LNCS 10669

Alexandre Madeira
Mário Benevides (Eds.)

Dynamic Logic

New Trends and Applications

First International Workshop, DALI 2017
Brasilia, Brazil, September 23–24, 2017
Proceedings

Springer

Lecture Notes in Computer Science 10669

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

Alexandre Madeira · Mário Benevides (Eds.)

Dynamic Logic

New Trends and Applications

First International Workshop, DALI 2017
Brasilia, Brazil, September 23–24, 2017
Proceedings

Springer

Editors
Alexandre Madeira
INESC TEC
University of Minho
Braga
Portugal

and

CIDMA
University Aveiro
Aveiro
Portugal

Mário Benevides
UFRJ
Rio de Janeiro
Brazil

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-73578-8 ISBN 978-3-319-73579-5 (eBook)
https://doi.org/10.1007/978-3-319-73579-5

Library of Congress Control Number: 2017962907

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

Printed on acid-free paper

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

Preface

Building on the pioneer intuitions of Floyd–Hoare logic, dynamic logic was introduced in the 1970s by Vaughan Pratt as a suitable logic to reason about, and verify, classic imperative programs. Since then, the original intuitions grew to an entire family of logics, which became increasingly popular for assertional reasoning about a wide range of computational systems. Simultaneously, their object (i.e., the very notion of a program) evolved in unexpected ways. This leads to dynamic logics tailored to specific programming paradigms and extended to new computing domains, including probabilistic, continuous, and quantum computation. Both its theoretical relevance and practical potential make dynamic logic a topic of interest in a number of scientific venues, from wide-scope software engineering conferences to modal logic-specific events. However, as far as we know, to date, no specific event was exclusively dedicated to it. This workshop emerged from this discussion, during the kick-off meeting of the project DaLí - Dynamic Logics for Cyber-Physical Systems, whose editors are participating as consultant and IR, respectively.

This volume contains the proceedings of the event that was held in the beautiful city of Brasilia during September 23–24, 2017, co-located with Tableaux, ITP, and FroCoS 2017.

We received 24 submissions, of which, after a careful revision process, with at least three revisions per work, 12 papers were accepted as regular papers and are published in this volume. Beyond these contributions, the workshop also included the following short papers:

– Fabricio Chalub, Alexandre Rademaker, Edward Hermann Haeusler and Christiano Braga: "Fixing the Proof of completeness of ALC Sequent Calculus"
– Diana Costa and Édi Duarte: "Checkers Game in Deontic Logic"
– Daniel Figueiredo and Manuel A. Martins: "Bisimulations for Reactive Frames"
– Konstantinos Gkikas and Alexandru Baltag: "Stable Beliefs and Conditional Probability Spaces"
– Leandro Gomes: "Contract-Based Design for Software Verification"
– Luiz Carlos Pereira: "Constructive Fragments of Classical Modal Logic and the Ecumenical Perspective"
– Carlos Tavares: "Toward a Quantum-Probabilistic Dynamic Logic"

From this list, four papers were originally submitted as short contribution and the other three were regular submissions invited to be converted into a short format. An informal volume with the extended abstracts (3–5 pages) of these contributions was provided in the conference.

We also had the invited talks of Alexandru Baltag, "Logic Goes Viral: Dynamic Modalities for Social Networks," and of Edward Hermann Haeusler, "Propositional Dynamic Logic with Petri Net Programs: A Discussion and a Logical System"; the latter is a joint work with Bruno Lopes and Mario Benevides. Finally, we had the pleasure of hosting a special talk entitled “Dynamic Logic, a Personal Perspective,” by the Dynamic Logics pioneer Vaughan Pratt.

The organization sincerely acknowledges the authors that submitted their works to our workshop, to the Program Committee for their careful and attentive revisions, to the Invited Speakers for their very interesting talks and to the local organizers for their valuable and very prompt support.

September 2017

Mario Benevides
Alexandre Madeira

Organization

Conference Chairs

Mario Benevides	UFRJ, Brazil
Alexandre Madeira	University of Minho, Portugal

Program Committee

Carlos Areces	University of Cordoba, Argentina
Phillippe Balbiani	University of Toulouse, France
Alexandru Baltag	Uva, The Netherlands
Luís S. Barbosa	University of Minho, Portugal
Johan van Benthem	University of Stanford, USA and University of Tsinghua, China
Patrick Blackburn	University of Roskilde, Denmark
Fredrik Dahlqvist	UCL, UK
Stéphane Demri	CNRS, France
Hans van Ditmarsch	LORIA, Nancy, France
Francicleber M. Ferreira	UFC, Brazil
Valentin Goranko	University of Stockholm, Sweden
Edward H. Hauesler	PUC-Rio, Brazil
Rolf Hennicker	LMU, Munich, Germany
Andreas Herzig	Toulouse, France
Dexter Kozen	Cornell, USA
Clemens Kupke	University of Strathclyde, UK
Bruno Lopes Vieira	UFF, Brazil
Paulo Mateus	IST, Portugal
Manuel A. Martins	University of Aveiro, Portugal
Carlos Olarte	UF RN, Brazil
José N. Oliveira	University of Minho, Portugal
André Platzer	CMU, USA
Eugénio Rocha	University of Aveiro, Portugal
Valéria de Paiva	NC, USA
Regivan Santiago	UFRN, Brazil
Luis Menasche Schechter	UFRJ, Brazil
Tinko Tinchev	Sofia University, Bulgaria
Petrucio Viana	UFF, Brazil
Sheila Veloso	UFRJ, Brazil
Yde Venema	ILLC, The Netherlands
Renata Wassermann	USP, Brazil

Additional Reviewers

Brandon Bohrer
Diana Costa
Daniel Figueiredo
Isaque Lima
Vitor Machado
Johannes Marti
António Pereira
Elaine Pimentel

Sponsoring

This workshop was promoted by DaLí, a research project financed by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalization, COMPETE 2020 Program, and by National Funds through the Portuguese funding agency, FCT, Fundação para a Ciência e a Tecnologia, within reference POCI-01-0145-FEDER-016692.

FCT, Fundação para a Ciência e Tecnologia
CIDMA, Center for Research and Development in Mathematics and Applications
High Assurance Laboratory INESC TEC

Contents

Undecidability of Relation-Changing Modal Logics

Carlos Areces[1,2], Raul Fervari[1,2(✉)], Guillaume Hoffmann[1,2], and Mauricio Martel[3]

[1] FaMAF, Universidad Nacional de Córdoba, Córdoba, Argentina
{areces,fervari,hoffmann}@famaf.unc.edu.ar
[2] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Buenos Aires, Argentina
[3] Fachbereich Mathematik und Informatik, Universität Bremen, Bremen, Germany
martel@informatik.uni-bremen.de

Abstract. Relation-changing modal logics are extensions of the basic modal logic that allow to change the accessibility relation of a model during the evaluation of a formula. In particular, they are equipped with dynamic modalities that are able to delete, add and swap edges in the model, both locally and globally. We investigate the satisfiability problem of these logics. We define satisfiability-preserving translations from an undecidable memory logic to relation-changing modal logics. This way we show that their satisfiability problems are undecidable.

Keywords: Modal logics · Dynamic logics · Satisfiability Undecidability

1 Introduction

Modal logics [12,14] were originally conceived as logics of necessary and possible truths. They are now viewed, more broadly, as logics that explore a wide range of modalities, or modes of truth: epistemic ("it is known that"), doxastic ("it is believed that"), deontic ("it ought to be the case that"), or temporal ("it has been the case that"), among others. From a model theoretic perspective, the field evolved into a discipline that deals with languages interpreted on various kinds of relational structures or graphs. Nowadays, modal logics are actively used in areas as diverse as software verification, artificial intelligence, semantics and pragmatics of natural language, law, philosophy, etc.

From an abstract point of view, modal logics can be seen as formal languages to navigate and explore properties of a given relational structure. But if we want to describe and reason about *dynamic aspects* of a given situation, e.g., how the relations between a set of elements *evolve* through time or through the application of certain operations, the use of modal logics (or actually, any kind of logic with classical semantics) becomes less clear. We can always resort

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 1–16, 2018.
https://doi.org/10.1007/978-3-319-73579-5_1

to modeling the whole space of possible evolutions as a graph, but this soon becomes unwieldy. It would be more elegant to use truly dynamic modal logics with operators that can mimic the changes the structure will undergo.

There exist several dynamic modal operators that fit in this approach. A clear example are the dynamic operators introduced in dynamic epistemic logics (see, e.g. [22]). Less obvious examples are given by hybrid logics [8,13] equipped with the down arrow operator ↓ which is used to 'rebind' names for states to the current point of evaluation, and memory logics [19], a kind of restricted form of hybrid logics that come equipped with a memory and operators to store and retrieve states from it. Finally, a classical example which can arguably be taken as the origin of the studies of logics in this approach is Sabotage Logic introduced by van Benthem in [21], which provides an operator that deletes individual edges in the model.

Generalizing this last logic, we study operators that do various kinds of change to the accessibility relation of a model: deleting, adding, and swapping edges, both locally (near the state of evaluation) and globally (anywhere). We call these operators *relation-changing*. In [2], the operators are introduced, and it is shown that the model checking problem is PSPACE-complete for the basic modal logic enriched with any of these operators. In this article, we consider the satisfiability problem of these logics. Previous results on this topic are the undecidability of (multimodal) global sabotage logic, via encoding of the Post Correspondence Problem [16] the undecidability of local swap logic with a single relation, by reduction from memory logic [4]; and non-terminating tableau methods for all six logics [3]. Here we present undecidability proofs for all six logics using reductions from memory logic.

The undecidability results can be surprising, considering for instance that dynamic epistemic logics are decidable [11,17,22]. However, other very expressive dynamic operators are undecidable, such as the hybrid logic with the ↓ operator [8]. As we mentioned before, ↓ binds states of the model to some particular names. We will show in this article that relation-changing operators can take advantage of adding, deleting or swapping around edges, to perform some sort of binding in the model, turning them undecidable.

Contributions

- We sketch the undecidability proof for the memory logic $\mathsf{ML}(Ⓡ, Ⓚ)$, by adapting the undecidability argument introduced in [18] for the description logic $\mathcal{ALC}\mathsf{self}$.
- We introduce undecidability proofs for the satisfiability problem of six relation-changing modal logics via satisfiability of memory logic. In this way, we complete the picture of the computational aspects of the family of languages defined in this framework.
- Our proofs improve previous ones for local swap [4] and global sabotage [16], by exploiting undecidability of memory logics. This allows for shorter proofs and avoid redundant encodings of the tiling problem.

The article is organized as follows. In Sect. 2 we introduce the syntax and semantics of relation-changing modal logics. In Sect. 3 we introduce the memory logic $\mathsf{ML}(ⓡ, ⓚ)$ and a sketch of the proof of its undecidability. We dedicate Sect. 4 to the translations from memory to global and local relation-changing modal logics. Finally we draw our conclusions in Sect. 5.

2 Relation-Changing Modal Logics

In this section, we formally introduce *relation-changing modal logics*. For more details and motivations, we direct the reader to [15].

Definition 1 (Syntax). *Let* PROP *be a countable, infinite set of propositional symbols. The set* FORM *of formulas over* PROP *is defined as:*

$$\mathsf{FORM} ::= p \mid \neg\varphi \mid \varphi \wedge \psi \mid \Diamond\varphi \mid \blacklozenge\varphi,$$

where $p \in \mathsf{PROP}$*,* $\blacklozenge \in \{\langle\mathsf{sb}\rangle, \langle\mathsf{br}\rangle, \langle\mathsf{sw}\rangle, \langle\mathsf{gsb}\rangle, \langle\mathsf{gbr}\rangle, \langle\mathsf{gsw}\rangle\}$ *and* $\varphi, \psi \in \mathsf{FORM}$*. Other operators are defined as usual. In particular,* $\blacksquare\varphi$ *is defined as* $\neg\blacklozenge\neg\varphi$*.*

Let ML *(the basic modal logic) be the logic without the* $\{\langle\mathsf{sb}\rangle, \langle\mathsf{br}\rangle, \langle\mathsf{sw}\rangle, \langle\mathsf{gsb}\rangle, \langle\mathsf{gbr}\rangle, \langle\mathsf{gsw}\rangle\}$ *operators, and* $\mathsf{ML}(\blacklozenge)$ *the extension of* ML *allowing also* $\blacklozenge$*, for* $\blacklozenge \in \{\langle\mathsf{sb}\rangle, \langle\mathsf{br}\rangle, \langle\mathsf{sw}\rangle, \langle\mathsf{gsb}\rangle, \langle\mathsf{gbr}\rangle, \langle\mathsf{gsw}\rangle\}$*.*

Semantically, formulas are evaluated in standard relational models, and the meaning of the operators of the basic modal logic remains unchanged (see [12] for details). When we evaluate formulas containing relation-changing operators, we will need to keep track of the edges that have been modified. To that end, let us define precisely the models that we will use.

Definition 2 (Models and model updates). *A model* M *is a triple* $\mathsf{M} = \langle W, R, V\rangle$*, where* W *is a non-empty set whose elements are called points or states,* $R \subseteq W \times W$ *is the accessibility relation, and* $V : \mathsf{PROP} \to \mathcal{P}(W)$ *is a valuation. We define the following notation:*

(sabotaging) $\mathsf{M}^-_S = \langle W, R^-_S, V\rangle$, *with* $R^-_S = R \backslash S$, $S \subseteq R$.
(bridging) $\mathsf{M}^+_S = \langle W, R^+_S, V\rangle$, *with* $R^+_S = R \cup S$, $S \subseteq (W \times W) \backslash R$.
(swapping) $\mathsf{M}^*_S = \langle W, R^*_S, V\rangle$, *with* $R^*_S = (R \backslash S^{-1}) \cup S$, $S \subseteq R^{-1}$.

Intuitively, M^-_S is obtained from M by deleting the edges in S, and similarly M^+_S adds the edges in S to the accessibility relation, and M^*_S adds the edges in S as inverses of edges previously in the accessibility relation.

Let w be a state in M, the pair (M, w) is called a pointed model (we will usually drop parentheses). In the rest of this article, we will use wv as a shorthand for $\{(w, v)\}$ or (w, v); context will always disambiguate the intended use.

Definition 3 (Semantics). *Given a pointed model* M, w *and a formula* φ*, we say that* M, w *satisfies* φ*, and write* $\mathsf{M}, w \models \varphi$*, when*

$$\begin{array}{lll}
\mathsf{M}, w \models p & \textit{iff} & w \in V(p) \\
\mathsf{M}, w \models \neg\varphi & \textit{iff} & \mathsf{M}, w \not\models \varphi \\
\mathsf{M}, w \models \varphi \wedge \psi & \textit{iff} & \mathsf{M}, w \models \varphi \textit{ and } \mathsf{M}, w \models \psi \\
\mathsf{M}, w \models \Diamond\varphi & \textit{iff} & \textit{for some } v \in W \textit{ s.t. } (w,v) \in R, \mathsf{M}, v \models \varphi \\
\mathsf{M}, w \models \langle \mathsf{sb} \rangle\varphi & \textit{iff} & \textit{for some } v \in W \textit{ s.t. } (w,v) \in R, \mathsf{M}^-_{wv}, v \models \varphi \\
\mathsf{M}, w \models \langle \mathsf{br} \rangle\varphi & \textit{iff} & \textit{for some } v \in W \textit{ s.t. } (w,v) \notin R, \mathsf{M}^+_{wv}, v \models \varphi \\
\mathsf{M}, w \models \langle \mathsf{sw} \rangle\varphi & \textit{iff} & \textit{for some } v \in W \textit{ s.t. } (w,v) \in R, \mathsf{M}^*_{vw}, v \models \varphi \\
\mathsf{M}, w \models \langle \mathsf{gsb} \rangle\varphi & \textit{iff} & \textit{for some } v, u \in W, \textit{ s.t. } (v,u) \in R, \mathsf{M}^-_{vu}, w \models \varphi \\
\mathsf{M}, w \models \langle \mathsf{gbr} \rangle\varphi & \textit{iff} & \textit{for some } v, u \in W, \textit{ s.t. } (v,u) \notin R, \mathsf{M}^+_{vu}, w \models \varphi \\
\mathsf{M}, w \models \langle \mathsf{gsw} \rangle\varphi & \textit{iff} & \textit{for some } v, u \in W, \textit{ s.t. } (v,u) \in R, \mathsf{M}^*_{uv}, w \models \varphi.
\end{array}$$

We say φ *is satisfiable if for some pointed model* M, w*, we have* $\mathsf{M}, w \models \varphi$*.*

Notice that $\langle \mathsf{br} \rangle$ and $\langle \mathsf{gbr} \rangle$ always add new edges in the model, and fail in case no new edge can be created. Other versions in which such edge is not necessarily new could be considered, but in that case the operators would behave sometimes as a $\Diamond$ or as a "do nothing", respectively. However, we conjecture that similar results could be proved for those and other versions of the operations.

Relation-changing operators can modify the accessibility relation and check for such changes in the model, and therefore can be used to mark and check for marked states, simulating some sort of binding. Adequately, marking and checking states are the basic dynamic operations *remember* and *known* that can be performed by *memory logics*, a formalism that we present in the next section.

3 Undecidability of Monomodal Memory Logic

Memory logics [1,19] are modal logics that can *store* the current state of evaluation into a memory and *check* whether the current state belongs to this memory. The memory is a subset of the domain of the model. We call $\mathsf{ML}(ⓡ, ⓚ)$ the memory logic that extends ML with the operators ⓡ and ⓚ, which stand for "remember" and "known", respectively.

Definition 4 (Syntax). *Let* PROP *be a countable, infinite set of propositional symbols. The set* FORM *of formulas over* PROP *is defined as:*

$$\mathsf{FORM} ::= p \mid ⓚ \mid \neg\varphi \mid \varphi \wedge \psi \mid \Diamond\varphi \mid ⓡ\varphi,$$

where $p \in \mathsf{PROP}$ *and* $\varphi, \psi \in \mathsf{FORM}$*. Other operators are defined as usual.*

Definition 5 (Semantics). *A model* $\mathsf{M} = \langle W, R, V, S \rangle$ *is a relational model equipped with a set* $S \subseteq W$ *called the* memory*. Let* w *be a state in* W*. The inductive definition of satisfiability for the cases specific to memory logic is:*

$$\begin{array}{lll}
\langle W, R, V, S \rangle, w \models ⓡ\varphi & \textit{iff} & \langle W, R, V, S \cup \{w\} \rangle, w \models \varphi \\
\langle W, R, V, S \rangle, w \models ⓚ & \textit{iff} & w \in S.
\end{array}$$

The remaining cases coincide with the semantics of ML*, and do not involve the memory.*

An $\mathsf{ML}(ⓡ, ⓚ)$*-formula* φ *is* satisfiable *if there are a model* $\mathsf{M} = \langle W, R, V, \emptyset \rangle$ *and* $w \in W$ *such that* $\mathsf{M}, w \models \varphi$*. The empty initial memory ensures that no state of the model satisfies the unary predicate* ⓚ *unless a formula* ⓡψ *has previously been evaluated there.*

Multimodal memory logic is shown to be undecidable in [7]. We strengthen this result, showing that undecidability holds also in the monomodal case.

Theorem 1. *The satisfiability problem of* $\mathsf{ML}(ⓡ, ⓚ)$ *is undecidable.*

Proof. The problem of concept consistency in the description logic $\mathcal{ALC}\mathsf{self}$ is undecidable [18]. Let us name $\mathsf{Tiling}(t)$ the concept defined in [18] that encodes an instance t of the (undecidable) problem of tiling the plane. A reduction of Tiling to the satisfiability problem of $\mathsf{ML}(ⓡ, ⓚ)$ can be done by replacing the $\mathcal{ALC}\mathsf{self}$ operator $\forall R$ by $\Box$, $\exists R$ by $\Diamond$, $\mathbf{I}$ by ⓡ and $\mathbf{me}$ by ⓚ.

We previously suggested that relation-changing operators could, each one in its own way, simulate remember and known operators. However, there is one important difference between the ⓡ operator and relation-changing operators like $\langle \mathsf{sb} \rangle$. While $\langle \mathsf{sb} \rangle\varphi$ always results in a change in the model, ⓡφ can leave the memory unchanged if the current state of evaluation is already memorized. We ignore this difference by observing that any $\mathsf{ML}(ⓡ, ⓚ)$-formula can be rewritten into an equivalent formula where every occurrence of ⓡ is "proper", in the sense that it actually modifies the memory.

Definition 6 (PNF). *An* $\mathsf{ML}(ⓡ, ⓚ)$*-formula is in* proper normal form *(PNF) if every occurrence of a sub-formula* ⓡψ *occurs within the following sub-formula:*

$$(\neg ⓚ \wedge ⓡ\psi) \vee (ⓚ \wedge \psi)$$

Finally, we define the notion of *modal depth* of an $\mathsf{ML}(ⓡ, ⓚ)$-formula.

Definition 7. *Given* φ *in* $\mathsf{ML}(ⓡ, ⓚ)$*, we define the* modal depth *of* φ *(notation* $\mathsf{md}\varphi$*) as*

$$\begin{aligned} \mathsf{md}(ⓚ) &= 0 \\ \mathsf{md}(p) &= 0 \textit{ for } p \in \mathsf{PROP} \\ \mathsf{md}(ⓡ\varphi) &= \mathsf{md}(\varphi) \\ \mathsf{md}(\neg\varphi) &= \mathsf{md}(\varphi) \\ \mathsf{md}(\varphi \wedge \psi) &= max\{\mathsf{md}(\varphi), \mathsf{md}(\psi)\} \\ \mathsf{md}(\Diamond\varphi) &= 1 + \mathsf{md}(\varphi). \end{aligned}$$

In the next section we prove that the satisfiability problem of relation-changing modal logics is undecidable via reductions from monomodal memory logic. We assume that memory logic formulas are always in PNF. This is important for structural inductive proofs.

4 Undecidability of Relation-Changing Logics

In this section, we present satisfiability-preserving translations from $\mathsf{ML}(\textcircled{r}, \textcircled{k})$ to relation-changing modal logics. Combining these translations with the undecidability result of Theorem 1, we can claim:

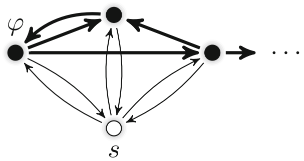
Theorem 2. *The satisfiability problem of* $\mathsf{ML}(\blacklozenge)$ *is undecidable, for* $\blacklozenge \in \{\langle \mathsf{sb} \rangle, \langle \mathsf{br} \rangle, \langle \mathsf{sw} \rangle, \langle \mathsf{gsb} \rangle, \langle \mathsf{gbr} \rangle, \langle \mathsf{gsw} \rangle\}$.

The main idea of these translations is to simulate the behavior of $\mathsf{ML}(\textcircled{r}, \textcircled{k})$ without having an external memory in the model. We simulate the ability to store states in a memory by changing the accessibility relation of a model. Checking for membership in the memory is simulated by checking for changes in the accessibility relation.

Every translation $\tau_{\blacklozenge}$ from $\mathsf{ML}(\textcircled{r}, \textcircled{k})$-formulas to $\mathsf{ML}(\blacklozenge)$-formulas proceeds in two steps. For a given target logic, the translation includes a fixed part called $\mathit{Struct}_{\blacklozenge}$, that enforces constraints on the structure of the model. The second part, called $\mathsf{Tr}_{\blacklozenge}$, is defined inductively on $\mathsf{ML}(\textcircled{r}, \textcircled{k})$-formulas, and uses the structure provided by $\mathit{Struct}_{\blacklozenge}$ to simulate the $\textcircled{r}$ and $\textcircled{k}$ operators.

Sabotage Logic

Local Sabotage. In the translation to local sabotage logic, the $\mathit{Struct}_{\langle \mathsf{sb} \rangle}$ subformula should ensure that every state of the model can be memorized using the expressivity of $\langle \mathsf{sb} \rangle$. This operator changes the point of evaluation after deleting an edge. To compensate for this, the $\mathit{Struct}_{\langle \mathsf{sb} \rangle}$ formula guarantees that every state has an edge that is deleted when the state is memorized, and a second edge back to the original state to ensure that evaluation can continue at the correct state. We use a spy point s to ensure this structure. The idea is illustrated in the following image.



We need to ensure that every satisfiable formula of $\mathsf{ML}(\textcircled{r}, \textcircled{k})$ is translated into a satisfiable formula (and vice-versa, if the translated formula is satisfiable, then the original formula is satisfiable, too). The image above shows an intended model for the translated formula $\tau_{\langle \mathsf{sb} \rangle}(\varphi)$. Intuitively, bold edges and arrows correspond to the model of φ. The complete translation is given in Definition 8. Here we discuss in detail how it works.

$\mathit{Struct}_{\langle \mathsf{sb} \rangle}$ adds a spy state with symmetric edges between itself and all other states. In particular, (1) in Definition 8 ensures that the evaluation state satisfies s and that it is irreflexive, and (2) guarantees that its immediate successors reach a state where s holds. Formulas (3) and (4) ensure that this state is the original s state. They work together as follows: (3) makes $\Box\Diamond s$ true in any s-state reachable in two steps, and by deleting the traversed edges we avoid a cycle of size two between this s-state and an immediate successor of the evaluation state, distinguishing the original s-state from any other s-state reachable in two steps. (4) then traverses one edge, deletes the next one, and reaches a state where s implies $\Diamond\Box\neg s$. This contradicts (3), unless we have arrived in the original s state. Formulas (5), (6) and (7) mimic (2), (3) and (4), but for edges which are removed twice. Observe that (6) now avoids a cycle of size three between any other s-state reachable in two steps and an immediate successor of the evaluation state. Finally, (8) and (9) ensure that the evaluation state is indeed a spy state, i.e., that it is linked to every other state of the input model.

$\mathsf{Tr}_{\langle \mathsf{sb} \rangle}$ starts by placing the translation $(\ )'$ of φ in a successor of the evaluation state. Boolean cases are obvious. For the diamond case, $\Diamond\psi$ is satisfied if there is a successor v where ψ holds, but we must ensure that v is not the spy state. For $(ⓡ\psi)'$, we do a round-trip of sabotaging from the current state to the spy state. Note that after reaching the spy state an edge does come back to the same state where it came from, since the only accessible state where $\neg\Diamond s$ holds is the one we are memorizing. For $(ⓚ)'$, we check whether there is an edge pointing to some s-state.

Definition 8. *Define* $\tau_{\langle \mathsf{sb} \rangle}(\varphi) = \mathit{Struct}_{\langle \mathsf{sb} \rangle} \wedge \mathsf{Tr}_{\langle \mathsf{sb} \rangle}(\varphi)$*, where:*

$$
\begin{aligned}
\mathit{Struct}_{\langle \mathsf{sb} \rangle} = {} & s \wedge \Box\neg s && (1)\\
& \wedge\ \Box\Diamond s && (2)\\
& \wedge\ [\mathsf{sb}][\mathsf{sb}](s \rightarrow \Box\Diamond s) && (3)\\
& \wedge\ \Box[\mathsf{sb}](s \rightarrow \Diamond\Box\neg s) && (4)\\
& \wedge\ \Box\Box(\neg s \rightarrow \Diamond s) && (5)\\
& \wedge\ \Box[\mathsf{sb}](s \rightarrow [\mathsf{sb}](\Box\neg s \rightarrow \Box\Box(s \rightarrow \Box\Diamond s))) && (6)\\
& \wedge\ \Box[\mathsf{sb}](s \rightarrow \Box(\Box\neg s \rightarrow \Box\Box(s \rightarrow \Diamond\Box\neg s))) && (7)\\
& \wedge\ \Box\Box\Box(s \rightarrow \Box\Diamond s) && (8)\\
& \wedge\ \Box\Box[\mathsf{sb}](s \rightarrow \Diamond\Box\neg s) && (9)
\end{aligned}
$$

$\mathsf{Tr}_{\langle \mathsf{sb} \rangle}(\varphi) = \Diamond(\varphi)'$, *with:*

$$
\begin{aligned}
(p)' &= p \quad \text{for } p \in \mathsf{PROP} \text{ appearing in } \varphi\\
(ⓚ)' &= \neg\Diamond s\\
(\neg\psi)' &= \neg(\psi)'\\
(\psi \wedge \chi)' &= (\psi)' \wedge (\chi)'\\
(\Diamond\psi)' &= \Diamond(\neg s \wedge (\psi)')\\
(ⓡ\psi)' &= \langle \mathsf{sb} \rangle(s \wedge \langle \mathsf{sb} \rangle(\neg\Diamond s \wedge (\psi)'))
\end{aligned}
$$

Proposition 1. *If* $\langle W, R, V\rangle, w \models \mathit{Struct}_{\langle \mathsf{sb} \rangle}$*, then for every state* $v \in W \setminus \{w\}$ *there exists exactly one state* v' *such that* $(v, v'), (v', v) \in R$ *and* $v' \in V(s)$*.*

Lemma 1. *Let* φ *be an* $\mathsf{ML}(ⓡ, ⓚ)$*-formula in PNF that does not contain the propositional symbol* s*. Then,* φ *is satisfiable iff* $\tau_{\langle \mathsf{sb} \rangle}(\varphi)$ *is satisfiable.*

Proof. (⇐) Suppose $\langle W, R, V\rangle, s \models \tau_{\langle \mathsf{sb}\rangle}(\varphi)$. Let $W' = W \backslash V(s)$, $R' = R \cap (W' \times W')$ and $V'(p) = V(p) \cap W'$ for all $p \in \mathsf{PROP}$. By definition of $\mathsf{Tr}_{\langle \mathsf{sb}\rangle}$ there is $w' \in W'$ such that $(s, w') \in R$ and $\langle W, R, V\rangle, w' \models (\varphi)'$.

Now, let ψ be a sub-formula of φ, $v \in W'$, $S \subseteq W'$ and $R_S = R \backslash \{(v, s), (s, v) \mid v \in S\}$. We prove by structural induction on ψ that $\langle W', R', V', S\rangle, v \models \psi$ if, and only if, $\langle W, R_S, V\rangle, v \models (\psi)'$. In particular, this will prove that $\langle W', R', V', \emptyset\rangle, w' \models \varphi$ if, and only if, $\langle W, R, V\rangle, w' \models (\varphi)'$.

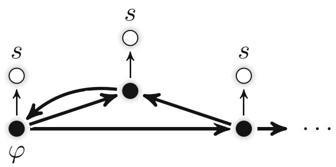
The propositional, Boolean and modal cases are trivial. For $\psi = ⓚ$, we should prove that $\langle W', R', V', S\rangle, v \models ⓚ$ if, and only if, $\langle W, R_S, V\rangle, v \models \neg\Diamond s$. However this is immediate by definition of S and R_S and Proposition 1.

For the last case, consider $\psi = \neg ⓚ \wedge ⓡ\chi$ (remember that formulas are in PNP), so we should prove that $\langle W', R', V', S\rangle, v \models \neg ⓚ \wedge ⓡ\chi$ if, and only if, $\langle W, R_S, V\rangle, v \models \Diamond s \wedge \langle \mathsf{sb}\rangle(s \wedge \langle \mathsf{sb}\rangle(\neg\Diamond s \wedge (\chi)'$. Again, the equivalence is immediate by Proposition 1.

(⇒) Suppose $\langle W, R, V, \emptyset\rangle, w \models \varphi$. We build a model for $\tau_{\langle \mathsf{sb}\rangle}(\varphi)$ by adding the necessary parts to this model, that are, the spy state and the round-trip paths. Define $\langle W', R', V'\rangle$ as follows. Let $s \notin W$ some state, $W' = W \cup \{s\}$, $R' = R \cup \{(x, s), (s, x) \mid x \in W\}$, $V'(s) = \{s\}$ and $V'(p) = V(p)$ for $p \in \mathsf{PROP} \backslash \{s\}$. By construction, $\langle W', R', V'\rangle, s \models \mathit{Struct}_{\langle \mathsf{sb}\rangle}$, so Proposition 1 holds. We prove that for all ψ sub-formula of φ, $v \in W$, $S \subseteq W$ and $R'_S = R' \backslash \{(x, s), (s, x) \mid x \in S\}$, $\langle W, R, V, S\rangle, v \models \psi$ iff $\langle W', R'_S, V'\rangle, v \models (\psi)'$. This can be done by structural induction on ψ using Proposition 1. This proves that $\langle W, R, V, \emptyset\rangle, w \models \varphi$ iff $\langle W', R', V'\rangle, s \models \tau_{\langle \mathsf{sb}\rangle}(\varphi)$, so $\tau_{\langle \mathsf{sb}\rangle}(\varphi)$ is satisfiable.

Global Sabotage. In [16] it is shown that multimodal sabotage logic is undecidable via a reduction of the Post Correspondence Problem. The present proof extends this result to the monomodal case via a reduction of the satisfiability problem of the memory logic $\mathsf{ML}(ⓡ, ⓚ)$. The notation $\Box^i\varphi$ is defined as $\Box^0\varphi = \varphi$ and $\Box^{n+1}\varphi = \Box\Box^n\varphi$.

One piece of data needed to build $\tau_{\langle \mathsf{gsb}\rangle}(\varphi)$ is the modal depth of the input formula ($\mathsf{md}(\varphi)$). Up to the depth indicated by this value, $\mathit{Struct}_{\langle \mathsf{gsb}\rangle}(\varphi)$ adds to every state a transition to some state where s holds (In fact, this latter state can be shared among several states of the input model.) It is as if each state of the input model had a flag that could be turned on to identify the state. Thus, remembering some state is simulated with $\mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(ⓡ)$ by deleting the edge between the state and its s-successor. For $\mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(ⓚ)$, we check whether the current state has an s-successor. The idea is illustrated in the following image.



Definition 9. *Define* $\tau_{\langle \mathsf{gsb}\rangle}(\varphi) = \mathit{Struct}_{\langle \mathsf{gsb}\rangle}(\varphi) \wedge \mathsf{Tr}_{\langle \mathsf{gsb}\rangle}(\varphi)$, *where:*

$$\mathit{Struct}_{\langle \mathsf{gsb}\rangle}(\varphi) = \neg s \wedge \bigwedge_{0 \leq i \leq \mathsf{md}(\varphi)} \Box^i(\neg s \rightarrow \Diamond s)$$

$$\begin{aligned}
\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(p) &= p \quad \textit{for } p \in \mathsf{PROP} \textit{ appearing in } \varphi \\
\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\circledk) &= \neg\Diamond s \\
\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\neg\psi) &= \neg\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\psi) \\
\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\psi\wedge\chi) &= \mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\psi)\wedge\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\chi) \\
\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\Diamond\psi) &= \Diamond(\neg s\wedge\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\psi)) \\
\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\circledr\psi) &= \langle\mathsf{gsb}\rangle(\neg\Diamond s\wedge\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\psi))
\end{aligned}$$

Proposition 2. *Let $dist(a,b)$ the minimal number of R-steps to reach some state b from some state a. Let φ some memory logic formula. If $\langle W,R,V\rangle,w\models Struct_{\langle\mathsf{gsb}\rangle}(\varphi)$, then for all $x\in W$ such that $dist(w,x)\le md(\varphi)$, x has a successor where s holds.*

Proposition 3. *If $\langle W,R,V\rangle,w\models\Diamond s\wedge\langle\mathsf{gsb}\rangle\neg\Diamond s$, then w has one and only one successor where s holds.*

Lemma 2. *Let φ be an $\mathsf{ML}(\circledr,\circledk)$-formula in PNF that does not contain the propositional symbol s. Then, φ is satisfiable iff $\tau_{\langle\mathsf{gsb}\rangle}(\varphi)$ is satisfiable.*

Proof. ($\Leftarrow$) Suppose $\langle W,R,V\rangle,w\models\tau_{\langle\mathsf{gsb}\rangle}(\varphi)$. Let $W'=W\backslash V(s)$, $R'=R\cap(W'\times W')$, and $V'(p)=V(p)\cap W'$ for $p\in\mathsf{PROP}\setminus\{s\}$. We should prove that for all ψ sub-formula of φ of modal depth $\mathsf{md}(\psi)\le\mathsf{md}(\varphi)-dist(w,v)$, $v\in W'$ accessible from w within $\mathsf{md}(\varphi)$ steps, $S\subseteq W'$, and $R_S=R\setminus\{(x,y)\mid |x\in S,y\in V(s)\}$, then $\langle W',R',V',S\rangle,v\models\psi$ iff $\langle W,R_S,V\rangle,v\models\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\psi)$.

The proof is by structural induction on ψ. The non-memory cases are easy. For the $\circledk$ case, we should show that $\langle W',R',V',S\rangle,v\models\circledk$ iff $\langle W,R_S,V\rangle,v\models\neg\Diamond s$, this is immediate by Proposition 2 and the definitions of S and R_S.

Then for the remaining case, we have to show that $\langle W',R',V',S\rangle,v\models\neg\circledk\wedge\circledr\chi$ iff $\langle W,R_S,V\rangle,v\models\Diamond s\wedge\langle\mathsf{gsb}\rangle(\neg\Diamond s\wedge\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\chi))$, which can be proved using the definition of $\models$, IH and Proposition 3.

($\Rightarrow$) Suppose $\langle W,R,V,\emptyset\rangle,w\models\varphi$. Let $s\notin W$. Define $\langle W',R',V'\rangle$, where $W'=W\cup\{s\}$, $R'=R\cup\{(v,s)\mid v\in W\}$, $V'(s)=\{s\}$, and $V'(p)=V(p)$, for $p\in\mathsf{PROP}$ appearing in φ. It is easy to check that $\langle W',R',V'\rangle,w\models Struct_{\langle\mathsf{gsb}\rangle}(\varphi)$, hence Proposition 2 holds. Then, let us prove that for all ψ sub-formula of φ of modal depth $\mathsf{md}(\psi)\le\mathsf{md}(\varphi)-dist(w,v)$, $v\in W$ accessible from w within $\mathsf{md}(\varphi)$ steps, $S\subseteq W$ and $R'_S=R'\setminus\{(x,s)\mid x\in S\}$, we have the equivalence $\langle W,R,V,S\rangle,v\models\psi$ iff $\langle W',R'_S,V'\rangle,v\models\mathsf{Tr}_{\langle\mathsf{gsb}\rangle}(\psi)$. This is done by structural induction on ψ. For the case $\circledk$ the equivalence is immediate, and for the case $\neg\circledk\wedge\circledr\chi$, Proposition 3 provides the equivalence needed.

Bridge Logic

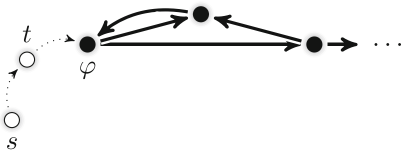
Local Bridge. For local bridge logic, we use a spy state that is initially disconnected from the input model. When some state should be memorized, the spy state gets connected (in both directions) to it. This construction is quite special since we do not have pre-built gadgets in the input model, as they get built on demand.

Let us first show the following result, that enables us to force the evaluation state to be the only one in the model to satisfy s:

Lemma 3. *Let* $\varphi = s \wedge \Box\bot \wedge [\mathsf{br}](s \to [\mathsf{br}]\neg s)$. *If* $\mathsf{M}, w \models \varphi$, *then* w *is the only state in the model* M *where* s *holds.*

Proof. First, w obviously satisfies s and does not have any successor. Now, we have $\mathsf{M}, w \models [\mathsf{br}](s \to [\mathsf{br}]\neg s)$. In particular this means that $\mathsf{M}^+_{ww}, w \models s \to [\mathsf{br}]\neg s$, hence $\mathsf{M}^+_{ww}, w \models [\mathsf{br}]\neg s$. Since in M^+_{ww}, the state w is only connected to itself, this means that for all $v \neq w$, we have $\mathsf{M}^+_{ww,wv}, v \models \neg s$, this also means that $\mathsf{M}, v \not\models s$ for all $v \neq w$.

For Bridge Logics, $Struct_{\langle \mathsf{br} \rangle}$ adds to the input model a spy state in which s holds. By Lemma 3, (1) in Definition 10 ensures that the evaluation state has no successor and is the only state in the model where s holds. And (2) ensures that there are no edges from $\neg s$-states (anywhere in the model) to the spy state. The idea is illustrated in the following image, where t is a propositional symbol used in $\mathsf{Tr}_{\langle \mathsf{br} \rangle}(\varphi)$ and dotted lines represent edges created with the $\langle \mathsf{br} \rangle$ operator.



Definition 10. *Define* $\tau_{\langle \mathsf{br} \rangle}(\varphi) = Struct_{\langle \mathsf{br} \rangle} \wedge \mathsf{Tr}_{\langle \mathsf{br} \rangle}(\varphi)$, *where:*

$$\begin{aligned} Struct_{\langle \mathsf{br} \rangle} = {} & s \wedge \Box\bot \wedge [\mathsf{br}](s \to [\mathsf{br}]\neg s) && (1) \\ & \wedge [\mathsf{br}](\neg s \to \Box\neg s) && (2) \end{aligned}$$

$\mathsf{Tr}_{\langle \mathsf{br} \rangle}(\varphi) = \langle \mathsf{br} \rangle(\neg s \wedge t \wedge \langle \mathsf{br} \rangle(\neg s \wedge \neg t \wedge (\varphi)'))$, *with:*

$$\begin{aligned} (p)' &= p \quad \textit{for } p \in \mathsf{PROP} \textit{ appearing in } \varphi \\ (\circledk)' &= \Diamond s \\ (\neg\psi)' &= \neg(\psi)' \\ (\psi \wedge \chi)' &= (\psi)' \wedge (\chi)' \\ (\Diamond\psi)' &= \Diamond(\neg s \wedge \neg t \wedge (\psi)') \\ (\circledr\,\psi)' &= \langle \mathsf{br} \rangle(s \wedge \langle \mathsf{br} \rangle(\neg s \wedge \Diamond s \wedge (\psi)')) \end{aligned}$$

$\mathsf{Tr}_{\langle \mathsf{br} \rangle}(\varphi)$ first creates two edges until a $\neg s$-state, where the translation of φ holds. For $\mathsf{Tr}_{\langle \mathsf{br} \rangle}(\circledr)$ we do a round-trip of bridging from the current state to the spy state. Note that the second part of this round-trip has to be from the spy state to the remembered state, since it is the only way to satisfy $\langle \mathsf{br} \rangle(\Diamond s)$. Also note that this would not work if the s state was directly connected to the input model; this is why we use the intermediate t-state. For $\mathsf{Tr}_{\langle \mathsf{br} \rangle}(\circledk)$ we check whether there is an edge to a state where s holds.

Proposition 4. *Let* $\langle W, R, V \rangle$ *a model such that there is a unique state* s *where* s *holds, there is no state* $x \in W$ *such that* $(x, s) \in R$, *and there is a component*

$C \subseteq W$ such that $s \notin C$ and for all $y \in C$, $(s,y) \notin R$. Let $S \subseteq C$ and $R_S = R \cup \{(x,s),(s,x) \mid x \in S\}$.

Then in the model $\langle W, R_S, V\rangle$, evaluating the formula $\langle \mathsf{br}\rangle(s \wedge \langle \mathsf{br}\rangle \Diamond s)$ at some state $y \in C \setminus S$ changes the evaluation state to s, then again to the same state y adding the edges (y,s) and (s,y) to the relation.

Lemma 4. *Let φ be an $\mathsf{ML}(ⓡ, ⓚ)$-formula in PNF that does not contain the propositional symbols s and t. Then, φ is satisfiable iff $\tau_{\langle \mathsf{br}\rangle}(\varphi)$ is satisfiable.*

Proof. ($\Leftarrow$) Suppose $\langle W, R, V\rangle, s \models \tau_{\langle \mathsf{br}\rangle}(\varphi)$. Define $\mathsf{M}' = \langle W', R', V', \emptyset\rangle$ with $W' = (W \setminus V(s)) \setminus V(t)$, $R' = R \cap (W' \times W')$, and $V'(p) = V(p) \cap W'$ for all $p \in \mathsf{PROP}$. By definition of $\mathsf{Tr}_{\langle \mathsf{br}\rangle}$ there is $w' \in W'$ such that $s \neq w'$ and $\langle W, R, V\rangle, w' \models (\varphi)'$.

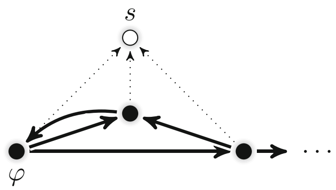
Let ψ a sub-formula of φ, $v \in W'$, $S \subseteq W'$, and $R_S = R \cup \{(x,s),(x,v) \mid x \in S\}$, then we will prove that $\langle W', R', V', S\rangle, v \models \psi$ iff $\langle W, R_S, V\rangle, v \models (\psi)'$.

We prove it by structural induction on ψ. For the $\neg ⓚ \wedge ⓡ\chi$ case, suppose $\langle W', R', V', S\rangle, v \models \neg ⓚ \wedge ⓡ\chi$. By definition, this is equivalent to $\langle W', R', V', S \cup \{v\}\rangle, v \models \chi$ with $v \notin S$, Then, by definition of R_S and inductive hypothesis we get $\langle W, (R_S)^+_{\{(v,s),(s,v)\}}, V\rangle, s \models (\chi)'$, with $(v,s) \notin R_S$ and $(s,v) \notin R_S$. By Proposition 4, this is equivalent to $\langle W, R_S, V\rangle, v \models \neg\Diamond s \wedge \langle \mathsf{br}\rangle(s \wedge \langle \mathsf{br}\rangle(\Diamond s \wedge (\chi)'))$. thus we have $\langle W, R_S, V\rangle, v \models (\neg ⓚ \wedge ⓡ\chi)'$.

($\Rightarrow$) Suppose $\langle W, R, V, \emptyset\rangle, w \models \varphi$. Let $s, t \notin W$. Define $\mathsf{M}' = \langle W', R, V'\rangle$ such that $W' = W \cup \{s,t\}$, $V'(s) = \{s\}$, $V'(t) = \{t\}$ and $V'(p) = V(p)$ for $p \in \mathsf{PROP}$ appearing in φ. We can easily check that $\langle W', R, V'\rangle, s \models Struct_{\langle \mathsf{br}\rangle}$, and we can also check by structural induction on φ that $\langle W, R, V, S\rangle, w \models \varphi$ iff $\langle W', R_S, V'\rangle, s \models \mathsf{Tr}_{\langle \mathsf{br}\rangle}(\varphi)$, where $R_S = R \cup \{(v,s),(s,v) \mid v \in S\}$.

Global Bridge. The global bridge operator is able to add edges in the model. This is why, to mark some state, we use this operator to add an edge to some s-state. Then, we enforce that the initial model does not have any reachable s-state.

Here $Struct_{\langle \mathsf{gbr}\rangle}(\varphi)$ ensures that no state of the input model has s-successors. Storing a state in the memory is simulated by creating an edge to an s-state, and checking whether the current state of evaluation is in the memory is simulated by checking the presence of an s-successor. Observe that we could have either one state where s holds or (possibly) different s-states for each state of the input model.



Definition 11. *Define* $\tau_{\langle \mathsf{gbr}\rangle}(\varphi) = Struct_{\langle \mathsf{gbr}\rangle}(\varphi) \wedge \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\varphi)$*, where:*

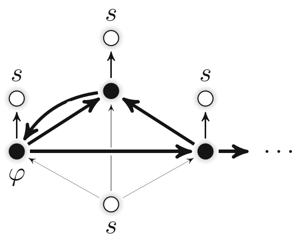
$$\begin{array}{ll} Struct_{\langle \mathsf{gbr}\rangle}(\varphi) = & \bigwedge_{0\leq i\leq \mathsf{md}(\varphi)+1} \Box^i \neg s \\ \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(p) & = p \;\; \text{for } p \in \mathsf{PROP} \text{ appearing in } \varphi \\ \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(Ⓚ) & = \Diamond s \\ \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\neg\psi) & = \neg \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\psi) \\ \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\psi \wedge \chi) & = \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\psi) \wedge \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\chi) \\ \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\Diamond\psi) & = \Diamond(\neg s \wedge \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\psi)) \\ \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(Ⓡ\psi) & = \langle \mathsf{gbr}\rangle(\Diamond s \wedge \mathsf{Tr}_{\langle \mathsf{gbr}\rangle}(\psi)) \end{array}$$

Lemma 5. *Let* φ *be an* $\mathsf{ML}(Ⓡ, Ⓚ)$*-formula in PNF that does not contain the propositional symbol* s*. Then,* φ *is satisfiable iff* $\tau_{\langle \mathsf{gbr}\rangle}(\varphi)$ *is satisfiable.*

Swap Logic

Local Swap. We introduce a new version of the translation given in [4] that uses only one propositional symbol. The idea is that we have each state pointing to some states called *switch states*, and memorizing a state is represented by swapping such edges. Then, no edge pointing to a switch means that the state has been memorized. We use the notation $\Box^{(n)}\varphi$ for $\bigwedge_{1\leq i\leq n} \Box^i\varphi$.

In this case $Struct_{\langle \mathsf{sw}\rangle}$ adds "switch states", which are in one-to-one correspondence with the states of the input model, together with a spy state. By (2) in Definition 12, each $\neg s$-state at one, two and three steps from the evaluation state, has a unique dead-end successor where s holds (switch state). By (3) and (4), switch states (corresponding to states at distance 1, 2 and 3) can be reached from the evaluation state by a unique path. (5) makes the evaluation state a spy state. All these conjuncts together ensure that switch states are independent one from another. The idea is illustrated in the following image.



Definition 12. *Define* $\tau_{\langle \mathsf{sw}\rangle} = Struct_{\langle \mathsf{sw}\rangle} \wedge \mathsf{Tr}_{\langle \mathsf{sw}\rangle}(\varphi)$*, where:*

$$\begin{array}{ll} Struct_{\langle \mathsf{sw}\rangle} = & \\ \quad s \wedge \Box\neg s & (1) \\ \quad \wedge\ \Box^{(3)}(\neg s \rightarrow Uniq) & (2) \\ \quad \wedge\ \Box[\mathsf{sw}](s \rightarrow \Box\Box\Box(s \rightarrow \Box\bot) & (3) \\ \quad \wedge\ \Box\Box[\mathsf{sw}](s \rightarrow \Box\Box\Box(s \rightarrow \Box\bot) & (4) \\ \quad \wedge\ [\mathsf{sw}][\mathsf{sw}](\neg s \rightarrow \langle \mathsf{sw}\rangle(s \wedge \Diamond((\Box\neg s) \rightarrow \Diamond\Diamond(s \wedge \Diamond\neg\Diamond s)))) & (5) \end{array}$$

$$Uniq = \Diamond(s \wedge \Box\bot) \wedge [\mathsf{sw}](s \rightarrow \Box\neg\Diamond s)$$

$\mathsf{Tr}_{\langle \mathsf{sw} \rangle}(\varphi) = \Diamond(\varphi)'$, *with:*

$$\begin{array}{ll} (p)' & = p \quad for\ p \in \mathsf{PROP}\ appearing\ in\ \varphi \\ (ⓚ)' & = \neg\Diamond s \\ (\neg\psi)' & = \neg(\psi)' \\ (\psi \wedge \chi)' & = (\psi)' \wedge (\chi)' \\ (\Diamond\psi)' & = \Diamond(\neg s \wedge (\psi)') \\ (ⓡ\psi)' & = \langle \mathsf{sw} \rangle (s \wedge \Diamond(\psi)') \end{array}$$

For $\mathsf{Tr}_{\langle \mathsf{sw} \rangle}(ⓡ\varphi)$ we traverse and swap the edge between the current state and its switch state, and come back to the same state. For $\mathsf{Tr}_{\langle \mathsf{sw} \rangle}(ⓚ)$, we check whether the current state has not an edge to its switch state.

Proposition 5. *Let $\langle W, R, V \rangle, s \models Struct_{\langle \mathsf{sw} \rangle}$, $W' = W \setminus V(s)$ and $S \subseteq W'$. Then $T = \{(v', v) \mid v \in S \wedge (v, v') \in R \wedge v' \in V(s)\}$ is a bijection.*

Lemma 6. *Let φ be an $\mathsf{ML}(ⓡ, ⓚ)$-formula in PNF that does not contain the propositional symbol s. Then, φ is satisfiable iff $\tau_{\langle \mathsf{sw} \rangle}(\varphi)$ is satisfiable.*

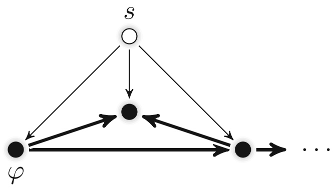
Proof. ($\Leftarrow$) From a pointed model $\langle W, R, V \rangle, w$ of $\tau_{\langle \mathsf{sw} \rangle}(\varphi)$ we can extract a pointed model $\langle W', R', V', \emptyset \rangle, w'$ satisfying φ following the same definition as in the proof of Lemma 1.

For all ψ sub-formula of φ, $v \in W'$, $S \subseteq W'$, $T = \{(v', v) \mid v \in S \wedge (v, v') \in R \wedge v' \in V(s)\}$ and $R_S = (R \backslash T^{-1}) \cup T$, we will prove that $\langle W', R', V', S \rangle, v \models \psi$ if, and only if, $\langle W, R_S, V \rangle, v \models (\psi)'$.

We do it by structural induction on ψ. We prove the $\neg ⓚ \wedge ⓡ\chi$ case. Suppose $\langle W', R', V', S \rangle, v \models \neg ⓚ \wedge ⓡ\chi$. Then by definition, $v \notin S$ and $\langle W', R', V', S \cup \{v\} \rangle, v \models \chi$, and by Proposition 5, we have $(v, v') \in R_S$ for a unique $v' \in V(s)$. Then, by definition of R_S and inductive hypothesis we get $\langle W, (R_S)^*_{v'v}, V \rangle, v \models (\chi)'$. By definition of $\models$ and by Proposition 5, $\langle W, (R_S)^*_{v'v}, V \rangle, v' \models s \wedge \Diamond(\chi)'$, and again, $\langle W, R_S, V \rangle, v \models \Diamond s \wedge \langle \mathsf{sw} \rangle (s \wedge \Diamond(\chi)')$, thus we have, equivalently, $\langle W, R_S, V \rangle, v \models (\neg ⓚ \wedge ⓡ\chi)'$.

($\Rightarrow$) Suppose $\langle W, R, V, \emptyset \rangle, w \models \varphi$. Let sw be a bijective function between W and a set U such that $U \cap W = \emptyset$, and $s \notin U \cup W$. Define $\mathsf{M}' = \langle W', R', V' \rangle$ such that $W' = W \cup \{s\} \cup U$, $R' = R \cup \{(s, w) \mid w \in W\} \cup \{(w, sw(w)) \mid w \in W\}$, $V'(s) = \{s\} \cup U$, and $V'(p) = V(p)$ for $p \in \mathsf{PROP}$ appearing in φ. It is easy to check that $\langle W', R', V' \rangle, s \models Struct_{\langle \mathsf{sw} \rangle}$, in particular, Proposition 5 is relevant. Then, we can easily prove that for all ψ sub-formula of φ, $v \in W$, $S \subseteq W$, $T = \{(sw(v), v) \mid v \in S\}$ and $R'_S = (R' \backslash T^{-1}) \cup T$, we have the equivalence $\langle W, R, V, S \rangle, v \models \psi$ iff $\langle W', R'_S, V' \rangle, v \models (\psi)'$. This is done by structural induction on ψ.

Global Swap. The global swap operator is able to change the direction of some edge in the model. In particular, we are interested in the ability to swap, for some state, an incoming edge (undetectable for the basic modal logic) into an outgoing edge. This is why this translation is similar to the one of global bridge logic. Initially, the model does not have any reachable state where s holds. As for global sabotage and global bridge, there may be many states where s holds in the model with edges to states of the input model. The idea is illustrated in the following image, where only one s state is shown.



Definition 13. *Define* $\tau_{\langle \mathsf{gsw} \rangle}(\varphi) = \mathit{Struct}_{\langle \mathsf{gsw} \rangle}(\varphi) \wedge \mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\varphi)$*, where:*

$$\mathit{Struct}_{\langle \mathsf{gsw} \rangle}(\varphi) = \bigwedge_{0 \leq i \leq \mathsf{md}(\varphi)+1} \Box^i \neg s$$

$$\begin{array}{ll}
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(p) & = p \quad \mathit{for}\ p \in \mathsf{PROP}\ \mathit{appearing\ in}\ \varphi \\
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(ⓚ) & = \Diamond s \\
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\neg \psi) & = \neg \mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\psi) \\
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\psi \wedge \chi) & = \mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\psi) \wedge \mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\chi) \\
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\Diamond \psi) & = \Diamond(\neg s \wedge \mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\psi)) \\
\mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(ⓡ \psi) & = \langle \mathsf{gsw} \rangle(\Diamond s \wedge \mathsf{Tr}_{\langle \mathsf{gsw} \rangle}(\psi))
\end{array}$$

Proposition 6. *Let* $\langle W, R, V \rangle, w \models \neg \Diamond s \wedge \langle \mathsf{gsw} \rangle \Diamond s$*. Then, by the semantics of the global swap operator, there exists a state* $v \in W \setminus \{w\}$ *such that* $(v, w) \in R$ *and* $v \in V(s)$*.*

Lemma 7. *Let* φ *be an* $\mathsf{ML}(ⓡ, ⓚ)$*-formula in PNF that does not contain the propositional symbol* s*. Then,* φ *is satisfiable iff* $\tau_{\langle \mathsf{gsw} \rangle}(\varphi)$ *is satisfiable.*

5 Conclusions

We exploited the similarities between memory logic and relation-changing logics to obtain simple and non-redundant undecidability proofs. We first presented an undecidability result for memory logics in the monomodal case, by adapting the proof introduced in [18] for $\mathcal{ALC}\mathsf{self}$. Then, we presented translations from the satisfiability problem of monomodal memory logics to all six relation-changing modal logics. Both results combined show undecidability of the satisfiability problem for relation-changing modal logics in a very simple way. These results complete the picture of the computational behaviour of relation-changing logics, given that we already know that model checking for them is PSPACE-complete [2,4,5,15].

This high complexity of the logics is a consequence of the degree of liberty we give to the operators. By replacing arbitrary modifications with conditional modifications (i.e., according to a pre- and a post-condition) it is possible to decrease the complexity and get decidable logics (e.g., as in [9,10]).

A related problem is the one of finite satisfiability. Indeed, for many applications of dynamic epistemic logic, we are only interested in looking for finite models. Finite satisfiability is known to be undecidable for multimodal global sabotage logic [20], and decidable for monomodal local sabotage and local swap logics [6]. It remains to see the status of this problem for all remaining cases.

Acknowledgements. This work was partially supported by grant ANPCyT-PICT-2013-2011, STIC-AmSud "Foundations of Graph Structured Data (FoG)", SeCyT-UNC, the Laboratoire International Associé "INFINIS", and the EU Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant No. 690974, project MIREL: MIning and REasoning with Legal texts.

