resp. TFL^q): classical logic, intuitionistic logic, linear logic (without exponentials), relevance logics, fuzzy logics such as Gödel-Dummett logic (see, e.g., Galatos et al. [3]). Given a set of axioms (to be precise, axiom schemata), say X, we denote by FL_X (resp. TFL^q_X) the corresponding extension of FL (resp. TFL^q) via X.

Lemma 1. *The following sequents-in-context are deducible in* TFL^q*:*

- $\varphi \otimes (\exists x\psi) \vdash \exists x(\varphi \otimes \psi) \ [\Gamma]$ *and* $\exists x(\varphi \otimes \psi) \vdash \varphi \otimes (\exists x\psi) \ [\Gamma]$.
- $(\exists x\psi) \otimes \varphi \vdash \exists x(\psi \otimes \varphi) \ [\Gamma]$ *and* $\exists x(\psi \otimes \varphi) \vdash (\exists x\psi) \otimes \varphi \ [\Gamma]$.

where it is supposed that φ *does not contain* x *as a free variable, and* Γ *contains type declarations on those free variables that appear in* φ *and* $\exists x\psi$.

A striking feature of typed predicate logic is that domains of discourse in semantics can be empty; they are assumed to be non-empty in the usual Tarski semantics of predicate logic. This means that a type σ can be interpreted as an initial object in a category. We therefore need no ad hoc condition on domains of discourse if we work with typed predicate logic. This resolution of the problem of empty domains is due to Joyal as noted in Marquis and Reyes [13].

A proof-theoretic manifestation of this feature is that the following sequent-in-context is not necessarily deducible in TFL^q: $\forall x\varphi \vdash \exists x\varphi \ [\ ]$ where the context is empty. Nonetheless, the following is deducible in TFL^q: $\forall x\varphi \vdash \exists x\varphi \ [x : \sigma, \Gamma]$ where Γ is an appropriate context including type declarations on free variables in φ. This means that we can prove the sequent above when a type σ is inhabited. Here, it is crucial that it is not allowed to delete redundant free variables (e.g., $[x : \sigma, \Gamma]$ cannot be reduced into $[\Gamma]$ even if x does not appear as a free variable in formulae involved); however, it is allowed to add fresh free variables to a context.

3 Full Lambek Hyperdoctrine

It is well known that FL algebras (defined below) provide sound and complete semantics for propositional logic FL (see, e.g., Galatos et al. [3]). In this section we show that fibred FL algebras, or FL hyperdoctrines (defined below), yield sound and complete semantics for typed (or many-sorted) predicate logic TFL^q.

We again emphasise the simple, algebro-logical idea that single algebras (symbolically, A with no indexing) correspond to propositional logic, and fibred algebras (symbolically, $(A_C)_{C\in\mathbf{C}}$ indexed by a category $\mathbf{C}$) correspond to predicate logic. As universal algebra gives foundations for algebraic propositional logic, so fibred universal algebra lays foundations for algebraic predicate logic.

Definition 2 ([3]). $(A, \otimes, \wedge, \vee, \backslash, /, 1, 0, \top, \bot)$ *is called an* FL *algebra iff*

- $(A, \otimes, 1)$ *is a monoid;* 0 *is a (distinguished) element of* A*;*
- $(A, \wedge, \vee, \top, \bot)$ *is a bounded lattice, which induces a partial order* $\leq$ *on* A*;*
- *for any* $a \in A$, $a\backslash(\text{-}) : A \to A$ *is a right adjoint of* $a \otimes (\text{-}) : A \to A$*: i.e.,* $a \otimes b \leq c$ *iff* $b \leq a\backslash c$ *for any* $a, b, c \in A$*;*
- *for any* $b \in A$, $(\text{-})/b : A \to A$ *is a right adjoint of* $(\text{-}) \otimes b : A \to A$*: i.e.,* $a \otimes b \leq c$ *iff* $a \leq c/b$ *for any* $a, b, c \in A$*.*

A homomorphism of FL *algebras is defined as a map preserving all the operations* $(\otimes, \wedge, \vee, \backslash, /, 1, 0, \top, \bot)$. **FL** *denotes the category of* FL *algebras and their homomorphisms.*

Although 0 is just a neutral element of A with no axiom, the rules for 0 are automatically valid by the definition of interpretations defined below.

FL is an algebraic category (i.e., a category monadic over **Set**), or a variety in terms of universal algebra, since the two adjointness conditions can be rephrased by equations (see, e.g., Galatos et al. [3]). An axiomatic extension FL_X of FL corresponds to an algebraic full subcategory (or sub-variety) of **FL**, denoted $\mathbf{FL}_X$ (algebraicity follows from definability by axioms); this is the well-known, logic-variety correspondence for logics over FL (see Galatos et al. [3]).

Definition 3. *An* FL *(Full Lambek) hyperdoctrine is an* $\mathbf{FL}$*-valued presheaf* $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}$ *such that* $\mathbf{C}$ *is a category with finite products, and the following conditions on quantifiers hold:*

- *For any projection* $\pi : X \times Y \to Y$ *in* $\mathbf{C}$, $P(\pi) : P(Y) \to P(X \times Y)$ *has a right adjoint, denoted*

$$\forall_\pi : P(X \times Y) \to P(Y).$$

And the corresponding Beck-Chevalley condition holds, i.e., the following diagram commutes for any arrow $f : Z \to Y$ *in* $\mathbf{C}$ *(*$\pi' : X \times Z \to Z$ *below denotes a projection):*

$$\begin{array}{ccc} P(X \times Y) & \xrightarrow{\forall_\pi} & P(Y) \\ {\scriptstyle P(X\times f)}\downarrow & & \downarrow{\scriptstyle P(f)} \\ P(X \times Z) & \xrightarrow[\forall_{\pi'}]{} & P(Z) \end{array}$$

– *For any projection* $\pi : X \times Y \to Y$ *in* $\mathbf{C}$, $P(\pi) : P(Y) \to P(X \times Y)$ *has a left adjoint, denoted*

$$\exists_\pi : P(X \times Y) \to P(Y).$$

The corresponding Beck-Chevalley condition holds:

$$\begin{array}{ccc} P(X \times Y) & \xrightarrow{\exists_\pi} & P(Y) \\ {\scriptstyle P(X \times f)}\downarrow & & \downarrow{\scriptstyle P(f)} \\ P(X \times Z) & \xrightarrow[\exists_{\pi'}]{} & P(Z) \end{array}$$

Furthermore, the Frobenius Reciprocity conditions hold: for any projection $\pi : X \times Y \to Y$ *in* $\mathbf{C}$, *any* $a \in P(Y)$, *and any* $b \in P(X \times Y)$,

$$a \otimes (\exists_\pi b) = \exists_\pi (P(\pi)(a) \otimes b)$$
$$(\exists_\pi b) \otimes a = \exists_\pi (b \otimes P(\pi)(a)).$$

For an axiomatic extension FL_X *of* FL, *an* FL_X *hyperdoctrine is defined by restricting the value category* $\mathbf{FL}$ *into* $\mathbf{FL}_X$. *An* FL *(resp.* FL_X*) hyperdoctrine is also called a fibred* FL *(resp.* FL_X*) algebra.*

The category $\mathbf{C}$ of an FL hyperdoctrine $P : \mathbf{C} \to \mathbf{FL}$ is called its base category or type category, and P is also called its predicate functor; intuitively, $P(C)$ is the algebra of predicates on a type, or domain of discourse, C.

Note that, in the definition above, we need two Frobenius Reciprocity conditions due to the non-commutativity of FL algebras.

An FL hyperdoctrine may be seen as an indexed category, and so as a fibration via the Grothendieck construction. Although we discuss in terms of indexed categories in this paper, we can do the job in terms of fibrations as well. In the view of fibrations, each $P(C)$ is called a fibre of an FL hyperdoctrine P.

The FL (resp. FL_X) hyperdoctrine semantics for TFL^q (resp. TFL^q_X) is defined as follows.

Definition 4. *Fix an* FL *hyperdoctrine* $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}$. *An interpretation* $[\![\text{-}]\!]$ *of* TFL^q *in the* FL *hyperdoctrine* P *consists of the following:*

– *assignment of an object* $[\![\sigma]\!]$ *in* $\mathbf{C}$ *to each basic type* σ *in* TFL^q*;*
– *assignment of an arrow* $[\![f : \tau\ [\Gamma]]\!] : [\![\sigma_1]\!] \times ... \times [\![\sigma_n]\!] \to [\![\sigma]\!]$ *in* $\mathbf{C}$ *to each typed function symbol* $f : \tau\ [\Gamma]$ *in* TFL^q *where* Γ *is supposed to be* $x_1 : \sigma_1, ..., x_n : \sigma_n$ *(note that* $[\![\sigma_1]\!] \times ... \times [\![\sigma_n]\!]$ *makes sense because* $\mathbf{C}$ *has finite products);*
– *assignment of an element* $[\![R\ [\Gamma]]\!]$ *in* $P([\![\Gamma]\!])$, *which is an* FL *algebra, to each typed predicate symbol* $R\ [\Gamma]$ *in* TFL^q*; if the context* Γ *is* $x_1 : \sigma_1, ..., x_n : \sigma_n$, *then* $[\![\Gamma]\!]$ *denotes* $[\![\sigma_1]\!] \times ... \times [\![\sigma_n]\!]$.

Then, terms are inductively interpreted in the following way:

– $[\![x : \sigma\ [\Gamma_1, x : \sigma, \Gamma_2]]\!]$ *is defined as the following projection in* $\mathbf{C}$*:*

$$\pi : [\![\Gamma_1]\!] \times [\![\sigma]\!] \times [\![\Gamma_2]\!] \to [\![\sigma]\!].$$

– $[\![f(t_1, ..., t_n) : \tau\ [\Gamma]]\!]$ *is defined as:*

$$[\![f]\!] \circ \langle [\![t_1 : \sigma_1\ [\Gamma]]\!], ..., [\![t_n : \sigma_n\ [\Gamma]]\!] \rangle$$

where it is supposed that $f : \tau\ [x_1 : \sigma_1, ..., x_n : \sigma_n]$*, and* $t_1 : \sigma_1\ [\Gamma], ..., t_n : \sigma_n\ [\Gamma]$*. Note that* $\langle [\![t_1 : \sigma_1\ [\Gamma]]\!], ..., [\![t_n : \sigma_n\ [\Gamma]]\!] \rangle$ *above is the product (or pairing) of arrows in* $\mathbf{C}$*.*

Formuli are then interpreted inductively in the following manner:

– $[\![R(t_1, ..., t_n)\ [\Gamma]]\!]$ *is defined as*

$$P(\langle [\![t_1 : \sigma_1[\Gamma]]\!], ..., [\![t_n : \sigma_n[\Gamma]]\!] \rangle)([\![R\ [x : \sigma_1, ..., x_n : \sigma_n]]\!])$$

where R *is a predicate symbol in context* $x_1 : \sigma_1, ..., x_n : \sigma_n$*.*

– $[\![\varphi \otimes \psi\ [\Gamma]]\!]$ *is defined as* $[\![\varphi\ [\Gamma]]\!] \otimes [\![\psi\ [\Gamma]]\!]$*. The other binary connectives* $\wedge, \vee, \backslash, /$ *are interpreted in the same way.* $[\![1\ [\Gamma]]\!]$ *is defined as the monoidal unit of* $P([\![\Gamma]\!])$*. The other constants* $0, \top, \bot$ *are interpreted in the same way.*

– $[\![\forall x\varphi\ [\Gamma]]\!]$ *is defined as*

$$\forall_\pi([\![\varphi\ [x : \sigma, \Gamma]]\!])$$

where $\pi : [\![\sigma]\!] \times [\![\Gamma]\!] \to [\![\Gamma]\!]$ *is a projection in* $\mathbf{C}$*, and* φ *is a formula in context* $[x : \sigma, \Gamma]$*. Similarly,* $[\![\exists x\varphi\ [\Gamma]]\!]$ *is defined as*

$$\exists_\pi([\![\varphi\ [x : \sigma, \Gamma]]\!]).$$

Finally, satisfaction of sequents is defined:

– $\varphi_1, ..., \varphi_n \vdash \psi\ [\Gamma]$ *is satisfied in an interpretation* $[\![\text{-}]\!]$ *in an* FL *hyperdoctrine* P *iff the following holds in* $P([\![\Gamma]\!])$*:*

$$[\![\varphi_1\ [\Gamma]]\!] \otimes ... \otimes [\![\varphi_n\ [\Gamma]]\!] \leq [\![\psi\ [\Gamma]]\!].$$

In case the right-hand side of a sequent is empty, $\varphi_1, ..., \varphi_n \vdash [\Gamma]$ *is satisfied in* $[\![\text{-}]\!]$ *iff* $[\![\varphi_1\ [\Gamma]]\!] \otimes ... \otimes [\![\varphi_n\ [\Gamma]]\!] \leq 0$ *in* $P([\![\Gamma]\!])$*. In case the left-hand side of a sequent is empty,* $\vdash \varphi\ [\Gamma]$ *is satisfied in* $[\![\text{-}]\!]$ *iff* $1 \leq [\![\varphi[\Gamma]]\!]$ *in* $P([\![\Gamma]\!])$*.*

An interpretation of TFL^q_X *in an* FL_X *hyperdoctrine is defined by replacing* **FL** *and* TFL^q *above with* $\mathbf{FL}_X$ *and* TFL^q_X *respectively.*

In the following, we show that the FL (resp. FL_X) hyperdoctrine semantics is sound and complete for TFL^q (resp. TFL^q_X). Let $[\![\Phi\ [\Gamma]]\!]$ denote $[\![\varphi_1\ [\Gamma]]\!] \otimes ... \otimes [\![\varphi_n\ [\Gamma]]\!]$ if Φ is $\varphi_1, ..., \varphi_n$.

Intuitively, an arrow f in $\mathbf{C}$ is a term, and $P(f)$ is a substitution operation (this is exactly true in syntactic hyperdoctrines defined later); then, the Beck-Chevalley conditions and the functoriality of P tell us that substitution commutes with all the logical operations (namely, both propositional connectives and quantifiers). From such a logical point of view, the meaning of the Beck-Chevalley conditions is crystal clear; they just say that substitution after quantification is the same as quantification after substitution.

Proposition 5. *If $\Phi \vdash \psi\ [\Gamma]$ is deducible in* TFL^q *(resp.* TFL^q_X*), then it is satisfied in any interpretation in any* FL *(resp.* FL_X*) hyperdoctrine.*

Proof. Fix an FL or FL_X hyperdoctrine P and an interpretation $[\![\text{-}]\!]$ in P. Initial sequents in context are satisfied because $a \leq a$ in any fibre $P(C)$. The cut rule preserves satisfaction, since tensoring preserves $\leq$ and $\leq$ has transitivity. It is easy to verify that all the rules for the logical connectives preserve satisfaction.

Let us consider universal quantifier $\forall$. To show the case of Rule $\forall R$, assume that $[\![\Phi\ [x : \sigma, \Gamma]]\!] \leq [\![\varphi\ [x : \sigma, \Gamma]]\!]$ in $P([\![\sigma]\!] \times [\![\Gamma]\!])$. It then follows that $[\![\Phi\ [x : \sigma, \Gamma]]\!] = P(\pi : [\![\sigma]\!] \times [\![\Gamma]\!] \to [\![\Gamma]\!])([\![\Phi\ [\Gamma]]\!])$ where π is a projection in $\mathbf{C}$, and note that Φ does not include x among its free variables by the eigenvariable condition. We thus have $P(\pi)([\![\Phi\ [\Gamma]]\!]) \leq [\![\varphi\ [x : \sigma, \Gamma]]\!]$. Since $\forall_\pi : P([\![\sigma]\!] \times [\![\Gamma]\!]) \to P([\![\Gamma]\!])$ is a right adjoint of $P(\pi)$, it follows that $[\![\Phi\ [\Gamma]]\!] \leq \forall_\pi([\![\varphi\ [x : \sigma, \Gamma]]\!]) = [\![\forall x\varphi\ [\Gamma]]\!]$. We next show the case of $\forall L$. Assume that $[\![\Phi_1\ [x : \sigma, \Gamma]]\!] \otimes [\![\varphi\ [x : \sigma, \Gamma]]\!] \otimes [\![\Phi_2\ [x : \sigma, \Gamma]]\!] \leq [\![\psi\ [x : \sigma, \Gamma]]\!]$. The adjunction condition for universal quantifier gives us $P(\pi)(\forall_\pi([\![\varphi\ [x : \sigma, \Gamma]]\!])) \leq [\![\varphi\ [x : \sigma, \Gamma]]\!]$ where $\pi : [\![\sigma]\!] \times [\![\Gamma]\!] \to [\![\Gamma]\!]$ is a projection. Yet we also have $P(\pi)(\forall_\pi([\![\varphi\ [x : \sigma, \Gamma]]\!])) = P(\pi)([\![\forall x\varphi\ [\Gamma]]\!]) = [\![\forall x\varphi\ [x : \sigma, \Gamma]]\!]$. Since tensoring respects $\leq$, these together imply that $[\![\Phi_1\ [x : \sigma, \Gamma]]\!] \otimes [\![\forall x\varphi\ [x : \sigma, \Gamma]]\!] \otimes [\![\Phi_2\ [x : \sigma, \Gamma]]\!] \leq [\![\psi\ [x : \sigma, \Gamma]]\!]$.

It remains to show the case of existential quantifier $\exists$. In order to prove that Rule $\exists L$ preserves satisfaction, assume that $[\![\Phi_1\ [x : \sigma, \Gamma]]\!] \otimes [\![\varphi\ [x : \sigma, \Gamma]]\!] \otimes [\![\Phi_2\ [x : \sigma, \Gamma]]\!] \leq [\![\psi\ [x : \sigma, \Gamma]]\!]$. This is equivalent to the following: $[\![\Phi_1\ [x : \sigma, \Gamma]]\!] \otimes [\![\varphi\ [x : \sigma, \Gamma]]\!] \otimes [\![\Phi_2\ [x : \sigma, \Gamma]]\!] \leq P(\pi)([\![\psi\ [\Gamma]]\!])$ where $\pi : [\![\sigma]\!] \times [\![\Gamma]\!] \to [\![\Gamma]\!]$ is a projection. Since $\exists_\pi : P([\![\sigma]\!] \times [\![\Gamma]\!]) \to P([\![\Gamma]\!])$ is left adjoint to $P(\pi)$, it follows that $\exists_\pi([\![\Phi_1\ [x : \sigma, \Gamma]]\!] \otimes [\![\varphi\ [x : \sigma, \Gamma]]\!] \otimes [\![\Phi_2\ [x : \sigma, \Gamma]]\!]) \leq [\![\psi\ [\Gamma]]\!]$. This is equivalent to the following: $\exists_\pi(\ P(\pi)([\![\Phi_1\ [\Gamma]]\!]) \otimes [\![\varphi\ [x : \sigma, \Gamma]]\!] \otimes P(\pi)([\![\Phi_2\ [\Gamma]]\!])\ ) \leq [\![\psi\ [\Gamma]]\!]$. Repeated applications of the two Frobenius Reciprocity conditions give us $[\![\Phi_1\ [\Gamma]]\!] \otimes \exists_\pi([\![\varphi\ [x : \sigma, \Gamma]]\!]) \otimes [\![\Phi_2\ [\Gamma]]\!] \leq [\![\psi\ [\Gamma]]\!]$. Then we finally have the following: $[\![\Phi_1\ [\Gamma]]\!] \otimes [\![\exists x\varphi\ [\Gamma]]\!] \otimes [\![\Phi_2\ [\Gamma]]\!] \leq [\![\psi\ [\Gamma]]\!]$. To show the case of $\exists R$, assume that $[\![\Phi\ [x : \sigma, \Gamma]]\!] \leq [\![\varphi\ [x : \sigma, \Gamma]]\!]$. The adjunction condition for existential quantifier tells us that $[\![\varphi\ [x : \sigma, \Gamma]]\!] \leq P(\pi)(\exists_\pi([\![\varphi\ [x : \sigma, \Gamma]]\!]))$ where $\pi : [\![\sigma]\!] \times [\![\Gamma]\!] \to [\![\Gamma]\!]$ is a projection. We thus have the following: $[\![\Phi\ [x : \sigma, \Gamma]]\!] \leq P(\pi)(\exists_\pi([\![\varphi\ [x : \sigma, \Gamma]]\!])) = [\![\exists x\varphi\ [x : \sigma, \Gamma]]\!]$. This completes the proof.

Syntactic hyperdoctrines are then defined as follows towards the goal of proving completeness. They are the categorificaton of Lindenbaum-Tarski algebras.

Definition 6. *The syntactic hyperdoctrine of* TFL^q *is defined as follows; that of* TFL^q_X *is defined by replacing* **FL** *and* TFL^q *below with* $\mathbf{FL}_X$ *and* TFL^q_X*.*

We first define the base category $\mathbf{C}$*. An object in* $\mathbf{C}$ *is a context* Γ *up to* α*-equivalence (i.e., the naming of variables does not matter). An arrow in* $\mathbf{C}$ *from an object* Γ *to another* Γ' *is a list of terms* $[t_1, ..., t_n]$ *(up to equivalence) such that* $t_1 : \sigma_1\ [\Gamma], ..., t_n : \sigma_n\ [\Gamma]$ *where* Γ' *is supposed to be* $x_1 : \sigma_1, ..., x_n : \sigma_n$*.*

The syntactic hyperdoctrine $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}$ *is then defined in the following way. For an object* Γ *in* $\mathbf{C}$*, let* $\mathrm{Form}_\Gamma = \{\varphi \mid \varphi \text{ is a formula in context } \Gamma\}$*. Define an equivalence relation* $\sim$ *on* Form_Γ *as follows: for* $\varphi, \psi \in \mathrm{Form}_\Gamma$*,* $\varphi \sim \psi$ *iff both* $\varphi \vdash \psi\ [\Gamma]$ *and* $\psi \vdash \varphi\ [\Gamma]$ *are deducible in* TFL^q*. We then define*

$$P(\Gamma) = \mathrm{Form}_\Gamma / \sim$$

with an FL *algebra structure induced by the logical connectives.*

The arrow part of P is defined as follows. Let $[t_1, ..., t_n] : \Gamma \to \Gamma'$ be an arrow in $\mathbf{C}$ where Γ' is $x_1 : \sigma_1, ..., x_n : \sigma_n$. Then we define $P([t_1, ..., t_n]) : P(\Gamma') \to P(\Gamma)$ by

$$P([t_1, ..., t_n])(\varphi) = \varphi[t_1/x_1, ..., t_n/x_n]$$

where it is supposed that $t_1 : \sigma_1\ [\Gamma], ..., t_n : \sigma_n\ [\Gamma]$, and that φ is a formula in context $x_1 : \sigma_1, ..., x_n : \sigma_n$.

Intuitively, $P(\Gamma)$ above is a Lindenbaum-Tarski algebra sliced with respect to each Γ. It is straightforward to verify that the operations of $P(\Gamma)$ above are well defined, and $P(\Gamma)$ forms an FL algebra. We still have to check that P defined above is a hyperdoctrine; this is done in the following lemma.

Lemma 7. *The syntactic hyperdoctrine $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}$ (resp. $\mathbf{FL}_X$) is an* FL *(resp.* FL_X*) hyperdoctrine. In particular, it has quantifier structures satisfying the Beck-Chevalley and Frobenius Reciprocity conditions.*

Proof. Since substitution commutes with all the logical connectives, $P([t_1, ..., t_n])$ defined above is always a homomorphism of FL algebras. Thus, P is a contravariant functor.

We have to verify that the base category $\mathbf{C}$ has finite products, or equivalently, binary products. For objects Γ, Γ' in $\mathbf{C}$, we define their product $\Gamma \times \Gamma'$ as follows. Suppose that Γ is $x_1 : \sigma_1, ..., x_n : \sigma_n$, and Γ' is $y_1 : \tau_1, ..., y_m : \tau_m$. Then, $\Gamma \times \Gamma'$ is defined as $x_1 : \sigma_1, ..., x_n : \sigma_n, y_1 : \tau_1, ..., y_m : \tau_m$. An associated projection $\pi : \Gamma \times \Gamma' \to \Gamma'$ is defined as $[y_1, ..., y_m] : \Gamma \times \Gamma' \to \Gamma'$ where the context of each y_i is taken to be $x_1 : \sigma_1, ..., x_n : \sigma_n, y_1 : \tau_1, ..., y_m : \tau_m$ (rather than $y_1 : \tau_1, ..., y_m : \tau_m$). The other projection is defined in a similar way. It is easily verified that these indeed form a categorical product in $\mathbf{C}$.

In order to show that P has quantifier structures, let $\pi : \Gamma \times \Gamma' \to \Gamma'$ denote the projection in $\mathbf{C}$ defined above, and then consider $P(\pi)$, which we have to show has right and left adjoints. The right and left adjoints of $P(\pi)$ can be constructed as follows. Recall Γ is $x : \sigma_1, ..., x_n : \sigma_n$. Let $\varphi \in P(\Gamma \times \Gamma')$; here we are identifying φ with the equivalence class to which φ belongs, since every argument below respects the equivalence. Then define $\forall_\pi : P(\Gamma \times \Gamma') \to P(\Gamma')$ by $\forall_\pi(\varphi) = \forall x_1 ... \forall x_n \varphi$ where the formula on the right-hand side actually denotes the corresponding equivalence class. Similarly, we define $\exists_\pi : P(\Gamma \times \Gamma') \to P(\Gamma')$ by $\exists_\pi(\varphi) = \exists x_1 ... \exists x_n \varphi$. Let us show that $\forall_\pi$ is the right adjoint of $P(\pi)$. We first assume $P(\pi)(\psi) \leq \varphi$ in $P(\Gamma \times \Gamma')$ for $\psi \in P(\Gamma')$ and $\varphi \in P(\Gamma \times \Gamma')$. Then it follows from the definition of P and π that $P(\pi)(\psi\ [\Gamma]) = \psi\ [\Gamma, \Gamma']$ where we are making explicit the two different contexts of ψ; the role of $P(\pi)$ just lies in changing contexts. Since the $\leq$ of $P(\Gamma \times \Gamma')$ is induced by its lattice structure, we have $\varphi \wedge \psi = \psi$. It follows from the definition of $P(\Gamma \times \Gamma')$ that $\varphi \wedge \psi \vdash \psi\ [\Gamma, \Gamma']$ and $\psi \vdash \varphi \wedge \psi\ [\Gamma, \Gamma']$ are deducible in TFL^q (resp. TFL^q_X), whence $\psi \vdash \varphi\ [\Gamma, \Gamma']$ is deducible as well. By repeated applications of rule

$\forall R$, it follows that $\psi \vdash \forall x_1 ... \forall x_n \varphi \ [\Gamma']$ is deducible. This implies that both $\psi \vdash \psi \wedge \forall x_1 ... \forall x_n \varphi \ [\Gamma']$ and $\psi \wedge \forall x_1 ... \forall x_n \varphi \vdash \psi \ [\Gamma']$ are deducible, whence $\psi \leq \forall x_1 ... \forall x_n \varphi$ in $P(\Gamma')$.

We show the converse. Assume that $\psi \leq \forall x_1 ... \forall x_n \varphi$ in $P(\Gamma')$. By arguing as in the above, $\psi \vdash \forall x_1 ... \forall x_n \varphi \ [\Gamma']$ is deducible. By enriching the context, $\psi \vdash \forall x_1 ... \forall x_n \varphi \ [\Gamma, \Gamma']$ is deducible. Since $\forall x_1 ... \forall x_n \varphi \vdash \varphi \ [\Gamma, \Gamma']$ is deducible by rule $\forall L$, the cut rule tells us that $\psi \vdash \varphi \ [\Gamma, \Gamma']$ is deducible; note that the contexts of two sequents-in-context must be the same when applying the cut rule to them. It finally follows that $P(\pi)(\psi) \leq \varphi$ in $P(\Gamma \times \Gamma')$. Thus, $\forall_\pi$ is the right adjoint of $P(\pi)$. Similarly, $\exists_\pi$ can be shown to be the left adjoint of $P(\pi)$.

The Beck-Chevalley condition for $\forall$ can be verified as follows. Let $\varphi \in P(\Gamma \times \Gamma')$, $\pi : \Gamma \times \Gamma' \to \Gamma'$ a projection in $\mathbf{C}$, and $\pi' : \Gamma \times \Gamma'' \to \Gamma''$ another projection in $\mathbf{C}$ for objects $\Gamma, \Gamma', \Gamma''$ in $\mathbf{C}$. Then, we have $P([t_1, ..., t_n]) \circ \forall_\pi(\varphi) = (\forall x_1 ... \forall x_n \varphi)[t_1/y_1, ..., t_n/y_m]$ where it is supposed that Γ is $x_1 : \sigma_1, ..., x_n : \sigma_n$, Γ' is $y_1 : \tau_1, ..., y_m : \tau_m$, and $t_1 : \tau_1 \ [\Gamma''], ..., t_m : \tau_m \ [\Gamma'']$. We also have the following $\forall_{\pi'} \circ P([t_1, ..., t_n])(\varphi) = \forall x_1 ... \forall x_n (\varphi[t_1/y_1, ..., t_n/y_m])$. The Beck-Chevalley condition for $\forall$ thus follows. The Beck-Chevalley condition for $\exists$ can be verified in a similar way. The two Frobenius Reciprocity conditions for $\exists$ follow immediately from Lemma 1.

The syntactic hyperdoctrine is a free or classifying hyperdoctrine in a suitable sense. It is the combination of the classifying category $\mathbf{C}$ above and the free algebras $P(\Gamma)$ above, which has the universal property inherited from both of them, though we do not have space to work out the details in this paper.

Now, there is the obvious, canonical interpretation of TFL^q (resp. TFL^q_X) in the syntactic hyperdoctrine of TFL^q (resp. TFL^q_X); it is straightforward to see:

Lemma 8. *If $\Phi \vdash \psi \ [\Gamma]$ is satisfied in the canonical interpretation in the syntactic hyperdoctrine of* TFL^q *(resp.* TFL^q_X*), it is deducible in* TFL^q *(resp.* TFL^q_X*).*

The lemmata above give us the completeness result: If $\Phi \vdash \psi \ [\Gamma]$ is satisfied in any interpretation in any FL (resp. FL_X) hyperdoctrine, then it is deducible in TFL^q (resp. TFL^q_X). Combining soundness and completeness, we obtain:

Theorem 9. *$\Phi \vdash \psi \ [\Gamma]$ is deducible in* TFL^q *(resp.* TFL^q_X*) iff it is satisfied in any interpretation in any* FL *(resp.* FL_X*) hyperdoctrine.*

4 Duality-Induced Set-Theoretical Hyperdoctrines

In this section, we discuss hyperdoctrines induced from dual adjunctions between **Set** and **FL**, which are, so to say, many-valued powerset hyperdoctrines, and give many-valued Tarski semantics with soundness and completeness, generalising the powerset hyperdoctrine $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \mathbf{2})$, which is equivalent to Tarski semantics. We mostly omit proofs in this section due to space limitations.

Theorem 10. *Let $\Omega \in \mathbf{FL}$. The following dual adjunction holds between* **Set** *and* **FL***, induced by Ω as a dualising object:*

$$\mathrm{Hom}_{\mathbf{FL}}(\text{-}, \Omega)^{\mathrm{op}} \dashv \mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega) : \mathbf{Set}^{\mathrm{op}} \to \mathbf{FL}.$$

Proposition 11. *Let $\Omega \in \mathbf{FL}$ with Ω complete. Then,* $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega) : \mathbf{Set}^{\mathrm{op}} \to \mathbf{FL}$ *(resp.* $\mathbf{FL}_X$*) is an* FL *(resp.* FL_X*) hyperdoctrine.*

Proof. Let $\pi : X \times Y \to Y$ be a projection in **Set**. We define $\forall_\pi$ and $\exists_\pi$ as follows: given $v \in \mathrm{Hom}(X \times Y, \Omega)$ and $y \in Y$, let $\forall_\pi(v)(y) := \bigwedge\{v(x, y) \mid x \in X\}$ and $\exists_\pi(v)(y) := \bigvee\{v(x, y) \mid x \in X\}$. These yield the required quantifier structures with the Beck-Chevalley and Frobenius Reciprocity conditions; details omitted.

Now, we aim at obtaining completeness with respect to models of form $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)$. The above proof tells us that $\forall$ and $\exists$ in $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)$ are actually meets and joins in Ω. This implies that if Ω is not complete, in general, $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)$ cannot interpret quantifiers. At the same time, however, assuming completeness prevents us from obtaining completeness for any axiomatic extension TFL^q_X of TFL^q; this is why we do not assume it. Such incompleteness phenomena have already been observed (see, e.g., Ono [14]). A standard remedy to this problem is to restrict attention to "safe" interpretations while considering general Ω. In our context, a safe interpretation $[\![\text{-}]\!]$ in $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)$ is such that $[\![\text{-}]\!]$ uses those joins and meets only that exist in Ω, i.e., quantifiers are always interpreted via existing joins and meets only. We then have completeness with respect to the special class of set-theoretical models $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)$.

Theorem 12. *$\Phi \vdash \psi$ $[\Gamma]$ is deducible in* TFL^q_X *iff it is satisfied in any safe interpretation in* $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)$ *for any $\Omega \in \mathbf{FL}$.*

In the special case of TFL^q*, it suffices to consider complete Ω's only: $\Phi \vdash \psi$ $[\Gamma]$ is deducible in* TFL^q *iff it is satisfied in any interpretation in any* FL *hyperdoctrine* $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)$ *with $\Omega \in \mathbf{FL}$ complete.*

Focusing on a more specific context, we can further reduce the class of models $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)$ into a smaller one. In the strongest case of classical logic, it suffices to consider $\{0, 1\}$ only in the place of Ω; this is exactly the Tarski completeness.

For an intermediate case, consider MTL (monoidal t-norm logic; see Hájek et al. [5]), which is FL expanded with exchange, weakening, and the pre-linearity axiom, $(\varphi \to \psi) \vee (\psi \to \varphi)$. The algebras of MTL are denoted by **MTL**. We denote by MTL^q the quantified version with the additional axiom of $\forall$-$\vee$ distributivity, i.e., $\forall x(\varphi \vee \psi) \leftrightarrow \forall x\varphi \vee \psi$ where x does not occur in ψ as a free variable, and by MTL^q_X an axiomatic extension of MTL^q.

Theorem 13. *$\Phi \vdash \psi$ $[\Gamma]$ is deducible in* MTL^q_X *iff it is satisfied in any interpretation in* $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)$ *for any linearly ordered $\Omega \in \mathbf{MTL}_X$.*

We briefly discuss the tripos-topos construction in the present context of FL hyperdoctrines; it is originally due to Hyland-Johnstone-Pitts [7]. To this end, we work in the internal logic of FL hyperdoctrines $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}$: i.e., we have types X and function symbols f corresponding to objects X and arrows f in $\mathbf{C}$ respectively, and also those predicate symbols R on a type $C \in \mathbf{C}$ that correspond to elements $R \in P(C)$.

Definition 14. *Let P be an* FL *hyperdoctrine. We define a category $\mathbf{T}[P]$ as follows. An object of $\mathbf{T}[P]$ is a partial equivalence relation, i.e., a pair (X, E_X) such*

that X is an object in the base category $\mathbf{C}$, and E_X is an element of $P(X\times X)$ and is symmetric and transitive in the internal logic of P: $E_X(x,y) \vdash E_X(y,x)$ $[x,y : X]$ and $E_X(x,y), E_X(y,z) \vdash E_X(x,z)$ $[x,y,z : X]$.

An arrow from (X, E_X) to (Y, E_Y) is $F \in P(X \times Y)$ such that (i) extensionality: $E_X(x_1,x_2), E_Y(y_1,y_2), F(x_1,y_1) \vdash F(x_2,y_2)$ $[x_1,x_2 : X, y_1,y_2 : Y]$; (ii) strictness: $F(x,y) \vdash E_X(x,x) \wedge E_Y(y,y)$ $[x : X, y : Y]$; (iii) single-valuedness: $F(x,y_1), F(x,y_2) \vdash E_Y(y_1,y_2)$ $[x : X, y_1,y_2 : Y]$; (iv) totality: $E_X(x,x) \vdash \exists y\, F(x,y)$ $[x : X]$. Such an F is called a functional relation.

For a complete FL algebra Ω, which is a quantale with additional operations, $\mathbf{T}[\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)]$ may be called the category of Ω-valued sets. Quantale sets in the sense of Höhle et al. [6] are objects in $\mathbf{T}[\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)]$, but not vice versa: our Ω-valued sets are slightly more general than their quantale sets.

Note that if Ω is a locale, $\mathbf{T}[\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)]$ is the Higgs topos of Ω-valued sets, which is in turn equivalent to the category of sheaves on Ω.

5 Girard's and Gödel's Translation Hyperdoctrinally

In this section, we discuss Girard's and Gödel's translation theorems on the hyperdoctrinal setting. The former embeds intuitionistic logic into linear logic via exponential !; the latter embeds classical logic into intuitionistic logic via double negation $\neg\neg$. Since logic is dual to algebraic semantics, we construct intuitionistic (resp. classical) hyperdoctrines from linear (resp. intuitionistic) hyperdoctrines. We omit proofs in this section as well for space limitations.

We first consider Gödel's translation. We think of $\neg\neg$ as a functor $\mathrm{Fix}_{\neg\neg}$ from **HA**, the category of heyting algebras, to **BA**, the category of boolean algebras: i.e., define $\mathrm{Fix}_{\neg\neg}(A) = \{a \in A \mid \neg\neg a = a\}$; the arrow part is defined by restriction. Here, $\mathrm{Fix}_{\neg\neg}(A)$ forms a boolean algebra.

Let us define IL hyperdoctrines as FL hyperdoctrines with values in **HA**. Likewise, CL hyperdoctrines are defined as FL hyperdoctrines with values in **BA**. Note that both kinds of hyperdoctrines are TFL^q_X hyperdoctrines with suitable choices of axioms X. Finally, Gödel's translation theorem can be understood in terms of hyperdoctrines as follows.

Theorem 15. *Let $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{HA}$ be an* IL *hyperdoctrine. Then, the following composed functor $\mathrm{Fix}_{\neg\neg} \circ P : \mathbf{C}^{\mathrm{op}} \to \mathbf{BA}$ forms a* CL *hyperdoctrine.*

This is a first-order and hyperdoctrinal version of the construction of boolean toposes from given toposes via double negation topologies on them.

We can treat Girard's translation along a similar line. An exponential ! on an FL algebra A is defined as a unary operation satisfying: (i) $a \leq b$ implies $!a \leq !b$; (ii) $!!a = !a \leq a$; (iii) $!\top = 1$; (iv) $!a \otimes !b = !(a \wedge b)$ (see Coumans et al. [2]). We denote by $\mathbf{FL}^!_c$ the category of commutative FL algebras with ! and maps preserving both ! and FL algebra operations; they give the algebraic counterpart of intuitionistic linear logic with !, denoted ILL.

We regard exponential ! as a functor $\mathrm{Fix}_!$ from $\mathbf{FL}^!_c$ to **HA**: define $\mathrm{Fix}_!(A) = \{a \in A \mid !a = a\}$; the arrow part is defined by restriction. $\mathrm{Fix}_!(A)$ is the set of

those elements of A that admit structural rules, and forms a heyting algebra. ILL hyperdoctrines are defined as FL hyperdoctrines with values in $\mathbf{FL}_c^!$.

Theorem 16. *Let* $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}_c^!$ *be an* ILL *hyperdoctrine. Then, the following composed functor* $\mathrm{Fix}_! \circ P : \mathbf{C}^{\mathrm{op}} \to \mathbf{HA}$ *forms an* IL *hyperdoctrine.*

The theorem above is slightly more general than Girard's translation theorem, in the sense that the latter corresponds to the case of syntactic hyperdoctrines in the former. Although in this paper we do not explicitly discuss substructural logics enriched with modalities and their hyperdoctrinal semantics, nevertheless, our method perfectly works for them as well, yielding the corresponding soundness and completeness results in terms of hyperdoctrines with values in FL algebras with modalities; Girard's ! is just a special case.

References

1. Adámek, J., Herrlich, H., Strecker, G.E.: Abstract and Concrete Categories. John Wiley and Sons, Inc. (1990)
2. Coumans, D., Gehrke, M., van Rooijen, L.: Relational semantics for full linear logic. To appear in J. Applied Logic
3. Galatos, N., Jipsen, P., Kowalski, T., Ono, H.: Residuated Lattices: An Algebraic Glimpse at Substructural Logics. Elsevier (2007)
4. Frey, J.: A 2-Categorical Analysis of the Tripos-to-Topos Construction (preprint)
5. Hájek, P., Cintula, P.: On theories and models in fuzzy predicate logics. J. Symb. Log. 71, 863–880 (2006)
6. Höhle, U., Kubiak, T.: A non-commutative and non-idempotent theory of quantale sets. Fuzzy Sets and Systems 166, 1–43 (2011)
7. Hyland, M., Johnstone, P.T., Pitts, A.: Tripos Theory. Math. Proc. Cambridge Philos. Soc. 88, 205–232 (1980)
8. Jacobs, B.: Categorical Logic and Type Theory. Elsevier (1999)
9. Johnstone, P.T.: Stone Spaces. CUP (1982)
10. Johnstone, P.T.: Sketches of an Elephant. OUP (2002)
11. Lambek, J., Scott, P.J.: Introduction to Higher-Order Categorical Logic (1986)
12. Lawvere, F.W.: Adjointness in Foundations. Dialectica 23, 281–296 (1969)
13. Marquis, J.-P., Reyes, G.: The History of Categorical Logic: 1963-1977. In: Handbook of the History of Logic, vol. 6, pp. 689–800. Elsevier (2011)
14. Ono, H.: Algebraic semantics for predicate logics and their completeness. RIMS Kokyuroku 927, 88–103 (1995)
15. Ono, H.: Crawley Completions of Residuated Lattices and Algebraic Completeness of Substructural Predicate Logics. Studia Logica 100, 339–359 (2012)
16. Pitts, A.: Categorical Logic. In: Handbook of Logic in Computer Science, vol. 5, ch. 2. OUP (2000)

A Finite Model Property for Gödel Modal Logics

Xavier Caicedo[1], George Metcalfe[2,⋆], Ricardo Rodríguez[3], and Jonas Rogger[2]

[1] Departamento de Matemáticas, Universidad de los Andes, Bogotá, Colombia
xcaicedo@uniandes.edu.co
[2] Mathematical Institute, University of Bern, Switzerland
{george.metcalfe,jonas.rogger}@math.unibe.ch
[3] Departamento de Computación, Universidad de Buenos Aires, Argentina
ricardo@dc.uba.ar

Abstract. A new semantics with the finite model property is provided and used to establish decidability for Gödel modal logics based on (crisp or fuzzy) Kripke frames combined locally with Gödel logic. A similar methodology is also used to establish decidability, and indeed co-NP-completeness for a Gödel S5 logic that coincides with the one-variable fragment of first-order Gödel logic.

1 Introduction

Gödel modal logics combine Kripke frames of modal logics with the semantics of the well-known fuzzy (and intermediate) Gödel logic. These logics, in particular, analogues GK (for "fuzzy" frames) and GK^{C} (for "crisp" frames) of the modal logic K, have been investigated in some detail by Caicedo and Rodríguez [7,6] and Metcalfe and Olivetti [13,14]. More general approaches, focussing mainly on finite-valued modal logics, have been developed by Fitting [9,10], Priest [15], and Bou et al. [4]. Multimodal variants of GK have also been proposed as the basis for fuzzy description logics in [12] and (restricting to finite models) [3].

Axiomatizations were obtained for the box and diamond fragments of GK (where the box fragments of GK and GK^{C} coincide) in [7] and for the diamond fragment of GK^{C} in [14]. It was subsequently shown in [6] that the full logic GK is axiomatized either by adding the Fischer Servi axioms for intuitionistic modal logic IK (see [8]) to the union of the axioms for both fragments, or by adding the prelinearity axiom for Gödel logic to IK. Decidability of the diamond fragment of GK was established in [7], using the fact that the fragment has the finite model property with respect to its Kripke semantics. This finite model property fails for the box fragment of GK and GK^{C} and the diamond fragment of GK^{C}, but decidability and PSPACE-completeness for these fragments was established in [13,14] using analytic Gentzen-style proof systems.

The first main contribution of this paper is to establish the decidability of validity in full GK and GK^{C} by providing an alternative Kripke semantics for

⋆ Supported by Swiss National Science Foundation grants 20002_129507 and 200021_146748.

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 226–237, 2013.

these logics that have the same valid formulas as the original semantics, but also admit the finite model property. The key idea of the new semantics is to restrict evaluations of modal formulas at a given world to a particular finite set of truth values. We then use a similar strategy to establish decidability, and indeed co-NP completeness, for the crisp Gödel modal logic $\mathsf{GS5}^{\mathsf{C}}$ based on $\mathsf{S5}$ frames where accessibility is an equivalence relation. Moreover, this logic, an extension of the intuitionistic modal logic MIPC of Bull [5] and Prior [16] with prelinearity and a further modal axiom, corresponds exactly to the one-variable fragment of first-order Gödel logic (see [11]).

2 Gödel Modal Logics

Gödel modal logics are defined based on a language $\mathcal{L}_{\Box\Diamond}$ consisting of a fixed countably infinite set Var of (propositional) variables, denoted $p, q, \ldots$, binary connectives $\to$, $\wedge$, $\vee$, constants $\bot$, $\top$, and unary operators $\Box$ and $\Diamond$. The set of *formulas* $\mathrm{Fml}_{\Box\Diamond}$, with arbitrary members denoted $\varphi, \psi, \chi, \ldots$ is defined inductively as usual, as are *subformulas* of formulas. We call formulas of the form $\Box\varphi$ and $\Diamond\varphi$ *box-formulas* and *diamond-formulas*, respectively, and fix the *length* of a formula φ, denoted $\ell(\varphi)$, to be the number of symbols occurring in φ. We also define $\neg\varphi = \varphi \to \bot$ and let $\mathrm{Var}(\varphi)$ denote the set of all variables occurring in the formula φ.

The standard semantics of Gödel logic is characterized by the Gödel t-norm min and its residuum $\to_{\mathsf{G}}$, defined on the real unit interval $[0, 1]$ by

$$x \to_{\mathsf{G}} y = \begin{cases} y & \text{if } x > y \\ 1 & \text{otherwise.} \end{cases}$$

The Gödel modal logics GK and GK^{C} are defined semantically as generalizations of the modal logic K where connectives behave at a given world as in Gödel logic.

A *fuzzy Kripke frame* is a pair $\mathfrak{F} = \langle W, R\rangle$ where W is a non-empty set of *worlds* and $R\colon W \times W \to [0, 1]$ is a binary *fuzzy accessibility relation* on W. If $Rxy \in \{0, 1\}$ for all $x, y \in W$, then R is called *crisp* and $\mathfrak{F}$, a *crisp Kripke frame*. In this case, we often write $R \subseteq W \times W$ and Rxy to mean $Rxy = 1$.

A GK-*model* is a triple $\mathfrak{M} = \langle W, R, V\rangle$, where $\langle W, R\rangle$ is a fuzzy Kripke frame and $V\colon \mathrm{Var} \times \mathrm{W} \to [0, 1]$ is a mapping, called a *valuation*, extended to $V\colon \mathrm{Fml}_{\Box\Diamond} \times \mathrm{W} \to [0, 1]$ as follows:

$$\begin{aligned} V(\bot, x) &= 0 \\ V(\top, x) &= 1 \\ V(\varphi \to \psi, x) &= V(\varphi, x) \to_{\mathsf{G}} V(\psi, x) \\ V(\varphi \wedge \psi, x) &= \min(V(\varphi, x), V(\psi, x)) \\ V(\varphi \vee \psi, x) &= \max(V(\varphi, x), V(\psi, x)) \\ V(\Box\varphi, x) &= \inf\{Rxy \to_{\mathsf{G}} V(\varphi, y) : y \in W\} \\ V(\Diamond\varphi, x) &= \sup\{\min(Rxy, V(\varphi, y)) : y \in W\}. \end{aligned}$$

A GK^{C}-*model* satisfies the extra condition that $\langle W, R\rangle$ is a crisp Kripke frame. In this case, the conditions for $\Box$ and $\Diamond$ may also be read as

$$V(\Box\varphi, x) = \inf(\{1\} \cup \{V(\varphi, y) : Rxy\})$$
$$V(\Diamond\varphi, x) = \sup(\{0\} \cup \{V(\varphi, y) : Rxy\}).$$

A formula $\varphi \in \mathrm{Fml}_{\Box\Diamond}$ is *valid* in a GK-model $\mathfrak{M} = \langle W, R, V\rangle$ if $V(\varphi, x) = 1$ for all $x \in W$. If φ is valid in all L-models for some logic L (in particular GK or GK^{C}), then φ is said to be L-*valid*, written $\models_{\mathsf{L}} \varphi$.

It is shown in [7] that validity in the box and diamond fragments of GK are axiomatized by extending any axiom system for Gödel logic (e.g., intuitionistic logic plus the prelinearity axiom $(\varphi \to \psi) \vee (\psi \to \varphi)$) with, respectively:

$$\begin{array}{ll} \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi) & \qquad\qquad \Diamond(\varphi \vee \psi) \to (\Diamond\varphi \vee \Diamond\psi) \\ \neg\neg\Box\varphi \to \Box\neg\neg\varphi & \qquad\qquad \Diamond\neg\neg\varphi \to \neg\neg\Diamond\psi \\ \varphi \;/\; \Box\varphi & \qquad\qquad \neg\Diamond\bot \\ & \qquad\qquad \varphi \to \psi \;/\; \Diamond\varphi \to \Diamond\psi. \end{array}$$

Moreover, it was shown in [6] that extending the union of these axiomatizations with the following Fischer Servi axioms (see [8]) axiomatizes the full logic GK (equivalently, extending the intuitionistic modal logic IK with prelinearity):

$$\Diamond(\varphi \to \psi) \to (\Box\varphi \to \Diamond\psi)$$
$$(\Diamond\varphi \to \Box\psi) \to \Box(\varphi \to \psi).$$

The box fragment of GK^{C} coincides with the box fragment of GK [7], while the diamond fragment of GK^{C} is axiomatized by adding the rule $\chi \vee (\varphi \to \psi) \;/\; \Diamond\chi \vee (\Diamond\varphi \to \Diamond\psi)$ to the diamond fragment of GK [14]. No axiomatization has yet been found for the full logic GK^{C}.

Let us agree to call a model *finite* if its set of worlds is finite, and say that a logic has the *finite model property* if validity in the logic coincides with validity in all finite models of the logic. In [7], it is shown that the formula $\Box\neg\neg p \to \neg\neg\Box p$ is valid in all finite GK-models, but not in the infinite crisp model $\langle \mathbb{N}, R, V\rangle$ where $Rxy = 1$ for all $x, y \in \mathbb{N}$ and $V(p, x) = 1/(x + 1)$ for all $x \in \mathbb{N}$. That is, neither GK nor GK^{C} has the finite model property. The diamond fragment of GK (but not of GK^{C}) does have the finite model property and this can be used to show that validity in the fragment is decidable [7]. Decidability and indeed PSPACE-completeness of validity in the box and diamond fragments of both GK and GK^{C} was established in [13,14] using analytic Gentzen-style proof systems. However, decidability of validity in the full logics GK and GK^{C} has not as yet been established, and indeed will be the main goal of the following two sections.

3 A New Semantics and Finite Model Property

In order for a GK^{C}-model to render the formula $\varphi = \Box\neg\neg p \to \neg\neg\Box p$ invalid at a world x, there must be values of p at worlds accessible to x that form an

infinite descending sequence tending to but never reaching 0. This ensures that the infinite model falsifies φ, but also that no particular world acts as a "witness" to the value of $\Box p$. Our strategy in what follows will be to redefine models to allow only a finite number of values at each world that can be taken by box-formulas and diamond-formulas. A formula such as $\Box p$ can then be "witnessed" at a world where the value of p is merely "sufficiently close" to the value of $\Box p$.

Let us define a GFK-*model* as a quadruple $\mathfrak{M} = \langle W, R, T, V\rangle$, where $\langle W, R, V\rangle$ is a GK-model and $T\colon W \to \mathcal{P}_{<\omega}([0,1])$ is a function from worlds to finite sets of truth values satisfying $\{0,1\} \subseteq T(x) \subseteq [0,1]$ for all $x \in W$. If $\langle W, R, V\rangle$ is also a GK^{C}-model, then $\mathfrak{M}$ will be called a $\mathsf{GFK}^{\mathsf{C}}$-*model*.

The GFK-valuation V is extended to formulas using the same clauses for non-modal connectives as for GK-valuations, together with the revised modal connective clauses:

$$V(\Box\varphi, x) = \max\{r \in T(x) : r \leq \inf\{Rxy \to_{\mathsf{G}} V(\varphi, y) : y \in W\}\}$$
$$V(\Diamond\varphi, x) = \min\{r \in T(x) : r \geq \sup\{\min(Rxy, V(\varphi, y)) : y \in W\}\}.$$

As before, a formula $\varphi \in \mathrm{Fml}_{\Box\Diamond}$ is *valid* in a GFK-model $\mathfrak{M} = \langle W, R, T, V\rangle$ if $V(\varphi, x) = 1$ for all $x \in W$, written $\mathfrak{M} \models_{\mathsf{GFK}} \varphi$.

Observe now that for the formula $\varphi = \Box\neg\neg p \to \neg\neg\Box p$, there are very simple finite $\mathsf{GFK}^{\mathsf{C}}$-counter-models: for example, $\mathfrak{M}_0 = \langle W, R, T, V\rangle$ with $W = \{a\}$, $Raa = 1$, $T(a) = \{0,1\}$, and $V(p,a) = \frac{1}{2}$. It is easy to see that $V(\neg p, a) = 0$, $Raa \to_{\mathsf{G}} V(\neg\neg p, a) = 1$, and so $V(\Box\neg\neg p, a) = 1$. Moreover, $V(\Box p, a) = 0$ (since $Raa \to_{\mathsf{G}} V(p,a) = \frac{1}{2}$, and 0 is the next smaller element of $T(a)$); hence $V(\neg\Box p, a) = 1$ and $V(\neg\neg\Box p, a) = 0$. So $1 = V(\Box\neg\neg p, a) > V(\neg\neg\Box p, a) = 0$ and $\mathfrak{M}_0 \not\models_{\mathsf{GFK}^{\mathsf{C}}} \Box\neg\neg p \to \neg\neg\Box p$.

Of course, such an observation is useful only if the new semantics characterizes the same logics. For convenience, let us agree to write $W_{\mathfrak{M}}$, $R_{\mathfrak{M}}$, $T_{\mathfrak{M}}$, and $V_{\mathfrak{M}}$ for, respectively, the set of worlds, accessibility relation, truth value function, and valuation function of an L-model $\mathfrak{M}$ where $\mathsf{L} \in \{\mathsf{GK}, \mathsf{GK}^{\mathsf{C}}, \mathsf{GFK}, \mathsf{GFK}^{\mathsf{C}}\}$. In the next section, we prove the following:

Theorem 1. *For each* $\varphi \in \mathrm{Fml}_{\Box\Diamond}$*:*

(a) $\models_{\mathsf{GK}} \varphi$ *iff* $\models_{\mathsf{GFK}} \varphi$ *iff* φ *is valid in all* GFK*-models* $\mathfrak{M}$ *satisfying* $|W_{\mathfrak{M}}| \leq (\ell(\varphi)+2)^{\ell(\varphi)}$ *and* $|T_{\mathfrak{M}}(x)| \leq \ell(\varphi)+2$ *for all* $x \in W_{\mathfrak{M}}$*.*

(b) $\models_{\mathsf{GK}^{\mathsf{C}}} \varphi$ *iff* $\models_{\mathsf{GFK}^{\mathsf{C}}} \varphi$ *iff* φ *is valid in all* $\mathsf{GFK}^{\mathsf{C}}$*-models* $\mathfrak{M}$ *satisfying* $|W_{\mathfrak{M}}| \leq (\ell(\varphi)+2)^{\ell(\varphi)}$ *and* $|T_{\mathfrak{M}}(x)| \leq \ell(\varphi)+2$ *for all* $x \in W_{\mathfrak{M}}$*.*

For decidability, we then reason as follows. Observe first that for any finite GFK-model $\mathfrak{M}$ and formula φ, the values taken by the subformulas of φ and the fuzzy accessibility relation $R_{\mathfrak{M}}$ are contained in the finite set

$$U = \bigcup_{x \in W_{\mathfrak{M}}} (\{V_{\mathfrak{M}}(p,x) : p \in \mathrm{Var}(\varphi)\} \cup \{R_{\mathfrak{M}}xy : y \in W_{\mathfrak{M}}\} \cup T_{\mathfrak{M}}(x)).$$

Moreover, using Lemma 1(c) below, we may assume without loss of generality that $U = \{0, \frac{1}{|U|-1}, \ldots, \frac{|U|-2}{|U|-1}, 1\}$. Hence, by Theorem 1, to check whether φ is

GK-valid or GK^{C}-valid, it suffices to consider finitely many different finite GFK-models or $\mathsf{GFK}^{\mathsf{C}}$-models $\mathfrak{M}$ (with $|W_{\mathfrak{M}}| \leq (\ell(\varphi)+2)^{\ell(\varphi)}$). So we obtain:

Theorem 2. *Validity in* GK *and* GK^{C} *are decidable.*

4 Proof of Theorem 1

We begin by fixing some useful notation. For a fuzzy Kripke frame $\langle W, R\rangle$, we define a crisp relation $R^+ = \{(x,y) \in W^2 : Rxy > 0\}$ and let $R^+[x] = \{y \in W : R^+xy\}$ for each $x \in W$.

We call $\langle W', R'\rangle$ a *subframe* of $\langle W, R\rangle$, written $\langle W', R'\rangle \subseteq \langle W, R\rangle$, if $W' \subseteq W$ and R' is R restricted to W'. A *submodel* $\widehat{\mathfrak{M}}$ of a model $\mathfrak{M}$ is based on a subframe $\langle W_{\widehat{\mathfrak{M}}}, R_{\widehat{\mathfrak{M}}}\rangle \subseteq \langle W_{\mathfrak{M}}, R_{\mathfrak{M}}\rangle$, with $T_{\widehat{\mathfrak{M}}}$ (if appropriate) and $V_{\widehat{\mathfrak{M}}}$ being, respectively, $T_{\mathfrak{M}}$ and $V_{\mathfrak{M}}$ restricted to $W_{\widehat{\mathfrak{M}}}$. In particular, given $X \subseteq W_{\mathfrak{M}}$, the *submodel of* $\mathfrak{M}$ *generated by* X is the smallest submodel $\widehat{\mathfrak{M}}$ of $\mathfrak{M}$ satisfying $X \subseteq W_{\widehat{\mathfrak{M}}}$ and for all $x \in W_{\widehat{\mathfrak{M}}}$, if $y \in R^+_{\mathfrak{M}}[x]$ then $y \in W_{\widehat{\mathfrak{M}}}$. Also, $\mathfrak{M}$ will be called a *tree-model* if $\langle W_{\mathfrak{M}}, R^+_{\mathfrak{M}}\rangle$ is a tree, and the *height* $hg(\mathfrak{M})$ of $\mathfrak{M}$ is the height of $\langle W_{\mathfrak{M}}, R^+_{\mathfrak{M}}\rangle$ (possibly ∞).

Parts (a) and (b) of the following lemma generalize well-known results for the modal logic K (see, e.g., [2]), while part (c) generalizes a useful result from [14] (Lemma 3.1). Their proofs will be omitted here, but follow very closely the ideas of the previous proofs from the references.