References

1. Areces, C.: Hybrid logics: the old and the new. In: Proceedings of LogKCA 2007, pp. 15–29 (2007)
2. Areces, C., Fervari, R., Hoffmann, G.: Moving arrows and four model checking results. In: Ong, L., de Queiroz, R. (eds.) WoLLIC 2012. LNCS, vol. 7456, pp. 142–153. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32621-9_11
3. Areces, C., Fervari, R., Hoffmann, G.: Tableaux for relation-changing modal logics. In: Fontaine, P., Ringeissen, C., Schmidt, R.A. (eds.) FroCoS 2013. LNCS (LNAI), vol. 8152, pp. 263–278. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40885-4_19
4. Areces, C., Fervari, R., Hoffmann, G.: Swap logic. Logic J. IGPL **22**(2), 309–332 (2014)
5. Areces, C., Fervari, R., Hoffmann, G.: Relation-changing modal operators. Logic J. IGPL **23**(4), 601–627 (2015)
6. Areces, C., Fervari, R., Hoffmann, G., Martel, M.: Relation-changing logics as fragments of hybrid logics. In: Cantone, D., Delzanno, G. (eds.) Proceedings of the Seventh International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2016, Italy. EPTCS, vol. 226, pp. 16–29 (2016)
7. Areces, C., Figueira, D., Figueira, S., Mera, S.: The expressive power of memory logics. Rev. Symb. Logic **4**(2), 290–318 (2011)
8. Areces, C., ten Cate, B.: Hybrid logics. In: Blackburn, P., Wolter, F., van Benthem, J. (eds.) Handbook of Modal Logic, pp. 821–868. Elsevier (2007)
9. Areces, C., van Ditmarsch, H., Fervari, R., Schwarzentruber, F.: Logics with copy and remove. In: Kohlenbach, U., Barceló, P., de Queiroz, R. (eds.) WoLLIC 2014. LNCS, vol. 8652, pp. 51–65. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44145-9_4
10. Areces, C., van Ditmarsch, H., Fervari, R., Schwarzentruber, F.: The modal logic of copy and remove. To Appear in Information and Computation (2015). Special Issue of WoLLIC 2014
11. Aucher, G., Schwarzentruber, F.: On the complexity of dynamic epistemic logic. In: Proceedings of TARK 2013, Chennai, India, January 2013
12. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, New York (2001)
13. Blackburn, P., Seligman, J.: Hybrid languages. J. Logic Lang. Inf. **4**(3), 251–272 (1995)

14. Blackburn, P., van Benthem, J.: Modal logic: a semantic perspective. In: Handbook of Modal Logic, pp. 1–84. Elsevier (2007)
15. Fervari, R.: Relation-Changing Modal Logics. Ph.D. thesis, Universidad Nacional de Córdoba, Argentina (2014)
16. Löding, C., Rohde, P.: Model checking and satisfiability for sabotage modal logic. In: Pandya, P.K., Radhakrishnan, J. (eds.) FSTTCS 2003. LNCS, vol. 2914, pp. 302–313. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-24597-1_26
17. Lutz, C.: Complexity and succinctness of public announcement logic. In: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2006, New York, NY, USA, pp. 137–143 (2006)
18. Marx, M.: Narcissists, stepmothers and spies. In: Proceedings of DL 2002, vol. 53. CEUR (2002)
19. Mera, S.: Modal memory logics. Ph.D. thesis, Université Henri Poincaré, Nancy, France and Universidad de Buenos Aires, Argentina (2009)
20. Rohde, P.: On games and logics over dynamically changing structures. Ph.D. thesis, RWTH Aachen (2006)
21. Benthem, J.: An essay on sabotage and obstruction. In: Hutter, D., Stephan, W. (eds.) Mechanizing Mathematical Reasoning. LNCS (LNAI), vol. 2605, pp. 268–276. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-32254-2_16
22. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Synthese Library. Springer, Dordrecht (2007). https://doi.org/10.1007/978-1-4020-5839-4

Axiomatization and Computability of a Variant of Iteration-Free PDL with Fork

Philippe Balbiani(✉) and Joseph Boudou

Institut de recherche en informatique de Toulouse,
CNRS — Université de Toulouse, 118 route de Narbonne,
31062 Toulouse Cedex 9, France
Philippe.Balbiani@irit.fr

Abstract. We devote this paper to the axiomatization and the computability of PDL_0^{Δ}—a variant of iteration-free PDL with fork.

Keywords: Iteration-free PDL · Fork · Axiomatization
Computability

1 Introduction

Propositional dynamic logic (PDL) is an applied non-classical logic designed for reasoning about the behaviour of programs [10]. The definition of its syntax is based on the idea of associating with each program α of some programming language the modal operator $[\alpha]$, formulas of the form $[\alpha]\phi$ being read "every execution of the program α from the present state leads to a state bearing the formula ϕ". Completeness and decidability results for the standard version of PDL in which programs are built up from program variables and tests by means of the operations of composition, union and iteration are given in [15,16]. A number of interesting variants have been obtained by extending or restricting the syntax or the semantics of PDL in different ways [7,9,14,18].

Some of these variants extend the ordinary semantics of PDL by considering sets W of states structured by means of a function $\star$ from the set of all pairs of states into the set of all states [5,11–13]: the state x is the result of applying the function $\star$ to the states y, z iff the information concerning x can be separated in a first part concerning y and a second part concerning z. The binary function $\star$ considered in [5,11] has its origin in the addition of an extra binary operation of fork denoted ∇ in relation algebras: in [5, Sect. 2], whenever x and y are related via R and z and t are related via S, states in $x \star z$ and states in $y \star t$ are related via $R \nabla S$ whereas in [11, Chap. 1], whenever x and y are related via R and x and z are related via S, x and states in $y \star z$ are related via $R \nabla S$.

This addition of fork in relation algebras gives rise to a variant of PDL which includes the program operation of fork denoted Δ. In this variant, for all programs α and β, one can use the modal operator $[\alpha \Delta \beta]$, formulas of the form $[\alpha \Delta \beta]\phi$ being read "every execution in parallel of the programs α and β from the

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 17–34, 2018.
https://doi.org/10.1007/978-3-319-73579-5_2

present state leads to a state bearing the formula ϕ". The binary operation of fork ∇ considered in Benevides *et al.* [5, Sect. 2] gives rise to $PRSPDL$, a variant of PDL with fork whose axiomatization is still open. We devote this paper to the axiomatization and the computability of $PDL_0^{\varDelta}$, a variant of iteration-free PDL with fork whose semantics is based on the interpretation of the binary operation of fork ∇ considered in Frias [11, Chap. 1].

The difficulty in axiomatizing or deciding $PRSPDL$ or $PDL_0^{\varDelta}$ originates in the fact that the program operations of fork considered above are not modally definable in the ordinary language of PDL. We overcome this difficulty by means of tools and techniques developed in [1,3,4]. Our results are based on the following: although fork is not modally definable, it becomes definable in a modal language strengthened by the introduction of propositional quantifiers. Instead of using axioms to define the program operation of fork in the language of PDL enlarged with propositional quantifiers, we add an unorthodox rule of proof that makes the canonical model standard for the program operation of fork and we use large programs for the proof of the Truth Lemma.

We will first present the syntax (Sect. 2) and the semantics (Sect. 3) of $PDL_0^{\varDelta}$ and continue with results concerning the expressivity of $PDL_0^{\varDelta}$ (Sect. 4), the axiomatization/completeness of $PDL_0^{\varDelta}$ (Sects. 5 and 6) and the decidability of $PDL_0^{\varDelta}$ (Sect. 7). We assume the reader is at home with tools and techniques in modal logic and dynamic logic. For more on this, see [6,15]. The proofs of our results can be found in [2].

2 Syntax

This section presents the syntax of $PDL_0^{\varDelta}$. As usual, we will follow the standard rules for omission of the parentheses.

Definition 1 (Programs and formulas). *The set PRG of all programs and the set FRM of all formulas are inductively defined as follows:*

- $\alpha, \beta ::= a \mid (\alpha; \beta) \mid (\alpha \varDelta \beta) \mid \phi?$;
- $\phi, \psi ::= p \mid \bot \mid \neg\phi \mid (\phi \vee \psi) \mid [\alpha]\phi \mid (\phi \circ \psi) \mid (\phi \triangleright \psi) \mid (\phi \triangleleft \psi)$;

where a ranges over a countably infinite set of program variables and p ranges over a countably infinite set of propositional variables.

We will use $\alpha, \beta, \ldots$ for programs and $\phi, \psi, \ldots$ for formulas. The Boolean constructs for formulas are defined as usual. A number of other constructs for formulas can be defined in terms of the primitive ones as follows.

Definition 2 (Abbreviations). *The modal constructs for formulas $\langle\cdot\rangle\cdot$, $(\cdot\bar{\circ}\cdot)$, $(\cdot\bar{\triangleright}\cdot)$ and $(\cdot\bar{\triangleleft}\cdot)$ are defined as follows: $\langle\alpha\rangle\phi ::= \neg[\alpha]\neg\phi$; $(\phi\bar{\circ}\psi) ::= \neg(\neg\phi \circ \neg\psi)$; $(\phi\bar{\triangleright}\psi) ::= \neg(\neg\phi \triangleright \neg\psi)$; $(\phi\bar{\triangleleft}\psi) ::= \neg(\neg\phi \triangleleft \neg\psi)$. Moreover, for all formulas ϕ, let $\phi^0 ::= \neg\phi$ and $\phi^1 ::= \phi$.*

It is well worth noting that programs and formulas are finite strings of symbols coming from a countable alphabet. It follows that there are countably many programs and countably many formulas. The construct $\cdot;\cdot$ comes from the class of algebras of binary relations [19]: the program $\alpha;\beta$ firstly executes α and secondly executes β. As for the construct $\cdot\Delta\cdot$, it comes from the class of proper fork algebras [11, Chap. 1]: the program $\alpha\Delta\beta$ performs a kind of parallel execution of α and β. The construct $[\cdot]\cdot$ comes from the language of PDL [10,15]: the formula $[\alpha]\phi$ says that "every execution of α from the present state leads to a state bearing the information ϕ". As for the constructs $\cdot\circ\cdot$, $\cdot \triangleright \cdot$ and $\cdot \triangleleft \cdot$, they come from the language of conjugated arrow logic [8,17]: the formula $\phi \circ \psi$ says that "the present state is a combination of states bearing the information ϕ and ψ", the formula $\phi \triangleright \psi$ says that "the present state can be combined to its left with a state bearing the information ϕ giving us a state bearing the information ψ" and the formula $\phi \triangleleft \psi$ says that "the present state can be combined to its right with a state bearing the information ψ giving us a state bearing the information ϕ".

Example 1. The formula $[a\Delta b](p \circ q)$ says that "the parallel execution of a and b from the present state always leads to a state resulting from the combination of states bearing the information p and q".

Obviously, programs are built up from program variables and tests by means of the constructs $\cdot;\cdot$ and $\cdot\Delta\cdot$. Let $\alpha(\phi_1?, \ldots, \phi_n?)$ be a program with $(\phi_1?, \ldots, \phi_n?)$ a sequence of some of its tests. The result of the replacement of $\phi_1?, \ldots, \phi_n?$ in their places with other tests $\psi_1?, \ldots, \psi_n?$ is another program which will be denoted $\alpha(\psi_1?, \ldots, \psi_n?)$. Now, we introduce the function f from the set of all programs into itself defined as follows.

Definition 3 (Test insertion). *Let f be the function from the set of all programs into itself inductively defined as follows:*

- $f(a) = a$;
- $f(\alpha;\beta) = f(\alpha);\top?;f(\beta)$;
- $f(\alpha\Delta\beta) = (f(\alpha);\top?)\Delta(f(\beta);\top?)$;
- $f(\phi?) = \phi?$.

Example 2. If $\alpha = a\Delta b$, $f(\alpha) = (a;\top?)\Delta(b;\top?)$.

Now, we introduce parametrized actions and admissible forms.

Definition 4 (Parametrized actions and admissible forms). *The set PAR of all parametrized actions and the set ADM of all admissible forms are inductively defined as follows:*

- $\breve{\alpha}, \breve{\beta} ::= (\breve{\alpha};\beta) \mid (\alpha;\breve{\beta}) \mid (\breve{\alpha}\Delta\beta) \mid (\alpha\Delta\breve{\beta}) \mid \neg\breve{\phi}?$;
- $\breve{\phi}, \breve{\psi} ::= \sharp \mid [\breve{\alpha}]\bot \mid (\breve{\phi}\bar{\circ}\psi) \mid (\phi\bar{\circ}\breve{\psi}) \mid (\breve{\phi}\bar{\triangleright}\psi) \mid (\phi\bar{\triangleright}\breve{\psi}) \mid (\breve{\phi}\bar{\triangleleft}\psi) \mid (\phi\bar{\triangleleft}\breve{\psi})$;

where $\sharp$ is a new propositional variable, α, β range over PRG and ϕ, ψ range over FRM.

We will use $\breve{\alpha}, \breve{\beta}, \ldots$ for parametrized actions and $\breve{\phi}, \breve{\psi}, \ldots$ for admissible forms. It is well worth noting that parametrized actions and admissible forms are finite strings of symbols coming from a countable alphabet. It follows that there are countably many parametrized actions and countably many admissible forms. Remark that in each expression $\breve{exp}$ (a parametrized action, or an admissible form), $\sharp$ has a unique occurrence. The result of the replacement of $\sharp$ in its place in $\breve{exp}$ with a formula ϕ is an expression which will be denoted $\breve{exp}(\phi)$.

Example 3. For all programs α, $\alpha;\neg[\neg\sharp?]\bot?$ is a parametrized action whereas for all formulas ϕ, $\phi\bar{\circ}[\neg\sharp?]\bot$ is an admissible form.

3 Semantics

Our task is now to present the semantics of PDL_0^{Δ}.

Definition 5 (Frames). *A frame is a 3-tuple $\mathcal{F} = (W, R, \star)$ where W is a nonempty set of states, R is a function from the set of all program variables into the set of all binary relations between states and $\star$ is a function from the set of all pairs of states into the set of all sets of states.*

We will use $x, y, \ldots$ for states. The set W of states in a frame $\mathcal{F} = (W, R, \star)$ is to be regarded as the set of all possible states in a computation process, the function R from the set of all program variables into the set of all binary relations between states associates with each program variable a the binary relation $R(a)$ on W with $xR(a)y$ meaning that "y can be reached from x by performing program variable a" and the function $\star$ from the set of all pairs of states into the set of all sets of states associates with each pair (x, y) of states the subset $x \star y$ of W with $z \in x \star y$ meaning that "z is a combination of x and y".

Definition 6 (Valuations and models). *A model on the frame $\mathcal{F} = (W, R, \star)$ is a 4-tuple $\mathcal{M} = (W, R, \star, V)$ where V is a valuation on $\mathcal{F}$, i.e. a function from the set of all propositional variables into the set of all sets of states.*

In the model $\mathcal{M} = (W, R, \star, V)$, the valuation V associates with each propositional variable p the subset $V(p)$ of W with $x \in V(p)$ meaning that "propositional variable p is true at state x in $\mathcal{M}$". We now define the property "state y can be reached from state x by performing program α in $\mathcal{M}$"—in symbols $xR_{\mathcal{M}}(\alpha)y$—and the property "formula ϕ is true at state x in $\mathcal{M}$"—in symbols $x \in V_{\mathcal{M}}(\phi)$.

Definition 7 (Accessibility via programs and truth of formulas). *In model $\mathcal{M} = (W, R, \star, V)$, $R_{\mathcal{M}} : \alpha \mapsto R_{\mathcal{M}}(\alpha) \subseteq W \times W$ and $V_{\mathcal{M}} : \phi \mapsto V_{\mathcal{M}}(\phi) \subseteq W$ are inductively defined as follows:*

– *$xR_{\mathcal{M}}(a)y$ iff $xR(a)y$;*
– *$xR_{\mathcal{M}}(\alpha;\beta)y$ iff there exists $z \in W$ such that $xR_{\mathcal{M}}(\alpha)z$ and $zR_{\mathcal{M}}(\beta)y$;*

– $xR_{\mathcal{M}}(\alpha\Delta\beta)y$ *iff there exists* $z, t \in W$ *such that* $xR_{\mathcal{M}}(\alpha)z$, $xR_{\mathcal{M}}(\beta)t$ *and* $y \in z \star t$;
– $xR_{\mathcal{M}}(\phi?)y$ *iff* $x = y$ *and* $y \in V_{\mathcal{M}}(\phi)$;
– $x \in V_{\mathcal{M}}(p)$ *iff* $x \in V(p)$;
– $x \notin V_{\mathcal{M}}(\bot)$;
– $x \in V_{\mathcal{M}}(\neg\phi)$ *iff* $x \notin V_{\mathcal{M}}(\phi)$;
– $x \in V_{\mathcal{M}}(\phi \vee \psi)$ *iff* $x \in V_{\mathcal{M}}(\phi)$, *or* $x \in V_{\mathcal{M}}(\psi)$;
– $x \in V_{\mathcal{M}}([\alpha]\phi)$ *iff for all* $y \in W$, *if* $xR_{\mathcal{M}}(\alpha)y$, $y \in V_{\mathcal{M}}(\phi)$;
– $x \in V_{\mathcal{M}}(\phi \circ \psi)$ *iff there exists* $y, z \in W$ *such that* $x \in y \star z$, $y \in V_{\mathcal{M}}(\phi)$ *and* $z \in V_{\mathcal{M}}(\psi)$;
– $x \in V_{\mathcal{M}}(\phi \rhd \psi)$ *iff there exists* $y, z \in W$ *such that* $z \in y \star x$, $y \in V_{\mathcal{M}}(\phi)$ *and* $z \in V_{\mathcal{M}}(\psi)$;
– $x \in V_{\mathcal{M}}(\phi \lhd \psi)$ *iff there exists* $y, z \in W$ *such that* $y \in x \star z$, $y \in V_{\mathcal{M}}(\phi)$ *and* $z \in V_{\mathcal{M}}(\psi)$.

It follows that

Proposition 1. *Let* $\mathcal{M} = (W, R, \star, V)$ *be a model. For all* $x \in W$, *we have:* $x \in V_{\mathcal{M}}(\langle\alpha\rangle\phi)$ *iff there exists* $y \in W$ *such that* $xR_{\mathcal{M}}(\alpha)y$ *and* $y \in V_{\mathcal{M}}(\phi)$; $x \in V_{\mathcal{M}}(\phi\bar{\circ}\psi)$ *iff for all* $y, z \in W$, *if* $x \in y \star z$, $y \in V_{\mathcal{M}}(\phi)$, *or* $z \in V_{\mathcal{M}}(\psi)$; $x \in V_{\mathcal{M}}(\phi\bar{\rhd}\psi)$ *iff for all* $y, z \in W$, *if* $z \in y \star x$, $y \in V_{\mathcal{M}}(\phi)$, *or* $z \in V_{\mathcal{M}}(\psi)$; $x \in V_{\mathcal{M}}(\phi\bar{\lhd}\psi)$ *iff for all* $y, z \in W$, *if* $y \in x \star z$, $y \in V_{\mathcal{M}}(\phi)$, *or* $z \in V_{\mathcal{M}}(\psi)$.

Example 4. Let $\mathcal{M} = (W, R, \star, V)$ be the model defined by:

– $W = \{x, y, z, t\}$;
– $R(a) = \{(x, y)\}$, $R(b) = \{(x, z)\}$, otherwise R is the empty function;
– $y \star z = \{t\}$, otherwise $\star$ is the empty function;
– $V(p) = \{y\}$, $V(q) = \{z\}$, otherwise V is the empty function.

Obviously, $xR_{\mathcal{M}}(a\Delta b)t$ and $t \in V_{\mathcal{M}}(p \circ q)$. Hence, $x \in V_{\mathcal{M}}(\langle a\Delta b\rangle(p \circ q))$.

We now define the property "state z can be reached from state x by performing parametrized action $\breve{\alpha}$ via state y in $\mathcal{M}$"—in symbols $xR_{\mathcal{M}}(\breve{\alpha}, y)z$—and the property "admissible form $\breve{\phi}$ is true at state x via state y in $\mathcal{M}$"—in symbols $x \in V_{\mathcal{M}}(\breve{\phi}, y)$.

Definition 8 (Accessibility via parametrized actions and truth of admissible forms). *In model* $\mathcal{M} = (W, R, \star, V)$, $R_{\mathcal{M}} : (\breve{\alpha}, y) \mapsto R_{\mathcal{M}}(\breve{\alpha}, y) \subseteq W \times W$ *and* $V_{\mathcal{M}} : (\breve{\phi}, y) \mapsto V_{\mathcal{M}}(\breve{\phi}, y) \subseteq W$ *are inductively defined as follows:*

– $xR_{\mathcal{M}}(\breve{\alpha};\beta, y)z$ *iff there exists* $t \in W$ *such that* $xR_{\mathcal{M}}(\breve{\alpha}, y)t$ *and* $tR_{\mathcal{M}}(\beta)z$;
– $xR_{\mathcal{M}}(\alpha;\breve{\beta}, y)z$ *iff there exists* $t \in W$ *such that* $xR_{\mathcal{M}}(\alpha)t$ *and* $tR_{\mathcal{M}}(\breve{\beta}, y)z$;
– $xR_{\mathcal{M}}(\breve{\alpha}\Delta\beta, y)z$ *iff there exists* $t, u \in W$ *such that* $xR_{\mathcal{M}}(\breve{\alpha}, y)t$, $xR_{\mathcal{M}}(\beta)u$ *and* $z \in t \star u$;
– $xR_{\mathcal{M}}(\alpha\Delta\breve{\beta}, y)z$ *iff there exists* $t, u \in W$ *such that* $xR_{\mathcal{M}}(\alpha)t$, $xR_{\mathcal{M}}(\breve{\beta}, y)u$ *and* $z \in t \star u$;
– $xR_{\mathcal{M}}(\neg\breve{\phi}?, y)z$ *iff* $x = z$ *and* $z \in V_{\mathcal{M}}(\breve{\phi}, y)$;

- $x \in V_{\mathcal{M}}(\sharp, y)$ *iff* $x = y$;
- $x \in V_{\mathcal{M}}([\breve{\alpha}]\bot, y)$ *iff there exists* $z \in W$ *such that* $xR_{\mathcal{M}}(\breve{\alpha}, y)z$;
- $x \in V_{\mathcal{M}}(\breve{\phi}\bar{\circ}\psi, y)$ *iff there exists* $z, t \in W$ *such that* $x \in z \star t$, $z \in V_{\mathcal{M}}(\breve{\phi}, y)$ *and* $t \notin V_{\mathcal{M}}(\psi)$;
- $x \in V_{\mathcal{M}}(\phi\bar{\circ}\breve{\psi}, y)$ *iff there exists* $z, t \in W$ *such that* $x \in z \star t$, $z \notin V_{\mathcal{M}}(\phi)$ *and* $t \in V_{\mathcal{M}}(\breve{\psi}, y)$;
- $x \in V_{\mathcal{M}}(\breve{\phi}\bar{\triangleright}\psi, y)$ *iff there exists* $z, t \in W$ *such that* $t \in z \star x$, $z \in V_{\mathcal{M}}(\breve{\phi}, y)$ *and* $t \notin V_{\mathcal{M}}(\psi)$;
- $x \in V_{\mathcal{M}}(\phi\bar{\triangleright}\breve{\psi}, y)$ *iff there exists* $z, t \in W$ *such that* $t \in z \star x$, $z \notin V_{\mathcal{M}}(\phi)$ *and* $t \in V_{\mathcal{M}}(\breve{\psi}, y)$;
- $x \in V_{\mathcal{M}}(\breve{\phi}\bar{\triangleleft}\psi, y)$ *iff there exists* $z, t \in W$ *such that* $z \in x \star t$, $z \in V_{\mathcal{M}}(\breve{\phi}, y)$ *and* $t \notin V_{\mathcal{M}}(\psi)$;
- $x \in V_{\mathcal{M}}(\phi\bar{\triangleleft}\breve{\psi}, y)$ *iff there exists* $z, t \in W$ *such that* $z \in x \star t$, $z \notin V_{\mathcal{M}}(\phi)$ *and* $t \in V_{\mathcal{M}}(\breve{\psi}, y)$;

It follows that

Proposition 2. *Let* $\mathcal{M} = (W, R, \star, V)$ *be a model. Let* $\breve{\alpha}$ *be a parametrized action. For all* $x, z \in W$, *the following conditions are equivalent:* $xR_{\mathcal{M}}(\breve{\alpha}(\phi))z$; *there exists* $y \in W$ *such that* $xR_{\mathcal{M}}(\breve{\alpha}, y)z$ *and* $y \notin V_{\mathcal{M}}(\phi)$. *Let* $\breve{\phi}$ *be an admissible form. For all* $x \in W$, *the following conditions are equivalent:* $x \in V_{\mathcal{M}}(\breve{\phi}(\psi))$; *for all* $y \in W$, *if* $x \in V_{\mathcal{M}}(\breve{\phi}, y)$, $y \in V_{\mathcal{M}}(\psi)$.

The concept of validity is defined in the usual way as follows.

Definition 9 (Validity). *We shall say that a formula* ϕ *is valid in a model* $\mathcal{M}$, *in symbols* $\mathcal{M} \models \phi$, *iff* $V_{\mathcal{M}}(\phi) = W$. *A formula* ϕ *is said to be valid in a frame* $\mathcal{F}$, *in symbols* $\mathcal{F} \models \phi$, *iff for all models* $\mathcal{M}$ *on* $\mathcal{F}$, $\mathcal{M} \models \phi$. *We shall say that a formula* ϕ *is valid in a class* $\mathcal{C}$ *of frames, in symbols* $\mathcal{C} \models \phi$, *iff for all frames* $\mathcal{F}$ *in* $\mathcal{C}$, $\mathcal{F} \models \phi$.

For technical reasons, we now consider three particular classes of frames.

Definition 10 (Separated, deterministic or serial frames). *A frame* $\mathcal{F} = (W, R, \star)$ *is said to be separated iff for all* $x, y, z, t, u \in W$, *if* $u \in x \star y$ *and* $u \in z \star t$, $x = z$ *and* $y = t$. *We shall say that a frame* $\mathcal{F} = (W, R, \star)$ *is deterministic iff for all* $x, y, z, t \in W$, *if* $z \in x \star y$ *and* $t \in x \star y$, $z = t$. *A frame* $\mathcal{F} = (W, R, \star)$ *is said to be serial iff for all* $x, y \in W$, *there exists* $z \in W$ *such that* $z \in x \star y$.

In separated frames, there is at most one way to decompose a given state; in deterministic frames, there is at most one way to combine two given states; in serial frames, it is always possible to combine two given states. Frias [11, Chap. 1] only considers separated, deterministic and serial frames. Here are some valid formulas and admissible rules of proof.

Proposition 3 (Validity). *The following formulas are valid in the class of all frames:*

$(A1)$ $[\alpha](\phi \to \psi) \to ([\alpha]\phi \to [\alpha]\psi)$;
$(A2)$ $\langle \alpha; \beta \rangle \phi \leftrightarrow \langle \alpha \rangle \langle \beta \rangle \phi$;
$(A3)$ $\langle \alpha \Delta \beta \rangle \phi \to \langle \alpha \rangle ((\phi \wedge \psi) \lhd \top) \vee \langle \beta \rangle (\top \rhd (\phi \wedge \neg \psi))$;
$(A4)$ $\langle \phi ? \rangle \psi \leftrightarrow \phi \wedge \psi$;
$(A5)$ $(\phi \to \psi) \bar{\circ} \chi \to (\phi \bar{\circ} \chi \to \psi \bar{\circ} \chi)$;
$(A6)$ $\phi \bar{\circ} (\psi \to \chi) \to (\phi \bar{\circ} \psi \to \phi \bar{\circ} \chi)$;
$(A7)$ $(\phi \to \psi) \bar{\rhd} \chi \to (\phi \bar{\rhd} \chi \to \psi \bar{\rhd} \chi)$;
$(A8)$ $\phi \bar{\rhd} (\psi \to \chi) \to (\phi \bar{\rhd} \psi \to \phi \bar{\rhd} \chi)$;
$(A9)$ $(\phi \to \psi) \bar{\lhd} \chi \to (\phi \bar{\lhd} \chi \to \psi \bar{\lhd} \chi)$;
$(A10)$ $\phi \bar{\lhd} (\psi \to \chi) \to (\phi \bar{\lhd} \psi \to \phi \bar{\lhd} \chi)$;
$(A11)$ $\phi \circ \neg (\phi \rhd \neg \psi) \to \psi$;
$(A12)$ $\phi \rhd \neg (\phi \circ \neg \psi) \to \psi$;
$(A13)$ $\neg (\neg \phi \lhd \psi) \circ \psi \to \phi$;
$(A14)$ $\neg (\neg \phi \circ \psi) \lhd \psi \to \phi$;
$(A15)$ $[(\alpha; \phi ?) \Delta (\beta; \psi ?)](\phi \circ \psi)$;
$(A16)$ $\langle \alpha(\phi ?) \rangle \psi \to \langle \alpha((\phi \wedge \chi)?) \rangle \psi \vee \langle \alpha((\phi \wedge \neg \chi)?) \rangle \psi$;
$(A17)$ $\langle f(\alpha) \rangle \phi \leftrightarrow \langle \alpha \rangle \phi$.

Proposition 4 (Validity). *The following formula is valid in the class of all separated frames:*

$(A18)$ $p \circ q \to (p \bar{\circ} \bot) \wedge (\bot \bar{\circ} q)$.

Proposition 5 (Admissibility). *The following rules of proof preserve validity in the class of all frames:*

(MP) *from* ϕ *and* $\phi \to \psi$, *infer* ψ;
(N) *from* ϕ, *infer* $[\alpha]\phi$; *from* ϕ, *infer* $\phi \bar{\circ} \psi$; *from* ϕ, *infer* $\psi \bar{\circ} \phi$.

$(A1)$ is the distribution axiom of PDL, $(A2)$ is the composition axiom, $(A4)$ is the test axiom, $(A5)$–$(A10)$ are the distribution axioms of conjugated arrow logic and $(A11)$–$(A14)$ are the tense axioms of conjugated arrow logic whereas $(A3)$ and $(A15)$–$(A18)$ are axioms concerning specific properties of the program operation of fork or the constructs $\cdot \circ \cdot$, $\cdot \rhd \cdot$ and $\cdot \lhd \cdot$. (MP) is the modus ponens rule of proof and (N) is the necessitation rule of proof. They are probably familiar to the reader. As for the following rule of proof, it concerns specific properties of the program operation of fork and the constructs $\cdot \rhd \cdot$ and $\cdot \lhd \cdot$.

Proposition 6 (Admissibility). *The following rule of proof preserves validity in the class of all separated frames:*

(FOR) *from* $\{\check{\phi}(\langle \alpha \rangle ((\psi \wedge p) \lhd \top) \vee \langle \beta \rangle (\top \rhd (\psi \wedge \neg p))) :$ p *is a propositional variable*$\}$, *infer* $\check{\phi}(\langle \alpha \Delta \beta \rangle \psi)$.

There is an important point we should make: (FOR) is an infinitary rule of proof, i.e. it has an infinite set of formulas as preconditions. In some ways, it is similar to the rule for intersection from [3,4].

4 Expressivity

This section studies the expressivity of PDL_0^{Δ}.

Definition 11 (Modal definability). *Let $\mathcal{C}$ be a class of frames. We shall say that $\mathcal{C}$ is modally definable by the formula ϕ iff for all frames $\mathcal{F}$, $\mathcal{F}$ is in $\mathcal{C}$ iff $\mathcal{F} \models \phi$.*

The following propositions show elementary classes of frames that are modally definable.

Proposition 7. *The elementary classes of frames defined by the first-order sentences in the hereunder table are modally definable by the associated formulas.*

1.	$\forall x\ \exists y\ y \in x \star x$	$\langle \top ? \Delta \top ? \rangle \top$
2.	$\forall x\ \forall y\ \forall z\ (y \in x \star x \wedge z \in x \star x \rightarrow y = z)$	$\langle \top ? \Delta \top ? \rangle p \rightarrow [\top ? \Delta \top ?] p$
3.	$\forall x\ \forall y\ (y \in x \star x \rightarrow x \in x \star y)$	$p \rightarrow [\top ? \Delta \top ?](p \triangleright p)$
4.	$\forall x\ \forall y\ (y \in x \star x \rightarrow x \in y \star x)$	$p \rightarrow [\top ? \Delta \top ?](p \triangleleft p)$
5.	$\forall x\ \forall y\ \forall z\ (z \in x \star y \leftrightarrow z \in y \star x)$	$p \circ q \leftrightarrow q \circ p$
6.	$\forall x\ \exists y\ \exists z\ x \in y \star z$	$\top \circ \top$
7.	$\forall x\ \exists y\ \exists z\ y \in z \star x$	$\top \triangleright \top$
8.	$\forall x\ \exists y\ \exists z\ z \in x \star y$	$\top \triangleleft \top$
9.	$\forall x\ \forall y\ \forall z\ \forall t\ (t \in (x \star y) \star z \leftrightarrow t \in x \star (y \star z))$	$(p \circ q) \circ r \leftrightarrow p \circ (q \circ r)$
10.	$\forall x\ \forall y\ \forall z\ x \notin y \star z$	$\bot \bar{\circ} \bot$

Proposition 8. *The class of all separated frames is modally definable by the formula $p \circ q \rightarrow (p \bar{\circ} \bot) \wedge (\bot \bar{\circ} q)$.*

The following proposition shows an elementary class of frames that is not modally definable.

Proposition 9. *The class of all deterministic frames is not modally definable.*

As for the class of all serial frames, its modal definability is still open. In other respect, the formula $\langle \phi ? \rangle \psi \leftrightarrow \phi \wedge \psi$, being valid in the class of all frames, seems to indicate that for all formulas, there exists an equivalent test-free formula. It is interesting to observe that this assertion is false.

Proposition 10. *For all test-free formulas ϕ, $\langle \top ? \Delta \top ? \rangle \top \leftrightarrow \phi$ is not valid in the class of all separated deterministic frames.*

The following proposition illustrates the fact that the program operation of fork cannot be defined from the fork-free fragment of the language.

Proposition 11. *Let a be a program variable. For all fork-free formulas ϕ, $\langle a\Delta a\rangle\top \leftrightarrow \phi$ is not valid in the class of all separated deterministic frames.*

The following proposition illustrates the fact that, in the presence of propositional quantifiers, the program operation of fork becomes definable from the fork-free fragment of the language in the class of all separated frames.

Proposition 12. *Let $\mathcal{M} = (W, R, \star, V)$ be a separated model and $x \in W$. For all admissible forms $\breve{\phi}$, for all programs α, β, for all formulas ψ and for all propositional variables p, if p does not occur in $\breve{\phi}, \alpha, \beta, \psi$, the following conditions are equivalent:* (1) *$x \in V_{\mathcal{M}}(\breve{\phi}(\langle\alpha\Delta\beta\rangle\psi))$;* (2) *for all $V' : q \mapsto V'(q) \subseteq W$, if $V' \sim_p V$, $x \in V_{(W,R,\star,V')}(\breve{\phi}(\langle\alpha\rangle((\psi \wedge p) \lhd \top) \vee \langle\beta\rangle(\top \rhd (\psi \wedge \neg p))))$.*

More precisely, in the presence of propositional quantifiers, the formulas $\langle\alpha\Delta\beta\rangle\phi$ and $\forall p(\langle\alpha\rangle((\phi\wedge p)\lhd\top)\vee\langle\beta\rangle(\top\rhd(\phi\wedge\neg p)))$ are logically equivalent in the class of all separated frames. The implication $\langle\alpha\Delta\beta\rangle\phi \to \forall p(\langle\alpha\rangle((\phi \wedge p) \lhd \top) \vee \langle\beta\rangle(\top\rhd(\phi\wedge\neg p)))$ can be expressed without propositional quantifiers by formulas: $\langle\alpha\Delta\beta\rangle\phi \to \langle\alpha\rangle((\phi\wedge\psi)\lhd\top)\vee\langle\beta\rangle(\top\rhd(\phi\wedge\neg\psi)))$. See axiom $(A3)$ in Proposition 3. As for the implication $\forall p(\langle\alpha\rangle((\phi \wedge p) \lhd \top) \vee \langle\beta\rangle(\top \rhd (\phi \wedge \neg p))) \to \langle\alpha\Delta\beta\rangle\phi$, it can be expressed by a rule of proof. The simplest form of such a rule of proof is: from $\{\langle\alpha\rangle((\phi \wedge p) \lhd \top) \vee \langle\beta\rangle(\top \rhd (\phi \wedge \neg p)) :$ p is a propositional variable$\}$, infer $\langle\alpha\Delta\beta\rangle\phi$. See Proposition 6.

5 Axiom System

We now define PDL_0^Δ.

Definition 12 **(PDL_0^Δ).** *Let PDL_0^Δ be the least set of formulas that contains all instances of propositional tautologies, that contains the formulas* $(A1)$–$(A18)$ *considered in Propositions 3 and 4 and that is closed under the rules of proof* (MP), (N) *and* (FOR) *considered in Propositions 5 and 6.*

It is easy to establish the soundness for PDL_0^Δ:

Proposition 13 **(Soundness for PDL_0^Δ).** *Let ϕ be a formula. If $\phi \in PDL_0^\Delta$, ϕ is valid in the class of all separated frames.*

The completeness for PDL_0^Δ is more difficult to establish and we defer proving it till next section. In the meantime, it is well worth noting that for all separated models $\mathcal{M} = (W, R, \star, V)$ and for all $x \in W$, $\{\phi :\ x \in V_{\mathcal{M}}(\phi)\}$ is a set of formulas that contains PDL_0^Δ and that is closed under the rule of proof (MP). Now, we introduce theories.

Definition 13 **(Theories).** *A set S of formulas is said to be a theory iff $PDL_0^\Delta \subseteq S$ and S is closed under the rules of proof* (MP) *and* (FOR).

We will use $S, T, \ldots$ for theories. Obviously, the least theory is PDL_0^Δ and the greatest theory is the set of all formulas. Not surprisingly, we have

Lemma 1. *Let S be a theory. The following conditions are equivalent: S is equal to the set of all formulas; there exists a formula ϕ such that $\phi \in S$ and $\neg\phi \in S$; $\bot \in S$.*

Referring to Lemma 1, we define what it means for a theory to be consistent.

Definition 14 (Consistency of theories). *We shall say that a theory S is consistent iff for all formulas ϕ, $\phi \notin S$, or $\neg\phi \notin S$.*

By Lemma 1, there is only one inconsistent theory: the set of all formulas. Now, we define what it means for a theory to be maximal.

Definition 15 (Maximality of theories). *A theory S is said to be maximal iff for all formulas ϕ, $\phi \in S$, or $\neg\phi \in S$.*

We will use the following lemma without explicit reference:

Lemma 2. *Let S be a maximal consistent theory. We have: $\bot \notin S$; for all formulas ϕ, $\neg\phi \in S$ iff $\phi \notin S$; for all formulas ϕ, ψ, $\phi \vee \psi \in S$ iff $\phi \in S$, or $\psi \in S$.*

To know more about theories, we need yet another definition.

Definition 16 (Operations on theories). *If α is a program, ϕ is a formula and S is a theory, let $[\alpha]S = \{\phi : [\alpha]\phi \in S\}$ and $S + \phi = \{\psi : \phi \rightarrow \psi \in S\}$.*

In the next lemmas, we summarize some properties of theories.

Lemma 3. *Let S be a theory. For all programs α and for all formulas ϕ, we have:* (1) $[\phi?]S = S + \phi$; (2) $[\alpha]S$ *is a theory;* (3) $S + \phi$ *is a theory;* (4) ϕ, $S + \phi$ *is the least theory containing S and ϕ;* (5) $S + \phi$ *is consistent iff $\neg\phi \notin S$.*

Lemma 4. *Let S be a theory. If S is consistent, for all formulas ϕ, $S + \phi$ is consistent, or there exists a formula ψ such that the following conditions are satisfied: $S + \psi$ is consistent; $\psi \rightarrow \neg\phi \in PDL_0^{\Delta}$; if ϕ is in the form $\breve{\chi}(\langle\alpha\Delta\beta\rangle\theta)$ of a conclusion of the rule of proof (FOR), there exists a propositional variable p such that $\psi \rightarrow \neg\breve{\chi}(\langle\alpha\rangle((\theta \wedge p) \triangleleft \top) \vee \langle\beta\rangle(\top \triangleright (\theta \wedge \neg p))) \in PDL_0^{\Delta}$.*

Now, we are ready for the Lindenbaum Lemma.

Lemma 5 (Lindenbaum Lemma). *Let S be a theory. If S is consistent, there exists a maximal consistent theory containing S.*

To define the canonical frame of PDL_0^{Δ} in next section, we need yet another definition.

Definition 17 (Composition of theories). *If S and T are theory, let $S \circ T = \{\phi \circ \psi : \phi \in S$ and $\psi \in T\}$.*

To end this section, we present useful results.

Lemma 6. *Let ϕ, ψ be formulas and $\otimes \in \{\circ, \triangleright, \triangleright\}$. For all maximal consistent theories S, if $\phi \otimes \psi \in S$, for all formulas χ, we have:* (1) *$(\phi \wedge \chi) \otimes \psi \in S$, or there exists a formula θ such that the following conditions are satisfied: $(\phi \wedge \theta) \otimes \psi \in S$; $\theta \rightarrow \neg\chi \in PDL_0^{\Delta}$; if χ is in the form $\breve{\tau}(\langle \alpha \Delta \beta \rangle \mu)$ of a conclusion of the rule of proof (FOR), there exists a propositional variable p such that $\theta \rightarrow \neg\breve{\tau}(\langle \alpha \rangle((\mu \wedge p) \triangleleft \top) \vee \langle \beta \rangle(\top \triangleright (\mu \wedge \neg p))) \in PDL_0^{\Delta}$;* (2) *$\phi \otimes (\psi \wedge \chi) \in S$, or there exists a formula θ such that the following conditions are satisfied: $\phi \otimes (\psi \wedge \theta) \in S$; $\theta \rightarrow \neg\chi \in PDL_0^{\Delta}$; if χ is in the form $\breve{\tau}(\langle \alpha \Delta \beta \rangle \mu)$ of a conclusion of the rule of proof (FOR), there exists a propositional variable p such that $\theta \rightarrow \neg\breve{\tau}(\langle \alpha \rangle((\mu \wedge p) \triangleleft \top) \vee \langle \beta \rangle(\top \triangleright (\mu \wedge \neg p))) \in PDL_0^{\Delta}$.*

Lemma 7. *Let ϕ, ψ be formulas. For all maximal consistent theories S, we have:* (1) *if $\phi \circ \psi \in S$, there exists maximal consistent theories T, U such that $T \circ U \subseteq S$, $\phi \in T$ and $\psi \in U$;* (2) *if $\phi \triangleright \psi \in S$, there exists maximal consistent theories T, U such that $T \circ S \subseteq U$, $\phi \in T$ and $\psi \in U$;* (3) *if $\phi \triangleleft \psi \in S$, there exists maximal consistent theories T, U such that $S \circ U \subseteq T$, $\phi \in T$ and $\psi \in U$.*

6 Completeness

Now, for the canonical frame of PDL_0^{Δ}.

Definition 18 (Canonical frame). *The canonical frame of PDL_0^{Δ} is the* 3*-tuple $\mathcal{F}_c = (W_c, R_c, \star_c)$ where W_c is the set of all maximal consistent theories, R_c is the function from the set of all program variables into the set of all binary relations between maximal consistent theories defined by $SR_c(a)T$ iff $[a]S \subseteq T$ and $\star_c$ is the function from the set of all pairs of maximal consistent theories into the set of all sets of maximal consistent theories defined by $U \in S \star_c T$ iff $S \circ T \subseteq U$.*

We show first that

Lemma 8. *$\mathcal{F}_c$ is separated.*

Now, for the canonical valuation of PDL_0^{Δ} and the canonical model of PDL_0^{Δ}.

Definition 19 (Canonical valuation and canonical model). *The canonical model of PDL_0^{Δ} is the* 4*-tuple $\mathcal{M}_c = (W_c, R_c, \star_c, V_c)$ where V_c is the canonical valuation of PDL_0^{Δ}, i.e. the function from the set of all propositional variables into the set of all sets of maximal consistent theories defined by $S \in V_c(p)$ iff $p \in S$.*

For the proof of the Truth Lemma, we have to consider large programs.

Definition 20 (Large programs). *The set of all large programs is inductively defined as follows:*

- $A ::= a \mid (A; B) \mid (A \Delta B) \mid \bar{S}?$;

where for all consistent theories S, $\bar{S}$ is a new symbol.

We will use $A, B, \ldots$ for large programs. Let us be clear that each large program is a finite string of symbols coming from an uncountable alphabet. It follows that there are uncountably many large programs. For convenience, we omit the parentheses in accordance with the standard rules. It is essential that large programs are built up from program variables and symbols for consistent theories by means of the operations ; and Δ. Let $A(\bar{S_1}?, \ldots, \bar{S_n}?)$ be a large program with $(\bar{S_1}, \ldots, \bar{S_n})$ a sequence of some of its symbols for consistent theories. The result of the replacement of $\bar{S_1}, \ldots, \bar{S_n}$ in their places with other symbols $\bar{T_1}, \ldots, \bar{T_n}$ for consistent theories is another large program which will be denoted $A(\bar{T_1}?, \ldots, \bar{T_n}?)$.

Definition 21 (Maximality of large programs). *A large program* $A(\bar{S_1}?, \ldots, \bar{S_n}?)$ *with* $(\bar{S_1}, \ldots, \bar{S_n})$ *the sequence of all its symbols for consistent theories will be defined to be maximal if the theories* $S_1, \ldots, S_n$ *are maximal.*

It appears that large programs, maximal, or not, can be associated with a set of programs.

Definition 22 (Kernel function). *The kernel function* $\ker: \ A \mapsto \ker(A) \subseteq PRG$ *is inductively defined as follows:*

- $\ker(a) = \{a\}$;
- $\ker(A; B) = \{\alpha; \beta : \ \alpha \in \ker(A)$ *and* $\beta \in \ker(B)\}$;
- $\ker(A \Delta B) = \{\alpha \Delta \beta : \ \alpha \in \ker(A)$ *and* $\beta \in \ker(B)\}$;
- $\ker(\bar{S}) = \{\phi? : \ \phi \in S\}$.

The following lemmas play an important role in the proof of the completeness for PDL_0^{Δ}.

Lemma 9. *Let* $\alpha(\phi?)$ *be a program. For all maximal consistent theories* S*, if* $\langle \alpha(\phi?) \rangle \top \in S$*, for all formulas* ψ*, we have:* $\langle \alpha((\phi \wedge \psi)?) \rangle \top \in S$*, or there exists a formula* χ *such that the following conditions are satisfied:* $\langle \alpha((\phi \wedge \chi)?) \rangle \top \in S$*;* $\chi \to \neg \psi \in PDL_0^{\Delta}$*; if* ψ *is in the form* $\breve{\theta}(\langle \beta \Delta \gamma \rangle \tau)$ *of a conclusion of the rule of proof* (FOR)*, there exists a propositional variable* p *such that* $\chi \to \neg \breve{\theta}(\langle \beta \rangle ((\tau \wedge p) \lhd \top) \vee \langle \gamma \rangle (\top \rhd (\tau \wedge \neg p))) \in PDL_0^{\Delta}$*.*

Lemma 10 (Diamond Lemma). *Let* α *be a program and* ϕ *be a formula. For all maximal consistent theories* S*, if* $[\alpha]\phi \notin S$*, there exists a maximal program* A *and there exists a maximal consistent theory* T *such that* $f(\alpha) \in \ker(A)$*, for all programs* β*, if* $\beta \in \ker(A)$*,* $[\beta]S \subseteq T$ *and* $\phi \notin T$*.*

With this established, we are ready for the Truth Lemma.

Lemma 11 (Truth Lemma). *Let* α *be a program. For all maximal consistent theories* S, T*, the following conditions are equivalent:* $SR_{\mathcal{M}_c}(\alpha)T$*; there exists a maximal program* A *such that* $f(\alpha) \in \ker(A)$ *and for all programs* β*, if* $\beta \in \ker(A)$*,* $[\beta]S \subseteq T$*. Let* ϕ *be a formula. For all maximal consistent theories* S*, the following conditions are equivalent:* $S \in V_{\mathcal{M}_c}(\phi)$*;* $\phi \in S$*.*

Now, we are ready for the completeness for PDL_0^{Δ}.

Proposition 14 (Completeness for PDL_0^{Δ}). *Let* ϕ *be a formula. If* ϕ *is valid in the class of all separated frames,* $\phi \in PDL_0^{\Delta}$*.*

7 Decidability

In this section, we prove that the logic completely axiomatized in the previous sections is decidable. We use the notation $\sim\phi$ which is defined by: $\sim\phi =$ if there exists a formula ψ such that $\phi = \neg\psi$ then ψ else $\neg\phi$. We use ν to denote an expression which may be either a program or a formula and $|\nu|$ to denote the number of occurrences of symbols in ν. The following size function provides a more semantical measure on programs.

Definition 23 (Size of programs). *Let* size *be the function from the set of all programs to* $\mathbb{N}$ *inductively defined as follows:*

- $\text{size}(\phi?) = 0$;
- $\text{size}(a) = 1$;
- $\text{size}(\alpha; \beta) = \text{size}(\alpha) + \text{size}(\beta)$;
- $\text{size}(\alpha \Delta \beta) = \min(\text{size}(\alpha), \text{size}(\beta)) + 1$.

Obviously, if $x \; R_{\mathcal{M}}(\alpha) \; y$ and $\text{size}(\alpha) = 0$ then $x = y$. Now we decompose expressions into subexpressions, associating a *depth* to each subformula.

Definition 24 (Localized expression and decomposition). *A localized expression is a tuple* $d\colon \nu$ *where* ν *is an expression and* $d \in \mathbb{N}$ *is called the depth. Given any localized expression* $d\colon \nu$*, the decomposition* $\text{Cl}(d\colon \nu)$ *of* $d\colon \nu$ *is the least set of localized expressions containing* $d\colon \nu$ *and closed by the application of the rules from Fig. 1. We write* $\text{Cl}(\phi)$ *for* $\text{Cl}(0\colon \phi)$*.*

$$\frac{d\colon \phi}{d\colon \sim\phi} \qquad \frac{d\colon \langle\alpha\rangle\phi}{d\colon \alpha \quad d+\text{size}(\alpha)\colon \phi} \qquad \frac{d\colon \phi \vee \psi}{d\colon \phi \quad d\colon \psi}$$

$$\frac{d\colon \phi?}{d\colon \phi} \qquad \frac{d\colon \alpha;\beta}{d\colon \alpha \quad d+\text{size}(\alpha)\colon \beta} \qquad \frac{d\colon \alpha\Delta\beta}{d\colon \alpha \quad d\colon \beta}$$

$$\frac{d\colon \phi \circ \psi}{d+1\colon \phi \quad d+1\colon \psi} \qquad \frac{d\colon \phi \rhd \psi}{d+1\colon \phi \quad d+1\colon \psi} \qquad \frac{d\colon \phi \lhd \psi}{d+1\colon \phi \quad d+1\colon \psi}$$

Fig. 1. Rules for the decomposition of localized programs and formulas

Lemma 12. *The cardinality of* $\text{Cl}(\phi)$ *is linear in* $|\phi|$*.*

Lemma 13. $\max\{d \mid \exists\nu, d\colon \phi \in \text{Cl}(\phi)\}$ *is linear in* $|\phi|$*.*

We now prove a strong finite model property for PDL_0^{Δ} interpreted over the class of all separated frames. The procedure SELECTION on the following page creates a model $\mathcal{M}_s$ from a model $\mathcal{M}_o$ satisfying a formula ϕ_0 at w_0. It uses the recursive procedure LINK described in Procedure 2.

Input: A formula ϕ_0, a model $\mathcal{M}_o = (W_o, R_o, \star_o, V_o)$ and an initial state $w_0 \in W_o$ such that $w_0 \in V_{\mathcal{M}_o}(\phi_0)$.
Result: A finite model $\mathcal{M}_s = (W_s, R_s, \star_s, V_s)$.
Data: A subset $K \subseteq W_s$ of marked nodes and an integer $n \in \mathbb{N}$.

initialisation
  $n = 0$;
  $W_s = \{(0, 0, w_0)\}$;
  $R_s(a) = \emptyset$ for all $a \in \Pi_0$;
  $(O, 0, w_0) \star_s (O, 0, w_0) = \emptyset$;
  $K = \emptyset$;
end
while $K \neq W_s$ **do**
  choose an unmarked state $(k, d, w) \in W_s \setminus K$;
  while $(k, d, w) \notin K$ **do**
    let $V_s(p) = \{(k_x, d_x, x) \in W_s \mid x \in V_o(p)\}$ for all $p \in \Phi_0$;
    if *there exists* d': $\langle\alpha\rangle\phi \in \mathrm{Cl}(\phi_0)$ *such that* $\mathrm{size}(\alpha) > 0$, $d' \geq d$, $w \in V_{\mathcal{M}_o}(\langle\alpha\rangle\phi)$ *and* $(k, d, w) \notin V_{\mathcal{M}_s}(\langle\alpha\rangle\phi)$ **then**
      choose y **s.t.** $w\ R_{\mathcal{M}_o}(\alpha)\ y$ and $y \in V_{\mathcal{M}_o}(\phi)$;
      let $d_y = d + \mathrm{size}(\alpha)$;
      let $n = n + 1$;
      add (n, d_y, y) **to** W_s ;
      call LINK $(\mathcal{M}_o, \mathcal{M}_s, n, (k, d, w), (n, d_y, y), \alpha)$;
    else if *there exists* d': $\phi \circ \psi \in \mathrm{Cl}(\phi_0)$ *such that* $d' \geq d$, $w \in V_{\mathcal{M}_o}(\phi \circ \psi)$ *and there is no* $(k_x, d_x, x), (k_y, d_y, y) \in W_s$ *such that* $(k, d, w) \in (k_x, d_x, x) \star_s (k_y, d_y, y)$ **then**
      choose x *and* y **s.t.** $w \in x \star_s y$, $x \in V_{\mathcal{M}_o}(\phi)$ and $y \in V_{\mathcal{M}_o}(\psi)$;
      add $(n + 1, d + 1, x)$ and $(n + 2, d + 1, y)$ **to** W_s ;
      add (k, d, w) **to** $(n + 1, d + 1, x) \star_s (n + 2, d + 1, y)$;
      let $n = n + 2$;
    else if *there exists* d': $\phi \triangleright \psi \in \mathrm{Cl}(\phi_0)$ *such that* $d' \geq d$, $w \in V_{\mathcal{M}_o}(\phi \triangleright \psi)$ *and* $(k, d, w) \notin V_{\mathcal{M}_s}(\phi \triangleright \psi)$ **then**
      choose x *and* y **s.t.** $y \in x \star_s w$, $x \in V_{\mathcal{M}_o}(\phi)$ and $y \in V_{\mathcal{M}_o}(\psi)$;
      add $(n + 1, d + 1, x)$ and $(n + 2, d + 1, y)$ **to** W_s ;
      add $(n + 2, d + 1, y)$ **to** $(n + 1, d + 1, x) \star_s (k, d, w)$;
      let $n = n + 2$;
    else if *there exists* d': $\phi \triangleleft \psi \in \mathrm{Cl}(\phi_0)$ *such that* $d' \geq d$, $w \in V_{\mathcal{M}_o}(\phi \triangleleft \psi)$ *and* $\mathcal{M}_s, (d, w) \triangleleft \notin V_\phi(\psi)$ **then**
      choose x *and* y **s.t.** $x \in w \star_s y$, $x \in V_{\mathcal{M}_o}(\phi)$ and $y \in V_{\mathcal{M}_o}(\psi)$;
      add $(n + 1, d + 1, x)$ and $(n + 2, d + 1, y)$ **to** W_s ;
      add $(n + 1, d + 1, x)$ **to** $(k, d, w) \star_s (n + 2, d + 1, y)$;
      let $n = n + 2$;
    else
      add (k, d, w) **to** K ;
    end
  end
end

Procedure 1. SELECTION

Input: Two models $\mathcal{M}_o = (W_o, R_o, \star_o, V_o)$ and $\mathcal{M}_s = (W_s, R_s, \star_s, V_s)$, an integer n, two states $(k_x, d_x, x), (k_y, d_y, y) \in W_s$ and a program α such that $x\ R_{\mathcal{M}_o}(\alpha)\ y$.

Result: $\mathcal{M}_s$ and n modified.

```
if α is of the form a ∈ Π_0 then
    add ((k_x, d_x, x), (k_y, d_y, y)) to R_s(a) ;
else if α is of the form (β;γ) then
    if size(β) = 0 then
        call Link (M_o, M_s, n, (k_x, d_x, x), (k_y, d_y, y), γ) ;
    else if size(γ) = 0 then
        call Link (M_o, M_s, n, (k_x, d_x, x), (k_y, d_y, y), β) ;
    else
        choose z s.t. x R_{M_o}(β) z and z R_{M_o}(γ) y;
        let n = n + 1 ;
        let d_z = d_x + size(α) ;
        add (n, d_z, z) to W_s ;
        call Link (M_o, M_s, n, (k_x, d_x, x), (n, d_z, z), β) ;
        call Link (M_o, M_s, n, (n, d_z, z), (k_y, d_y, y), γ) ;
    end
else if α is of the form (βΔγ) then
    if size(β) = 0 and size(γ) = 0 then
        add (k_y, d_y, y) to (k_x, d_x, x) ⋆_s (k_x, d_x, x) ;
    else if size(β) = 0 then
        choose z s.t. x R_{M_o}(γ) z and y ∈ x ⋆_o z;
        let n = n + 1 ;
        let d_z = min (d_y + 1, d_x + size(γ)) ;
        add (n, d_z, z) to W_s ;
        add (k_y, d_y, y) to (k_x, d_x, x) ⋆_s (n, d_z, z) ;
        call Link (M_o, M_s, n, (k_x, d_x, x), (n, d_z, z), γ) ;
    else if size(γ) = 0 then
        choose w s.t. x R_{M_o}(β) w and y ∈ w ⋆_o x;
        let n = n + 1 ;
        let d_w = min (d_y + 1, d_x + size(β)) ;
        add (n, d_w, w) to W_s ;
        add (k_y, d_y, y) to (n, d_w, w) ⋆_s (k_x, d_x, x) ;
        ;
        call Link (M_o, M_s, n, (k_x, d_x, x), (n, d_w, w), β) ;
    else
        choose w and z s.t. x R_{M_o}(β) w, x R_{M_o}(γ) z and y ∈ w ⋆_o z;
        let n = n + 2 ;
        let d_w = min (d_y + 1, d_x + size(β), d_x + size(γ) + 1) ;
        let d_z = min (d_y + 1, d_x + size(γ), d_x + size(β) + 1) ;
        add (n − 1, d_w, w) and (n, d_z, z) to W_s ;
        add (k_y, d_y, y) to (n − 1, d_w, w) ⋆_s (n, d_z, z) ;
        call Link (M_o, M_s, n, (k_x, d_x, x), (n − 1, d_w, w), β) ;
        call Link (M_o, M_s, n, (k_x, d_x, x), (n, d_z, z), γ) ;
    end
end
```

Procedure 2. Link

Lemma 14. *The procedure* SELECTION *terminates and the cardinality of* W_s *is exponential in* $|\phi_0|$.

Lemma 15. *Whenever* LINK *is called,* $d_y \leq d_x + \text{size}(\alpha)$.

Lemma 16. *For all* $(k_y, d_y, y), (k_w, d_w, w), (k_z, d_z, z) \in W_s$, *such that* $(k_y, d_y, y) \in (k_w, d_w, w) \star_s (k_z, d_z, z)$ *then* $y \in w \star_o z$, $|d_y - d_w| \leq 1$, $|d_y - d_z| \leq 1$ *and* $|d_w - d_z| \leq 1$.

Lemma 17. *For all* $(k_x, d_x, x), (k_y, d_y, y) \in W_s$ *and all* α, *if* $(k_x, d_x, x)\ R_{\mathcal{M}_s}(\alpha)$ (k_y, d_y, y), *then* $d_y \leq d_x + \text{size}(\alpha)$.

Lemma 18. *If* $\mathcal{M}_o$ *is separated, then* $\mathcal{M}_s$ *is separated too.*

Lemma 19 (Truth lemma). *If* $\mathcal{M}_o$ *is separated, then* $(0, 0, w_0) \in V_{\mathcal{M}_s}(\phi_0)$.

Proposition 15. *Any* PDL_0^{Δ} *formula* ϕ *satisfiable in a separated model is satisfiable in a separated finite model with a number of states bounded by an exponential in* $|\phi|$.

Since the model-checking problem for PDL_0^{Δ} is obviously polynomial in the size of the model, therefore we have the following corollary:

Corollary 1. *The satisfiability problem for* PDL_0^{Δ} *in the class of separated frames is decidable in non-deterministic exponential time.*

8 Conclusion

In modal logic, standard proofs of completeness for a given logic are usually based on the canonical frame construction consisting of the set of all maximal consistent sets of the logic equipped with standard definitions for the canonical accessibility relations. Since the program operation of fork considered in [11, Chap. 1] is not modally definable in the ordinary language of PDL, this method cannot work in our case. As a result, we have given an axiomatization of PDL_0^{Δ}, our variant of iteration-free PDL with fork, using an unorthodox rule of proof and we have proved its completeness using large programs. So, we have extended the canonical frame construction introducing new tools and techniques connected with an unorthodox rule of proof and large programs.

We anticipate a number of further investigations. First, there is the following general question: is it possible to eliminate the rule of proof (FOR) and to replace it with a finite set of additional axiom schemes? Second, more details on decidability/complexity issues would be relevant. Third, there is the question of the complete axiomatization of validity with respect to other classes of frames like the class of frames considered in [11, Chap. 1], i.e. the class of all separated, deterministic and serial frames. Fourth, is the validity problem with respect to the class of all separated, deterministic and serial frames decidable? If it is, what is its complexity? Fifth, it remains to see whether our approach can be extended to the full language of PDL with fork, this time with iteration.

A novelty in the paper is the proof that fork is modally definable in a language with propositional quantifiers and that the rule (FOR) in a sense simulates the quantifier rule for universal quantification in the context of the definition of fork. This is a new look on the nature of some context dependent rules of proof like (FOR). In some ways, (FOR) is similar to the rule for intersection from [3,4]. See also [1] for ideas about its elimination from the axiomatization of PDL_0^{Δ} we have given. We expect that our variant of the canonical frame construction can be applied to other logics, for instance $PRSPDL$, the variant of PDL with fork given rise by the binary operation of fork ∇ considered in Benevides *et al.* [5, Sect. 2] and whose axiomatization is still open.

Acknowledgement. Our research has been supported by the ANR project no. ANR-11-BS02-011.