Lemma 1. *Let* $\mathsf{L} \in \{\mathsf{GK}, \mathsf{GK}^{\mathsf{C}}, \mathsf{GFK}, \mathsf{GFK}^{\mathsf{C}}\}$ *and let* $\mathfrak{M}$ *be an* L*-model.*

(a) *Given any generated submodel* $\widehat{\mathfrak{M}}$ *of* $\mathfrak{M}$, $V_{\widehat{\mathfrak{M}}}(\varphi, x) = V_{\mathfrak{M}}(\varphi, x)$ *for all* $x \in W_{\widehat{\mathfrak{M}}}$, *and* $\varphi \in \mathrm{Fml}_{\Box\Diamond}$.

(b) *Given* $x_0 \in W_{\mathfrak{M}}$ *and* $\varphi \in \mathrm{Fml}_{\Box\Diamond}$, *there is an* L*-tree-model* $\widehat{\mathfrak{M}}$ *with root* $\widehat{x}_0$ *and* $hg(\widehat{\mathfrak{M}}) \leq \ell(\varphi)$ *satisfying* $V_{\widehat{\mathfrak{M}}}(\varphi, \widehat{x}_0) = V_{\mathfrak{M}}(\varphi, x_0)$.

(c) *Given an order-embedding* $h\colon [0,1] \to [0,1]$ *satisfying* $h(0) = 0$ *and* $h(1) = 1$, *consider* $\widehat{\mathfrak{M}}$ *with* $W_{\widehat{\mathfrak{M}}} = W_{\mathfrak{M}}$, $R_{\widehat{\mathfrak{M}}}xy = h(R_{\mathfrak{M}}xy)$, $T_{\widehat{\mathfrak{M}}}(x) = h(T_{\mathfrak{M}}(x))$, *and* $V_{\widehat{\mathfrak{M}}}(p, x) = h(V_{\mathfrak{M}}(p, x))$ *for all* $x, y \in W_{\mathfrak{M}}$ *and* $p \in \mathrm{Var}$. *Then* $V_{\widehat{\mathfrak{M}}}(\varphi, x) = h(V_{\mathfrak{M}}(\varphi, x))$ *for all* $\varphi \in \mathrm{Fml}_{\Box\Diamond}$ *and* $x \in W_{\mathfrak{M}}$.

Note that the tree in (b), although it is of finite height, can still be infinitely branching and thus contain infinitely many nodes (i.e., worlds).

We now provide the key construction of a GK-tree-model taking the same values for formulas at its root as a given GFK-tree-model. Note first that the original GFK-model without the function T cannot play this role in general since the infimum or supremum required for calculating the value of a box or diamond formula might not be in the set $T(x_0)$ (where x_0 is the root world). This problem is resolved by taking infinitely many order-isomorphic copies of the original GFK-model (without T) in such a way that the open intervals between members of $T(x_0)$ are "squeezed" closer to either their lower or upper bounds. The obtained infima and suprema will then coincide with the next smaller or

larger member of $T(x_0)$, that is, the required values of the formulas at x_0 in the original GFK-model.

Lemma 2. *For any* GFK*-tree-model* $\mathfrak{M} = \langle W, R, T, V\rangle$ *of finite height with root* x_0*, there is a* GK*-tree-model* $\widehat{\mathfrak{M}} = \langle \widehat{W}, \widehat{R}, \widehat{V}\rangle$ *with root* $\widehat{x}_0$*, such that* $\widehat{V}(\varphi, \widehat{x}_0) = V(\varphi, x_0)$ *for all* $\varphi \in \mathrm{Fml}_{\Box\Diamond}$*. Moreover, if* $\mathfrak{M}$ *is crisp, then so is* $\widehat{\mathfrak{M}}$*.*

Proof. The lemma is proved by induction on $hg(\mathfrak{M})$. The base case $hg(\mathfrak{M}) = 0$ is immediate, fixing $\widehat{\mathfrak{M}} = \langle \widehat{W}, \widehat{R}, \widehat{V}\rangle$ with $\widehat{W} = W = \{x_0\}$, $\widehat{R} = R = \emptyset$, and $\widehat{V} = V$. For the inductive step $hg(\mathfrak{M}) = n + 1$, define for all $y \in R^+[x_0]$, $\mathfrak{M}_y = \langle W_y, R_y, T_y, V_y\rangle$ as the submodel of $\mathfrak{M}$ generated by $\{y\}$. That is, $\mathfrak{M}_y$ is a GFK-tree-model of finite height with root y, $hg(\mathfrak{M}_y) \leq n$, and, by Lemma 1(a), $V_y(\varphi, x) = V(\varphi, x)$ for all $x \in W_y$ and $\varphi \in \mathrm{Fml}_{\Box\Diamond}$. So, by the induction hypothesis, for each $y \in R^+[x_0]$, there is a GK-tree-model $\widehat{\mathfrak{M}}_y = \langle \widehat{W}_y, \widehat{R}_y, \widehat{V}_y\rangle$ (crisp if $\mathfrak{M}$ is crisp) with root $\widehat{y}$ such that $\widehat{V}_y(\varphi, \widehat{y}) = V_y(\varphi, y)$ $(= V(\varphi, y))$ for all $\varphi \in \mathrm{Fml}_{\Box\Diamond}$.

We now define infinitely many copies of our models $\widehat{\mathfrak{M}}_y$ such that at each copy, all the values of our formulas (and fuzzy accessibility relation) get "squeezed" closer and closer towards the next smaller (or next larger) element of $T(x_0)$. This is achieved by defining for each $k \in \mathbb{Z}^+$, an order-embedding (using Lemma 1(c)) that "squeezes" the open interval between two members α_i and α_{i+1} of $T(x_0)$ into the interval $(\alpha_i, \alpha_i + \frac{1}{k})$ (or $(\alpha_{i+1} - \frac{1}{k}, \alpha_{i+1})$), which gets infinitely small as k approaches infinity.

More formally, consider $T(x_0) = \{\alpha_1, \ldots, \alpha_m\}$ with $0 = \alpha_1 < \ldots < \alpha_m = 1$ and define a family of order-embeddings $\{h_k\}_{k\in\mathbb{Z}^+}$ from $[0,1]$ into $[0,1]$ satisfying $h_k(0) = 0$ and $h_k(1) = 1$, such that

$$\begin{array}{rl} h_k(\alpha_i) = \alpha_i & \text{for all } i \leq m \text{ and } k \in \mathbb{Z}^+ \\ h_k[(\alpha_i, \alpha_{i+1})] = (\alpha_i, \min(\alpha_i + \frac{1}{k}, \alpha_{i+1})) & \text{for all } i \leq m-1 \text{ and even } k \in \mathbb{Z}^+ \\ h_k[(\alpha_i, \alpha_{i+1})] = (\max(\alpha_i, \alpha_{i+1} - \frac{1}{k}), \alpha_{i+1}) & \text{for all } i \leq m-1 \text{ and odd } k \in \mathbb{Z}^+. \end{array}$$

Furthermore, for each $y \in R^+[x_0]$ and $k \in \mathbb{Z}^+$, we define a GK-model $\widehat{\mathfrak{M}}^k_y = \langle \widehat{W}^k_y, \widehat{R}^k_y, \widehat{V}^k_y\rangle$ such that for each $k \in \mathbb{Z}^+$ and $y \in R^+[x_0]$:

(1) $\widehat{W}^k_y$ is a copy of $\widehat{W}_y$ with distinct worlds, where $\widehat{x}^k_y$ is the corresponding copy of $\widehat{x}_y$ (the root is denoted by $\widehat{y}^k$)

(2) $\widehat{R}^k_y \widehat{x}^k_y \widehat{z}^k_y = h_k(\widehat{R}_y \widehat{x}_y \widehat{z}_y)$, for all $\widehat{x}^k_y, \widehat{z}^k_y \in \widehat{W}^k_y$

(3) $\widehat{V}^k_y(\varphi, \widehat{x}^k_y) = h_k(\widehat{V}_y(\varphi, \widehat{x}_y))$ for all $\varphi \in \mathrm{Fml}_{\Box\Diamond}$ and $\widehat{x}^k_y \in \widehat{W}^k_y$.

Note that for all $y \in R^+[x_0]$, $\widehat{x}_y, \widehat{z}_y \in \widehat{W}_y$, and $\varphi \in \mathrm{Fml}_{\Box\Diamond}$, if $\widehat{R}_y \widehat{x}_y \widehat{z}_y \to_{\mathsf{G}} \widehat{V}_y(\varphi, \widehat{x}_y) \in (\alpha_i, \alpha_{i+1})$, then $\widehat{R}^k_y \widehat{x}^k_y \widehat{z}^k_y \to_{\mathsf{G}} \widehat{V}^k_y(\varphi, \widehat{x}^k_y) \in (\alpha_i, \alpha_i + \frac{1}{k})$, for each even $k \in \mathbb{Z}^+$, and if $\min(\widehat{R}_y \widehat{x}_y \widehat{z}_y, \widehat{V}_y(\varphi, \widehat{x}_y)) \in (\alpha_i, \alpha_{i+1})$, then $\min(\widehat{R}^k_y \widehat{x}^k_y \widehat{z}^k_y, \widehat{V}^k_y(\varphi, \widehat{x}^k_y)) \in (\alpha_{i+1} - \frac{1}{k}, \alpha_{i+1})$, for each odd $k \in \mathbb{Z}^+$.

We now define the GK-tree-model $\widehat{\mathfrak{M}} = \langle \widehat{W}, \widehat{R}, \widehat{V}\rangle$ with

$$\widehat{W} = \bigcup_{y \in R^+[x_0]} \bigcup_{k \in \mathbb{Z}^+} \widehat{W}_y^k \cup \{\widehat{x}_0\}$$

$$\widehat{R}xz = \begin{cases} h_k(Rx_0y) & \text{if } x = \widehat{x}_0 \text{ and } z = \widehat{y}^k \text{ for some } y \in R^+[x_0] \text{ and } k \in \mathbb{Z}^+ \\ \widehat{R}_y^k xz & \text{if } x, z \in \widehat{W}_y^k \text{ for some } y \in R^+[x_0] \text{ and } k \in \mathbb{Z}^+ \\ 0 & \text{otherwise} \end{cases}$$

$$\widehat{V}(p,x) = \begin{cases} V(p,x_0) & \text{if } x = \widehat{x}_0 \\ \widehat{V}_y^k(p,x) & \text{if } x \in \widehat{W}_y^k \text{ for some } y \in R^+[x_0] \text{ and } k \in \mathbb{Z}^+. \end{cases}$$

If $\mathfrak{M}$ is crisp, then for all $y \in R^+[x_0]$, $\widehat{\mathfrak{M}}_y$ is crisp and so also are $\widehat{\mathfrak{M}}_y^k$ for all $k \in \mathbb{Z}^+$. Hence, by construction, $\widehat{\mathfrak{M}}$ is crisp. Moreover, $\widehat{V}_y^k(\varphi, \widehat{x}_y^k) = \widehat{V}(\varphi, \widehat{x}_y^k)$ for all $\varphi \in \mathrm{Fml}_{\Box\Diamond}$ and $\widehat{x}_y^k \in \widehat{W} \setminus \{\widehat{x}_0\}$.

Now we prove that $\widehat{V}(\varphi, \widehat{x}_0) = V(\varphi, x_0)$ for all $\varphi \in \mathrm{Fml}_{\Box\Diamond}$, proceeding by induction on $\ell(\varphi)$. The base case $\ell(\varphi) = 1$ follows directly from the definition of $\widehat{V}$. For the inductive step, the cases for the non-modal connectives follow easily using the induction hypothesis. Let us just consider the case $\varphi = \Box\psi$, the case $\varphi = \Diamond\psi$ being very similar. There are two possibilities. Suppose first that

$$V(\Box\psi, x_0) = \max\{r \in T(x_0) : r \leq \inf\{Rx_0y \to_{\mathsf{G}} V(\psi, y) : y \in W\}\} = 1.$$

Then for all $y \in R^+[x_0]$, $Rx_0y \leq V(\psi, y)$ and by Lemma 1(a), $V(\psi, y) = V_y(\psi, y) = \widehat{V}_y(\psi, \widehat{y})$. Thus $Rx_0y \leq \widehat{V}_y(\psi, \widehat{y})$ and therefore, for all $k \in \mathbb{Z}^+$ and $y \in R^+[x_0]$,

$$\widehat{R}\widehat{x}_0\widehat{y}^k = h^k(Rx_0y) \leq h^k(\widehat{V}_y(\psi, \widehat{y})) = \widehat{V}_y^k(\psi, \widehat{y}^k) = \widehat{V}(\psi, \widehat{y}^k).$$

It follows that

$$\widehat{V}(\Box\psi, \widehat{x_0}) = \inf\{\widehat{R}\widehat{x}_0 z \to_{\mathsf{G}} \widehat{V}(\psi, z) : z \in \widehat{W}\} = 1 = V(\Box\psi, x_0).$$

Now suppose that $V(\Box\psi, x_0) = \alpha_i < 1$ for some $i \leq m-1$. Then $Rx_0z \to_{\mathsf{G}} V(\psi, z) \geq \alpha_i$ for all $z \in W$, and thus, $(\star)$, $\widehat{R}\widehat{x}_0 z \to_{\mathsf{G}} \widehat{V}(\psi, z) \geq \alpha_i$ for all $z \in \widehat{W}$, by construction using the order-embeddings $\{h_k\}_{k \in \mathbb{Z}^+}$.

There are two subcases. First, suppose that there is at least one $y \in W$ such that $Rx_0y \to_{\mathsf{G}} V(\psi, y) = \alpha_i$; call it y_0. This means that $Rx_0y_0 > V(\psi, y_0) = \alpha_i$ and for all $k \in \mathbb{Z}^+$, $\widehat{V}(\psi, \widehat{y}_0^k) = \widehat{V}_{y_0}^k(\psi, \widehat{y}_0^k) = h_k(\widehat{V}_{y_0}(\psi, \widehat{y}_0)) = h_k(V_{y_0}(\psi, y_0)) = h_k(V(\psi, y_0)) = h_k(\alpha_i) = \alpha_i$. Since $Rx_0y > \alpha_i$, also for all $k \in \mathbb{Z}^+$, $\widehat{R}\widehat{x}_0\widehat{y}_0^k = h_k(Rx_0y_0) > \alpha_i = \widehat{V}(\psi, \widehat{y}_0^k)$, and hence, using $(\star)$,

$$\widehat{V}(\Box\psi, \widehat{x}_0) = \inf\{\widehat{R}\widehat{x}_0 z \to_{\mathsf{G}} \widehat{V}(\psi, z) : z \in \widehat{W}\} = \alpha_i = V(\Box\psi, x_0).$$

Now suppose that $Rx_0y \to_{\mathsf{G}} V(\psi, y) > \alpha_i$ for all $y \in W$. Since $V(\Box\psi, x_0) = \max\{r \in T(x_0) : r \leq \inf\{Rx_0y \to_{\mathsf{G}} V(\psi, y) : y \in W\}\} = \alpha_i$, there is at least one $y \in W$ such that $Rx_0y \to_{\mathsf{G}} V(\psi, y) \in (\alpha_i, \alpha_{i+1})$; call it y_0. Then, by construction, for any $\varepsilon > 0$ there is a $k \in \mathbb{Z}^+$ such that $h_k(Rx_0y_0 \to_{\mathsf{G}} V(\psi, y_0)) = h_k(Rx_0y_0) \to_{\mathsf{G}} h_k(V(\psi, y_0)) = \widehat{R}\widehat{x}_0\widehat{y}_0^k \to_{\mathsf{G}} \widehat{V}(\psi, \widehat{y}_0^k) \in (\alpha_i, \alpha_i + \varepsilon)$. Using $(\star)$,

$$\widehat{V}(\Box\psi, \widehat{x}_0) = \inf\{\widehat{R}\widehat{x}_0 z \to_{\mathsf{G}} \widehat{V}(\psi, z) : z \in \widehat{W}\} = \alpha_i = V(\Box\psi, x_0). \qquad \square$$

A subset $\Sigma \subseteq \mathrm{Fml}_{\Box\Diamond}$ will be called a *fragment* iff it is closed with respect to taking subformulas and contains $\bot$ and $\top$. For a formula $\varphi \in \mathrm{Fml}_{\Box\Diamond}$, we let $\Sigma(\varphi)$ be the smallest fragment containing φ. Clearly, $|\Sigma(\varphi)| \leq \ell(\varphi) + 2$.

We now show that given any finite fragment Σ and GK-tree-model $\mathfrak{M}$, we are able to "prune" (i.e., remove branches from) $\mathfrak{M}$ and introduce a suitable function T to obtain a finite GFK-tree-model $\widehat{\mathfrak{M}}$ such that the evaluations of formulas in Σ at the roots of $\mathfrak{M}$ and $\widehat{\mathfrak{M}}$ coincide.

Lemma 3. *Let $\Sigma \subseteq \mathrm{Fml}_{\Box\Diamond}$ be a finite fragment. Then for any* GK*-tree-model $\mathfrak{M} = \langle W, R, V\rangle$ of finite height with root x_0, there is a finite* GFK*-tree-model $\widehat{\mathfrak{M}} = \langle \widehat{W}, \widehat{R}, \widehat{T}, \widehat{V}\rangle$ with $\langle \widehat{W}, \widehat{R}\rangle \subseteq \langle W, R\rangle$, root $x_0 \in \widehat{W}$, $|\widehat{W}| \leq |\Sigma|^{hg(\mathfrak{M})}$, and $|\widehat{T}(x)| \leq |\Sigma|$ for all $x \in \widehat{W}$, such that $\widehat{V}(\varphi, x_0) = V(\varphi, x_0)$ for all $\varphi \in \Sigma$. Moreover, if $\mathfrak{M}$ is crisp, then so is $\widehat{\mathfrak{M}}$.*

Proof. Let $\Sigma_\Box$ be the set of all box-formulas in Σ, $\Sigma_\Diamond$ the set of all diamond-formulas in Σ, and Σ_{Var} the set of all variables in Σ. Let us also define $V_x[\Delta] = \{V(\varphi, x) : \varphi \in \Delta\}$ for any $x \in W$ and $\Delta \subseteq \mathrm{Fml}_{\Box\Diamond}$. We prove the lemma by induction on $hg(\mathfrak{M})$. For the base case, it suffices to define $\widehat{W} = W$, $\widehat{R} = R = \emptyset$, $\widehat{V} = V$, and $\widehat{T}(x_0) = \{0, 1\}$.

For the induction step $hg(\mathfrak{M}) = n + 1$, consider for each $y \in R^+[x_0]$, the submodel $\mathfrak{M}_y = \langle W_y, R_y, V_y\rangle$ of $\mathfrak{M}$ generated by $\{y\}$. It is clear that each $\mathfrak{M}_y$ is a GK-tree-model of finite height with root y and $hg(\mathfrak{M}_y) \leq n$. Hence, by the induction hypothesis, for each $y \in R^+[x_0]$ there is a finite GFK-tree model $\widehat{\mathfrak{M}}_y = \langle \widehat{W}_y, \widehat{R}_y, \widehat{T}_y, \widehat{V}_y\rangle$ with $\langle \widehat{W}_y, \widehat{R}_y\rangle \subseteq \langle W_y, R_y\rangle$ and root y, such that for all $\varphi \in \Sigma$: $\widehat{V}_y(\varphi, y) = V_y(\varphi, y)$ $(= V(\varphi, y))$. Moreover, we know for all $y \in R^+[x_0]$ that $|\widehat{W}_y| \leq |\Sigma|^n$ and $|\widehat{T}_y(x)| \leq |\Sigma|$ for all $x \in \widehat{W}_y$.

We now choose a finite number of appropriate $y \in R^+[x_0]$ in order to build our finite GFK-model. To this end, note that we can view $V_{x_0}[\Sigma_\Box \cup \Sigma_\Diamond] \cup \{0, 1\}$ as a finite set $\{\alpha_1, \ldots, \alpha_m\}$ with $0 = \alpha_1 < \ldots < \alpha_m = 1$. Then, for each $\Box\psi \in \Sigma_\Box$, such that $V(\Box\psi, x_0) = \alpha_i < 1$, choose a $y = y_{\Box\psi} \in R^+[x_0]$ such that $Rx_0y_{\Box\psi} \to_{\mathsf{G}} V(\psi, y_{\Box\psi}) < \alpha_{i+1}$, and for each $\Diamond\psi \in \Sigma_\Diamond$, such that $V(\Diamond\psi, x) = \alpha_i > 0$, choose a $y = y_{\Diamond\psi} \in R^+[x_0]$ such that $\min(Rx_0y_{\Diamond\psi}, V(\psi, y_{\Diamond\psi})) > \alpha_{i-1}$. Then let $Y = \{y_{\Box\psi} \in R^+[x_0] : \Box\psi \in \Sigma_\Box\} \cup \{y_{\Diamond\psi} \in R[x_0] : \Diamond\psi \in \Sigma_\Diamond\}$, noting that Y is finite and $|Y| \leq |\Sigma_\Box \cup \Sigma_\Diamond|$.

Now we define $\widehat{\mathfrak{M}} = \langle \widehat{W}, \widehat{R}, \widehat{T}, \widehat{V}\rangle$ with $\widehat{W} = \bigcup_{y \in Y} \widehat{W}_y \cup \{x_0\}$ and

$$\widehat{R}xz = \begin{cases} Rx_0z, & \text{if } x = x_0 \text{ and } z \in R^+[x_0] \\ \widehat{R}_yxz & \text{if } x, z \in \widehat{W}_y, \text{ for some } y \in Y \\ 0 & \text{otherwise} \end{cases}$$

$$\widehat{T}(x) = \begin{cases} V_{x_0}[\Sigma_\Box \cup \Sigma_\Diamond] \cup \{0, 1\} & \text{if } x = x_0 \\ \widehat{T}_y(x), & \text{if } x \in \widehat{W}_y \text{ for some } y \in Y \end{cases}$$

$$\widehat{V}(p, x) = \begin{cases} V(p, x_0) & \text{if } x = x_0 \\ \widehat{V}_y(p, x) & \text{if } x \in \widehat{W}_y, \text{ for some } y \in Y. \end{cases}$$

Note that, since for all $y \in W$, $\langle \widehat{W}_y, \widehat{R}_y \rangle \subseteq \langle W_y, R_y \rangle \subseteq \langle W, R \rangle$, it follows that $\langle \widehat{W}_y, \widehat{R}_y \rangle \subseteq \langle \widehat{W}, \widehat{R} \rangle \subseteq \langle W, R \rangle$ for all $y \in Y$. Furthermore, because $\widehat{W}_y$ is finite for all $y \in Y \subseteq W$, $\widehat{W}$ is finite. Therefore, it is clear that, ($\star$), $\widehat{R}^+[x_0] = Y \subseteq R^+[x_0]$ and for all $y \in Y$, $\widehat{R}x_0y = Rx_0y$ and $\widehat{V}(\varphi, y) = V(\varphi, y)$. Then, by an induction on the length of φ, we further show that for all $\varphi \in \Sigma$: $\widehat{V}(\varphi, x_0) = V(\varphi, x_0)$.

The base case follows directly from the definition of $\widehat{V}$. For the inductive step, let $\varphi \in \Sigma$ be of the form $\varphi = \Box\psi$ (the non-modal cases follow directly, using the induction hypothesis). We need to consider two cases. First, let $Rx_0y \to_{\mathsf{G}} V(\psi, y) = 1$ for all $y \in R^+[x_0]$. This implies that for all $y \in R^+[x_0]$: $Rx_0y \leq V(\psi, y)$, and by ($\star$), that $\widehat{R}x_0y \to_{\mathsf{G}} \widehat{V}(\psi, y) = 1$ for all $y \in \widehat{R}^+[x_0]$ and thus for all $y \in \widehat{W}$. Because $1 \in \widehat{T}(x_0)$, we conclude that

$$\widehat{V}(\Box\psi, x_0) = \max\{r \in \widehat{T}(x_0) : r \leq \inf\{\widehat{R}x_0y \to_{\mathsf{G}} \widehat{V}(\psi, y) : y \in \widehat{W}\}\} = 1.$$

For the second case, let $V(\Box\psi, x_0) = \inf\{Rx_0y \to_{\mathsf{G}} V(\psi, y) : y \in W\} = \alpha_i < 1$ for some $i \in \{1, \ldots, m-1\}$, call it i_0. By ($\star$), it follows that for all $y \in \widehat{W}$, $\widehat{R}x_0y \to_{\mathsf{G}} \widehat{V}(\psi, y) = Rx_0y \to_{\mathsf{G}} V(\psi, y)$. Because $\widehat{W} \subseteq W$, this implies that $\inf\{\widehat{R}x_0y \to_{\mathsf{G}} \widehat{V}(\psi, y) : y \in \widehat{W}\} \geq \inf\{Rx_0y \to_{\mathsf{G}} V(\psi, y) : y \in W\} = \alpha_{i_0}$. Furthermore, because of our choice of $y_{\Box\psi} \in \widehat{W}$, we know that $\widehat{R}x_0y_{\Box\psi} \to_{\mathsf{G}} \widehat{V}(\psi, y_{\Box\psi}) = Rx_0y_{\Box\psi} \to_{\mathsf{G}} V(\psi, y_{\Box\psi}) < \alpha_{i_0+1}$. Thus $\alpha_{i_0} \leq \inf\{\widehat{R}x_0y \to_{\mathsf{G}} \widehat{V}(\psi, y) : y \in \widehat{W}\} < \alpha_{i_0+1}$ and, by the construction of $\widehat{T}$ ($\widehat{T}(x_0) = \{\alpha_1, \ldots, \alpha_m\}$),

$$\widehat{V}(\Box\psi, x) = \max\{r \in \widehat{T}(x_0) : r \leq \inf\{\widehat{R}x_0y \to_{\mathsf{G}} \widehat{V}(\psi, y) : y \in \widehat{W}\}\} = \alpha_{i_0}.$$

The diamond-case case follows similarly to the box-case and is therefore omitted. Note also that since $\langle \widehat{W}, \widehat{R} \rangle \subseteq \langle W, R \rangle$, crispness is clearly preserved. Finally, we note that $|\widehat{W}| \leq |Y||\Sigma|^n + 1 \leq |\Sigma||\Sigma|^n = |\Sigma|^{hg(\mathfrak{M})}$ and $|\widehat{T}(x_0)| \leq |\Sigma_\Box \cup \Sigma_\Diamond| + 2$, thus $|\widehat{T}(x)| \leq |\Sigma|$ for all $x \in \widehat{W}$. This concludes the induction on the height of the model and therefore the proof of Lemma 3. □

We now have all the tools required to prove Theorem 1. Suppose first that $\not\models_{\mathsf{GFK}} \varphi$. By Lemma 1(b), φ is not valid in a GFK-tree model of finite height, and hence, by Lemma 2, $\not\models_{\mathsf{GK}} \varphi$. Conversely, suppose that $\not\models_{\mathsf{GK}} \varphi$. By Lemma 1(b), φ is not valid in a GK-tree model $\mathfrak{M}$ with $hg(\mathfrak{M}) \leq \ell(\varphi)$. But then, by Lemma 3, φ is not valid in a GFK-tree model $\widehat{\mathfrak{M}} = \langle \widehat{W}, \widehat{R}, \widehat{T}, \widehat{V} \rangle$ with (since $|\Sigma(\varphi)| \leq \ell(\varphi)+2$) $|\widehat{W}| \leq (\ell(\varphi)+2)^{\ell(\varphi)}$ and $|\widehat{T}(x)| \leq \ell(\varphi)+2$ for all $x \in \widehat{W}$. This completes the reasoning for (a), and (b) follows in exactly the same manner, using the fact that Lemmas 2 and 3 preserve crispness.

5 A Crisp Gödel S5 Logic

The crisp Gödel modal logic $\mathsf{GS5^C}$ is characterized by validity in $\mathsf{GK^C}$-models where R is an equivalence relation. In fact, it is easily seen that $\mathsf{GS5^C}$-validity corresponds to validity in *universal* $\mathsf{GS5^C}$*-models* where all worlds are related

(i.e., GK^{C}-models $\mathfrak{M}$ where $R_{\mathfrak{M}} = W_{\mathfrak{M}} \times W_{\mathfrak{M}}$). Such models may be written $\mathfrak{M} = \langle W, V\rangle$ with simplified valuation clauses

$$V(\Box\varphi, x) = \inf\{V(\varphi, y) : y \in W\} \quad \text{and} \quad V(\Diamond\varphi, x) = \sup\{V(\varphi, y) : y \in W\}.$$

$\mathsf{GS5}^{\mathsf{C}}$ can be axiomatized as an extension of the intuitionistic modal logic MIPC [5,16] with prelinearity and $\Box(\Box\varphi \vee \psi) \to (\Box\varphi \vee \Box\psi)$ [6]. It may also be viewed as the one-variable fragment of first-order Gödel logic $\mathsf{G}\forall$ (see [11]). Given a formula $\varphi \in \mathrm{Fml}_{\Box\Diamond}$, let φ^* be the first-order formula obtained by replacing each propositional variable p with the predicate $P(x)$, $\Box$ with $\forall x$, and $\Diamond$ with $\exists x$. Then $\models_{\mathsf{GS5^C}} \varphi$ if and only if $\models_{\mathsf{G}\forall} \varphi^*$. Similarly, if φ is a first-order formula with one variable, let φ° be the modal formula obtained by replacing each $P(x)$ with p, $\forall x$ with $\Box$, and $\exists x$ with $\Diamond$. Then $\models_{\mathsf{G}\forall} \varphi$ if and only if $\models_{\mathsf{GS5^C}} \varphi^\circ$.

We define a $\mathsf{GFS5}^{\mathsf{C}}$-*model* as a $\mathsf{GFK}^{\mathsf{C}}$-model $\mathfrak{M} = \langle W, R, T, V\rangle$ such that $\langle W, R, V\rangle$ is a $\mathsf{GS5}^{\mathsf{C}}$-model, and also $T(x) = T(y)$ whenever Rxy. Again, $\mathsf{GFS5}^{\mathsf{C}}$-validity amounts to validity in *universal* $\mathsf{GFS5}^{\mathsf{C}}$-*models*, written $\mathfrak{M} = \langle W, T, V\rangle$, where T may now be understood as a single fixed finite subset of $[0, 1]$, and

$$V(\Box\varphi, x) = \max\{r \in T : r \leq \inf\{V(\varphi, y) : y \in W\}\}$$
$$V(\Diamond\varphi, x) = \min\{r \in T : r \geq \sup\{V(\varphi, y) : y \in W\}\}.$$

Note in particular that in both $\mathsf{GS5}^{\mathsf{C}}$-models and $\mathsf{GFS5}^{\mathsf{C}}$-models, the truth values of box-formulas and diamond-formulas are independent of the world.

Lemma 4. *For any universal* $\mathsf{GFS5}^{\mathsf{C}}$*-model* $\mathfrak{M}$*, there is a universal* $\mathsf{GS5}^{\mathsf{C}}$*-model* $\widehat{\mathfrak{M}}$ *with* $W_{\mathfrak{M}} \subseteq W_{\widehat{\mathfrak{M}}}$*, such that* $V_{\widehat{\mathfrak{M}}}(\varphi, x) = V_{\mathfrak{M}}(\varphi, x)$ *for all* $\varphi \in \mathrm{Fml}_{\Box\Diamond}$ *and* $x \in W_{\mathfrak{M}}$*.*

Proof. We proceed similarly to the proof of Lemma 2, but since there is no accessibility relation here, an induction is not required. Given a universal $\mathsf{GFS5}^{\mathsf{C}}$-model $\mathfrak{M}$, we construct the universal $\mathsf{GS5}^{\mathsf{C}}$-model $\widehat{\mathfrak{M}}$ directly by taking infinitely many copies of $\mathfrak{M}$. Consider $T_{\mathfrak{M}} = \{\alpha_1, \ldots, \alpha_n\}$ with $0 = \alpha_1 < \ldots < \alpha_n = 1$ and, using Lemma 1(c), define a family of order-embeddings $\{h_k\}_{k\in\mathbb{Z}^+}$ exactly as in the proof of Lemma 2. For all $k \in \mathbb{Z}^+$, we define a universal $\mathsf{GS5}^{\mathsf{C}}$-model $\widehat{\mathfrak{M}}_k = \langle \widehat{W}_k, \widehat{V}_k\rangle$ such that each $\widehat{W}_k$ is a copy of $W_{\mathfrak{M}}$ with distinct worlds and $\widehat{V}_k(\varphi, x_k) = h_k(V_{\mathfrak{M}}(\varphi, x))$ for each copy x_k of $x \in W_{\mathfrak{M}}$ and $\varphi \in \mathrm{Fml}_{\Box\Diamond}$. We also let $\widehat{W}_0 = W_{\mathfrak{M}}$ and $\widehat{V}_0 = V_{\mathfrak{M}}$. Then $\widehat{\mathfrak{M}} = \langle \widehat{W}, \widehat{V}\rangle$ where

$$\widehat{W} = \bigcup_{k\in\mathbb{N}} \widehat{W}_k \qquad \text{and} \qquad \widehat{V}(p, x) = \widehat{V}_k(p, x) \ \text{ for } x \in \widehat{W}_k.$$

It then suffices to prove that $\widehat{V}(\varphi, x) = V(\varphi, x)$ for all $\varphi \in \mathrm{Fml}_{\Box\Diamond}$ and $x \in W$, proceeding by an induction on $\ell(\varphi)$ similar to the proof of Lemma 2. □

Lemma 5. *Let* $\Sigma \subseteq \mathrm{Fml}_{\Box\Diamond}$ *be a finite fragment. Then, for any universal* $\mathsf{GS5}^{\mathsf{C}}$*-model* $\mathfrak{M}$*, there is a finite universal* $\mathsf{GFS5}^{\mathsf{C}}$*-model* $\widehat{\mathfrak{M}}$ *with* $W_{\widehat{\mathfrak{M}}} \subseteq W_{\mathfrak{M}}$*, such that* $V_{\widehat{\mathfrak{M}}}(\varphi, x) = V_{\mathfrak{M}}(\varphi, x)$ *for all* $\varphi \in \Sigma$ *and* $x \in W_{\widehat{\mathfrak{M}}}$*. Moreover,* $|W_{\widehat{\mathfrak{M}}}| + |T_{\widehat{\mathfrak{M}}}| \leq 2|\Sigma|$*.*

Proof. Let $\Sigma \subseteq \mathrm{Fml}_{\Box\Diamond}$ be a finite fragment, $\mathfrak{M} = \langle W, V\rangle$ a universal $\mathsf{GS5^C}$-model, and fix $x_0 \in W$. First, define $\Sigma_\Box$, $\Sigma_\Diamond$, Σ_{Var}, and $V_x[\Delta]$ as in Lemma 3 and let $V_{x_0}[\Sigma_\Box \cup \Sigma_\Diamond] \cup \{0,1\} = \{\alpha_1, \ldots, \alpha_n\}$ with $0 = \alpha_1 < \ldots < \alpha_n = 1$. As in Lemma 3, we choose a finite number of $y \in W$. For each $\Box\psi \in \Sigma_\Box$ such that $V(\Box\psi, x_0) = \alpha_i < 1$, choose a $y = y_{\Box\psi} \in W$ such that $V(\psi, y_{\Box\psi}) < \alpha_{i+1}$, and for each $\Diamond\psi \in \Sigma_\Diamond$, such that $V(\Diamond\psi, x_0) = \alpha_i > 0$, choose a $y = y_{\Diamond\psi} \in W$ such that $V(\psi, y_{\Diamond\psi}) > \alpha_{i-1}$. Then let $\widehat{W} = \{x_0\} \cup \{y_{\Box\psi} \in W : \Box\psi \in \Sigma_\Box\} \cup \{y_{\Diamond\psi} \in W : \Diamond\psi \in \Sigma_\Diamond\}$. Clearly $\widehat{W} \subseteq W$ is finite. We define $\widehat{\mathfrak{M}} = \langle \widehat{W}, \widehat{T}, \widehat{V}\rangle$ where $\widehat{T} = V_{x_0}[\Sigma_\Box \cup \Sigma_\Diamond] \cup \{0,1\}$, and $\widehat{V}$ is V restricted to $\widehat{W}$. It then follows by induction on $\ell(\varphi)$ (omitted here) as in Lemma 3 that $\widehat{V}(\varphi, x) = V(\varphi, x)$ for all $x \in \widehat{W}$ and $\varphi \in \Sigma$. Moreover, $|\widehat{W}| \leq |\Sigma_\Box \cup \Sigma_\Diamond| + 1 \leq |\Sigma|$ and $|\widehat{T}| \leq |\Sigma_\Box \cup \Sigma_\Diamond| + 2 \leq |\Sigma|$, and therefore $|\widehat{W}| + |\widehat{T}| \leq 2|\Sigma|$. □

Hence, immediately by Lemmas 4 and 5:

Theorem 3. *For each* $\varphi \in \mathrm{Fml}_{\Box\Diamond}$*:* $\models_{\mathsf{GS5^C}} \varphi$ *iff* $\models_{\mathsf{GFS5^C}} \varphi$ *iff* φ *is valid in all universal* $\mathsf{GFS5^C}$*-models* $\mathfrak{M}$ *where* $|W_{\mathfrak{M}}| + |T_{\mathfrak{M}}| \leq 2(\ell(\varphi) + 2)$.

Finally, to check non-deterministically if a formula $\varphi \in \mathrm{Fml}_{\Box\Diamond}$ is not $\mathsf{GS5^C}$-valid, it suffices, using Lemmas 4 and 5, to guess a universal $\mathsf{GFS5^C}$-model $\mathfrak{M}$ with $|W_{\mathfrak{M}}| = \ell(\varphi)$ and values in $U = \{0, \frac{1}{\ell(\varphi)^2+1}, \ldots, \frac{\ell(\varphi)^2}{\ell(\varphi)^2+1}, 1\}$ and then to guess a world $x \in W_{\mathfrak{M}}$ and check whether $V_{\mathfrak{M}}(\varphi, x) < 1$. This means choosing $\ell(\varphi)|\mathrm{Var}(\varphi)|$ values in U for $V_{\mathfrak{M}}$ and also a subset of U for $T_{\mathfrak{M}}$. Choosing these values and the world $x \in W_{\mathfrak{M}}$ and then computing the value of $V_{\mathfrak{M}}(\varphi, x) = 1$ can be achieved in time polynomial in $\ell(\varphi)^2$. So validity in $\mathsf{GS5^C}$ is in co-NP. Since checking validity in Gödel logic is co-NP hard (see, e.g., [11]), we obtain:

Theorem 4. *Validity in* $\mathsf{GS5^C}$ *and the one-variable fragment of first-order Gödel logic is decidable and indeed co-NP-complete.*

6 Concluding Remarks

In this paper, we have established the decidability of validity in the Gödel modal logics GK and $\mathsf{GK^C}$ based, respectively, on fuzzy and crisp Kripke frames. We have also established decidability and co-NP completeness for validity in the one-variable fragment of first-order Gödel logic and, equivalently, the logic $\mathsf{GS5^C}$ based on crisp Kripke frames where accessibility is an equivalence relation. In ongoing work, we aim to determine the complexity of validity in GK and $\mathsf{GK^C}$ (both of which we conjecture to be PSPACE complete), possibly via Gentzen-style proof systems. We also intend to extend our approach to other logics. From a modal perspective, we plan to consider logics with multiple modalities (in particular, Gödel description logics) and modalities whose accessibility relations satisfy conditions such as reflexivity, symmetry, and transitivity. Moreover, in the propositional setting, we intend to treat modal logics based on so-called "projective logics" (see [1]) where similar methods should apply.

References

1. Baaz, M., Fermüller, C.G.: Analytic calculi for projective logics. In: Murray, N.V. (ed.) TABLEAUX 1999. LNCS (LNAI), vol. 1617, pp. 36–51. Springer, Heidelberg (1999)
2. Blackburn, P., de Rijke, M., Venema, Y.: Modal logic. Cambridge University Press, Cambridge (2001)
3. Bobillo, F., Delgado, M., Gómez-Romero, J., Straccia, U.: Fuzzy description logics under Gödel semantics. International Journal of Approximate Reasoning 50(3), 494–514 (2009)
4. Bou, F., Esteva, F., Godo, L., Rodríguez, R.: On the minimum many-valued logic over a finite residuated lattice. Journal of Logic and Computation 21(5), 739–790 (2011)
5. Bull, R.A.: MIPC as formalisation of an intuitionist concept of modality. Journal of Symbolic Logic 31, 609–616 (1966)
6. Caicedo, X., Rodríguez, R.: Bi-modal Gödel logic over [0,1]-valued Kripke frames. To appear in Journal of Logic and Computation
7. Caicedo, X., Rodríguez, R.: Standard Gödel modal logics. Studia Logica 94(2), 189–214 (2010)
8. Fischer Servi, G.: Axiomatizations for some intuitionistic modal logics. Rend. Sem. Mat. Polit de Torino 42, 179–194 (1984)
9. Fitting, M.C.: Many-valued modal logics. Fundamenta Informaticae 15(3-4), 235–254 (1991)
10. Fitting, M.C.: Many-valued modal logics II. Fundamenta Informaticae 17, 55–73 (1992)
11. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
12. Hájek, P.: Making fuzzy description logic more general. Fuzzy Sets and Systems 154(1), 1–15 (2005)
13. Metcalfe, G., Olivetti, N.: Proof systems for a Gödel modal logic. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS (LNAI), vol. 5607, pp. 265–279. Springer, Heidelberg (2009)
14. Metcalfe, G., Olivetti, N.: Towards a proof theory of Gödel modal logics. Log. Methods Comput. Sci. 7(2), 1–27 (2011)
15. Priest, G.: Many-valued modal logics: a simple approach. Review of Symbolic Logic 1, 190–203 (2008)
16. Prior, A.: Time and Modality. Clarendon Press, Oxford (1957)

Model Checking for Modal Dependence Logic: An Approach through Post's Lattice

Julian-Steffen Müller and Heribert Vollmer

Institut für Theoretische Informatik
Leibniz Universität Hannover
Appelstr. 4, 30167 Hannover, Germany
{mueller,vollmer}@thi.uni-hannover.de

Abstract. In this paper we investigate an extended version of modal dependence logic by allowing arbitrary Boolean connectives. Modal dependence logic was recently introduced by Jouko Väänänen by extending modal logic by a the dependence atom dep(·). In this paper we study the computational complexity of the model checking problem. For a complete classification of arbitrary Boolean functions we are using a Lattice approach introduced by Emil Post. This classification is done for all fragments of the logical language allowing modalities $\Diamond$ and $\Box$, the dependence atom, and logical symbols for arbitrary Boolean functions.

1 Introduction

Many algorithmic problems for propositional logic and its extensions are presumably computationally intractable, the most prominent of course the simple satisfiability problem SAT known to be NP-complete. For propositional modal logic, satisfiability is even PSPACE-complete [7]. Much effort has therefore been spent on identifying fragments of the logical language that admit efficient algorithms for satisfiability, see [8] for propositional logic and [6] for modal logic. These studies first extend propositional (modal) logic by allowing arbitrary Boolean connectives (i.e., logical symbols for arbitrary Boolean functions) in the formulas, and then classify the computational complexity of satisfiability for each finite subset B of allowed Boolean functions/connectives. An important tool in these complexity classifications is Post's lattice of all closed classes of Boolean functions, also known as Boolean clones, since it can be shown that the complexity of satisfiability for a logic with connectives from B depends only on the clone $[B]$ generated by B.

In this paper we are interested in modal dependence logic (MDL). This logic extends (propositional) modal logic by dependence atoms, i.e., atomic formulas that describe functional dependencies between variables. This logic was introduced recently by Väänänen [13] and examined from a complexity theoretic point of view in [11,4,9]. While the model checking problem for propositional modal logic is known to be efficiently solvable (i.e., in polynomial time) [5], it gets PSPACE-complete for modal dependence logic. The above sketched approach to identify efficiently solvable fragments making use of the structure of Post's

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 238–250, 2013.

lattice does not work here, because the semantics of the Boolean connectives is not immediate in dependence logic. For example, $\otimes$ (here called splitjunction) and $\to$ (intuitionistic implication) are defined in somewhat non-classical ways making use of so called team-semantics, see [13,11]. Ebbing and Lohmann [4] examined the complexity of a few fragments, but the fragments were given by somewhat arbitrary bases; their results determine the complexity of model checking in some important special cases, but the full picture is still missing.

In the present paper we introduce a novel approach to the study of fragments of dependence logic: We do not aim at a classification of all fragments defined by arbitrary dependence connectives like splitjunction or intuitionistic implication. Instead we make a distinction between dependence connectives on the one side and classical Boolean connectives on the other side. In other words, we introduce connectives given by Boolean function into dependence logic and define their semantics in the classical way. Then it can be observed that for this latter class of connectives an approach via Post's lattice is possible, and this is what we exploit in this paper. We achieve a classification of the model checking problem for modal dependence logic for all fragments of the language making use of dependence atoms, one or both modalities, and arbitrary Boolean connectives. As we will explain, the complexity will depend not on the particular choice of Boolean functions that we allow in our formulas, but on the clones in Post's lattice that is defined by the set of Boolean functions. In this way, the mentioned results from [4] will allow us more generally to determine the complexity of model checking for all monotone clones. We then extend these observations to all the remaining clones by considering also the connectives of logical negation and exclusive-or.

For the results presented here, we do not consider dependence connectives (splitjunction, intuitionistic implication, etc.), but we come back to this question in the conclusion.

After introducing the reader to dependence logic and our extension via arbitrary classical Boolean connectives we shortly recall basic results about Post's lattice in Sect. 2. Then, in Sect. 3 we prove our classification results. We will see that when restricting the language to the modality $\Box$, model checking becomes a very efficiently solvable task, independently of what else we allow in our language. Introducing the modality $\Diamond$, however, makes model checking hard. We obtain fragments that are NP-complete, some are complete for $\mathrm{P}^{\mathrm{NP[1]}}$. The technically most interesting theorem of our paper shows that as soon as the connective exclusive-or is present or can be simulated, model checking reaches its maximal complexity and becomes PSPACE-complete.

2 The Modal Language and Its Fragments

We first define syntax and semantics of modal dependence logic.

Definition 1. Let B be a set of Boolean functions. Then we define the set of MDL_B-formulae (B-formulae for short) as follows: Every variable p is a B-formula. If $p_1, \ldots, p_n, q$ are variables, then $\mathrm{dep}(p_1, \ldots, p_n, q)$ is a B-formula.

If f is an n-ary function in B and $\phi_1, \ldots, \phi_n$ are B-formulae, then $f(\phi_1, \ldots, \phi_n)$ is a B-formula. If ϕ is a B-formula, then $\Diamond\phi$ and $\Box\phi$ are B-formulae.

For $U \subseteq \{\Box, \Diamond, \mathrm{dep}(\cdot)\}$ we say that a B-formula is a (B, U)-formula, if it uses only logical symbols from $B \cup U$.

We remark that, as usual, we do not distinguish in our notation between a Boolean function f and a logical symbol for f.

The dependence atom $\mathrm{dep}(p_1, \ldots, p_n, q)$ is meant to express that the value of q functionally depends on those of $p_1, \ldots, p_n$. Unlike in usual modal logic, it does not make sense to evaluate such a formula in a single state but in a set of states (in this context called *team*), and this is different from evaluating the formula in each state separately.

As usual, in a Kripke structure $\mathscr{M} = (W, R, \pi)$ the set of all successors of $T \subseteq W$ is defined as $R(T) = \{s \in W \mid \exists s' \in T : (s', s) \in R\}$. Furthermore we define $R\langle T\rangle = \{T' \subseteq R(T) \mid \forall s \in T\ \exists s' \in T' : (s, s') \in R\}$.

Definition 2. Let $\mathscr{M}$ be a Kripke structure, T be a team over $\mathscr{M}$ and ϕ be a B-formula. The semantic evaluation (denoted as $\mathscr{M}, T \models \phi$) is defined by the induction below. We also define the function $\langle\cdot\rangle_T^{\mathscr{M}}$ which maps a formula to a truth value, where $\langle\phi\rangle_T^{\mathscr{M}}$ is true if and only if $\mathscr{M}, T \models \phi$.

$\mathscr{M}, \emptyset \models \phi$	true	
$\mathscr{M}, T \models p$	if	for all $w \in T : p \in \pi(w)$
$\mathscr{M}, T \models \overline{p}$	if	for all $w \in T : p \notin \pi(w)$
$\mathscr{M}, T \models \mathrm{dep}(p_1, \ldots, p_n, q)$	if	for all $w, w' \in T$: $\pi(w) \cap \{p_1, \ldots, p_n\} = \pi(w') \cap \{p_1, \ldots, p_n\}$ implies $q \in \pi(w) \Leftrightarrow q \in \pi(w')$
$\mathscr{M}, T \models f(\phi_1, \ldots, \phi_n)$	if	$f(\langle\phi_1\rangle_T^{\mathscr{M}}, \ldots, \langle\phi_n\rangle_T^{\mathscr{M}}) = 1$
$\mathscr{M}, T \models \Diamond\phi$	if	there is a $T' \in R\langle T\rangle$ such that $\langle\phi\rangle_{T'}^{\mathscr{M}}$
$\mathscr{M}, T \models \Box\phi$	if	$\langle\phi\rangle_{R(T)}^{\mathscr{M}}$

These modalities, as defined by Väänänen, do not fulfill the usual dualities; as a technical tool for our upcoming results we therefore define a further modality by $\boxdot\phi \equiv \neg\Diamond\neg\phi$. Also, note that $\Box\phi \equiv \neg\Box\neg\phi$ holds and that the empty team satisfies all formulae, including $\neg\phi$.

We collect some important observations, all of which follows quite immediately from the definitions.

Lemma 1. *Let ϕ, ϕ' be* MDL *formulae. Then the following axioms are satisfied on all Kripke models $\mathscr{M}$.*

1. $\Box(\phi \wedge \phi') \to \Box\phi \wedge \Box\phi'$.
2. $\Box(\phi \vee \phi') \to \Box\phi \vee \Box\phi'$.
3. $\Box(\neg\phi) \to \neg\Box\phi$.
4. *Let f^n be a n-ary Boolean formula over the basis* B. *Then* $\Box f(\phi_1, \ldots, \phi_n) \to f(\Box\phi_1, \ldots, \Box, \phi_n)$ *holds.*

Proof. Let ϕ, ϕ' be MDL formulae, $\mathscr{M}$ be a Kripke model and T be a team over $\mathscr{M}$. Let $\odot \in \{\vee, \wedge\}$. Then the axioms 1 and 2 will follow by simple equivalencies from the definition as follows:

$$\begin{aligned}\mathscr{M}, T \models \Box(\phi \odot \phi') &\Leftrightarrow \mathscr{M}, R(T) \models (\phi \odot \phi') \\ &\Leftrightarrow \mathscr{M}, R(T) \models \phi \text{ and/or } \mathscr{M}, R(T) \models \phi' . \\ &\Leftrightarrow \mathscr{M}, T \models \Box\phi \odot \Box\phi'\end{aligned}$$

The axiom 3 is stating a self duality property of $\Box$ and can be proven by the following equivilance.

$$\begin{aligned}\mathscr{M}, T \models \Box\neg\phi &\Leftrightarrow \mathscr{M}, R(T) \models \neg\phi \Leftrightarrow \mathscr{M}, R(T) \not\models \phi \\ &\Leftrightarrow \mathscr{M}, T \not\models \Box\phi \Leftrightarrow \mathscr{M}, T \models \neg\Box\phi\end{aligned}$$

Axiom 4 directly follows from the axioms 1-3, because each Boolean function f can be efficiently transformed into a logically equivalent function f' over the basis $\{\wedge, \vee, \neg\}$. By applying the axioms 1-3 iteratively on f' we obtain the axiom 4. □

The algorithmic problem family whose computational complexity we want to determine in this paper is defined as follows. Here, B denotes a finite set of Boolean functions, and $U \subseteq \{\Box, \Diamond, \mathrm{dep}(\cdot)\}$.

Problem:	MDL-MC(B, U)
Description:	Model checking problem for (B, U)-formulae.
Input:	(B, U)-formula ϕ, Kripke model $\mathscr{M}$ and team T.
Question:	Is ϕ satisfied in $\mathscr{M}$ on T?

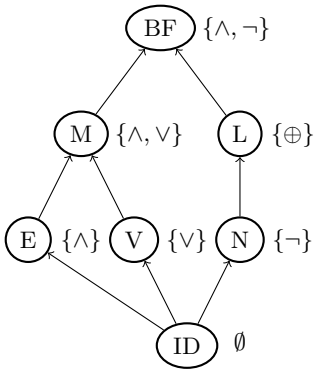


Fig. 1. Post's Lattice for Boolean clones with both constants and their "standard" basises

Post's Lattice

Emil Post [10] classified the lattice of all closed sets of Boolean functions—called *clones*—and found a finite base for each clone. For an arbitrary finite set B of Boolean functions we define $[B]$ to be the clone generated by B, i.e., the class of all Boolean functions that contains B, all projections (identities), and is closed under composition. A list of all clones as well as the full inclusion graph can be found, for example, in [2]. Whereas in general there is an infinite set of clones, for model checking luckily there are only seven different clones [1]. This is essentially due to the fact that the constants for *false* and *true* do not need to be part of the language but can be expressed by atoms that are either nowhere or everywhere satisfied in the model. In other words, MDL-MC(B) $\equiv$ MDL-MC(B $\cup \{0, 1\}$) ($\equiv$ denotes a suitable reduction, e.g., polynomial-time, logspace, or even constant-depth).

But there are only seven clones that contain the constants, see Fig. 1. This means (and is proved formally in [12]) that if one wishes to study the computational complexity of model checking for propositional formulas with logical connectives restricted to some set B of Boolean functions, it is not necessary to consider all infinite possibilities for such sets B but actually suffices to consider these seven clones, depicted in Fig. 1, where we describe the clones by their standard bases (we use $\oplus$ to denote the exclusive or).

As an example, notice that even though $\{\wedge, \oplus\}$ is not a base for all Boolean functions, it suffices to express all Boolean functions w.r.t. model checking problems because of the "free" existence of the constants; e.g., $\neg x = x \oplus 1$ and $x \vee y = ((x \oplus 1) \wedge (y \oplus 1)) \oplus 1$.

To summarize, given any finite set B of Boolean functions/propositional connectives, the computational complexity of model checking for formulas over B is equivalent to the complexity of model checking for one of the bases given in Fig. 1.

Hence, in all upcoming results, if we classify the computational complexity of a model checking problem for the bases in Fig. 1, we have in fact achieved a full complexity classification for all finite sets B of Boolean connectives.

3 Complexity Results

We first study fragments of the modal language with $\Box$ as only modality. The following theorem completely clarifies the complexity of all arising fragments.

Theorem 1. *For all finite sets B,* MDL-MC$(B, \{\Box, \mathrm{dep}(\cdot)\})$ *is* NL*-complete.*

Proof. To prove hardness, we give a reduction from the standard NL-complete graph reachability problem. Let $\langle G = (V, E), s, t\rangle$ be a instance of REACH, then we construct a Kripke model $\mathscr{M} = (W, R, \pi)$ as follows:

$$(W, R) := (V, E \cup \{(v, v) \mid v \in V\})$$
$$\pi^{-1}(q) := V \setminus \{t\}$$

Now we conclude: If there is no path in G from s to t, then t is not contained in one of the first $|V|-1$ breath depth first search levels. By the definition of π it holds, that all vertices in the first $|V|-1$ levels are labeled with the proposition q and therefore $\Box^{(|W|-1)}\mathrm{dep}(q)$ holds on $\mathscr{M}$ at the starting team $\{s\}$. The converse direction is proved similarly.

The membership result uses a well known fact by Buss [3], that propositional formulae can be evaluated in NL. Because of Lemma 1 we know that modalities can only occur as a sequence at the leafs of the formula tree followed by an atomic formula. Every time the Buss algorithm needs to evaluate such a modal leaf we evaluate that leaf in NL with Algorithm 1 and the Buss algorithm can proceed with the corresponding Boolean value. This procedure is shown in the algorithm, where ϕ_{leaf} is such a modal leaf and $\mathrm{Depth}_{\mathrm{modal}}(\phi_{\mathrm{leaf}})$ gives the length of the modal sequence.

Input : $\mathrm{MDL}_{\mathrm{BF}}$ formula ϕ, Kripke model $\mathscr{M} = (W, R, \pi)$, team $T \subseteq W$ and leaf ϕ_{leaf}
Output: Is ϕ_{leaf} satisfied in $\mathscr{M}$ on T?
universally guess $w_1 \in W$ with $d(t, w_1) = \mathrm{Depth}_{\mathrm{modal}}(\phi_{\mathrm{leaf}})$ for $t \in T$
universally guess $w_2 \in W$ with $d(t, w_2) = \mathrm{Depth}_{\mathrm{modal}}(\phi_{\mathrm{leaf}})$ for $t \in T$
if $\phi_{leaf} = \mathrm{dep}(p_1, \ldots, p_n)$ **then**
  labellingAgrees $\leftarrow$ **true**
  for $i \leftarrow 0$ **to** $n-1$ **do**
    if not $(p_i \in \pi(w_1) \Leftrightarrow p_i \in \pi(w_2))$ **then**
      labellingAgrees $\leftarrow$ **False**
    end
  end
  if *labellingAgrees* **then**
    if not $(p_n \in \pi(w_1) \Leftrightarrow p_n \in \pi(w_2))$ **then**
      reject
    end
  end
else if $\phi_{leaf} = p$ **then**
  if $p \notin \pi(w_1)$ **then**
    reject
  end
else if $\phi_{leaf} = \overline{p}$ **then**
  if $p \in \pi(w_1)$ **then**
    reject
  end
end
accept

Algorithm 1. co-NL leaf checking algorithm MDL-MC(BF, $\{\Box\}$)

□

In the rest of the section we study the modal language with modality $\Diamond$. However, we will see that all obtained classifications hold as well for the modal language with both modalities $\Diamond$ and $\Box$. The results we will obtain are summarized in Fig. 2.

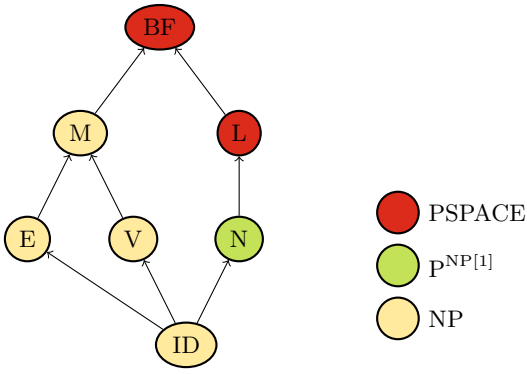


Fig. 2. Complexity of MDL-MC($\{\Diamond, \mathrm{dep}(\cdot)\}$) and MDL-MC($\{\Diamond, \Box, \mathrm{dep}(\cdot)\}$)

In order to prove the upper bound for the negation clone N, we have to prove a property called downwards closure for the universal modal operator.

We say that a logic has the *downwards closure property* if for all Kripke models $\mathscr{M}$, all teams T over $\mathscr{M}$ and all formulae ϕ, if ϕ is satisfied in $\mathscr{M}$ on T, then it is satisfied on any subset T' of T. We also say that in this case, ϕ is downards closed.