References

1. Balbiani, P.: Eliminating unorthodox derivation rules in an axiom system for iteration-free PDL with intersection. Fundam. Inf. **56**, 211–242 (2003)
2. Balbiani, P., Boudou, J.: About the axiomatization and the computability of a variant of iteration-free PDL with fork. Institut de recherche en information de Toulouse (2017)
3. Balbiani, P., Vakarelov, D.: Iteration-free PDL with intersection: a complete axiomatization. Fundam. Inf. **45**, 173–194 (2001)
4. Balbiani, P., Vakarelov, D.: PDL with intersection of programs: a complete axiomatization. J. Appl. Non-Classical Logics **13**, 231–276 (2003)
5. Benevides, M., de Freitas, R., Viana, P.: Propositional dynamic logic with storing, recovering and parallel composition. Electron. Notes Theor. Comput. Sci. **269**, 95–107 (2011)
6. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, Cambridge (2001)
7. Danecki, R.: Nondeterministic propositional dynamic logic with intersection is decidable. In: Skowron, A. (ed.) SCT 1984. LNCS, vol. 208, pp. 34–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-16066-3_5
8. Došen, K.: A brief survey of frames for the Lambek calculus. Zeitschrift für mathematische Logik und Grundlagen der Mathematik **38**, 179–187 (1992)
9. del Cerro, L.F., Orłowska, E.: DAL—a logic for data analysis. Theoret. Comput. Sci. **36**, 251–264 (1985)
10. Fisher, M., Ladner, R.: Propositional dynamic logic of regular programs. J. Comput. Syst. Sci. **18**, 194–211 (1979)
11. Frias, M.: Fork Algebras in Algebra, Logic and Computer Science. World Scientific, River Edge (2002)
12. Frias, M., Baum, G., Haeberer, A.: Fork algebras in algebra, logic and computer science. Fund. Inf. **32**, 1–25 (1997)
13. Frias, M., Veloso, P., Baum, G.: Fork algebras: past, present and future. J. Relat. Methods Comput. Sci. **1**, 181–216 (2004)
14. Gargov, G., Passy, S.: A note on Boolean modal logic. In: Petkov, P. (ed.) Mathematical Logic, pp. 299–309. Plenum Press, New York (1990)
15. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press, Cambridge (2000)

16. Kozen, D., Parikh, R.: An elementary proof of the completeness of PDL. Theoret. Comput. Sci. **14**, 113–118 (1981)
17. Mikulás, S.: Complete calculus for conjugated arrow logic. In: Marx, M., Pólos, L., Masuch, M. (eds.) Arrow Logic and Multi-Modal Logic, pp. 125–139. CSLI, Stanford (1996)
18. Mirkowska, G.: PAL—propositional algorithmic logic. Fund. Inf. **4**, 675–760 (1981)
19. Tarski, A.: On the calculus of relations. J. Symb. Logic **6**, 73–89 (1941)

A Dynamic Logic for Learning Theory

Alexandru Baltag[1], Nina Gierasimczuk[2], Aybüke Özgün[1,3], Ana Lucia Vargas Sandoval[1(✉)], and Sonja Smets[1]

[1] ILLC, University of Amsterdam, Amsterdam, The Netherlands
ana.varsa@gmail.com
[2] DTU Compute, Technical University of Denmark, Copenhagen, Denmark
[3] LORIA, CNRS-Université de Lorraine, Nancy, France

Abstract. Building on previous work [4,5] that bridged Formal Learning Theory and Dynamic Epistemic Logic in a topological setting, we introduce a *Dynamic Logic for Learning Theory* (DLLT), extending Subset Space Logics [9,17] with dynamic *observation modalities* $[o]\varphi$, as well as with a *learning operator* $L(\vec{\sigma})$, which encodes the learner's *conjecture* after observing a finite sequence of data $\vec{\sigma}$. We completely axiomatise DLLT, study its expressivity and use it to characterise various notions of knowledge, belief, and learning.

Keywords: Learning theory · Dynamic epistemic logic Modal Logic · Subset Space Semantics · Inductive knowledge Epistemology

1 Introduction

The process of learning consists of incorporating new information into one's prior information state. Dynamic epistemic logic (DEL) studies such one-step information changes from a logical perspective [6,19,23]. But the general concept of *learning* encompasses not only one-step revisions, but also their *long-term horizon*. In the long run, learning should lead to knowledge — an epistemic state of a particular value. Examples include language learning (inferring the underlying grammar from examples of correct sentences), and scientific inquiry (inferring a theory of a phenomenon on the basis of observations). Our goal in this paper is to provide a simple logic for reasoning about this process of *inductive learning* from successful observations. Understanding inductive inference is of course an infamously difficult open problem, and there are many different approaches in the literature.[1] However, in this paper we *do not try to solve* the problem of induction, but only to *reason about* a (rational) inductive learner. For this, we adopt the more flexible and open-ended approach of Formal Learning

[1] From probabilistic and statistical formalisms based on Bayesian reasoning, Popper-style measures of corroboration, through default and non-monotonic logics, Carnap-style 'inductive logic', to AGM-style rational belief revision and theory change.

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 35–54, 2018.
https://doi.org/10.1007/978-3-319-73579-5_3

Theory (FLT). While most other approaches adopt a normative stance, aimed at prescribing 'the' correct algorithm for forming and changing rational beliefs from observations (e.g., Bayesian conditioning), or at least at prescribing some general rational constraints that any such algorithm should obey (e.g., the AGM postulates for belief revision), FLT gives the learner a high degree of freedom, allowing the choice of *any learner* that produces conjectures *based on the data* (no matter how 'crazy' or unjustified are these conjectures, or how erratic is the process of belief change). In FLT the only criterion of success is... success: tracking the truth in the limit. In other words, the only thing that matters is whether or not the iterated belief revision process will eventually stabilise on a conjecture which matches the truth (about some given issue). Of course, we are not interested in cases of convergence to the truth 'by accident', but in determining whether or not a given learner is *guaranteed* to eventually track the truth; hence, the focus on 'The Logic of *Reliable* Inquiry'.[2]

We propose a formalism that combines ideas from: Subset Space Logics, as introduced by Moss and Parikh [17], investigated further by Dabrowski et al. [9] and already merged with the DEL tradition in prior work [3,7,8,20,22,25]; the topological approach to FLT in [5,16]; and the general agenda of bridging DEL and FLT in [13]. Semantically, we take *intersection spaces* (a type of subset spaces that are closed under finite non-empty intersections), with points interpreted as possible worlds and neighbourhoods interpreted as *observations* (or *information states*) (see, e.g., [18] for a survey on subset space logics). We enhance these structures with a *learner* $\mathbb{L}$, mapping every information state to a *conjecture*, representing the learner's strongest belief in this state. As in Subset Space logics, our language features an S5-type *'knowledge-with-certainty' modality*, capturing the learner's *hard information*, as well as the so-called *'effort' modality*, which we interpret as 'stable truth' (i.e., truth immune to further observations). We add to this *observation modalities* $[o]\varphi$, analogous to the dynamic modalities in Public Announcement Logic (PAL), as well as a *learning operator* $L(\vec{\sigma})$, which encodes the learner's conjecture after observing a finite sequence of pieces of evidence $\vec{\sigma}$. This can be used to give a natural definition of *belief*: a learner believes P iff she knows that P is entailed by her current conjecture.

We present a sound and complete axiomatisation of DLLT with respect to our learning models. The completeness uses a neighbourhood version of the standard canonical model construction. We use this logic to characterise various learnability notions. In particular, we are able to model inductive learning as *coming to stably believe a true fact after observing an incoming sequence of true data*. The possibility of such learning corresponds to a key concept in FLT, namely *identifiability in the limit* first introduced and studied by Gold in [15]. Finally, we discuss the expressivity of DLLT, showing that the dynamic observation modalities are in principle eliminable via reduction laws.

[2] 'The Logic of Reliable Inquiry' is the title of a classic text in FLT-based epistemology [16].

Due to page-limit constraints, we include only the shortest proofs in the main text of this paper. The other relevant proofs can be found in the Appendix of the long version of this paper, available online at https://sites.google.com/site/ozgunaybuke/publications.

1.1 Effort Modality and Knowledge

In [24], Vickers reconstructed general topology as a logic of observation, in which the points of the space represent possible states of the world, while basic open neighborhoods of a point are interpreted as *information states* produced by accumulating finitely many observations. Moss and Parikh [17] gave an account of learning in terms of *observational* effort. Making the epistemic effort to obtain more information about a possible world has a natural topological interpretation—it can be seen as *shrinking* the open neighborhood (representing the current information state), thus providing a more accurate approximation of the actual state of the world [9,11,12,17,18]. A similar line was proposed in Formal Epistemology [5,16], where it was combined with more sophisticated notions of learning borrowed from FLT. The following example relates the effort modality with knowledge.

Example 1 ([18]*).* Let us consider some *measurement*, say of a vehicle's velocity. Suppose a policeman uses radar to determine whether a car is speeding in a 50-mile speed-limit zone. The property *speeding* can be identified with the interval $(50, \infty)$. Suppose the radar shows 51 mph, but the radar's accuracy is ± 2 mph. The intuitive meaning of a speed measurement of 51 ± 2 is that the car's true speed v is in the *open interval* $(49, 53)$. According to [18], "anything which we *know* about v must hold not only of v itself, but also of any v' in the same interval" [18, p. 300]. Since the interval $(49, 53)$ is not fully included in the 'speeding' interval $(50, \infty)$, the policeman does *not know* that the car is speeding. But suppose that he does another measurement, using a more accurate radar with an accuracy of ± 1 mph, which shows 51.5 mph. Then he will *come to know* that the car is speeding: the open interval $(50.5, 52.5)$ *is* included in $(50, \infty)$ (Fig. 1).

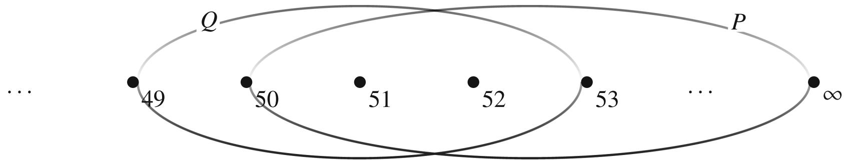


Fig. 1. Example 1; P := "the car is speeding", Q := "the reading of the radar is 51 km/h"

Infallible Knowledge Versus Inductive Knowledge. Let us now extend this picture with learning as understood in FLT. We start by setting the stage—briefly introducing *learning frame*, the underlying structure of learning.[3] Using them we will be able to explain and model various epistemic notions.

First, consider a pair $(X, \mathscr{O})$, where X is a non-empty set of *possible worlds*; $\mathscr{O} \subseteq \mathcal{P}(X)$ is a non-empty set of *information states* (or 'observables', or 'evidence'). We take $\mathscr{O}$ to be closed on intersections, i.e., for any $O_1, O_2 \in \mathscr{O}$, we have $O_1 \cap O_2 \in \mathscr{O}$, the resulting $(X, \mathscr{O})$ is called an intersection space. A *learning frame* is a triplet $(X, \mathscr{O}, \mathbb{L})$, where $\mathbb{L} : \mathscr{O} \to \mathcal{P}(X)$ is a *learner*, i.e., a map associating to every finite sequence of observations $O \in \mathscr{O}$ some 'conjecture' $\mathbb{L}(O) \subseteq X$.

Let us now reconstruct Example 1 as a learning frame. We take $X = (0, \infty)$ as the set of possible worlds (representing possible velocities of the car, where we assume the car is known to be *moving*); $\mathscr{O} = \{(a, b) \in Q \times Q : 0 < a < b < \infty\}$ is the set of all open intervals with positive rational endpoints (representing possible measurement results by arbitrarily accurate radars). The pair $(X, \mathscr{O})$ is an intersection frame, and the topology generated by $\mathscr{O}$ is the standard topology on real numbers (restricted to X).

Certain (Infallible) Knowledge. In an information state $U \in \mathscr{O}$, the learner is said to *infallibly know a proposition $P \subseteq X$ conditional on observation O* if her conditional information state entails P, i.e., if $U \cap O \subseteq P$. The learner *(unconditionally) knows P* if $U \subseteq P$. The possibility of achieving certain knowledge about a proposition $P \subseteq X$ in a possible world $x \in X$ by a learner $\mathbb{L}$ if given enough evidence (true at x) is called *learnability with certainty*. In other worlds P is learnable with certainty if there exists some observable property $O \in \mathscr{O}$ (with $x \in O$) such that the learner infallibly knows P in information state O. Learnability can be used to define verifiability and falsifiability: a proposition $P \subseteq X$ is *verifiable (resp. falsifiable) with certainty (by $\mathbb{L}$)* if it is learnable with certainty by $\mathbb{L}$ whenever it is true (resp. false); i.e. if P is learnable with certainty at all worlds $x \in P$ (resp. $x \notin P$). Finally, a proposition $P \subseteq X$ is *decidable with certainty (by $\mathbb{L}$)* if it is both verifiable and falsifiable with certainty (by $\mathbb{L}$).

In the context of Example 1, let us consider the certain knowledge of the policeman. In the information state $U = (49, 53)$, the learner/policeman does not know the proposition $P = (50, \infty)$, so he cannot be certain that the car is speeding. However, the speeding property P is verifiable with certainty: whenever P is actually true, he could perform a more accurate speed measurement, by which he can get to an information state in which P is infallibly known. In our example, the policeman refined his measurement getting to the information state $O = (50.5, 52.5)$, thus coming to know P. In contrast, the property $X - P = (0, 50]$ ('not speeding') is not verifiable with certainty: if by some kind of miraculous coincidence, the speed of the car is exactly 50 mph, then the car is not speeding, but the policeman will never know that for certain (since every speed measurement, of any degree of accuracy, will be consistent both with P

[3] We will return to it, with complete definitions, later in the paper. Our DLLT is interpreted over such frames.

and with $X - P$). Nevertheless, $X - P$ is always falsifiable with certainty: if false (i.e. if the speed is in P, so that car is speeding), then as we saw the policeman will come to infallibly know that (by some more accurate measurement).

Inductive (Defeasible) Knowledge

Before we proceed to Inductive Knowledge let us consider epistemic states weaker than certainty, *belief*. In an information state $U \in \mathscr{O}$, the learner $\mathbb{L}$ is said to:

- *un-conditionally believe* $P \subseteq X$ if $\mathbb{L}(U) \subseteq P$.[4]
- *believe a proposition* $P \subseteq X$ *conditional on observation* O if $\mathbb{L}(U \cap O) \subseteq P$;
- *have undefeated belief in a proposition* $P \subseteq X$ *at world* x if she believes P in every information state $O \in \mathscr{O}$ that is true at x (i.e., $x \in O$) and is at least as strong as U (i.e., $O \subseteq U$). This means that, once she reaches information state U, no further evidence can defeat the learner's belief in P.

One of the central problems in epistemology is to define a realistic notion of knowledge that fits the needs of empirical sciences. It should allow fallibility, while requiring a higher standards of evidence and robustness than simple belief. One of the main contenders is the so-called Defeasibility Theory of Knowledge, which defines *defeasible (fallible) knowledge* as true undefeated belief. In the learning-theoretic context, this gives us an evidence-based notion of 'inductive knowledge': in an information state U, P is *inductively known* at world x if it is true at x (i.e., $x \in P$) and it is undefeated belief (in the sense defined above). This is the kind of knowledge that can be gained by empirical (incomplete) induction, based on experimental evidence.

As in the case of learnability with certainty, achieving inductive knowledge is defined as learnability. A proposition $P \subseteq X$ is *inductively learnable* (or 'learnable in the limit') *by the learner* $\mathbb{L}$ at world x if $\mathbb{L}$ will come to inductively know P if given enough evidence (true at x); i.e. if there exists some observable property $O \in \mathscr{O}$ of world x (i.e., with $x \in O$) such that $\mathbb{L}$ inductively knows P in information state O. Inductive verifiability and falsifiability are defined in terms of learnability. A proposition $P \subseteq X$ is *inductively verifiable (resp. falsifiable) by the learner* $\mathbb{L}$, if it is inductively learnable whenever it is true (resp. false); i.e., if P is inductively learnable at all worlds $x \in P$ (resp. $x \notin P$). A proposition $P \subseteq X$ is *inductively decidable by* $\mathbb{L}$ if it is both inductively verifiable and inductively falsifiable by $\mathbb{L}$.

In the context of Example 1, let us now turn to inductive knowledge of the policeman. Both speeding (P) and non-speeding ($X - P$) are inductively decidable (and thus both inductively verifiable and inductively falsifiable): for instance, they are inductively decidable by the learner $\mathbb{L}$, defined by putting $\mathbb{L}(O) := O \cap P$ for every open interval $O = (a, b) \in \mathscr{O}$ s.t. $O \cap P \neq \emptyset$, and putting $\mathbb{L}(O) := O (\subseteq X - P)$ otherwise.

[4] In the tautological information state X, the learner believes P iff $\mathbb{L}(X) \subseteq P$.

Intuitively, this learner is the 'suspicious cop', who believes the car to be *speeding* whenever the available evidence cannot settle the issue, and keeps this conjecture until it is disproven by some more accurate measurement. Regardless of the car's speed, this policeman will be right 'in the limit': after doing enough accurate measurements, he will *eventually settle on the correct belief* (about speeding or not); though of course (in case the car's speed is exactly 50 mph) he may still never be certain. Obviously, the dual learner (the 'judge', who assumes innocence until proven guilty) will also inductively decide the speeding issue. An example of property which is *inductively decidable but neither verifiable with certainty nor falsifiable with certainty* is the proposition $S = [50, 51)$. It is not verifiable with certainty, since if the car's speed is exactly 50 mph, then Q is true but the learner will never be certain of this; and it is not falsifiable with certainty, since if the car's speed is exactly 51 mph, then S is false but the learner will never be certain of that. Nevertheless, S is *inductively decidable*, e.g. by the learner defined by: $\mathbb{L}(a, b) := (a, b) \cap S$ for open intervals with $a < 50 < b$; $\mathbb{L}(a, b) := (a, b)$ for open intervals (a, b) s.t. either $(a, b) \subseteq S$ or $(a, b) \cap S = \emptyset$; and $\mathbb{L}(a, b) := [51, b)$ whenever $50 < a < 51 < b$.

Dependence on the Learner. It is easy to see that learnability (verifiability, falsifiability, decidability) with certainty are *learner-independent* notions (since they are directed towards achieving infallible knowledge), so they do not depend on $\mathbb{L}$ but only on the underlying intersection model. In contrast, *the corresponding inductive notions above are learner-dependent.* As a consequence, the interesting concepts in Learning Theory are obtained from them by *quantifying existentially over learners*: a proposition P is *inductively learnable (verifiable, falsifiable, decidable)* if there exists *some* learner $\mathbb{L}$ s.t. P is respectively inductively learnable verifiable falsifiable decidable by $\mathbb{L}$. This property of a learning frame is called *generic inductive learnability.*

Topological Characterizations. As it is well-known in learning theory and formal epistemology [16,24], the above notions are *topological* in nature: P is learnable with certainty at world x iff x is in the *interior* of P with respect to the topology generated by $\mathscr{O}$; P is verifiable with certainty iff it is *open* in the same topology; P is falsifiable with certainty iff it is *closed* in this topology; finally, P is decidable with certainty iff it is *clopen.* The corresponding inductive notions can be easily characterized [16] in the case that the topology generated by $\mathscr{O}$ satisfies the separation condition[5] $T1$: in this case, P is inductively verifiable iff it is Σ_2 in the Borel hierarchy for this topology (i.e. a countable union of closed sets); in the same conditions, P is inductively falsifiable iff it is Π_2 (a countable intersection of open sets), and it is inductively decidable iff it is Δ_2 (i.e. Σ_2 and Π_2). More recently, in work by three of this paper's coauthors [5], these characterisations were generalised to arbitrary topologies satisfying the

[5] This topology is $T1$ iff for every two distinct points $x \neq y$ there exist an observation $O \in \mathscr{O}$ with $x \in O$ and $y \notin O$.

weaker separation condition[6] $T0$; in particular, P is inductively verifiable iff it is a countable union of locally closed sets.[7]

2 Dynamic Logic for Learning Theory

In this section we introduce our 'dynamic logic for learning theory' DLLT. As already mentioned, this is obtained by adding two ingredients to the language of Subset Space Logics: *dynamic observation modalities* $[o]\varphi$ and a *learning operator* $L(\vec{o})$.

2.1 Syntax and Semantics of DLLT

Let $\mathrm{Prop} = \{p, q, \ldots\}$ be a countable set of *propositional variables*, denoting arbitrary 'ontic' (i.e., non-epistemic) facts that might hold in a world (even if they might never be observed), and let $\mathrm{Prop}_{\mathscr{O}} = \{o, u, v, \ldots\}$ be a countable set of *observational variables*, denoting 'observable facts' (which, if true, will eventually be observed).

Definition 1. *The syntax of our language $\mathfrak{L}$ is defined by the grammar:*

$$\varphi ::= p \mid o \mid L(\vec{o}) \mid \neg\varphi \mid \varphi \wedge \varphi \mid K\varphi \mid \Box\varphi \mid [o]\varphi$$

where $p \in \mathrm{Prop}$ and $o \in \mathrm{Prop}_{\mathscr{O}}$, while $\vec{o} = (o_1, \ldots, o_n) \in \mathrm{Prop}^*_{\mathscr{O}}$ is a finite sequence of observational variables. (In particular, empty sequence is denoted by λ.) We employ the usual abbreviations for propositional connectives $\top, \bot, \vee, \rightarrow, \leftrightarrow$ and for the dual modalities $\langle K\rangle, \Diamond, \langle o\rangle$.

The informal meaning of our formulas is as follows. Propositional variables denote *ontic facts* (i.e. factual, non-epistemic features of a world), while observational variables o denote *observable facts* (i.e. facts that, if true, will eventually be observed). We read $K\varphi$ as 'the learner *knows* φ (with absolute certainty)'. $\Box\varphi$ is the so-called 'effort modality' from Subset Space Logic; we read $\Box\varphi$ as 'φ is *stably true*'. Indeed, $\Box\varphi$ holds iff φ is true and will stay true *no matter what new (true) evidence is observed.* The operator $[o]\varphi$ is similar to the operator $[\psi]\varphi$ in Public Announcement Logic, but it is restricted to the cases when ψ is a particular kind of atomic formula, namely an observational variable $o \in \mathrm{Prop}_{\mathscr{O}}$. So we read $[o]\varphi$ as 'after o is observed, φ will hold'. Finally, $L(\vec{o})$ denotes the learner's *conjecture* given observations $\vec{o}$; i.e. her strongest belief (i.e., the set of worlds considered to be most plausible) after observing $\vec{o}$.

[6] The observational topology is $T0$ iff points can be distinguished by observations; i.e. if x and y satisfy the same observable properties in $\mathscr{O}$, then $x = y$. Obviously, $T0$ is a minimally necessary condition for any kind of learnability of the real world from observations.

[7] A set is locally closed if it is the intersection of a closed and an open set.

Definition 2 (Intersection Frame/Model and Learning Frame/Model). An *intersection frame* [9,17] is a pair $(X, \mathscr{O})$, where: X is a non-empty set of *possible worlds* (or 'ontic states'); $\mathscr{O} \subseteq \mathcal{P}(X)$ is a non-empty set of subsets, called *information states* (or 'observables', or 'evidence'), which is assumed to be *closed under finite intersections*: if $\mathcal{F} \subseteq \mathcal{O}$ is finite then $(\bigcap \mathcal{F}) \in \mathcal{O}$. An *intersection model* $(X, \mathscr{O}, \|\cdot\|)$ is an intersection frame $(X, \mathscr{O})$ together with a valuation map $\|\cdot\| : \mathrm{Prop} \cup \mathrm{Prop}_{\mathscr{O}} \to \mathcal{P}(X)$, that maps propositional variables p into arbitrary sets $\|p\| \subseteq X$ and observational variables o into observable properties $\|o\| \in \mathscr{O}$.

A *learning frame* is a triplet $(X, \mathscr{O}, \mathbb{L})$, where $(X, \mathscr{O})$ is an intersection frame and $\mathbb{L} : \mathscr{O} \to \mathcal{P}(X)$ is a *learner*, i.e. a map associating to every information state $O \in \mathscr{O}$ some 'conjecture' $\mathbb{L}(O) \subseteq X$, and satisfying two properties: (1) $\mathbb{L}(O) \subseteq O$ (*conjectures fit the evidence*), and (2) if $O \neq \emptyset$ then $\mathbb{L}(O) \neq \emptyset$ (*consistency of conjectures* based on consistent evidence). We can extend $\mathbb{L}$ to range over *strings* of information states $\vec{O} = (O_1, \ldots, O_n) \in \mathscr{O}^*$ in a natural way, by putting $\mathbb{L}(\vec{O}) := \mathbb{L}(\bigcap \vec{O})$, where $\bigcap \vec{O} := O_1 \cap \ldots \cap O_n$. A *learning model* $\mathcal{M} = (X, \mathscr{O}, \mathbb{L}, \|\cdot\|)$ is a learning frame $(X, \mathscr{O}, \mathbb{L})$ together with a valuation map $\|\cdot\| : \mathrm{Prop} \cup \mathrm{Prop}_{\mathscr{O}} \to \mathcal{P}(X)$ as above; equivalently, it consists of an intersection model $(X, \mathscr{O}, \|\cdot\|)$ together with a learner, as defined above.

Intuitively, the states in X represent possible worlds. The tautological evidence $X = \bigcap \emptyset$ represents the state of 'no information' (before anything is observed), while the contradictory evidence $\emptyset$ represents inconsistent information. Finally, $\mathbb{L}(O)$ represents the learner's conjecture after observing O, while $\mathbb{L}(O_1, \ldots, O_n) = \mathbb{L}(O_1 \cap \ldots \cap O_n)$ represents the conjecture after observing a finite sequence of observations $O_1, \ldots, O_n$. (The fact that $\mathscr{O}$ is closed under finite intersections is important here for identifying any finite sequence $O_1, \ldots, O_n$ with a single observation $O = O_1 \cap \ldots O_n \in \mathscr{O}$.)

Epistemic Scenarios. As in Subset Space Semantics, the formulas of our logic are not interpreted at possible worlds, but at so-called *epistemic scenarios*, i.e. pairs (x, U) of an ontic state $x \in X$ and an information state $U \in \mathscr{O}$ such that $x \in U$. Therefore, only the truthful observations about the actual state play a role in the evaluation of formulas. We denote by $ES(\mathcal{M}) := \{(x, U) : x \in U \in \mathscr{O}\}$ the set of all epistemic scenarios.

Definition 3 (Semantics). Given a learning model $\mathcal{M} = (X, \mathscr{O}, \mathbb{L}, \|\cdot\|)$ and an epistemic scenario (x, U), the semantics of the language $\mathfrak{L}$ is given by a binary relation $(x, U) \models_{\mathcal{M}} \varphi$ between epistemic scenario and formulas, called the *satisfaction relation*, as well as a *truth set* (interpretation) $[\![\varphi]\!]^U_{\mathcal{M}} =: \{x \in U \mid (x, U) \models_{\mathcal{M}} \varphi\}$, for all formulas φ. We typically omit the subscript, simply writing $(x, U) \models \varphi$ and $[\![\varphi]\!]^U$, whenever the model $\mathcal{M}$ is understood. The satisfaction relation is defined by the following recursive clauses:

$$
\begin{array}{ll}
(x, U) \models p & \text{iff } \ x \in \|p\| \\
(x, U) \models o & \text{iff } \ x \in \|o\| \\
(x, U) \models L(o_1, \ldots, o_n) & \text{iff } \ x \in \mathbb{L}(U, \|o_1\|, \ldots, \|o_n\|)
\end{array}
$$

$$
\begin{array}{lll}
(x,U) \models \neg\varphi & \text{iff} & (x,U) \not\models \varphi \\
(x,U) \models \varphi \wedge \psi & \text{iff} & (x,U) \models \varphi \text{ and } (x,U) \models \psi \\
(x,U) \models K\varphi & \text{iff} & (\forall y \in U)\,((y,U) \models \varphi) \\
(x,U) \models \Box\varphi & \text{iff} & (\forall O \in \mathscr{O})\,(x \in O \subseteq U \text{ implies } (x,O) \models \varphi) \\
 & \text{iff} & (\forall O \in \mathscr{O})\,(x \in O \text{ implies } (x, U \cap O) \models \varphi) \\
(x,U) \models [o]\varphi & \text{iff} & (x \in \|o\| \text{ implies } (x, U \cap \|o\|) \models \varphi)
\end{array}
$$

where $p \in \mathrm{Prop}$, $o, o_1, \ldots, o_n \in \mathrm{Prop}_{\mathscr{O}}$, $\vec{o} \in \mathrm{Prop}^*_{\mathscr{O}}$, and where we used the notation $\mathbb{L}(O_1, \ldots, O_n) := \mathbb{L}(O_1 \cap \ldots \cap O_n)$ introduced above. We say that a formula φ is *valid in a learning model* $\mathcal{M}$, and write $\mathcal{M} \models \varphi$, if $(x,U) \models_{\mathcal{M}} \varphi$ for all epistemic scenarios $(x,U) \in ES(\mathcal{M})$. We say φ is *validable in an intersection model* $(X, \mathscr{O}, \|\cdot\|)$, and write $(X, \mathscr{O}, \|\cdot\|) \models \varphi$, if there exists some learner $\mathbb{L} : \mathscr{O} \to \mathcal{P}(X)$ such that φ is valid in the learning model $(X, \mathscr{O}, \mathbb{L}, \|\cdot\|)$. We say φ is *valid*, and write $\models \varphi$, if it is valid in *all* learning models.

Abbreviations: For any string $\vec{o} = (o_1, \ldots, o_n) \in \mathrm{Prop}^*_{\mathscr{O}}$ of observational variables, and any formula φ we set:

$$\bigwedge \vec{o} \;:=\; o_1 \wedge \ldots \wedge o_n \text{ (with the convention that } \bigwedge \lambda := \top)$$

$$\vec{o} \Leftrightarrow \vec{u} \;:=\; K\left((\bigwedge \vec{o}) \leftrightarrow (\bigwedge \vec{u})\right) \quad \text{(extensional equivalence of observations)}$$

$$[\vec{o}]\varphi \;:=\; [o_1]\ldots[o_n]\varphi \text{ (with the convention that } [\lambda]\varphi := \varphi\text{); similarly for } \langle \vec{o} \rangle)$$

$$B^{\vec{o}}\varphi \;:=\; K(L(\vec{o}) \to \varphi)$$

$$B\varphi \;:=\; B^{\lambda}\varphi$$

(where λ is the empty string). We read $B\varphi$ as the 'observer *believes* φ' (given no observations), and $B^{\vec{o}}\varphi$ as 'the observer *believes* φ *conditional on evidence* $\vec{o}$'.

2.2 Axiomatization and Proof System

We will now provide the formal definition of our proposed system **L** of the Dynamic Logic for Learning Theory (DLLT) by listing the axioms and derivation rules, see Table 1 below. Given a formula $\varphi \in \mathfrak{L}$, we denote by P_φ and O_φ the set of all propositional variables and observational variables respectively occurring in φ (we will use the same notation for the necessity and possibility forms defined below).

The intuitive reading of the $S5$ axioms for epistemic modality K expresses the fact that K is factive and (positively and negatively) introspective. The intuitive nature of the reduction axioms should be as in Public Announcement Logic [2], when we take into account the natural atomic behaviour of observables. The learning axioms (CC), (EC) and (SP) express pre-conditions in formal learning theory on *observations*, namely that: they are truthful observations about the world (CC); that the history of observations is irrelevant for the learner, except for the extensional evidence provided by observations (EC) and that conjectures fit what is observed (SP). Since the effort modality $\Box$ quantifies over possible

Table 1. The axiom schemas for the Dynamic Logic of Learning Theory, **L**

	Basic axioms:	
(P)	All instantiations of propositional tautologies	
(K_K)	$K(\varphi \to \psi) \to (K\varphi \to K\psi)$	
(T_K)	$K\varphi \to \varphi$	
(4_K)	$K\varphi \to KK\varphi$	
(5_K)	$\neg K\varphi \to K\neg K\varphi$	
$(\mathrm{K}_{[o]})$	$[o](\psi \to \chi) \to ([o]\psi \to [o]\chi)$	
	Basic rules:	
(MP)	From $\vdash \varphi$ and $\vdash \varphi \to \psi$, infer $\vdash \psi$	
(Nec_K)	From $\vdash \varphi$, infer $\vdash K\varphi$	
$(\mathrm{Nec}_{[o]})$	From $\vdash \varphi$, infer $\vdash [o]\varphi$	
	Learning axioms:	
(CC)	$(\bigwedge \vec{\sigma}) \to \langle K \rangle L(\vec{\sigma})$	Consistency of conjectures
(EC)	$(\vec{\sigma} \Leftrightarrow \vec{u}) \to (L(\vec{\sigma}) \leftrightarrow L(\vec{u}))$	Extensionality of conjectures
(SP)	$L(\vec{\sigma}) \to \bigwedge \vec{\sigma}$	Success postulate
	Reduction axioms:	
(R_p)	$[o]p \leftrightarrow (o \to p)$	
(R_u)	$[o]u \leftrightarrow (o \to u)$	
(R_L)	$[o]L(\vec{u}) \leftrightarrow (o \to L(o, \vec{u}))$	
$(\mathrm{R}_\neg)$	$[o]\neg\psi \leftrightarrow (o \to \neg[o]\psi)$	
(R_K)	$[o]K\psi \leftrightarrow (o \to K[o]\psi)$	
$(\mathrm{R}_\Box)$	$[o]\Box\psi \leftrightarrow \Box[o]\psi$	
	Effort axiom and rule:	
$(\Box$-Ax)	$\Box\varphi \to [\vec{\sigma}]\varphi$ for all $\vec{\sigma} \in \mathrm{Prop}^*_{\mathscr{O}}$ arbitrary	
$(\Box$-Rule)	From $\vdash \psi \to [o]\varphi$, infer $\vdash \psi \to \Box\varphi$, where $o \notin O_\psi \cup O_\varphi$	

observations, we could think of the Effort axiom and the Effort rule as elimination and introduction rules for $\Box$. The former one expresses the fact that: if a property is stably true then it holds after any observations. Finally, the latter says that, if a property holds after any arbitrary observation, it is stably true.

So each of our axioms is simple and readable and has a transparent intuitive interpretation, in contrast to other axiomatizations of (the less expressive) Subset Space Logic over intersection spaces (i.e., the L-free analogues of our models). Having such a simple axiomatization is one of the advantages brought by the addition of dynamic observation modalities. See more discussion of this issue in the Conclusions section.

We now give reduction laws for strings of observational variables in $\mathrm{Prop}^*_{\mathscr{O}}$.

Proposition 1 (Reduction laws for strings of observational variables). *The following reduction laws are provable in* $\mathbf{L}$ *for all* $\varphi \in \mathfrak{L}$*:*

1. $[\vec{u}]p \leftrightarrow (\bigwedge \vec{u} \rightarrow p)$
2. $[\vec{u}]o \leftrightarrow (\bigwedge \vec{u} \rightarrow o)$
3. $[\vec{u}]L(\vec{o}) \leftrightarrow (\bigwedge \vec{u} \rightarrow L(\vec{u}, \vec{o}))$
4. $[\vec{u}]\neg\varphi \leftrightarrow (\bigwedge \vec{u} \rightarrow \neg[\vec{u}]\varphi)$
5. $[\vec{u}]K\varphi \leftrightarrow (\bigwedge \vec{u} \rightarrow K[\vec{u}]\varphi)$
6. $[\vec{u}]\Box\varphi \leftrightarrow \Box[\vec{u}]\varphi$

Proposition 2. *The following reduction laws are provable in* $\mathbf{L}$ *for all formulas* $\varphi \in \mathfrak{L}$*:*

$(R_\wedge)$	$[u](\varphi \wedge \psi) \leftrightarrow ([u]\varphi \wedge [u]\psi)$
$(R_{\rightarrow\wedge})$	$[\vec{u}](\varphi \wedge \psi) \leftrightarrow ([\vec{u}]\varphi \wedge [\vec{u}]\psi)$
$(\langle o \rangle)$	$\langle o \rangle\psi \leftrightarrow (o \wedge [o]\psi)$
$(\langle \vec{o} \rangle)$	$\langle \vec{o} \rangle\psi \leftrightarrow (\bigwedge \vec{o} \wedge [\vec{o}]\psi)$

In our framework, belief (B) and conditional beliefs ($B^{\vec{o}}\varphi$) are defined in terms of the operators K and L. The axiomatic system $\mathbf{L}$ given in Table 1 over the language $\mathfrak{L}$ can therefore derive the properties describing the type of belief and conditional belief modalities we intend to formalize in this paper. More precisely, as stated in Proposition 3, the system $\mathbf{L}$ yields the standard belief system KD45 for B. More generally, if we replace the D axiom for a 'weaker' version $D' := (\langle K \rangle \vec{o} \rightarrow \neg B^{\vec{o}}\bot)$ then we have a weak version of KD45 system for conditional belief $B^{\vec{o}}$, namely wKD45.

Proposition 3 (wKD45 axioms and rules for Conditional Belief). *The standard axioms and rules of the doxastic logic KD45 are derivable for our belief operator* B *in the system* $\mathbf{L}$*. More generally, the axioms of rules of the weaker system wKD45 are derivable for our conditional-belief operator* B^o *in the system* $\mathbf{L}$*.*

Proposition 4. *The S4 axioms for the effort modality* $\Box$ *are derivable in* $\mathbf{L}$*.*

3 Soundness and Completeness

In this section we prove soundness and completeness. Note that, although our logic is more expressive than Subset Space Logic (interpreted on intersection spaces), our completeness proof is *much simpler*, via a *canonical construction*: this is one of the advantages of having the (expressively redundant) dynamic observation modalities!

Soundness. We first prove soundness, for which we need the following lemma. Note that by the definition of the valuation $||\cdot||$ in a learning model $\mathcal{M}$, we have that for all $U \in \mathscr{O}$, $U \cap ||p|| = [\![p]\!]^U_{\mathcal{M}}$ and $U \cap ||o|| = [\![o]\!]^U_{\mathcal{M}}$ for all $p \in$ Prop and $o \in \text{Prop}_{\mathscr{O}}$ in $\mathfrak{L}$.

Lemma 1. *Let* $\mathcal{M} = (X, \mathscr{O}, \mathbb{L}, \|\cdot\|)$ *and* $\mathcal{M} = (X, \mathscr{O}, \mathbb{L}, \|\cdot\|')$ *be two learning models and* $\varphi \in \mathfrak{L}$ *such that* $\mathcal{M}$ *and* $\mathcal{M}'$ *differ only in the valuation of some* $o \notin O_\varphi$*. Then, for all* $U \in \mathscr{O}$*, we have* $[\![\varphi]\!]^U_{\mathcal{M}} = [\![\varphi]\!]^U_{\mathcal{M}'}$*.*

Theorem 1. *The system* **L** *in Table 1 is sound wrt the class of learning models.*

Completeness. We now move to the completeness proof for our logic **L**, which will be shown via a 'simple' canonical model construction. But its simplicity is deceiving, due to two main technical differences between our construction and the standard canonical model from Modal Logic. First, this is *not* a relational (Kripke) model, but a *neighborhood model*; so the closest analogue is the type of canonical construction used in Topological Modal Logic or Neighborhood Semantics [1]. Second, the standard notion of maximally consistent theory is *not* very useful for our logic, since such theories do not 'internalize' the $\Box$-Rule. To do this, we need instead to consider *'witnessed' (maximally consistent) theories*, in which every occurrence of a $\Diamond\varphi$ in any "existential context" is 'witnessed' by some $\langle o\rangle\varphi$ (with o observational variable). The appropriate notion of "existential contexts" is represented by *possibility forms*, in the following sense:

Definition 4 ('Pseudo-modalities': necessity and possibility forms). *The* set of necessity-form expressions *of our language is given by* $NF_{\mathfrak{L}} := (\{\varphi \to \mid \varphi \in \mathfrak{L}\} \cup \{K\} \cup \text{Prop}_{\mathscr{O}})^*$. *For any finite string* $s \in NF_{\mathfrak{L}}$, *we define 'pseudo-modalities'* $[s]$ *(called* necessity form*) and* $\langle s\rangle$ *(called* possibility form*), that generalize our dynamic modalities* $[o]$ *and* $\langle o\rangle$. *These pseudo-modalities are just functions mapping any formula* $\varphi \in \mathfrak{L}$ *to another formula* $[s]\varphi \in \mathfrak{L}$, *and respectively* $\langle s\rangle\varphi \in \mathfrak{L}$. *Necessity forms are defined recursively, by putting:* $[\lambda]\varphi := \varphi$, $[\varphi \to, s]\varphi := \varphi \to [s]\varphi$, $[K, s]\varphi := K[s]\varphi$, $[o, s]\varphi := [o][s]\varphi$. *As for possibility forms, we put* $\langle s\rangle\varphi := \neg[s]\neg\varphi$.

Lemma 2. *For every necessity form* $[s]$, *there exist an observational variable* $o \in \mathfrak{L}$ *and a formula* $\psi \in \mathfrak{L}$ *such that for all* $\varphi \in \mathfrak{L}$, *we have:* $\vdash [s]\varphi$ *iff* $\vdash \psi \to [o]\varphi$.

Proof. The proof follows similarly as in [2, Lemma 4.8], but is even simpler since our dynamic modalities only involve observational variables which are atomic formulas.

Lemma 3. *The following rule is admissible in* **L***:*

$$\textit{if } \vdash [s][o]\varphi \textit{ then } \vdash [s]\Box\varphi, \textit{ where } o \notin O_s \cup O_\varphi.$$

Proof. Suppose $\vdash [s][o]\varphi$ where $o \notin O_s \cup O_\varphi$. Then, by Lemma 2, there exist $u \in \text{Prop}_{\mathscr{O}}$ and $\psi \in \mathfrak{L}$ such that $\vdash \psi \to [u][o]\varphi$. Thus we get $\vdash \psi \to [u, o]\varphi$. It is not hard to see that $\vdash [u, o]\varphi \leftrightarrow [o, u]\varphi$ (which follows by subformula induction on φ, using the corresponding reduction axiom given in Proposition 1, and the fact that $\vdash u \wedge o \leftrightarrow o \wedge u$). Therefore, $\vdash \psi \to [u, o]\varphi$ iff $\vdash \psi \to [o, u]\varphi$. Hence we obtain $\vdash \psi \to [o, u]\varphi$, i.e., $\vdash \psi \to [o][u]\varphi$. By the construction of the formulas ψ and u, we know that $O_\psi \cup O_u \subseteq O_s$, and so $o \notin O_\psi \cup \{u\} \cup O_\varphi$. Therefore, by the Effort rule ($\Box$-Rule) we have $\vdash \psi \to \Box[u]\varphi$, implying, by the reduction axiom ($\text{R}_\Box$), that $\vdash \psi \to [u]\Box\varphi$. Applying again Lemma 2, we obtain $\vdash [s]\Box\varphi$.

Definition 5. *For every countable set* O*, let* $\mathfrak{L}^{\mathrm{O}}$ *be the language of the logic* $\mathbf{L}^{\mathrm{O}}$ *based only on the observational variables in* O *(i.e. having as set of observational variables* $\mathrm{Prop}_{\mathscr{O}} := \mathrm{O}$*). Let* $NF^{\mathrm{O}}_{\mathfrak{L}}$ *denote the set of necessity-form expressions of* $\mathbf{L}^{\mathrm{O}}$ *(i.e. necessity forms involving only observational variables in* O*). An* O-theory *is a* consistent *set of formulas in* $\mathfrak{L}^{\mathrm{O}}$*. Here, 'consistent' means consistent with respect to the axiomatization* $\mathbf{L}$ *formulated for* $\mathfrak{L}^{\mathrm{O}}$*. A* maximal O-theory *is an* O*-theory* Γ *that is maximal with respect to* $\subseteq$ *among all* O*-theories; in other words,* Γ *cannot be extended to another* O*-theory. An* O-witnessed theory *is an* O*-theory* Γ *such that, for every* $s \in NF^{\mathrm{O}}_{\mathfrak{L}}$ *and* $\varphi \in \mathfrak{L}^{\mathrm{O}}$*, if* $\langle s\rangle\Diamond\varphi$ *is consistent with* Γ *then there is* $o \in \mathrm{O}$ *such that* $\langle s\rangle\langle o\rangle\varphi$ *is consistent with* Γ*. A* maximal O-witnessed theory Γ *is an* O*-witnessed theory that is not a proper subset of any* O*-witnessed theory.*

Lemma 4. *For every* $\Gamma \subseteq \mathfrak{L}^{\mathrm{O}}$*, if* Γ *is an* O*-theory and* $\Gamma \not\vdash \neg\varphi$ *for some* $\varphi \in \mathfrak{L}^{\mathrm{O}}$*, then* $\Gamma \cup \{\varphi\}$ *is an* O*-theory. Moreover, if* Γ *is* O*-witnessed, then* $\Gamma \cup \{\varphi\}$ *is also* O*-witnessed.*

Lemma 5 (Lindenbaum's Lemma). *Every* O*-witnessed theory* Γ *can be extended to a maximal* O*-witnessed theory* T_Γ*.*

Lemma 6 (Extension Lemma). *Let* O *be a set of observational variables and* O′ *be a countable set of* fresh *observational variables, i.e.,* $\mathrm{O} \cap \mathrm{O}' = \emptyset$*. Let* $\widetilde{\mathrm{O}} = \mathrm{O} \cup \mathrm{O}'$*. Then, every* O*-theory* Γ *can be extended to a* $\widetilde{\mathrm{O}}$*-witnessed theory* $\widetilde{\Gamma} \supseteq \Gamma$*, and hence to a maximal* $\widetilde{\mathrm{O}}$*-witnessed theory* $T_\Gamma \supseteq \Gamma$*.*

We are now ready to build the canonical model.

Canonical Model for T_0. For any consistent set of formulas Φ, consider a maximally consistent O-witnessed extension $T_0 \supseteq \Phi$. As our canonical set of worlds, we take the set $\mathcal{X}^c := \{T : T$ maximally consistent O-witnessed theory with $T \sim_K T_0\}$, where we put $T \sim_K T'$ iff $\forall\varphi \in \mathfrak{L}_{\mathrm{O}}\left((K\varphi) \in T \rightarrow \varphi \in T'\right)$. As usual, it is easy to see (given the $S5$ axioms for K) that $\sim_K$ is an equivalence relation. For any formula φ, we use the notation $\widehat{\varphi} := \{T \in \mathcal{X}^c : \varphi \in T\}$. In particular, for any observational variable $o \in \mathrm{O}$, we have $\widehat{o} = \{T \in \mathcal{X}^c : o \in T\}$. We can generalize this notation to *finite sequences* $\vec{o} = (o_1, \ldots, o_n) \in \mathrm{O}^*$ of observational variables, by putting: $\widehat{\vec{o}} := \{T \in \mathcal{X}^c : o_1, \ldots, o_n \in T\}$.

As canonical set of information states, we take $\mathscr{O}^c := \{\widehat{\vec{o}} : \vec{o} \in \mathrm{O}^*\}$. Finally, our canonical learner is given by $\mathbb{L}^c(\widehat{\vec{o}}) := \widehat{L(\vec{o})}$, and the canonical valuation $\|\cdot\|_c$ is given as $||p||_c = \widehat{p}$ and $||o||_c = \widehat{o}$. The learning model $\mathcal{M}^c = (\mathcal{X}^c, \mathscr{O}^c, \mathbb{L}^c, \|\cdot\|_c)$ is called the *canonical model*. Note that we use c as a subindex instead of a superindex for the canonical valuation $\|\cdot\|_c$, this is in order to avoid confusion with our 'open-restriction' notation for the truth set of a formula $[\![\varphi]\!]^U$.

Before proving that the canonical model is well-defined, we need the following.

Lemma 7. *For every maximal* O*-witnessed theory* T*, the set* $\{\theta : K\theta \in T\}$ *is an* O*-witnessed theory.*

Lemma 8. *Let* $T \in \mathcal{X}^c$*. Then,* $K\varphi \in T$ *iff* $\varphi \in S$ *for all* $S \in \mathcal{X}^c$*.*

Corollary 1. *Let $T \in \mathcal{X}^c$. Then, $\langle K\rangle\varphi \in T$ iff there is $S \in \mathcal{X}^c$, such that $\varphi \in S$.*

Proposition 5. *The canonical model is well-defined.*

Proof. We need to show that the following properties hold:

1. *If $F = \{\widehat{\vec{o_1}}, \ldots, \widehat{\vec{o_m}}\} \subseteq \mathcal{O}^c$ is finite then $\bigcap F \in \mathcal{O}^c$*: Let $F = \{\widehat{\vec{o_1}}, \ldots, \widehat{\vec{o_m}}\} \subseteq \mathcal{O}^c$. It is easy to see that $\bigcap\{\widehat{\vec{o_1}}, \ldots, \widehat{\vec{o_m}}\} = \widehat{\vec{o}}$, where $\vec{o}$ is the concatenation of all the $\vec{o_i}$'s with $1 \leq i \leq m$. Since each $\vec{o_i}$ is finite, $\vec{o}$ is finite. Therefore, by the definition of $\mathcal{O}^c$, we obtain $\bigcap\{\widehat{\vec{o_1}}, \ldots, \widehat{\vec{o_m}}\} = \widehat{\vec{o}} \in \mathcal{O}^c$.
2. $\mathbb{L}^c$ *is a well-defined function and a learner*: For this, note that $\mathbb{L}^c(\widehat{\vec{o}}) := \widehat{L(\vec{o})} \subseteq X^c$. We will first prove that:
   (2a) *if $\widehat{\vec{o}} = \widehat{\vec{u}}$ then $\mathbb{L}^c(\widehat{\vec{o}}) = \mathbb{L}^c(\widehat{\vec{u}})$*: Suppose $\widehat{\vec{o}} = \widehat{\vec{u}}$. This means that $(\forall T \in X^c)(\bigwedge \vec{o} \in T$ iff $\bigwedge \vec{u} \in T)$. Therefore, we obtain $\vdash \bigwedge \vec{o} \leftrightarrow \bigwedge \vec{u}$. Then, by (Nec$_K$), we have $\vdash K\,(\bigwedge \vec{o} \leftrightarrow \bigwedge \vec{u})$, i.e., $\vdash \vec{o} \Leftrightarrow \vec{u}$. Since $\mathbb{L}^c(\widehat{\vec{o}}) := \widehat{L(\vec{o})}$, showing $\mathbb{L}^c(\widehat{\vec{o}}) = \mathbb{L}^c(\widehat{\vec{u}})$ boils down to showing that $\widehat{L(\vec{o})} = \widehat{L(\vec{u})}$, i.e., that $\vdash L(\vec{o}) \leftrightarrow L(\vec{u})$, which follows from axiom (EC) and the assumption that $\vdash \vec{o} \Leftrightarrow \vec{u}$.
   Next, we must prove that
   (2b) $\mathbb{L}^c$ *is a learner*, i.e., $\mathbb{L}^c$ satisfies the properties of a learner given in Definition 2. To show this, we first check that $\mathbb{L}^c(\widehat{\vec{o}}) \subseteq \widehat{\vec{o}}$ holds. Let $T \in \mathbb{L}^c(\widehat{\vec{o}})$. This means, by the definition of $\mathbb{L}^c(\widehat{\vec{o}})$, that $L(\vec{o}) \in T$. Since $(L(\vec{o}) \rightarrow \bigwedge \vec{o}) \in T$ (by the axiom (SP)), we have that $\bigwedge \vec{o} \in T$. Therefore, as T is maximally consistent, we obtain $o_1, \ldots, o_m \in T$ for $\vec{o} = (o_1, \ldots, o_m)$, meaning that $\vec{o} \in T$. Thus, $T \in \widehat{\vec{o}}$. Finally we show that if $\widehat{\vec{o}} \neq \emptyset$ then $\mathbb{L}^c(\widehat{\vec{o}}) \neq \emptyset$. Suppose $\widehat{\vec{o}} \neq \emptyset$, i.e., there is $T \in X^c$ with $T \in \widehat{\vec{o}}$. This means, by the definition of $\widehat{\vec{o}}$, that $\vec{o} \in T$. Then, since T is a maximal consistent theory, we have $\bigwedge \vec{o} \in T$, and $((\bigwedge \vec{o}) \rightarrow \langle K\rangle L(\vec{o})) \in T$ (by the axiom (CC)). Thus we obtain $\langle K\rangle L(\vec{o}) \in T$. Then, by Corollary 1, there is $S \in X^c$ such that $L(\vec{o}) \in S$. Thus, by the definition of $\widehat{L(\vec{o})}$, we have $S \in \widehat{L(\vec{o})}$, and therefore, $\widehat{L(\vec{o})} = \mathbb{L}^c(\widehat{\vec{o}}) \neq \emptyset$.

Our aim is to prove a Truth Lemma for the canonical model, that will immediately imply completeness, as usual. But for this we first need the following result.

Lemma 9. *Let $T \in \mathcal{X}^c$. Then, $\Box\varphi \in T$ iff $[\vec{u}]\varphi \in T$ for all $\vec{u} \in \mathrm{Prop}_O^*$.*

We now proceed to our key result:

Lemma 10 (Truth Lemma). *For all formulas φ, all $T \in X^c$ and all $\widehat{\vec{o}} \in \mathcal{O}^c$, we have:*

$$\langle \vec{o}\rangle\varphi \in T \quad \textit{iff} \quad (T, \widehat{\vec{o}}) \models_{\mathcal{M}^c} \varphi.$$

Proof. The proof is by induction over subformulas. The cases for propositional and observational variables, as well as for Boolean connectives are as usual. So we only check the remaining cases. At each step of the proof, $\bigwedge \vec{o} \in T$ guarantees that the pair $(T, \widehat{\vec{o}})$ is a well-defined epistemic scenario of the canonical model since $\widehat{\bigwedge \vec{o}} = \widehat{\vec{o}}$.

– Case $\varphi := L(\vec{u})$.

$$\begin{aligned}
\langle \vec{o} \rangle L(\vec{u}) \in T \text{ iff } & (\bigwedge \vec{o} \wedge [\vec{o}]L(\vec{u})) \in T && \text{(Proposition 2-}(\langle \vec{o} \rangle)) \\
\text{iff } & (\bigwedge \vec{o} \wedge L(\vec{o}, \vec{u})) \in T && \text{(Proposition 1.3.)} \\
\text{iff } & \bigwedge \vec{o} \in T \text{ and } L(\vec{o}, \vec{u}) \in T && \\
\text{iff } & T \in \widehat{\vec{o}} \text{ and } T \in \widehat{L(\vec{o}, \vec{u})} = \mathbb{L}^c(\widehat{\vec{o}}, \widehat{\vec{u}}) && \text{(since } \widehat{\bigwedge \vec{o}} = \widehat{\vec{o}}) \\
\text{iff } & (T, \widehat{\vec{o}}) \models_{\mathcal{M}^c} L(\vec{u}) && \text{(by the semantics of } L)
\end{aligned}$$

– Case $\varphi := K\psi$.

$$\begin{aligned}
\langle \vec{o} \rangle K\psi \in T \text{ iff } & (\bigwedge \vec{o} \wedge K[\vec{o}]\psi) \in T && \text{(Propositions 2-}(\langle \vec{o} \rangle)\text{) and 1.5)} \\
\text{iff } & \bigwedge \vec{o} \in T \text{ and } K[\vec{o}]\psi \in T && \\
\text{iff } & \bigwedge \vec{o} \in T \text{ and } (\forall S \sim_K T)([\vec{o}]\psi \in S) && \text{(by Lemma 8)} \\
\text{iff } & \bigwedge \vec{o} \in T \text{ and } (\forall S \in \widehat{\vec{o}})(\langle \vec{o} \rangle \psi \in S) && \text{(Propositions 2-}(\langle \vec{o} \rangle)) \\
\text{iff } & \widehat{\vec{o}} \in T \text{ and } (\forall S \in \widehat{\vec{o}})((S, \widehat{\vec{o}}) \models \psi) && \text{(by } \widehat{\bigwedge \vec{o}} = \widehat{\vec{o}} \text{ and I.H)} \\
\text{iff } & (T, \widehat{\vec{o}}) \models_{\mathcal{M}^c} K\psi && \text{(by the semantics of } K)
\end{aligned}$$

– Case $\varphi := \langle \vec{u} \rangle \psi$.

$$\begin{aligned}
\langle \vec{o} \rangle \langle \vec{u} \rangle \psi \in T \text{ iff } & \langle \vec{o}, \vec{u} \rangle \psi \in T && \text{(by the abbreviation for } \langle \vec{o} \rangle \psi) \\
\text{iff } & (\bigwedge \vec{o} \wedge \bigwedge \vec{u}) \wedge \langle \vec{o}, \vec{u} \rangle \psi \in T && \text{(Propositions 2-}(\langle \vec{o} \rangle)) \\
\text{iff } & (\bigwedge \vec{o} \wedge \bigwedge \vec{u}) \in T \text{ and } \langle \vec{o}, \vec{u} \rangle \psi \in T && \\
\text{iff } & T \in \widehat{\vec{o}} \cap \widehat{\vec{u}} \text{ and } \langle \vec{o}, \vec{u} \rangle \psi \in T && \text{(since } \widehat{\bigwedge \vec{o} \wedge \bigwedge \vec{u}} = \widehat{\vec{o}} \cap \widehat{\vec{u}}) \\
\text{iff } & T \in \widehat{(\vec{o}, \vec{u})} \text{ and } (T, \widehat{(\vec{o}, \vec{u})}) \models \psi && \text{(since } \widehat{(\vec{o}, \vec{u})}) = \widehat{\vec{o}} \cap \widehat{\vec{u}}) \\
\text{iff } & T \in ||\vec{o}, \vec{u}||_c \text{ and } (T, ||\vec{o}, \vec{u}||_c) \models \psi && \\
& && \text{(by the definition of } || \cdot ||_c) \\
\text{iff } & (T, ||\vec{o}, \vec{u}||_c) \models \langle \vec{o}, \vec{u} \rangle \psi && \text{(by the semantics)} \\
\text{iff } & (T, ||\vec{o}, \vec{u}||_c) \models \langle \vec{o} \rangle \langle \vec{u} \rangle \psi && \text{(by the abbreviation for } \langle \vec{o} \rangle \psi)
\end{aligned}$$

- Case $\varphi := \Box\psi$.
  ($\Leftarrow$) Suppose $\langle\vec{o}\rangle\Box\psi \in T$. Then, by Propositions 2-($\langle\vec{o}\rangle$) and 1.6, we obtain that (1) $\bigwedge\vec{o} \in T$, i.e., $T \in \widehat{\vec{o}}$, and (2) $\Box[\vec{o}]\psi \in T$. Thus, by Lemma 9 and (2), we have $[\vec{u}][\vec{o}]\psi \in T$, i.e., $[\vec{u},\vec{o}]\psi \in T$, for all $\vec{u} \in \text{Prop}_O^*$. Now let $O \in \mathscr{O}^c$ such that $T \in O$. By the construction of $\mathscr{O}^c$, we know that $O = \widehat{\vec{v}}$ for some $\vec{v} \in \text{Prop}_O^*$. We want to show that $(T, \widehat{\vec{o}} \cap \widehat{\vec{v}}) \models \psi$. Since $T \in \widehat{\vec{o}} \cap \widehat{\vec{v}}$ and $\mathcal{M}^c$ is an intersection space, we know that $(T, \widehat{\vec{o}} \cap \widehat{\vec{v}})$ is a well-defined epistemic scenario. $T \in \widehat{\vec{o}} \cap \widehat{\vec{v}}$ also implies that $(\bigwedge\vec{o} \wedge \bigwedge\vec{v}) \in T$ as in the above case. Hence, by Proposition 2-($\langle\vec{o}\rangle$) and the fact that $[\vec{v},\vec{o}]\psi \in T$, we obtain $\langle\vec{v},\vec{o}\rangle\psi \in T$. Then, by I.H, we obtain $(T, \widehat{\vec{v}} \cap \widehat{\vec{o}}) \models \psi$ as in the previous case. Therefore, by the semantics of $\Box$, we obtain $(T, \widehat{\vec{o}}) \models \Box\psi$.
  ($\Rightarrow$) Suppose $(T, \widehat{\vec{o}}) \models \Box\psi$. This means, by the definition of $\mathscr{O}^c$, that for all $\vec{u} \in \text{Prop}_O^*$, if $T \in \widehat{\vec{u}}$ then $(T, \widehat{\vec{o}} \cap \widehat{\vec{u}}) \models \psi$. Now let $\vec{v} \in \text{Prop}_O^*$ such that $T \in \widehat{\vec{v}}$. Therefore, $T \in \widehat{\vec{v}} \cap \widehat{\vec{o}}$. Since $(\vec{v},\vec{o}) \in \text{Prop}_O^*$ and $\widehat{\vec{v}} \cap \widehat{\vec{o}} = \widehat{(\vec{v},\vec{o})}$, we obtain by the assumption that $(T, \widehat{\vec{v}} \cap \widehat{\vec{o}}) \models \psi$. Thus, by I.H., we have $\langle\vec{v},\vec{o}\rangle\psi \in T$. As $\vdash \langle\vec{v},\vec{o}\rangle\psi \to [\vec{v},\vec{o}]\psi$ and T is maximal, we obtain $[\vec{v},\vec{o}]\psi \in T$, i.e., $[\vec{v}][\vec{o}]\psi \in T$. Hence, by Lemma 9, we have $\Box[\vec{o}]\psi \in T$. Then, by Proposition 1.6, the fact that $\bigwedge\vec{o} \in T$ and Propositions 2-($\langle\vec{o}\rangle$) and, we obtain $\langle\vec{o}\rangle\Box\psi \in T$.