Lemma 2. *Let ϕ be a downwards closed* MDL *formula. Then the formula $\boxdot\phi$ is logically equivalent to $\Box\phi$.*

Proof. Let ϕ be a downwards closed MDL formula, $\mathscr{M}$ be a Kripke model, T be team over $\mathscr{M}$ and $\boxdot\phi$ be satisfied in $\mathscr{M}$ on T. By definition ϕ has to be satisfied on all successor teams in $R\langle T\rangle$, especially on $R(T)$. Clearly all teams T' in $R\langle T\rangle$ are subsets of the team of all successors $R(T)$. Because of this and the fact that ϕ is downwards closed, it is sufficient to check if ϕ evaluates to true on $R(T)$. □

Theorem 2. *Let $[B] = \mathrm{N}$, then* MDL-MC$(B, \{\Diamond, \mathrm{dep}(\cdot)\})$ *is* $\mathrm{P}^{\mathrm{NP}[1]}$*-complete.*

Proof. Let ϕ be a $(B, \{\Diamond, \mathrm{dep}(\cdot)\})$-formula and $k_1, \ldots, k_n \in \mathbb{N}$. Then ϕ is always either of the form $\phi := \neg\phi'$ or $\phi := \phi'$, where

$$\phi' := \Diamond^{k_1}\boxdot^{k_2} \ldots \boxdot^{k_n}\lambda, \quad \lambda \text{ is a literal or a dependence atom.}$$

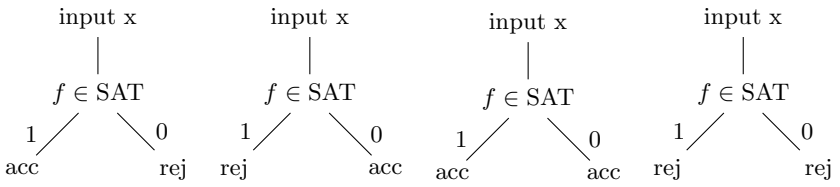
Let $\hat{\phi}$ be any sub formula of ϕ'. Then it follows from Lemma 2, that $\boxdot\hat{\phi}$ can be replaced by $\Box\hat{\phi}$. Hence we can rewrite ϕ' as

$$\phi' := \Diamond^{k_1}\Box^{k_2} \ldots \Box^{k_n}\lambda, \quad \lambda \text{ is a literal or a dependence atom.}$$

Thus, model checking for ϕ can be reduced clearly to model checking for ϕ'. Since Ebbing and Lohmann showed in [4] that MDL model checking for the operator fragment $\{\Diamond, \Box, \mathrm{dep}(\cdot), \bar{\ }\}$ is NP-complete, we conclude MDL-MC$(\mathrm{N}, \{\Diamond, \mathrm{dep}(\cdot)\})$ is in $\mathrm{P}^{\mathrm{NP}[1]}$.

It remains to show that MDL-MC($\{\mathrm{N}, \Diamond, \mathrm{dep}(\cdot)\}$) is $\mathrm{P}^{\mathrm{NP}[1]}$-hard. Let $A \in \mathrm{P}^{\mathrm{NP}[1]}$, and let the corresponding Turing-Machine be M_A. We have to show that $A \leq_m^p$ MDL-MC($\mathrm{N}, \{\Diamond, \mathrm{dep}(\cdot)\}$).

In the polynomial many-one reduction, we can simulate the polynomial part of the machine. Therefore the only thing that is left, is the oracle question and four possible acceptance behaviours of M_A as shown in Figure 3.



(a) $f \in \mathrm{SAT}$ (b) $f \in \overline{\mathrm{SAT}}$ (c) Accept always (d) Reject always

Fig. 3. Acceptance cases in $\mathrm{P}^{\mathrm{NP}[1]}$

SAT is represented in MDL-MC($\mathrm{N}, \{\Diamond, \mathrm{dep}(\cdot)\}$) in the same way it is represented by Ebbing and Lohmann in [4], but we have to adjust our formula to represent the four possible acceptance cases.

Let $\psi := \bigwedge_i C_i$ be the SAT oracle question and $g(\psi) = \langle \mathscr{M}, T, \phi \rangle$ the reduction function.

The Kripke structure $\mathscr{M} = (W, R, \pi)$ is defined as follows:

$$W := \{c_1, \ldots, c_n, s_1, \ldots, s_m, \overline{s}_1, \ldots, \overline{s}_m\},$$

$$R \supseteq \begin{cases} \{(c_i, s_j)\} & \text{, if } x_j \text{ occurs in } C_i \\ \{(c_i, \overline{s}_j)\} & \text{, if } \overline{x}_j \text{ occurs in } C_i, \end{cases}$$

$$\pi(s_i) \supseteq \{p_i, q\}, \qquad \pi(\overline{s}_i) \supseteq \{p_i\}.$$

The initial team is defined by the worlds, which representing the clauses of ψ.

$$T := \{c_1, \ldots, c_n\},$$

$$\phi := \begin{cases} \Diamond \mathrm{dep}(p_1, \ldots, p_m, q) & \text{, } \psi \in \mathrm{SAT} \\ \neg \Diamond \mathrm{dep}(p_1, \ldots, p_m, q) & \text{, } \psi \in \overline{\mathrm{SAT}} \\ \mathrm{t} & \text{, accept always} \\ \mathrm{f} & \text{, reject always.} \end{cases}$$

The correctness follows directly from the MDL-MC($\mathrm{N}, \{\Diamond, \mathrm{dep}(\cdot)\}$) correctness proof in [4] and the definition of $\neg$. □

Theorem 3. *Let* $[B] \supseteq \mathrm{L}$*, then* MDL-MC($B, \{\Diamond, \mathrm{dep}(\cdot)\}$) *is* PSPACE*-complete.*

Proof. To prove hardness, we give a reduction from the standard PSPACE-complete problem QBF, the validity problem for quantified Boolean formulae, to MDL-MC($\{\oplus\}, \{\Diamond, \mathrm{dep}(\cdot)\}$).

Let $\phi := \exists x_1 \forall x_2 \ldots \exists x_n (C_i)$ be a QBF formula. The QBF formula will be transformed to a MDL-MC instance $\langle \mathscr{M} = (W, R, \pi), T, \psi \rangle$ as follows. For each quantified variable we will construct one connected component. In these connected component the nesting of the variables will be simulated by *value states* $x_i, \overline{x}_i$ and *delay states* d_i. There are also components for each clause.

$$W := \bigcup_{i=1}^{n} \left(\{x_i, \overline{x}_i\} \cup \{x_i^j, \overline{x}_i^j | i \le j \le n\} \ \cup \{d_i^j | 1 \le j \le i\} \right) \\ \cup \ \{c_i^j | 1 \le i \le m, \ 1 \le j \le n+1\}$$

For the quantified variable x_i the variables value decision will be made at point i. At the decision point the natural ordering of delay states will branch in a natural ordering of the different values states.

$$R := \bigcup_{i=1}^{n} \Bigg(\{(x_i^j, x_i^{j+1}) | 1 \le j < n-i\} \ \cup \{(\overline{x}_i^j, \overline{x}_i^{j+1}) | i \le j \le n\} \cup \\ \{(x_i^j, x_i) | j = n-i\} \quad \cup \ \{(\overline{x}_i^j, \overline{x}_i) | j = n-i\} \cup \\ \{(d_i^j, d_i^{j+1}) | 1 \le j < i\} \cup \ \{(d_i^i, x_i^{i+1}), (d_i^i, \overline{x}_i^{i+1})\} \cup \\ \{(c_i^j, c_i^{j+1}) | 1 \le i \le n\} \Bigg) \cup \{(c_i, x_i | 1 \le i \le m, \ x_i \in C_i\}$$

The starting team is the set of all initial delay nodes and the initial clause nodes.

$$T := \{d_i^0 | 1 \le i \le n\} \cup \{c_i^1 | 1 \le i \le m\}$$

At last we have to define the labelling of the Kripke structure. For each positive value world $x_i^j, \overline{x}_i^j, \overline{x}_i, x_i$ p_i is labelled to represent the variable and to represent the value on each positive value node q is labelled also.

$$\pi^{-1}(p_i) := \{x_i^j, x_i, \overline{x}_i^j, \overline{x}_i | 1 \le i \le n, i \le j \le n\} \\ \pi^{-1}(q) := \{x_i^j, x_i | 1 \le i \le n, i \le j \le n\}$$

The following formula ψ will simulate the QBF evaluation on the given Kripke model $\mathscr{M} = (W, R, \pi)$ over the team starting team T.

$$\psi := \underbrace{\Diamond \boxdot \ldots \Diamond}_{\text{n-times}} \underbrace{(\neg \mathrm{dep}(p_2, p_4, \ldots, p_{n-1}, q) \oplus \Diamond \mathrm{dep}(p_1, \ldots, p_n, q))}_{\psi'}$$

Let ϕ' be a CNF over the variables $x_1, \ldots, x_n$ and $\phi = \exists x_1 \forall x_2 \ldots \exists x_n \phi'$ be a satisfied QBF formula. Then $\mathscr{M}, T \models \psi$. After n modal quantification steps, the formula ψ' will be evaluated on teams T' over $\{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\}$, because in the i's modal step we pick one or both variable vertices in the connected

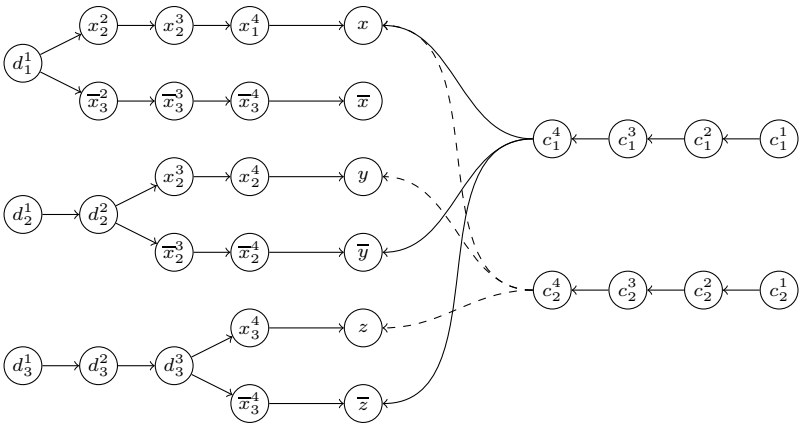


Fig. 4. Kripke Model for QBF formula $\phi := \exists x \forall y \exists z (x \vee \overline{y} \vee \overline{z}) \wedge (x \vee y \vee z)$

component corresponding to the i's variable. For convenience we say that a team is consistent if is does not contain a variable positively and negatively and not consistent if it does. In the following we want to choose satisfying teams with respect to the QBF assignment tree and show that these teams satisfy ψ'. In the case of existential quantification we can choose the variable path with respect to the QBF assignment, but for the universal quantification we have to ensure that the case of both variable assignments are picked, does not falsify ψ'.

Claim (1). Let T' be a team over $\{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\}$, where the universal quantified variable x_i is contained positively and negatively. Then $\mathscr{M}, T' \models \psi'$ holds.

To prove the claim, let x_i and $\overline{x}_i$ be in T'. Then $\neg\mathrm{dep}(p_2, p_4, \ldots, p_{n-1}, q)$ is true, because T' does not satisfy the dependence atom. By this it follows that $\Diamond\mathrm{dep}(\boldsymbol{p}, q)$ will also be false, because $\{p_2, p_4, \ldots, p_{n-1}\} \subseteq \{p_1, \ldots, p_n\}$ and the modal operator does not shrink the team T'. This proves the claim.

With this claim in mind, we construct all consistent successor teams of the form $T' = \bigcup_{i=1}^{n}\{t_i^n\} \cup \bigcup_{i=1}^{m}\{c_i^k\}$, which are representing the assignments of a QBF assignment tree. In the last claim we show that a consistent team evaluated to true on ψ' if and only if the corresponding variable assignments satisfies ϕ'.

Hardness now follows from the following claim.

Claim (2). Let T be of the form $\bigcup_{i=1}^{n}\{t_i^n\} \cup \bigcup_{i=1}^{m}\{c_i^k\}$, where $t_i \in \{x_i^n, \overline{x}_i^n\}$ and $\alpha(x_i) = \begin{cases} 1 & , t_i = \{x_i^n\} \\ 0 & , \text{otherwise} \end{cases}$. Then $\mathscr{M}, T \models \psi'$ if and only if $\alpha \models \phi'$.

We prove the claim. Because of team T is being consistent the sub formula $\neg\mathrm{dep}(p_2, p_4, \ldots, p_{n-1}, q)$ is false. It remains to show that $\Diamond\mathrm{dep}(p_1, \ldots, p_n, q)$ is true if and only if $\alpha \models \phi'$ holds and false otherwise. Let $\alpha \models \phi'$. Then each clause C_j is satisfied by at least one variable assignment in α. W.lo.g. let C_j be satisfied by $x_i = 0$. By definition it follows that C_j is connected with the vertex

$\overline{x}_i$ and by $\alpha(x_i) = 0$ that $\overline{x}_i^n \in T$. Then by consistency of T and α it directly follows that a consistent successor team T' with $\overline{x}_i \in R\langle T\rangle$ and $x_i \notin R\langle T\rangle$ exists.

Finally, MDL-MC$(B, \{\Diamond, \mathrm{dep}(\cdot)\}) \in$ PSPACE follows from Algorithm 2 evaluating the formula in the obvious way.

Input : $\mathrm{MDL_{BF}}$ formula ϕ, Kripke model $\mathscr{M} = (W, R, \pi)$ and team $T \subseteq W$
Output: Is ϕ is satisfied in $\mathscr{M}$ on T?
if $\phi = f(\phi_1, \ldots, \phi_n))$ **then**
  return f(**check** $(\mathscr{M}, T, \phi_1), \ldots,$ **check**$(\mathscr{M}, T, \phi_n)$)
else if $\phi = p$ **then**
  foreach $w_i \in T$ **do**
    if $p \notin \pi(w_i)$ **then return** false
  end
  return true
else if $\phi = \overline{p}$ **then**
  foreach $w_i \in T$ **do**
    if $p \in \pi(w_i)$ **then return** false
  end
  return true
else if $\phi = \mathrm{dep}(p_1, \ldots, p_n, q)$ **then**
  forall the p_i ***in*** $\{p_1, \ldots, p_n\}$ **do**
    forall the w_{t_j} ***in*** $\{w_{t_1}, \ldots, w_{t_m}\} = T$ **do**
      $c_{ij} = \mathrm{valid}(\mathscr{M}, \{w_{t_j}\}, p_i)$
    end
  end
  for $w_{t_i} \in T$ **do**
    for $w_{t_j} \in T$ **do**
      if $c_i = c_j$ **then**
        if $\mathrm{valid}(\mathscr{M}, \{w_{t_i}\}, q) \neq \mathrm{valid}(\mathscr{M}, \{w_{t_j}\}, q)$ **then**
          return false
        end
      end
    end
  end
  return true
else if $\phi = \Diamond\phi'$ **then**
  guess existentially $T' \in R\langle T\rangle$
  return check $(\mathscr{M}, T', \phi')$
else if $\phi = \Box\phi'$ **then**
  return check $(\mathscr{M}, R(T), \phi')$
end

Algorithm 2. PSPACE algorithm **check**$(\mathscr{M}, T, \phi)$

□

From results presented by Ebbing and Lohmann in [4], in which the NP-completeness of some particular fragments for modal dependence logic model checking was shown, it follows that if a set B of Boolean connectives forms a base for one of the Boolean clones $\mathrm{ID}, \mathrm{E}, \mathrm{V}, \mathrm{M}$, it yields an NP-complete model checking problem.

Together with Theorem 2 and 3 above, we thus obtain a complete picture for the complexity of model checking for modal dependence logic, as given in Fig. 2. For any set B of Boolean connectives, the complexity falls in one of the cases given there.

4 Conclusion

We obtained a complete classification of the complexity of the model checking problem for modal dependence logic for formulas that may contain dependence atoms, one or two modalities, and symbols for arbitrary Boolean functions.

What we did not address here is formulas that besides the arbitrary sets of Boolean connectives also involve dependence connectives, e.g., splitjunction, intuitionistic implication, linear implication, etc. While partial results, at least for the first two mentioned connectives, are known, a full classification is still missing. For the case of splitjunction combined with the diamond modality it is known that the classification shown in Figure 2 is still valid expect for the N case. In this case no result is known. The classification of splitjucntion combined with the $\Box$ modality misses the cases L and N.

Even more interesting is maybe the question how to develop a general concept of what a dependence connective is, and then study complexity issues concerning formulas with arbitrary sets of dependence connectives, maybe via a similar lattice as Post's. First steps into the direction of a concept of such general connectives have been made by Antti Kuusisto (personal communication).

References

1. Bauland, M., Mundhenk, M., Schneider, T., Schnoor, H., Schnoor, I., Vollmer, H.: The tractability of model checking for LTL: The good, the bad, and the ugly fragments. ACM Trans. Comput. Log. 12(2), 26 (2011)
2. Böhler, E., Creignou, N., Reith, S., Vollmer, H.: Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory. SIGACT News 34(4), 38–52 (2003)
3. Buss, S.R.: The boolean formula value problem is in alogtime. In: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, pp. 123–131 (1987)
4. Ebbing, J., Lohmann, P.: Complexity of model checking for modal dependence logic. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) SOFSEM 2012. LNCS, vol. 7147, pp. 226–237. Springer, Heidelberg (2012)
5. Fischer, M.J., Ladner, R.E.: Propositional dynamic logic of regular programs. Journal of Computer and Systems Sciences 18(2), 194–211 (1979)
6. Hemaspaandra, E., Schnoor, H., Schnoor, I.: Generalized modal satisfiability. J. Comput. Syst. Sci. 76(7), 561–578 (2010)
7. Ladner, R.E.: The computational complexity of provability in systems of modal propositional logic. SIAM J. Comput. 6(3), 467–480 (1977)
8. Lewis, H.R.: Satisfiability problems for propositional calculi. Mathematical Systems Theory 13, 45–53 (1979)

9. Lohmann, P., Vollmer, H.: Complexity results for modal dependence logic. In: Dawar, A., Veith, H. (eds.) CSL 2010. LNCS, vol. 6247, pp. 411–425. Springer, Heidelberg (2010)
10. Post, E.: The two-valued iterative systems of mathematical logic. Annals of Mathematical Studies 5, 1–122 (1941)
11. Sevenster, M.: Model-theoretic and computational properties of modal dependence logic. Journal of Logic and Computation 19(6), 1157–1173 (2009), http://logcom.oxfordjournals.org/cgi/content/abstract/exn102v1
12. Thomas, M.: On the applicability of Post's lattice. Inf. Process. Lett. 112(10), 386–391 (2012)
13. Väänänen, J.: Modal dependence logic. In: Apt, K.R., van Rooij, R. (eds.) New Perspectives on Games and Interaction, Texts in Logic and Games, vol. 4, pp. 237–254. Amsterdam University Press (2008)

Ockhamist Propositional Dynamic Logic: A Natural Link between PDL and CTL*

Philippe Balbiani and Emiliano Lorini

Université de Toulouse, IRIT-CNRS, France

Abstract. We present a new logic called Ockhamist Propositional Dynamic Logic, OPDL, which provides a natural link between PDL and CTL*. We show that both PDL and CTL* can be polynomially embedded into OPDL in a rather simple and direct way. More generally, the semantics on which OPDL is based provides a unifying framework for making the dynamic logic family and the temporal logic family converge in a single logical framework. Decidability of the satisfiability problem for OPDL is studied in the paper.

1 Introduction

Different logical systems are traditionally used in theoretical computer science and in artificial intelligence for the verification of programs and for modelling reactive systems and multi-agent systems. Among them we should mention Propositional Dynamic Logic PDL [12], Propositional Linear Temporal Logic PLTL [20], Computation Tree Logic CTL [11], Full Computation Tree Logic CTL* [22] and Alternating-time Temporal Logic ATL [1]. Some relationships between these different logical systems have been studied. For instance, it is well-known that PLTL and CTL are fragments of CTL* and that CTL is a fragment of ATL [13]. However, at the current stage, the general picture remains incomplete. For example, it is clear (and well-known) that the logic of programs PDL can express properties that Full Computation Tree Logic CTL* cannot and vice-versa. Moreover, there are no clear relationships between PDL and logics of strategic reasoning such as ATL. More precisely, it is not known whether there exists natural embeddings of PDL into ATL or of ATL in PDL. Even more importantly, there is still no logical system that can be said to be more general than the others. For instance, there is no logic that embeds in a *natural* and *simple* way both PDL and CTL*. Indeed, although there exist some logics that embed both PDL and CTL*, they do it in a rather complicated and unnatural way. For example, it is well-known that PDL and CTL* can be embedded in modal μ-calculus. However, although the embedding of PDL into modal μ-calculus is simple and direct, the embedding of CTL* into modal μ-calculus is rather complicated and doubly exponential in the length of the input formula [7]. Another logic that links PDL with CTL* is the extension of PDL with a repetition construct (PDL-Δ) by [26]. But again, the embedding of CTL* into PDL-Δ too

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 251–265, 2013.

is rather complicated and doubly exponential in the length of the input formula [29].[1] For this reason, a challenge arises of making the previous competing logical systems converge into a single logical system. The aim of this paper is to make a step into this direction by proposing an Ockhamist variant of Propositional Dynamic Logic, OPDL, that provides a natural link between PDL and CTL*. Specifically, we show that both PDL and CTL* can be polynomially embedded into OPDL in a rather simple and direct way. More generally, the Ockhamist semantics on which OPDL is based provides a unifying framework for making the dynamic logic family and the temporal logic family converge into a single logical framework. Ockhamist semantics for temporal logic have been widely studied in the 80ies and in the 90ies [27,30,5]. The logic of agency STIT (the logic of "seeing to it that") by Belnap *et al.* [4] is based on such semantics. According to the Ockhamist conception of time (also called *indeterminist actualist*, see [30]) the truth of statements is evaluated with respect to a moment and to a particular *actual* linear history passing through that moment, and the temporal operators are relativized to the actual history of the evaluation.

The rest of the paper is organized as follows. We first present the syntax and the semantics of OPDL and provide a decidability result for this logic (Section 2). Then, we discuss, in Section 3, about the relationship of OPDL with PDL and CTL* (Section 3). In particular, we provide polynomial reductions of PDL and CTL* to OPDL. In Section 4 we present a variant of OPDL whose semantics is based on the notion of labeled transition system (LTS). In Section 5 we conclude by discussing some perspectives for future work.[2]

2 Ockhamist Propositional Dynamic Logic

The distinction between the 'Ockhamist' semantics and the 'Peircean' semantics for branching-time temporal logic was proposed by Prior in his seminal work on the logic of time [21] (see also [27]). According to the 'Peircean' view the truth of a temporal formula should be evaluated with respect either to some history or all histories starting in a given state. In the 'Ockhamist' semantics for branching time a notion of *actual* course of events is given. In particular, according to the 'Ockhamist' view, the truth of a temporal formula should be evaluated with respect to a particular *actual* history starting in a given state. While the branching-time temporal logic CTL* is compatible with the Ockhamist conception of time, the semantics for PDL in terms of labelled transition systems is closer to the Peircean view than to the Ockhamist view since it does not consider a notion of actual history or actual path in a transition system. The logic OPDL can be conceived as a variant of the logic of programs PDL based on the

[1] It is worth noting that Axelsson et al. [2] have recently studied generic extensions of CTL in which temporal operators are parameterized with different kinds of formal languages recognized by different classes of automata (*e.g.*, regular languages, visibly pushdown and context-free languages). They compare the expressive power of these extensions of PDL to CTL, PDL and extensions of PDL such as PDL-Δ. However, they also show that CTL* cannot be embedded in any of these extensions of CTL, as the property of fairness is expressible in CTL* but is not expressible is any of these logics (see [2, Theorem 4.3]).

[2] An extended version of this paper containing detailed proofs is available at [3].

Ockhamist view of time. Specifically, OPDL is a variant of PDL in which the truth of a formula is evaluated with respect to a given actual history. The syntax and the semantics of this logic are presented in Sections 2.1 and 2.2.

2.1 Syntax

Assume a countable set $Prop$ of atomic propositions (with typical members denoted $p, q, \ldots$) and a countable set Atm of atomic programs (or atomic actions) (with typical members denoted $a, b, \ldots$). Let $2^{Atm*} = 2^{Atm} \setminus \{\emptyset\}$. The language $\mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ of OPDL consists of a set Prg of programs and a set Fml of formulae. It is defined as follows:

$$\begin{array}{l} Prg: \ \pi ::= a \mid \equiv \mid (\pi_1; \pi_2) \mid (\pi_1 \cup \pi_2) \mid \pi^* \mid \varphi? \\ Fml: \varphi ::= p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid [\![\pi]\!]\varphi \end{array}$$

where p ranges over $Prop$ and a ranges over Atm. We adopt the standard definitions for the remaining Boolean operations. The dual $\langle\!\langle \pi \rangle\!\rangle$ of the operator $[\![\pi]\!]$ is defined in the expected way: $\langle\!\langle \pi \rangle\!\rangle\varphi \stackrel{\mathtt{def}}{=} \neg[\![\pi]\!]\neg\varphi$. We follow the usual rules for omission of the parentheses. Given a formula φ, let $FL(\varphi)$ denote its Fischer-Ladner closure. See [12, Chapter 6] for details. It is a well-known fact that $card(FL(\varphi))$ is linear in the length of φ.

Complex programs of sequential composition ($\pi_1; \pi_2$), non-deterministic choice ($\pi_1 \cup \pi_2$), iteration (π^*) and test ($\varphi?$) are built from atomic programs in Atm, from the special program $\equiv$ and from formulae in Fml. The special program $\equiv$ allows to move from a history to an alternative history passing through the same moment. The behavior of this program will become clearer in Section 2.2 when presenting the OPDL semantics.

The formula $[\![\pi]\!]\varphi$ has to be read "φ will be true at the end of all possible executions of program π" whereas $\langle\!\langle \pi \rangle\!\rangle\varphi$ has to be read "φ will be true at the end of some possible execution of program π". As it is assumed that atomic programs in Atm are linear (*i.e.*, all atomic programs in Atm occurring at a given state lead to the same successor state), $[\![a]\!]\varphi$ can also be read "if the atomic program a occurs, φ will be true afterwards". Indeed, from the assumption of linearity, it follows that atomic programs in Atm are deterministic (*i.e.*, there is at most one possible execution of an atomic program a at a given state). Finally, the formula $[\![\equiv]\!]\varphi$ has to be read "φ is true in all histories passing through the current moment" or, more shortly, "φ is necessarily true in the current moment".

2.2 Semantics

OPDL frames are structures with two dimensions: a vertical dimension corresponding to the concept of history, a horizontal dimension corresponding to the concept of moment.

Definition 1 (OPDL frame). *An OPDL frame is a tuple $F = (W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_{\equiv})$ where:*

- *W is a nonempty set of states (or worlds),*
- *$\mathcal{Q}$ is a partial function $\mathcal{Q} : W \longrightarrow W$,*

- *$\mathcal{L}$ is a mapping $\mathcal{L} : \mathcal{Z} \longrightarrow 2^{Atm*}$ from state transitions to non-empty sets of atomic programs, $\mathcal{Z} = \{(w, v) \mid w, v \in W$ and $\mathcal{Q}(w) = v\}$ being the transition relation induced by the successor state function $\mathcal{Q}$,*
- *$\mathcal{R}_{\equiv} \subseteq W \times W$ is an equivalence relation between states in W such that for all $w, v, u \in W$:*
  (C1) *if $\mathcal{Q}(w) = v$ and $(v, u) \in \mathcal{R}_{\equiv}$ then there is $z \in W$ such that $(w, z) \in \mathcal{R}_{\equiv}$ and $\mathcal{Q}(z) = u$ and $\mathcal{L}(z, u) = \mathcal{L}(w, v)$.*

For every $w, v \in W$, $\mathcal{Q}(w) = v$ means that v is the successor state of w. If $\mathcal{Q}(w) = v$ then we also say that w is a predecessor of v. If $\mathcal{L}(w, v) = \{a, b\}$, then the actions a and b are responsible for the transition from the state w to the state v. In other words, the function $\mathcal{L}$ labels every state transition with a set of atomic actions (*viz.* the actions that are responsible for the transition). The assumption that the set $\mathcal{L}(w, v)$ should be non-empty means that every state transition is due to the execution of at least one atomic action.

$\mathcal{R}_{\equiv}$-equivalence classes are called *moments*. If w and v belong to the same moment then they are called *alternatives*. A maximal sequence of states according to the transition relation $\mathcal{Z}$ starting at a given state w is called *history starting in* w. If w and v belong to the same moment, then the history starting in w and the history starting in v are alternative histories (*viz.* histories starting at the same moment).

Constraint (**C1**) corresponds to what in Ockhamist semantics is called property of *weak diagram completion* [30]. This means that if two worlds v and u are in the same moment and world w is a predecessor of v then, there exists a world z such that (i) w and z are in the same moment, (ii) u is the successor of z, (iii) the transition from w to v and the transition from z to u are labeled with the same set of action names.

Figure 1 is an example of OPDL frame. The $\mathcal{R}_{\equiv}$-equivalences classes $\{w_1, w_2, w_3, w_4\}$, $\{w_5, w_6\}$, $\{w_7, w_8\}$, $\{w_9\}$, $\{w_{10}\}$, $\{w_{11}\}$, $\{w_{12}\}$, $\{w_{13}\}$, $\{w_{14}\}$, $\{w_{15}\}$ and $\{w_{16}\}$ are the moments. The sequences of states (w_1, w_5, w_9, w_{13}), $(w_2, w_6, w_{10}, w_{14})$, $(w_3, w_7, w_{11}, w_{15})$ and $(w_4, w_8, w_{12}, w_{16})$ are the alternative histories starting at the same moment $\{w_1, w_2, w_3, w_4\}$. Actions a and c are responsible for the transition from the state w_1 to the state w_5 and, because of Constraint (**C1**), actions a and c are also responsible for the transition from the state w_2 to the state w_6. Moreover, actions b and c are responsible for the transition from the state w_3 to the state w_7 and, because of Constraint (**C1**), actions b and c are also responsible for the transition from the state w_4 to the state w_8.

Definition 2 (Atomic transitions). *Given an OPDL frame $F = (W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_{\equiv})$ and an atomic program $a \in Atm$, let*

$$\mathcal{R}_a = \{(w, v) \mid \mathcal{Q}(w) = v \text{ and } a \in \mathcal{L}(w, v)\}$$

be the set of a-transitions in the frame F.

An OPDL model is an OPDL frame supplemented with a valuation function mapping each state to the set of propositional atoms which are true in it, under the assumption that two states belonging to the same moment agree on the atoms. More precisely:

Definition 3 (OPDL model). *An OPDL model is a tuple $M = (W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_{\equiv}, \mathcal{V})$ where:*

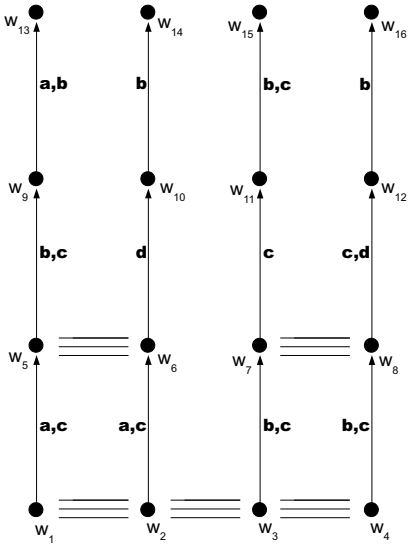


Fig. 1. An OPDL frame

- $(W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_{\equiv})$ *is a* OPDL *frame and*
- $\mathcal{V} : W \longrightarrow 2^{Prop}$ *is a valuation function for atomic propositions such that for all* $w, v \in W$*:*
  (C2) *if* $(w, v) \in \mathcal{R}_{\equiv}$ *then* $\mathcal{V}(w) = \mathcal{V}(v)$.

The truth of a OPDL formula is evaluated with respect to a world w in an OPDL model M.

Definition 4 (π-transitions and truth conditions). *Let* $M = (W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_{\equiv}, \mathcal{V})$ *be an* OPDL *model. Given a program* π*, let us define a binary relation* $\mathcal{R}_\pi$ *on* W *with* $(w, v) \in \mathcal{R}_\pi$ *(or* $w\mathcal{R}_\pi v$*) meaning that* v *is accessible from* w *by performing program* π*. Let us also define a binary relation* $\models$ *between worlds in* M *and formulae with* $M, w \models \varphi$ *meaning that formula* φ *is true at* w *in* M*. The rules inductively defining* $\mathcal{R}_\pi$ *and* $\models$ *are:*

$$\begin{aligned}
\mathcal{R}_{\pi_1;\pi_2} &= \mathcal{R}_{\pi_1} \circ \mathcal{R}_{\pi_2} \\
\mathcal{R}_{\pi_1 \cup \pi_2} &= \mathcal{R}_{\pi_1} \cup \mathcal{R}_{\pi_2} \\
\mathcal{R}_{\pi^*} &= (\mathcal{R}_\pi)^* \\
\mathcal{R}_{\varphi?} &= \{(w, w) \mid w \in W \text{ and } M, w \models \varphi\}
\end{aligned}$$

and

$$\begin{aligned}
M, w \models p &\iff p \in \mathcal{V}(w); \\
M, w \models \neg\varphi &\iff M, w \not\models \varphi; \\
M, w \models \varphi \wedge \psi &\iff M, w \models \varphi \text{ AND } M, w \models \psi; \\
M, w \models [\![\pi]\!]\varphi &\iff \forall v \in \mathcal{R}_\pi(w) : M, v \models \varphi
\end{aligned}$$

with $\mathcal{R}_\pi(w) = \{v \in W \mid (w,v) \in \mathcal{R}_\pi\}$.

An OPDL formula φ is said to be OPDL valid, denoted by $\models_{\mathsf{OPDL}} \varphi$, if and only if φ is true in all OPDL models (*i.e.*, for every OPDL model M and for every world w in M, we have $M, w \models \varphi$). An OPDL formula φ is said to be OPDL satisfiable if and only if $\neg\varphi$ is not OPDL valid.

OPDL formulae can also be interpreted over standard Kripke structures.

Definition 5 (Kripke OPDL model). *A Kripke OPDL model is a tuple* $M = (W, \{\mathcal{R}_a \mid a \in Atm\}, \mathcal{R}_\equiv, \mathcal{V})$ *where:*

- W *is a set of states (or worlds),*
- $\mathcal{R}_\equiv$ *is an equivalence relation on* W *and all* $\mathcal{R}_a$ *are binary relations on* W *satisfying the following two constraints for all* $w, v, u \in W$*:*
  *(**C1***)* *if* $(w,v) \in \mathcal{R}_\mathsf{X}$ *and* $(w,u) \in \mathcal{R}_\mathsf{X}$ *then* $u = v$,
  *(**C2***)* *if* $(w,v) \in \mathcal{R}_\mathsf{X}$ *and* $(v,u) \in \mathcal{R}_\equiv$ *then there is* $z \in W$ *such that* $(w,z) \in \mathcal{R}_\equiv$ *and for all* $a \in Atm$, $(w,v) \in \mathcal{R}_a$ *if and only if* $(z,u) \in \mathcal{R}_a$,
  with $\mathcal{R}_\mathsf{X} = \bigcup_{a \in Atm} \mathcal{R}_a$,
- $\mathcal{V} : W \longrightarrow 2^{Prop}$ *is a valuation function for atomic propositions such that for all* $w, v \in W$*:*
  *(**C3***)* *if* $(w,v) \in \mathcal{R}_\equiv$ *then* $\mathcal{V}(w) = \mathcal{V}(v)$.

Constraints (**C2***) and (**C3***) are respectively the counterparts of Constraint (**C1**) and Constraint (**C2**) in the definition of an OPDL model. Constraint (**C1***) forces the successor relation $\mathcal{R}_\mathsf{X}$ to be deterministic (*i.e.*, every state has at most one successor).

As stated by the following proposition, the notion of satisfiability with respect to the class of OPDL models is equivalent to the notion of satisfiability with respect to the class of Kripke OPDL models.

Proposition 1. *Let* φ *be an OPDL formula. Then, there exists an OPDL model which satisfies* φ *if and only if there exists a Kripke OPDL model which satisfies* φ.

We shall say that φ is a global logical consequence of a finite set of global axioms $\Gamma = \{\chi_1, \ldots, \chi_n\}$, denoted by $\Gamma \models_{\mathsf{OPDL}} \varphi$, if and only if for every OPDL model M, if Γ is true in M (*i.e.*, for every world w in M, we have $M, w \models \chi_1 \wedge \ldots \wedge \chi_n$) then φ is true in M too (*i.e.*, for every world w in M, we have $M, w \models \varphi$).

As the following proposition highlights, when the set of atomic programs Atm is finite, the problem of logical consequence in OPDL with a finite set of global axioms is reducible to the validity problem for OPDL formulae.

Proposition 2. *Let* $\Gamma = \{\chi_1, \ldots, \chi_n\}$ *be a finite set of OPDL formulae. If* Atm *is finite,* $\Gamma \models_{OPDL} \varphi$ *if and only if* $\models_{OPDL} [\![\mathbf{any}^*]\!](\chi_1 \wedge \ldots \wedge \chi_n) \rightarrow \varphi$ *with* $\mathbf{any} \stackrel{\mathtt{def}}{=} (\bigcup_{a \in Atm} \cup \equiv)$.

The model checking problem for OPDL is the following decision problem: given a finite OPDL model M and an OPDL formula φ, is there a world in M such that $M, w \models \varphi$? With finite OPDL model, we mean a OPDL model $M = (W, \mathcal{Q}, \mathcal{L}, \mathcal{R}_\equiv, \mathcal{V})$ that satisfies the following three conditions: (1) W is *finite*; (2) $\mathcal{L}$ associates to every transition $(w,v) \in \mathcal{Z} = \{(w,v) \mid w, v \in W \text{ and } \mathcal{Q}(w) = v\}$ a non-empty *finite* set of atomic

actions in Atm; (3) $\mathcal{V}$ associates to every world $w \in W$ a *finite* set of atomic formulas in $Prop$. In order to determine whether there exists a world w in M such that $M, w \models \varphi$, we can use the model checking algorithm for PDL showing that the model checking problem for PDL is PTIME-complete with respect to the size of the input model and the input formula. It follows that the model checking problem for OPDL is PTIME-complete too with respect to $size(M) + size(\varphi)$.

2.3 Decidability of OPDL

Using the "mosaic method", a technique used in algebraic logic [18] to prove the decidability of equational theories, we will prove the decidability of SAT, the following decision problem: determine whether a given OPDL formula φ is satisfiable with respect to the class of OPDL models.

Theorem 1. *SAT is decidable.*

Proof (Sketch). Let φ be a formula. In order to simplify the proof, we assume that at most one atomic action, namely a, occurs in φ. A type for φ is a subset t of $FL(\varphi)$. It is normal iff it satisfies the conditions of atomicity considered in [14, Definition 2.2]. A group for φ is a finite set G of normal types for φ. A mosaic for φ is a finite set M of groups for φ. It is normal iff it satisfies the following conditions: **(i)** if $p \in FL(\varphi)$ then for all $G \in M$, for all $t \in G$, if $p \in t$ then for all $H \in M$, for all $u \in H$, $p \in u$; **(ii)** if $[\![\equiv]\!]\psi \in FL(\varphi)$ then for all $G \in M$, for all $t \in G$, if $[\![\equiv]\!]\psi \in t$ then for all $H \in M$, for all $u \in H$, $\psi \in u$; **(iii)** if $\neg[\![\equiv]\!]\psi \in FL(\varphi)$ then for all $G \in M$, for all $t \in G$, if $\neg[\![\equiv]\!]\psi \in t$ then there exists $H \in M$, there exists $u \in H$ such that $\neg\psi \in u$. A system for φ is a finite set S of normal mosaics for φ. A context for S is a structure of the form (M, G, t) where $M \in S$, $G \in M$ and $t \in G$. Obviously, there exists finitely many types, groups, mosaics and systems for φ. Since the normality conditions for types and mosaics are decidable, the set of all contexts can be computed. Let $\Sigma = \{a, \equiv\} \cup \{\psi?\colon \psi \in FL(\varphi)\}$. For all $\alpha \in \Sigma$, we define the transition relation $\longrightarrow^S_\alpha$ between contexts for S as follows: $(M, G, t) \longrightarrow^S_\alpha (N, H, u)$ iff one of the following conditions is satisfied: **(i)** $\alpha = a$ and there exists a bijection $f : G \to N$ such that **(a)** $f(t) = u$, **(b)** if $[\![a]\!]\psi \in FL(\varphi)$ then for all $v \in G$, if $[\![a]\!]\psi \in v$ then $\psi \in f(v)$, **(c)** if $\neg[\![a]\!]\psi \in FL(\varphi)$ then for all $v \in G$, if $\neg[\![a]\!]\psi \in v$ then $\neg\psi \in f(v)$; **(ii)** $\alpha = \equiv$ and $M = N$; **(iii)** $\alpha = \psi?$, $M = N$, $G = H$, $t = u$ and $\psi \in u$. For all programs π, we inductively define the transition relation $\longrightarrow^S_\pi$ between contexts for S as follows: $\longrightarrow^S_{\pi_1;\pi_2} = \longrightarrow^S_{\pi_1} \circ \longrightarrow^S_{\pi_2}$, $\longrightarrow^S_{\pi_1 \cup \pi_2} = \longrightarrow^S_{\pi_1} \cup \longrightarrow^S_{\pi_2}$, $\longrightarrow^S_{\pi^\star} = (\longrightarrow^S_\pi)^\star$. Since the set of all contexts can be computed, the transition relations $\longrightarrow^S_\pi$ are all decidable. A system S for φ is said to be saturated iff it satisfies the following condition: if $\neg[\![\pi]\!]\psi \in FL(\varphi)$ then for all contexts (M, G, t) for S, if $\neg[\![\pi]\!]\psi \in t$ then there exists a context (N, H, u) for S such that $(M, G, t) \longrightarrow^S_\pi (N, H, u)$ and $\neg\psi \in u$. Since the transition relations $\longrightarrow^S_\pi$ are all decidable, checking the saturation of a given system for φ is decidable. The proof of the decidability of SAT proceeds in two steps. First, in Proposition 3, we prove that φ is satisfiable iff there exists a saturated system S for φ and there exists a context (M, G, t) in S such that $\varphi \in t$. Second, in Proposition 4, we prove the decidability of the decision problem SYS defined as follows: determine whether there exists a saturated system S for a given formula φ and there exists a context (M, G, t) in S such that $\varphi \in t$.

Proposition 3. *Let φ be a formula. The following conditions are equivalent:* **(i)** *φ is satisfiable;* **(ii)** *there exists a saturated system S for φ, there exists a context (M, G, t) in S such that $\varphi \in t$.*

Proposition 4. *SYS is decidable.*

As a result, SAT is decidable. □

3 Relationships between OPDL, PDL and CTL*

In this section we study the relationships between OPDL and PDL, and between OPDL and CTL*. In particular, we provide a polynomial embedding of PDL into OPDL and a polynomial embedding of CTL* into OPDL.

3.1 Relationships between OPDL and PDL

Propositional Dynamic Logic PDL [15] is the well-known logic of programs. Again assume the countable set of atomic propositions $Prop = \{p, q, \ldots\}$ and the countable set of atomic programs $Atm = \{a, b, \ldots\}$. The language $\mathcal{L}_{\mathsf{PDL}}(Prop, Atm)$ of PDL is defined by the following grammar in Backus-Naur Form (BNF):

$$\begin{array}{l} Prg: \ \pi ::= a \mid (\pi_1; \pi_2) \mid (\pi_1 \cup \pi_2) \mid \pi^* \mid \varphi? \\ Fml: \varphi ::= p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid [\pi]\varphi \end{array}$$

where p ranges over $Prop$ and a ranges over Atm.

PDL models are nothing but labeled transition systems, *i.e.*, transition systems where transitions between states are labeled with atomic programs.

Definition 6. *PDL models are tuples $M = (W, \{\mathcal{R}_a \mid a \in Atm\}, \mathcal{V})$ where:*

- *W is a nonempty set of possible worlds or states;*
- *$\{\mathcal{R}_a \mid a \in Atm\}$ is a set of binary relations on W;*
- *$\mathcal{V} : W \longrightarrow 2^{Prop}$ is a valuation function.*

The accessibility relations for atomic programs are generalized to complex programs in the usual way (see Definition 2).

The truth conditions of PDL formulae are standard for the Boolean constructions plus the following one for the dynamic operators $[\pi]$:

$$M, w \models [\pi]\varphi \Longleftrightarrow \forall v \in \mathcal{R}_\pi(w) : M, v \models \varphi$$

A PDL formula φ is said to be PDL valid if and only if φ is true in all PDL models.

We can embed PDL in OPDL. Consider the following polynomial translation $tr_1 : \mathcal{L}_{\mathsf{PDL}}(Prop, Atm) \longrightarrow \mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ from the language of PDL to the OPDL language.

$$\begin{array}{ll} tr_1(p) & = p \text{ for all } p \in Prop \\ tr_1(\neg\varphi) & = \neg tr_1(\varphi) \\ tr_1(\varphi_1 \wedge \varphi_2) & = tr_1(\varphi_1) \wedge tr_1(\varphi_2) \\ tr_1([\pi]\varphi) & = [\![tr_2(\pi)]\!] tr_1(\varphi) \end{array}$$

where

$$\begin{array}{ll} tr_2(a) & = \equiv; a \text{ for all } a \in Atm \\ tr_2(\pi_1;\pi_2) & = tr_2(\pi_1); tr_2(\pi_2) \\ tr_2(\pi_1 \cup \pi_2) & = tr_2(\pi_1) \cup tr_2(\pi_2) \\ tr_2(\pi^*) & = (tr_2(\pi))^* \\ tr_2(\varphi?) & = tr_1(\varphi)? \end{array}$$

As the following theorem shows, the preceding translation is a correct embedding.

Theorem 2. *Let φ be a PDL formula. φ is PDL valid if and only if $tr_1(\varphi)$ is OPDL valid.*

3.2 Relationships between OPDL and CTL*

Full Computation Tree Logic CTL* was first described in [10,9] as an extension of Computation Tree Logic CTL [6] and Propositional Linear Temporal Logic PLTL [20]. The language of CTL* is built recursively from the atomic propositions using the temporal operators of PLTL, and the existential path switching operator of CTL as well as classical connectives.

Again assume the countable set of atomic propositions $Prop = \{p, q, \ldots\}$. The language $\mathcal{L}_{\mathsf{CTL}^*}(Prop)$ of CTL* is defined by the following grammar in Backus-Naur Form (BNF):

$$\varphi ::= p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \mathsf{X}\varphi \mid \varphi\, \mathcal{U}\, \psi \mid \mathsf{A}\varphi$$

where p ranges over $Prop$. The constructs X and $\mathcal{U}$ are respectively the operators *next* and *until* of PLTL, the formulas $\mathsf{X}\varphi$ and $\varphi\, \mathcal{U}\, \psi$ being respectively read "φ will be true in the next state along the current path" and "ψ will be true at some point in the future along the current path and φ has to hold until ψ". These two operators can be used to express other kinds of temporal notions such as *eventually* $\mathsf{F}\psi \stackrel{\mathtt{def}}{=} \top\, \mathcal{U}\, \psi$, *henceforth* $\mathsf{G}\psi \stackrel{\mathtt{def}}{=} \neg\mathsf{F}\neg\psi$ and *before* $\varphi\, \mathcal{B}\, \psi \stackrel{\mathtt{def}}{=} \neg(\neg\varphi\, \mathcal{U}\, \psi)$. The construct A is a modal operator quantifying over possible paths, the formula $\mathsf{A}\varphi$ being read "φ is true in all possible paths". The existential path-quantifier operator E, is defined by $\mathsf{E}\varphi \stackrel{\mathtt{def}}{=} \neg\mathsf{A}\neg\varphi$.

Different semantics for CTL* have been given in the literature. One of this semantics is based on the notion of Ockhamist structure. Here we mainly follow the presentation of the Ockhamist semantics for CTL* given by Reynolds [22] who introduces a special kind of Ockhamist structures called $(\mathbb{N} \times W)$ structures.

Definition 7. *A $(\mathbb{N} \times W)$ structure is a tuple $(W, \sim, g)$ where:*

- *W is a set of points;*
- *$\sim$ is an equivalence relation over $\mathbb{N} \times W$ such that for all $w, v \in W$ and for all $n, m \in \mathbb{N}$:*
  - ***(S1)*** *if $(n, w) \sim (m, v)$ then $n = m$,*
  - ***(S2)*** *if $(n, w) \sim (n, v)$ and $m < n$ then $(m, w) \sim (m, v)$,*
  - ***(S3)*** *$(0, w) \sim (0, v)$;*

– $g : \mathbb{N} \times W \longrightarrow 2^{Prop}$ *is a valuation function mapping each integer and each point into a set of atoms such that for all* $w, v \in W$ *and for all* $n \in \mathbb{N}$*:*
(S4) *if* $(n, w) \sim (n, v)$ *then* $g(n, w) = g(n, v)$.

Given a $(\mathbb{N} \times W)$ structure $(W, \sim, g)$ and a CTL^* formula φ, $(W, \sim, g), (n, w) \models \varphi$ means that φ is true at the index (n, w) in the $(\mathbb{N} \times W)$ structure $(W, \sim, g)$. The rules defining the truth conditions of CTL^* formulae are inductively defined as follows:

$$
\begin{aligned}
(W, \sim, g), (n, w) \models p &\Longleftrightarrow p \in g(n, w);\\
(W, \sim, g), (n, w) \models \neg\varphi &\Longleftrightarrow (W, \sim, g), (n, w) \not\models \varphi;\\
(W, \sim, g), (n, w) \models \varphi_1 \wedge \varphi_2 &\Longleftrightarrow (W, \sim, g), (n, w) \models \varphi_1 \text{ AND } (W, \sim, g), (n, w) \models \varphi_2;\\
(W, \sim, g), (n, w) \models \mathsf{X}\varphi &\Longleftrightarrow (W, \sim, g), (n + 1, w) \models \varphi\\
(W, \sim, g), (n, w) \models \varphi\, \mathcal{U}\, \psi &\Longleftrightarrow \exists m \in \mathbb{N} : m \geq n \text{ AND } (W, \sim, g), (m, w) \models \psi \text{ AND}\\
&\qquad\qquad \forall k \in \mathbb{N} : \text{IF } n \leq k < m \text{ THEN } (W, \sim, g), (k, w) \models \varphi\\
(W, \sim, g), (n, w) \models \mathsf{A}\varphi &\Longleftrightarrow \forall v \in W : \text{IF } (n, w) \sim (n, v) \text{ THEN } (W, \sim, g), (n, v) \models \varphi
\end{aligned}
$$

As shown by Reynolds [22] the CTL^* semantics in terms of $(\mathbb{N} \times W)$ structures is equivalent to the CTL^* semantics in terms of *bundled trees*. However, it is more general than the common CTL^* semantics in terms of $\mathcal{R}$*-generable models* used by [9], *i.e.*, Kripke structures with states, a total accessibility relation $\mathcal{R}$ between them and the set of all paths which arise by moving from state to state along the accessibility relation. The difference between the CTL^* semantics in terms of bundled trees and the CTL^* semantics in terms of R-generable models is that the latter quantifies over all paths induced by the relation R whereas the former quantifies over a bundle of paths. Although this bundle is suffix and fusion closed, it does not need to be be limit closed. For example, it may be the case that all paths include a right branch even though at every world there is a path where the next branch goes left, which violates the limit closure property. In order to distinguish full computation tree logic interpreted over R-generable models and full computation tree logic interpreted over bundled trees (and equivalently over $(\mathbb{N} \times W)$ structures), some authors (see, *e.g.*, [23,16,17]) use the term CTL^* to indicate the former logic and the term BCTL^* (bundled CTL^*) to indicate the latter (in [25] it is called $\forall LTFC$).

It is well-known that CTL^* interpreted over R-generable models is 2-EXPTIME complete: [10] provides a doubly exponential automaton based satisfiability checker, and [28] gives a lowerbound. As pointed by [17], an argument for the 2-EXPTIME hardness of the satisfiability problem could also be made for CTL^* interpreted over bundled trees in a way similar to the argument for CTL^* interpreted over R-generable models. Therefore, as the CTL^* semantics in terms of bundled trees is equivalent to the CTL^* semantics in terms of $(\mathbb{N} \times W)$ structures [22], it follows that the satisfiability problem for CTL^* interpreted over $(\mathbb{N} \times W)$ structures is also 2-EXPTIME hard. Therefore, CTL^* interpreted over bundled trees (or over $(\mathbb{N} \times W)$ structures) is not easier to deal with than CTL^* interpreted over R-generable models. However, one interesting aspect of the former kind of CTL^* is that it is closely connected to Ockhamist temporal logics studied in the field of philosophical logic [30]. Moreover, one might argue that reasoning in BCTL^* is relatively easier than reasoning in CTL^*. For example the specification for the tableau method for BCTL^* proposed by [23] was much simpler than the CTL^* tableau that originated from it [24].

Consider the following translation $tr_3 : \mathcal{L}_{\mathsf{CTL}^*}(Prop) \longrightarrow \mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ from the language of CTL* to the OPDL language where x is an arbitrary atomic program in Atm:

$$\begin{array}{ll} tr_3(p) & = p \text{ for all } p \in Prop \\ tr_3(\neg\varphi) & = \neg tr_3(\varphi) \\ tr_3(\varphi_1 \wedge \varphi_2) & = tr_3(\varphi_1) \wedge tr_3(\varphi_2) \\ tr_3(\mathsf{X}\varphi) & = \langle\!\langle x \rangle\!\rangle tr_3(\varphi) \\ tr_3(\varphi\, \mathcal{U}\, \psi) & = \langle\!\langle (tr_3(\varphi)?; x)^* \rangle\!\rangle tr_3(\psi) \\ tr_3(\mathsf{A}\varphi) & = [\![\equiv]\!] tr_3(\varphi) \end{array}$$

The preceding translation is polynomial and, as the following theorem shows, it provides an embedding of the variant of CTL* interpreted over $(\mathbb{N} \times W)$ structures into OPDL.

Theorem 3. *Let φ be a CTL* formula. φ is valid with respect to the class of $(\mathbb{N} \times W)$ structures if and only if $\{\langle\!\langle x \rangle\!\rangle \top\} \models_{\mathsf{OPDL}} tr_3(\varphi)$.*

From Theorem 3 and Proposition 2 in Section 2.2, it follows that the satisfiability problem in the variant of CTL* interpreted over $(\mathbb{N} \times W)$ structures can be reduced to the satisfiability problem in OPDL with a finite number of atomic programs.

Corollary 1. *Let φ be a CTL* formula and let Atm be finite. φ is valid with respect to the class of $(\mathbb{N} \times W)$ structures if and only if $\models_{\mathsf{OPDL}} [\![\mathbf{any}^*]\!] \langle\!\langle x \rangle\!\rangle \top \rightarrow tr_3(\varphi)$.*

Since the satisfiability problem of $(\mathbb{N} \times W)$ structures is 2-EXPTIME-hard (see above), the preceding polynomial embedding of CTL* into OPDL provides an argument for the 2-EXPTIME-hardness of the satisfiability problem of our logic OPDL.

3.3 Relationships with Other Logics: Discussion

The logic OPDL has interesting connections with other logical systems proposed in the field of theoretical computer science such as propositional linear time temporal logic PLTL and Nishimura's combination of PDL and PLTL (call it, NL) [19]. As for PLTL, in the previous Section 3.2 we have provided a polynomial embedding of CTL* into OPDL. As PLTL is nothing but the fragment of CTL* without the path quantifier operator A, the translation tr_3 also provides a polynomial embedding of PLTL into OPDL. As for NL, we just need to put together the translation tr_1 from PDL to OPDL given in Section 3.1 and the translation from PLTL to OPDL in order to provide a polynomial reduction of Nishimura's logic NL to OPDL. Another logic that is related with OPDL is ACTL*, the action based version of CTL* proposed by [8]. ACTL* extends CTL* with temporal operators X_a indexed by atomic programs a in the set of atomic programs Atm. The ACTL* formula $\mathsf{X}_a\varphi$ has to be read "the next transition is labeled with the atomic program a and φ will be true in the next state along the current path". By adding the following item to the preceding translation tr_3 from CTL* to OPDL, we get a polynomial embedding of ACTL* into OPDL:

$$tr_3(\mathsf{X}_a\varphi) = \langle\!\langle a \rangle\!\rangle tr_3(\varphi) \text{ for all } a \in Atm$$

4 A Variant of OPDL Interpreted over Labeled Transition Systems

As we have shown in Section 3.2, the logic OPDL interpreted over OPDL models (Definition 3) embeds the variant of CTL* interpreted over $(\mathbb{N} \times W)$ structures which in turn is equivalent to the variant of CTL* interpreted over bundled trees.

A second variant of CTL*, first introduced by [9], is the one interpreted over $\mathcal{R}$-generable models of the form $M = (W, \mathcal{R}, \mathcal{V})$ where W is a set of states, $\mathcal{R} \subseteq W \times W$ is a *total* binary relation on W (*i.e.*, for every $w \in W$, there is some $v \in W$ such that $(w, v) \in \mathcal{R}$) and $\mathcal{V} : W \longrightarrow 2^{Prop}$ is a valuation function for atomic propositions.[3]

Given a model $M = (W, \mathcal{R}, \mathcal{V})$, a *fullpath* in M is defined to be an infinite sequence $(w_1, w_2, w_3, \ldots)$ of states of M such that for each $i \geq 1$, $(w_i, w_{i+1}) \in \mathcal{R}$. Given a fullpath $\sigma = (w_1, w_2, w_3, \ldots)$ and an integer $i \geq 1$, the symbol $\sigma_{\geq i}$ denotes the fullpath $(w_i, w_{i+1}, \ldots)$. As usual, $\sigma[1]$ denotes the first element of the sequence σ. Truth of a CTL* formula is evaluated with respect to a $\mathcal{R}$-generable model M and a fullpath σ in M. Specifically, the rules defining the truth conditions of CTL* formulae are inductively defined as follows:

$$
\begin{aligned}
M, \sigma \models p &\Longleftrightarrow p \in \mathcal{V}(\sigma[1]);\\
M, \sigma \models \neg\varphi &\Longleftrightarrow M, \sigma \not\models \varphi;\\
M, \sigma \models \varphi_1 \wedge \varphi_2 &\Longleftrightarrow M, \sigma \models \varphi_1 \text{ AND } M, \sigma \models \varphi_2;\\
M, \sigma \models \mathsf{X}\varphi &\Longleftrightarrow M, \sigma_{\geq 2} \models \varphi\\
M, \sigma \models \varphi\,\mathcal{U}\,\psi &\Longleftrightarrow \exists i \geq 1 : M, \sigma_{\geq i} \models \psi \text{ AND } \forall j : \text{IF } 1 \leq j < i \text{ THEN } M, \sigma_{\geq j} \models \varphi\\
M, \sigma \models \mathsf{A}\varphi &\Longleftrightarrow \forall \sigma' : \text{IF } \sigma[1] = \sigma'[1] \text{ THEN } M, \sigma' \models \varphi
\end{aligned}
$$

Here we consider a variant of OPDL which embeds the preceding variant of CTL* interpreted over $\mathcal{R}$-generable models. We call OPDL^{lts} this variant of OPDL, where OPDL^{lts} means 'OPDL interpreted over labeled transition systems'. The semantics for OPDL^{lts} is given in terms of PDL models as defined in Section 3.1 (Definition 6), which are nothing but labeled transition systems, *i.e.*, transition systems where transitions between states are labeled with atomic programs. Given a PDL model $M = (W, \{\mathcal{R}_a \mid a \in Atm\}, \mathcal{V})$, let the successor state function $succ$ be defined by $succ(w) = \bigcup_{a \in Atm} \{v \in W \mid (w, v) \in \mathcal{R}_a\}$ for each $w \in W$. $succ(w)$ identifies the successors of world w in M. Moreover, for every $w \in W$, let

$$PA = \{(w_1, \ldots, w_n) \mid w_1, \ldots, w_n \in W \text{ and } w_{i+1} = succ(w_i) \text{ for all } 1 \leq i < n\}$$

be the set of all *paths* in M. For every $w \in W$, let MPA_w be the set of all maximal paths starting in w, also called *histories starting in* w. That is, $\sigma \in MPA_w$ if and only if $\sigma \in PA$ and $\sigma[1] = w$ and there is no $\sigma' \in PA$ such that $\sigma'[1] = w$ and $\sigma \sqsubset \sigma'$ (*i.e.*, σ is a proper initial subsequence of σ'). Finally, let $IN = \{w/\sigma \mid w \in W \text{ and } \sigma \in MPA_w\}$ be the set of all *indexes* in the model M.