Theorem 2. **L** *is complete with respect to the class of all learning models.*

Proof. Let φ be an **L**-consistent formula, i.e., it is an O_φ-theory. Then, by Lemma 6, it can be extended to a maximal O_φ-witnessed theory T. Then, we have $\langle\lambda\rangle\varphi \in T$ where λ is the empty string, i.e., $T \in \widehat{\langle\lambda\rangle\varphi}$. Note that $\widehat{\bigwedge\lambda} = \bigcap\emptyset = X^c$. Then, by Truth Lemma (Lemma 10), we obtain that $(T, X^c) \models_{\mathcal{M}^c} \varphi$, where $\mathcal{M}^c = (X^c, \mathscr{O}^c, \mathbb{L}^c, \|\cdot\|_c)$ is the canonical model for T. This proves completeness.

4 Expressivity

We first investigate how various notions of learnability can be expressed in our language. In fact, the following result was already noticed in [17]:

Proposition 6. $\Diamond Kp$ *is true at* (x, U) *in a model* $\mathcal{M}$ *iff* $\|p\|$ *is* learnable with certainty *at state* x*. Similarly,* $p \to \Diamond Kp$ *is valid (i.e. true at all epistemic scenarios) in a model* $\mathcal{M}$ *iff* $\|p\|$ *is* verifiable with certainty *(i.e. 'finitely identifiable' in the sense of FLT* [10,14]*). A similar statement holds for* falsifiability with certainty.

Proof. As we know from Sect. 1.1, $\|p\|$ is learnable with certainty x iff $x \in Int\|p\|$, and $\|p\|$ is verifiable with certainty iff it is open in the topology generated by $\mathscr{O}$. It is well-known [17] that these properties are expressible in SSL via the above validities.

In particular, the following validity of our logic expresses the fact that *all observable properties are verifiable with certainty*:

$$o \to \Diamond K o.$$

By adding the learning operator to subset space logic, DLLT can capture, not only belief, but also the various inductive notions of knowledge and learnability:

Proposition 7. *[Inductive notions of knowledge and learnability]*

- $\Box Bp$ *holds at* (x, U) *in a model* $\mathcal{M}$ *iff the learner* $\mathbb{L}$ *has* undefeated belief *in* $\|p\|$ *(at world* x *in information state* U*). Hence,* $p \wedge \Box Bp$ *captures* inductive knowledge *of* p*, and so* $p \wedge \Diamond\Box Bp$ *captures* inductive learnability *of* p *by learner* $\mathbb{L}$.
- *Similarly,* $p \to \Diamond\Box Bp$ *is valid in a model* $\mathcal{M}$ *iff* $\|p\|$ *is* inductively verifiable by L. *For the corresponding generic notion:* $\|p\|$ is inductively verifiable *(by* some *learner) iff* $p \to \Diamond\Box Bp$ is validable in the intersection space $(X, \mathscr{O})$. *Similar statements hold for* inductive falsifiability.
- *Finally,* $\Diamond L(\lambda)$ *is true if (given enough observations) the observer* will eventually reach a true conjecture *(though he might later fall again into false ones); and similarly,* $\Diamond\Box L(\lambda)$ *is true if (given enough observations) the observer will* eventually produce only true conjectures thereafter.

Proof. This is an easy verification, given the relevant definitions and our semantics.

As usual in Dynamic Epistemic Logic, the dynamic ‘observation’ modalities $[u]\varphi$ are only a convenient way to express complex properties in a succinct manner, but they can in principle be eliminated. To show this, we first need the following lemma.

Lemma 11. *There is a well-founded strict partial order* $<$ *on formulas (called ‘complexity order’), satisfying the following conditions:*

- *if* φ *is a (proper) subformula of* ψ *then* $\varphi < \psi$
- $(u \to p) < [u]p$
- $(u \to o) < [u]o$
- $L(u, \vec{o}) < [u]L(\vec{o})$
- $([u]\varphi \wedge [u]\psi) < [u](\varphi \wedge \psi)$
- $(u \to K[u]\varphi) < [u]K\varphi$
- $\Box[u]\varphi < [u]\Box\varphi$

Proposition 8 *(Expressivity)***.** *The above language is co-expressive with the one obtained by removing all dynamic modalities* $[u]\varphi$*. Moreover, this can be proved in the above proof system: for every formula* φ *there exists some formula* φ' *free of any dynamic modalities, such that* $\varphi \leftrightarrow \varphi'$ *is a theorem in the above proof system. Furthermore, if* φ *contains dynamic modalities then* φ' *can be chosen such that* $\varphi' < \varphi$.

5 Conclusion and Comparison with Other Work

In this paper we proposed a dynamic logic which allows reasoning about inductive inference. Our Dynamic Logic of Learning Theory (DLLT) is an extension of previously studied Subset Space Logics, and a natural continuation of the work bridging Dynamic Epistemic Logic and Formal Learning Theory. Together with a syntax, featuring dynamic observation operators, and a topological semantics, we give a sound and complete axiomatization of this logic. We show how natural learnability properties, as learnability in the limit and learnability with certainty, can be expressed in DLLT.

Our technical results (the complete axiomatization and expressivity results), as well as the methods used to prove them (the canonical neighborhood model and the reduction laws), may look deceivingly simple. But in fact, achieving this simplicity is one of the major contributions of our paper! The most well-known relative to our logic is *Subset Space Logic (SSL) over intersection spaces*, completely axiomatized by Weiss and Parikh [26] (- and indeed our operator $\Box$ originates in the 'effort modality' of the SSL formalism introduced in [9,17]). Although less expressive than our logic (since it has no notion of belief B or conjecture L), the Weiss-Parikh axiomatization of SSL over intersection spaces is in a sense more complex and less transparent (as is their completeness proof, which is non-canonical). That axiomatization consists of the following list:

$S5_K$	The $S5$ axioms and rules for K
$S4_\Box$	The $S4$ axioms and rules for $\Box$
Cross Axiom	$K\Box\varphi \to \Box K\varphi$
Weak Directedness	$\Diamond\Box\varphi \to \Box\Diamond\varphi$
M_n (for all n)	$(\Box\langle K\rangle\varphi \wedge \Diamond K\psi_1 \wedge \ldots \wedge \Diamond K\psi_n) \to \langle K\rangle(\Diamond\varphi \wedge \Diamond K\psi_1 \wedge \ldots \wedge \Diamond K\psi_n)$

Though this list looks shorter than our list in Table 1, each of our axioms is simple and readable and has a transparent intuitive interpretation. In contrast, note the complexity and opaqueness of the last axiom schemata M_n above (having one schema for each natural number n)! Our completeness result implies that all these complex validities are provable in our simple system (and in fact in the even simpler system obtained by deleting from ours all the axioms that refer to the learner L). This shows the usefulness of adding the (expressively redundant) dynamic observation modalities: they help to describe the behavior of the effort modality $\Box$ in a much simpler and natural manner, via the combination of the Effort axiom and the Effort rule (which together capture the meaning of $\Box$ as universally quantifying over observation modalities).

Moreover, our completeness proof is also much simpler (though with some technical twists). Traditionally, the use of canonical models has been considered impossible for Subset Space Logics, and so authors had to use other, more ad-hoc methods (e.g. step-by-step constructions). The fact that in this paper we can get away with a canonical construction is again due to the addition of the dynamic modalities.

More recent papers, closely related to our logic, are Bjorndahl [8], van Ditmarsch et al. [20,21], and Baltag et al. [7]. Bjorndahl [8] introduces dynamic modalities $[\varphi]$ for arbitrary formulas (rather than restricting to observational variables $[o]$, as we do), though with a different semantics (according to which $[\varphi]$ restricts the space to the interior of φ, in contrast to our simpler semantics, that follows the standard definition of update or "public announcement"). His syntax does *not* contain the effort modality, or any other form of quantifying over observations. The work of van Ditmarsch et al. [20,21] uses Bjorndahl-style dynamic modalities in combination with a topological version of the so-called "arbitrary public announcement" operator, which is a more syntactic-driven relative of the effort modality. This syntactic nature comes with a price: the logic of arbitrary public announcements is much less well-behaved than SSL (or our logic), in particular it has non-compositional features (-the meaning of a formula may depend on the meaning of *all* atomic variables, including the ones that do not occur in that formula!). As a consequence, the soundness of (the arbitrary-public announcement analogue of) our Effort Rule is not at all obvious for their logic, which instead relies on an infinitary inference rule. Since that rule makes use of infinitely many premisses, their complete axiomatization is truly infinitary, and impossible to automatize: indeed, it does not even necessarily imply that the set of their validities is recursively enumerable (in contrast with our finitary axiomatization, which immediately implies such a result). The recent, unpublished work by Baltag et al. [7] (due to a subset of the current authors, using techniques similar to the ones we used in this paper) fixes these problems by replacing the arbitrary announcement modality with the effort modality (or equivalently, extending SSL with Bjorndahl-style dynamic modalities). But note that, in contrast to the work presented here, *all the above papers are concerned only with axiomatizations over topological spaces* (rather than the wider class of intersection spaces), and that *none of them has any belief B or conjecture operators L*. Hence, none of them can be used to capture any learning-theoretic notions going beyond finite identifiability.

Acknowledgements. The research of Nina Gierasimczuk is supported by an Innovational Research Incentives Scheme Veni grant 275-20-043, Netherlands Organisation for Scientific Research (NWO) and by the OPUS grant 2015/19/B/HS1/03292, National Science Centre Poland (NCN). Aybüke Özgün acknowledges financial support from European Research Council grant EPS 313360.

References

1. Aiello, M., Pratt-Hartmann, I., van Benthem, J. (eds.): Handbook of Spatial Logics, vol. 4. Springer, Dordrecht (2007). https://doi.org/10.1007/978-1-4020-5587-4
2. Balbiani, P., Baltag, A., van Ditmarsch, H., Herzig, A., Hoshi, T., Lima, T.D.: 'Knowable' as 'known after an announcement'. Rev. Symb. Logic **1**, 305–334 (2008)
3. Balbiani, P., van Ditmarsch, H., Kudinov, A.: Subset space logic with arbitrary announcements. In: Lodaya, K. (ed.) ICLA 2013. LNCS, vol. 7750, pp. 233–244. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36039-8_21
4. Baltag, A., Gierasimczuk, N., Smets, S.: Belief revision as a truth-tracking process. In: Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge, pp. 187–190. ACM (2011)

5. Baltag, A., Gierasimczuk, N., Smets, S.: On the solvability of inductive problems: a study in epistemic topology. In: Ramanujam, R. (ed.) Proceedings of the 15th Conference TARK, Also Available as a Technical Report in ILLC Prepublication Series PP-2015-13 (2015)
6. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In: Proceedings of the 7th Conference TARK, pp. 43–56. Morgan Kaufmann Publishers Inc. (1998)
7. Baltag, A., Özgün, A., Vargas Sandoval, A.L.: Topo-logic as a dynamic-epistemic logic. In: Baltag, A., Seligman, J., Yamada, T. (eds.) LORI 2017. LNCS, vol. 10455, pp. 330–346. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-55665-8_23
8. Bjorndahl, A.: Topological subset space models for public announcements. In: Trends in Logic, Outstanding Contributions: Jaakko Hintikka (2016, to appear)
9. Dabrowski, A., Moss, L.S., Parikh, R.: Topological reasoning and the logic of knowledge. Ann. Pure Appl. Logic **78**, 73–110 (1996)
10. Dégremont, C., Gierasimczuk, N.: Finite identification from the viewpoint of epistemic update. Inf. Comput. **209**, 383–396 (2011)
11. Georgatos, K.: Knowledge theoretic properties of topological spaces. In: Masuch, M., Pólos, L. (eds.) Logic at Work 1992. LNCS, vol. 808, pp. 147–159. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58095-6_11
12. Georgatos, K.: Knowledge on treelike spaces. Stud. Logica. **59**, 271–301 (1997)
13. Gierasimczuk, N.: Knowing one's limits. Logical analysis of inductive inference. Ph.D. thesis, Universiteit van Amsterdam, The Netherlands (2010)
14. Gierasimczuk, N., de Jongh, D.: On the complexity of conclusive update. Comput. J. **56**, 365–377 (2013)
15. Gold, E.M.: Language identification in the limit. Inf. Control **10**, 447–474 (1967)
16. Kelly, K.T.: The Logic of Reliable Inquiry. Oxford University Press, Oxford (1996)
17. Moss, L.S., Parikh, R.: Topological reasoning and the logic of knowledge. In: Proceedings of the 4th Conference TARK, pp. 95–105. Morgan Kaufmann (1992)
18. Parikh, R., Moss, L.S., Steinsvold, C.: Topology and epistemic logic. In: Aiello, M., Pratt-Hartmann, I., Van Benthem, J. (eds.) Handbook of Spatial Logics, pp. 299–341. Springer, Dordrecht (2007). https://doi.org/10.1007/978-1-4020-5587-4_6
19. van Benthem, J.: Logical Dynamics of Information and Interaction. Cambridge University Press, New York (2011)
20. van Ditmarsch, H., Knight, S., Özgün, A.: Arbitrary announcements on topological subset spaces. In: Bulling, N. (ed.) EUMAS 2014. LNCS (LNAI), vol. 8953, pp. 252–266. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-17130-2_17
21. van Ditmarsch, H., Knight, S., Özgün, A.: Announcement as effort on topological spaces. In: Proceedings of the 15th TARK, pp. 95–102 (2015)
22. van Ditmarsch, H., Knight, S., Özgün, A.: Announcement as effort on topological spaces. Extended version, Submitted (2015). http://www.lix.polytechnique.fr/sophia/papers/effort.pdf
23. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic, 1st edn. Springer, Dordrecht (2008). https://doi.org/10.1007/978-1-4020-5839-4
24. Vickers, S.: Topology via Logic. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge (1989)
25. Wang, Y.N., Ågotnes, T.: Multi-agent subset space logic. In: Proceedings of the 23rd IJCAI, pp. 1155–1161. IJCAI/AAAI (2013)
26. Weiss, M.A., Parikh, R.: Completeness of certain bimodal logics for subset spaces. Stud. Logica. **71**, 1–30 (2002)

Layered Logics, Coalgebraically

Luís Soares Barbosa(✉)

UNU-EGOV, United Nations University,
Campus de Couros, Guimarães, Portugal
barbosa@unu.edu

Abstract. This note revisits layered logics from a coalgebraic point of view, and proposes a naturality condition to express the typical hierarchical requirement under which all abstract transitions should be traceable in more specialised layers.

Keywords: Layered logics · Hierarchical models · Coalgebra

1 Introduction

A plethora of logics is used in Software Engineering to support the specification of systems' requirements and properties, as well as to verify whether, or to what extent, they are enforced in specific implementations. Broadly speaking, the logics of dynamical systems are *modal*, *i.e.* they provide operators which qualify formulas as holding in a certain *mode*. In mediaeval Scholastics such modes represented the strength of assertion (e.g. 'necessity' or 'possibility'). In temporal reasoning they can refer to a future or past instant, or a collection thereof. Similarly, one may express epistemic states (e.g. 'as everyone knows'), deontic obligations (e.g. 'when legally entitled'), or spatial states (e.g. 'in every point of a surface').

Regarding dynamical systems as transformations of state spaces according to specific transition shapes, *i.e.* as coalgebras for particular functors [11] such modes refer to particular configurations of successor states as defined, or induced, by the coalgebra dynamics. Coalgebra provides a *uniform* characterisation inducing 'canonical' notions of modality and the corresponding logic with respect to the underlying functor [5,6]. General questions in modal logic, such as the trade-off between expressiveness and computational tractability, or the relationship between logical equivalence and bisimilarity, can be addressed at this (appropriate) level of abstraction.

In this sense, modal logic is essentially coalgebraic [3]. Its classical extensions, for example hybrid logic, which is able to pinpoint specific states and index to them the satisfaction relation, can also be easily accommodated in the framework [9,12].

This short note revisits a logic suitable to express properties of, and reason about, n-layered, hierarchical transition systems, from a coalgebraic perspective,

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 55–63, 2018.
https://doi.org/10.1007/978-3-319-73579-5_4

building on previous results reported in references [7,8]. In particular it is shown how the *hierachical condition*, informaly stated under the *motto* 'upper transitions should be traceable in the layer below' can be expressed as a naturality condition in the models.

A very brief introduction do Coalgebra is made in the next session to highlight the paper's background. Hierarchical systems and a language to express them are discussed in Sect. 3. Finally, Sect. 4 introduces the paper's original idea on how layered logics can be framed coalgebraically.

2 Coalgebra

Often referred to as *the mathematics of dynamical, state-based systems*, Coalgebra claims to provide a compositional and uniform framework to specify, analyse and reason about state and behaviour in computing. A flavour of the basic definition is given in the sequel.

To define an (inductive) data structure, as typically taught in an undergraduate course on programming, one essentially specifies its 'assembly process'. For example, one builds a sequence in a data domain D, either by taking an empty list or by adjoining a fresh element to an existing sequence. Thus, declaring a sequence data type yields a function $\zeta : \mathbf{1} + D \times U \longrightarrow U$, where U stands for the data type being defined. The structured domain of function ζ captures a signature of *constructors* ($nil : \mathbf{1} \longrightarrow U$, $cons : D \times U \longrightarrow U$), composed additively (*i.e.* $\zeta = [nil, cons]$). The whole procedure resembles the way in which an algebraic structure is defined.

Reversing an 'assembly process' swaps structure from the domain to the codomain of the arrow, which now captures the result of a 'decomposition' or 'observation' process. In the example at hand this is performed by the familiar *head* and *tail selectors* joined together into

$$\alpha : U \longrightarrow \mathbf{1} + D \times U \tag{1}$$

where α either returns a token $*$ (representing the unique element of singleton set $\mathbf{1}$), when observing an empty sequence, or its decomposition in the top element and the remaining tail.

This reversal of perspective also leads to a different understanding of what U may stand for. The product $D \times U$ captures the fact that both the head and the tail of a sequence are selected (or *observed*) simultaneously. In fact, once one is no longer focused on how to construct U, but simply on what can be observed of it, finiteness is no longer required: both finite or infinite sequences can be observed through the process above. Therefore, U can be more accurately thought of as a *state space* of a machine generating a finite or infinite sequence of values of type D. Elements of U, in this example, can no longer be distinguished by construction, but should rather be identified when generating the same sequence. That is to say, when it becomes impossible to distinguish them through the observations allowed by the 'shape' structuring the codomain of α.

Function (1) is an example of a *coalgebra* living in the category *Set* of sets and functions.

A category, the reader is recalled, is simply a universe of typed arrows bearing the structure of a partial monoid: arrows are associatively composed whenever domain and codomain types match, and every type exhibits an arrow which is the identity for composition. Categories themselves can be thought as types in broader categories whose arrows, called *functors*, are morphisms preserving identities and composition. Again functors can be taken as types in yet broader categories whose arrows, traditionally called *natural transformations*, convey a notion which nicely captures parametricity in datatype theory, and is central to the construction proposed in this paper.

Look again at coalgebra α defined in (1). Its ingredients are: a carrier U (intuitively, the state space of a machine), the *shape* of allowed observations, technically a functor $\mathcal{F}(X) = \mathbf{1} + D \times X$, and the observation *dynamics* given by function α, *i.e.* the machine itself. Formally, a $\mathcal{F}$-coalgebra is a pair $\langle U, \alpha \rangle$ consisting of an object U and a map $\alpha : U \longrightarrow \mathcal{F}\ U$. The latter maps states to structured collections of successor states. What shapes the underlying transition system, therefore structuring the set of successor states, is encoded in functor $\mathcal{F}$. In (1), $\mathcal{F}(X) = \mathbf{1} + D \times X$ entails a deterministic, partial transition system, whereas, for example, the power set construction, as in $\mathcal{F}(X) = \mathcal{P}(D \times X)$ introduces non determinism. Actually, by varying $\mathcal{F}$, one may capture a large class of semantic structures used to model computational phenomena as (more or less complex) transition systems. Going even further, $\mathcal{F}$ is not restricted to be an endofunctor in the category of sets. For example, as recently discussed in [10], the category of topological spaces emerges as the natural host for coalgebras modelling continuous systems. The study of the common properties of all these systems is the subject of *Universal Coalgebra*, as developed systematically by a number of authors from the pioneering work of Rutten [11].

For each functor $\mathcal{F}$ over a category C, $\mathcal{F}$-coalgebras are types in a corresponding category $C_{\mathcal{F}}$ where both composition and identities are inherited from C. An arrow there, between two $\mathcal{F}$-coalgebras, $\langle U, \alpha \rangle$ and $\langle V, \beta \rangle$, is a map h between carriers U and V which preserves the dynamics, *i.e.* such that $\beta \cdot h = \mathcal{F}\ h \cdot \alpha$.

This sets Coalgebra as a suitable mathematical framework for the study of dynamical systems in both a *compositional* and *uniform* way. The qualifier *uniform* requires some extra explanation: coalgebraic concepts (*i.e.* models, constructions, logics, and proof principles) are parametric on, or *typed* by, the functor that characterises the underlying transition structure. Actually, the essence of the coalgebraic method boils down to a very basic observation: that from a suitable characterisation of the *type* of a system's dynamics, encoded in a functor $\mathcal{F}$, canonical notions of behaviour and observational reasoning (equational and inequational) can be derived in a uniform (*i.e.* parametric) way. In Mathematics, as in Software Engineering, going parametric allows us to focus on the abstract structure of a problem such that, on solving it, what we actually solve is a whole class of problems.

In software design one is often interested in properties that are preserved along the system's evolution, the so-called 'business rules', as well as in 'future warranties', stating that *e.g.* some desirable outcome will be eventually produced. As mentioned in the introductory section, both classes are examples of *modal* assertions, *i.e.* properties that are to be interpreted across a transition system capturing the software dynamics. Again in Coalgebra also modalities acquire a *shape*. That is, their definitions become parametric on whatever type of behaviour seems appropriate for addressing the problem at hand. The following sections explore such a potential in designing a family of logics to reason about hierarchical design.

3 Reasoning About Hierarchical Designs

Hierarchical transition systems are a popular mathematical structure to represent state-based software applications in which different layers of abstraction are captured by interrelated state machines. The decomposition of high-level states into inner sub-states, and of their transitions into inner sub-transitions, is a common refinement procedure adopted in a number of specification formalisms.

In a recent paper [7] the author and his collaborators proposed an hybrid layered logic to reason about (non deterministic) transition systems. The diagram in Fig. 1, representing a partial view of a strongbox controller, is taken from that paper as an illustration of the sort of examples we have in mind.

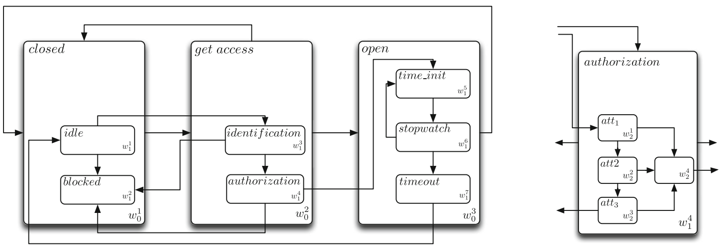


Fig. 1. An hierarchical transition system.

The strongbox controller is specified at three different levels of abstraction, expressing the progressive decomposition not only of its internal states, but also of its transitions. Thus, each 'high-level' state gives rise to a new, local transition system, and each 'upper-level' transition is decomposed into a number of 'intrusive' transitions from sub-states of the 'lower-level' transition system corresponding to the refinement of the original source state, to sub-states of the corresponding refinements of original target states. For instance, the (upper) *close* state can be refined into a (inner) transition system with two (sub) states: one, *idle*, representing the system waiting for the order to proceed for the *get access* state, and another one, *blocked*, capturing a system which is unable to proceed with the opening process (e.g. when authorised access for a given user was definitively denied). In this scenario, the upper level transition from *closed* to *get access* can be realised by, at least, one intrusive transition between the *closed* sub-state *idle* and the *getaccess* sub-state $identification$, in which the user identification is to be checked before proceeding. This refinement is illustrated in the left part of Fig. 1.

The logic proposed in [7] to reason about this sort of systems is modal (so that state transitions can be expressed), combined with hybrid features to refer to specific, individual states. The qualifier *hybrid* [1,2] refers to an extension of modal languages with symbols, called *nominals*, which explicitly refer to individual states in the underlying Kripke structure. A satisfaction operator $@_i\varphi$ stands for φ holding in the state named by nominal i.

Signatures are n-families of disjoint, possible empty, sets of symbols

$$\Delta^n = \big(\mathrm{Prop}_k, \mathrm{Nom}_k\big)_{k\in\{0,\cdots,n\}}\,.$$

For example, to specify the strongbox above, one considers a signature Δ^2 for the three layers presented, numbered from 0 (the most abstract) to 2.

The set of formulas $Fm(\Delta^n)$ is the n-family recursively defined, for each k, by

$$\varphi_0 \ni i_0 \mid p_0 \mid \neg\varphi_0 \mid \varphi_0 \wedge \varphi_0 \mid @_{i_0}\varphi_0 \mid \Box_0\varphi_0$$

$$\varphi_0^b \ni i_0 \mid p_0 \mid @_{i_0}\varphi_0 \mid \Box_0\varphi_0$$

and

$$\varphi_k \ni \varphi_{k-1}^b \mid i_k \mid p_k \mid \neg\varphi_k \mid \varphi_k \wedge \varphi_k \mid @_{i_k}\varphi_k \mid \Box_k\varphi_k$$

where for any $k \in \{1,\ldots,n\}$, the basic formulas are defined by

$$\varphi_{k-1}^b \ni i_{k-1} \mid p_{k-1} \mid \varphi_{k-2}^b \mid @_{i_{k-1}}\varphi_{k-1} \mid \Box_{k-1}\varphi_{k-1}$$

for $k \in \{2,\cdots,n\}$, $p_k \in \mathrm{Prop}_k$ and $i_k \in \mathrm{Nom}_k$.

This language is able to express properties of very different natures. For instance, one may express inner-outer relations between named states (e.g. $@_{idle_1}closed_0$ or $@_{att_{12}}open_0$) as well as a variety of transitions. Those include, for example, the layered transition $@_{get\_access_0}\,\Diamond_0\; open_0$, a 0-internal one $@_{identification_1}\,\Diamond_1\; authorisation_1$, and intrusive transitions such as $@_{idle_1}\,\Diamond_1$ $authorisation_1$ and $get\_access_0 \to \Diamond_1 open_0$.

4 ... Coalgebraically

The whole programme can actually be carried out in a coalgebraic setting. The basic observation is that when defining a model for this logic the family of accessibility relations considered in [7] is replaced by a family of coalgebras for the same endofunctor, each of which captures the dynamics of the appropriate layer.

Thus, a n-layered model $M \in \text{Mod}^n(\Delta^n)$ is a tuple

$$M = \langle W^n, D^n, \alpha^n, V^n \rangle$$

where $W^n = (W_k)_{k\in\{0,\cdots,n\}}$ is a family of disjoint sets of states, and $D^n \subseteq W_0 \times \cdots \times W_n$ is a definition predicate that singles out the chains of states across the n levels which are considered meaningful 'global' states. Denoting by D_k the k-restriction $D^n|_k$ to the first $k+1$ columns, for each $k \in \{0, \cdots, n\}$, it is the case that

$$W_k = \{v_k | D_k\langle w_0, \cdots, w_{k-1}, v_k\rangle \ ,$$

for some $w_0, \cdots, w_{k-1}$, such that $D_{k-1}\langle w_0, \cdots, w_{k-1}\rangle\}$. The 'dynamics':

$$\alpha^n = \big(\alpha_k : D_k \longrightarrow \mathcal{F}(D_k)\big)_{k\in\{0,\cdots,n\}}$$

is a family of $\mathcal{F}$-coalgebras specifying the system's evolution at each level in the hierarchy. Finally, $V^n = (V_k^{\text{Prop}}, V_k^{\text{Nom}})_{k\in\{0,\cdots,n\}}$ is a family of pairs of valuations defined as one would expect: $V_k^{\text{Prop}} : \text{Prop}_k \to \mathcal{P}(D_k)$, and $V_k^{\text{Nom}} : \text{Nom}_k \to W_k$.

The advantage of expressing the transition structure coalgebraically is the genericity of the approach. Actually, making $\mathcal{F} = \mathcal{P}$, the powerset monad, we are brought back the usual Kripke structure, modal formulas being interpreted over a non deterministic transition system. Different alternatives can be considered by varying $\mathcal{F}$. For example, modalities can be interpreted in a probabilistic setting by instantiating $\mathcal{F}$ with the sub-distribution monad $\mathcal{D}_{\leq}(X) = \{\mu : X \to \mathbb{R}_{\geq 0} | \sum_{x\in X} \mu\, x \leq 1\}$ which captures probabilistic transitions — note that what is missing to 1 above can be seen as the probability of some sort of 'systemic' failure, such as deadlock, to occur. As expected, the satisfaction relation is a family $\models^n = (\models_k)_{k\in\{0,\cdots,n\}}$ defined, for each $w_r \in W^r$, $r \in \{0, \cdots, k\}$, $k \leq n$, such that $D_k\langle w_0, \cdots w_k\rangle$. The case of interest in the context of this note is the one for modalities, *i.e.* $M_k, w_0, \cdots, w_k \models_k \Box_k \varphi_k$ iff

$$\forall_{v_0\in W_0,\cdots,v_k\in W_k} \cdot \ \langle v_0, \cdots, v_k\rangle \in \alpha_k\langle w_0, \cdots, w_k\rangle \ \text{ implies } \ M, v_0, \cdots, v_k \models_k \varphi_k \ .$$

The hybrid part is given by

- $M_k, w_0, \cdots, w_k \models_k i_k$ iff $w_k = V_k^{\text{Nom}}(i_k)$ and $D_k\langle w_0, \cdots, w_{k-1}, V_k^{\text{Nom}}(i_k)\rangle$,
- $M_k, w_0, \cdots, w_k \models_k @_{i_k}\varphi_k$ iff $M_k, w_0, \cdots w_{k-1}, V_k^{\text{Nom}}(i_k) \models_k \varphi_k$ and $D_k\langle w_0, \cdots w_{k-1}, V_k^{\text{Nom}}(i_k)\rangle$.

The Boolean part, finally, is defined as usual, just taking care of the definability interdependence captured by D^n. The only aspect one needs to take into account is the interplay between the satisfaction operators and the modalities induced (or built over) the coalgebra. For example, one has to specify that a formula like $@_i\ \varphi$ must be valid either in the whole model or nowhere. In an Hilbert calculus this can be achieved through an extra axiom, for each modal operator $\circledast$:

$$@_i\varphi \ \Rightarrow \ (\circledast(\varphi_1, \cdots, \varphi_k) \ \Leftrightarrow \ \circledast(\varphi_1 \wedge @_i\varphi, \cdots, \varphi_k \wedge @_i\varphi)$$

capturing the intended validity of $@_i\varphi$ irrespective to the interpretation of each φ_j.

As mentioned in the Introduction, there is a specific, particularly well-behaved class of layered models, called *hierarchical*, in which all upper transitions are traceable in the layer below. Technically, this amounts to the requirement that the restriction of a coalgebra α_k to the state space of α_{k-1} coincides with the latter. In this case, the family of coalgebras α^n is called *hierarchically compatible.*

The example sketched in Fig. 1, is clearly an hierarchical model. Examples of non-hierarchical layered models can be achieved by removing some 0-transitions depicted in the diagram above (e.g. the one linking the named states $closed_0$ and get_access_0). The hierarchical condition is quite natural and somehow inherent to well-known design formalisms such as Harel's statecharts [4] and the subsequent UML hierarchical state machines, among others.

What is worth to notice is that the *hierarchical* requirement can be expressed as a naturality condition as follows. The first step is to regard the family of coalgebras α^n as a coalgebra in a *functor category*, for a suitable finite chain:

$$\begin{array}{ccc} \bullet & \rightarrow & Set \\ \uparrow & & \\ \bullet & & \\ \uparrow & & \\ \bullet & & \end{array}$$

Arrows (thus, component coalgebras) are families of natural transformations, making the following diagram to commute for all k,

$$\begin{array}{ccc} D_k & \xrightarrow{\alpha_k} & \mathcal{F}(D_k) \\ \downarrow{\scriptstyle \pi_k} & & \downarrow{\scriptstyle \mathcal{F}(\pi_k)} \\ D_{k-1} & \xrightarrow{\alpha_{k-1}} & \mathcal{F}(D_{k-1}) \end{array}$$

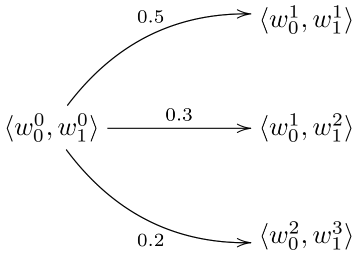
where $\pi_k : D_k \longrightarrow D_{k-1}$ be given by $\pi_k\ \langle w_0, \cdots, w_{k-1}, w_k\rangle \mathrel{\widehat{=}} \langle w_0, \cdots, w_{k-1}\rangle$.

Let us illustrate this construction. For $\mathcal{F} = \mathcal{P}$, transitions

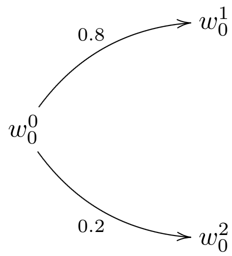
$$\begin{array}{ccc} & & \langle w_0^1, w_1^1\rangle \\ & \nearrow & \\ \langle w_0^0, w_1^0\rangle & & \\ & \searrow & \\ & & \langle w_0^1, w_1^2\rangle \end{array}$$

exist at level 1 iff a transition $w_0^0 \longrightarrow w_0^1$ exists at level 0.

For another example, consider $\mathcal{F} = \mathcal{D}_{\leq}$. In an hierarchical (probabilistic) system naturality entails that the existence of transitions



requires



at level 0.

5 Concluding

As it happens in other domains of Computer Science, also at this level of 'logics engineering' Coalgebra is a source of genericity. Clearly, whenever functor $\mathcal{F}$ is a monad, the monadic structure can be taken as the basis for a model algebra in which the composition of hierarchical systems can be addressed. Canonical formats for bisimulation come for free, and often Hennessy-Milner like theorems can be proved in a generic setting.

On the other hand, the *motto* for hierarchical models – 'upper transitions should be traceable in the layer below' – can be re-phrased, in a more precise way, as *layer compatibility is natural.* This, again, is a source of genericity: the

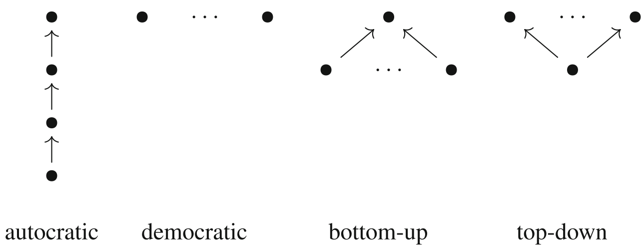


Fig. 2. Some shapes for layered structures.

very notion of *hierarchical* is made relative to whatever poset P is used to define the *layer structure* intended to be respected. We leave the reader of this short note with the (easy) quiz of re-phrasing this notion for the four different posets depicted in Fig. 2.

Acknowledgments. This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation through (a) COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, project `POCI-01-0145-FEDER-016826`, and (b) Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, within project `NORTE-01-0145-FEDER-000037`.

References

1. Blackburn, P.: Representation, reasoning, and relational structures: a hybrid logic manifesto. Logic J. IGPL **8**(3), 339–365 (2000)
2. Brauner, T.: Hybrid Logic and its Proof-Theory. Applied Logic Series. Springer, Dordrecht (2010). https://doi.org/10.1007/978-94-007-0002-4
3. Cîrstea, C., Kurz, A., Pattinson, D., Schröder, L., Venema, Y.: Modal logics are coalgebraic. Comput. J. **54**(1), 31–41 (2011)
4. Harel, D.: Statecharts: a visual formalism for complex systems. Sci. Comput. Program. **8**(3), 231–274 (1987)
5. Kupke, C., Pattinson, D.: Coalgebraic semantics of modal logics: an overview. Theor. Comput. Sci. **412**(38), 5070–5094 (2011)
6. Kurz, A., Leal, R.L.: Modalities in the stone age: a comparison of coalgebraic logics. Theor. Comput. Sci. **430**, 88–116 (2012)
7. Madeira, A., Martins, M.A., Barbosa, L.S.: A logic for n-dimensional hierarchical refinement. In: Derrick, J., Boiten, E.A., Reeves, S. (eds.) Proceedings 17th International Workshop on Refinement, Refine@FM 2015, Oslo, Norway, 22nd June 2015, EPTCS, vol. 209, pp. 40–56 (2016)
8. Madeira, A., Martins, M.A., Barbosa, L.S., Hennicker, R.: Refinement in hybridised institutions. Formal Aspects Comput. **27**(2), 1–21 (2014)
9. Myers, R., Pattinson, D., Schröder, L.: Coalgebraic hybrid logic. In: de Alfaro, L. (ed.) FoSSaCS 2009. LNCS, vol. 5504, pp. 137–151. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00596-1_11
10. Neves, R., Barbosa, L.S., Hofmann, D., Martins, M.A.: Continuity as a computational effect. J. Log. Algebr. Meth. Program. **85**(5), 1057–1085 (2016)
11. Rutten, J.J.M.M.: Universal coalgebra: a theory of systems. Theor. Comput. Sci **249**(1), 3–80 (2000). Revised version of CWI Technical report CS-R9652 (1996)
12. Schröder, L., Pattinson, D.: Named models in coalgebraic hybrid logic. In: Marion, J.-Y., Schwentick, T. (eds.) 27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, Leibniz International Proceedings in Informatics, vol. 5, pp. 645–656. Schloss Dagstuhl - Leibniz-Center of Informatics, Dagstuhl (2010)

A Dynamic Informational-Epistemic Logic

Yuri David Santos(✉)

University of Groningen, Groningen, The Netherlands
y.david.santos@rug.nl

Abstract. Epistemic logic is usually employed to model two aspects of a situation: the ontic and the epistemic aspects. Truth, however, is not always attainable, and in many cases we are forced to reason only with whatever information is available to us. In this paper, we will explore a four-valued epistemic logic designed to deal with situations of this sort. The technical results include a set of reduction axioms for public announcements, correspondence proofs, and a complete tableau system.

Keywords: Many-valued logics · Epistemic logic
Paraconsistent logics · Public announcements
Multi-agent systems

1 Introduction

Is drinking a glass of red wine per day good for your heart? The answer may be yes or no depending on where you look. We do not dare to try to answer this question here, but we want to offer a logical formalism that can help us understand situations like this, where the available information about a certain topic can be conflicting or incomplete. In most practical settings, obtaining the ultimate truth about anything is out of the question, and one has to deal with whatever information is available to them, which might be incomplete and sometimes even contradictory.

We will carry out our analysis on a four-valued epistemic logic, a variant of the so-called **BK** logic [13], a Belnapian version of modal logic **K**. As an important component of modern dynamic epistemic logics, public announcements will also be examined. In this way, we also intend to contribute to the motivation for the use of many-valued modal logics. As remarked by Fitting in the conclusion of [6], very little has been said about intuitions underlying many-valued modal logics, a situation which seems to persist in the current literature.

In four-valued logics, a proposition p can be, besides true or false, *both* (true and false) or *neither* (true nor false), denoted in this paper by the valuations $V(p) = \{0, 1\}$ and $V(p) = \emptyset$, respectively. One can, as was done by Belnap in his influential paper [2], interpret these truth-values as the status of information possibly coming from several sources. For example, if *both* is the value assigned to p, then this means that some source points to the truth and another to the falsity of p. The value *none* could mean that no information is available about p.

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 64–81, 2018.
https://doi.org/10.1007/978-3-319-73579-5_5

In this way, the valuation already represents the epistemic level, instead of the ontic level. This was not a problem since Belnap was not dealing with a modal logic. Now, the addition of a modal operator of belief to this logic will create two separate epistemic "layers".

Look at the classical epistemic model of Fig. 1(left). It represents a situation wherein an agent cannot distinguish between the truth and falsity of proposition p, or, equivalently, wherein the agent does not know whether p.

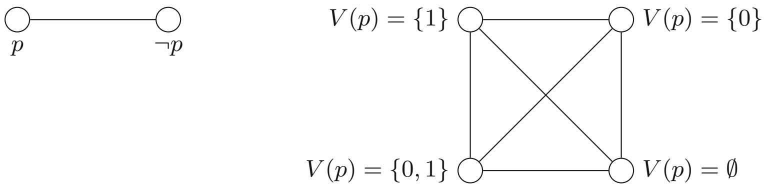


Fig. 1. An epistemic model (left) and a four-valued epistemic model (right).

Now, compare this situation with the four-valued model of Fig. 1(right). What is a plausible interpretation for this model? Here, the agent not only cannot distinguish between worlds where p is *true* or *false*, but also between worlds where it is *neither* true nor false, or *both*. If we adopt an epistemic interpretation of the valuations, what kind of interpretation is left for the operator $\Box$?

As mentioned before, we should think of two layers: the first concerning information, and the second concerning knowledge. The (four-valued) valuation function embodies the informational layer, while the accessibility relations account for the (multi-agent) epistemic layer. For example, we can regard the valuation as representing the information about some propositions stored in a database. The database only registers the information it receives, so it is well possible that at first it receives the information that p is *true*, but subsequently it receives (possibly from another source) the information that p is *false*. In this case the database contains contradictory information about p. The second level (the epistemic/doxastic level), represented by the accessibility relations, may be illustrated, for instance, by the knowledge of a user of this database. The user may be in a state like the one in Fig. 1(right), where she considers it possible that the database is in any of the four possible states regarding p.

Notice that our agents do not possess real knowledge (knowledge about facts), but only a superficial knowledge about information itself – whence we say that one layer concerns *information*, as opposed to reality, and the other concerns *knowledge* about the "informational layer".

This interpretation also makes clear the difference between *none* and *both*, which could otherwise be equally understood as no information or useless information. If the database has *both* as the value of p, deleting information that supports the truth of p would result in p being just *false*, whereas if p was *none* this would have no effect. Similarly, receiving information when p is *none* can lead to a consistent state, but, if p is *both*, receiving new information has no qualitative effect. Such changes in information could be modelled through dynamic operators, but in this paper we only use public announcements (Sect. 5), and with a different purpose.

Another example not involving databases can be given. Let us consider a typical epistemic logic scenario. Anne lives in Groningen, so she knows whether *It is raining in Groningen* ($\Box_a g \vee \Box_a \neg g$). Likewise, Bart lives in Rotterdam and knows whether *It is raining in Rotterdam* ($\Box_b r \vee \Box_b \neg r$). The traditional epistemic model for this situation is depicted in Fig. 2.

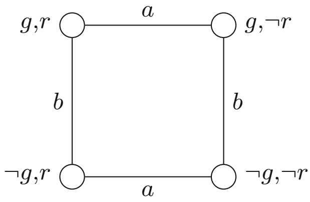


Fig. 2. A classic epistemic model.

Now suppose both of them usually inform themselves of the weather by watching the local television's newscast, and g and r mean that *It will rain in Groningen tonight* and *It will rain in Rotterdam tonight*, respectively. This changes nothing in the model of Fig. 2. However, imagine the situation in which Anne heard that g in the newscast of Channel 1, but $\neg g$ in the newscast of Channel 2. The status of g for Anne is now contradictory, which is denoted by the truth value *both*. Moreover, assuming that Anne is always up to date with the weather news from Channels 1 and 2, she will always be aware of the four-valued status of g. In this example, the sources of information, namely the television channels, play the role of the database. We are not endorsing the position that proposition g can actually be true and false at the same time, but only that there may be different pieces of information available, one supporting the truth and the other the falsity of g.

In this way, the logic preserves the standard meaning of the accessibility relations, namely that of epistemic alternatives (or uncertainty). So, in a state where g was announced to be both true and false, Anne is aware of that. She does not consider a world to be possible where only $\neg g$ was announced, for she already knows this is not the case. Bart, on the other hand, does not have access to Groningen weather in his local newscast, so he considers all of the four values to be possible for g. Now we can have a formula like $\Box_a(g \wedge \neg g)$, meaning that *Anne knows that there is information supporting both the truth and the falsity of* g.

The rest of this paper will explore in detail this logic with two epistemic layers, which we will simply call *four-valued epistemic logic* (FVEL, in short). The remaining content is organized as follows. In Sect. 2 we define the syntax and semantics of the logic, and present some of its basic properties. In Sect. 3 we present a sound and complete tableau system. In Sect. 4 we show some correspondence results concerning classical epistemic logic axioms. In Sect. 5 we add public announcements to FVEL and show that they do not increase expressivity. We also extend the tableau system with rules for public announcements, and prove completeness. In Sect. 6 we give an illustrative example of FVEL in action. To wrap up, we comment on related work in Sect. 7 and conclude with Sect. 8. Some of the proofs can be found in the appendix[1].

2 Four-Valued Epistemic Logic

In this section, we will define the syntax and the semantics of the logical language being examined.

2.1 Syntax

Let P be a countable set of atomic propositions and A a finite set of agents. A well-formed formula φ in our language $\mathcal{L}$ is inductively defined as follows:

$$\varphi ::= p \mid {\sim}\varphi \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \Box_i\varphi$$

with $p \in P$ and $i \in A$. The following abbreviations will be employed throughout the text: $(\varphi \vee \psi) \overset{\text{def}}{=} \neg(\neg\varphi \wedge \neg\psi)$; $(\varphi \rightarrow \psi) \overset{\text{def}}{=} (\neg\varphi \vee \psi)$; $(\varphi \leftrightarrow \psi) \overset{\text{def}}{=} ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$; $\Diamond_i\varphi \overset{\text{def}}{=} \neg\Box_i\neg\varphi$. Parentheses will be left out when there is no room for ambiguity.

2.2 Semantics

Given the non-empty finite set $A = \{1, 2, ..., n\}$ of agents, an interpretation is a tuple $M = \langle S, R, V\rangle$, where S is a non-empty set of states, $R = \langle R_1, R_2, ..., R_n\rangle$ is an n-tuple of binary relations on S and $V : P \times S \rightarrow 2^{\{0,1\}}$ is a valuation function that assigns to each proposition one of four truth values ($\{0\}$ is *false*, $\{1\}$ is *true*, $\{\}$ is *none* and $\{0, 1\}$ is *both*). Although the results in this paper do not depend on the accessibility relations being equivalence relations, Sect. 4 presents some results that illustrate the effects of restricting R. With $p \in P$, $s \in S$, $i \in A$ and $\varphi, \psi \in \mathcal{L}$, the satisfaction relation $\models$ is inductively defined as follows:

$M, s \models p$	iff $1 \in V(p, s)$
$M, s \models \neg p$	iff $0 \in V(p, s)$
$M, s \models (\varphi \wedge \psi)$	iff $M, s \models \varphi$ and $M, s \models \psi$
$M, s \models \neg(\varphi \wedge \psi)$	iff $M, s \models \neg\varphi$ or $M, s \models \neg\psi$
$M, s \models \Box_i\varphi$	iff $\forall t \in S$ s.t. sR_it, it holds that $M, t \models \varphi$
$M, s \models \neg\Box_i\varphi$	iff $\exists t \in S$ such that sR_it and $M, t \models \neg\varphi$
$M, s \models {\sim}\varphi$	iff $M, s \not\models \varphi$
$M, s \models \neg{\sim}\varphi$	iff $M, s \models \varphi$
$M, s \models \neg\neg\varphi$	iff $M, s \models \varphi$

[1] Some proofs have been omitted due to space limitations, but are available at https://www.ime.usp.br/~yurids/appendix-dali17.pdf.

Now, we can talk not only about 4-valued atoms but also about 4-valued formulas in general. We define the *extended valuation function* $\overline{V} : \mathcal{L} \times S \to 2^{\{0,1\}}$ as follows:

$$1 \in \overline{V}(\varphi, s) \text{ iff } M, s \models \varphi$$
$$0 \in \overline{V}(\varphi, s) \text{ iff } M, s \models \neg\varphi$$

Using the above definition, we say that a formula φ has value *both* at s, for example, if and only if $\overline{V}(\varphi, s) = \{0, 1\}$, which is the case when both $M, s \models \varphi$ and $M, s \models \neg\varphi$. Truth and falsity of formulas are evaluated independently, and for that reason we define semantic conditions for each negated formula separately. Even though the semantics of $\neg$ as defined above is non-compositional[2], the connective is still truth-functional, as we will see in the next section.

2.3 Basic Properties

Now we build the truth tables for the truth-functional connectives according to the truth definitions given above (compare truth Tables 1, 2, 3, 4 and 5 below to the ones in [16, p. 146]). *True*, *false*, *none* and *both* are abbreviated to t, f, n and b, respectively.

Table 1. $\neg\varphi$.

φ	n	f	t	b
	n	t	f	b

Table 2. $\sim\varphi$.

φ	n	f	t	b
	t	t	f	f

Table 3. $\varphi \wedge \psi$.

$\varphi \backslash \psi$	n	f	t	b
n	n	f	n	f
f	f	f	f	f
t	n	f	t	b
b	f	f	b	b

Table 4. $\varphi \vee \psi$.

$\varphi \backslash \psi$	n	f	t	b
n	n	n	t	t
f	n	f	t	b
t	t	t	t	t
b	t	b	t	b

Example for Table 1 ($\neg \boldsymbol{b} = \boldsymbol{b}$): $\overline{V}(\varphi, s) = \{0, 1\}$ iff $0 \in \overline{V}(\varphi, s)$ and $1 \in \overline{V}(\varphi, s)$ iff $M, s \models \neg\varphi$ and $M, s \models \varphi$ iff $M, s \models \neg\varphi$ and $M, s \models \neg\neg\varphi$ iff $1 \in \overline{V}(\neg\varphi, s)$ and $0 \in \overline{V}(\neg\varphi, s)$ iff $\overline{V}(\neg\varphi, s) = \{0, 1\}$.

Example for Table 4 ($\boldsymbol{n} \vee \boldsymbol{b} = \boldsymbol{t}$): Recall that disjunction is defined in terms of conjunction and negation. $M, s \models \neg(\neg\varphi \wedge \neg\psi)$ iff $M, s \models \neg\neg\varphi$ or $M, s \models \neg\neg\psi$ iff $M, s \models \varphi$ or $M, s \models \psi$ iff $1 \in \overline{V}(\varphi, s)$ or $1 \in \overline{V}(\psi, s)$, which is true, for $\overline{V}(\psi, s) = \{0, 1\}$. $M, s \models \neg\neg(\neg\varphi \wedge \neg\psi)$ iff $M, s \models \neg\varphi \wedge \neg\psi$ iff $M, s \models \neg\varphi$ and $M, s \models \neg\psi$ iff $0 \in \overline{V}(\varphi, s)$ and $0 \in \overline{V}(\psi, s)$, which is false, for $\overline{V}(\varphi, s) = \emptyset$. Therefore $M, s \models \varphi \vee \psi$ holds, but $M, s \models \neg(\varphi \vee \psi)$ does not, thus $1 \in \overline{V}(\varphi \vee \psi)$ and $0 \notin \overline{V}(\varphi \vee \psi)$, hence $\overline{V}(\varphi \vee \psi) = \{1\}$.

[2] The semantics would be compositional if we used two support relations $\models^+$ and $\models^-$, as was done in [13]. While these two formalisms have the same expressivity, ours has a larger number of formulas (see more on this comparison in Sect. 7).

Table 5. $\varphi \to \psi$.

$\varphi \backslash \psi$	n	f	t	b
n	n	n	t	t
f	t	t	t	t
t	n	f	t	b
b	t	b	t	b

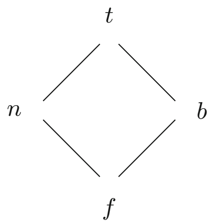


Fig. 3. Lattice L4.

If we leave $\neg$ out, we have classical modal logic, with $\{1\}$ and $\{0,1\}$ (to which [16] refers as *designated values*) behaving as *true*, and $\emptyset$ and $\{0\}$ (accordingly, *non-designated values*) behaving as *false*.

Moreover, observing these truth tables, we notice that the fragment resulting from leaving $\sim$ and $\Box$ out behaves exactly as *first degree entailment* (**FDE**) [5,16]. Conjunction and disjunction are given by the meet and join, respectively, of the values in the lattice depicted in Fig. 3, called $L4$ in [2]. Now, adding the modal operator to **FDE** we obtain $\mathbf{K_{FDE}}$, a logic which Priest has studied [16]. He provides a complete tableau system for this logic. Moreover, he shows that this logic contains no validities, as is the case for **FDE** itself.

We can also build the truth tables for the connectives $\vee$ and $\to$ defined over $\sim$ instead of $\neg$ (which we will denote by $\tilde{\vee}$ and $\tilde{\to}$, respectively). Despite these connectives being binary functions accepting two four-valued parameters, they behave analogously to their classical (Boolean) counterparts. They can be viewed as a composition of a function that compresses designated values into *true* and non-designated values into *false* (just like the $\sim$ operator itself) with the corresponding Boolean function. In other words, if *or* is classical disjunction and *imp* is classical implication, $x \tilde{\vee} y = or({\sim}{\sim}x, {\sim}{\sim}y)$ and $x \tilde{\to} y = imp({\sim}{\sim}x, {\sim}{\sim}y)$. It is also relevant to remark that when the operands take on only classical values, both pairs of operators ($\vee$,$\to$ and $\tilde{\vee}$,$\tilde{\to}$) behave exactly alike.

Validity. We say that $M \models \varphi$ if and only if $M, s \models \varphi$ for all $s \in S$, where $M = \langle S, R, V \rangle$. A formula φ is valid ($\models \varphi$) if and only if $M \models \varphi$ for all models M. A frame is a pair $\mathcal{F} = \langle S, R \rangle$. We say a formula φ is valid in a frame $\mathcal{F} = \langle S, R \rangle$, that is, $\mathcal{F} \models \varphi$, if and only if, for all valuations V, it holds that $M \models \varphi$, where $M = \langle S, R, V \rangle$ (and we say M is based on frame $\mathcal{F}$). If for all models M and all states s it is the case that $M, s \models \Sigma$ implies $M, s \models \varphi$, we say that $\Sigma \models \varphi$ (φ is a logical consequence of Σ).

We can define $\top$, a validity, as $\top \stackrel{\text{def}}{=} (p \vee {\sim}p)$. While **FDE** has no validities, FVEL has an infinity of them, including $\top$. Moreover, all propositional tautologies (built with $\sim$) are still validities in FVEL, as expected, but there are other valid formulas with both $\sim$ and $\neg$, such as ${\sim}p \vee \neg{\sim}p$.

Equivalence. Logical equivalence (sameness in truth value) cannot be expressed by $\varphi \leftrightarrow \psi$ in FVEL. Look at Table 6. The diagonal should be *designated*, and the rest *non-designated*. In fact, in this case even the logical equivalence connective ($\tilde{\leftrightarrow}$) derived using ~ instead of ¬ does not give a truth table which is designated in the diagonal and non-designated everywhere else, for it treats b and t as equals (and the same goes for f and n), resulting in a weaker type of equivalence.

Table 6. $\varphi \leftrightarrow \psi$.

$\varphi \backslash \psi$	n	f	t	b
n	n	n	n	t
f	n	t	f	b
t	n	f	t	b
b	t	b	b	b

Table 7. φ^n, φ^f, φ^t and φ^b.

φ	φ^n	φ^f	φ^t	φ^b
n	t	f	f	f
f	f	t	f	f
t	f	f	t	f
b	f	f	f	t

The reason for adding the classical negation (~) to a language which already has a negation operator (¬) is that this increases the expressivity of the language[3]. For instance, we can now define formulas discriminating which of the four truth values a formula φ has: $\varphi^n \overset{\text{def}}{=} ({\sim}\varphi \wedge {\sim}\neg\varphi)$; $\varphi^f \overset{\text{def}}{=} {\sim}{\sim}({\sim}\varphi \wedge \neg\varphi)$; $\varphi^t \overset{\text{def}}{=} {\sim}{\sim}(\varphi \wedge {\sim}\neg\varphi)$; $\varphi^b \overset{\text{def}}{=} {\sim}{\sim}(\varphi \wedge \neg\varphi)$. As can be seen in Table 7, φ^i is true if and only if φ has truth value i, for $i \in \{n, f, t, b\}$, and false otherwise. Using these connectives, it is easy to see that a stronger notion of logical equivalence can be expressed in FVEL:

$$\varphi \Leftrightarrow \psi \overset{\text{def}}{=} (\varphi^n \wedge \psi^n) \vee (\varphi^f \wedge \psi^f) \vee (\varphi^t \wedge \psi^t) \vee (\varphi^b \wedge \psi^b)$$

Since this formula is complex and difficult to evaluate, we will often favor the use of the metalanguage operator $\equiv$, defined by:

$$\varphi \equiv \psi \overset{\text{def}}{=} (M, s \models \varphi \text{ iff } M, s \models \psi) \text{ and } (M, s \models \neg\varphi \text{ iff } M, s \models \neg\psi),$$
$$\text{for all models } M \text{ and all states } s.$$

The formula $\varphi \Leftrightarrow \psi$ is designated if and only if φ and ψ have the same truth value. We can use $\Leftrightarrow$ and $\equiv$ interchangeably for it holds that:

---

[3] Interestingly, Girard and Tanaka [8] show that the standard definition of $p \rightarrow q$ as $\neg p \vee q$ does not suffice to prove reduction axioms for public announcements when working with an epistemic extension of Priest's three-valued Logic of Paradox. To circumvent that, they introduced an alternative implication. Our classical negation has a similar role w.r.t. our reduction axioms of Sect. 5.

Proposition 1. *$\varphi \equiv \psi$ iff $\models \varphi \Leftrightarrow \psi$.*

The operator $\equiv$ will be widely used for the demonstrations of Sect. 5.

3 Tableaux

In this section we will describe a tableaux system for FVEL[4]. A tableau is a tree-like structure used for checking derivability and theoremhood. Each branch of the tableau is a set of restrictions that may ultimately determine a model, which is said to be the model *induced* by that branch. In the system used in this paper, each node of the tree is of the form $(\varphi, \pm i)$ or $(ir_m j)$. The first type of node tells if the formula φ is designated (+) or non-designated (−) in state $f(i)$, where f is a function from $\mathbb{N}$ to states (of the induced model). The second type of node says that $f(i)R_m f(j)$ for $i, j \in \mathbb{N}$, where R_m is the accessibility relation of agent m (in the induced model).

The root of the tableau is of the form $(\varphi, -0)$, where φ is the desired conclusion. This root node asserts that the conclusion is non-designated in an arbitrary state $f(0)$. Below the root comes a series of nodes $P_1, P_2, ..., P_n$ such that P_1 is child of the root, P_2 is child of P_1, P_3 is child of P_2, and so on. Each node in this sequence is in the form $(\psi, +0)$, where ψ is a premise. A *branch* is a path from the root to a leaf of the tableau. A branch is called *closed* if it contains a contradiction, that is, a pair of nodes $(\chi, +k)$ and $(\chi, -k)$ for some formula χ and $k \in \mathbb{N}$. If the branch does not contain a contradiction and its leaf node does not fulfill the conditions for the application of any rule (that was not yet applied), then we say the branch is *open*. If the branch contains no contradictions and not all applicable rules were applied, the branch is neither closed nor open, it is *incomplete*. If no branch is incomplete we say the tableau is *complete*.

A successful proof is one where all the branches of the tableau are closed, showing that it is impossible that φ is non-designated in a state where all the premises are designated, and therefore φ is provable from the premises. In that case we say $\Sigma \vdash \varphi$, where Σ is the finite set of premises and φ is the conclusion. If φ is proven from an empty set of premises we write $\vdash \varphi$, and call φ a *theorem*.

To apply a rule to the tableau, a leaf node must be chosen. If the branch to which the leaf node belongs satisfies the conditions of the rule (which are represented in the left-hand side of the rule), certain nodes can be appended as child nodes to that leaf (according to the specification in the right-hand side). Conditions require simply the existence of a set of nodes with a particular format. Some rules allow the creation of one child node, other rules allow the creation of two child nodes, in some cases in series (denoted by a comma in the rules below), in other cases in parallel (denoted by a vertical bar: |).

To obtain a tableau calculus for FVEL, we started with the rules from the tableau system given in [16, p. 248], which, for the paper to be self-contained, are reproduced here (rules $R1$–$R14$). We then added four more rules for classical negation (rules $R15$–$R18$). This tableau system will be further augmented in

[4] Compare [13], which provides a tableaux for **BK** (discussed in Sect. 7).

Sect. 4 to prove some correspondence results between the tableau system and classes of frames and in Sect. 5 to cope with public announcements.

$$(\varphi \wedge \psi, +i) \Longrightarrow (\varphi, +i), (\psi, +i) \tag{R1}$$

$$(\varphi \wedge \psi, -i) \Longrightarrow (\varphi, -i) \mid (\psi, -i) \tag{R2}$$

$$(\varphi \vee \psi, +i) \Longrightarrow (\varphi, +i) \mid (\psi, +i) \tag{R3}$$

$$(\varphi \vee \psi, -i) \Longrightarrow (\varphi, -i), (\psi, -i) \tag{R4}$$

$$(\neg(\varphi \vee \psi), \pm i) \Longrightarrow (\neg\varphi \wedge \neg\psi, \pm i) \tag{R5}$$

$$(\neg(\varphi \wedge \psi), \pm i) \Longrightarrow (\neg\varphi \vee \neg\psi, \pm i) \tag{R6}$$

$$(\neg\neg\varphi, +i) \Longrightarrow (\varphi, +i) \tag{R7}$$

$$(\neg\neg\varphi, -i) \Longrightarrow (\varphi, -i) \tag{R8}$$

$$(\Box_m\varphi, +i), (ir_mj) \Longrightarrow (\varphi, +j) \tag{R9}$$

$$(\Box_m\varphi, -i) \Longrightarrow (ir_mj), (\varphi, -j) \tag{R10}$$

$$(\Diamond_m\varphi, +i) \Longrightarrow (ir_mj), (\varphi, +j) \tag{R11}$$

$$(\Diamond_m\varphi, -i), (ir_mj) \Longrightarrow (\varphi, -j) \tag{R12}$$

$$(\neg\Box_m\varphi, \pm i) \Longrightarrow (\Diamond_m\neg\varphi, \pm i) \tag{R13}$$

$$(\neg\Diamond_m\varphi, \pm i) \Longrightarrow (\Box_m\neg\varphi, \pm i) \tag{R14}$$

$$(\neg{\sim}\varphi, +i) \Longrightarrow (\varphi, +i) \tag{R15}$$

$$(\neg{\sim}\varphi, -i) \Longrightarrow (\varphi, -i) \tag{R16}$$

$$({\sim}\varphi, +i) \Longrightarrow (\varphi, -i) \tag{R17}$$

$$({\sim}\varphi, -i) \Longrightarrow (\varphi, +i) \tag{R18}$$

Notice that rules $R17$ and $R18$ invert the sign before i. In rules $R10$ and $R11$, the number j must be fresh in the branch. Figures 4 and 5 show two examples of proofs using the tableau system. In the first proof, no rule can be applied to the leaf node (its branch does not fulfill the conditions of any rule that was not already applied), and therefore the formula ${\sim}\Box_m p \vee p$ is not a theorem. The second example proves the validity $(p \vee {\sim}p) \wedge (\neg p \vee {\sim}\neg p)$.

Now we can prove soundness and completeness of this enhanced tableau with respect to FVEL.

$$\dfrac{{\sim}\Box_m p \vee p, -0}{{\sim}\Box_m p, -0}\,R4 \qquad \dfrac{p, -0}{\Box_m p, +0}\,R18$$

Fig. 4. An open tableau

$$\dfrac{(p \vee {\sim}p) \wedge (\neg p \vee {\sim}\neg p), -0}{\dfrac{p \vee {\sim}p, -0}{\begin{matrix} p, -0 \\ \dfrac{{\sim}p, -0}{\begin{matrix} p, +0 \\ \times \end{matrix}}\,R18 \end{matrix}}\,R4 \quad \Big| \quad \dfrac{\neg p \vee {\sim}\neg p, -0}{\begin{matrix} \neg p, -0 \\ \dfrac{{\sim}\neg p, -0}{\begin{matrix} \neg p, +0 \\ \times \end{matrix}}\,R18 \end{matrix}}\,R4}\,R2$$

Fig. 5. A closed tableau: $p, +0$ contradicts $p, -0$, and $\neg p, +0$ contradicts $\neg p, -0$.

Theorem 1. *For any finite set of formulas* $\Sigma \cup \{\varphi\}$, $\Sigma \vdash \varphi$ *iff* $\Sigma \models \varphi$.

4 Correspondence Results

Now we will take a look at standard axioms and inference rules from modal logics. *Modus Ponens* (abbreviated as MP, $\{\varphi, \varphi \to \psi\} \vdash \psi$) is not a sound inference rule, as is the case for **FDE** (see Proposition 2). Necessitation (NEC, $\vdash \varphi \Longrightarrow \vdash \Box_m\varphi$), on the other hand, is sound. The axiom K is not a theorem of our logic. However, if K is built using classical negation instead of **FDE** negation ($\Box_m(\varphi \tilde{\to} \psi) \tilde{\to} (\Box_m\varphi \tilde{\to} \Box_m\psi)$), then it is a theorem of FVEL. Axioms T ($\Box_m\varphi \to \varphi$), 4 ($\Box_m\varphi \to \Box_m\Box_m\varphi$) and 5 ($\neg\Box_m\varphi \to \Box_m\neg\Box_m\varphi$) are not theorems (neither in their regular version, nor in their version with classical negation). Whether these or any other formulas are theorems can be easily checked using the tableau method.

Proposition 2. *MP is not a sound inference rule for FVEL.*

Proposition 3. *NEC is a sound inference rule for FVEL.*

The version of K derived from $\sim$ (let us call it $\tilde{K}$) is valid in all frames. Not surprisingly, the correspondence between some properties of frames and validity of formulas still hold, as shown by the propositions below (where the versions of T, B, 4, D and 5 derived using the classical negation are named $\tilde{T}$, $\tilde{B}$, $\tilde{4}$, $\tilde{D}$ and $\tilde{5}$, respectively).

Proposition 4. *$\mathcal{F} \models \tilde{T}$ iff $\mathcal{F}$ is reflexive.*

Proposition 5. *$\mathcal{F} \models \tilde{4}$ iff $\mathcal{F}$ is transitive.*

Proposition 6. *$\mathcal{F} \models \tilde{B}$ iff $\mathcal{F}$ is symmetric.*

Proposition 7. *$\mathcal{F} \models \tilde{D}$ iff $\mathcal{F}$ is serial.*

Proposition 8. *$\mathcal{F} \models \tilde{5}$ iff $\mathcal{F}$ is Euclidian.*

Now, it can be shown that the tableau system is complete with respect to the class of models satisfying the above properties if we augment the system with the following rules:

$$\bullet \Longrightarrow (ir_m i) \qquad (\mathrm{R}\rho)$$
$$(ir_m j), (jr_m k) \Longrightarrow (ir_m k) \qquad (\mathrm{R}\tau)$$
$$(ir_m j) \Longrightarrow (jr_m i) \qquad (\mathrm{R}\sigma)$$
$$\bullet \Longrightarrow (ir_m j) \qquad (\mathrm{R}\eta)$$
$$(ir_m j), (ir_m k) \Longrightarrow (jr_m k) \qquad (\mathrm{R}\epsilon)$$

Rules (Rρ) and (Rη) can only be applied if there is a previous appearance of the label i in the branch, and (Rη) additionally requires that j be fresh in the branch. We use the symbol $\vdash_\rho$ for the provability relation of the tableau system augmented with the rule (Rρ), and similarly for the other rules. Likewise, we use $\models_\rho$ to represent satisfiability restricted only to reflexive models, $\models_\tau$ for transitive models, $\models_\sigma$ for symmetric models, $\models_\eta$ for serial models and $\models_\epsilon$ for Euclidian models.

Theorem 2. *For all finite sets of formulas $\Sigma \cup \varphi$, the following statements hold:*

- $\Sigma \vdash_\rho \varphi$ *iff* $\Sigma \models_\rho \varphi$
- $\Sigma \vdash_\tau \varphi$ *iff* $\Sigma \models_\tau \varphi$
- $\Sigma \vdash_\sigma \varphi$ *iff* $\Sigma \models_\sigma \varphi$
- $\Sigma \vdash_\eta \varphi$ *iff* $\Sigma \models_\eta \varphi$
- $\Sigma \vdash_\epsilon \varphi$ *iff* $\Sigma \models_\epsilon \varphi$

5 Public Announcements

In this section, we extend the language with public announcements. The semantics for the new operator is defined as follows (cf. [4,15]):

$$M, s \models [\varphi]\psi \quad \text{iff} \quad M, s \models \varphi \text{ implies } M|_\varphi, s \models \psi$$
$$M, s \models \neg[\varphi]\psi \quad \text{iff} \quad M, s \models \varphi \text{ and } M|_\varphi, s \models \neg\psi$$

where $M = \langle S, R, V\rangle$ and $M|_\varphi = \langle S', R', V'\rangle$, with $S' = \{s \in S \mid M, s \models \varphi\}$, $R' = R \cap (S' \times S')$ and $V' = V|_{P\times S'}$.

The model of Fig. 1(right), upon the public announcement of $\neg p$, would be transformed according to Fig. 6.

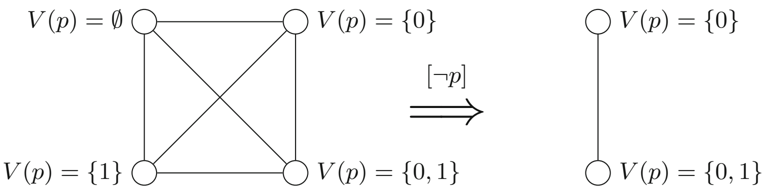


Fig. 6. The announcement of $\neg p$.

Notice that, for propositional atoms, the announcement of p does not delete worlds where $\neg p$ holds, but only worlds where p does not hold, that is, worlds where $\sim p$ holds. To delete worlds where $\neg p$ holds we would have to announce $\sim\neg p$, so that only worlds s with $M, s \models \sim\neg p$ (which is equivalent to $M, s \not\models \neg p$) would survive. Resorting again to our database analogy, it is possible to understand a public announcement of p (or $\neg p$) as showing to all agents the result of the query "p?" (or "$\neg p$?") to the database. More generally, the effect of the public announcement of p (or $\neg p$) is to make everybody aware that p was said true (or false) by the information source, which differs from the intuition about public announcements in standard logics.

As is the case for Public Announcement Logic [7,15], public announcements in FVEL do not increase expressivity. Any formula with public announcements can be rewritten as a standard FVEL formula, through the use of the following reduction axioms.

$$[\varphi]p \Leftrightarrow {\sim}\varphi \vee p \qquad \text{(AnAt)}$$
$$[\varphi]\neg p \Leftrightarrow {\sim}\varphi \vee \neg p \qquad (\text{An}\neg)$$
$$[\varphi](\psi \wedge \chi) \Leftrightarrow [\varphi]\psi \wedge [\varphi]\chi \qquad (\text{An}\wedge)$$
$$[\varphi]\neg(\psi \wedge \chi) \Leftrightarrow [\varphi]\neg\psi \vee [\varphi]\neg\chi \qquad (\text{An}\neg\wedge)$$
$$[\varphi]\Box_m\psi \Leftrightarrow {\sim}\varphi \vee \Box_m[\varphi]\psi \qquad (\text{An}\Box)$$
$$[\varphi]\neg\Box_m\psi \Leftrightarrow {\sim}\varphi \vee \neg\Box_m[\varphi]\psi \qquad (\text{An}\neg\Box)$$
$$[\varphi]{\sim}\psi \Leftrightarrow {\sim}\varphi \vee {\sim}[\varphi]\psi \qquad (\text{An}{\sim})$$
$$[\varphi]\neg{\sim}\psi \Leftrightarrow {\sim}\varphi \vee {\sim}{\sim}[\varphi]\psi \qquad (\text{An}\neg{\sim})$$

Proposition 9. *All above formulas for public announcements in FVEL are valid.*

Before proving that any formula with public announcements can be rewritten as an equivalent formula of FVEL where the public announcement operator does not occur, we need to prove the following lemma:

Lemma 1. *For all formulas φ, ψ, χ of FVEL with public announcements, $\varphi \equiv \psi$ implies $\chi \equiv \chi[\psi/\varphi]$. ($\chi[\psi/\varphi]$ is the formula that results from χ after uniform substitution of φ by ψ.)*

Now we can prove the following:

Proposition 10. *For any formula φ of FVEL with public announcements, a formula φ' of FVEL without public announcements can be found such that $\varphi \equiv \varphi'$.*

To account for public announcements, the tableau system can be extended with the following rule schema (which actually represents nine rules):

$$(\varphi, \pm i) \Longrightarrow (\varphi[\chi/\psi], \pm i) \quad \text{(RPA)}$$

where $\psi \Leftrightarrow \chi$ or $\chi \Leftrightarrow \psi$ is one of the public announcement axioms above[5]. Finally we can prove completeness of the extended tableau system with respect to FVEL with public announcements.

Theorem 3. *For any finite set of formulas $\Sigma \cup \{\varphi\}$ of FVEL with public announcements, $\Sigma \vdash \varphi$ iff $\Sigma \models \varphi$.*

---

[5] See [1] for a different approach to tableaux for logics with public announcements, and [9] for tableaux for logics with public announcements that use translations as rules in a similar fashion.

6 A Simple Example

Now we describe the situation depicted in Fig. 7. John (j) knows that there are studies regarding health benefits of coffee consumption, for he often sees headlines about the subject. However, he never cared enough to read those articles, so he is sure that there is evidence *for* or *against* (or even *both for and against*) *coffee being beneficial for health* (p), but he does not know exactly what is the status of the evidence about p, he only knows that there is some information. Looking at Fig. 7 it is easy to see that $\Box_j((p \wedge {\sim}\neg p) \vee (\neg p \wedge {\sim} p) \vee (p \wedge \neg p))$, which is equivalent to $\Box_j(p \vee \neg p)$, holds in the actual world (s_3).

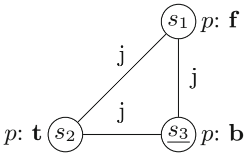


Fig. 7. Some evidence for p

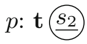


Fig. 8. No false evidence.

Kate (k), on the other hand, is a researcher on the effects of coffee on health, and for this reason she knows exactly what evidence is available (her relation R_k has only reflexive arrows, which are not represented). We can see that $M, s_3 \models \Box_k(p \wedge \neg p)$, that is, Kate actually knows that there is evidence both for and against the benefits of coffee. Moreover, John knows Kate and her job, so he also knows that she knows about p, whatever its status is (using abbreviations defined in Sect. 2.3: $\Box_j(\Box_k p^f \vee \Box_k p^t \vee \Box_k p^b)$). Likewise, Kate knows that John simply knows that there is some information about p ($\Box_k(\Box_j(p \vee \neg p) \wedge {\sim}\Box_j(p \wedge \neg p))$).

Now suppose the actual world is s_2, and so p is *true*, i.e., there is only positive evidence for p (and Kate knows that). Suppose also that Kate announces that a paper was published in a very respectable journal reassessing all the main studies that concluded that coffee was not beneficial for health, and concluded that those studies were not reliable due to sloppy methodology. Now this is equivalent to an announcement of $\Box_k{\sim}\neg p$ (Kate knows that there is no evidence for the falsity of p). This announcement results in the removal of the worlds where evidence for the falsity of p is present, namely s_1 and s_3. The resulting model is the one in Fig. 8, where John knows the status of p too. The formula ${\sim}\Box_j(p \wedge {\sim}\neg p) \wedge [\Box_k{\sim}\neg p]\Box_j(p \wedge {\sim}\neg p)$, which is satisfied in s_2 before the announcement, reflects the fact that John does not know the status of p, but after Kate's announcement he learns that p is true.

This example shows the dynamics of the agents' knowledge about available information/evidence. It might be puzzling, however, to notice that these models actually do not say much about factual knowledge. Nevertheless, it is based on information and evidence that one can form knowledge and beliefs. This observation calls for an extension of FVEL in which knowledge about evidence could be converted into factual knowledge or belief.

7 Related Work

Many authors have studied the subject of many-valued modal logics [6,11–14,17–19,21]. Of these, the most closely related to ours are Odintsov and Wansing's and Rivieccio's papers. Both papers explore some kind of four-valued epistemic logics. We will now discuss the similarities and differences between these and our approach.

In their paper [13], Odintsov and Wansing describe a logic called **BK** (a Belnapian variant of **K**), which is closely related to FVEL. They also provide a tableaux system similar to ours, but their paper does not cover public announcements, nor the correspondence results presented here. There are other small differences between the two formalisms. The logic **BK** uses two entailment symbols – support for truth ($\models^+$) and support for falsity ($\models^-$) – whereas we opted for an additional negation. While this small change still results in equi-expressive logics, we can express statements like $M, s \models \neg p \wedge \neg q$ directly, when **BK** always place the "negation" in front of the formula: $M, s \models^- p \vee q$. The latter has a more natural equivalent in our logic: $\neg(p \vee q)$. Moreover, this choice allows us to announce a formula like $\neg p$, which in **BK** is only expressible w.r.t. a state of a model ($M, s \models^- p$).

Rivieccio [17], with a very different formalism (focused on algebraic semantics), describes a logic that seems to be an extension of the one presented here, with a more expressive language and a four-valued accessibility relation. Rivieccio's logic has a symbol $\perp$ which is always evaluated to *none*, while in our language this is not expressible (no formula is evaluated to *none* if $V(p, s) \neq \emptyset$ for all p and s). His work features a Hilbert-style calculus instead of a tableau. He provides an axiomatisation which includes reduction axioms for public announcements, but it is not obvious how both axiomatisations for public announcements compare, since the languages used are slightly different.

Another work closely related to ours is being done by Majer and Sedlár [10]. They also study the logic **BK**. Their work, however, does not include public announcements nor a tableau system (as far as we know).

Finally, a unique contribution of our paper is the intuitive interpretation given to FVEL. These insights show a way in which many-valued modal logics could be used in practical applications, and open some new possibilities for research that will be discussed in the next section.

8 Conclusions and Future Work

In this paper, we presented a multi-agent four-valued logic with two distinct layers: one informational and the other epistemic. The idea of having two separate layers may be useful in the modelling of realistic scenarios where agents have access to an inconsistent or incomplete base of information. Some examples are the database scenario described in the introduction, or a robot who collects data through several sensors, which may result in inconsistent data due to sensors' inaccuracy.

First degree entailment was used as the propositional basis for the logic, with its four-valued atoms playing the role of the "informational layer", in which a proposition could be both true and false or have no value at all. A modal layer was built on top of that, introducing an epistemic aspect to the logic. The accessibility relation, then, defines the knowledge of the agents about the possibly contradictory or incomplete informational layer.

Moreover, classical negation was added to the language, increasing its expressivity. That addition allowed us to define an equivalence operator and reduction axioms for public announcements. A tableau calculus and some correspondence results were provided. While on the technical side there are similarities among our approach and others, new results have been presented and, not least, some intuition for these logics have been given.

For further work, a number of possibilities were opened. There are other possible intuitive readings for FVEL, besides the two-layered interpretation presented here (Majer and Sedlár's work offers one alternative). Furthermore, besides the public announcements studied here, a range of dynamic operators can be considered in combination with this logic. Some of these operators will act on the informational layer, and some on the epistemic layer – and perhaps some of them could act on both layers. A useful example of dynamic operation would be a method for "filtering" those inconsistent sets of beliefs in order to produce a consistent epistemic state (along the lines of belief revision, in particular [20]). It might be valuable as well to understand which actions the agents are justified to carry out on the basis of such inconsistent belief states. Other update actions (along the lines of [3]) could also be studied, for example, the actions mentioned in the introduction, which change the informational layer instead of only changing the knowledge about it. These actions, instead of removing states, could just add or remove truth (or falsity) from the value of a proposition in all worlds.

Acknowledgements. I am very grateful to Barteld Kooi, Rineke Verbrugge and Allard Tamminga for comments that greatly improved this work. I would also like to thank my colleagues from the Philosophy Department who participated in a discussion of a preliminary version of this paper, and the anonymous referees for the useful suggestions.

Appendix

Proof (Lemma 1). First, let us label each atom of χ with a unique integer. In such a way, we can distinguish occurrences $\varphi_1, \varphi_2, ..., \varphi_n$ of any subformula φ that appears in χ n times, according to the labels in their atoms. The notation $\chi_\#$ will denote a labelled version of χ (that is, an occurrence of χ) in whose number we are not particularly interested. We will first prove that for any occurrence φ_k of any subformula φ of χ, if $\varphi \equiv \psi$ then $\chi \equiv \chi[\psi/\varphi_k]$, where $\chi[\psi/\varphi_k]$ denotes the formula χ after replacing the occurrence φ_k by ψ. Before starting the proof, we need to define a function sub^l from labelled formulas to a set of labelled formulas. Intuitively, $sub^l(\chi_\#)$ denotes the labelled subformulas of χ at level l, level 0 being the root, that is, $\chi_\#$ itself.

$$sub^0(\chi_\#) = \{\chi_\#\}$$

$$sub^1(\chi_\#) = \begin{cases} \emptyset, & \text{if } \chi_\# = p_\# \text{ for some atom } p. \\ \{\varphi_\#\}, & \text{if } \chi_\# \in \{{\sim}\varphi_\#, \neg\varphi_\#, \Box\varphi_\#\} \\ \{\varphi_\#, \psi_\#\}, & \text{if } \chi_\# \in \{\varphi_\# \wedge \psi_\#, [\varphi_\#]\psi_\#\} \end{cases}$$

$$sub^i(\chi_\#) = \bigcup_{\zeta_\# \in sub^{i-1}(\chi_\#)} sub^1(\zeta_\#), \qquad \text{for } i > 1$$

The proof will be by induction on the level l of φ_k. The base case is when $l = 0$, that is, $\varphi_k \in sub^0(\chi_\#)$, or simply $\varphi_k = \chi_\#$. Trivially, if $\varphi \equiv \psi$ then $\chi \equiv \chi[\psi/\varphi_k]$, since $\chi[\psi/\varphi_k] = \psi$ and $\chi \equiv \psi$ follows from $\chi = \varphi$. Induction Hypothesis (I.H.): for all $l < n$, if $\varphi_k \in sub^l(\chi_\#)$ and $\varphi \equiv \psi$, then $\chi \equiv \chi[\psi/\varphi_k]$. Given that for all occurrences $\delta_\#$ of subformulas of χ at level $n-1$ it holds that $\delta \equiv \psi$ implies $\chi \equiv \chi[\psi/\delta_\#]$, in the induction step we need to show that for all occurrences φ_k of subformulas of χ at level n it holds that $\varphi \equiv \psi$ implies $\chi \equiv \chi[\psi/\varphi_k]$. We will divide the step in cases according to the formula $\delta_\# \in sub^{n-1}(\chi_\#)$ such that $\varphi_k \in sub^1(\delta_\#)$, and to the position of φ_k in $\delta_\#$.

$\delta_\# = \varphi_k \wedge \xi$: suppose $\varphi \equiv \psi$. Then (for all models M and states s) $M, s \models \varphi_k \wedge \xi$ iff ($M, s \models \psi$ and $M, s \models \xi$) iff $M, s \models \psi \wedge \xi$. Also, $M, s \models \neg(\varphi_k \wedge \xi)$ iff ($M, s \models \neg\varphi_k$ or $M, s \models \neg\xi$) iff ($M, s \models \neg\psi$ or $M, s \models \neg\xi$) iff $M, s \models \neg(\psi \wedge \xi)$. Therefore, $\varphi \wedge \xi \equiv \psi \wedge \xi$, and by the I.H. $\chi \equiv \chi[\psi \wedge \chi/\varphi_k \wedge \xi] = \chi[\psi/\varphi_k]$. The case for $\delta_\# = \xi \wedge \varphi_k$ is completely analogous.

$\delta_\# = \neg\varphi_k$: Suppose $\varphi \equiv \psi$. Then ($M, s \models \neg\varphi_k$ iff $M, s \models \neg\psi$) and ($M, s \models \neg\neg\varphi_k$ iff $M, s \models \neg\neg\psi$), from which it follows that $\neg\varphi \equiv \neg\psi$. By the I.H. we have that $\chi \equiv \chi[\neg\psi/\neg\varphi_k] = \chi[\psi/\varphi_k]$.

$\delta_\# = {\sim}\varphi_k$: Suppose $\varphi \equiv \psi$. Then ($M, s \models {\sim}\varphi_k$ iff $M, s \models {\sim}\psi$) and ($M, s \models \neg{\sim}\varphi_k$ iff $M, s \models \neg{\sim}\psi$), from which it follows that ${\sim}\varphi \equiv {\sim}\psi$. By the I.H. we have that $\chi \equiv \chi[{\sim}\psi/{\sim}\varphi_k] = \chi[\psi/\varphi_k]$.

$\delta_\# = \Box_i\varphi_k$. Suppose $\varphi \equiv \psi$. Then ($M, s \models \Box_i\varphi_k$ iff $\forall t$ such that sR_it $M, t \models \varphi_k$ iff $\forall t$ such that $sR_itM, t \models \psi$ iff $M, s \models \Box_i\psi$) and ($M, s \models \neg\Box_i\varphi_k$ iff $\exists t$ such that sR_it and $M, t \models \neg\varphi_k$ iff $\exists t$ such that sR_it and $M, t \models \neg\psi$ iff $M, s \models \neg\Box_i\psi$). From that it follows that $\Box_i\varphi_k \equiv \Box_i\psi$, and by the I.H. we get $\chi \equiv \chi[\Box_i\psi/\Box_i\varphi_k] = \chi[\psi/\varphi_k]$.

$\delta_\# = [\varphi_k]\xi$. Suppose $\varphi \equiv \psi$. Then $M, s \models [\varphi_k]\xi$ iff ($M, s \not\models \varphi_k$ or $M|_{\varphi_k}, s \models \xi$). But since $\varphi \equiv \psi$, $M|_{\varphi_k} = M|_\psi$. Then $M, s \models [\varphi_k]\xi$ iff ($M, s \not\models \psi$ or $M|_\psi, s \models \xi$) iff $M, s \models [\psi]\xi$. $M, s \models \neg[\varphi_k]\xi$ iff ($M, s \models \varphi_k$ and $M|_{\varphi_k}, s \models \neg\xi$) iff ($M, s \models \psi$ and $M|_\psi \models \neg\xi$) iff $M, s \models \neg[\psi]\xi$. So $[\varphi_k]\xi \equiv [\psi]\xi$, then by the I.H. $\chi \equiv \chi[[\psi]\xi/[\varphi_k]\xi] = \chi[\psi/\varphi_k]$. The case for $\delta_\# = [\xi]\varphi_k$ is similar, but even easier.

Now the induction is finished and we have proven that for any occurrence φ_k of a subformula φ of χ, if $\varphi \equiv \psi$ then $\chi \equiv \chi[\psi/\varphi_k]$. From this it is easy to see that for all φ, ψ, χ, if $\varphi \equiv \psi$ then $\chi \equiv \chi[\psi/\varphi]$. Suppose a subformula φ of χ has occurrences $\varphi_1, \varphi_2, ..., \varphi_n$ (any formula of FVEL must be finite) and $\varphi \equiv \psi$.

From the previous proof, it follows that $\chi \equiv \chi[\psi/\varphi_1] \equiv \chi[\psi/\varphi_1][\psi/\varphi_2] \equiv ... \equiv \chi[\psi/\varphi_1]...[\psi/\varphi_n] \equiv \chi[\psi/\varphi]$.

Proof (Proposition 10). First we will assume the following claims:

(Claim 1). If $[\varphi]\psi \equiv \chi$, then $\zeta([\varphi]\psi) \equiv \zeta[\chi/[\varphi]\psi]$.

(Claim 2). Given a formula $[\varphi]\psi$, where ψ contains no announcements, we can always find a formula χ without announcements such that $[\varphi]\psi \equiv \chi$.

Given any formula φ of FVEL with public announcements, we can choose a subformula $[\psi]\chi$ of it such that χ contains no announcements and, if (Claim 2) is true, find an equivalent ζ without announcements ($\zeta \equiv [\psi]\chi$). If (Claim 1) is true, we can replace $[\psi]\chi$ by ζ in φ preserving the truth value, that is, $\varphi \equiv \varphi[\zeta/[\psi]\chi]$. Since any FVEL formula with public announcements is finite, we can repeat this procedure until we reach an equivalent formula without public announcements.

Now, (Claim 1) is a corollary of Lemma 1. We now prove (Claim 2) by structural induction on ψ. Induction Hypothesis (I.H.): for all proper subformulas ψ' of ψ, $[\varphi]\psi'$ has an equivalent formula without announcements. Base: if ψ has form p, $\neg p$ or $\neg\Box\psi'$, by axioms (AnAt), (An$\neg$) and (An$\neg\Box$), respectively, we can find an equivalent formula without announcements, since ψ itself does not contain announcements. Step: for each possible connective we have a reduction axiom which reduces the original formula into another such that the formulas under the announcement φ are simpler. By the I.H., these formulas have an equivalent formula without announcements.

Proof (Theorem 3). The proof system being considered here is the tableau calculus for FVEL augmented with the public announcements' axioms and a substitution rule (if $\varphi \equiv \psi$ and $\Sigma \vdash \chi$ then $\Sigma \vdash \chi[\psi/\varphi]$). Soundness is already proven (soundness for the tableau for FVEL is proven in Theorem 1, soundness of public announcements' axioms is proven in Proposition 9 and soundness of the substitution rule follows from Lemma 1). For completeness, if φ is a formula without announcements, then $\Sigma \models \varphi$ implies $\Sigma \vdash \varphi$ due to completeness of the FVEL proof system. If φ contains announcements, then, by Proposition 10, we can apply a finite sequence of reduction axioms $Ax_1, Ax_2, ..., Ax_n$ on φ to obtain an equivalent formula $t(\varphi)$ without announcements. Since the proof system for FVEL is complete, $t(\varphi)$ can be proven in it. Now, if we apply the substitution rule n times with the reduction axioms $Ax_n, Ax_{n-1}, ..., Ax_1$, we will obtain the original formula φ.

References

1. Balbiani, P., van Ditmarsch, H., Herzig, A., de Lima, T.: Tableaux for public announcement logic. J. Logic Comput. **20**(1), 55 (2010)
2. Belnap, N.: A useful four-valued logic. In: Dunn, J.M., Epstein, G. (eds.) Modern Uses of Multiple-Valued Logic. Episteme, vol. 2, pp. 5–37. Springer, Dordrecht (1977). https://doi.org/10.1007/978-94-010-1161-7_2
3. van Benthem, J., van Eijck, J., Kooi, B.: Logics of communication and change. Inf. Comput. **204**(11), 1620–1662 (2006)

4. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic, vol. 337. Springer, Dordrecht (2007). https://doi.org/10.1007/978-1-4020-5839-4
5. Dunn, J.: Intuitive semantics for first-degree entailments and 'coupled trees'. Philos. Stud. **29**(3), 149–168 (1976)
6. Fitting, M.: Many-valued modal logics. Fundamenta Informaticae **15**(3–4), 235–254 (1991)
7. Gerbrandy, J., Groeneveld, W.: Reasoning about information change. J. Logic Lang. Inform. **6**(2), 147–169 (1997)
8. Girard, P., Tanaka, K.: Paraconsistent dynamics. Synthese **193**(1), 1–14 (2016)
9. Hansen, J.: Terminating tableaux for dynamic epistemic logics. Electron. Not. Theoret. Comput. Sci. **262**, 141–156 (2010)
10. Majer, O., Sedlár, I.: Paraconsistent epistemic logic. Talk given in the LogiCIC Workshop 2016 in the University of Amsterdam (2016)
11. Morgan, C.: Local and global operators and many-valued modal logics. Notre Dame J. Formal Logic **20**(2), 401–411 (1979)
12. Morikawa, O.: Some modal logics based on a three-valued logic. Notre Dame J. Formal Logic **30**(1), 130–137 (1988)
13. Odintsov, S., Wansing, H.: Modal logics with Belnapian truth values. J. Appl. Non-Class. Logics **20**(3), 279–301 (2010)
14. Ostermann, P.: Many-valued modal propositional calculi. Math. Logic Q. **34**(4), 343–354 (1988)
15. Plaza, J.: Logics of public communications. Synthese **158**(2), 165–179 (2007)
16. Priest, G.: An Introduction to Non-Classical Logic: From If to Is. Cambridge University Press, Cambridge (2008)
17. Rivieccio, U.: Algebraic semantics for bilattice public announcement logic. In: Indrzejczak, A., Kaczmarek, J., Zawidzki, M. (eds.) Proceedings of the Trends in Logic XIII, Lodz, Poland, 2–5 July 2014, pp. 199–215. Lodz University Press (2014)
18. Schotch, P., Jensen, J., Larsen, P., MacLellan, E.: A note on three-valued modal logic. Notre Dame J. Formal Logic **19**(1), 63–68 (1978)
19. Segerberg, K.: Some modal logics based on a three-valued logic. Theoria **33**(1), 53–71 (1967)
20. Tamminga, A.: Belief dynamics: (Epistemo) logical investigations. Ph.D. thesis, ILLC (2001)
21. Thomason, S.: Possible worlds and many truth values. Stud. Logica **37**(2), 195–204 (1978)

Dynamic Epistemic Logics of Introspection

Raul Fervari[1] and Fernando R. Velázquez-Quesada[2(✉)]

[1] FaMAF, Universidad Nacional de Córdoba, and CONICET, Córdoba, Argentina
rfervari@conicet.gov.ar
[2] ILLC, Universiteit van Amsterdam, Amsterdam, The Netherlands
FRVelazquezQuesada@uva.nl

Abstract. This work studies positive and negative introspection not as properties, but rather as actions that change the agent's knowledge. The actions are introduced as model update operations, with matching modalities expressing their effects. Sound and complete axiom systems are provided, and some properties are explored.

Keywords: Positive introspection · Negative introspection
Epistemic logic · Dynamic epistemic logic

1 Introduction

One of the reasons of the widespread use of *epistemic logic* (*EL*; [1]) is that it deals not only with an agent's knowledge about propositional facts, but also with her knowledge about her own (and eventually other agents') knowledge (*high-order* knowledge). This has been the starting point for the study of more complex multi-agent epistemic notions (e.g., common knowledge) that are crucial in multi-agent interaction, thus allowing *EL* to extend its range of applications, including not only philosophy (epistemology [2]), but also computer science (artificial intelligence [3]) and economics (game theory [4]).

In the study of agents with high-order knowledge, two of the most important concepts have been *positive introspection* (if the agent knows something, she knows that she knows it) and *negative introspection* (if the agent does not know something, she knows that she does not know it). One of the main advantages of the standard *EL* semantic structure, relational models, is that these two properties correspond, at the level of frames, to simple relational properties: to work with full positively introspection, it is enough to consider a transitive indistinguishability relation, and to deal with full negative introspection, it is enough to ask for such relation to be Euclidean. When these properties are not enforced, the agent might lack introspection, thus making her more 'real'. But, as in real life, not being introspective should not imply one will never be.

Recent works have studied properties of an *EL* agent's knowledge from a *dynamic* point of view, thinking about them in terms of the actions the agent can perform to achieve them. For example, closure under logical consequence

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 82–97, 2018.
https://doi.org/10.1007/978-3-319-73579-5_6

can be seen not as a 'static' property, but rather as the eventual result of awareness raising and 'syntactic' inference steps within awareness relational models [5,6], and also as the result of dynamics of evidence or deductive inference within neighbourhood models [7–9]. Following this idea, the present work studies introspection properties by defining epistemic actions that allow a non-introspective agent to reach them. These actions are represented in a *dynamic epistemic logic* (*DEL*; [10,11]) style: as accessibility-changing model operations. There are several examples of such operations in the literature, as the actions for belief revision and/or preference change studied in [12–15] and the logics for reasoning about dynamic policies investigated in [16,17]. There are also the more 'abstract' edge-deleting sabotage operation of [18], the edge-adding and swapping proposals in [19–22] and the general arrow update approach of [23].

The article is organised as follows. Section 2 introduces basic definitions about epistemic logic and propositional dynamic logic. Section 3 defines model operations to achieve positive introspection for general knowledge and also with respect to a formula. Section 4 focuses on similar operations for negative introspection. In all cases we study some properties of the operations, providing also sound and complete axiomatizations for their respective modalities. Finally, Sect. 5 draws conclusions.

2 Basic Definitions

This section recalls not only the basic definitions of basic epistemic logic, but also extensions that will be useful when providing axiom systems for modalities representing the introspection operations. Throughout this paper, let $\mathtt{P}$ be a countable set of atomic propositions.

Definition 2.1 (Relational Frame, Relational Model, Relational State). *A* relational frame *is a tuple* $F = \langle W, R \rangle$ *with* W *a non-empty set of possible worlds and* $R \subseteq (W \times W)$ *a binary relation, the agent's* indistinguishability *relation (which is not required to satisfy any property). A* relational model *is a tuple* $M = \langle F, V \rangle$ *with* F *a relational frame and* $V : \mathtt{P} \to \wp(W)$ *an atomic valuation. A tuple* (M, w) *with* M *a relational model and* w *a world in it (the* evaluation point*) is called a* relational state*.*

Next we introduce the *basic epistemic language* $\mathcal{L}_\Diamond$.

Definition 2.2 (Language $\mathcal{L}_\Diamond$). *Formulas* φ, ψ *of* $\mathcal{L}_\Diamond$ *are given by*

$$\varphi, \psi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid \Diamond \varphi,$$

with $p \in \mathtt{P}$*. Other Boolean connectives and constants as well as the modality* $\Box$ *are defined as usual (*$\Box \varphi := \neg \Diamond \neg\varphi$ *for the latter), and formulas of the form* $\Box \varphi$ *are read as* "the agent knows φ"*. For the semantic interpretation, given a relational state* (M, w) *with* $M = \langle W, R, V \rangle$*, formulas in* $\mathcal{L}_\Diamond$ *are interpreted as usual, with the cases of atomic propositions and the 'diamond' modality being*

$$(M, w) \Vdash p \quad \textit{iff} \quad w \in V(p)$$
$$(M, w) \Vdash \Diamond\, \varphi \quad \textit{iff} \quad \textit{there is } u \in W \textit{ such that } Rwu \textit{ and } (M, u) \Vdash \varphi.$$

A formula φ *is* true *at* w *in* M *when* $(M, w) \Vdash \varphi$. *A formula* φ *is* valid *(notation:* $\Vdash \varphi$*) when it is true in every world* w *of every model* M*.*

Theorem 2.1 (Axiom System for $\mathcal{L}_\Diamond$). *As it is well-known (e.g., [24, 25]), axiom schemes and rules on the first block of Table 1 form a sound and strongly complete axiom system ($\mathsf{L}_\Diamond$) for formulas of $\mathcal{L}_\Diamond$ w.r.t. relational models.*

Table 1. Axiom systems for $\mathcal{L}_\Diamond$ and some of its extensions.

Prop	$\vdash \varphi$ for φ a propositional tautology	*MP*	If $\vdash \varphi \to \psi$ and $\vdash \varphi$, then $\vdash \psi$
K	$\vdash \Box\,(\varphi \to \psi) \to (\Box\,\varphi \to \Box\,\psi)$	*N*	If $\vdash \varphi$, then $\vdash \Box\,\varphi$
Dual	$\vdash \Box\,\varphi \leftrightarrow \neg \Diamond \neg\varphi$		
$K_{\boxplus}$	$\vdash \boxplus\,(\varphi \to \psi) \to (\boxplus\,\varphi \to \boxplus\,\psi)$	$Nec_{\boxplus}$	If $\vdash \varphi$, then $\vdash \boxplus\,\varphi$
$Dual_{\boxplus}$	$\vdash \boxplus\,\varphi \leftrightarrow \neg \Diamond^{+} \neg\varphi$		
$FP_{\boxplus}$	$\vdash \Diamond^{+}\,\varphi \;\leftrightarrow\; \Diamond\,(\varphi \vee \Diamond^{+}\,\varphi)$	$Ind_{\boxplus}$	$\vdash \boxplus\,(\varphi \to \Box\,\varphi) \;\to\; (\Box\,\varphi \to \boxplus\,\varphi)$
Prop	$\vdash \varphi$ for φ a propositional tautology	*MP*	If $\vdash \varphi \to \psi$ and $\vdash \varphi$, then $\vdash \psi$
K_α	$\vdash [\alpha](\varphi \to \psi) \to ([\alpha]\,\varphi \to [\alpha]\,\psi)$	Nec_α	If $\vdash \varphi$, then $\vdash [\alpha]\,\varphi$
$Dual_\alpha$	$\vdash [\alpha]\,\varphi \leftrightarrow \neg \langle\alpha\rangle \neg\varphi$	?	$\vdash \langle ?\varphi \rangle\,\psi \leftrightarrow (\varphi \wedge \psi)$
$\triangleleft_1$	$\vdash \varphi \to [\triangleright]\,\langle\triangleleft\rangle\,\varphi$	$\triangleleft_2$	$\vdash \varphi \to [\triangleleft]\,\langle\triangleright\rangle\,\varphi$
$\cup$	$\vdash \langle \alpha \cup \beta \rangle\,\varphi \leftrightarrow (\langle\alpha\rangle\,\varphi \vee \langle\beta\rangle\,\varphi)$	;	$\vdash \langle \alpha\,;\beta \rangle\,\varphi \leftrightarrow \langle\alpha\rangle\,\langle\beta\rangle\,\varphi$
FP_*	$\vdash \langle \alpha^* \rangle\,\varphi \leftrightarrow (\varphi \vee \langle\alpha\rangle\,\langle\alpha^*\rangle\,\varphi)$	Ind_*	$\vdash [\alpha^*](\varphi \to [\alpha]\,\varphi) \to (\varphi \to [\alpha^*]\,\varphi)$

The following sections study languages with modalities for actions of introspection. To introduce their corresponding axiom systems, some extensions of the basic epistemic language will be useful. First, a transitive closure modality.

Definition 2.3 (Language $\mathcal{L}_{\Diamond,\Diamond^{+}}$). *The language $\mathcal{L}_{\Diamond,\Diamond^{+}}$ adds $\Diamond^{+}$ to $\mathcal{L}_\Diamond$. Given a relational state (M, w) with $M = \langle W, R, V \rangle$ and R^+ the transitive closure of R,*

$$(M, w) \Vdash \Diamond^{+}\varphi \quad \textit{iff} \quad \textit{there is } u \in W \textit{ such that } R^+wu \textit{ and } (M, u) \Vdash \varphi.$$

The dual modality $\boxplus$ is defined in the usual way ($\boxplus\,\varphi := \neg\Diamond^{+}\neg\varphi$).

Theorem 2.2 (Axiom System for $\mathcal{L}_{\Diamond,\Diamond^{+}}$). *The axioms and rules on the first and second block of Table 1 form sound and weakly complete axiom system ($\mathsf{L}_{\Diamond,\Diamond^{+}}$) for formulas of $\mathcal{L}_{\Diamond,\Diamond^{+}}$ w.r.t. relational models [3].*

Second, the propositional dynamic logic (*PDL*; [26]) framework with a converse modality, with operations for building more complex relations (cf. [27]).

Definition 2.4 (Language $\mathcal{L}_{PDL^{\lhd,?}}$). *Formulas φ, ψ and program expressions α, β in $\mathcal{L}_{PDL^{\lhd,?}}$ are given, respectively, by*

$$\varphi, \psi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid \langle\alpha\rangle\,\varphi \qquad \alpha, \beta ::= \rhd \mid \lhd \mid \alpha \cup \beta \mid \alpha\,;\beta \mid \alpha^* \mid ?\varphi,$$

with $p \in \mathsf{P}$. The fragment of $\mathcal{L}_{PDL^{\lhd,?}}$ without ? is called $\mathcal{L}_{PDL^{\lhd}}$. Given (M, w) with $M = \langle W, R, V\rangle$, the semantics of the new modality is defined as

$$(M, w) \Vdash \langle\alpha\rangle\,\varphi \quad \textit{iff} \quad \text{there is } u \in W \text{ such that } R_\alpha wu \text{ and } (M, u) \Vdash \varphi,$$

with the relation R_α defined inductively as

$$R_\rhd{:=}R,\ R_\lhd{:=}\mathfrak{R},\ R_{\alpha\cup\beta}{:=}R_\alpha \cup R_\beta,\ R_{\alpha;\beta}{:=}R_\alpha \circ R_\beta,\ R_{\alpha^*}{:=}(R_\alpha)^*,\ R_{?\varphi}{:=}\mathrm{Id}^M_\varphi,$$

where $\mathfrak{R} := \{(v, u) \mid Ruv\}$, $\mathrm{Id}^M_\varphi := \{(u, u) \mid (M, u) \Vdash \varphi)\}$ and $R^* := R^+ \cup \mathrm{Id}^M_\top$.

Theorem 2.3 (Axiom System for $\mathcal{L}_{PDL^{\lhd,?}}$). *The axioms and rules on the third block of Table 1 form sound and weakly complete axiom system ($\mathsf{L}_{PDL^{\lhd,?}}$) for formulas of $\mathcal{L}_{PDL^{\lhd,?}}$ w.r.t. relational models [26,28,29]. $\mathsf{L}_{PDL^{\lhd}}$ denotes the axiom system for the fragment $\mathcal{L}_{PDL^{\lhd}}$, given by $\mathsf{L}_{PDL^{\lhd,?}}$ minus axiom ?.*

3 Positive Introspection

3.1 General Positive Introspection

When looking for a model operation for representing an action of positive introspection, the first idea is simple: if transitivity makes the positive introspection axiom $\Box\,\varphi \to \Box\,\Box\,\varphi$ valid, then make the accessibility relation transitive.

Definition 3.1 (General Positive Introspection Operation). *Take a relational model $M = \langle W, R, V\rangle$. The* general positive introspection *operation yields the model $M^+ = \langle W, R^+, V\rangle$.*

Definition 3.2 (Language $\mathcal{L}_{\Diamond,+}$). *The language $\mathcal{L}_{\Diamond,+}$ extends $\mathcal{L}_\Diamond$ with $\langle+\rangle$. For its semantic interpretation, let (M, w) be a relational state. Then,*

$$(M, w) \Vdash \langle+\rangle\,\varphi \quad \textit{iff} \quad (M^+, w) \Vdash \varphi.$$

As the model operation is deterministic and its associated modality lacks a precondition, the dual modality $[+]\,\varphi := \neg\,\langle+\rangle\,\neg\varphi$ is equivalent to $\langle+\rangle$ (self-duality).

Some Properties. The operation makes the accessibility relation transitive; then, after applying it, the agent has full positive introspection about any φ.

Proposition 3.1. *Let φ an $\mathcal{L}_{\Diamond,+}$-formula. Then $\Vdash [+]\,(\Box\,\varphi \to \Box\,\Box\,\varphi)$.*

However, the operation does not take the agent from a state in which she knows a given φ without knowing she knows it, $\Box\,\varphi \wedge \neg\,\Box\,\Box\,\varphi$, to a state in which she knows φ and is positively introspective about it, $\Box\,\varphi \wedge \Box\,\Box\,\varphi$.

Fact 3.1. *The formula* $\Box\varphi \rightarrow [+](\Box\varphi \wedge \Box\Box\varphi)$ *is not valid, not even for* φ *propositional.*

Proof. Take φ *as* p*. In the relational state below on the left (reflexivity assumed),* $(M, w_1) \Vdash \Box p \wedge \neg\Box\Box p$*. Nevertheless, after the operation (relational state on the right), she does not know* p *anymore:* $(M^+, w_1) \Vdash \neg\Box p$*, i.e.,* $(M, w_1) \Vdash \langle + \rangle \neg\Box p$*. Thus,* $(M, w_1) \Vdash \Box p \wedge \langle + \rangle \neg\Box p$*.*

Making the accessibility relation transitive might increase the worlds reachable in one step. Thus, although the operation makes the agent's knowledge positively introspective, it does not do it by increasing her knowledge; rather, it discards the knowledge that was non-introspective.

Axiom System. When providing an axiom system for a modality representing a model operation, a useful *DEL* strategy is to provide *reduction axioms*: valid formulas and validity-preserving rules indicating how to translate a formula with occurrences of this model-changing modality (a formula in the 'dynamic' language) into a provably equivalent one without them (a formula in the 'basic' language). Then, while soundness follows from the validity and validity-preserving properties of the new axioms and rules, completeness follows from the completeness of the axiom system for the basic language.

Note how this strategy requires a basic language expressive enough to describe the changes the model operation induces. In this case, $\mathcal{L}_{\Diamond}$ is not enough to deal with the changes the general positive introspection operation brings about: it cannot describe what holds in worlds that can be reached by an *arbitrary* (finite non-zero) number of R-steps (i.e., a single R^+-step). Thus, in order to provide reduction axioms for $\langle + \rangle$, the basic language will be $\mathcal{L}_{\Diamond,\oplus}$.

Table 2. Axioms and rule for the modality $\langle + \rangle$.

$+_p$	$\vdash \langle + \rangle p \leftrightarrow p$	$+_\Diamond$	$\vdash \langle + \rangle \Diamond\varphi \leftrightarrow \oplus \langle + \rangle \varphi$
$+_\neg$	$\vdash \langle + \rangle \neg\varphi \leftrightarrow \neg \langle + \rangle \varphi$	$+_\oplus$	$\vdash \langle + \rangle \oplus\varphi \leftrightarrow \oplus \langle + \rangle \varphi$
$+_\vee$	$\vdash \langle + \rangle (\varphi \vee \psi) \leftrightarrow (\langle + \rangle \varphi \vee \langle + \rangle \psi)$	*Nec*$_+$	If $\vdash \varphi$, then $\vdash [+]\varphi$
SE	If $\vdash \psi_1 \leftrightarrow \psi_2$ then $\vdash \varphi \leftrightarrow \varphi[\psi_2/\psi_1]$, with $\varphi[\psi_2/\psi_1]$ any formula obtained by replacing one or more occurrences of ψ_1 in φ with ψ_2.		

Theorem 3.2 (Axiom System for $\mathcal{L}_{\Diamond,\oplus,+}$). *The axioms and rules of Table 2, together with* $\mathsf{L}_{\Diamond,\oplus}$ *(first and second blocks in Table 1), form a sound and weakly complete axiom system for formulas of* $\mathcal{L}_{\Diamond,\oplus,+}$ *w.r.t. relational models.*

3.2 Particular Positive Introspection

The operation of Definition 3.1 allows the agent to have positive introspection at the cost of losing knowledge. As such, it does not follow the intuition of what an actual positive introspection reasoning step should do. An operation closer to this intuition would take the agent from knowing χ without knowing she knows it, to knowing χ and knowing she knows it. But then the operation should be radically different. If at (M, w) the agent knows a given χ without having full positive introspection about it, then although every world R-reachable from w *in one step* satisfies χ, there is at least one world R-reachable from w (in two or more steps) where χ fails. In order for the agent to have full positive introspection about χ, such $\neg\chi$-worlds should not be R-reachable anymore. In other words, the operation should not add edges, but rather remove them.

Definition 3.3 (U-disconnecting Operation). *Let $M = \langle W, R, V\rangle$ be a relational model; take $U \subseteq W$. The U-*disconnecting *operation yields the model $M_{+U} = \langle W, R', V\rangle$, with $R' := R \setminus (U \times \overline{U})$ (for $\overline{U} := W \setminus U$). Thus, this operation removes edges from worlds on U to worlds not in U.*

When the parameter U of this model operation is given by the truth-set of a formula χ, then the operation can be understood as a *particular positive χ-introspection operation*: it removes edges from worlds satisfying χ to worlds not satisfying χ. The modality for this operation will be introduced in two stages, the first one being the definition of an auxiliary modality.

Definition 3.4 (Language $\mathcal{L}_{\diamond,+'\chi}$). *The language $\mathcal{L}_{\diamond,+'\chi}$ extends $\mathcal{L}_{\diamond}$ with a modality $\langle +'\chi\rangle$ for each formula χ. For the semantic interpretation, let (M, w) be a relational state; use $[\![\chi]\!]^M$ to denote the set of worlds in M in which χ holds.*

$$(M, w) \Vdash \langle +'\chi\rangle\, \varphi \quad \textit{iff} \quad (M_{+[\![\chi]\!]^M}, w) \Vdash \varphi.$$

The operation is deterministic and its modality does not have a precondition, so the modality $[+']$, defined as $[+'\chi]\,\varphi := \neg\,\langle +'\chi\rangle\,\neg\varphi$, is equivalent to $\langle +'\rangle$.

This auxiliary modality allows the language to describe the effects of the positive χ-introspection operation. Still, it differs from what one might expect in one crucial way: its semantic interpretation has no precondition, thus indicating that the epistemic action it represents, an introspective reasoning step, can take place in any situation (even in those in which the agent does not know χ). This issue can be solved in a second stage by introducing another modality:

$$\langle +\chi\rangle\,\varphi := \Box\,\chi \wedge \langle +'\chi\rangle\,\varphi.$$

The reader familiar with *DEL* might notice here a departure from the standard approach: why the auxiliary 'preconditionless' modality $\langle +'\chi\rangle$ instead of defining $\langle +\chi\rangle$ directly with the appropriate precondition? The reason is that the former simplifies the formulation of reduction axioms.

Some Properties. First, here it is a validity characterizing the knowledge of the agent after the operation.

Proposition 3.2. *Let χ and φ be formulas in $\mathcal{L}_{\Diamond,+'\chi}$. The agent can perform a particular positive introspection step for χ after which she will know φ iff she knows both χ and that, after the 'preconditionless' operation, φ will be the case. More precisely,* $\Vdash \langle +\chi \rangle \Box \varphi \leftrightarrow \Box (\chi \wedge [+'\chi]\, \varphi)$.

Proof. *Take any (M, w) with $M = \langle W, R, V \rangle$. From left to right, $(M, w) \Vdash \langle +\chi \rangle \Box \varphi$ yields, by definition, $(M, w) \Vdash \Box \chi$ and $(M, w) \Vdash \langle +'\chi \rangle \Box \varphi$. From the first, Rwu implies $(M, u) \Vdash \chi$; from the latter, $(M_{+\chi}, w) \Vdash \Box \varphi$, i.e. $R'wu$ implies $(M_{+\chi}, u) \Vdash \varphi$. Take now any $u \in W$ with Rwu: then $(M, u) \Vdash \chi$ and hence, from the definition of R' in $M_{+\chi}$, $R'wu$, so $(M_{+\chi}, u) \Vdash \varphi$ and then $(M, u) \Vdash [+'\chi]\, \varphi$. Thus, Rwu implies $(M, u) \Vdash \chi \wedge [+'\chi]\, \varphi$; hence, $(M, w) \Vdash \Box (\chi \wedge [+'\chi]\, \varphi)$. From right to left, $(M, w) \Vdash \Box (\chi \wedge [+'\chi]\, \varphi)$ implies, first, $(M, w) \Vdash \Box \chi$, and second, $(M, w) \Vdash \Box [+'\chi]\, \varphi$, with the latter stating that Rwu implies $(M, u) \Vdash [+'\chi]\, \varphi$. Take now any $u \in W$ with $R'wu$: since $R' \subseteq R$, then Rwu and hence $(M, u) \Vdash [+'\chi]\, \varphi$, i.e., $(M_{+\chi}, u) \Vdash \varphi$. Thus, $(M_{+\chi}, w) \Vdash \Box \varphi$ and so $(M, w) \Vdash \langle +'\chi \rangle \Box \varphi$. But recall the first: $(M, w) \Vdash \Box \chi$. Hence, $(M, w) \Vdash \Box \chi \wedge \langle +'\chi \rangle \Box \varphi$ and thus, by definition, $(M, w) \Vdash \langle +\chi \rangle \Box \varphi$.*

In order to show how this operation behaves as expected, consider the instance of the previous validity with φ replaced by $\Box \chi$:

$$\Vdash \langle +\chi \rangle \Box \Box \chi \leftrightarrow \Box (\chi \wedge [+'\chi] \Box \chi).$$

The formula states what is needed for the agent to have a one-level positive introspection about χ ($\Box \Box \chi$) after the operation. One might expect for the second conjunct inside the scope of $\Box$ in the right-side, $[+'\chi] \Box \chi$, to collapse to $\top$, so the necessary and sufficient condition for the agent to reach one-level positive χ-introspection is for her to know χ. This is not the case.

Fact 3.3 *The formula $\Box \chi \rightarrow [+'\chi] \Box \chi$ is not valid, and so neither is $[+'\chi] \Box \chi$.*

Proof. *Take $\chi := p \wedge \Diamond \Diamond \neg p$ and the relational state below on the left (reflexivity assumed); χ holds at w_1 and w_2 (so $(M, w_1) \Vdash \Box \chi$), but fails at w_3. Thus, the operation yields the relational state on the right, with χ false at w_2; then, $(M_{+\chi}, w_1) \Vdash \neg \Box \chi$ and hence $(M, w_1) \Vdash \Box \chi \wedge \langle +'\chi \rangle \neg \Box \chi$: the agent knows χ, but she will not know it anymore after a positive χ-introspection action.*

Note how $(M, w_1) \Vdash \neg \Box \Box \chi$, so the introspection action is not redundant. Even more, $(M, w_1) \Vdash \Box \chi$, so the state satisfies $\langle +\chi \rangle \neg \Box \chi$ and hence $\neg [+\chi] \Box \chi$.

Fact 3.3 is just one more instance of Moorean phenomena, commonly known as formulas which, after being truthfully announced, become false [30].[1] Here it appears as formulas that are known but, after a particular positive introspection action, are not known anymore. This is because, though the operation does not

[1] The paradigmatic example is $p \wedge \neg \Box p$.

change the atomic valuation, it changes the accessibility relation, thus affecting the agent's knowledge. Nevertheless, the operation behaves as expected when the truth-value of the involved formula χ is preserved by the operation.

Proposition 3.3. *If* $\Vdash \chi \rightarrow [+'\chi]\,\chi$*, then after the operation the agent will have positive introspection about* χ*,* $\Vdash \langle +\chi \rangle \Box\Box\chi \leftrightarrow \Box\chi$*.*

Proof. The '$\rightarrow$' direction follows by replacing φ *with* $\Box\chi$ *in the validity of Proposition 3.2. For '$\leftarrow$', take any* (M, w) *with* $M = \langle W, R, V \rangle$*, and suppose* $(M, w) \Vdash \Box\chi$*; then* Rwu *implies* $(M, u) \Vdash \chi$*. Now take any* $u \in W$ *with* $R'wu$ *and any* $v \in W$ *with* $R'uv$*. Since* $R' \subseteq R$*, then* Rwu *and hence* $(M, u) \Vdash \chi$*. But* $R'uv$ *so the definition of* R' *yields* $(M, v) \Vdash \chi$*. Then, by the assumption,* $(M, v) \Vdash [+'\chi]\,\chi$*, that is,* $(M_{+\chi}, v) \Vdash \chi$*. Since* v *is an arbitrary* R'*-successor of* u*,* $(M_{+\chi}, u) \Vdash \Box\chi$*; since* u *is an arbitrary* R'*-successor of* w*,* $(M_{+\chi}, w) \Vdash \Box\Box\chi$*. Hence,* $(M, w) \Vdash \langle +'\chi \rangle \Box\Box\chi$ *and, as the precondition holds,* $(M, w) \Vdash \langle +\chi \rangle \Box\Box\chi$*.*

The right-to-left direction of this validity, $\Box\chi \rightarrow \langle +\chi \rangle \Box\Box\chi$, is a *dynamic* version of the positive introspection axiom $\Box\chi \rightarrow \Box\Box\chi$: the agent might lack positive introspection for χ, but she can achieve it. Even more: under the same requirement for χ, after the action the agent will have *full* positive χ-introspection.