[3] It has been proved that the variant of CTL* interpreted over $(\mathbb{N} \times W)$ structures is more general than the variant of CTL* interpreted $\mathcal{R}$-generable models, in the sense that the former have less validities than the latter. For instance, as shown by [22], the formula $\mathsf{AG}(p \rightarrow \mathsf{EX}p) \rightarrow (p \rightarrow \mathsf{EG}p)$ is valid in the latter variant of CTL* but is not valid in the former.

Truth of an OPDL formula is evaluated at a given index $w/\sigma \in IN$ of a PDL model M. The rules inductively defining the truth conditions of OPDL formulae are:

$$\begin{aligned} M, w/\sigma \models p &\iff p \in \mathcal{V}(w); \\ M, w/\sigma \models \neg\varphi &\iff M, w/\sigma \not\models \varphi; \\ M, w/\sigma \models \varphi \wedge \psi &\iff M, w/\sigma \models \varphi \text{ AND } M, w/\sigma \models \psi; \\ M, w/\sigma \models [\![\pi]\!]\varphi &\iff \forall\, v/\sigma' \in \rho_\pi(w/\sigma) : M, v/\sigma' \models \varphi \end{aligned}$$

where

$$\begin{aligned} \rho_a &= \{(w/\sigma, v/\sigma') \mid w/\sigma, v/\sigma' \in IN, (w,v) \in \mathcal{R}_a \text{ and } \sigma = (w,\sigma')\} \\ \rho_{\equiv} &= \{(w/\sigma, v/\sigma') \mid w/\sigma, v/\sigma' \in IN \text{ and } w = v\} \\ \rho_{\pi_1;\pi_2} &= \rho_{\pi_1} \circ \rho_{\pi_2} \\ \rho_{\pi_1 \cup \pi_2} &= \rho_{\pi_1} \cup \rho_{\pi_2} \\ \rho_{\pi^*} &= (\rho_\pi)^* \\ \rho_{\varphi?} &= \{(w/\sigma, w/\sigma) \mid w/\sigma \in IN \text{ and } M, w/\sigma \models \varphi\} \end{aligned}$$

and $\rho_\pi(w/\sigma) = \{v/\sigma' \mid (w/\sigma, v/\sigma') \in \rho_\pi\}$.

A formula φ of the language $\mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ is said to be OPDL^{lts} valid, denoted by $\models_{\mathsf{OPDL}^{lts}} \varphi$, if and only if φ is true in all PDL models. As usual, a formula φ of the language $\mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ is said to be OPDL^{lts} satisfiable if and only if $\neg\varphi$ is not OPDL^{lts} valid. We shall say that a formula φ of the language $\mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ is a global logical consequence in OPDL^{lts} of a finite set of global axioms $\Gamma = \{\chi_1, \ldots, \chi_n\}$, denoted by $\Gamma \models_{\mathsf{OPDL}^{lts}} \varphi$, if and only if for every PDL model M, if Γ is true in M then φ is true in M too.

As the following theorem shows, the translation given in Section 3.2 provides an embedding of the variant of CTL* interpreted over $\mathcal{R}$-generable models into OPDL^{lts}.

Theorem 4. *Let φ be a CTL* formula. φ is valid with respect to the class of $\mathcal{R}$-generable models if and only if $\{\langle\!\langle x \rangle\!\rangle \top\} \models_{\mathsf{OPDL}^{lts}} tr_3(\varphi)$.*

From Theorem 4 and the fact that, as in OPDL, the problem of global logical consequence in OPDL^{lts} with a finite number of global axioms is reducible to the problem of OPDL^{lts} validity, it follows that the satisfiability problem in the variant of CTL* interpreted over $\mathcal{R}$-generable models can be reduced to the satisfiability problem in OPDL^{lts} with a finite number of atomic programs.

Corollary 2. *Let φ be a CTL* formula and let Atm be finite. φ is valid with respect to the class of $\mathcal{R}$-generable models if and only if $\models_{\mathsf{OPDL}^{lts}} [\![\mathbf{any}^*]\!]\langle\!\langle x \rangle\!\rangle \top \rightarrow tr_3(\varphi)$.*

5 Perspectives

We have presented a new logic called Ockhamist Propositional Dynamic Logic OPDL and studied its relationship with PDL and CTL*. An interesting issue for future research is the study of the relationship between OPDL and PDL with intersections of programs. Intersections of *atomic* programs can be simulated in OPDL as follows:

$$[\![a \cap b]\!]\varphi \stackrel{\text{def}}{=} [\![\equiv]\!](\langle\!\langle a \rangle\!\rangle \top \rightarrow [\![b]\!]\varphi)$$

However, it is not clear whether we can find a simple translation from PDL with intersection of (not necessarily atomic) programs to OPDL that preserves validity.

Another direction of future research is the study of the exact complexity of the satisfiability problem for OPDL. The embedding of CTL* into OPDL ensures that it is 2-EXPTIME hard. However, the construction based on the "mosaic method" given in the Section 2.3 does not provide an optimal decision procedure for OPDL. Future work will be devoted to find an optimal decision procedure for OPDL showing that its satisfiability problem is in 2-EXPTIME. Indeed, at the current stage, we conjecture that CTL* is not easier to deal with than OPDL. We also plan to find a sound and complete axiomatization for the logic OPDL.

As to the logic OPDL^{lts} whose semantics has been sketched in Section 4, much work remains to be done. First of all, we plan to study more in detail the differences between OPDL and OPDL^{lts}, taking inspiration from Reynolds' work [22] on the comparison between the CTL* semantics in terms of $\mathcal{R}$-generable models and the CTL* semantics in terms of $(\mathbb{N} \times W)$ structures (or bundled trees). For instance, we plan to find some interesting examples of formulae of the language $\mathcal{L}_{\mathsf{OPDL}}(Prop, Atm)$ which are valid in OPDL^{lts} but are not valid in OPDL. Secondly, we plan to adapt the proof of the decidability of the satisfiability problem for OPDL given in the Section 2.3 in order to prove the decidability of the satisfiability problem for OPDL^{lts}. Another aspect of the logic OPDL^{lts} that we plan to investigate in the future is its relationship with the extension of PDL with a repetition construct (PDL-Δ) by [26]. PDL-Δ extends PDL with constructions of the form $\Delta\pi$ meaning that "the program π can be repeatedly executed infinitely many times". We believe that the construction $\Delta\pi$ can be simulated in OPDL^{lts} as follows:

$$\Delta\pi \stackrel{\mathtt{def}}{=} \langle\!\langle \equiv \rangle\!\rangle [\![\pi^*]\!] \langle\!\langle \pi \rangle\!\rangle \top$$

We postpone to future work the definition of the exact translation from PDL-Δ formulae to OPDL^{lts} formulae and a theorem stating that, for every PDL-Δ formula φ, φ is PDL-Δ valid if and only if the translation of φ in OPDL^{lts} is OPDL^{lts} valid.

References

1. Alur, R., Henzinger, T., Kupferman, O.: Alternating-time temporal logic. Journal of the ACM 49, 672–713 (2002)
2. Axelsson, R., Hague, M., Kreutzer, S., Lange, M., Latte, M.: Extended computation tree logic. In: Fermüller, C.G., Voronkov, A. (eds.) LPAR-17. LNCS, vol. 6397, pp. 67–81. Springer, Heidelberg (2010)
3. Balbiani, P., Lorini, E.: Ockhamist propositional dynamic logic: a natural link between PDL and CTL*. Technical Report IRIT/RT–2013-12–FR, Institut de Recherche en Informatique de Toulouse (2013)
4. Belnap, N., Perloff, M., Xu, M.: Facing the future: agents and choices in our indeterminist world. Oxford University Press (2001)
5. Brown, M., Goranko, V.: An extended branching-time ockhamist temporal logic. Journal of Logic, Language and Information 8(2), 143–166 (1999)
6. Clarke, E.M., Emerson, E.A.: Design and synthesis of synchronization skeletons using branching time temporal logic. In: Kozen, D. (ed.) Logic of Programs 1981. LNCS, vol. 131, pp. 52–71. Springer, Heidelberg (1982)

7. Dam, M.: CTL* and ECTL* as fragments of the modal mu-calculus. Theoretical Computer Science 126(1), 77–96 (1994)
8. De Nicola, R., Vaandrager, F.W.: Action versus state based logics for transition systems. In: Guessarian, I. (ed.) LITP 1990. LNCS, vol. 469, pp. 407–419. Springer, Heidelberg (1990)
9. Emerson, E.A., Halpern, J.: 'Sometimes' and 'not never' revisited: on branching versus linear time. Journal of the ACM 33, 151–178 (1986)
10. Emerson, E.A., Sistla, A.: Deciding full branching time logic. Information and Control 61, 175–201 (1984)
11. Emerson, E.A.: Temporal and modal logic. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science. Formal Models and Semantics, vol. B. North-Holland Pub. Co./MIT Press (1990)
12. Fischer, M.J., Ladner, R.E.: Propositional dynamic logic of regular programs. Journal of Computer Systems Science 18(2), 194–211 (1979)
13. Goranko, V.: Coalition games and alternating temporal logics. In: Proc. of TARK 2001, pp. 259–272. Morgan Kaufmann (2001)
14. Harel, D.: Dynamic logic. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. 2, pp. 497–604. Reidel, Dordrecht (1984)
15. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press (2000)
16. Masini, A., Viganò, L., Volpe, M.: Labelled natural deduction for a bundled branching temporal logic. Journal of Logic and Computation 21(6), 1093–1163 (2011)
17. McCabe-Dansted, J.C.: A tableau for the combination of CTL and BCTL. In: Proc. of TIME 2012, pp. 29–36. IEEE Computer Society (2012)
18. Németi, I.: Decidable versions of first order logic and cylindric-relativized set algebras. In: Csirmaz, L., Gabbay, D., de Rijke, M. (eds.) Logic Colloquium 1992, pp. 171–241. CSLI Publications (1995)
19. Nishimura, H.: Descriptively complete process logic. Acta Informatica 14, 359–369 (1980)
20. Pnueli, A.: The temporal logic of programs. In: Proc. of the Eighteenth Symposium on Foundations of Computer Science, pp. 46–57. IEEE Computer Society (1977)
21. Prior, A.: Past, Present, and Future. Clarendon Press, Oxford (1967)
22. Reynolds, M.: An axiomatization of full computation tree logic. Journal of Symbolic Logic 66(3), 1011–1057 (2001)
23. Reynolds, M.: A tableau for bundled CTL*. Journal of Logic and Computation 17(1), 117–132 (2007)
24. Reynolds, M.: A tableau for CTL*. In: Cavalcanti, A., Dams, D.R. (eds.) FM 2009. LNCS, vol. 5850, pp. 403–418. Springer, Heidelberg (2009)
25. Stirling, C.: Modal and temporal logics. In: Abramsky, S., Gabbay, D.M., Maibaum, T.S.E. (eds.) Handbook of Logic in Computer Science, vol. 2. Clarendon Press, Oxford (1992)
26. Streett, R.S.: Propositional dynamic logic of looping and converse is elementarily decidable. Information and Control 54(1-2), 121–141 (1982)
27. Thomason, R.: Combinations of tense and modality. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. 2, pp. 135–165. Reidel, Dordrecht (1984)
28. Vardi, M.Y., Stockmeyer, L.J.: Improved upper and lower bounds for modal logics of programs. In: Proc. of the 17th Annual ACM Symposium on Theory of Computing, pp. 240–251. ACM (1985)
29. Wolper, P.: A translation from full branching time temporal logic to one letter propositional dynamic logic with looping. Unpublished manuscript (1982)
30. Zanardo, A.: Branching-time logic with quantification over branches: The point of view of modal logic. Journal of Symbolic Logic 61(1), 143–166 (1996)

Information, Awareness and Substructural Logics

Igor Sedlár

Faculty of Arts, Comenius University in Bratislava
Gondova 2, 814 99 Bratislava, Slovak Republic
sedlar@fphil.uniba.sk

Abstract. The paper outlines a generalisation of the awareness-based epistemic semantics by Fagin and Halpern. Awareness is construed as a relation between agents and pieces of information instead of formulas. The main motive for introducing the generalisation is that it shows substructural logics to be a natural component of information-based epistemic logic: substructural logics can be seen as describing the logical behaviour of pieces of information. Substructural epistemic logics are introduced and some of their properties are discussed. In addition, extensions of substructural epistemic logics invoking group-epistemic and dynamic modalities are sketched.

Keywords: Awareness, Epistemic Actions, Epistemic Logic, Information, Substructural Logic.

1 Introduction

The present paper provides a generalisation of the awareness-based epistemic framework by Fagin and Halpern [17]: awareness is construed as a relation between agents and *pieces of information*. The main motive for introducing the generalisation is that it connects epistemic logics with *substructural logics* [29,32]. It is shown that the latter are a natural component of information-based epistemic logics: substructural logics can be seen as describing the logical behaviour of pieces of information.

The generalisation provides a framework for studying a large class of *substructural epistemic logics*. The paper is an introductory overview of the framework, focusing on a general discussion of substructural epistemic logics. Consequently, many standard investigations of specific logics (such as completeness proofs) are postponed to a sequel. We note that our approach owes much to justification logics [3,4,5,6] and to the Fitting semantics [19,20] in particular. While being similar in some respects, our approach and Fitting semantics are motivated by different goals: we are aiming solely at explaining the possible applications of substructural logics within epistemic logic.

The paper is organised as follows. Section 2 outlines the awareness-based framework [17]. Section 3 suggests a generalisation of the framework: 'pieces of information' are discussed explicitly and awareness is construed as pertaining to

L. Libkin, U. Kohlenbach, and R. de Queiroz (Eds.): WoLLIC 2013, LNCS 8071, pp. 266–281, 2013.

these. Building on the relational semantics for distributive substructural logics, Sect. 4 explains that the logics can be seen as natural 'logics of information'. Section 5 expands on the above observations and introduces substructural epistemic logics. Section 6 outlines an information-based generalisation of public announcements and discusses some of its idiosyncrasies. Section 7 concludes the paper and outlines some interesting directions for future work. Proofs of some of the propositions are given in a technical appendix.

2 Logical Omniscience and Awareness

It has been argued since the inception of epistemic logic that the modal-logic-based approach[1] is rather optimistic as to the agents' epistemic abilities. If 'α believes that F' is equivalent to 'F holds in *every* α*-accessible* alternative' and

$$F_1 \wedge \ldots \wedge F_n \rightarrow G \tag{1}$$

holds in *every* alternative, then if α believes $F_1, \ldots, F_n$, then she is bound to believe G as well.[2] This is rather optimistic indeed. There are no logical reasons why α should 'realise' that (1) holds in every alternative and adjust her beliefs accordingly. The problem is known as the *logical omniscience problem*.[3]

This is a conceptual issue: the notion of belief embodied in the modal-logic-based epistemic logics is rather specific and it does not conform to many intuitions associated with our use of the word 'belief'.[4] The intuitions are numerous, unclear, and perhaps mutually inconsistent. However, one may try to explicate one's intuitions in a little more detail and provide appropriate formalisations. This has been done by many, which led to a number of sophisticated contributions to epistemic logic.[5]

One may argue that the 'pre-theoretical notion of belief' includes a crucial element ignored by the modal-logic-based approach, namely the agent's *active attitudes* towards the believed proposition. Thus, Fagin and Halpern [17] couple the true-in-every-alternative condition with α's *awareness* of the believed formula. Formally, an *awareness model* is a quadruple

$$M = \langle W, R, A, V \rangle \tag{2}$$

where W is the set of alternatives (or 'possible worlds'), R is a binary relation on W (the 'accessibility' relation) and V is a valuation. The crucial element is

[1] This approach originates in Hintikka's classic [21]. For a more recent overview, see [10], for example.

[2] To ensure that or discussion is as general as possible, we shall be using the notion of belief throughout the paper. However, some contexts will allow us to use the stronger 'true belief' and even 'knowledge'.

[3] The term has been coined by Hintikka, see [22]. For more details on the problem, see [18, Ch. 9].

[4] Hintikka [21] concludes his discussion of the problem in a similar vein. For a more recent incarnation of the idea, see [10].

[5] See [18, Ch. 9] for an overview.

A, a function from W to sets of formulas. Informally, $F \in A(w)$ means that α is aware of F at w. 'α believes that F' holds at a world $w \in W$ iff i) Rwv implies that F holds at v (α 'implicitly' believes that F) and, importantly, ii) $F \in A(w)$. The syntactic nature of the awareness function makes it clear that one may avoid the logical omniscience problem for every instance of (1). Since $A(w)$ may be arbitrary, it is always possible to construct a pointed model M, w such that 'α believes that F_i' holds at M, w for every $1 \leq i \leq n$, but $G \notin A(w)$.

However, one can argue that the syntactic flavour of the approach is, in fact, a shortcoming. The syntactical nature of the awareness function can be seen as depriving the approach from the capacity to offer a deeper *internal* justification of the respective failures of omniscience. One may strive for a logic of non-omniscient belief where the properties of belief are arrived at by using a subtler 'inner semantic mechanism'.

3 Awareness and Information

This section introduces the core notions of the paper. First, the information-based generalisation of the awareness framework is discussed on an intuitive level (3.1). After that, information models and related technical notions are defined (3.2).

3.1 Awareness Generalised

Note that propositions, represented by formulas, can be seen as a special case of *pieces of information*. A piece of information can be tentatively characterised as everything that can *corroborate* (give support to) a proposition.[6]

Example 1. Assume that during a murder trial the jury is shown a video of the defendant entering the victim's home around the established time of death and carrying something that could be the murder weapon. *The video itself* can be seen as a piece of information that corroborates the proposition 'The defendant is guilty'. The prosecution can be said to have made the jury aware of this piece of information by introducing it during the trial. A statement 'The defendant threatened to kill the victim on numerous occasions' of a witness is another possible piece of information that corroborates the same proposition.

[6] Unfortunately, we do not have space in this paper to provide a satisfactory philosophical analysis of the notions of information and corroboration. However, our being vague about these notions can be justified. First, depending on the academic discipline, distinct notions are associated with the term 'information', see [1]. Second, a similar ambiguity pertains to the often used 'agent' as well. However, our use of 'piece of information' is close to the standard use of 'signal', see [16,31]. 'Corroboration' can be seen as a generalisation of 'carrying' information: we do not adopt Dretske's [16] assumption that if a signal carries information that F, then F is the case. 's corroborates F' can be tentatively seen as being close to 'If accepted by an agent, s is likely to cause the agent's belief that F'.

The picture can be complemented by adding general *corroboration conditions*. For example, it may be stipulated that if a piece of information s corroborates a conjunction $F \wedge G$, then it corroborates both conjuncts F, G. In this manner, pieces of information can be endowed with 'logical character'.

Example 2. A perceptual image of my two hands can be seen as a piece of information that corroborates the proposition 'Both of my hands exist now'. The image then obviously corroborates *both* 'My left hand exists now' and 'My right hand exists now'.

Moreover, taking pieces of information into consideration opens door for considering *relations* on pieces of information in addition to the pieces themselves. These may be, in turn, called upon within corroboration conditions. Examples will be provided later on.

3.2 Information Models

Definition 1 (The basic epistemic language). *Let $\Phi = \{p_1, p_2, \ldots\}$ be a denumerable set of propositional variables and let $\mathcal{G}$ be a non-empty set ('agents'). The set of formulas of the basic epistemic language $L_{\mathcal{G}}(\Phi)$ is given by:*

$$F ::= p \mid \neg F \mid F \wedge F \mid F \vee F \mid F \rightarrow F \mid F \leftrightarrow F \mid \Box_\alpha F \tag{3}$$

where $p \in \Phi$ and $\alpha \in \mathcal{G}$. The set of $L_{\mathcal{G}}$-formulas will be denoted as $Form(L_{\mathcal{G}})$. Formulas $\Box_\alpha F$ are read 'α believes that F'. The Boolean fragment of $L_{\mathcal{G}}$ is the subset of $Form(L_{\mathcal{G}})$ consisting of formulas that do not contain occurrences of $\Box_\alpha$, for any $\alpha \in \mathcal{G}$.

Definition 2 (Information structure and L-structure). *An information structure is a couple*

$$\mathcal{I} = \langle I, \triangle \rangle \tag{4}$$

where I is a non-empty set ('pieces of information') and $\triangle$ is a set of relations on I. Let '$X \subseteq \mathcal{I}$' and '$s \in \mathcal{I}$' be short for '$X \subseteq I \in \mathcal{I}$' and '$s \in I \in \mathcal{I}$', respectively.

Let L be a language. An information L-structure is a couple

$$\mathcal{I}(L) = \langle \mathcal{I}, \Vdash \rangle \tag{5}$$

where $\mathcal{I}$ is an information structure and $\Vdash$ is a subset of $I \times Form(L)$ ('corroboration relation'). We shall use '$s \Vdash F$' instead of '$\langle s, F \rangle \in \Vdash$'.

A familiar special case of information structure are sets of atomic programs together with the program operators, known from propositional dynamic logic [23]. A special case of information L-structure are action models, known from dynamic epistemic logics [7,8].

Definition 3 (Information frames and models). *An information frame for $L_{\mathcal{G}}$ is a tuple*

$$\mathcal{F} = \langle W, R, \mathcal{I}, A \rangle \tag{6}$$

where W is a non-empty set ('possible worlds'), $R : \mathcal{G} \to \mathcal{P}(W \times W)$ is a function from the set of agents to binary relations on W, $\mathcal{I}$ is an information structure, $A : (\mathcal{G} \times W) \to (\mathcal{P}(\mathcal{I}) - \emptyset)$ is a function that assigns to every agent α and possible world w a non-empty set of pieces of information $A(\alpha, w) \subseteq \mathcal{I}$. We shall write '$R_\alpha wv$' instead of '$\langle w, v\rangle \in R(\alpha)$' and R_α shall be referred to as the 'α-accessibility' relation.

An information model for $L_\mathcal{G}$ is a tuple

$$\mathcal{M} = \langle \mathcal{F}, V, \Vdash \rangle \tag{7}$$

where $\mathcal{F}$ is an information frame for $L_\mathcal{G}$, $V : \Phi \to \mathcal{P}(W)$ is a valuation and $\Vdash$ is a corroboration relation.

The α-accessibility relations are interpreted in the usual way (implicit belief). $A(\alpha, w)$ is to be thought of as the set of pieces of information α is aware of at w. Our target notion of belief can be characterised as follows: α believes that F iff α implicitly believes that F and is aware of a piece of information that corroborates F.[7]

It is clear that awareness models are a special case of information models. Just consider information structures where $I = Form(L_\mathcal{G})$ and $\Vdash$ is the identity relation on I. Specific closure principles may be validated by incorporating extra corroboration conditions. For example, closure under $\wedge$-elimination corresponds to the condition:

- If $s \Vdash F \wedge G$, then $s \Vdash F$ and $s \Vdash G$

Definition 4 (Truth conditions). *The truth conditions of the Boolean formulas are as usual and we state only some:*

- $M, w \models p$ *iff* $w \in V(p)$
- $M, w \models \neg F$ *iff* $M, w \not\models F$
- $M, w \models F \wedge G$ *iff* $M, w \models F$ *and* $M, w \models G$

The target notion of belief is formalised in an obvious way:

- $M, w \models \Box_\alpha F$ *iff i)* $R_\alpha wv$ *implies* $M, v \models F$ *and ii) there is a* $s \in A(\alpha, w)$ *such that* $s \Vdash F$

The usual notions of validity in a model, frame and class of frames are assumed.

Note that, to ensure maximal generality, we have not introduced specific corroboration conditions yet. However, Sect. 4 points out that truth conditions in substructural models are natural candidates.

[7] The requirement that the awareness sets be non-empty is a useful idealisation, see the proof of Prop. 3.

4 Information and Substructural Logics

'Pieces of information' have been invoked within informal interpretations of the semantics of many substructural logics. For example, Kripke [24] describes the points of intuitionistic models as 'points in time (or "evidential situations"), at which we may have various pieces of information' [24, p. 98]. Urquhart's interpretation of his semi-lattice semantics for relevant implication [33] invokes 'pieces of information' together with specific operations on them. More recent interpretations [26,28] invoke a related notion of situation. Hence, there is hope that epistemic models where pieces of information are considered explicitly will be a natural 'meeting point' of epistemic and substructural logics.

Definition 5 (Substructural frames and models). *We shall use a slight modification of the standard definitions [29, Ch. 11]. A substructural frame is a tuple*

$$\mathfrak{F} = \langle P, \sqsubseteq, \bullet, C \rangle \tag{8}$$

where P is a non-empty set ('points'), $\sqsubseteq$ is a partial order on P ('informational containment'), $\bullet$ is a binary operation on P ('application') and C is a symmetric binary relation on P ('compatibility'). It is assumed that

- *If Cxy, $x' \sqsubseteq x$ and $y' \sqsubseteq y$, then $Cx'y'$*
- *if $x \bullet y \sqsubseteq z$, $x' \sqsubseteq x$, $y' \sqsubseteq y$ and $z \sqsubseteq z'$, then $x' \bullet y' \sqsubseteq z'$*

A substructural model for $L_{\mathcal{G}}$ is a couple

$$\mathfrak{M} = \langle \mathfrak{F}, \Vdash \rangle \tag{9}$$

where $\mathfrak{F}$ is a frame and $\Vdash$ is a relation between points and members of $Form(L_{\mathcal{G}})$ such that:

- *$x \sqsubseteq y$ and $x \Vdash p$ implies $y \Vdash p$*
- *$x \Vdash \neg F$ iff Cxy implies $y \nVdash F$ for all y*
- *$x \Vdash F \wedge G$ iff $x \Vdash F$ and $x \Vdash G$*
- *$x \Vdash F \vee G$ iff $x \Vdash F$ or $x \Vdash G$*
- *$x \Vdash F \rightarrow G$ iff $y \Vdash F$ and $x \bullet y \sqsubseteq z$ imply $z \Vdash G$, for all y, z*
- *$x \Vdash F \leftrightarrow G$ iff $x \Vdash F \rightarrow G$ and $x \Vdash G \rightarrow F$*

F entails G in $\mathfrak{M}$ iff $x \Vdash F$ implies $x \Vdash G$ for every $x \in \mathfrak{M}$. Entailment in frames and classes of frames is then defined in the usual way.

Substructural frames (models) clearly are a special case of information structures ($L_{\mathcal{G}}$-structures). Substructural frames correspond to a set of pieces of information together with a binary relation $\sqsubseteq$ of informational containment, a relation of compatibility C and an operation of application $\bullet$. Let us discuss these in reverse order.

Example 3. Application corresponds to 'taking two pieces of information together': the two pieces of information taken together can be seen as a 'new' piece of information. For example, two consecutive statements s, t by witnesses during a trial can be seen as a 'complex' piece of information $s \bullet t$, considered by the jury.

Example 4. As an example of two compatible pieces of information, consider a sworn statement of a witness to the effect that the defendant's car was parked somewhere far away from the crime scene around the time of the victim's death (s) and the video from Exam. 1 (t). The two are obviously compatible. Consequently, s does not corroborate the proposition 'The defendant is not guilty', $\neg F$, since it is consistent with t, a piece of information that corroborates F.

Proposition 1. *Let F be a member of the Boolean fragment of $L_{\mathcal{G}}$, $\mathfrak{M}$ a substructural model and x, y points of $\mathfrak{M}$. If $x \Vdash F$ and $x \sqsubseteq y$, then $y \Vdash F$.*

Hence, if x is informationally contained in y, then every Boolean formula that holds at x holds at y as well. The Proposition is a standard result in substructural logic and a simple consequence of Def. 5.

Example 5. Informational containment can be seen as a complex relation: $s \sqsubseteq t$ iff i) s is contained in t *and* ii) every F corroborated by s is corroborated by t as well. For an example of a piece of information contained in a 'larger' piece, consider two fingerprints found on a crime scene. The couple of prints can be seen as a piece of information containing the two single prints. It is plausible to assume automatically that if one of the prints corroborates a proposition (e.g. that a given suspect is guilty), then the couple does so as well. However, this is not plausible in general. For example, consider a sworn statement that the defendant was playing poker at a local casino at the estimated time of the murder (s) and, again, the video from Exam. 1 (t). The jury can take these together, i.e. consider $s \bullet t$. In a sense, both s, t are contained in $s \bullet t$. However, they are not so in the *informational sense*: while s alone can be said to corroborate $\neg F$, $s \bullet t$ cannot. The new piece of information t 'neutralised' the force of s.

Consequently, the corroboration condition for $F \to G$ makes sense: s can be said to corroborate $F \to G$ iff taking s together with any possible piece of information t that corroborates F results in $s \bullet t$ that corroborates G, and so does every u such that $s \bullet t \sqsubseteq u$.

Observe that the 'boxed' formulas $\Box_\alpha F$ have not received attention yet. To retain generality, we shall not provide truth conditions, but we will focus on a specific class of substructural models.

Definition 6 (Intended substructural models). *An intended substructural model for $L_{\mathcal{G}}$ is a substructural model for $L_{\mathcal{G}}$ such that*

$$s \Vdash F \text{ only if } s \Vdash \Box_\alpha F \text{ for every } \alpha \in \mathcal{G} \tag{10}$$

In intended models, boxed formulas behave somewhat like propositional atoms. Extension of Prop. 1 to the whole $L_{\mathcal{G}}$ within intended models is a trivial consequence of Def. 6. The clause (10) is included also for technical reasons that will become clear in Sect. 5.2. But it can be motivated independently as well: if s corroborates F, then s should be a sufficient reason to believe F. Of course, (10) can be dropped in case it is not considered intuitive enough, but we choose to

keep it.[8] Another reason to keep the clause is that it allows for a natural way of dealing with common belief (Sect. 5.3).

5 Substructural Epistemic Logics

Substructural epistemic logics emerge as soon as we use substructural models as the information structures in information models.

5.1 Substructural Information Models

Definition 7 (Substructural information $\mathcal{C}$-frames and $\mathcal{C}$-models). *Let $\mathcal{C}$ be a class of substructural frames. A substructural information $\mathcal{C}$-frame is a tuple*

$$\mathbf{F} = \langle W, R, \mathfrak{F}, A \rangle \tag{11}$$

where W, R, A are as in Def. 3 and $\mathfrak{F}$ is a substructural frame such that $\mathfrak{F} \in \mathcal{C}$. Moreover, let us assume for technical convenience[9] that

$$s \in A(\alpha, w) \text{ and } s' \sqsubseteq s \text{ only if } s' \in A(\alpha, w) \tag{12}$$

A substructural information $\mathcal{C}$-model built on $\mathbf{F} = \langle W, R, \mathfrak{F}, A \rangle$ is a tuple

$$\mathbf{M} = \langle W, R, \mathfrak{M}, A, V \rangle \tag{13}$$

where W, R, A are as in Def. 3, $\mathfrak{M} = \langle \mathfrak{F}, \Vdash \rangle$ is an intended substructural model and V is a valuation.

The truth conditions of $L_{\mathcal{G}}$-formulas are those of Def. 4. The usual notions of validity in a model and a frame are assumed. F is $\mathcal{C}$-valid iff it is valid in every substructural information $\mathcal{C}$-frame. The set of $\mathcal{C}$-valid formulas shall be denoted as $\mathsf{K}(\mathcal{C})$.

The sets $\mathsf{K}(\mathcal{C})$ can be seen as basic information-based epistemic logics where the pieces of information are 'described' by the logic of $\mathcal{C}$. The actual choice of $\mathcal{C}$ will depend on the application.[10] However, since this paper is focused on the general framework, we shall not discuss such special cases here. We shall limit our discussion to a rather general observation instead.

Proposition 2. *F is $\mathcal{C}$-valid if (but not only if) i) F is a propositional tautology or ii) $F = \Box_\alpha G \to \Box_\alpha G'$, where G entails G' in $\mathcal{C}$.*

[8] There is a standard way of dealing with 'boxes' that invokes additional relations, see [29]. A different evidence-based approach that builds only on C and $\sqsubseteq$ is discussed in [12].

[9] See the proof of Prop. 4 in the appendix.

[10] For example, Sequoiah-Grayson [30] argues that, when modelling the flow of information in inference, associativity $s \bullet (t \bullet u) = (s \bullet t) \bullet u$, contraction $s \bullet s = s$ and other assumptions have to be rejected, leaving only weak commutativity $s \bullet t = t \bullet s$.

Hence, substructural epistemic logics in general respect propositional validity and belief is closed under $\mathcal{C}$-entailment.

Example 6. An example of such a closure principle would be closure under conjunction elimination:

$$\Box_\alpha(F \wedge G) \to (\Box_\alpha F \wedge \Box_\alpha G) \tag{14}$$

On the other hand, some of the more problematic closure principles are not valid.

Example 7. Examples of invalid closure principles include closure under conjunction introduction and Modus Ponens:

$$(\Box_\alpha F \wedge \Box_\alpha G) \to \Box_\alpha(F \wedge G) \tag{15}$$

$$\Box_\alpha(F \to G) \to (\Box_\alpha F \to \Box_\alpha G) \tag{16}$$

The construction of counterexamples is easy and the reader may try it as an exercise.

Closure under logical equivalence does not hold either, i.e. it is not the case that if $F \leftrightarrow G$ is $\mathcal{C}$-valid, then $\Box_\alpha F \leftrightarrow \Box_\alpha G$ is $\mathcal{C}$-valid as well. For example, counterexamples concerning the classical tautology $(p \to q) \leftrightarrow (\neg p \vee q)$ are easily constructed for most classes $\mathcal{C}$.

Consequently, substructural epistemic logics achieve the goal mentioned in Sect. 2: the non-omniscient properties of belief are explained by reference to the 'logical character' of pieces of information.

5.2 Factive and Introspective Models

Definition 8 (Factive frames and models). *A substructural information frame*

$$\mathbf{F} = \langle W, R, \mathfrak{F}, A \rangle$$

is factive iff every $R(\alpha)$ is reflexive on W. A substructural information model $\mathbf{M}$ *built on* $\mathbf{F}$ *is factive iff* $\mathbf{F}$ *is factive. The set of formulas valid in every factive $\mathcal{C}$-frame will be denoted* $\mathsf{T}(\mathcal{C})$.

Proposition 3. $\Box_\alpha F \to F$ *is* $\mathbf{F}$*-valid iff* $\mathbf{F}$ *is a factive frame. Consequently,* $\Box_\alpha F \to F \in \mathsf{T}(\mathcal{C})$ *for every* $\mathcal{C}$.

Hence, in the context of factive models and frames, $\Box_\alpha$ may be read in terms of 'true belief' or even 'knowledge'.

Definition 9 (Introspective frames and models). *A substructural information frame* $\mathbf{F} = \langle W, R, \mathfrak{F}, A \rangle$ *is introspective iff every $R(\alpha)$ is transitive on W and $R_\alpha wv$ implies $A(\alpha, w) \subseteq A(\alpha, v)$ for every α and w. A substructural information model* $\mathbf{M}$ *built on* $\mathbf{F}$ *is introspective iff* $\mathbf{F}$ *is introspective. The set of formulas valid in every introspective $\mathcal{C}$-frame will be denoted* $\mathsf{K4}(\mathcal{C})$ *and the set of formulas valid in every factive and introspective $\mathcal{C}$-frame will be denoted* $\mathsf{S4}(\mathcal{C})$.

Proposition 4. $\Box_\alpha F \to \Box_\alpha \Box_\alpha F$ *is* $\mathbf{F}$*-valid iff* $\mathbf{F}$ *is an introspective frame. Consequently,* $\Box_\alpha F \to \Box_\alpha \Box_\alpha F \in \mathsf{K4}(\mathcal{C})$ *for every* $\mathcal{C}$.

5.3 Common Belief

This section outlines a way to deal with common belief in the substructural epistemic framework. We will work with the standard construal of common belief as an infinite iteration of the 'everybody-believes-operator'. It is noted that common belief lacks some of the standard properties.

Definition 10 (Language with common belief). *The language with common belief $L^*_{\mathcal{G}}$ extends the basic epistemic language with a family of operators $\circledast_B$ for every $B \subseteq \mathcal{G}$. Formulas $\circledast_B F$ are read 'It is common belief in B that F'. Moreover, $\Box_B F$ is a shorthand for $\bigwedge_{\alpha \in B} \Box_\alpha F$, read 'every agent in B believes that F'.*

Definition 11 (Common belief information structures). *A B-sequence σ_B is a sequence of belief-operators $\Box_{\alpha_1} \ldots \Box_{\alpha_n}$ where $n \geq 1$ and every $\alpha_i \in B$. $\mathcal{I}(L^*_{\mathcal{G}})$ is a common belief information $L^*_{\mathcal{G}}$-structure iff it is the case that*

- *$s \Vdash \circledast_B F$ iff $s \Vdash \sigma_B F$ for every B-sequence σ_B*

A common belief $\mathcal{C}$-model is a substructural information $\mathcal{C}$-model where the information $L^*_{\mathcal{G}}$-structure is a common belief information $L^*_{\mathcal{G}}$-structure. $\mathsf{K}^*(\mathcal{C})$, $\mathsf{T}^*(\mathcal{C})$, $\mathsf{K4}^*(\mathcal{C})$ and $\mathsf{S4}^*(\mathcal{C})$ are sets of $L^*_{\mathcal{G}}$-formulas valid in every common belief $\mathcal{C}$-model, every factive, introspective and factive and introspective common belief $\mathcal{C}$-model, respectively.

Lemma 1. *If $s \Vdash F$, then $s \Vdash \sigma_B F$ for every B-sequence σ_B and every $B \subseteq \mathcal{G}$.*

Definition 12 (Group accessibility). *Let $B \subseteq \mathcal{G}$. Let a B-path from w to v be a sequence of couples $\langle w_1, w_2\rangle, \ldots, \langle w_{n-1}, w_n\rangle$ such that $w_1 = w$, $w_n = v$ and every $\langle w_i, w_{i+1}\rangle \in R(\alpha)$ for some $\alpha \in B$. Let $R(B)$ ('B-accessibility') be a binary relation on W such that $\langle w, v\rangle \in R(B)$ iff there is a B-path from w to v.*

Definition 13 (Truth conditions for common belief). *The truth conditions for every $L^*_{\mathcal{G}}$-formula are specified by adding the following clause to Def. 4:*

- *$\mathbf{M}, w \models \circledast_B F$ iff i) $\mathbf{M}, w \models \Box_B F$ and ii) $R_B wv$ implies $\mathbf{M}, v \models \Box_B F$.*

Let us close the section by pointing out that two of the well known axioms for 'common knowledge' hold also for common belief in the substructural epistemic setting, if we limit our attention to factive frames (i.e. if we are studying 'common true belief' or 'common knowledge').

Proposition 5. *The following belong to $\mathsf{T}^*(\mathcal{C})$ for every $\mathcal{C}$:*

1. *$\circledast_B F \to (F \wedge \Box_B \circledast_B F)$ ('Mix')*
2. *$\circledast_B (F \to \Box_B F) \to (F \to \circledast_B F)$ ('Induction')*

However, other standard axioms, such as $\circledast_B$-closure under Modus Ponens and $\circledast_B$-necessitation, do not hold due to the specifics of the simple $\Box_\alpha$-belief.

6 Public Information Introduction

This section investigates into a generalisation of public announcements (not necessarily truthful). If we see formulas as special cases of pieces of information, then the action of publicly announcing a formula is a special case of publicly introducing a piece of information. Hence, it is interesting to look at the more general case.

Definition 14 (The announcement language). *Let* AI *('active pieces of information') and* $\mathcal{G}$ *('agents') be non-empty sets of labels. Formulas of the announcement language* $L^+_{\mathcal{G}}(\Phi, \mathsf{AI})$ *are constructed as follows:*

$$F ::= p \mid \neg F \mid F \wedge F \mid F \vee F \mid F \to F \mid F \leftrightarrow F \mid \Box_\alpha F \mid s : F \mid [+s]F \tag{17}$$

where $p \in \Phi$, $\alpha \in \mathcal{G}$ *and* $s \in \mathsf{AI}$.

Formulas $s : F$ are read 's corroborates F' and $[+s]F$ is read 'F is the case after the public introduction of s'. We shall not assume special corroboration conditions for formulas $s : F$ and $[+s]F$.[11]

Definition 15 (Information models for the announcement language). *We shall use the models of Def. 7 with the proviso that* $I \subseteq \mathsf{AI}$. *Validity of formulas in models, frames and classes of frames is defined in the usual way. Truth conditions for the 'basic epistemic fragment' of the announcement language are as before (Def. 4). Moreover:*

- $\mathbf{M}, w \models s : F$ *iff* $s \in \mathcal{I}$ *and* $s \Vdash F$
- $\mathbf{M}, w \models [+s]F$ *iff* $s \in \mathcal{I}$ *implies* $M^{+s}, w \models F$

where

$$\mathbf{M}^{+s} = \langle W^{+s}, R^{+s}, \mathcal{I}(L^+_{\mathcal{G}})^{+s}, A^{+s}, V^{+s} \rangle \tag{18}$$

such that

- $W^{+s} = W$, $\mathcal{I}(L^+_{\mathcal{G}})^{+s} = \mathcal{I}(L^+_{\mathcal{G}})$ *and* $V^{+s}(p) = V(p)$ *for all* $p \in \Phi$
- $R^{+s}_\alpha(w) = R_\alpha(w) - [\![\overline{s}]\!]_M$ *for every* α, w
- $A^{+s}(\alpha, w) = A(\alpha, w) \cup \{s\}$ *for every* α, w

where $R_\alpha(w) = \{v \mid R_\alpha wv\}$ *and* $[\![\overline{s}]\!]_{\mathbf{M}} = \{w \mid \mathbf{M}, w \models \neg F$ *for some* F *such that* $s \Vdash F\}$.

$\mathsf{K}^+(\mathcal{C})$ *is the class of* $L^+_{\mathcal{G}}$*-formulas valid in every information* $\mathcal{C}$*-model for* $L^+_{\mathcal{G}}$. $\mathsf{T}^+(\mathcal{C})$, $\mathsf{K4}^+(\mathcal{C})$ *and* $\mathsf{S4}^+(\mathcal{C})$ *are the classes of* $L^+_{\mathcal{G}}$*-formulas valid in every factive, introspective, and factive and introspective* $\mathcal{C}$*-model for* $L^+_{\mathcal{G}}$.

's corroborates F' holds in a pointed model only if s is 'active' in the model, i.e. if $s \in \mathcal{I} \subseteq \mathsf{AI}$. A public introduction of s inserts s into every $A(\alpha, w)$ and 'cuts off' the accessibility arrows leading to points where the negation of a formula corroborated by s holds. Such an introduction is 'persuasive' and 'monotonic':

[11] However, it might be plausible to assume that $s \Vdash [+t]F$ iff $s \bullet t \Vdash F$.

Lemma 2 (Persuasiveness and monotonicity). *The following are contained in* $\mathsf{K}^+(\mathcal{C})$*, for every* $\mathcal{C}$*:*

1. $s : F \to [+s]\Box_\alpha F$
2. $\Box_\alpha F \to [+s]\Box_\alpha F$

There is a stronger version of information introduction for which these properties do not hold. It is possible to add the assumption that Cst holds for every $t \in A^{+s}(\alpha, w)$. In other words, we could assume that the introduction of s results in 'removing' every t that is not consistent with s from the awareness set. For sake of simplicity, we shall not discuss this version in more detail here.[12]

Note that application of the standard 'reduction-axioms-technique' is seriously limited in the substructural epistemic framework. Importantly, there is no hope of being able to find an equivalent $L_{\mathcal{G}}$-formula for every $L^+_{\mathcal{G}}$-formula. The reason is explained by Exam. 7: it is possible that there are formulas $[+s]F$ and G such that $[+s]F \leftrightarrow G$ is valid, but $\Box_\alpha[+s]F \leftrightarrow \Box_\alpha G$ is not. However, variants of some of the well-known reduction axioms are still valid.

Proposition 6. *The following belong to* $\mathsf{K}^+(\mathcal{C})$*, for every* $\mathcal{C}$*:*

1. $s : G \to ([+s]p \leftrightarrow p)$
2. $s : G \to ([+s]\neg F \leftrightarrow \neg[+s]F)$
3. $[+s](F \wedge G) \leftrightarrow ([+s]F \wedge [+s]G)$
4. $[+s](F \vee G) \leftrightarrow ([+s]F \vee [+s]G)$
5. $[+s](F \to G) \leftrightarrow ([+s]F \to [+s]G)$
6. $[+s](F \leftrightarrow G) \leftrightarrow ([+s]F \leftrightarrow [+s]G)$
7. $s : G \to ([+s]\Box_\alpha F \leftrightarrow (s : F \vee \Box_\alpha F))$
8. $s : G \to ([+s]t : F \leftrightarrow t : F)$

Notice items 1., 2., 7. and 8.: the antecedent $s : G$ is necessary, since not every s is 'active' in every model.[13] To sidestep this, we could narrow our attention down to models where every piece of information expressible in the language is active.

Definition 16 (Full frames). *An information frame is* AI*-full iff* $\mathcal{I} = \langle \mathsf{AI}, \triangle \rangle$.

Corollary 1. *The following are valid in every* AI*-full information frame:*

1. $[+s]p \leftrightarrow p$
2. $[+s]\neg F \leftrightarrow \neg[+s]F$
3. $[+s]\Box_\alpha F \leftrightarrow (s : F \vee \Box_\alpha F)$
4. $[+s]t : F \leftrightarrow t : F$

A note on related work. Combinations of public announcements and information-based epistemic logics are widely studied within dynamic justification logics. However, there are notable differences between the justification-logic-based

[12] However, there is hope that working with both of these versions will yield interesting results concerning the relation of the present framework to the AGM belief revision theory, see [2].

[13] This fact partly justifies our inclusion of formulas $s : F$ into the announcement language. The other part of the justification is the fact that without such formulas, no interesting 'recursion implication' for formulas $[+s]\Box_\alpha F$ would be provable.

approaches and the approach of the present paper. Bucheli et al. [13,14] and Renne [27] combine justification logic with announcements, but the latter are classical formula announcements. Kuznets and Studer [25] combine formula announcements with evidence introduction, in that the announcement itself is considered as a new piece of evidence. The rich framework of Baltag et al. [9] deals with various versions of evidence dynamics, but does so only for the singe-agent case and the 'pieces of information' are considered from the viewpoint of justification logic, not substructural logic. Apart from the justification-logic-based approaches, an interesting contribution has been made by van Benthem and Pacuit [11], but they construe evidence in terms of sets of possible worlds and their announcements are formula-based as well.

7 Conclusion

Our primary goal in this paper was to explain that substructural logics are a natural part of information-based epistemic logic. This observation may stimulate productive collaborations between sub-fields of logic that have perhaps been thought of as rather remote from one another. The paper is an introductory outline and, consequently, there are many interesting directions for future work. First, we plan to concentrate on specific substructural epistemic logics: to explain their respective philosophical motivations in more detail and to provide axiomatisations. Second, as the present framework is rather general, its will be interesting to expound connections to the established formalisms. Third, the information-introduction-extensions of substructural epistemic logics deserve systematic attention: sound and complete axiomatisations are a natural goal, as is establishing connections with the well-known version of dynamic-epistemic logics. Moreover, there are many other dynamic extensions that have been left out of the present outline. In addition, it is interesting to dwell upon the 'philosophical background' of the present framework: one could formulate different readings of 'corroboration' and 'piece of information' and provide various versions of information-based logics built to fit the different readings. There is hope that this will result in non-trivial applications of the present framework in epistemology.

Acknowledgements. I gratefully acknowledge the constructive comments of the anonymous referees. The paper builds upon a talk given at MCMP, Munich in February 2013, where the audience provided helpful feedback as well. This work was carried out at the Department of Logic and Methodology of Sciences, Comenius University in Bratislava as a part of the research project 'Semantic models, their explanatory power and applications', supported by the grant VEGA 1/0046/11.

Appendix: Proofs of Propositions

This technical appendix contains proofs of some of the Propositions stated in the paper.

Proof of Prop. 3. The implication from right to left is trivial. The converse implication is easily demonstrated in the usual way. Assume that $\mathbf{F}$ is not a factive frame. Hence, there are α and w such that $\neg R_\alpha ww$. Since $A(\alpha, w)$ is non-empty by Def. 3, we may choose an arbitrary $s \in A(\alpha, w)$. Now define a model $\mathbf{M}$ built on $\mathbf{F}$ such that $s \Vdash p$, $\mathbf{M}, w \not\models p$ and $R_\alpha wv$ implies $\mathbf{M}, v \models p$ for every v. It is obvious that $\mathbf{M} \not\models \Box_\alpha p \to p$.

Proof of Prop. 4. The implication from right to left is trivial. Again, the converse implication is easily demostrated in the usual way. Assume that $\mathbf{F}$ is not introspective. The assumption entails that 1) there are α, w, v, v' such that $R_\alpha wv$, $R_\alpha vv'$ and $\neg R_\alpha wv'$, *or* 2) there are α, w, v such that $R_\alpha wv$ and $A(\alpha, w) \not\subseteq A(\alpha, v)$.

Assume 1). Build a model $\mathbf{M}$ as follows. Choose an arbitrary $s \in A(\alpha, w)$ and set $s \Vdash p$. In addition, set $\mathbf{M}, u \models p$ for every u such that $R_\alpha wu$ and $\mathbf{M}, u' \not\models p$ for every u' such that $R_\alpha vu'$. It is plain that $\mathbf{M}, w \not\models \Box_\alpha p \to \Box_\alpha \Box_\alpha p$.

Now assume 2). There is a $s \in A(\alpha, w)$ such that $s \notin A(\alpha, v)$. Build a model $\mathbf{M}$ as follows. Let $t \Vdash p$ iff $t \in \{s' \mid s \sqsubseteq s'\}$ for all t. Moreover, let $t \not\Vdash p$ for every $t \in A(\alpha, v)$. This choice is possible due to (12) of Def. 7. Moreover, set $\mathbf{M}, u \models p$ for every u such that $R_\alpha wu$. It is plain that $\mathbf{M}, w \not\models \Box_\alpha p \to \Box_\alpha \Box_\alpha p$.

Proof of Prop. 5. Item 1. $\circledast_B F$ obviously entails F (Def. 8, 13 and Prop. 3). It remains to prove that $\circledast_B F$ entails $\Box_B \circledast_B F$. Now $\mathbf{M}, w \models \circledast_B F$ entails $\mathbf{M}, w \models \Box_\alpha F$ for every $\alpha \in B$ (Def. 13). The latter entails that there is a $t \in A(\alpha, w)$ such that $t \Vdash F$. Consequently, $t \Vdash \sigma_B F$ for every B-sequence σ_B (Lem. 1) and $t \Vdash \circledast_B F$.

Hence, it remains to prove that $R_\alpha wv$ and $\mathbf{M}, w \models \circledast_B F$ together imply $\mathbf{M}, v \models \circledast_B F$ for every v and $\alpha \in B$. Assume to the contrary. The assumption entails that a) $\mathbf{M}, v \not\models \Box_B F$ or b) $R_B vu$ and $\mathbf{M}, u \not\models \Box_B F$ for some u. However, both are impossible, since $R_B wv$ and $R_B wu$: consequently, the assumption entails that $\mathbf{M}, v \models \Box_B F$ and $\mathbf{M}, u \models \Box_B F$.

Item 2. The proof is virtually identical to the standard inductive proof of a similar claim in modal-logic-based epistemic logic [15, p. 37]. Assume that $\mathbf{M}, w \models \circledast_B(F \to \Box_B F) \wedge F$. We have to show that $\mathbf{M}, w \models \circledast_B F$, i.e. that $R_B^n wv$ entails $\mathbf{M}, v \models \Box_B F$ for every $n \geq 0$, where $R_B^0 wv$ iff $w = v$ and $R_B^m wv$ iff v is reachable from w by a B-path of length m. The base case for $n = 0$ is trivial. Now assume that the claim holds for a specific m: there is a v such that $R_B^m wv$ and $\mathbf{M}, v \models \Box_B F$. To prove the claim for $m + 1$, pick an $\alpha \in B$ and a u such that $R_\alpha vu$. Now $\mathbf{M}, v \models \Box_\alpha F$ obviously entails $\mathbf{M}, u \models F$. But $R_B wu$ and, consequently, $\mathbf{M}, u \models F \to \Box_B F$. Thus, $\mathbf{M}, u \models \Box_B F$ as desired.

Proof of Prop. 6. Item 1. For every $\mathbf{M}, w$: $\mathbf{M}, w \models s : G \to [+s]p$ iff ($s \in \mathcal{I}$ and $s \Vdash G$) implies ($s \in \mathcal{I}$ and $\mathbf{M}^{+s}, w \models p$) iff ($s \in \mathcal{I}$ and $s \Vdash G$) implies ($s \in \mathcal{I}$ and $\mathbf{M}, w \models p$) iff $\mathbf{M}, w \models s : G \to p$. By propositional logic, $\mathbf{M}, w \models s : G \to ([+s]p \leftrightarrow p)$, for every $\mathbf{M}, w$.

Item 2. First, let us prove that $s : G \wedge \neg[+s]F$ implies $[+s]\neg F$. By Def. 15, $\neg[+s]F$ is equivalent to the conjunction of $s \in \mathcal{I}$ and $\mathbf{M}^{+s}, w \models \neg F$. The conjunction implies that $s \in \mathcal{I} \Rightarrow \mathbf{M}^{+s}, w \models \neg F$, i.e. that $\mathbf{M}, w \models [+s]\neg F$. The desired result follows by propositional logic. Second, let us prove that $s : G \wedge [+s]\neg F$ implies $\neg[+s]F$. The assumption $\mathbf{M}, w \models s : G \wedge [+s]\neg F$ is equivalent to the conjunction of $s \in \mathcal{I}$, $s \Vdash G$ and ($s \in \mathcal{I} \Rightarrow \mathbf{M}^{+s}, w \models \neg F$). The conjunction obviously entails $s \in \mathcal{I}$ and $\mathbf{M}^{+s}, w \not\models F$, i.e. $\mathbf{M}, w \models \neg[+s]F$.

Items 3. – 6. can be demonstrated by simple propositional reasoning. Item 7. One half of the result follows from Lemma 2. To prove the second half, assume that $\mathbf{M}, w \models s : G \wedge \neg s : F \wedge \neg\Box_\alpha F$. The first two conjuncts entail that $s \nVdash F$. The third conjunct entails that i) there is a v such that $R_\alpha wv$ and $\mathbf{M}, v \models \neg F$, or ii) there is no $t \in A(\alpha, w)$ such that $t \Vdash F$. Assume i). Since $s \nVdash F$, $R_\alpha^{+s} wv$ for the $\neg F$-world v. Consequently, $\mathbf{M}, w \not\models [+s]\Box_\alpha F$. Assume ii). Since, $s \nVdash F$, there is no $t' \in A^{+s}(\alpha, w)$ such that $t' \Vdash F$ and, consequently, $\mathbf{M}, w \not\models [+s]\Box_\alpha F$.

Item 8. can be proved easily by propositional reasoning and by using the fact that $\mathcal{I}(L_\mathcal{G}^+)^{+s} = \mathcal{I}(L_\mathcal{G}^+)$ (Def. 15).

References

1. Adriaans, P., van Benthem, J. (eds.): Philosophy of Information. Elsevier, Amsterdam (2008)
2. Alchourron, C., Gärdenfors, P., Makinson, D.: On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. J. Symbolic Logic 50, 510–530 (1985)
3. Artemov, S.: Operational Modal Logic. Technical report, Cornell University (1995)
4. Artemov, S.: Explicit Provability and Constructive Semantics. B. Symb. Log. 7, 1–36 (2001)
5. Artemov, S.: The Logic of Justification. Rev. Symb. Log. 1, 477–513 (2008)
6. Artemov, S.: Why Do We Need Justification Logic? In: van Benthem, J., Gupta, A., Pacuit, E. (eds.) Games, Norms and Reasons: Logic at the Crossroads, pp. 23–38. Springer, Dordrecht (2011)
7. Baltag, A., Moss, L., Sołecki, S.: The Logic of Common Knowledge, Public Announcements, and Private Suspicions. In: Gilboa, I. (ed.) Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 1998), pp. 43–56. Morgan Kaufmann, San Francisco (1998)
8. Baltag, A., Moss, L.: Logics for Epistemic Programs. Synthese 139, 165–224 (2004)
9. Baltag, A., Renne, B., Smets, S.: The Logic of Justified Belief Change, Soft Evidence and Defeasible Knowledge. In: Ong, L., de Queiroz, R. (eds.) WoLLIC 2012. LNCS, vol. 7456, pp. 168–190. Springer, Heidelberg (2012)
10. van Benthem, J.: Logical Dynamics of Information and Interaction. Cambridge University Press, Cambridge (2011)
11. van Benthem, J., Pacuit, E.: Dynamic Logics of Evidence-Based Beliefs. Studia Logica 99, 61–92 (2011)

12. Bílková, M., Majer, O., Peliš, M., Restall, G.: Relevant Agents. In: Beklemishev, L., Goranko, V., Shehtman, V. (eds.) Advances in Modal Logic, vol. 8, pp. 22–38. College Publications, London (2010)
13. Bucheli, S., Kuznets, R., Renne, B., Sack, J., Studer, T.: Justified Belief Change. In: Arrazola, X., Ponte, M. (eds.) Proceedings of the Second ILCLI International Workshop on Logic and Philosophy of Knowledge, Communication and Action (LogKCA 2010). The University of the Basque Country Press, San Sebastian (2010)
14. Bucheli, S., Kuznets, R., Studer, T.: Realizing public announcements by justifications. To appear in J. Comput. Syst. Sci.
15. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer, Dordrecht (2008)
16. Dretske, F.: Knowledge and the Flow of Information. MIT Press, Cambridge (1981)
17. Fagin, R., Halpern, J.: Belief, Awareness, and Limited Reasoning. Artif. Intell. 34, 39–76 (1988)
18. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning About Knowledge. MIT Press, Cambridge (1995)
19. Fitting, M.: Logic of Proofs, Semantically. Ann. Pure Appl. Logic 132, 1–25 (2005)
20. Fitting, M.: Reasoning with Justifications. In: Makinson, D., Malinowski, J., Wansing, H. (eds.) Towards Mathematical Philosophy: Papers from the Studia Logica Conference Trends in Logic IV, pp. 107–123. Springer, Dordrecht (2009)
21. Hintikka, J.: Knowledge and Belief: An Introduction to the Logic of the Two Notions. Cornell University Press, Ithaca (1962)
22. Hintikka, J.: Impossible Possible Worlds Vindicated. J. Philos. Logic 4, 475–484 (1975)
23. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press, Cambridge (2000)
24. Kripke, S.: Semantical Analysis of Intuitionistic Logic. In: Crossley, J., Dummett, M.A.E. (eds.) Formal Systems and Recursive Functions, pp. 92–130. North-Holland Publishing Company, Amsterdam (1965)
25. 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)
26. Mares, E.: Relevant Logic. A Philosophical Interpretation. Cambridge University Press, Cambridge (2004)
27. Renne, B.: Public Communication in Justification Logic. J. Logic Comput. 21, 1005–1034 (2011)
28. Restall, G.: Information Flow and Relevant Logics. In: Seligman, J., Westerstøahl, D. (eds.) Logic, Language, and Computation, vol. 1, pp. 463–467. CSLI Publications, Stanford (1995)
29. Restall, G.: An Introduction to Substructural Logics. Routledge, London (2000)
30. Sequoiah-Grayson, S.: Epistemic Closure and Commutative, Nonassociative Residuated Structures. Synthese 190, 113–128 (2013)
31. Shannon, C.E.: A Mathematical Theory of Communication. AT&T Tech. J. 27, 379–423, 623–656 (1948)
32. Schröder-Heister, P., Došen, K.: Substructural Logics. Oxford University Press, Oxford (1993)
33. Urquhart, A.: Semantics for Relevant Logics. J. Symbolic Logic 37, 159–169 (1972)

Author Index