Proposition 3.4. *If* $\Vdash \chi \rightarrow [+'\chi]\,\chi$*, then after the operation the agent will have* full *positive introspection about* χ*, that is,* $\Vdash \Box\chi \rightarrow \langle +\chi \rangle \Box^n \Box\chi$ *for any* $n \geq 0$*, with* $\Box^0 \varphi := \varphi$ *and* $\Box^{k+1} \varphi := \Box^k \Box\varphi$*.*

Proof. Take a relational state (M, w) *with* $M = \langle W, R, V \rangle$*, and suppose* $(M, w) \Vdash \Box\chi$*; then* Rwu *implies* $(M, u) \Vdash \chi$*. The first step is to show, by induction on* $n \geq 0$*, how* $(R')^{n+1}wu$ *implies* $(M, u) \Vdash \chi$*. The base case is immediate:* $(R')^1wu$ *is* $R'wu$*, and since* $R' \subseteq R$*, then* Rwu *and thus* $(M, u) \Vdash \chi$*. For the inductive case, suppose* $(R')^{n+2}wu$*. Then there is* $v \in W$ *such that* $(R')^{n+1}wv$ *and* $R'vu$*, and hence* $(M, v) \Vdash \chi$ *(from the first and inductive hypothesis) and* Rvu *(from the second and* $R' \subseteq R$*). But* $R'vu$ *so, from the definition of* R'*, it follows that* $(M, u) \Vdash \chi$*.*

For $(M, w) \Vdash \langle +\chi \rangle \Box^n \Box\chi$*, take* $n \geq 0$ *and any* $u \in W$ *with* $(R')^{n+1}wu$*. Then* $(M, u) \Vdash \chi$ *and hence, by the assumption,* $(M, u) \Vdash [+'\chi]\,\chi$*, i.e.,* $(M_{+\chi}, u) \Vdash \chi$*. Thus,* $(R')^{n+1}wu$ *implies* $(M_{+\chi}, u) \Vdash \chi$*, that is,* $(M_{+\chi}, w) \Vdash \Box^n \Box\chi$ *so* $(M, w) \Vdash \langle +'\chi \rangle \Box^n \Box\chi$*. But* $\langle +\chi \rangle$ *'s precondition holds; thus,* $(M, w) \Vdash \langle +\chi \rangle \Box^n \Box\chi$*.*

Thus, if the operation does not affect χ's truth-value, the action's precondition (to know χ) guarantees that the agent will have (knowledge and) full positive introspection about χ. This operation is closer to what comes to mind when one thinks about 'real life': the agent knows χ without noticing it, and thus she only needs to make a further 'introspective' effort to realise it. The operation does not yield positive introspection for all formulas, but it does the work for the particular χ (modulo the extra assumption).

Particular Introspection vs Public Announcement. The reader familiar with *public announcement logic* (*PAL*; [31]) will have noted the similarities between the operation of Definition 3.3 and the public announcement operation: in the new model, former χ-worlds can only reach former χ-worlds. Thus, when the evaluation point is a χ-world, the resulting models are bisimilar. There is, however, an important difference in the precondition of their associated modalities: the one for a χ-announcement requires χ, but the one for a χ-introspection requires $\Box\,\chi$. This is why, while the public announcement modality has 'straightforward' reduction axioms (there is a match between the precondition and the requirement for a world to be reachable after the operation), the introspection modality requires an auxiliary 'preconditionless' version.

Despite the technical similarities, the two operations represent actions of a very different nature: a public announcement is about external communication, but introspection is about self-reflection. It is then remarkable how, in this setting, their representations are so similar. It could be argued that the presented introspection action is too drastic: it removes any 'eventual' (i.e., possibility of having a possibility) uncertainty the agent might have about the given formula. This is indeed the case, but it is the interpretation of edges in relational models what gives no other choice in order to represent this specific epistemic action.

Axiom System. For an axiom system for the modality $\langle +\chi\rangle$, the first step is to provide reduction axioms for its 'preconditionless' counterpart.

Theorem 3.4 (Axiom System for $\mathcal{L}_{\Diamond,+'\chi}$). *The axioms and rules of Table 3, together with the axiom system* $\mathsf{L}_\Diamond$ *(see Table 1), form a sound and strongly complete axiom system for formulas of* $\mathcal{L}_{\Diamond,+'\chi}$ *w.r.t. relational models.*

Table 3. Axioms and rule for the modality $+'\chi$.

$+'\chi_p$	$\vdash \langle +p\rangle \leftrightarrow p$	$+'\chi_\vee$	$\vdash \langle +'\chi\rangle\,(\varphi \vee \psi) \leftrightarrow \langle +'\chi\rangle\,\varphi \vee \langle +'\chi\rangle\,\psi$
$+'\chi_\neg$	$\vdash \langle +'\chi\rangle\,\neg\varphi \leftrightarrow \neg\,\langle +'\chi\rangle\,\varphi$	$+'\chi_\Diamond$	$\vdash \langle +'\chi\rangle\,\Diamond\,\varphi \leftrightarrow (\neg\chi \wedge \Diamond\,\langle +'\chi\rangle\,\varphi) \vee \Diamond\,(\chi \wedge \langle +'\chi\rangle\,\varphi)$
$N_{+'\chi}$	If $\vdash \varphi$, then $\vdash [+'\chi]\,\varphi$		
SE'	If $\vdash \psi_1 \leftrightarrow \psi_2$ then $\vdash \varphi \leftrightarrow \varphi[\psi_2/\psi_1]$, with $\varphi[\psi_2/\psi_1]$ any formula obtained by replacing one or more *non-modality* occurrences of ψ_1 in φ (occurrences of ψ_1 which are *not* inside any 'dynamic' modality $\langle +'\chi\rangle$.) with ψ_2.		

The previous theorem provides a sound and strongly complete axiom system for $\langle +'\chi\rangle$. As $\langle +\chi\rangle$ is just an abbreviation, it requires no axioms; still, its definition makes $\langle +\chi\rangle\,\varphi \leftrightarrow (\Box\,\chi \wedge \langle +'\chi\rangle\,\varphi)$ valid.

4 Negative Introspection

4.1 General Negative Introspection

Analogous to its positive introspection counterpart, the operation for achieving full negative introspection is simply an Euclidean closure operation.

Definition 4.1 (General Negative Introspection Operation). *Take a relational model* $M = \langle W, R, V\rangle$*. The* general negative introspection *operation yields the model* $M^- = \langle W, R^E, V\rangle$ *in which* R^E *is the Euclidean closure of* R*, that is,*

$$R^E := R \cup (\mathfrak{K} \circ (R \cup \mathfrak{K})^* \circ R).$$

Definition 4.2 (Language $\mathcal{L}_{\diamond,-}$). *The language* $\mathcal{L}_{\diamond,-}$ *extends* $\mathcal{L}_{\diamond}$ *with* $\langle - \rangle$ *(*$[-]$ *defined as usual). For its semantic interpretation, let* (M, w) *be a relational state.*

$$(M, w) \Vdash \langle - \rangle\, \varphi \quad \textit{iff} \quad (M^-, w) \Vdash \varphi.$$

Clearly, R^E can be equivalently defined in *PDL* plus the converse operator. This suggests that $\mathcal{L}_{PDL^{\triangleleft}}$ from Definition 2.4 will be useful to provide reduction axioms for this operation. But first, here are some of its properties.

Some Properties. Since R^E is indeed R's Euclidean closure, after the operation the agent has negative introspection.

Lemma 4.1. *For any* $R \subseteq (W \times W)$*, the relation* $R^E := R \cup (\mathfrak{K} \circ (R \cup \mathfrak{K})^* \circ R)$ *is* R*'s Euclidean closure, i.e., the smallest Euclidean relation containing* R*.*[2]

Proposition 4.1. *Let* φ *an* $\mathcal{L}_{\diamond,-}$*-formula. Then,* $\Vdash [-]\,(\neg\,\Box\,\varphi \rightarrow \Box\,\neg\,\Box\,\varphi)$*.*

Even more. Different from the positive introspection case, in the propositional case the operation makes the agent's knowledge negatively introspective in the sense of taking her from $\neg\,\Box\,\varphi \wedge \neg\,\Box\,\neg\,\Box\,\varphi$ to $\neg\,\Box\,\varphi \wedge \Box\,\neg\,\Box\,\varphi$.

Proposition 4.2. *If* φ *is propositional, then* $\Vdash \neg\,\Box\,\varphi \rightarrow [-]\,(\neg\,\Box\,\varphi \wedge \Box\,\neg\,\Box\,\varphi)$*.*

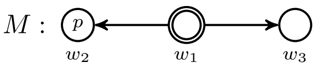
Proof. Take a relational state (M, w) *with* $M = \langle W, R, V\rangle$*, and suppose* $(M, w) \Vdash \neg\,\Box\,\varphi$*; then there is* $u \in W$ *such that* Rwu *and* $(M, u) \Vdash \neg\varphi$*, so* $R^E wu$ *(definition) and* $(M^-, u) \Vdash \neg\varphi$ *(*φ *is propositional). Thus, first,* $(M^-, w) \Vdash \Diamond\neg\varphi$*, i.e.,* $(M^-, w) \Vdash \neg\,\Box\,\varphi$*. Second, for every* $u' \in W$*,* $R^E wu'$ *implies* $R^E u'u$ *(*R^E *is Euclidean), and hence* $(M^-, u') \Vdash \Diamond\neg\varphi$ *so* $(M^-, w) \Vdash \Box\,\Diamond\neg\varphi$*, i.e.,* $(M^-, w) \Vdash \Box\,\neg\,\Box\,\varphi$*. Thus,* $(M, w) \Vdash [-]\,(\neg\,\Box\,\varphi \wedge \Box\,\neg\,\Box\,\varphi)$*.*

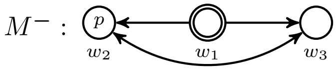
This validity, a *dynamic* version of the negative introspection axiom $\neg\,\Box\,\varphi \rightarrow \Box\,\neg\,\Box\,\varphi$, shows how the operation behaves properly in the propositional case. Still, as expected, it also has Moorean behaviour for arbitrary formulas.

Fact 4.1 *The formula* $\neg\,\Box\,\varphi \rightarrow [-]\,\Box\,\neg\,\Box\,\varphi$ *is not valid.*

Proof. Consider $\varphi := \neg\,\Box\,p$ *and the relational state* (M, w_1) *below on the left (reflexivity assumed). Note how* $(M, w_1) \Vdash \Diamond\,\Box\,p$*, i.e.,* $(M, w_1) \Vdash \neg\,\Box\,(\neg\,\Box\,p)$*. The operation produces the relational state on the right, where* $(M^-, w_1) \Vdash \Diamond\,\Box\,\Diamond\,\neg p$*, i.e.,* $(M^-, w_1) \Vdash \neg\,\Box\,\neg\Box\,(\neg\,\Box\,p)$ *so* $(M, w_1) \Vdash \langle - \rangle\,\neg\,\Box\,\neg\Box\,(\neg\,\Box\,p)$*.*

[2] Proof: http://homepages.cwi.nl/~jve/courses/lai0506/Solutions2.pdf.





Axiom System. $\mathcal{L}_\Diamond$ is not expressive enough to describe the effects of this operation, but the clearly more expressive $\mathcal{L}_{PDL^\lhd}$ is. The Boolean cases are as those in Tables 2 and 3; the modal case is different: in $\langle \alpha \rangle \varphi$, the expression α is an arbitrary program expression, and thus an appropriate translation in each case must be presented. The *program transformer* defined below, a simplification of that of [32] for providing reduction axioms for *PDL*-expressions after action-model operations, captures this: it takes a program α describing a path in the new model M^-, returning its 'matching' path $\mathrm{T}(\alpha)$ in M (Proposition 4.3).

Definition 4.3 (Program Transformer). *A* program transformer T *is a function from program expressions to program expressions defined inductively as*

$$\mathrm{T}(\rhd) := \rhd \cup (\lhd\,;(\rhd \cup \lhd)^*\,;\rhd), \qquad \mathrm{T}(\alpha \cup \beta) := \mathrm{T}(\alpha) \cup \mathrm{T}(\beta), \qquad \mathrm{T}(\alpha^*) := (T(\alpha))^*.$$
$$\mathrm{T}(\lhd) := \lhd \cup (\rhd\,;(\lhd \cup \rhd)^*\,;\lhd), \qquad \mathrm{T}(\alpha\,;\beta) := \mathrm{T}(\alpha)\,;\mathrm{T}(\beta),$$

Proposition 4.3. *Let $M = \langle W, R, V \rangle$ be any relational model, and recall that $M^- = \langle W, R^E, V \rangle$. Then, for every program expression α, $R^E_\alpha = R_{\mathrm{T}(\alpha)}$.*

Proof. The proof is by structural induction on α. For $R^E_\rhd$ ($R^E_\lhd$ is similar),

- $R^E_\rhd = R^E = R \cup (\mathfrak{R} \circ (R \cup \mathfrak{R})^* \circ R) = R_\rhd \cup (R_\lhd \circ (R_\rhd \cup R_\lhd)^* \circ R_\rhd)$
  $= R_\rhd \cup (R_\lhd \circ (R_{\rhd\cup\lhd})^* \circ R_\rhd) = R_\rhd \cup (R_\lhd \circ R_{(\rhd\cup\lhd)^*} \circ R_\rhd)$
  $= R_\rhd \cup R_{\lhd;(\rhd\cup\lhd)^*;\rhd} = R_{\rhd\cup(\lhd;(\rhd\cup\lhd)^*;\rhd)} = R_{\mathrm{T}(\rhd)}$

For the inductive cases (inductive hypothesis: $R^E_\alpha = R_{\mathrm{T}(\alpha)}$, $R^E_\beta = R_{\mathrm{T}(\beta)}$),

- $R^E_{\alpha\cup\beta} = R^E_\alpha \cup R^E_\beta = R_{\mathrm{T}(\alpha)} \cup R_{\mathrm{T}(\beta)} = R_{\mathrm{T}(\alpha)\cup\mathrm{T}(\beta)} = R_{\mathrm{T}(\alpha\cup\beta)}$.
- $R^E_{\alpha;\beta} = R^E_\alpha \circ R^E_\beta = R_{\mathrm{T}(\alpha)} \circ R_{\mathrm{T}(\beta)} = R_{\mathrm{T}(\alpha);\mathrm{T}(\beta)} = R_{\mathrm{T}(\alpha;\beta)}$.
- $R^E_{\alpha^*} = (R^E_\alpha)^* = (R_{\mathrm{T}(\alpha)})^* = R_{(\mathrm{T}(\alpha))^*} = R_{\mathrm{T}(\alpha^*)}$.

Theorem 4.2 (Axiom System for $\mathcal{L}_{PDL^\lhd,-}$). *The axioms and rules of Table 4, together with the axiom system $\mathsf{L}_{PDL^\lhd}$ (see Table 1), form a sound and weakly complete axiom system for formulas of $\mathcal{L}_{PDL^\lhd,-}$ w.r.t. relational models.*

Table 4. Axioms and rule for the modality $\langle - \rangle$.

$-_p$	$\vdash \langle - \rangle p \leftrightarrow p$	$-_{\langle\alpha\rangle}$	$\vdash \langle - \rangle \langle \alpha \rangle \varphi \leftrightarrow \langle \mathrm{T}(\alpha) \rangle \langle - \rangle \varphi$
$-_\neg$	$\vdash \langle - \rangle \neg\varphi \leftrightarrow \neg \langle - \rangle \varphi$	*Nec*$_-$	If $\vdash \varphi$, then $\vdash [-]\,\varphi$
$-_\vee$	$\vdash \langle - \rangle (\varphi \vee \psi) \leftrightarrow (\langle - \rangle \varphi \vee \langle - \rangle \psi)$	*SE*	As in Table 2

4.2 Particular Negative Introspection

Different from the positive introspection counterpart, the operation of Definition 4.1 already behaves as expected: it preserves the agent's (propositional) lack of knowledge while giving her negative introspection (Proposition 4.2). Still, for uniformity, this section explores a negative introspection action over a given χ.

A model operation for achieving full negative introspection about χ should then make sure that all worlds R-reachable from the evaluation point (in zero or more steps, so the original lack of knowledge is preserved and full introspection is reached) can see a $\neg\chi$-world. Assuming that initially the agent does not know χ, this property can be achieved by using a particular instance of the Euclidean closure operation of Definition 4.1 in which the new edges point only to $\neg\chi$-worlds.

Definition 4.4 (U-connecting Operation). *Let $M = \langle W, R, V \rangle$ be a relational model; let $U \subseteq W$. The U-*connecting *operation gives the model $M_{-U} = \langle W, R', V \rangle$, with its indistinguishability relation R' given (with* $\mathrm{Id}_U^M := \{(u, u) \mid u \in U\}$*) by*

$$R' := R \cup (\mathfrak{R} \circ (R \cup \mathfrak{R})^* \circ R \circ \mathrm{Id}_U^M).$$

A modality for a particular full negative introspection can be defined by instantiating U with the set of worlds satisfying $\neg\chi$ in the original model. Here is a 'preconditionless' version.

Definition 4.5 (Language $\mathcal{L}_{\Diamond,-'\chi}$). *The language $\mathcal{L}_{\Diamond,-'\chi}$ extends $\mathcal{L}_{\Diamond}$ with a modality $\langle -'\chi \rangle$ for each formula χ. For the semantic interpretation,*

$$(M, w) \Vdash \langle -'\chi \rangle\, \varphi \quad \textit{iff} \quad (M_{-[\![\neg\chi]\!]^M}, w) \Vdash \varphi.$$

A modality with an appropriate precondition is defined in the obvious way:

$$\langle -\chi \rangle\, \varphi := \neg \Box\, \chi \wedge \langle -'\chi \rangle\, \varphi.$$

Thus, the agent can perform an act of particular negative χ-introspection after which φ is the case, $\langle -\chi \rangle\, \varphi$, iff she does not know χ, $\neg \Box\, \chi$, and after the particular negative χ-introspection operation, φ is the case, $\langle -'\chi \rangle\, \varphi$.

Some Properties. As expected, the analogous of Proposition 3.4 holds.

Proposition 4.4. *If $\Vdash \chi \rightarrow [-'\chi]\, \chi$, then after the operation the agent will have* full *negative introspection about χ, $\Vdash \neg \Box\, \chi \rightarrow \langle -\chi \rangle\, \Box^n \neg \Box\, \chi$ for any $n \geq 0$.*

Proof. *Take a relational state (M, w) with $M = \langle W, R, V \rangle$, and suppose $(M, w) \Vdash \neg \Box\, \chi$; then there is $v \in W$ such that Rwv and $(M, v) \Vdash \neg\chi$, with the latter implying $\mathrm{Id}_{\neg\chi}^M vv$ (by definition) and $(M_{-\chi}, v) \Vdash \neg\chi$ (by the assumption). The first step is to show (by induction on $n \geq 0$) how, in $M_{-\chi}$, any world that can be reached from w in zero or more steps can also reach v, that is, how $(R')^n wu$ implies $R'uv$. The base case ($n = 0$, i.e., $u = w$) is immediate, as the supposition states Rwv, and thus $R'wv$. For the inductive case, suppose $(R')^{n+2} wu$.*

Then there is $u' \in W$ such that $(R')^{n+1}wu'$ and $R'u'u$, and hence (inductive hypothesis) $R'u'v$ and $R'u'u$. It is not hard to see that, in each of the four cases the definition of R' yields, $R'uv$.

Now, in order to prove $(M, w) \Vdash \langle -\chi \rangle \Box^n \neg \Box \chi$, take any $n \geq 0$ and any $u \in W$ such that $(R')^n wu$. Then $R'uv$ and, from $(M_{-\chi}, v) \Vdash \neg\chi$, it follows that $(M_{-\chi}, u) \Vdash \Diamond \neg\chi$, that is, $(M_{-\chi}, w) \Vdash \Box^n \neg \Box \chi$ so $(M, w) \Vdash \langle -'\chi \rangle \Box^n \neg \Box \chi$. But $\langle -\chi \rangle$'s precondition holds; thus, $(M, w) \Vdash \langle -\chi \rangle \Box^n \neg \Box \chi$, as required.

Note how both negative introspection operations add edges. This differs from the positive introspection case: the general operation adds edges, but the particular one needs to remove them.

Axiom System. The basic language will be now $\mathcal{L}_{PDL^{\lhd,?}}$ (Definition 2.4), as the 'test' operator ? is required. Thus, $\mathcal{L}_{PDL^{\lhd,?,-'\chi}}$ extends $\mathcal{L}_{PDL^{\lhd,?}}$ with the 'dynamic' negative χ-introspection modality; for reduction axioms, the program transformer of Definition 4.3 is redefined in the following way.

Definition 4.6 (Program Transformer). *A* χ-program transformer T_χ *is a function from program expressions to program expressions defined as follows*

$$\mathrm{T}_\chi(\rhd) := \rhd \cup (\lhd\, ;\, (\rhd \cup \lhd)^*\, ;\, \rhd\, ;\, ?\neg\chi), \qquad \mathrm{T}_\chi(?\varphi) := ?\langle -'\chi \rangle \varphi.$$
$$\mathrm{T}_\chi(\lhd) := \lhd \cup (?\neg\chi\, ;\, \rhd\, ;\, (\lhd \cup \rhd)^*\, ;\, \lhd)\ ,$$

The remaining cases (for $\cup$, ; and *) are as in Definition 4.3.

Proposition 4.5. *Let $M = \langle W, R, V \rangle$ be any relational model, and recall that $M_{-\chi} = \langle W, R', V \rangle$. Then, for every program expression α, $R'_\alpha = R_{\mathrm{T}_\chi(\alpha)}$.*

Proof. *As in Proposition 4.3, the proof is by structural induction on α. The common cases are similar; for the 'test',*

- $R'_{?\varphi} = \{(w, w) \mid (M_{-\chi}, w) \Vdash \varphi\} = \{(w, w) \mid (M, w) \Vdash \langle -'\chi \rangle \varphi\} = R_{?\langle -'\chi \rangle \varphi} = R_{\mathrm{T}_\chi(?\varphi)}$.

Theorem 4.3 (Axiom System for $\mathcal{L}_{PDL^{\lhd},-}$). *The axioms and rules of Table 5, together with the axiom system $\mathsf{L}_{PDL^{\lhd,?}}$ (see Table 1) form a sound and weakly complete axiom system for formulas of $\mathcal{L}_{PDL^{\lhd,?,-'\chi}}$ w.r.t. relational models.*

Table 5. Axioms and rule for the modality $\langle -'\chi \rangle$.

$-'\chi_p$	$\vdash \langle -'\chi \rangle p \leftrightarrow p$	$-'\chi_{\langle\alpha\rangle}$	$\vdash \langle -'\chi \rangle \langle \alpha \rangle \varphi \leftrightarrow \langle \mathrm{T}_\chi(\alpha) \rangle \langle -'\chi \rangle \varphi$
$-'\chi_\neg$	$\vdash \langle -'\chi \rangle \neg\varphi \leftrightarrow \neg \langle -'\chi \rangle \varphi$	$Nec_{-'\chi}$	If $\vdash \varphi$, then $\vdash [-'\chi]\, \varphi$
$-'\chi_\vee$	$\vdash \langle -'\chi \rangle (\varphi \vee \psi) \leftrightarrow (\langle -'\chi \rangle \varphi \vee \langle -'\chi \rangle \psi)$	*SE"*	Analogous to *SE'* in Table 3

With the language extended and the axiom system introduced, it is possible to provide further validities describing the behaviour of the operation. First, here is how the operation affects the agent's knowledge (now described by $[\rhd]$).

Proposition 4.6. *Suppose* χ *and* φ *are both formulas in* $\mathcal{L}_{PDL^{\lhd,?,-'\chi}}$*; then,* $\Vdash \langle -\chi \rangle [\rhd] \varphi \leftrightarrow (\neg [\rhd] \chi \wedge [\mathrm{T}_\chi(\rhd)] [-'\chi] \varphi)$.

From this and the axiom system, one can obtain a validity characterising the requirements for the agent to have negative introspection about a given χ after the operation: $\Vdash \langle -\chi \rangle [\rhd] \neg [\rhd] \chi \ \leftrightarrow \ (\neg [\rhd] \chi \wedge [\mathrm{T}_\chi(\rhd)] [-'\chi] \neg [\rhd] \chi)$.

5 Conclusion and Further Work

This paper studies positive and negative introspection as epistemic actions that modify the agent's knowledge. In both cases two possibilities are considered: a general operation, and a particular one working relative to a given formula. In all cases, the basic epistemic language is extended with modalities representing the effects of the model operations, presenting their sound and complete axiom systems, and exploring some properties of the new languages.

In the case of positive introspection, the general operation follows a straightforward idea: make the accessibility relation transitive. Yet, this approach boils down to assume that introspection fails not because of what the agent knows about her knowledge, but rather because of what she knows; thus, as a result, non-introspective knowledge is lost, and only the introspective one is preserved. The particular operation has the opposite perspective: to get positive introspection about a given χ, it eliminates edges from χ-worlds to $\neg\chi$-worlds, thus forcing positive introspection on χ while keeping the rest of her knowledge 'as before'. For the negative introspection case, the general operation makes the accessibility relation Euclidean, and thus reaches negative introspection by ensuring the agent knows what she does not know. The particular operation follows the same idea while adding only edges pointing to $\neg\chi$-worlds. Both cases about edge-addition; thus, they have a similar behaviour.

For future work, one direction is to explore operations that raise the agent's introspection in just one level (e.g., from $\Box p \wedge \neg \Box \Box p$ to $\Box p \wedge \Box \Box p \wedge \neg \Box \Box \Box p$). A more interesting project is to investigate similar operations in a multi-agent setting (e.g., public, private versions of these operations), focusing also on operations for reaching common knowledge.

Acknowledgements. This work was partially supported by grant ANPCyT-PICT-2013-2011, STIC-AmSud "Foundations of Graph Structured Data (FoG)", SeCyT-UNC, the Laboratoire International Associé "INFINIS", and the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No. 690974 for the project MIREL: MIning and REasoning with Legal texts.

References

1. Hintikka, J.: Knowledge and Belief. Cornell University Press, Ithaca (1962)
2. Hendricks, V.F.: 8 bridges between formal and mainstream epistemology. Philos. Stud. **128**(1), 1–5 (2006)

3. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning About Knowledge. MIT Press, Cambridge (1995)
4. de Bruin, B.: Explaining Games: The Epistemic Programme in Game Theory. Synthese Library, vol. 346. Springer, Dordrecht (2010). https://doi.org/10.1007/978-1-4020-9906-9
5. van Ditmarsch, H., French, T.: Semantics for knowledge and change of awareness. J. Logic Lang. Inf. **23**(2), 169–195 (2014)
6. Grossi, D., Velázquez-Quesada, F.R.: Syntactic awareness in logical dynamics. Synthese **192**(12), 4071–4105 (2015)
7. van Benthem, J., Pacuit, E.: Dynamic logics of evidence-based beliefs. Stud. Logica **99**(1), 61–92 (2011)
8. Velázquez-Quesada, F.R.: Explicit and implicit knowledge in neighbourhood models. In: Grossi, D., Roy, O., Huang, H. (eds.) LORI 2013. LNCS, vol. 8196, pp. 239–252. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40948-6_19
9. Balbiani, P., Fernández-Duque, D., Lorini, E.: A logical theory of belief dynamics for resource-bounded agents. In: Jonker, C.M., Marsella, S., Thangarajah, J., Tuyls, K. (eds.) Proceedings AAMAS 2016, pp. 644–652. ACM (2016)
10. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer, Dordrecht (2008). https://doi.org/10.1007/978-1-4020-5839-4
11. van Benthem, J.: Logical Dynamics of Information and Interaction. CUP, New York (2011)
12. van Benthem, J.: Dynamic logic for belief revision. J. Appl. Non-Class. Logics **17**(2), 129–155 (2007)
13. van Benthem, J., Liu, F.: Dynamic logic of preference upgrade. J. Appl. Non-Class. Logics **17**(2), 157–182 (2007)
14. Ghosh, S., Velázquez-Quesada, F.R.: Agreeing to agree: reaching unanimity via preference dynamics based on reliable agents. In: Weiss, G., Yolum, P., Bordini, R.H., Elkind, E. (eds.) Proceedings AAMAS 2015, pp. 1491–1499. ACM (2015)
15. Ghosh, S., Velázquez-Quesada, F.R.: A note on reliability-based preference dynamics. In: van der Hoek, W., Holliday, W.H., Wang, W. (eds.) LORI 2015. LNCS, vol. 9394, pp. 129–142. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48561-3_11
16. Pucella, R., Weissman, V.: Reasoning about dynamic policies. In: Walukiewicz, I. (ed.) FoSSaCS 2004. LNCS, vol. 2987, pp. 453–467. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24727-2_32
17. Göller, S.: On the complexity of reasoning about dynamic policies. In: Duparc, J., Henzinger, T.A. (eds.) CSL 2007. LNCS, vol. 4646, pp. 358–373. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74915-8_28
18. Benthem, J.: An essay on sabotage and obstruction. In: Hutter, D., Stephan, W. (eds.) Mechanizing Mathematical Reasoning. LNCS (LNAI), vol. 2605, pp. 268–276. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-32254-2_16
19. Areces, C., Fervari, R., Hoffmann, G.: Moving arrows and four model checking results. In: Ong, L., de Queiroz, R. (eds.) WoLLIC 2012. LNCS, vol. 7456, pp. 142–153. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32621-9_11
20. Areces, C., Fervari, R., Hoffmann, G.: Swap logic. Logic J. IGPL **22**(2), 309–332 (2014)
21. Fervari, R.: Relation-Changing Modal Logics. Ph.D. thesis, Facultad de Matemática, Astronomía y Física, Universidad Nacional de Córdoba, Argentina (2014)

22. Areces, C., Fervari, R., Hoffmann, G.: Relation-changing modal operators. Logic J. IGPL **23**(4), 601–627 (2015)
23. Kooi, B., Renne, B.: Arrow update logic. Rev. Symb. Logic **4**, 536–559 (2011)
24. Chellas, B.F.: Modal Logic: An Introduction. Cambridge University Press, Cambridge (1980)
25. Blackburn, P., de Rijke, M., Venema, Y.: Modal logic. CUP, New York (2001)
26. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press, Cambridge (2000)
27. van Eijck, J., Wang, Y.: Propositional dynamic logic as a logic of belief revision. In: Hodges, W., de Queiroz, R. (eds.) WoLLIC 2008. LNCS (LNAI), vol. 5110, pp. 136–148. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69937-8_13
28. Prior, A.N.: Time and Modality. Clarendon Press, Oxford (1957)
29. Parikh, R.: The completeness of propositional dynamic logic. In: Winkowski, J. (ed.) MFCS 1978. LNCS, vol. 64, pp. 403–415. Springer, Heidelberg (1978). https://doi.org/10.1007/3-540-08921-7_88
30. Holliday, W., Icard, T.: Moorean phenomena in epistemic logic. In: Beklemishev, L., Goranko, V., Shehtman, V. (eds.) Advances in Modal Logic, College Publications, pp. 178–199 (2010)
31. Plaza, J.A.: Logics of public communications. In: Emrich, M.L., Pfeifer, M.S., Hadzikadic, M., Ras, Z.W. (eds.) Proceedings 4th International Symposium on Methodologies for Intelligent Systems, Oak Ridge National Laboratory, pp. 201–216 (1989)
32. van Benthem, J., van Eijck, J., Kooi, B.: Logics of communication and change. Inf. Comput. **204**(11), 1620–1662 (2006)

Logics for Actor Networks: A Case Study in Constrained Hybridization

José Fiadeiro[1], Ionuţ Ţuţu[1(✉)], Antónia Lopes[2], and Dusko Pavlovic[3]

[1] Department of Computer Science,
Royal Holloway University of London, Egham, UK
jose.fiadeiro@rhul.ac.uk, ittutu@gmail.com
[2] Department of Informatics, Faculty of Sciences,
University of Lisbon, Lisbon, Portugal
malopes@ciencias.ulisboa.pt
[3] Department of Information and Computer Sciences,
University of Hawaii, Honolulu, USA
dusko@hawaii.edu

Abstract. Actor Networks are a modeling framework for cyber-physical system protocols based on Latour's actor-network theory that addresses the way we now create and exploit the power of computational networks. We advance a logic for modeling and reasoning about such actor networks, which is obtained through a two-stage constrained hybridization process. The first stage results in a logic that captures the structure of actor networks and the way knowledge flows across them; the second addresses the dynamic aspects of actor networks, that is the way they can evolve as a result of the interactions that occur within them. For each of these stages, we develop a sound and complete proof system.

1 Introduction

Over the past few years, there has been a renewed interest in modal logics for computer science through the family of the so-called *hybrid logics* (see [1] for a comprehensive overview). The development of hybrid logics originated in Arthur Prior's work in the 1960s [2]. In their most basic form, these are logics obtained by enriching ordinary modal logics with nominals—symbols that name individual states (possible worlds) in Kripke models—and dedicated satisfaction operators $@_a$ that enable a change of perspective from the current state to the one that corresponds to the nominal a. A significant body of research exists around this class of logics, among which [3–6] are recent publications.

In this paper, we are specifically interested in the present-day applications of hybrid logics to the specification and verification of reconfigurable systems [7]. In a nutshell, the idea is that system configurations (and the functionalities associated with them) can be regarded as local models of a Kripke structure, and that they can change simply by switching from one mode of operation to another via an accessibility relation. The key advancement here lies in the fact that the characteristic features of hybrid logic can be developed, through a process known as

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 98–114, 2018.
https://doi.org/10.1007/978-3-319-73579-5_7

hybridization [8], on top of an arbitrary logic used for expressing configuration-specific requirements. This means that, depending on the base logic, configurations can be captured, for example, as algebras, relational structures or, when the hybridization process is iterated, even as Kripke models.

Actor Networks. Our interest in this area results from a new modeling framework for cyber-physical system protocols proposed in [9] around the concept of Actor Network (or ANt), which addresses networks whose components are no longer limited to programs but can also include humans or physical artifacts as actors. ANts should therefore be understood in the wider sense of Latour's actor-network theory [10]: actors are cyber-physical entities that have shared agency—from people, to objects, and to locations; they interact through so-called channels that account, for example, for observations that an actor may make of another, of control that an actor may exert on another, or of movement of an actor inside another (say, a person to a location). Interaction, rather than computation, has become the critical source of complexity, thus giving rise to new challenges in ensuring the reliability of the systems that are now operating in cyberspace.

Contributions. The ordinary hybridization process yields logical systems that are suitable for dealing with either the static/structural aspects or with the dynamic aspects of actor networks. From a static perspective, hybrid logics can be naturally used, for example, to give faithful descriptions of the shapes of networks, of the (states of the) actors involved, or of the channels through which interactions can take place. However, accommodating at the same time both the structure and the behaviour of ANts raises some difficulties because these two aspects require distinct, and possibly conflicting, interpretations of the hybrid features. For example, from a structural point of view, modalities denote channels, whereas from a behavioural point of view they stand for graphs of interactions between actors. That is, the challenge raised by ANts lies precisely in capturing the way in which the structure of such networks evolves. This leads us to propose a two-layered hybridization process, where the first level corresponds to the structure, and the second to the dynamics of actor networks.

The paper consists of two main technical sections. In Sect. 2 we introduce the underlying model theory of actor networks. We start by formalizing the main static concepts: actors, interaction channels, the knowledge that actors may have and the way it may be acquired across certain channels, and the placement of some actors relative to other actors. Then, we formalize the key notion of interaction and the way an interaction can change the state of a network.

Section 3 is devoted to the logics through which we can specify and reason about the states of an actor network and about the state transitions associated with interactions. These logics are obtained through a sequence of two processes of constrained hybridization, meaning that (a) the models of the hybrid logics implicitly satisfy additional semantic constraints, and (b) we actually operate across three logical levels—each level captures a different aspect of actor networks (knowledge, structure, or dynamics), and each is defined as an exogenous enrichment (with new hybrid features) of the previous level.

Notational Conventions. Most of the structures we deal with in this paper are presented as tuples—whose components, in turn, may also be tuple-based structures—that satisfy certain cohesion properties. To keep the notations as simple as possible, and avoid spelling out all the components of a given structure, we make use of subscripts. For example, we may denote the set $\mathcal{N}$ of nodes of a graph $\mathcal{G}$ by $\mathcal{N}_{\mathcal{G}}$, the underlying graph $\mathcal{G}$ of an ANt schema $\mathcal{A}$ by $\mathcal{G}_{\mathcal{A}}$, and the domain $\mathcal{D}$ of an actor network ν by $\mathcal{D}_{\nu}$. When there is no risk of confusion, we overload this notation in order to refer to the hereditary components of a structure. That is, we may denote, for example, the set $\mathcal{N}$ of nodes of the underlying graph of an ANt schema $\mathcal{A}$ by $\mathcal{N}_{\mathcal{A}}$—even if $\mathcal{N}$ is not a direct component of $\mathcal{A}$.

2 Actor Networks

2.1 Schemas

Definition 1 (Schema). An *ANt schema* $\mathcal{A}$ consists of:

- a finite directed graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{C}, \delta, \rho \rangle$, where $\mathcal{N}$ is a non-empty set (of nodes, called *actors*), $\mathcal{C}$ is a set (of edges, called *channels*), and δ and ρ are maps $\mathcal{C} \to \mathcal{N}$ that give the domain (origin) and codomain (target) of every channel;
- a partially ordered set $\mathcal{T}$ (of *channel types*, with a subtype relationship);
- a function $\tau \colon \mathcal{C} \to 2^{\mathcal{T}}$ that assigns a non-empty upper set[1] of channel types to every channel, such that for every $n, n' \in \mathcal{N}$ and $\kappa \in \mathcal{T}$ there is at most one channel $c \in \mathcal{C}$ such that $\delta(c) = n$, $\rho(c) = n'$, and $\kappa \in \tau(c)$; and
- a set $\mathcal{P}$ (of *propositional symbols*).

The nodes of an ANt schema represent *actors* executing a given protocol and edges represent *channels* that link together those actors. Channels are typed in order to account for different modes of relationship between actors. The propositional symbols are used to represent knowledge that is held by the different actors, including data. Pieces of data (or knowledge) have by themselves no agency in the context of the protocol, otherwise they would be actors; for example, in a given protocol, money could be data but, in another protocol, bank notes could be actors, in the sense that they can change hands, be lost, and so on. Knowledge/data can be transmitted across channels as appropriate.

Example 2. Consider the ANt schema Elevator whose graph and typing function are depicted in Fig. 1. The nodes *F0* and *F1* correspond to the ground and first floor of a building, and *E* to the elevator proper, which we often refer to as

[1] We recall that an *upper set* of $\mathcal{T}$ is an upward closed subset U of $\mathcal{T}$. That is, U contains all channel types $\kappa' \in \mathcal{T}$ for which there exists $\kappa \in U$ such that $\kappa \leq \kappa'$.

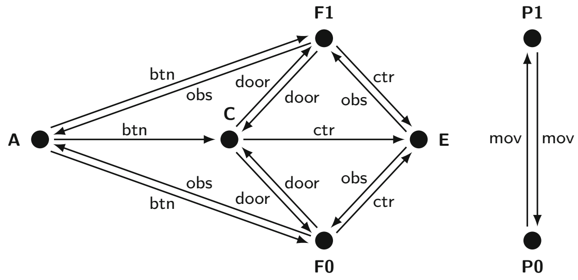


Fig. 1. The graph and typing function of the ANt schema Elevator

Elevator unless it is ambiguous. The node C corresponds to the elevator cabin, which we often refer to as Cabin, and *P0* and *P1* correspond to the two platforms where the cabin can be—*P0* for the ground floor and *P1* for the first floor. The node A represents a user of the elevator, which we refer to as Alice.

Elevator has a number of channels of different types:

- The channel type *mov* captures the movement of one actor inside another. The two channels of type *mov* that connect *P0* and *P1* allow the cabin to move between the two platforms (up or down).
- The type *door* is a subtype of *mov*. The two channels of type *door* connect *F0* and *F1* to C in order to allow users to enter or exit the cabin from or to the floor. Although these two channels are also of type *mov*, for readability we tend to depict only the minimal types when representing ANt schemas.
- The channel type *obs* captures observations that an actor may make of another. The channels of type *obs* that connect E to *F0* and *F1* account for observations of the state of Elevator at either floor, while those that connect *F0* and *F1* to A account for observations that Alice makes of either floor.
- The channel type *ctr* captures forms of control that one actor may exert on another. The two channels of type *ctr* that connect *F0* to E and *F1* to E are for transmitting requests from floors to Elevator, and the channel that connects C to E is for transmitting requests from Cabin to Elevator.
- The channel subtype *btn* of *ctr* captures the special case of control achieved through a button. The three channels of type *btn* that connect A to *F0*, *F1*, and C account for the buttons that Alice can press at either floor or at Cabin.

Last but not least, the ANt schema Elevator has two propositional symbols, C_at_P0 and C_at_P1. These are used to capture knowledge of where Cabin is.

2.2 States

A structure for an ANt schema $\mathcal{A}$ consists of a subgraph of $\mathcal{G}_{\mathcal{A}}$ together with a forest (or *placement graph*, as in [11]) over its nodes that captures 'location'.

Definition 3 (Structure). A *structure* for a schema $\mathcal{A}$ is a pair $\langle \mathcal{H}, \mathcal{F} \rangle$ where:

- $\mathcal{H}$ is a subgraph of $\mathcal{G}_{\mathcal{A}}$, and
- $\mathcal{F}$ is a forest over $\mathcal{N}_{\mathcal{H}}$, meaning that every node n has either none or a unique parent, denoted $\mathcal{F}(n)$. Nodes without a parent are called *roots*.

We say that $\langle \mathcal{H}_1, \mathcal{F}_1 \rangle$ is a *substructure* of (or is included in) $\langle \mathcal{H}_2, \mathcal{F}_2 \rangle$ if:

- $\mathcal{H}_1$ is a subgraph of $\mathcal{H}_2$, and
- for every $n \in \mathcal{N}_{\mathcal{H}_1}$ such that $\mathcal{F}_1(n)$ is defined, $\mathcal{F}_2(n)$ is also defined and equal to $\mathcal{F}_1(n)$—that is, the hierarchy is strictly preserved.

The notion of substructure defines a partial order, which we denote by $\preceq$.

Example 4. An ANt structure for Elevator is depicted in Fig. 2. The forest, on the right, places the two platforms inside Elevator, the Cabin inside the platform of the first floor, and Alice at the ground floor; both floors are outside Elevator.

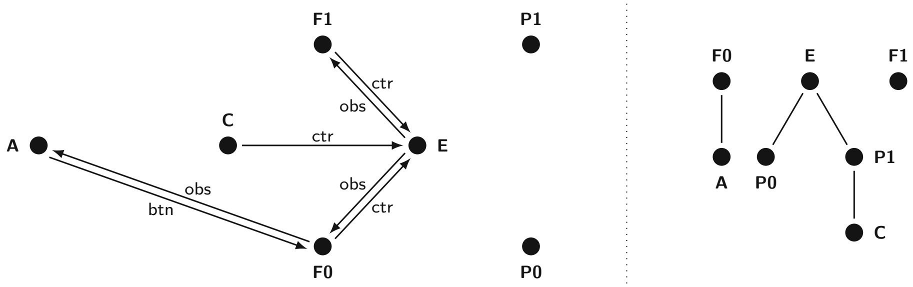


Fig. 2. An ANt structure for Elevator: the graph on the left and the forest on the right

The graph, on the left, indicates the channels that are available: for example, the channel that corresponds to the button that Alice can press to call the elevator, and the one that connects *F0* to *E* and allows the floor to transmit requests to Elevator. Notice that, for readability, we always include the channel types in figures, even though they are not formally part of ANt structures.

To better visualize ANt structures, we combine the graph and the forest components through the nesting of nodes in the graph. This can be seen in Fig. 3, where three ANt structures are presented.

A state is an ANt structure together with a *valuation* of the propositional symbols, which assigns to each node and propositional symbol the truth value of the propositional symbol at that node. We work with a three-valued Łukasiewicz logic, i.e., propositions may have values $\oplus$ (true), $\ominus$ (false), or $\boxplus$ (undefined).

Definition 5 (State). A *state* of an ANt schema $\mathcal{A}$ consists of a structure $\mathcal{S}$ for $\mathcal{A}$ such that $\mathcal{N}_{\mathcal{S}} = \mathcal{N}_{\mathcal{A}}$ (i.e., the structure has all the nodes of the schema) together with, for each node n, a valuation $\mathcal{V}_n : \mathcal{P}_{\mathcal{A}} \to \{\oplus, \ominus, \boxplus\}$.

We denote the set of states of an ANt schema $\mathcal{A}$ by $\mathbb{S}_{\mathcal{A}}$ and, following our notational convention, the structure underlying a state σ by $\mathcal{S}_{\sigma}$.

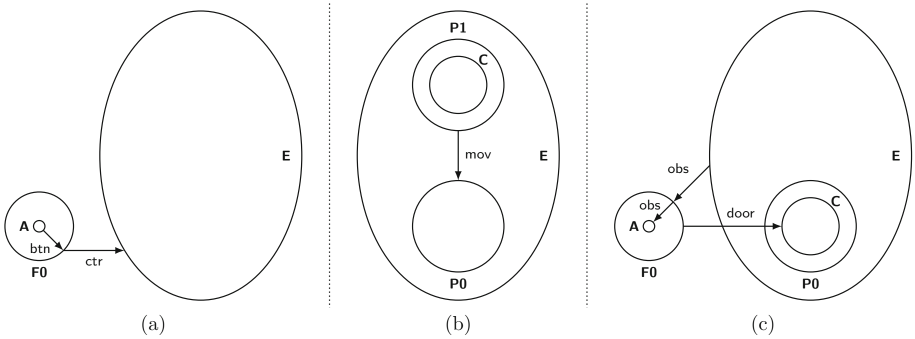


Fig. 3. Interactions of Elevator: (a) callElevator0, (b) moveCabin0, and (c) enterCabin0

Example 6. As an example, we define the state $\mathsf{elevator}_0$ whose underlying structure is shown in the top-left part of Fig. 4 and whose valuation is given by:

$$\mathcal{V}_E(\mathsf{C\_at\_P0}) = \mathcal{V}_{F0}(\mathsf{C\_at\_P0}) = \mathcal{V}_{F1}(\mathsf{C\_at\_P0}) = \mathcal{V}_A(\mathsf{C\_at\_P0}) = \ominus,$$
$$\mathcal{V}_E(\mathsf{C\_at\_P1}) = \mathcal{V}_{F0}(\mathsf{C\_at\_P1}) = \mathcal{V}_{F1}(\mathsf{C\_at\_P1}) = \mathcal{V}_A(\mathsf{C\_at\_P1}) = \oplus, \text{ and}$$
$$\mathcal{V}_n(\mathsf{C\_at\_P0}) = \mathcal{V}_n(\mathsf{C\_at\_P1}) = \oplus \text{ for all other nodes.}$$

That is, the actors/nodes *E*, *F0*, *F1* and *A* all know that Cabin is at the platform *P1* (and that it is not at *P0*); no other node knows where Cabin is.

2.3 Interactions

Channels provide the means for actors to interact with each other. The interactions through which actors change protocol states can be more complex (in the sense that they can involve many actors and channels) and are therefore defined as ANt structures: given an interaction, its nodes are the actors of the ANt schema that are involved in the interaction, and its channels are those through which those actors interact with each other.

Definition 7 (Interaction). An interaction in the context of an ANt schema $\mathcal{A}$ is a structure for $\mathcal{A}$. We denote by $\mathbb{I}_{\mathcal{A}}$ the set of all interactions of $\mathcal{A}$.

Example 8. Figure 3 depicts three interactions for the ANt schema Elevator:

(a) callElevator0: Alice is at the ground floor and presses the button to call the elevator; the request is transmitted to Elevator through a *ctr* channel.
(b) moveCabin0: Cabin is at the first floor and the channel through which it can move to the ground floor is available.
(c) enterCabin0: Alice is at the ground floor and observes the position of the cabin through the two channels of type *obs*; the channel of type *door* that connects *F0* to *C* is available for Alice to enter the cabin.

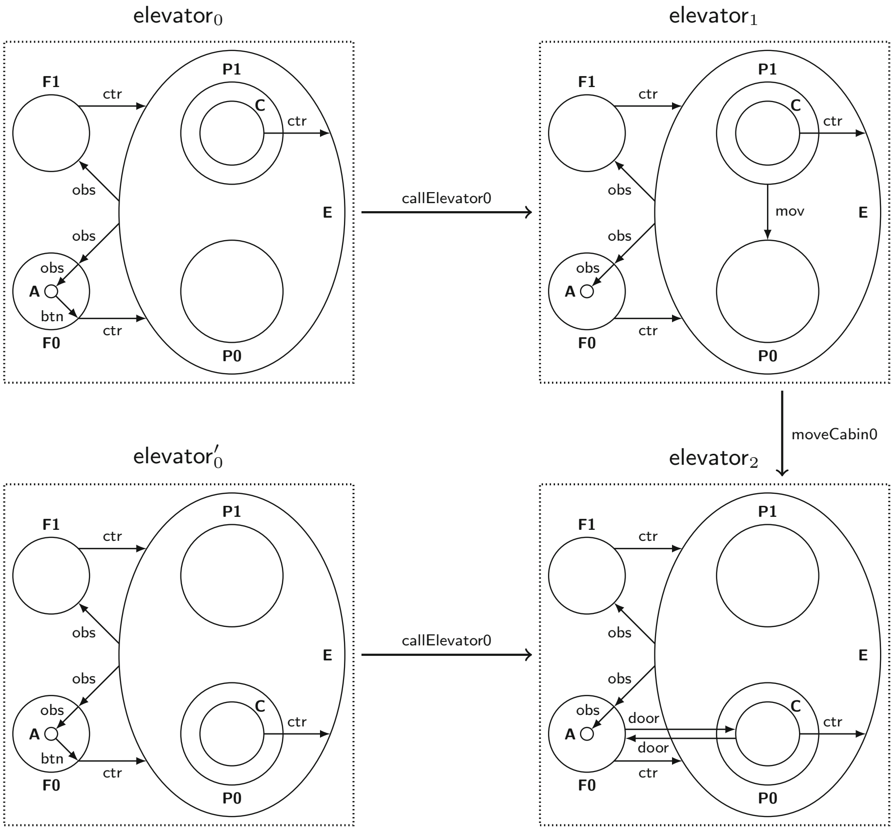


Fig. 4. Transitions performed by the interactions callElevator0 and moveCabin0

2.4 Networks

Protocols are formalized as actor networks. An actor network consists of (a) an ANt schema, which contains all the actors and the channels that connect them; (b) a set of possible worlds—each being associated with a so-called admissible state—including a subset of designated initial worlds; (c) a set of all possible interactions through which the actor network can evolve; and (d) for every such interaction, a transition relation on the set of worlds. Formally,

Definition 9 (Actor network). An actor network ν consists of:

– an ANt schema $\mathcal{A}$,
– a *domain* (set of *worlds*) $\mathcal{D}$ together with a labeling function $\varsigma\colon \mathcal{D} \to \mathbb{S}_{\mathcal{A}}$,
– a non-empty subset $\mathcal{D}_0 \subseteq \mathcal{D}$ of *initial worlds* (whose labels are *initial states*),
– a set $\mathcal{I} \subseteq \mathbb{I}_{\mathcal{A}}$ of interactions for $\mathcal{A}$,
– a *transition relation* $(\longrightarrow) \subseteq \mathcal{D} \times \mathcal{I} \times \mathcal{D}$ such that, for each interaction $\iota \in \mathcal{I}$, $w \xrightarrow{\iota} w'$ implies $\iota \preceq \mathcal{S}_{\varsigma(w)}$ (interactions are substructures of the source states).

We say that a state of $\mathcal{A}$ is *admissible* for ν if it corresponds to one of its worlds; we denote by $\mathbb{S}_\nu$ the image of $\mathcal{D}$ under ς (i.e., the set of admissible states of ν).

Therefore, an actor network can be regarded as a labeled transition system over a set of states of the schema, transitions being labeled with interactions.

Example 10. An actor network ν_{Elevator} with Elevator as its schema could have, for example, $\mathsf{elevator}_0$ (labeled with the state defined in Example 4) as one of its initial worlds, and the worlds and transitions presented in Fig. 4, among others. Note that the valuations are not included in these diagrams; an axiomatic presentation of the valuations is discussed in Sect. 3.

The 'horizontal' transitions in Fig. 4 are performed by the interaction callElevator0 (cf. Fig. 3(a)). The one at the top starts at $\mathsf{elevator}_0$. Although several actors and channels are present in $\mathsf{elevator}_0$, the interaction callElevator0 indicates that the actors that are active in the transition are Alice, Elevator and *F0* (the ground floor), and that the active channels are those that connect *A* to *F0* and *F0* to *E*. That is to say, Alice presses the button at *F0* and the request is transmitted to Elevator. The transition to $\mathsf{elevator}_1$ activates the channel of type *mov* that connects *P1* to *P0* through which Elevator can respond to the request (i.e., move the cabin), and closes the channel of type *btn* from *A* to *F0*, i.e., Alice is no longer able to call the elevator.

The other transition (at the bottom) performed by callElevator0 starts in a different world, $\mathsf{elevator}'_0$, where Cabin is in *P0*. It opens the two channels of type *door* between *F0* and *C* that allow users to enter or exit the cabin.

The 'vertical' transition from $\mathsf{elevator}_1$ to $\mathsf{elevator}_2$ is performed by the interaction moveCabin0 (cf. Fig. 3(b)). As indicated by the interaction, this computation is local to *P0*, *P1*, *F0*, *F1*, *E*, and *C*. The transition moves the cabin from *P1* to *P0*, closes the channel of type *mov* that connects the two platforms and—just like the transition between $\mathsf{elevator}'_0$ and $\mathsf{elevator}_2$—opens the two channels of type *door* that allow users to enter or exit the cabin.

3 Logics for ANts

3.1 The Base Logic

The logics through which we can specify and reason about actor networks are obtained through an iterated process of constrained hybridization. At the base of this construction is the three-valued propositional Łukasiewicz logic, which we recall below. A signature for the Łukasiewicz logic is given by a set $\mathcal{P}$ of *atomic propositions* (the propositional symbols defined by an ANt schema).

Definition 11 (Syntax). The set $\mathsf{Ł}(\mathcal{P})$ of sentences of the base logic is the least set that includes $\mathcal{P}$ and is closed under negation ($\overline{p}$) and implication ($p \supset q$).

Definition 12 (Semantics). Base-logic sentences are interpreted over functions $[\![\_]\!] \colon \mathcal{P} \to \{\oplus, \ominus, \boxplus\}$ as follows: The negation maps $\oplus$ to $\ominus$ and vice versa, and leaves $\boxplus$ unchanged. The implication $p \supset q$ evaluates to $\ominus$ if $p = \oplus$ and $q = \ominus$, to $\boxplus$ if $p = \oplus$ and $q = \boxplus$ or $p = \boxplus$ and $q = \ominus$, and to $\oplus$ in all other cases. We say that a proposition p is valid if $[\![p]\!] = \oplus$ for all valuations $[\![\_]\!]$.

The following modalities, which return Boolean values, are useful:

$\mathsf{M}\,p \triangleq (\overline{p} \supset p)$	p is possibly true—i.e., it has value $\oplus$ or $\boxplus$
$\mathsf{L}\,p \triangleq \overline{\mathsf{M}\,\overline{p}}$	p is necessarily true—i.e., it has value $\oplus$
$\mathsf{N}\,p \triangleq \overline{\mathsf{M}\,p} = \mathsf{L}\,\overline{p}$	p is necessarily false—i.e., it has value $\ominus$
$\mathsf{I}\,p \triangleq \overline{\mathsf{M}\,p \supset \mathsf{L}\,p}$	p is unknown—i.e., it has value $\boxplus$

3.2 The State Logic

The logic through which we can specify and reason about the states of an ANt is a constrained hybridization of $\text{Ł}(\mathcal{P})$. Therefore, there are two main ingredients to consider here. Firstly, an *ANt-schema signature*, denoted Σ in what follows, which consists of a set $\mathcal{P}$ of propositional symbols (i.e., a signature of the base Łukasiewicz logic), a countably infinite set $\mathcal{N}om$ of nominals that includes a set $\mathcal{N}$ of actor names, and a set $\mathcal{T}$ of channel types (regarded as modalities). Secondly, a partial order on $\mathcal{T}$ and an edge-labeled directed finite graph $\mathcal{G}$ with components $\mathcal{N}$, $\mathcal{C}$, δ, ρ, and τ as in Definition 1. These provide the constraints that we impose on the models of the hybrid logic.

Definition 13 (Syntax). The syntax of the state logic is given by the grammar

$$\phi ::= p \mid a \mid \neg\,\phi \mid \phi \to \phi \mid \langle\kappa\rangle\,\phi \mid \langle\pi\rangle\,\phi \mid @_a\,\phi \mid \exists b\,\phi$$

where $p \in \text{Ł}(\mathcal{P})$, $a \in \mathcal{N}om$, $b \in \mathcal{N}om \setminus \mathcal{N}$, $\kappa \in \mathcal{T}$, and π is a distinguished and new *parent* modality. We denote this set of sentences by $\mathsf{State}(\Sigma)$.

In the most general setting, hybrid sentences are evaluated over unconstrained Kripke models, that is over triples $\langle W, R, V\rangle$, where W is a set of nodes or possible worlds, R is a family of accessibility relations $R_\lambda \subseteq W \times W$, indexed by modalities λ, and V is a family of interpretations of the symbols from $\mathcal{P}$ indexed by possible worlds. The semantics of hybrid logics often includes additional constraints; for example, in the S4 variant of hybrid propositional logic, the accessibility relations are reflexive and transitive, and in the S5 variant they are reflexive and Euclidean. The constraints that we consider for the state logic follow from the underlying graph structure of the ANt schema used:

- There is a one-to-one correspondence between actors and possible worlds. For notational convenience, we do not distinguish possible worlds from actors.
- The accessibility relations conform to the channels and the channel types of the schema: for each channel type κ, R_κ consists of those pairs of nodes (n, n') that are connected through a channel of type κ.
- The interpretation of the parent modality π is functional and acyclic.

In other words, the constrained models of the hybridization of $\text{Ł}(\mathcal{P})$ that we consider here are states of the actor-network schema $\mathcal{A} = \langle\mathcal{G}, \mathcal{T}, \tau, \mathcal{P}\rangle$.

Definition 14 (Semantics). Given a state $\sigma = \langle \mathcal{S}, \mathcal{V} \rangle$ of $\mathcal{A}$, an *assignment* is a map $\alpha : \mathcal{N}om \to \mathcal{N}_\mathcal{S}$ whose restriction to the set $\mathcal{N}$ of actor names is the identity.[2] The satisfaction relation between ANt states and state-logic sentences is parameterized by assignments α and by actors n (i.e., by nodes of $\mathcal{S}$):

- $\sigma, \alpha, n \vDash a$ iff $\alpha(a) = n$;
- $\sigma, \alpha, n \vDash p$ iff $[\![p]\!]_n = \oplus$ where $[\![\_]\!]_n$ is the valuation defined by $\mathcal{V}_n$ over $\mathcal{P}$;
- $\sigma, \alpha, n \vDash \neg\, \phi$ iff $\sigma, \alpha, n \nvDash \phi$;
- $\sigma, \alpha, n \vDash \phi_1 \to \phi_2$ iff $\sigma, \alpha, n \vDash \phi_1$ implies $\sigma, \alpha, n \vDash \phi_2$;
- $\sigma, \alpha, n \vDash \langle \kappa \rangle\, \phi$ iff there is $c \in \mathcal{C}$ such that $\delta(c) = n$, $\kappa \in \tau(c)$, and $\sigma, \alpha, \rho(c) \vDash \phi$;
- $\sigma, \alpha, n \vDash \langle \pi \rangle\, \phi$ iff $\mathcal{F}(n)$ is defined and $\sigma, \alpha, \mathcal{F}(n) \vDash \phi$;
- $\sigma, \alpha, n \vDash @_a\, \phi$ iff $\sigma, \alpha, \alpha(a) \vDash \phi$;
- $\sigma, \alpha, n \vDash \exists b\, \phi$ iff $\sigma, \alpha', n \vDash \phi$ for some α' that agrees with α on $\mathcal{N}om \setminus \{b\}$.

We also define validity of a sentence at a state to mean that it is satisfied for every assignment at every node, validity of a sentence at an ANt structure to mean that it is satisfied at every state for it, and absolute validity of a sentence (at a schema) to mean that it is valid at every state of the schema:

- $\sigma \vDash \phi$ iff $\sigma, \alpha, n \vDash \phi$ for all assignments $\alpha : \mathcal{N}om \to \mathcal{N}$ and all $n \in \mathcal{N}$;
- $\mathcal{S} \vDash \phi$ iff $\sigma \vDash \phi$ for all states σ such that $\mathcal{S} \preceq \mathcal{S}_\sigma$;
- $\mathcal{A} \vDash \phi$, or simply $\vDash \phi$, iff $\sigma \vDash \phi$ for all $\sigma \in \mathbb{S}_\mathcal{A}$.

The validity relations extend to sets of sentences to mean that every sentence in the set is valid. Given a set Φ of sentences, we denote by $\mathbb{S}_\Phi$ the set of states over which all the sentences in Φ are valid.

We use the usual propositional connectives for conjunction ($\wedge$) and disjunction ($\vee$), as well as the dual modal operators ($[\_]$) and quantifier ($\forall$) given by:

- $[\kappa]\, \phi \triangleq \neg\, \langle \kappa \rangle \neg\, \phi$
  That is, $\sigma, \alpha, n \vDash [\kappa]\, \phi$ iff $\sigma, \alpha, \rho(c) \vDash \phi$ for all $c \in \mathcal{C}$ with $\delta(c) = n$ & $\kappa \in \tau(c)$.
- $[\pi]\, \phi \triangleq \neg\, \langle \pi \rangle \neg\, \phi$
  That is, $\sigma, \alpha, n \vDash [\pi]\, \phi$ iff $\sigma, \alpha, \mathcal{F}(n) \vDash \phi$ if $\mathcal{F}(n)$ is defined.
- $\forall b\, \phi \triangleq \neg\, \exists b \neg\, \phi$
  That is, $\sigma, \alpha, n \vDash \forall b\, \phi$ iff $\sigma, \alpha', n \vDash \phi$ for all α' that agree with α on $\mathcal{N}om \setminus \{b\}$.

Notice that the symbols used for the negation ($\bar{\ }$) and implication ($\supset$) in the base logic are different from those of the state logic ($\neg$ and $\to$, respectively) to mark the difference between the two levels.

Example 15. The sentences presented below are properties of (valid at) the state $\mathsf{elevator}_0$ from Example 6, of its underlying structure $\mathcal{S}_{\mathsf{elevator}_0}$, or of the ANt schema itself. Naturally, any property of $\mathcal{S}_{\mathsf{elevator}_0}$ is also a property of $\mathsf{elevator}_0$, and any property of $\mathsf{Elevator}$ is also a property of $\mathcal{S}_{\mathsf{elevator}_0}$. Notice, however, that the converse does not hold for the following sentences:

[2] Recall that, by Definition 5, $\mathcal{N}_\mathcal{S} = \mathcal{N}$.

S1 $\mathsf{elevator}_0 \vDash (E \vee F0 \vee F1 \vee A) \rightarrow \mathbf{L}\,(\mathsf{C\_at\_P1})$
The actors E, $F0$, $F1$, and A know that the cabin is at the first platform.
S2 $\mathcal{S}_{\mathsf{elevator}_0} \vDash @_A\,[btn]\,\langle ctr \rangle\, E$
Whenever Alice calls the elevator, the request is transmitted to the Elevator.
S3 $\mathsf{Elevator} \vDash \big((F0 \vee F1) \rightarrow [obs]\,A\big) \wedge \big(\langle obs \rangle\, A \rightarrow (F0 \vee F1)\big)$
The floors can only be observed by Alice, and that is all Alice can observe.

Definition 16 (State specification). A *state specification* for an ANt schema $\mathcal{A}$ consists of a signature Σ for $\mathcal{A}$ (i.e., with the same actor names, channel types, and propositional symbols as the schema) and a set of sentences in $\mathsf{State}(\Sigma)$.

The sentences of a state specification of an ANt schema are used to restrict the set of admissible states of the actor networks defined over that schema.

Example 17. A state specification of Elevator could consist of the instances of the following sentence schemas, which we denote by Φ_{Elevator}:

E1 $@_C\,\langle \pi \rangle\, Pi \leftrightarrow @_E\,\mathbf{L}\,(\mathsf{C\_at\_P}i)$ for $i \in \{0,1\}$
The cabin is at platform i if and only if the elevator knows it.
E2 $p \rightarrow [obs]\,p$ for every $p \in \mathsf{Ł}(\mathcal{P})$
Knowledge is propagated through observation channels.

That is, this specification of Elevator determines what knowledge nodes have about the whereabouts of Cabin: the elevator proper always knows where Cabin is and the other nodes can acquire that information through observation channels.

The state logic is useful for deriving properties of states, ANt structures, and of ANt schemas and their specifications.

Definition 18 (Entailment). Given a finite set Φ of sentences and a sentence ϕ (defined over the same signature as Φ), we say that ϕ is a *semantic consequence* of Φ, or that Φ *entails* ϕ, and write $\Phi \vDash \phi$, if $\sigma \vDash \phi$ for all $\sigma \in \mathbb{S}_\Phi$.

The following two propositions allow us to redefine the three kinds validity of a state sentence in terms of entailment. They provide a syntactic characterization (as a set of sentences) $\Phi_\sigma / \Phi_{\mathcal{S}}$ for every state σ or structure $\mathcal{S}$. A sentence ϕ is valid at a state σ if and only if $\Phi_\sigma \vDash \phi$, and is valid at an ANt structure $\mathcal{S}$ if and only if $\Phi_{\mathcal{S}} \vDash \phi$; obviously, ϕ is absolutely valid if and only if $\emptyset \vDash \phi$.

Proposition 19. *Let $\Phi_{\mathcal{S}}$ be the (finite) set of all sentences of the form $@_a\,\langle \lambda \rangle\, b$ that are valid at a structure $\mathcal{S}$, where a and b are actor names and λ is a modality (i.e., a channel type or the parent modality π). Then $\mathbb{S}_{\Phi_{\mathcal{S}}} = \{\sigma \in \mathbb{S}_{\mathcal{A}} \mid \mathcal{S} \preceq \mathcal{S}_\sigma\}$.*

Proposition 20. *For every state σ, let Φ_σ be the (finite) set that extends $\Phi_{\mathcal{S}_\sigma}$ with all state sentences of the form $\neg @_a\,\langle \kappa \rangle\, b$, $@_a\,[\pi]\,\mathsf{false}$, $@_a\,\mathbf{L}p$, $@_a\,\mathbf{N}p$, or $@_a\,\mathbf{I}p$ that are valid at σ, where a and b are actor names, κ is a type that labels one of the channels between a and b in the ANt schema, and p is a propositional symbol. Then σ is the only state that satisfies Φ_σ.*

Entailment can be derived syntactically through the use of a proof system. The Hilbert-style axiomatization of the basic, unconstrained hybrid logic in Fig. 5 is both a simplification (because we do not consider the binder ↓) and an extension (due to the multi-modality setting and the different base logic) of the axiom system given in [3, Chap. 2].

Axioms

BLT	Axiom schemata for the base Łukasiewicz logic (see e.g. [12])
CT	Axiom schemata for $\neg$ and $\rightarrow$
Distr	$@_a (\phi_1 \rightarrow \phi_2) \leftrightarrow (@_a \phi_1 \rightarrow @_a \phi_2)$
SD	$@_a \phi \leftrightarrow \neg @_a \neg \phi$
Scope	$@_a @_b \phi \leftrightarrow @_b \phi$
Ref	$@_a a$
Intro	$(a \wedge \phi) \rightarrow @_a \phi$
λE	$([\lambda] \phi \wedge \langle\lambda\rangle a) \rightarrow @_a \phi$
$\forall E$	$\forall b \, \phi \rightarrow \phi[a/b]$

Inference rules

$$\frac{\phi_1 \qquad \phi_1 \rightarrow \phi_2}{\phi_2} \ (MP) \qquad \frac{\phi}{@_a \phi} \ (@I) \qquad \frac{@_a \phi}{\phi} \ (@E)^*$$

$$\frac{(\phi_1 \wedge \langle\lambda\rangle a) \rightarrow @_a \phi_2}{\phi_1 \rightarrow [\lambda] \phi_2} \ (\lambda I)^* \qquad \frac{\phi_1 \rightarrow \phi_2[a/b]}{\phi_1 \rightarrow \forall b \, \phi_2} \ (\forall I)^*$$

$*$ a does not occur free in ϕ ($@E$), in ϕ_1 or ϕ_2 (λI), or in ϕ_1 or $\forall b \, \phi_2$ ($\forall I$).

Fig. 5. Hilbert-style axiom schemata and rules for basic hybrid logic (Here, λ stands both for the regular modalities defined by channel types and for π.)

Proposition 21 **([3, Chap. 2]).** *The axiom schemata and inference rules presented in Fig. 5 are sound and complete with respect to the unconstrained Kripke-frame semantics of basic hybrid logic.*

Unlike the models of the basic hybrid logic, the models of the state logic are subject to the constraints defined by the ANt schema. Because of this, the axiom schemata and inference rules from Fig. 5 are no longer complete—though, obviously, they remain sound—with respect to the constrained Kripke semantics of the state logic. There are two main categories of new tautologies: those that arise from the ANt schema used, and those that are innate to the state logic. The former category contains, for instance, when considering the ANt schema Elevator from Example 2, sentences like $\neg @_A \langle ctr \rangle E$ (Alice cannot control the elevator directly.), while the latter contains sentences like $\neg @_a b$, where a and b are distinct actor names, or $\langle\pi\rangle \phi \rightarrow [\pi] \phi$.

In order to regain completeness, we introduce new axioms that reflect the semantic constraints of the state logic. The axiom schemata *N1* and *N2* from Fig. 6 ensure that all possible worlds correspond to actor names, and that no two distinct names are interpreted in the same way. The axiom schema *C1* rules out those channels that are not defined in the ANt schema, while *C2* captures the channel subtyping relation. Lastly, π*1* and π*2* specify that the interpretations of the distinguished parent modality are functional and acyclic, respectively.

N1	$\bigvee\{a \mid a \in \mathcal{N}\}$
N2	$\neg\, @_a\, b \quad \text{for } a \neq b \in \mathcal{N}$
C1	$\neg\, @_a \langle\kappa\rangle\, b \quad \text{if there is no } c \in \mathcal{C} \text{ with } \delta(c) = a,\ \rho(c) = b, \text{ and } \kappa \in \tau(c)$
C2	$\langle\kappa\rangle\, a \rightarrow \langle\kappa'\rangle\, a \quad \text{for } \kappa \leq \kappa' \text{ and } a \in \mathcal{N}$
π*1*	$\langle\pi\rangle\, a \rightarrow [\pi]\, a \quad \text{for } a \in \mathcal{N}$
π*2*	$\neg\, @_a \langle\pi\rangle^n a \quad \text{for } 1 \leq n \leq \lvert\mathcal{N}\rvert \text{ and } a \in \mathcal{N}$

Fig. 6. Additional axiom schemata for the state logic

Definition 22. Under the notations and assumptions of Definition 18, we say that ϕ is provable from Φ, and write $\Phi \vdash \phi$, if and only if ϕ can be derived from Φ using the axiom schemata and inference rules from Figs. 5 and 6.

The soundness and completeness of the proof system for the state logic follow from Proposition 21 and the lemma below.

Lemma 23. *For any* ANt *schema* $\mathcal{A}$*, the states in* $\mathbb{S}_{\mathcal{A}}$ *are given precisely by those Kripke structures that satisfy the axioms in Fig. 6.*

Proposition 24. *The extension of the proof system for hybrid logic with the axiom schemata in Fig. 6 yields a sound and complete axiomatization of the state logic.*

$$\Phi \vDash \phi \qquad \textit{iff} \qquad \Phi \vdash \phi$$

Example 25. The properties $@_{F0}\, \mathsf{L}\, (\mathsf{C\_at\_P1})$ and $@_A\, \mathsf{L}\, (\mathsf{C\_at\_P1})$ can be derived for $\mathcal{S}_{\mathsf{elevator}_0}$ under the specification Φ_{Elevator} from Example 17. In symbols,

$$\Phi_{\mathcal{S}_{\mathsf{elevator}_0}} \cup \Phi_{\mathsf{Elevator}} \vdash @_{F0}\, \mathsf{L}\, (\mathsf{C\_at\_P1}), @_A\, \mathsf{L}\, (\mathsf{C\_at\_P1})$$

This example shows that valuations can sometimes be fully determined by the ANt structure and the axioms associated with the ANt schema. In this particular case, the valuation of the atomic proposition $\mathsf{C\_at\_P1}$ at the nodes *F0* and *A* can be derived from the structural properties of the ANt structure and from the axiomatization of the way in which knowledge is propagated (see Example 17).

There are other general properties of $\mathsf{Elevator}$ that we might want to prove. For example, $@_C \langle\pi\rangle\, (P0 \vee P1)$—the cabin is either at *P0* or at *P1*. Because such properties are not structural, in the sense that they do not hold at every state of Elevator, they should be proved instead at the level of actor networks, which define the way states can evolve through repeated interactions. The corresponding logic for this kind of proofs is defined in the next sub-section.

3.3 The ANt Logic

The logic through which we can reason about the actor networks of an ANt schema $\mathcal{A}$ requires a further level of hybridization. In this case, a higher-level *actor-network signature* Ω consists of a signature Σ of the state logic (now playing the role of the base logic), a countably infinite set $\mathcal{N}om$ of nominals together with a non-empty subset $Init \subseteq \mathcal{N}om$ of names of initial states, and a set $\mathcal{I}$ of interactions for $\mathcal{A}$ (regarded as modalities).

Definition 26 (Syntax). The syntax of the ANt logic is given by the grammar

$$\psi ::= \phi \mid i \mid \neg\!\!\neg\, \psi \mid \psi \Rightarrow \psi \mid \langle\!|\iota|\!\rangle\, \psi \mid i : \psi \mid \exists\!\exists j\, \psi$$

where $\phi \in \mathsf{State}(\Sigma)$, $i \in \mathcal{N}om$, $j \in \mathcal{N}om \setminus Init$, and $\iota \in \mathcal{I}$. We denote by $\mathsf{ANt}(\Omega)$ the set of ANt-logic sentences defined over the signature Ω.

Notice that we use double symbols for the connectives of the ANt logic, and that the satisfaction operators are denoted using a colon. We extend the use of the double-symbol notation to the dual modal operators ($[\![\_]\!]$) and to the universal quantifier ($\forall\!\!\forall$), which are defined as in Sect. 3.2.

Example 27. We can now write sentences about the dynamics of Elevator like

$$\bigwedge \Phi_{\mathsf{elevator}_0} \Rightarrow \langle\!|\mathsf{callElevator0}|\!\rangle \bigwedge \Phi_{\mathsf{elevator}_1}$$

meaning that at the state $\mathsf{elevator}_0$ (which, by Proposition 20, is characterized by the sentences in $\Phi_{\mathsf{elevator}_0}$) there is a transition to the state $\mathsf{elevator}_1$ performed by the interaction callElevator0. Note that, by Proposition 20, the sets of sentences $\Phi_{\mathsf{elevator}_0}$ and $\Phi_{\mathsf{elevator}_1}$ are finite, hence the conjunctions in the antecedent and consequent of the implication above are well formed.

The semantics of the ANt logic is defined once more by means of constrained Kripke models. This time, we restrict only the interpretations of the modalities: all interactions $\iota \in \mathcal{I}$ are substructures of the underlying structures of the states on which (the relational interpretations of) ι are defined (see Definition 9). That is, the models of the ANt logic are actor networks.

Definition 28 (Semantics). The satisfaction relation for the ANt logic is defined for an actor network ν with interactions according to Ω, an assignment $\alpha\colon \mathcal{N}om \to \mathcal{D}_\nu$ (which, in this case, is just a function), and a world $w \in \mathcal{D}_\nu$:

- $\nu, \alpha, w \Vdash i$ iff $\alpha(i) = w$;
- $\nu, \alpha, w \Vdash \phi$ iff $\varsigma_\nu(w) \vDash \phi$;
- $\nu, \alpha, w \Vdash \neg\!\!\neg\, \psi$ iff $\nu, \alpha, w \nVdash \psi$;

- $\nu, \alpha, w \Vdash \psi_1 \Rightarrow \psi_2$ iff $\nu, \alpha, w \Vdash \psi_1$ implies $\nu, \alpha, w \Vdash \psi_2$;
- $\nu, \alpha, w \Vdash \langle\!| \iota |\!\rangle\, \psi$ iff there is a transition $w \xrightarrow{\iota} w'$ in ν such that $\nu, \alpha, w' \Vdash \psi$;
- $\nu, \alpha, w \Vdash i : \psi$ iff $\nu, \alpha, \alpha(i) \Vdash \psi$;
- $\nu, \alpha, w \Vdash \exists j\, \psi$ iff $\nu, \alpha', w \Vdash \psi$ for some α' that agrees with α on $\mathcal{N}om \setminus \{j\}$.

Similarly to the first level of hybridization, we say that an ANt-logic sentence ψ defined over Ω is valid in an actor network if it is satisfied, for every assignment, at every world of the network, and that a sentence ψ is absolutely valid if it is valid in every actor network:

- $\nu \Vdash \psi$ iff $\nu, \alpha, w \Vdash \psi$ for all assignments $\alpha \colon \mathcal{N}om \to \mathcal{D}_\nu$ and all $w \in \mathcal{D}_\nu$;
- $\Vdash \psi$ iff $\nu \Vdash \psi$ for all actor networks ν over Ω.

Given a set Ψ of ANt sentences, we denote by $\mathbb{N}_\Psi$ the set of actor networks over which all the sentences in Ψ are valid; and, given another sentence ψ, we say that Ψ entails ψ, which we denote $\Psi \Vdash \psi$, if $\nu \Vdash \psi$ for all $\nu \in \mathbb{N}_\Psi$.

The proof theory for the ANt logic builds once again on the proof theory for the basic hybrid logic. To that end, we use the same axiom schemata and inference rules from Fig. 5, only that in this case the tautologies of the Łukasiewicz logic are replaced by those of the state logic, and the Boolean and hybrid connectives of the state logic are replaced by those of the ANt logic. In addition, through the axiom schema *Inter* from Fig. 7, we introduce new axioms that reflect the semantic constraints of the models of the ANt logic: state properties of interactions hold in the states where the transitions occur.

Inter	$\langle\!| \iota |\!\rangle\, \mathsf{true} \Rightarrow \phi$	for all interactions ι and state-logic sentences $\phi \in \Phi_\iota$

Fig. 7. Additional axiom schema for the ANt logic

Definition 29. An ANt sentence ψ is provable from a set Ψ of sentences (of the same signature as ψ), denoted $\Psi \Vdash\!- \psi$, if ψ can be derived from Ψ using the axiom system for hybrid logic and the additional axiom schema defined in Fig. 7.

Example 30. Consider the following axiomatization of the transitions of an actor network for the ANt schema $\mathsf{Elevator}$. Most of the sentences below are of the form $\phi_1 \Rightarrow [\![\iota]\!]\, \phi_2$. They generalize Hoare triples and express properties of the transitions performed by interactions: intuitively, the sentence ϕ_1 is a precondition under which the interaction ι ensures the postcondition ϕ_2.

T1 $@_C \langle \pi \rangle\, P1 \Rightarrow [\![\mathsf{callElevator0}]\!]\, @_{P1} \langle mov \rangle\, P0$
When the elevator is called (at the ground floor) and the cabin is at the first platform, a request to move the cabin to the ground platform is issued.
T2 $@_C \langle \pi \rangle\, P0 \Rightarrow [\![\mathsf{callElevator0}]\!]\, @_{F0} \langle door \rangle\, (C \wedge \langle door \rangle\, F0)$
If the cabin is already at the ground platform, then the doors are opened.

T3 $[\![\mathsf{moveCabin0}]\!]\, @_{F0}\, \langle door \rangle\, (C \wedge \langle door \rangle\, F0)$
The doors are opened whenever the cabin moves to the (ground) platform.
T4 $@_A\, \langle \pi \rangle\, (F0 \wedge \langle door \rangle\, C) \Rightarrow \langle\!|\mathsf{enterCabin0}|\!\rangle\, \mathsf{true}$
If Alice is at *F0* and the doors are open, then she can enter the cabin.
T5 $@_a\, \langle \pi \rangle\, s \Rightarrow [\![\iota]\!]\, @_a\, \langle \pi \rangle\, t$ for interactions ι such that $\iota \vDash @_s\, \langle mov \rangle\, t$
Any interaction that involves a channel of type *mov* between actors s and t (regarded as locations) determines the movement to t of any actor in s.
T6 $@_a\, \langle \pi \rangle\, s \Rightarrow [\![\iota]\!]\, @_a\, \langle \pi \rangle\, s$ for interactions ι such that $\iota \nvDash @_s\, \langle mov \rangle\, \mathsf{true}$
But if the interaction does not involve a *mov* channel starting as s, then the actors in s maintain their location.

Then we can derive complex actor-network sentences such as:

$$@_C\, \langle \pi \rangle\, (P0 \vee P1) \Rightarrow [\![\mathsf{callElevator0}]\!]\, \big(@_A\, \mathbf{L}\, (\mathsf{C\_at\_P0}) \vee [\![\mathsf{moveCabin0}]\!]\, @_A\, \mathbf{L}\, (\mathsf{C\_at\_P0})\big)$$

That is, provided that we start at a world where Cabin is at one of the platforms, if the elevator is called, then Alice either knows immediately that the cabin is at the ground platform, or she discovers this as soon as the cabin is moved.

Proposition 31. *The extension of the hybrid-logic proof system with the axiom schema in Fig. 7 yields a sound and complete axiomatization of the* ANt *logic.*

$$\Psi \Vdash\!\!= \psi \qquad \textit{iff} \qquad \Psi \Vdash\!- \psi$$

4 Concluding Remarks

In this paper, we have shown how a suite of logics can be developed through a two-stage constrained-hybridization process, providing in this way support for the specification and verification of cyber-physical system protocols modeled as actor networks (ANts) in the sense of [9]. The first stage of the hybridization process results in a logic that captures the structure of actor networks and the way knowledge flows across such networks; the second addresses the dynamic aspects of actor networks, that is the way their structure can evolve as a result of the interactions that occur within them.

One of the main novelties of our approach is that we rely on unconventional semantic constraints, derived from the structural characteristics of actor-network states, or from the general properties of the state transitions. This results in faithful representations, at a logical level, of the way computation is performed in actor networks. That is, constrained models capture the relationship between the higher-level reconfigurations of networks and the lower-level interactions between actors that trigger them. Besides expressivity, a key property of these constraints is that they can be axiomatized within hybrid logic. This enables the use of conventional (sound and complete) proof systems for hybrid logic as a tool through which we can formally verify properties of actor networks.

Two main research directions are ongoing. The first aims to use the expressive power of our formalism to reason about security protocols in cyber-physical systems, in particular the existence of covert channels. The second aims to extend the logic to support the modeling of the transition system defined by interactions through graph transformations.

Acknowledgment. The work of D. Pavlovic was partially supported by NSF. J. Fiadeiro and A. Lopes received support from AFOSR for research visits to the University of Hawaii.

References

1. Blackburn, P.: Representation, reasoning, and relational structures: a hybrid logic manifesto. Logic J. IGPL **8**(3), 339–365 (2000)
2. Prior, A.: Past, Present and Future. Oxford University Press, Oxford (1967)
3. Braüner, T.: Hybrid Logic and its Proof-Theory. Applied Logic Series, vol. 37. Springer, Dordrecht (2011). https://doi.org/10.1007/978-94-007-0002-4
4. Neves, R., Madeira, A., Martins, M.A., Barbosa, L.S.: Proof theory for hybrid(ised) logics. Sci. Comput. Program. **126**, 73–93 (2016)
5. Diaconescu, R.: Quasi-varieties and initial semantics for hybridized institutions. J. Logic Comput. **26**(3), 855–891 (2016)
6. Găină, D.: Birkhoff style calculi for hybrid logics. Formal Aspects Comput. **29**(5), 805–832 (2017)
7. Madeira, A., Neves, R., Barbosa, L.S., Martins, M.A.: A method for rigorous design of reconfigurable systems. Sci. Comput. Program. **132**, 50–76 (2016)
8. Martins, M.A., Madeira, A., Diaconescu, R., Barbosa, L.S.: Hybridization of institutions. In: Corradini, A., Klin, B., Cîrstea, C. (eds.) CALCO 2011. LNCS, vol. 6859, pp. 283–297. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22944-2_20
9. Pavlovic, D., Meadows, C.: Actor-network procedures (extended abstract). In: Ramanujam, R., Ramaswamy, S. (eds.) ICDCIT 2012. LNCS, vol. 7154, pp. 7–26. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28073-3_2
10. Latour, B.: Reassembling the Social: An Introduction to Actor-Network Theory. Oxford University Press, Oxford (2005)
11. Milner, R.: The space and motion of communicating agents. CUP (2009)
12. Malinowski, G.: Many-Valued Logics. Oxford Logic Guides. Clarendon Press, Oxford (1993)

Parity Games and Automata for Game Logic

Helle Hvid Hansen[1], Clemens Kupke[2(✉)], Johannes Marti[2], and Yde Venema[3]

[1] Delft University of Technology, Delft, The Netherlands
[2] University of Strathclyde, Glasgow, UK
clemens.kupke@strath.ac.uk
[3] University of Amsterdam, Amsterdam, The Netherlands

Abstract. Parikh's game logic is a PDL-like fixpoint logic interpreted on monotone neighbourhood frames that represent the strategic power of players in determined two-player games. Game logic translates into a fragment of the monotone μ-calculus, which in turn is expressively equivalent to monotone modal automata. Parity games and automata are important tools for dealing with the combinatorial complexity of nested fixpoints in modal fixpoint logics, such as the modal μ-calculus. In this paper, we (1) discuss the semantics a of game logic over neighbourhood structures in terms of parity games, and (2) use these games to obtain an automata-theoretic characterisation of the fragment of the monotone μ-calculus that corresponds to game logic. Our proof makes extensive use of structures that we call syntax graphs that combine the ease-of-use of syntax trees of formulas with the flexibility and succinctness of automata. They are essentially a graph-based view of the alternating tree automata that were introduced by Wilke in the study of modal μ-calculus.

1 Introduction

Game logic was introduced by Parikh [23] as a modal logic for reasoning about strategic power in determined 2-player games, and it can be seen as a generalisation of PDL [16] both in terms of syntax and semantics. On the syntax side, game logic is a multi-modal language in which modalities are labelled by games, which in turn are built from atomic games, the PDL program constructs together with the operation *dual* which switches the role of the players. A modal formula $\langle\alpha\rangle\varphi$ should be read as *"player 1 has a strategy in the game α to achieve an outcome that satisfies the formula φ"*. On the semantic side, one goes from PDL to game logic by moving from Kripke frames to monotone neighbourhood frames. A game perspective on this generalisation is that nondeterministic programs (i.e., relations) are 1-player games in which the player chooses his move from a set of successors, and monotone neighbourhood frames are 2-player games where player 1 first chooses a neighbourhood U, and then player 2 chooses an element in U. The shift from Kripke frames to monotone neighbourhood frames also means that we go from normal modal logic to monotone modal logic. Just

C. Kupke and J. Marti—Supported by EPSRC grant EP/N015843/1.

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 115–132, 2018.
https://doi.org/10.1007/978-3-319-73579-5_8

as PDL (and other fixpoint logics such as LTL and CTL*) can be viewed as a fragment of the modal μ-calculus [2,20], game logic can be naturally viewed as a fragment of the *monotone μ-calculus* [24], which is monotone (multi-) modal logic with explicit fixpoint operators. A notable difference is that PDL, LTL and CTL* are all contained in level 1 or 2 of the alternation hierarchy whereas game logic, due to the combination of dual and iteration, spans all levels of the alternation hierarchy [1]. This high level of expressiveness could be an explanation for why a completeness proof for game logic is still missing.

In this paper we contribute to the theory of game logic. We discuss the semantics of game logic over neighbourhood structures using parity games and then use these games to characterise a class of automata that is exactly as expressive as formulas in game logic. Parity games are an intuitive way of dealing with the nesting of least and greatest fixpoint operators, and together with automata they play a fundamental role in the theory of fixpoint logics [12]. For instance, parity games and automata have been used in proving complexity results for the modal μ-calculus [7,8] and also Walukiewicz' completeness result [27] is proved by automata-theoretic means. Some of these results have been extended to the setting of coalgebraic fixpoint logic [10]. In particular, they are applicable to the monotone μ-calculus. Since monotone modal μ-calculus is expressively equivalent to a naturally defined class of *(unguarded) monotone modal automata* [11], it is of interest to find out which subclass of these automata corresponds to game logic. The main result in our paper is a characterisation of a class of unguarded monotone modal automata that effectively corresponds to game logic, in the sense that there are effective translations in both directions. This result can be seen as the game logic analogue of the characterisation of PDL in automata-theoretic terms [3]. The case of game logic, however, is more involved because composition of games does not distribute from the left over choice as is the case for the programs in PDL. This is related to the fact that in the relational semantics of PDL, diamonds distribute over disjunctions; this property, which is heavily exploited in the mentioned results on PDL, does not apply to the diamonds of game logic. Finally, note that our characterisation can also be seen as an automata-theoretic counterpart to the results in [4, Sect. 3.3] that characterise a fragment of the μ-calculus that is expressively equivalent to game logic interpreted over Kripke frames.

Our characterisation goes via a class of structures that we call *syntax graphs*. Syntax graphs combine the ease-of-use of syntax trees of formulas with the flexibility and succinctness of automata. They are essentially the same as Wilke's alternating tree automata (ATAs) [29] except they are described in terms of their transition graphs, and they run on monotone neighbourhood models rather than Kripke models. Unguarded monotone modal automata can, in turn, be viewed as Wilke's ATAs with complex transition condition [29] (again with a semantics over monotone neighbourhood models). As noted in [19,29] an ATA with complex transition conditions can be effectively translated into an equivalent ATA, and this construction is easily seen to work also for monotone semantics. Concretely, our characterisation consists of a number of conditions that define a subclass **GG** of syntax graphs that correspond to game logic formulas. We call these *game logic graphs*. A game automaton is then a monotone modal automaton whose corresponding syntax graph (i.e. ATA) is in **GG**. The translation from formulas to game logic graphs is an inductive construction similar to the construction of a nondeterministic automaton from a regular expression. Conversely, the defining conditions on game logic graphs allow us to decompose a game logic graph into components that correspond to formulas.

The rest of the paper is structured as follows. In Sect. 2 we recall the syntax and neighbourhood semantics of game logic and describe a normal form that is needed for our results. In Sect. 3 we introduce the game semantics for game logic and prove it to be equivalent to the neighbourhood semantics. In Sect. 4 we discuss syntax graphs and their game semantics. In Sect. 5 we define game logic graphs and prove them to be expressively equivalent to formulas in game logic. Due to space constraints, proofs are provided in an extended version of this paper [15].

2 Game Logic

Most definitions and results in this section are from [23,25]. The syntax of game logic is based on the syntax of propositional modal logic with the additional feature that modal operators are labelled with terms that denote games. Since we have "test games" of the form $\varphi?$, the definition of the syntax is a simultaneous recursion on the structure of formulas and games.

Definition 1. *Throughout the paper we fix a countable set* Prop *of atomic propositions (proposition letters) and a set* Gam *of atomic games. The sets $\mathcal{F}$ of formulas and $\mathcal{G}$ of game terms of game logic are defined recursively as follows:*

$$\mathcal{F} \ni \varphi ::= p \in \mathsf{Prop} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle\alpha\rangle\varphi, \quad \textit{where } \alpha \in \mathcal{G}$$
$$\mathcal{G} \ni \alpha ::= g \in \mathsf{Gam} \mid \alpha^d \mid \alpha \cup \alpha \mid \alpha \cap \alpha \mid \alpha;\alpha \mid \alpha^* \mid \alpha^\times \mid \varphi? \mid \varphi!, \quad \textit{where } \varphi \in \mathcal{F}$$

We use the standard definitions of $\to$ and $\leftrightarrow$, and note that $\top$ can be defined as $p \vee \neg p$ for any $p \in$ Prop. In the following we denote formulas by $\varphi, \psi, \ldots$ and game terms with $\alpha, \beta, \rho, \ldots$. We use the letter χ to denote arbitrary terms that could either be a formula or a game term.

The formulas of game logic express strategic power in 2-player determined, zero-sum games. A formula $\langle\alpha\rangle\varphi$ says that player 1 has a strategy in the game α to ensure that the outcome of the game satisfies φ. The assumption that the games are determined and zero-sum means that in a given game α, player 2 has a strategy to achieve φ iff player 1 does not have a strategy to achieve $\neg\varphi$. Hence the formula $\neg\langle\alpha\rangle\neg\varphi$, usually written as $[\alpha]\varphi$, says that player 2 has a strategy in α to ensure an outcome that satisfies φ. For technical reasons we do not include boxes as primitive operators.

It will be convenient to refer to player 1 as Angel and player 2 as Demon. The game operations can then be explained as follows. The composition $\alpha\,;\beta$ is the game consisting of playing α followed by β. The angelic choice $\alpha \cup \beta$ (resp. demonic choice $\alpha \cap \beta$) is the game in which Angel (resp. Demon) chooses whether to play α or β. The angelic iteration α^* is the game in which α is played 0 or more times, and after each time, Angel chooses whether to stop or play again, but she must stop after some finite number of iterations. The demonic iteration $\alpha^\times$ is the iterated game in which Demon chooses when to stop, and he may choose to play forever. The formula $\langle \alpha^* \rangle \varphi$ thus says that Angel has a strategy to reach a φ-state by playing α some finite number of rounds (where her strategy may depend on what Demon did in previous rounds, so that in particular, the number of rounds needed to reach φ is not determined at the start of the game). The formula $\langle \alpha^\times \rangle \varphi$ says that Angel has a strategy for maintaining φ indefinitely when playing α repeatedly. Finally, the dual game α^d is the same as α but with the roles of the two players reversed, i.e., Angel has a strategy to achieve φ in α^d iff Demon has a strategy to achieve φ in α, and vice versa.

In [23,25], the language of game logic only contained the game operations $;\, \cup, ^*, ^d$, and the demonic operations were defined as $\alpha \cap \beta = (\alpha^d \cup \beta^d)^d$ and $\alpha^\times = ((\alpha^d)^*)^d$. We take the demonic operations as primitives, since later we want to reduce formulas to dual and negation normal form.

The formal semantics of game logic is given by representing games as monotone neighbourhood frames. These are well known semantic structures in modal logic [5,13].

Definition 2. *Let S be a set. We denote by $\mathcal{M}(S)$ the set of up-closed subsets of $\mathcal{P}(S)$, i.e., $\mathcal{M}(S) = \{N \subseteq \mathcal{P}(S) \mid \forall U, U' : U \in N, U \subseteq U' \Rightarrow U' \in N\}$. A* monotone neighbourhood frame on *S is a function $f \colon S \to \mathcal{M}(S)$. We denote by* $\mathrm{MF}(S)$ *the set of all monotone neighbourhood frames on S.*

For $f \in \mathrm{MF}(S)$ and $s \in S$, the subsets U in $f(s)$ are called the neighbourhoods of s. We point out that such neighbourhoods are not necessarily neighbourhoods in the topological sense. In particular, we do not require that a state s is an element of all its neighbourhoods. In our setting, the neighbourhoods will be the subsets that Angel can force in the game represented by f.

We note that $(\mathcal{M}(S), \subseteq)$ is a complete partial order with associated join and meet given by union and intersection of neighbourhood collections. This CPO structure lifts pointwise to a CPO $(\mathrm{MF}(S), \sqsubseteq)$ in which we also denote join and meet by $\cup$ and $\cap$.

In analogue with how the PDL program operations are interpreted in relation algebra, we interpret game operations via algebraic structure on $\mathrm{MF}(S)$.[1]

Definition 3 (Game operations). *Let $f, f_1, f_2 \in \mathrm{MF}(S)$ be monotone neighbourhood frames. We define*

- *the unit frame η_S by:* $U \in \eta_S(s)$ *iff* $s \in U$ *for* $s \in S$ *and* $U \subseteq S$.

[1] It is well-known that $\mathcal{M}$ is a monad, [14]. Readers who are familiar with monads will recognise that unit and composition correspond to the unit and Kleisli composition.

– *the composition* $f_1 \,;\, f_2$ *by:*

$$U \in (f_1 \,;\, f_2)(s) \text{ iff } \{s' \in S \mid U \in f_2(s')\} \in f_1(s) \qquad \text{for } s \in S \text{ and } U \subseteq S.$$

– *the Angelic choice and Demonic choice between* f_1 *and* f_2 *by:*

$$(f \cup g)(s) = f(s) \cup g(s) \qquad (f \cap g)(s) = f(s) \cap g(s), \qquad \text{for } s \in S.$$

– *the dual* f^d *by:* $U \in f^d(s)$ *iff* $S \setminus U \notin f(s)$ *for* $s \in S$ *and* $U \subseteq S$.
– *the angelic iteration* $f^* := \mathrm{LFP}(\mathsf{A}_f)$,
– *the demonic iteration* $f^\times := \mathrm{GFP}(\mathsf{D}_f)$,

where $\mathrm{LFP}(\mathsf{A}_f)$ *and* $\mathrm{GFP}(\mathsf{D}_f)$ *are the least and greatest fixed points of the maps*

$$\begin{aligned} \mathsf{A}_f : \mathrm{MF}(S) &\to \mathrm{MF}(S) \\ g &\mapsto \eta_S \cup (f \,;\, g) \end{aligned} \qquad\qquad \begin{aligned} \mathsf{D}_f : \mathrm{MF}(S) &\to \mathrm{MF}(S) \\ g &\mapsto \eta_S \cap (f \,;\, g) \end{aligned}$$

Note that for any $f \in \mathrm{MF}(S)$, *the map* $g \mapsto f \,;\, g$ *is a monotone operation on* $(\mathrm{MF}(S), \sqsubseteq)$ *and hence so are* A_f *and* D_f. *By the Knaster-Tarski theorem,* A_f *and* D_f *have unique least and greatest fixed points.*

It is straightforward to verify that $\mathrm{MF}(S)$ is closed under the above operations. The following lemma lists a number of identities that will be useful in reasoning about game logic semantics.

Lemma 1. *For all* $f, g \in \mathrm{MF}(S)$, *we have:*

1. $(f^d)^d = f$	*4.* $(f \cup g)^d = f^d \cap g^d$	*7.* $(f^*)^d = (f^d)^\times$
2. $(f \,;\, g)^d = f^d \,;\, g^d$	*5.* $(f \cap g)^d = f^d \cup g^d$	*8.* $(f^\times)^d = (f^d)^*$
3. $(\eta_S)^d = \eta_S$	*6.* $f \subseteq g \;\Rightarrow\; g^d \subseteq f^d$	

We now have all the definitions in place to define game models and the semantics of formulas and games. We first give some intuitions. A game model consists of a state space together with interpretations of atomic propositions (as subsets of the state space) and atomic games (as monotone neighbourhood frames). The semantics of complex formulas and complex games is then defined by mutual induction. For a formula φ, the semantics $[\![\varphi]\!]$ is defined via the usual definitions from monotone modal logic. For a game α, the semantics $\langle\!|\alpha|\!\rangle$ is a monotone neighbourhood frame defined via the game constructions given above. The subsets U in $\langle\!|\alpha|\!\rangle(s)$ are the sets of outcomes that Angel can "force" when playing the game α in state s.

Definition 4. *A* game model *is a triple* $\mathbb{S} = (S, \gamma, \Upsilon)$ *where* S *is a set of states,* $\gamma : \mathsf{Gam} \to \mathrm{MF}(S)$ *is a* Gam*-indexed collection of monotone neighbourhood frames, which provides an interpretation of atomic games, and* $\Upsilon : \mathsf{Prop} \to \mathcal{P}(S)$ *is a valuation of atomic propositions. For* $\varphi \in \mathcal{F}$ *and* $\alpha \in \mathcal{G}$ *we define the* *semantics* $[\![\varphi]\!]_{\mathbb{S}} \subseteq S$ *and* $(\!|\alpha|\!)_{\mathbb{S}} \in \mathrm{MF}(S)$ *by induction on the term structure:*

$$
\begin{array}{rlrl}
[\![p]\!]_{\mathbb{S}} := \Upsilon(p) & \textit{for } p \in \mathsf{Prop} & [\![\neg\varphi]\!]_{\mathbb{S}} & := S \setminus [\![\varphi]\!]_{\mathbb{S}} \\
[\![\varphi_1 \vee \varphi_2]\!]_{\mathbb{S}} := [\![\varphi_1]\!]_{\mathbb{S}} \cup [\![\varphi_2]\!]_{\mathbb{S}} & & [\![\varphi_1 \wedge \varphi_2]\!]_{\mathbb{S}} & := [\![\varphi_1]\!]_{\mathbb{S}} \cap [\![\varphi_2]\!]_{\mathbb{S}} \\
[\![\langle\alpha\rangle\varphi]\!]_{\mathbb{S}} := \{s \in S \mid [\![\varphi]\!]_{\mathbb{S}} \in (\!|\alpha|\!)_{\mathbb{S}}(s)\} & & (\!|\alpha;\beta|\!)_{\mathbb{S}} & := (\!|\alpha|\!)_{\mathbb{S}} \, ; (\!|\beta|\!)_{\mathbb{S}} \\
(\!|g|\!)_{\mathbb{S}} := \gamma(g) \textit{ for } g \in \mathsf{Gam} & & (\!|\alpha^d|\!)_{\mathbb{S}} & := ((\!|\alpha|\!)_{\mathbb{S}})^d \\
(\!|\alpha \cup \beta|\!)_{\mathbb{S}} := (\!|\alpha|\!)_{\mathbb{S}} \cup (\!|\beta|\!)_{\mathbb{S}} & & (\!|\alpha \cap \beta|\!)_{\mathbb{S}} & := (\!|\alpha|\!)_{\mathbb{S}} \cap (\!|\beta|\!)_{\mathbb{S}} \\
(\!|\alpha^*|\!)_{\mathbb{S}} := ((\!|\alpha|\!)_{\mathbb{S}})^* & & (\!|\alpha^\times|\!)_{\mathbb{S}} & := ((\!|\alpha|\!)_{\mathbb{S}})^\times \\
(\!|\psi?|\!)_{\mathbb{S}} := \lambda x. \begin{cases} \eta_S(x) & \textit{if } x \in [\![\psi]\!]_{\mathbb{S}} \\ \emptyset & \textit{otherwise.} \end{cases} & & (\!|\psi!|\!)_{\mathbb{S}} & := \lambda x. \begin{cases} \eta_S(x) & \textit{if } x \notin [\![\psi]\!]_{\mathbb{S}} \\ \mathcal{P}S & \textit{otherwise.} \end{cases}
\end{array}
$$

We write $\varphi \equiv \psi$ if for all $\mathbb{S}$, $[\![\varphi]\!]_{\mathbb{S}} = [\![\psi]\!]_{\mathbb{S}}$. Similarly, we write $\alpha \equiv \beta$ if for all $\mathbb{S}$, $(\!|\alpha|\!)_{\mathbb{S}} = (\!|\beta|\!)_{\mathbb{S}}$. We will often omit the subscript $\mathbb{S}$, if $\mathbb{S}$ is clear from the context, or irrelevant.

The following lemma states some basic identities involving the dual operator, and a congruence property.

Lemma 2. *Let* $\varphi, \psi \in \mathcal{F}$ *and* $\alpha, \beta \in \mathcal{G}$. *We have:*

1. $(\alpha^d)^d \equiv \alpha$
2. $(\alpha;\beta)^d \equiv \alpha^d;\beta^d$
3. $(\alpha \cup \beta)^d \equiv \alpha^d \cap \beta^d$
4. $(\alpha \cap \beta)^d \equiv \alpha^d \cup \beta^d$
5. $(\alpha^*)^d \equiv (\alpha^d)^\times$
6. $(\alpha^\times)^d \equiv (\alpha^d)^*$
7. $(\psi?)^d \equiv (\neg\psi)!$
8. $(\psi!)^d \equiv (\neg\psi)?$
9. $\langle\alpha^d\rangle\varphi \equiv \neg\langle\alpha\rangle\neg\varphi$
10. *If* $\alpha \equiv \beta$ *and* $\varphi \equiv \psi$ *then* $\langle\alpha\rangle\varphi \equiv \langle\beta\rangle\psi$

We will make frequent use of the fact that all formulas and game terms can be reduced to a dual and negation normal form.

Definition 5. *A formula* $\varphi \in \mathcal{F}$, *resp. game term* $\alpha \in \mathcal{G}$, *is in* dual and negation normal form (DNNF) *if dual is only applied to atomic games and negations occur only in front of proposition letters. We denote by* $\mathcal{F}_{\mathrm{DNNF}}$ *the set of formulas in DNNF, and by* $\mathcal{G}_{\mathrm{DNNF}}$ *the set of game terms in DNNF.*

Lemma 3. *For all* $\varphi \in \mathcal{F}$, *there is a DNNF formula* $\mathrm{nf}(\varphi)$ *such that* $\varphi \equiv \mathrm{nf}(\varphi)$. *For all* $\alpha \in \mathcal{G}$, *there is a DNNF game term* $\mathrm{nf}(\alpha)$ *such that* $\alpha \equiv \mathrm{nf}(\alpha)$.

From now on we will generally assume that formulas are in DNNF. The following lemma lists some crucial validities that form the basis for the definition of the game semantics in the next section. It is straightforward to verify that these formulas are valid.

Lemma 4. *The following formulas are valid in all game models:*

$$
\begin{array}{ll}
\langle\alpha;\beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\langle\beta\rangle\varphi & \langle\alpha^d\rangle\varphi \leftrightarrow \neg\langle\alpha\rangle\neg\varphi \\
\langle\alpha \cup \beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\varphi \vee \langle\beta\rangle\varphi & \langle\alpha \cap \beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\varphi \wedge \langle\beta\rangle\varphi \\
\langle\alpha^*\rangle\varphi \leftrightarrow \varphi \vee \langle\alpha\rangle\langle\alpha^*\rangle\varphi & \langle\alpha^\times\rangle\varphi \leftrightarrow \varphi \wedge \langle\alpha\rangle\langle\alpha^\times\rangle\varphi \\
\langle\psi?\rangle\varphi \leftrightarrow \psi \wedge \varphi & \langle\psi!\rangle\varphi \leftrightarrow \psi \vee \varphi
\end{array}
$$

3 Game Semantics for Game Logic

In this section we will see how games provide an operational semantics for game logic. In particular, we will develop a two-player evaluation game for game logic, very much in the spirit of Berwanger [1]. Note however, that the ambient model-theoretic structures in our setting are *monotone neighbourhood structures*, whereas Berwanger restricts to (relational) Kripke structures. Our approach allows for a neat formulation of some useful additional observations involving the unfolding games related to monotone operations on full powersets [26].

3.1 Game Preliminaries

Two-player *graph games* are an important tool for fixpoint logics. We will briefly recall their definition and the related terminology. For a more comprehensive account of these games, the reader is referred to [12]. A graph game is played on a *board* B, that is, a set of *positions*. Each position $b \in B$ *belongs* to one of the two *players*, Eloise (abbr. $\exists$) and Abelard (abbr. $\forall$). Formally we write $B = B_\exists \cup B_\forall$, and for each position b we use $P(b)$ to denote the player i such that $b \in B_i$. Furthermore, the board is endowed with a binary relation E, so that each position $b \in B$ comes with a set $E[b] \subseteq B$ of *successors*. Note that we do not require the games to be strictly alternating, i.e., successors of positions in $B_\exists$ or $B_\forall$ can lie again in $B_\exists$ or $B_\forall$, respectively. Formally, we say that the *arena* of the game consists of a directed two-sorted graph $\mathbb{B} = (B_\exists, B_\forall, E)$.

A *match* or *play* of the game consists of the two players moving a pebble around the board, starting from some *initial position* b_0. When the pebble arrives at a position $b \in B$, it is player $P(b)$'s turn to move; (s)he can move the pebble to a new position of their liking, but the choice is restricted to a successor of b. Should $E[b]$ be empty then we say that player $P(b)$ *got stuck* at the position. A *match* or *play* of the game thus constitutes a (finite or infinite) sequence of positions $b_0 b_1 b_2 \ldots$ such that $b_i E b_{i+1}$ (for each i such that b_i and b_{i+1} are defined). A *full play* is either (i) an infinite play or (ii) a finite play in which the last player got stuck. A non-full play is called a *partial* play. Each full play of the game has a *winner* and a *loser*. A finite full play is lost by the player who got stuck; the winning condition for infinite games is usually specified using a so-called *parity function*, i.e., a function $\Omega : B \to \mathbb{N}$ that maps each position to a natural number (its *priority*) and that has finite range. An infinite play $\Pi = b_0 b_1 \ldots b_n \cdots \in B^\omega$ is won by Eloise if $\max\{\Omega(b) \mid b \in \mathrm{Inf}(\Pi)\}$ is even, where $\mathrm{Inf}(\Pi)$ denotes the positions from B that occur infinitely often in Π. Otherwise Abelard wins this play. A graph game with parity function Ω is a *parity game*. All graph games used in this paper are parity games, but we will not specify the parity function explicitly in simple cases (e.g. when one of the players is supposed to win all infinite plays).

A *strategy* for player i tells player i how to play at all positions where it is i's turn to move. A strategy can be represented as a *partial* function which maps partial plays $\beta = b_0 \cdots b_n$ with $P(b_n) = i$ to legal next positions (that is, to elements of $E[b_n]$), and which is undefined for partial plays $\beta = b_0 \cdots b_n$ with $E[b_n] = \emptyset$. We say that a play $\Pi = b_1 \dots b_n \cdots \in B^* \cup B^\omega$ follows a strategy f if for all positions b_j in Π on which f is defined we have $f(b_j) = b_{j+1}$. A strategy is *positional* if it only depends on the current position of the match. A strategy is *winning for player* i from position $b \in B$ if it guarantees i to win any match with initial position b, no matter how the adversary plays—note that this definition also applies to positions b for which $P(b) \neq i$. A position $b \in B$ is called a *winning position* for player i, if i has a winning strategy from position b; the set of winning positions for i in a game $\mathcal{F}$ is denoted as $\mathrm{Win}_i(\mathcal{F})$. Parity games are *positionally determined*, i.e., at each position of the game board exactly one of the players has a positional winning strategy (cf. [9,22]).

3.2 Definition of the Evaluation Game

In order to be able to trace the unfoldings of fixpoint operators within games we need some terminology concerning the nesting of fixpoints. Firstly, we need notation for the subterm relation and the definition of a parity map for a formula.

Definition 6. *We let* $\lhd \subseteq (\mathcal{F} \cup \mathcal{G})^2$ *be the subterm relation on formulas and game terms, i.e.,* $\xi_1 \lhd \xi_2$ *if either* $\xi_1 = \xi_2$ *or* ξ_1 *is a proper subterm of* ξ_2*.*

Definition 7. *For a term* $\xi \in \mathcal{F} \cup \mathcal{G}$ *we let* $\mathrm{Fix}(\xi) := \{\alpha^* \mid \alpha \in \mathcal{G}, \alpha^* \lhd \xi\} \cup \{\alpha^\times \mid \alpha \in \mathcal{G}, \alpha^\times \lhd \xi\}$*. A parity function for a formula* φ *in DNNF is a partial map* $\Omega : \mathrm{Fix}(\varphi) \to \omega$ *such that*

1. $\alpha_1 \lhd \alpha_2$ *implies* $\Omega(\alpha_1) < \Omega(\alpha_2)$ *for all* $\alpha_1, \alpha_2 \in \mathrm{Fix}(\varphi)$ *with* $\alpha_1 \neq \alpha_2$*, and*
2. *for all* $\alpha \in \mathrm{Fix}(\varphi)$*,* $\Omega(\alpha)$ *is even iff* $\alpha = \rho^\times$ *is a demonic iteration.*

We define the canonical parity function $\Omega_{\mathrm{can}} : \mathrm{Fix}(\varphi) \to \omega$ *associated with* φ *as the partial function given by* $\Omega_{\mathrm{can}}(\alpha^*) = 2n + 1$ *and* $\Omega_{\mathrm{can}}(\alpha^\times) = 2n$ *where* $n = \#\mathrm{Fix}(\alpha^*)$ *and* $n = \#\mathrm{Fix}(\alpha^\times)$*, respectively. The canonical parity function formalises the fact that any fixpoint operator dominates any other fixpoint operator in its scope.*

Definition 8. *Let* $\mathbb{S} = (S, \gamma, \Upsilon)$ *be a game model, let* $\varphi \in \mathcal{F}$ *be a formula in DNNF and let* $\Omega : \mathrm{Fix}(\varphi) \to \omega$ *be a parity function for* φ*. We define the evaluation game* $\mathcal{E}(\mathbb{S}, \varphi)$ *as the parity graph game with the game board specified in Fig. 1 and the parity function* $\Omega_{\mathcal{E}}$ *given by*

$$\Omega_{\mathcal{E}}(b) := \begin{cases} \Omega(\alpha) & \text{if } b = (x, \langle\alpha\rangle\psi) \text{ for some } \alpha \in \mathrm{Fix}(\varphi) \\ 0 & \text{otherwise.} \end{cases}$$

3.3 Adequacy of Game Semantics

In this section we show that the game semantics of Definition 8 is equivalent to the standard semantics of game logic from Definition 4 where we assume w.l.o.g. that formulas are in DNNF.

Formula Part

Position b	P(b)	Moves E[b]
$(s,p), s \in \Upsilon(p)$	$\forall$	$\emptyset$
$(s,p), s \notin \Upsilon(p)$	$\exists$	$\emptyset$
$(s,\neg p), s \in \Upsilon(p)$	$\exists$	$\emptyset$
$(s,\neg p), s \notin \Upsilon(p)$	$\forall$	$\emptyset$
$(s,\varphi \wedge \psi)$	$\forall$	$\{(s,\varphi),(s,\psi)\}$
$(s,\varphi \vee \psi)$	$\exists$	$\{(s,\varphi),(s,\psi)\}$
$(s,\langle g\rangle\varphi)$	$\exists$	$\{(U,\langle g\rangle\varphi) \mid U \in \langle\!\lvert g\rvert\!\rangle(s)\}$
$(U,\langle g\rangle\varphi)$	$\forall$	$\{(s,\varphi) \mid s \in U\}$
$(s,\langle g^d\rangle\varphi)$	$\forall$	$\{(U,\langle g^d\rangle\varphi) \mid U \in \langle\!\lvert g\rvert\!\rangle(s)\}$
$(U,\langle g^d\rangle\varphi)$	$\exists$	$\{(s,\varphi) \mid s \in U\}$

Game Part

Position b	P(b)	Moves E[b]
$(s,\langle\alpha\,;\beta\rangle\varphi)$	$\star$	$\{(s,\langle\alpha\rangle\langle\beta\rangle\varphi)\}$
$(s,\langle\alpha\cup\beta\rangle\varphi)$	$\star$	$\{(s,\langle\alpha\rangle\varphi \vee \langle\beta\rangle\varphi)\}$
$(s,\langle\alpha\cap\beta\rangle\varphi)$	$\star$	$\{(s,\langle\alpha\rangle\varphi \wedge \langle\beta\rangle\varphi)\}$
$(s,\langle\alpha^*\rangle\varphi)$	$\star$	$\{(s,\varphi \vee \langle\alpha\rangle\langle\alpha^*\rangle\varphi)\}$
$(s,\langle\alpha^\times\rangle\varphi)$	$\star$	$\{(s,\varphi \wedge \langle\alpha\rangle\langle\alpha^\times\rangle\varphi)\}$
$(s,\langle\psi?\rangle\varphi)$	$\star$	$\{(s,\psi \wedge \varphi)\}$
$(s,\langle\psi!\rangle\varphi)$	$\star$	$\{(s,\psi \vee \varphi)\}$

Fig. 1. Game board of the evaluation game. We use $P(b) = \star$ to express that it is irrelevant which player moves, since there is exactly one possible move.

To compare the two different semantics we need a game characterisation of the $(\_)^*$ and $(\_)^\times$-operations. As both operations are defined as fixpoints they can be characterised via fixpoint games (these games are straightforward adaptation of the unfolding game described in [26]). We provide some intuition below the definition.

Definition 9. *Let $\alpha \in \mathcal{G}$ be a game term, let $\mathbb{S} = (S,\gamma,\Upsilon)$ be a game model and let $U \subseteq S$. The games $\mathcal{F}(\mathbb{S},\alpha^*,U)$ and $\mathcal{F}(\mathbb{S},\alpha^\times,U)$ have the following game boards:*

Board of $\mathcal{F}(\mathbb{S},\alpha^*,U)$:

Pos. b	*$P(b)$*	*Moves $E[b]$*
$s \in S$	$\exists$	$\begin{cases}\{\emptyset\} & \text{if } s \in U\\ \langle\!\lvert\alpha\rvert\!\rangle(s) & \text{otherwise.}\end{cases}$
$U' \in \mathcal{P}(S)$	$\forall$	U'

Board of $\mathcal{F}(\mathbb{S},\alpha^\times,U)$:

Pos. b	*$P(b)$*	*Moves $E[b]$*
$s \in S$	$\exists$	$\begin{cases}\langle\!\lvert\alpha\rvert\!\rangle(s) & \text{if } s \in U\\ \emptyset & \text{otherwise.}\end{cases}$
$U' \in \mathcal{P}(S)$	$\forall$	U'

The winning conditions in these games are as usual: finite complete plays are lost by the player that gets stuck. Infinite plays of $\mathcal{F}(\mathbb{S},\alpha^*,U)$ and $\mathcal{F}(\mathbb{S},\alpha^\times,U)$ are won by Abelard and Eloise, respectively.

The fixpoint game $\mathcal{F}(\mathbb{S},\alpha^*,U)$ works as follows. The objective of Eloise is to reach U in finitely many rounds of α. At a position $s \in U$, Eloise can win by choosing the move $\emptyset$ which causes Abelard to get stuck in the next step, since he must choose from the empty set of moves. At a position $s \notin U$, Eloise chooses an α-neighbourhood U' of s, and in the next step Abelard then chooses a state $s' \in U'$, and the game continues. In the game $\mathcal{F}(\mathbb{S},\alpha^\times,U)$, the objective of Eloise is to stay in U indefinitely. At a position $s \notin U$, she therefore loses immediately (indeed, she is stuck at such positions, since her set of moves is empty). But at a position $s \in U$, the players play another round of α, and the game continues.

Lemma 5. *For all* $\mathbb{S} = (S, \gamma, \Upsilon)$, $\alpha \in \mathcal{G}$, $s \in S$ *and* $U \subseteq S$, *we have:*

$$s \in \mathrm{Win}_\exists(\mathcal{F}(\mathbb{S}, \alpha^*, U)) \quad \textit{iff} \quad U \in \langle\!|\alpha^*|\!\rangle(s), \textit{ and}$$
$$s \in \mathrm{Win}_\exists(\mathcal{F}(\mathbb{S}, \alpha^\times, U)) \quad \textit{iff} \quad U \in \langle\!|\alpha^\times|\!\rangle(s).$$

The lemma easily follows because the games $\mathcal{F}(\mathbb{S}, \alpha^*, U)$ and $\mathcal{F}(\mathbb{S}, \alpha^\times, U)$ are instances of Tarski's fixpoint games that characterise least and greatest fixpoints of a monotone operator.

The following technical lemma demonstrates that winning strategies for Eloise in the evaluation game entail the existence of certain neighbourhood sets in the game model that witness the truth of a modal formula. There is no requirement on the witness to be non-empty, e.g., $s \models \langle\alpha\rangle\bot$ if $\emptyset \in \langle\!|\alpha|\!\rangle(s)$.

Lemma 6. *Let* $\varphi \in \mathcal{F}$, *let* $\mathbb{S} = (S, \gamma, \Upsilon)$ *be a game model and consider the game* $\mathcal{E} = \mathcal{E}(\mathbb{S}, \varphi)$. *Assume that* $f_\exists$ *is a winning strategy for Eloise in* $\mathcal{E}$, *and that* $(s, \langle\alpha\rangle\psi) \in Win_\exists(\mathcal{E})$. *Let* $\mathrm{Win}_\psi(\mathcal{E}) := \{s' \in S \mid (s', \psi) \in \mathrm{Win}_\exists(\mathcal{E})\}$ *and suppose* $\mathrm{Win}_\psi(\mathcal{E}) \subseteq [\![\psi]\!]$. *Then* $\mathrm{Win}_\psi(\mathcal{E}) \in \langle\!|\alpha|\!\rangle(s)$.

The lemma is the key to prove one direction of the adequacy of our game semantics.

Proposition 1. *Let* $\varphi \in \mathcal{F}$, *let* $\mathbb{S} = (S, \gamma, \Upsilon)$ *be a game model and consider* $\mathcal{E} = \mathcal{E}(\mathbb{S}, \varphi)$. *For all* ψ *occurring in* $\mathcal{E}$ *we have* $\mathrm{Win}_\psi(\mathcal{E}) \subseteq [\![\psi]\!]_\mathbb{S}$.

The claim is proven by induction on ψ and follows easily from Lemma 6. For the second half of the adequacy theorem we again need a technical lemma.

Lemma 7. *Let* $\mathbb{S} = (S, \gamma, \Upsilon)$ *be a game model and let* $\varphi \in \mathcal{F}$. *For any position* $(s, \langle\alpha\rangle\psi)$ *of the game* $\mathcal{E} = \mathcal{E}(\mathbb{S}, \varphi)$ *and for all* $U \subseteq [\![\psi]\!]_\mathbb{S}$ *with* $U \in \langle\!|\alpha|\!\rangle(s)$ *Eloise has a strategy* $f_\exists$ *such that for each finite* $\mathcal{E}$*-play* Π *starting at* $(s, \langle\alpha\rangle\psi)$ *and following* $f_\exists$ *either Abelard gets stuck or* Π *reaches a state* $(s', \xi') \in S \times \mathcal{F}$ *that satisfies one of the following conditions: (i)* $\xi' \lhd \alpha$ *and* $s' \in [\![\xi']\!]$, *or (ii)* $\xi' = \psi$ *and* $s' \in U$.

Proposition 2. *Let* $\mathbb{S} = (S, \gamma, \Upsilon)$ *be a game model and consider the game* $\mathcal{E} = \mathcal{E}(\mathbb{S}, \varphi)$ *for some* $\varphi \in \mathcal{F}$. *There is a strategy* $f_\exists$ *for Eloise that is winning for Eloise for all game positions* (s, ψ) *such that* $s \in [\![\psi]\!]_\mathbb{S}$.

In summary, Propositions 1 and 2 imply that our game semantics for game logic is adequate:

Theorem 1. *Let* $\mathbb{S} = (S, \gamma, \Upsilon)$ *be a game model and consider the game* $\mathcal{E} = \mathcal{E}(\mathbb{S}, \varphi)$ *for some* $\varphi \in \mathcal{F}$. *Then for all positions* (s, ψ) *in* $\mathcal{E}$ *we have* $(s, \psi) \in \mathrm{Win}_\exists(\mathcal{E})$ *iff* $\mathbb{S}, s \models \psi$.

4 Syntax Graphs

In this section we introduce syntax graphs which we then use later to provide an automata-theoretic characterisation of game logic. Syntax graphs are a generalisation of syntax trees that allow cycles and sharing of subterms. Another perspective is that they are a graph-based description of the alternating tree automata from [19,29]. We discuss the precise connection after the definition of syntax graphs and their game semantics.

4.1 Graph Basics

We first recall some basic notions and fix notation. A graph is a pair $\mathbb{G} = (V, E)$ where V is a set of vertices V and $E \subseteq V \times V$ is a set of edges. We will use the following notation: vEw iff $(v, w) \in E$ iff $w \in E(v)$, and call w a successor of v.

Let $\mathbb{G} = (V, E)$ be a graph. A *path* p in $\mathbb{G}$ is a sequence of vertices $p = v_1 \dots v_n$ such that $v_i E v_{i+1}$ for all $i < n$. We say that v_n is *reachable* from v_1 if a path $p = v_1 \dots v_n$ exists. Note that every vertex is always reachable from itself. A *cycle* $c = v_1 \dots v_n$ is any path such that $v_1 = v_n$ and $n \geq 2$.

A path $p = v_1 \dots v_n$ is *simple* if all the v_i for $i \leq n$ are distinct. A cycle $c = v_1 \dots v_n$ is *simple* if all the v_i for $i < n$ are distinct Every path can be *contracted* to a simple path with the same start and end points, To see how this works consider a path p that contains a repetition of some vertex $u \in V$. This means that p is of the form $p = qumur$, for paths q, m and r. We contract p to the path qur with the same starting and end points, in which there is one less occurrence of u. We can repeat this procedure until we obtain a simple path.

A *pointed* graph $\mathbb{G} = (V, E, v_I)$ is a graph (V, E) together with a $v_I \in V$ that we call the *initial* vertex of $\mathbb{G}$. If $\mathbb{G}$ is a graph (V, E) or a pointed graph (V, E, v) and v_I is a vertex in $\mathbb{G}$, we define $\mathbb{G}@v_I = (V', E', v_I)$ to be the *subgraph generated by v_I in* $\mathbb{G}$, i.e., V' is the set of vertices that are reachable from v_I and $E' = E \cap (V' \times V')$.

A pointed graph $\mathbb{G} = (V, E, v_I)$ is *reachable* if every $v \in V$ is reachable from v_I. Note that $\mathbb{G}@v_I$ is always reachable.

4.2 Syntax Graphs

We define the following sets of label symbols: $\mathsf{Lit} = \mathsf{Lb}_0 := \{p, \neg p \mid p \in \mathsf{Prop}\}$, $\mathsf{Latt} = \mathsf{Lb}_2 := \{\wedge, \vee\}$ and $\mathsf{Mod} = \mathsf{Lb}_1 := \{\langle g \rangle \mid g \in \mathsf{Gam}\} \cup \{\langle g^d \rangle \mid g \in \mathsf{Gam}\}$. The labels $\mathsf{Lb}_0, \mathsf{Lb}_1, \mathsf{Lb}_2$ can be given an arity in the expected manner, namely, for $l \in \mathsf{Lb}i$, $\mathsf{arity}(l) = i$. We let $\mathsf{Lb} := \mathsf{Lb}_0 \cup \mathsf{Lb}_1 \cup \mathsf{Lb}_2$.

Definition 10. *A* syntax graph $\mathbb{G} = (V, E, L, \Omega)$ *is a finite graph* (V, E) *together with a* labelling function $L : V \to \mathsf{Lb}$ *and a partial* priority function $\Omega : V \rightharpoonup \omega$ *satisfying the following two conditions:*

(arity condition). *For all* $v \in V$, $|E(v)| = \mathsf{arity}(L(v))$.
(priority condition). *On every simple cycle of* (V, E) *there is at least one vertex on which* Ω *is defined.*

Later we will show that formulas correspond to syntax graphs, and game terms correspond to syntax graphs with a special atomic proposition that marks an "exit" from the graph. The idea is that a game term α is viewed as the modality $\langle\alpha\rangle$ which still needs a formula φ in order to become a formula $\langle\alpha\rangle\varphi$, and an exit marks a place in the graph where φ can be inserted.

Definition 11. *A proposition letter e is an* exit *of a syntax graph $\mathbb{G} = (V, E, L, \Omega)$ if there is a vertex $v \in V$ with $L(v) = e$ and there is no $v \in V$ with $L(v) = \neg e$.*

We say that a proposition letter p is reachable *from a vertex v in $\mathbb{G}$ if there is some vertex u that is reachable from v in $\mathbb{G}$ with $L(u) = p$ or $L(u) = \neg p$. The* priority *of a path (or cycle) $p = v_1 \dots v_n$ is defined by*

$$\Omega(p) = \max\left(\{-1\} \cup \{\Omega(v_i) \mid 1 \leq i \leq n\}\right),$$

i.e., $\Omega(p) = -1$ if Ω is undefined on all the v_i.

Due to the close connection between formulas and syntax graphs, we can define an acceptance game for syntax graphs in essentially the same way as in Definition 8, using that successors in the syntax graph can be viewed as subformulas.

Definition 12. *Let $\mathbb{G} = (V, E, L, \Omega, v_I)$ be a pointed syntax graph and $\mathbb{S} = (S, \gamma, \Upsilon, s_I)$ be a pointed game model. We define the* acceptance game *$\mathcal{A} = \mathcal{A}(\mathbb{G}, \mathbb{S})$ as a parity game with the game board as specified in Fig. 2, initial position (v_I, s_I) and priority function $\Omega_{\mathcal{A}}$ such that $\Omega_{\mathcal{A}}(v, s) = \Omega(v)$ if $\Omega(v)$ is defined and $\Omega_{\mathcal{A}}(v, s) = 0$ otherwise. If Eloise has a winning strategy in the game $\mathcal{A}(\mathbb{G}, \mathbb{S})$ then we say that $\mathbb{G}$* accepts *$\mathbb{S}$. We also write $\mathbb{S}, s \models \mathbb{G}$ to mean that Eloise has a winning strategy in the game $\mathcal{A}(\mathbb{G}, \mathbb{S})$ starting from position (v_I, s).*

Given a pointed syntax graph $\mathbb{G}$ and a formula φ, we write $\mathbb{G} \equiv \varphi$ if for all $\mathbb{S}$, Eloise has a winning strategy in $\mathcal{E}(\mathbb{S}, \varphi)$ iff she has one in $\mathcal{A}(\mathbb{G}, \mathbb{S})$.

Position b	$P(b)$	Moves $E[b]$
$(v, s), L(v) = p, s \in \Upsilon(p)$	$\forall$	$\emptyset$
$(v, s), L(v) = p, s \notin \Upsilon(p)$	$\exists$	$\emptyset$
$(v, s), L(v) = \neg p, s \in \Upsilon(p)$	$\exists$	$\emptyset$
$(v, s), L(v) = \neg p, s \notin \Upsilon(p)$	$\forall$	$\emptyset$
$(v, s), L(v) = \wedge$	$\forall$	$\{(w_0, s), (w_1, s)\}$, where $E(v) = \{w_0, w_1\}$
$(v, s), L(v) = \vee$	$\exists$	$\{(w_0, s), (w_1, s)\}$, where $E(v) = \{w_0, w_1\}$
$(v, s), L(v) = \langle g\rangle$	$\exists$	$\{(v, U) \mid U \in \langle\!\langle g\rangle\!\rangle(s)\}$
$(v, U), L(v) = \langle g\rangle$	$\forall$	$\{(w, s) \mid s \in U, L(v) = \{w\}\}$
$(v, s), L(v) = \langle g^d\rangle$	$\forall$	$\{(v, U) \mid U \in \langle\!\langle g\rangle\!\rangle(s)\}$
$(v, U), L(v) = \langle g^d\rangle$	$\exists$	$\{(w, s) \mid s \in U, L(v) = \{w\}\}$

Fig. 2. Game board of the acceptance game $\mathcal{A}(\mathbb{G}, \mathbb{S})$

A syntax graph is essentially a multi-modal version of an alternating tree automaton (ATA) with partial priority function as described in [29, Sect. 2.2.5]. Namely, taking the transition graph of an ATA as defined in [29, Sect. 2.2.4] and equipping this graph with the evident labelling function, yields a syntax graph. Conversely, given a syntax graph one constructs for each vertex a transition condition from its label and successors in the obvious manner. If desired, a partial priority function Ω can be made into a total map Ω' by defining $\Omega'(v) = \Omega(v)+2$ if $v \in V_P$ and $\Omega'(v) = 0$ otherwise. One easily adapts the notion of a run on a pointed Kripke structure from [29] to a run on a pointed game model (by dealing with modal transition conditions as in the modal positions of Definition 12) such that there exists an accepting run for the ATA on $\mathbb{S}$ iff Eloise has a winning strategy in the acceptance game for the corresponding syntax graph on $\mathbb{S}$.

As described in [29, Sect. 2.2.5] and in more detail in [19, Sect. 9.3.4] ATAs can be generalised to allow complex transition conditions (i.e. arbitrary formulas) without increasing their expressive power. The basic idea in transforming an ATA with complex transition condition into an equivalent ATA is to introduce new states for each node in the syntax tree of the transition conditions.

Monotone modal automata are obtained by instantiating the definition of Λ-automaton from [11] with the function $\mathcal{M}^{\mathsf{Gam}}$ and taking Λ to be a suitable set of predicate liftings. Monotone modal automata and their unguarded variants are expressively complete for the monotone (multi-modal) μ-calculus. On the other hand, unguarded monotone modal automata are essentially the same as ATAs with complex transition condition (running on monotone neighbourhood models for a multi-modal signature), hence by the above transformation, unguarded monotone modal automata can be viewed as syntax graphs, and vice versa.

We have chosen to work with syntax graphs rather than ATAs or monotone modal automata, since we characterise the game logic fragment mainly in terms of the graph structure. In the following section, we identify a class **GG** of syntax graphs that correspond to game logic formulas. By the correspondence just outlined, we can define game automata as those unguarded monotone modal automata for which the corresponding syntax graph (ATA) is in **GG**.

5 The Game Logic Fragment

In this section we define game logic graphs, which are a class of syntax graphs that has the same expressivity over neighbourhood frames as formulas in game logic. After giving the definition of game logic graphs, we show that for each game logic formula there is a game logic graph that accepts a pointed game model iff the formula is true at the model and, vice versa, for every game logic graph there is a game logic formula that is true at a pointed game model iff the game logic graphs accepts the model.

5.1 Game Logic Graphs

The idea behind the definition of game logic graphs is that cycles in the graph correspond to formulas of the form $\langle \alpha^* \rangle \varphi$ and $\langle \alpha^\times \rangle \varphi$. Consider e.g. the axiom for

$\langle\alpha^*\rangle\varphi$ (in Lemma 4). We see that the vertex v corresponding to the disjunction in $\varphi \vee \langle\alpha\rangle\langle\alpha^*\rangle\varphi$ has a special role as a vertex on the corresponding cycle. Namely, let v_l and v_r be the two successors of v where going to v_l means *leaving* the cycle (going to subformula φ) and going to v_r means *remaining* on the cycle (going to subformula $\langle\alpha\rangle\langle\alpha^*\rangle\varphi$). We will refer to this v as the *head* of the cycle corresponding to $\langle\alpha^*\rangle\varphi$. If the cycles in the syntax graph arise from a nesting of fixpoint formulas, and Ω is the parity function of some formula (cf. Definition 7), then certain conditions will need to hold for the cycles and Ω. This is made precise in the following definition.

Definition 13. *Given a syntax graph $\mathbb{G} = (V, E, L, \Omega)$ in which Ω is injective, we let $h := \Omega^{-1} : \mathrm{ran}(\Omega) \to V$ denote the inverse of Ω on its range. We use the abbreviation $h_n := h(n)$ and call h_n the* head of priority n*. Whenever we write h_n, we presuppose that $n \in \mathrm{ran}(\Omega)$.*

A game logic graph *is a syntax graph $\mathbb{G} = (V, E, L, \Omega)$ in which Ω is injective and the following conditions hold for all $n \in \mathrm{ran}(\Omega)$:*

(parity). *$L(h_n) = \vee$ if n is odd and $L(h_n) = \wedge$ if n is even.*
(head). *There are maps $r, l : \mathrm{ran}(\Omega) \to V$, for which we also use the abbreviations $r_n := r(n)$ and $l_n := l(n)$, such that $E(h_n) = \{l_n, r_n\}$ and*
  (leave). *For all simple paths $p = l_n \dots h_n$ we have that $\Omega(p) > n$.*
  (remain). *There is no simple path $h_n r_n \dots h_m$ for any $m > n$.*

A game logic graph with exit *is a syntax graph with exit $\mathbb{G} = (V, E, L, \Omega, e)$ for which (V, E, L, Ω) is a game logic graph that additionally satisfies:*

(exit). *For all $n \in \mathrm{ran}(\Omega)$ and all $v \in V$ with $L(v) = e$, there is no simple path $h_n r_n \dots v$.*

5.2 From Formulas to Game Logic Graphs

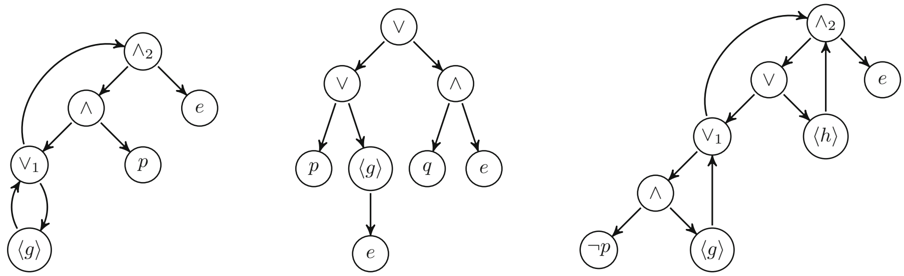
Our first result in characterising the game logic fragment of syntax graphs shows that we can translate game logic formulas into equivalent game logic graphs.

Theorem 2. *For every game $\alpha \in \mathcal{G}_{\mathrm{DNNF}}$ in which the proposition letter e does not occur, there is a pointed syntax graph $\mathbb{G}$ with exit e such that $\mathbb{G} \equiv \langle\alpha\rangle e$. For every game logic formula $\varphi \in \mathcal{F}_{\mathrm{DNNF}}$ there is a pointed syntax graph $\mathbb{G}$ such that $\mathbb{G} \equiv \varphi$.*

The proof of Theorem 2 is by a mutual induction on the structure of games and formulas, and is similar to the construction of a nondeterministic finite automaton from a regular expression [17], that is, we define constructions on syntax graphs that correspond to game operations and logical connectives. The recursive procedure itself is similar to the translation of game logic into the μ-calculus [24], with the difference that we directly translate into syntax graphs instead of formulas of the μ-calculus.

For example, we construct $\mathbb{G}_1 ; \mathbb{G}_2$ where $\mathbb{G}_1$ and $\mathbb{G}_2$ are given by the induction hypothesis by rerouting the edges that went to an exit vertex in $\mathbb{G}_1$ to go to the initial state of $\mathbb{G}_2$. The priority function Ω for $\mathbb{G}_1\,;\mathbb{G}_2$ is unchanged on the $\mathbb{G}_2$ part, but in order to make sure Ω is injective we shift all priority values in $\mathbb{G}_1$ by adding to them a number k that preserves the parity and ensures that all priorities in the $\mathbb{G}_1$ part are higher than those in the $\mathbb{G}_2$ part. The correctness of the construction is proved by constructing winning strategies in the evaluation game from winning strategies in the acceptance game, and vice versa. A detailed proof is provided in [15].

Example 1. Below we show the syntax graphs of some formulas. The initial vertex is the topmost vertex, and priorities are indicated as subscripts on the vertex labels.



$\varphi = \langle (p?; g^*)^\times \rangle e$ $\quad$ $\varphi = \langle (p!; g) \cup q? \rangle e$ $\quad$ $\varphi = \langle (((\neg p)?; g)^* \cup h)^\times \rangle e$

5.3 From Game Logic Graphs to Formulas

We now show how to transform game logic graphs into equivalent game logic formulas.

Theorem 3. *For every pointed game logic graph with exit* $\mathbb{G} = (V, E, L, \Omega, e, v_I)$ *there is a game term* $\delta \in \mathcal{G}$*, not containing* e *and only containing propositional letters that are reachable from* v_I*, such that* $\mathbb{G} \equiv \langle\delta\rangle e$*.*

The proof of Theorem 3 is by induction on the number of heads in the game logic graph. In the base case there are no heads which implies that there are no cycles in the graph, which makes it easy to recursively decompose the graph into a game term. In the inductive step we use a construction that removes some of the edges at the head with the highest priority and thus cutting all cycles that pass through the highest priority head. This allows us to remove the priority from this head and obtain a simpler game logic graph to which we can apply the induction hypothesis. A detailed proof is provided in [15].

Because any propositional letter e that does not occur in $\mathbb{G}$ can be added as an exit to a game logic graph $\mathbb{G}$ we obtain the following corollary from Theorem 3:

Corollary 1. *For every pointed game logic syntax graph* $\mathbb{G}$ *there is a formula* $\varphi \in \mathcal{F}$ *such that* $\mathbb{G} \equiv \varphi$*.*

Example 2. We apply the construction from Theorem 3 to the graph on the left in Example 1. The heads h_1 and h_2 are the disjunction with priority 1 and the conjunction with priority 2, respectively. We start the decomposition at h_2. We then first obtain a game $\delta_2 = \underline{\lambda_2^\times}$, where $\underline{\lambda_2^\times}$ is a dummy game term that is a place holder for the game through the left child of h_2, that describes how to reach the exit from the initial state without iterating at h_2. We also apply the induction hypothesis to obtain a new game λ_2 that describes one iteration from the left node to h_2, which we replace by a fresh exit e'. In this inductive step we then need to cut h_1. At h_1 we have $\delta_1 = \underline{\lambda_1^*} \cap p?\,;p!$ and $\lambda_1 = \langle g \rangle$. We then obtain λ_2 by substituting $\underline{\lambda_1^*}$ in δ_1 with λ_1^* and thus obtain $\lambda_1 = g^* \cap (p?\,;p!)$. Substituting $\lambda_1^\times$ for $\underline{\lambda_1^\times}$ in δ_2 yields the overall game $(g^* \cap (p?\,;p!))^\times$. Hence the game graph is equivalent to the formula $\langle (g^* \cap (p?\,;p!))^\times \rangle e$.

6 Conclusion

We have provided a semantics for game logic in terms of parity games. This was the key to obtain our main technical result, the characterisation of game logic graphs, i.e., a class of parity automata that correspond to game logic formulas.

These automata open several avenues for future research: Firstly, we would like to study normal forms in game logic. In the μ-calculus, automata are the key to obtain the (semi-)disjunctive normal forms of formulas which can be used to prove further results, e.g., completeness, interpolation and the characterisation of the expressivity of the logic [6,18,28]. Our experience suggests that a similar normal form for game logic is out of reach, but a careful analysis of the cycle structure of game logic graphs might yield useful insights concerning the structure of game logic formulas. As a first step in this direction we are currently investigating how to obtain guarded game logic graphs and, consequently, a definition of guarded game logic formulas.

Furthermore, game logic constitutes a very general dynamic logic that makes very few assumptions on the algebraic properties of the modal operators. Therefore we believe that our game logic automata have the potential to help us understand a wider class of automata for families of dynamic logics such as coalgebraic dynamic logics [14] or many-valued dynamic logics as described in [21] or for a combination of these frameworks.

References

1. Berwanger, D.: Game logic is strong enough for parity games. Stud. Logica **75**(2), 205–219 (2003)
2. Bradfield, J., Stirling, C.: Modal μ-calculi. In: Handbook of Modal Logic, pp. 721–756. Elsevier (2006)
3. Carreiro, F., Venema, Y.: PDL inside the μ-calculus: a syntactic and an automata-theoretic characterization. In: Goré, R., Kooi, B., Kurucz, A. (eds.) Advances in Modal Logic, vol. 10, pp. 74–93. College Publications (2014)

4. Carreiro, F.: Fragments of fixpoint logics. Ph.D. thesis, University of Amsterdam (2015)
5. Chellas, B.F.: Modal Logic: An Introduction. Cambridge University Press, Cambridge (1980)
6. d'Agostino, G., Hollenberg, M.: Logical questions concerning the μ-calculus. J. Symbol. Logic **65**, 310–332 (2000)
7. Emerson, E.A., Jutla, C.S.: The complexity of tree automata and logics of programs (extended abstract). In: Proceedings of the 29th Symposium on the Foundations of Computer Science, pp. 328–337. IEEE Computer Society Press (1988)
8. Emerson, E.A., Jutla, C.S., Sistla, P.: On model checking for the μ-calculus and its fragments. Theor. Comput. Sci. **258**, 491–522 (2001)
9. Emerson, E., Jutla, C.: Tree automata, mu-calculus and determinacy. In: Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science (FoCS 1991), pp. 368–377. IEEE (1991)
10. Enqvist, S., Seifan, F., Venema, Y.: Completeness for μ-calculi: a coalgebraic approach. Technical rep. PP-2017-04, ILLC, Universiteit van Amsterdam (2017)
11. Fontaine, G., Leal, R., Venema, Y.: Automata for coalgebras: an approach using predicate liftings. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010. LNCS, vol. 6199, pp. 381–392. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14162-1_32
12. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata, Logics, and Infinite Games. LNCS, vol. 2500. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36387-4
13. Hansen, H.H.: Monotonic modal logic. Master's thesis, University of Amsterdam (2003). iLLC Preprint PP-2003-24
14. Hansen, H.H., Kupke, C.: Weak completeness of coalgebraic dynamic logics. In: Fixed Points in Computer Science (FICS). EPTCS, vol. 191, pp. 90–104 (2015)
15. Hansen, H., Kupke, C., Marti, J., Venema, Y.: Parity games and automata for game logic (extended version) (2017). http://www.arxiv.org
16. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. The MIT Press, Cambridge (2000)
17. Hopcroft, J., Ullman, J.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading (1979)
18. Janin, D., Walukiewicz, I.: Automata for the modal μ-calculus and related results. In: Wiedermann, J., Hájek, P. (eds.) MFCS 1995. LNCS, vol. 969, pp. 552–562. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60246-1_160
19. Kirsten, D.: Alternating tree automata and parity games. In: Grädel, E., Thomas, W., Wilke, T. (eds.) Automata Logics, and Infinite Games. LNCS, vol. 2500, pp. 153–167. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36387-4_9
20. Kozen, D.: Results on the propositional μ-calculus. Theoret. Comput. Sci. **27**, 333–354 (1983)
21. Madeira, A., Neves, R., Martins, M.: An exercise on the generation of many-valued dynamic logics. J. Logic Algebraic Method Program. **85**(5), 1011–1037 (2016)
22. Mostowski, A.: Games with forbidden positions. Technical rep. 78, Instytut Matematyki, Uniwersytet Gdański, Poland (1991)
23. Parikh, R.: The logic of games and its applications. In: Topics in the Theory of Computation. Annals of Discrete Mathematics, vol. 14. Elsevier (1985)
24. Pauly, M.: Logic for social software. Ph.D. thesis, University of Amsterdam (2001)
25. Pauly, M., Parikh, R.: Game logic: an overview. Stud. Logica **75**(2), 165–182 (2003)
26. Venema, Y.: Lectures on the modal μ-calculus (2012). https://staff.science.uva.nl/y.venema/

27. Walukiewicz, I.: On completeness of the mu-calculus. In: Proceedings of the Eighth Annual Symposium on Logic in Computer Science (LICS 1993), pp. 136–146. IEEE Computer Society (1993)
28. Walukiewicz, I.: Completeness of Kozen's axiomatisation of the propositional μ-calculus. Inf. Comput. **157**(1–2), 142–182 (2000). LICS 1995. San Diego, CA
29. Wilke, T.: Alternating tree automata, parity games, and modal μ-calculus. Bull. Belg. Math. Soc. **8**, 359–391 (2001)

Model Checking Against Arbitrary Public Announcement Logic: A First-Order-Logic Prover Approach for the Existential Fragment

Tristan Charrier(✉), Sophie Pinchinat, and François Schwarzentruber

Université de Rennes 1, 35000 Rennes, France
{tristan.charrier,sophie.pinchinat}@irisa.fr,
francois.schwarzentruber@ens-rennes.fr

Abstract. In this paper, we investigate the model checking problem of symbolic models against epistemic logic with arbitrary public announcements and group announcements. We reduce this problem to the satisfiability of Monadic Monadic Second Order Logic (MMSO), the fragment of monadic-second order logic restricted to monadic predicates. In particular, for the case of epistemic formulas in which all arbitrary and group announcements are existential, the proposed reduction lands in monadic first-order logic. We take advantage of this situation to report on few experiments we made with first-order provers.

1 Introduction

In a multi-robot system, agents collect knowledge from what they perceive with their sensors and from the information acquired from some communication channel [31,32]. In order to formalize the notion of knowledge, epistemic modal logics have been developed. For instance, Dynamic epistemic logic [9,43] aims at expressing properties about the knowledge of agents and at modeling information change in multi-agent settings. Public announcement logic PAL [38] is a noticeable fragment of Dynamic epistemic logic, where possible events are public announcements. Since then, variants/extensions of PAL have been developed: typically, arbitrary public announcement logic APAL [6] and group announcement logic GAL [2]. The family of announcement logics has been the subject of much work as they open the way to formal reasoning in many practical applications. We here mention a few, at the intuitive level only. For example, such logics enable one to reason about human/robot interaction via a public channel of communication: message exchanges between robots can be modeled by public announcements when there is common knowledge of the reliability of the network and when it is assumed that messages are received instantaneously [32]. Announcement logics, as well as dynamic epistemic logic, are also relevant in games [34]: in the Battleships, players publicly announce that there is a ship

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 133–152, 2018.
https://doi.org/10.1007/978-3-319-73579-5_9

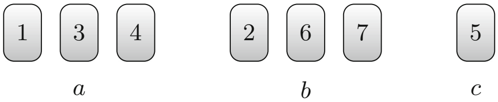


Fig. 1. Example of hands for the Russian cards puzzle

in a given cell. In card games, players often publicly show some cards to other players or announce something. Some issues in security may also be approached with announcement logic: for example, one may wish to verify that no announcement leads the system to a critical/bad state, say, where Intruder knows some secret [17]. Finally, gossip-based algorithms in distributed systems, where agents privately share their secrets in order to achieve shared knowledge of all secrets[1], may be analyzed with announcement logic [28, 41].

In order to get started with announcement logic, we develop the classic Russian card example.

Example 1 (Russian card [42, 44]*).* We consider three agents, a, b and c. Agent a has 3 cards in her hand, b has 3 cards in his hand and c has 1 card in his hand. The cards range from 1 to 7. Given a hand, say as in Fig. 1, the question is whether a and b can publicly announce truthful facts so that they commonly know all players' hands but c not learning any card from a's or b's hands from the course of announcements.

In the case where a and b have 3 cards and c1 card, it is shown in [44] that it is possible for a and b to share information about their hands in any possible configuration in one single public announcement for a and one single public announcement for b.

Of course, a cannot just announce what her hand is, because it would cause c to learn the content of her hand. The trick for a consists in announcing a set of possible hands such that b can deduce what a's hand is, and c cannot. In the example of Fig. 1, if a announces the sentence (Δ) "My hand is either 134, 126, 367, 465 or 275", she ensures that for any possible configuration of hands for b and c, b will always be able to deduce a's hand and c will never deduce any card of a's hand. After a has announced (Δ), b actually knows all hands of the players. Therefore, b announces "c has card 5 is his hand" so that a knows all hands.

Regarding logics APAL and GAL, it has been proved that their satisfiability problem are undecidable [3, 25]. It has been shown that the satisfiability problem with iterations over public announcements is undecidable too [36], so the satisfiability problem with any protocol is also undecidable. Nevertheless, these logics are very relevant for model checking, that is verifying that a given model satisfies a given property. The model checking problem is at the heart of this contribution. Additionally, the setting we consider is the one of *symbolic* models. These models are not specified in extension but described by means of all

[1] In a minimal number of communications.

the possible valuations of a finite set of propositions (each valuation denotes a possible world) and the indistinguishability relations (one for each agent) are specified by *accessibility programs*.

We introduce a second example, the standard muddy children puzzle [43], and we pull its definition to a symbolic model. Both Russian cards and muddy children examples will be useful in the paper.

Example 2 (muddy children). We consider n children playing in their garden. Some of them have mud on their forehead, some have not. Each child can see the others' forehead[2], but she cannot see her own. We suppose that all children are honest and clever. Their father comes to them and says: "At least one of you has mud on her head". Then he repeatedly asks "Does any one of you know for sure whether he/she is muddy?". He stops asking when at least one child tells that she knows.

The solution to this very classic puzzle is that if k children are muddy with $k \leq n$, no child knows its status before round k, and the muddy children know their status in round k[3].

Formally, the initial situation is modeled by a Kripke model containing all combinations of possible children's forehead's status, that is 2^n possible worlds. In a given situation/world, each child considers one other possible world that differs from the current one regarding her own forehead's status. Figure 2 shows the Kripke model for two agents. Proposition p_a stands for "a is muddy" while proposition p_b symmetrically stands for "b is muddy".

Because Kripke models may be large – in the muddy children example the model is exponential in the number of children – many symbolic representations have been considered in the model checking literature (see for example [5]) and more recently in epistemic logic [19,20,40]. We use here the notion of *symbolic accessibility relations* that we call *accessibility programs*, or simply *programs*, that can modify propositional variables. These programs are akin to a dialect used in PDL [24], called *DL-PA*, for "dynamic logic of propositional assignments" [8]. These programs turn out to be the natural way of defining Kripke models. For instance, for the muddy children puzzle, the program of agent a (resp. b) is: *Non-deterministically choose between setting the value of* p_a *(resp.* p_b*) to false or to true.* As observed in [20], the size of a symbolic Kripke model (that is the size needed to describe the collection of agent programs) may be exponentially smaller than the size any equivalent non-symbolic Kripke model[4]. Thus it is polynomial in the number of children in the muddy children's example.

The symbolic model checking of APAL was already studied in [19]. Its complexity was proved to be A_{pol}EXPTIME-complete, and NEXPTIME-complete when restricted to existential arbitrary announcements. Recall that the class A_{pol}EXPTIME[14,15,29] stands for the class of problems decided by alternating Turing machines [16] that run in exponential time but with only a polynomial

[2] Henceforth if there is mud.

[3] Clean children know their status during round $k + 1$.

[4] For non-symbolic Kripke models, the size is the one of its graph.

number of alternations along the computation, hence it is in between EXPTIME and AEXPTIME (=EXPSPACE).

In this paper, instead of building specific algorithms for model checking symbolic models against arbitrary public announcement and group announcement logic (AGPAL, the natural combination of APAL and GAL), we bring closer this logic and first-order logic. More precisely:

1. We show a polynomial reduction from the symbolic model checking[5] against AGPAL to the satisfiability problem of the *monadic monadic second order logic*, written MMSO here, that is the fragment of monadic second order logic where all predicates in the formula are monadic.
2. We prove that this reduction leads to a reduction from the symbolic model checking of *existential* AGPAL[6] (∃AGPAL) to the satisfiability problem of *monadic first-order logic*, that we write MFO. This reduction is supported by the fact that the symbolic model checking against ∃AGPAL and the satisfiability problem of monadic first-order logic are both NEXPTIME-complete (see respectively [19] and [4,33,35]).
3. We build a set of benchmarks for FO provers and report on our experiments.

We claim that the relationship we establish between announcement logics and first-order logic cross-fertilizes two communities: the one in dynamic epistemic logic would benefit from the expertise of researchers in first-order provers in term of efficiency of algorithms and theorem proving techniques; the other community from first-order logic will collect new benchmarks that correspond to instances of the symbolic model checking problem of ∃AGPAL.

The article is organized as follows. In Sect. 2, we recall the setting of MMSO and MFO. Next, in Sect. 3, we describe the language AGPAL and its existential fragment ∃AGPAL. Sections 4 (resp. Sect. 5) is dedicated to the reduction of the symbolic model checking problem against AGPAL (resp. ∃AGPAL) to the satisfiability problem for MMSO (resp. MFO). In Sect. 6, we benefit from the use of FO provers to solve the symbolic model checking problem against ∃AGPAL, and report on our experiments. Finally, we open perspectives for future work in Sect. 7.

In the rest of this paper, we fix a countable set of atomic propositions $\mathtt{AP} = \{p, q, p_1, p_2, \ldots\}$.

2 Brief Recall on First and Second-Order Logics

Monadic monadic second-order logic MMSO and its fragment monadic first-order logic MFO are central in the proposed approach. These monadic fragments of MSO and FO respectively disallow the use of non-unary predicates and of function symbols: MMSO-formulas are thus monadic second-order formulas with first-order and second-order variables but with no occurrence of non-unary predicates; MFO-formulas have only first-order variables. The signature of MMSO

[5] A short way for model checking of symbolic models.

[6] The fragment of AGPAL with only existential arbitrary and group announcements.

mimics the set of atomic propositions $\mathtt{AP}$: to each atomic proposition $p \in \mathtt{AP}$, we introduce a corresponding unary predicate symbol $P(.)$[7].

A model $\mathscr{M}$ of MMSO is a structure $(D, (P^{\mathscr{M}})_{p \in \mathtt{AP}})$ where D is a non-empty domain and each $P^{\mathscr{M}} \subseteq D$. We will use the classical notation of the form $\mathscr{M}[...]$ for the model $\mathscr{M}$ extended with (first-order and second-order) variable assignments: for instance, $\mathscr{M}[\mathsf{x} \leftarrow e, \mathsf{y} \leftarrow e', \mathsf{X} \leftarrow D', \mathsf{Y} \leftarrow D'']$ is the model $\mathscr{M}$ in which first-order variables x and y are interpreted by element $e \in D$ and $e' \in D$ respectively, and second-order variables X and Y are interpreted by element $D' \subseteq D$ and $D'' \subseteq D$ respectively.

Regarding the properties of MMSO and MFO, it is known that the satisfiability problem of a MFO-formula is NEXPTIME-complete [4,33]. Also, there are plenty of FO provers: Isabelle, iprover, Z3 [21], CVC4 [10]. In particular, the prover iprover won CASC 2016 in EPR division [39].

3 Background on Arbitrary/Group Public Announcement

In this section, we define the logic AGPAL that extends both arbitrary public announcement logic and group announcement logic, as well at its fragment ∃AGPAL. Moreover, we consider symbolic models to interpret these logics, and state the symbolic model checking problem.

3.1 Syntax of AGPAL

Let $\mathtt{AP}$ be a countable set of atomic propositions. Let $\mathtt{Agt}$ be a finite set of agents. We define the logic AGPAL that extends both arbitrary public announcement logic and group announcement logic, but we simply call it *announcement logic.*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \mid \langle \varphi! \rangle \varphi \mid \langle \bullet! \rangle \varphi \mid \langle \bullet!_G \rangle \varphi$$

where p ranges over $\mathtt{AP}$ and a over $\mathtt{Agt}$. Formula $K_a\varphi$ reads as "agent a knows that φ holds". Construction $\langle \psi! \rangle \varphi$ reads as "ψ is true and after having announced ψ, formula φ holds". $\langle \bullet! \rangle \varphi$ reads as "there exists a true formula ψ such that makes φ true after announcing it". Formula $\langle \bullet!_G \rangle \varphi$ reads as "agents of group G can make φ hold by announcing at the same time each a formula she knows". In other words, it means that "there exists a true formula of the form $\bigwedge_{a \in G} K_a \psi_a$ such that make φ hold after announcing it". As usual, we write $(\varphi \vee \psi)$ for $\neg(\neg\varphi \wedge \neg\psi)$, $\hat{K}_a\varphi$ for $\neg K_a \neg\varphi$, $[\psi!]\varphi$ for $\neg\langle \psi! \rangle \neg\varphi$. We concisely write $\langle \psi! \rangle^n \varphi$ for $\langle \psi! \rangle \ldots \langle \psi! \rangle \varphi$ where the announcement of ψ takes place n times.

Example 3 (Muddy children with n children). Suppose that all children are muddy. Formula $\langle \bigvee_{a \in \mathtt{Agt}} p_a! \rangle \langle (\bigwedge_{a \in \mathtt{Agt}} \neg K_a p_a)! \rangle^n \bigwedge_{a \in \mathtt{Agt}} K_a p_a$ states that all children know that they are muddy after the father announces that one of them is muddy and then announces n times that no child knows that she is muddy. It is known that this formula holds in the initial situation of the muddy children puzzle.

[7] We take the convention that atomic propositions are written in lowercase while the corresponding predicates are written in uppercase.

Example 4 (Russian cards). We introduce propositions $p_{i,a}$ for "agent a has card i". Let $\mathtt{AP}_h$ be the set of all propositions $p_{i,a}$, $p_{i,b}$, $p_{i,c}$ for $i \in \{1, \ldots, 7\}$.

Let S_7 be the set of all permutations of $\{1, \ldots, 7\}$. Given $\boldsymbol{h} = (h_1, ..., h_7)$ an element of S_7, we define

$$\varphi_{Rh}(\boldsymbol{h}) = p_{h_1,a} \wedge p_{h_2,a} \wedge p_{h_3,a} \wedge p_{h_4,b} \wedge p_{h_5,b} \wedge p_{h_6,b} \wedge p_{h_7,c} \wedge \bigwedge_{p \in \mathtt{AP}_h \setminus \{p_{h_1,a}, \ldots, p_{h_7,c}\}} \neg p.$$

$\varphi_{Rh}(\boldsymbol{h})$ describes a particular configuration $\boldsymbol{h}$ of the hands for the players. The rules of the game are defined by the formula $\varphi_R = \bigvee_{\boldsymbol{h} \in S_7} \varphi_{Rh}(\boldsymbol{h})$.

The following formula φ_G states that both a and b know the card configurations while c does not:

$$\varphi_G = \bigvee_{\boldsymbol{h} \in S_7} (K_a \varphi_{Rh}(\boldsymbol{h}) \wedge K_b \varphi_{Rh}(\boldsymbol{h}))) \wedge \bigwedge_{p \in \{p_{1,a}, \ldots, p_{7,a}, p_{1,b}, \ldots, p_{7,b}\}} \neg(K_c p) \wedge \neg(K_c \neg p)$$

In the Russian card situation, the goal is to check that $\langle \bullet !_a \rangle \langle \bullet !_b \rangle \varphi_G$ holds.

3.2 Syntax of ∃AGPAL

We now define the fragment ∃AGPAL of AGPAL, where arbitrary and group announcement operators are only existential. Formally, ∃AGPAL is defined by the following grammar.

$$\begin{aligned} \exists\text{AGPAL} \ni \varphi &::= \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \hat{K}_a \varphi \mid \langle \varphi ! \rangle \varphi \mid \langle \bullet ! \rangle \varphi \mid \langle \bullet !_G \rangle \varphi \\ \psi &::= p \mid \neg \psi \mid \psi \vee \psi \mid K_a \psi \end{aligned}$$

where $p \in \mathtt{AP}$ and a is an agent.

Example 5. The formula $\langle \bullet !_a \rangle \langle \bullet !_b \rangle \varphi_G$ given in the Russian card Example is in ∃AGPAL.

Example 6. Formula $K_b \langle \bullet !_a \rangle K_c p$ is not in ∃AGPAL since $\langle \bullet !_a \rangle$ occurs after K_b. Formula $\hat{K}_b \langle \bullet !_a \rangle K_c p$ is in ∃AGPAL.

3.3 Semantics of AGPAL

Formulas of AGPAL are interpreted on classic Kripke models with the *possible world* semantics, widely used in logics of knowledge [23].

Definition 1. *A* Kripke model *is a tuple* $\mathcal{M} = (W, \{\xrightarrow{a}\}_{a \in \mathtt{Agt}}, V)$*, where:*

- W *is the non-empty set of* worlds,
- *for each* $a \in \mathtt{Agt}$, $\xrightarrow{a} \subseteq W \times W$ *is the* accessibility relation *for agent* a*,*
- $V : W \rightarrow 2^{\mathtt{AP}}$ *is the* valuation *on worlds, that reveals the set of propositions that hold.*

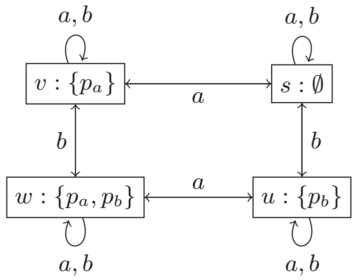


Fig. 2. Kripke model for the muddy children puzzle for two agents

For the sake of generality, we do not require the accessibility relations to be equivalence relations.

Example 7 (muddy children). Figure 2 shows a Kripke model for muddy children with $n = 2$ agents. It has four worlds w, u, v, s. The arrows represent the agents' accessibility relations. For an arbitrary number n of agents, the Kripke model is $\mathcal{M} = (W, \{\xrightarrow{i}\}_{i \in \texttt{Agt}}, V)$ where:

- $W = 2^{\{p_a | a \in \texttt{Agt}\}}$;
- $\xrightarrow{a} = \{(w, u) \mid w \setminus \{p_a\} = u \setminus \{p_a\}\}$;
- $V(w) = w$.

This Kripke model is a graph containing 2^n nodes and $2^{n+1} \times |\texttt{Agt}|$ edges.

Example 8 (Russian cards). A Kripke model corresponding to the Russian card puzzle is $\mathcal{M} = (W, \{\xrightarrow{a}\}_{a \in \texttt{Agt}}, V)$ where:

- W is the set of valuations over $\texttt{AP}_h$ that satisfy formula φ_R; where φ_R is defined in Example 4;
- $w \xrightarrow{a} u$ if $w \cap \{p_{i,a} \mid i \in \{1, \ldots, 7\}\} = u \cap \{p_{i,a} \mid i \in \{1, \ldots, 7\}\}$;
- $V(w) = w$.

Informally, W is the set of all distributions of cards, $w \xrightarrow{a} u$ if a holds the same cards in both worlds w and u, and the valuation $V(w)$ is given by w.

Back to the semantics of AGPAL, we now define the truth conditions for $\mathcal{M}, w \models \varphi$ (read as "formula φ is true in world w of model $\mathcal{M}$") and the restriction $\mathcal{M}^\psi$ of a model $\mathcal{M}$ to a formula ψ.

Definition 2. *We define $\mathcal{M}, w \models \varphi$ (read as "formula φ is true in world w of model $\mathcal{M}$") and $\mathcal{M}^\psi$ (the ψ-*restriction *of $\mathcal{M}$) by mutual induction:*

- *$\mathcal{M}, w \models p$ if $p \in V(w)$;*
- *$\mathcal{M}, w \models (\varphi_1 \land \varphi_2)$ if $\mathcal{M}, w \models \varphi_1$ and $\mathcal{M}, w \models \varphi_2$;*
- *$\mathcal{M}, w \models \neg\varphi$ if $\mathcal{M}, w \not\models \varphi$;*
- *$\mathcal{M}, w \models K_a\varphi$ if for all u such that $w \xrightarrow{a} u$, $\mathcal{M}, u \models \varphi$;*

– *$\mathcal{M}, w \models \langle \psi ! \rangle \varphi$ if $\mathcal{M}, w \models \psi$ and $\mathcal{M}^\psi, w \models \varphi$;*
– *$\mathcal{M}, w \models \langle \bullet ! \rangle \varphi$ if there exists a formula ψ without any occurrence of $\langle \bullet ! \rangle$ or $\langle \bullet !_G \rangle$ such that $\mathcal{M}, w \models \langle \psi ! \rangle \varphi$;*
– *$\mathcal{M}, w \models \langle \bullet !_G \rangle \varphi$ if there exist formulas $(\psi_a)_{a \in G}$ without any occurrence of $\langle \bullet ! \rangle$ or $\langle \bullet !_G \rangle$, such that $\mathcal{M}, w \models \langle \bigwedge_{a \in G} K_a \psi_a ! \rangle \varphi$.*

and $\mathcal{M}^\psi$ is the model $(W^\psi, \{\xrightarrow{a}^\psi\}_{i \in \mathtt{Agt}}, V^\psi)$ where

– *$W^\psi = \{u \in W \mid \mathcal{M}, u \models \psi\}$ (namely, only worlds satisfying ψ are preserved);*
– *$\xrightarrow{a}^\psi = \xrightarrow{a} \cap (W^\psi \times W^\psi)$;*
– *V^ψ is the restriction of V to W^ψ.*

Example 9 (muddy children continued). Let $\mathcal{M}$ be the model of Fig. 2. We have:

$$\mathcal{M}, w \models \langle K_b p_a ! \rangle K_a p_a \wedge \langle \bullet ! \rangle K_a p_a \wedge \langle \bullet !_{\{b\}} \rangle K_a p_a.$$

3.4 Symbolic Presentations of Models

As in [19,20], a *symbolic accessibility relation*, simply called an *accessibility program*, or even a *program*, describes a relation between valuations by executing an explicit sequence of propositional variable assignments. We write $\mathtt{u} \xrightarrow{\pi} \mathtt{v}$ for "$\mathtt{v}$ is a π-successor of $\mathtt{u}$ by π". The syntax for symbolic programs is the following.

$$\pi \ ::= \ p {\leftarrow} \beta \mid \beta ? \mid (\pi ; \pi) \mid (\pi \cup \pi) \mid (\pi \cap \pi) \mid \pi^{-1}$$

where $p \in \mathtt{AP}$, β is a Boolean formula over $\mathtt{AP}$.

The intuitive meaning of the constructions for programs is given in Table 1.

Table 1. Informal meaning of programs

$p {\leftarrow} \beta$	Set p to the value of Boolean formula β
$\beta ?$	Test that β holds
$\pi ; \pi'$	Execute π then execute π'
$\pi \cup \pi'$	Non-deterministically execute π or π'
$\pi \cap \pi'$	Execute the intersection of π and π'
π^{-1}	Converse of π

In what follows, we let $set(p_1, \ldots, p_n)$ denote the program $(p_1 {\leftarrow} \bot \cup p_1 {\leftarrow} \top); \ldots ; (p_n {\leftarrow} \bot \cup p_n {\leftarrow} \top)$ that sets arbitrary values to $p_1, \ldots, p_n$.

Example 10 (Programs for the muddy children example). Since child a sees the forehead of child b but not her own, the program of a amounts to varying the truth value of p_a. That is, $\pi_a = set(p_a)$, and symmetrically for b, $\pi_b = set(p_b)$.

The semantics of programs is defined by induction.

- $\mathtt{w} \xrightarrow{p \leftarrow \beta} \mathtt{u}$ iff ($\mathtt{w} \not\models \beta$ and $\mathtt{u} = \mathtt{w} \setminus \{p\}$) or ($\mathtt{w} \models \beta$ and $\mathtt{u} = \mathtt{w} \cup \{p\}$);
- $\mathtt{w} \xrightarrow{\beta?} \mathtt{u}$ iff $\mathtt{w} \models \beta$ and $\mathtt{w} = \mathtt{u}$;
- $\mathtt{w} \xrightarrow{\pi_1;\pi_2} \mathtt{u}$ iff there exists $\mathtt{v}$ s.t. $\mathtt{w} \xrightarrow{\pi_1} \mathtt{v}$ and $\mathtt{v} \xrightarrow{\pi_2} \mathtt{u}$;
- $\mathtt{w} \xrightarrow{\pi_1 \cup \pi_2} \mathtt{u}$ iff $\mathtt{w} \xrightarrow{\pi_1} \mathtt{u}$ or $\mathtt{w} \xrightarrow{\pi_2} \mathtt{u}$;
- $\mathtt{w} \xrightarrow{\pi_1 \cap \pi_2} \mathtt{u}$ iff $\mathtt{w} \xrightarrow{\pi_1} \mathtt{u}$ and $\mathtt{w} \xrightarrow{\pi_2} \mathtt{u}$;
- $\mathtt{w} \xrightarrow{\pi^{-1}} \mathtt{u}$ iff $\mathtt{u} \xrightarrow{\pi} \mathtt{w}$.

The size of a program is the number of nodes its syntax tree, or equivalently the number of symbols needed to write it, parenthesis omitted. For instance, the program $(p \leftarrow \top) \cup (q?; p \leftarrow \bot)$ has size 10.

As we have seen, the models are symbolically described by means of programs. They yield *symbolic Kripke models* that denote classic Kripke models[8]. However, the former may be exponentially more succinct than the latter.

Definition 3 (Symbolic Kripke models). *A* symbolic Kripke model *is a tuple* $\mathfrak{M} = \langle \mathtt{AP}_M, (\pi_a)_{a \in \mathtt{Agt}} \rangle$ *where* $\mathtt{AP}_M \subseteq \mathtt{AP}$ *is a finite set of atomic propositions and* π_a *is a program over* $\mathtt{AP}_M$ *for each agent* a*.*

Intuitively, each program π_a symbolically describes the accessibility relation for an agent a.

Example 11. The symbolic Kripke model corresponding to the initial situation of the muddy children puzzle is $\mathfrak{M} = \langle \mathtt{AP}_M, (\pi_a)_{a \in \mathtt{Agt}} \rangle$ where:

- $\mathtt{AP}_M = \{p_a \mid a \in \mathtt{Agt}\}$;
- $\pi_a = set(p_a)$ for all agents a.

A pointed symbolic Kripke model is a pair $(\mathfrak{M}, \mathtt{w})$ where $\mathfrak{M} = \langle \mathtt{AP}_M, (\pi_a)_{a \in \mathtt{Agt}} \rangle$ is a symbolic Kripke model and $\mathtt{w}$ is a valuation over $\mathtt{AP}_M$.

We define the explicit Kripke model $\hat{M}(\mathfrak{M})$ associated to the symbolic Kripke model $\mathfrak{M}$: the set of worlds is the set of valuations over $\mathtt{AP}_M$ and the accessibility relation $\xrightarrow{a}$ is the relation $\xrightarrow{\pi_a}$.

Definition 4. *Given a symbolic Kripke model* $\mathfrak{M} = \langle \mathtt{AP}_M, (\pi_a)_{a \in \mathtt{Agt}} \rangle$*, the* Kripke model represented by $\mathfrak{M}$*, noted* $\hat{M}(\mathfrak{M})$ *is the model* $(W, (\xrightarrow{a})_{a \in \mathtt{Agt}}, V)$ *where:*

- $W = \mathcal{V}(\mathtt{AP}_M)$ *where* $\mathcal{V}(\mathtt{AP}_M)$ *is the set of valuations over* $\mathtt{AP}_M$*;*
- $\xrightarrow{a} = \{(\mathtt{w}, \mathtt{u}) \in W^2 \mid \mathtt{w} \xrightarrow{\pi_a} \mathtt{u}\}$*;*
- $V(\mathtt{w}) = \mathtt{w}$.

We write $\mathfrak{M}, w \models \varphi$ instead of $\hat{M}(\mathfrak{M}), w \models \varphi$.

[8] Actually, and *vice versa* [20].

Example 12 (muddy children continued). The Kripke model corresponding to M is $\hat{M}(\mathfrak{M}) = (W, \{\xrightarrow{a}\}_{a\in\mathtt{Agt}}, V)$ where $W = \mathcal{V}(\mathtt{AP}_M)$; for every $a \in \mathtt{Agt}$, $\xrightarrow{a} = \{(\mathtt{w}, \mathtt{u}) \in W^2 \mid \mathtt{w} \setminus p_a = \mathtt{u} \setminus p_a\}$; $V(\mathtt{w}) = \mathtt{w}$. Compared to the Kripke model given in Example 7 whose size is exponential in $|\mathtt{Agt}|$, the symbolic Kripke model is of size $3|\mathtt{Agt}|$.

Example 13 (Russian cards). First we consider the following symbolic Kripke model $\mathfrak{M} = \langle \mathtt{AP}_M, (\pi_a)_{a\in\mathtt{Agt}}\rangle$ where: $\mathtt{AP}_M = \{p_{i,a}, p_{i,b}, p_{i,c} \mid i \in \{1, \dots, 7\}\}$; $\pi_x = set\{p_{i,y} \mid i \in \{1, \dots, 7\}$ and $y \in \{a, b, c\} \setminus \{x\}\}$ for agent $x \in \{a, b, c\}$. The Kripke model corresponding to the initial situation of the Russian card is $\hat{M}(\mathfrak{M})^{\varphi_R}$, which corresponds to model $\hat{M}(\mathfrak{M})$ after the *fake* announcement φ_R that enforces common knowledge that agents a and b have 3 cards each and c has 1.

We finally define the symbolic model checking problem against AGPAL which is central in our contribution, and that we write **AGPAL-mc**.

- Input: a symbolic model $\mathfrak{M}$, a valuation w, and a formula φ;
- Output: yes if $\mathfrak{M}, w \models \varphi$, no otherwise.

4 Announcement Logic into Monadic Monadic Second-Order Logic

We reduce the model checking against AGPAL to the satisfiability problem of MMSO. Intuitively, second-order variables denote current sets of valuations, called *contexts*, and first-order variables denote possible worlds/valuations. We present the reduction in four steps:

1. we define an MMSO-theory that enforce the MMSO-model to contain all valuations (Theorem 1);
2. we translate arbitrary accessibility programs into first-order logic (Theorem 2);
3. we translate AGPAL formulas into MMSO (Theorem 3);
4. we give the reduction of the AGPAL-model checking into the MMSO-satisfiability problem (Theorem 4).

4.1 The Theory of Models of Valuations

In this section, we fix a set of atomic propositions A. Since we evaluate AGPAL-formulas on a symbolic model $\mathfrak{M}$ meant to denote the Kripke model with all valuations, we therefore need to enforce that all such valuations are captured.

Definition 5. *The model of valuations $\mathscr{M}_A$ on A is the structure $\mathscr{M}_A = (D, (P^{\mathscr{M}_A})_{p\in A})$ with D is the domain of all valuations on A and the interpretation of P is defined by as $P^{\mathscr{M}_A}(\mathtt{w})$ iff $p \in \mathtt{w}$.*

In what follows, we write P_A for the set of atomic predicates associated to some $p \in A$.

Definition 6. *Let β be a Boolean formula over A. We define the first-order formula $tr(\beta)(\mathsf{x})$ to be formula β in which each occurrence of $p \in \mathtt{AP}$ is replaced by $P(\mathsf{x})$. Similarly, for a valuation $\mathtt{w}$, we define $tr(\mathtt{w})(\mathsf{x})$ for the formula describing $\mathtt{w}$ where all p are replaced by $P(\mathsf{x})$.*

Example 14. Let $\beta = (p \vee q) \wedge (\neg p \vee q)$. Then $tr(\beta)(\mathsf{x}) = (P(\mathsf{x}) \vee Q(\mathsf{x})) \wedge (\neg P(\mathsf{x}) \vee Q(\mathsf{x}))$.

Example 15. Let $\mathtt{w} = \{p, q\}$ a valuation over $A = \{p, q, r\}$. $tr(\mathtt{w})(\mathsf{x}) = P(\mathsf{x}) \wedge Q(\mathsf{x}) \wedge \neg R(\mathsf{x})$.

We define a theory $\mathscr{T}_A$ such that $\mathscr{M}_A$ satisfies $\mathscr{T}_A$ and every model satisfying $\mathscr{T}_A$ is isomorphic to $\mathscr{M}_A$.

Currently, in an arbitrary structure $(D, (P_i^{\mathscr{M}})_{p_i \in \mathtt{AP}})$, two distinct elements e, e' in D may be such that $e \in P_i^{\mathscr{M}}$ iff $e' \in P_i^{\mathscr{M}}$ for all $p_i \in \mathtt{AP}$. To prevent it, we define $\varphi_{unique} = \forall \mathsf{x} \forall \mathsf{y} (\mathsf{x} = \mathsf{y}) \leftrightarrow \bigwedge_{p \in A} (P(\mathsf{x}) \leftrightarrow P(\mathsf{y}))$. It says that two elements satisfy the same predicates (i.e. are the same valuation) iff they are equal. We define too φ_{exists} says that for each valuation, for each atomic proposition p, there exists another valuation that differs only on p. In other words, $\varphi_{exists} = \forall \mathsf{x} \bigwedge_{p \in A} \left(\exists \mathsf{y} \left((P(\mathsf{x}) \leftrightarrow \neg P(\mathsf{y})) \wedge \bigwedge_{q \in A, q \neq p} (Q(\mathsf{x}) \leftrightarrow Q(\mathsf{y})) \right) \right)$, imposing all valuations to appear in the model.

By letting $\mathscr{T}_A = \{\varphi_{unique}, \varphi_{exists}\}$, we get the following.

Theorem 1. *For all* MMSO*-models $\mathscr{M}$, we have $\mathscr{M} \models \mathscr{T}_A$ iff $\mathscr{M}$ is isomorphic to $\mathscr{M}_A$.*

Proof. $\Leftarrow$: It is sufficient to prove that $\mathscr{M}_A \models \mathscr{T}_A$:

- $\mathscr{M}_A \models \varphi_{unique}$ because each valuation is represented exactly one time in D and by Definition 5, P mimics the role of the atomic propositions in the valuations.
- $\mathscr{M}_A \models \varphi_{exists}$ because all valuations are represented in D.

Therefore $\mathscr{M}_A \models \mathscr{T}_A$ and thus $\mathscr{M} \models \mathscr{T}_A$.

$\Rightarrow$: Let $\mathscr{M}$ be such that $\mathscr{M} \models \mathscr{T}_A$. Let D' be the domain of $\mathscr{M}$ and P' be the monadic predicates of $\mathscr{M}$. We define the mapping $f : D' \to D$ such that for all $e \in D$, $f(e)$ is the valuation $\{p \mid e \in P'\} \in D$. We conclude by showing that f is an isomorphism.

- f is injective: if $f(e) = f(e')$, it means that for all P, $e \in P^{\mathscr{M}}$ iff $e' \in P^{\mathscr{M}}$. With $\mathscr{M} \models \varphi_{unique}$, we conclude that $e = e'$.
- f is surjective: let $\mathtt{w}$ be an element of D. As D' is non-empty, let e be in D'. As $\mathscr{M} \models \varphi_{exists}$, we can, from e, guarantee the existence of an element e' of D' such that $f(e') = \mathtt{w}$.

From Theorem 1, we obtain the following.

Corollary 1. *Let φ be an* MMSO*-formula. Then $\mathscr{M}_A \models \varphi$ if, and only if, $\mathscr{T}_A \wedge \varphi$ is* MMSO*-satisfiable.*

4.2 From Programs to FO-Formulas

Definition 7. *Let π be a program and x, y be two first-order variables. We define the first-order formula $\pi(\mathsf{x}, \mathsf{y})$ by induction π as follows:*

$$\begin{array}{ll} (p \leftarrow \beta)(\mathsf{x}, \mathsf{y}) & = (P(\mathsf{y}) \leftrightarrow tr(\beta)(\mathsf{x})) \wedge \bigwedge_{q \in A, q \neq p} (Q(\mathsf{x}) \leftrightarrow Q(\mathsf{y})); \\ \beta?(\mathsf{x}, \mathsf{y}) & = tr(\beta)(\mathsf{x}) \wedge (\mathsf{x} = \mathsf{y}); \\ (\pi_1; \pi_2)(\mathsf{x}, \mathsf{y}) & = \exists \mathsf{z}\ \pi_1(\mathsf{x}, \mathsf{z}) \wedge \pi_2(\mathsf{z}, \mathsf{y}). \\ (\pi_1 \cup \pi_2)(\mathsf{x}, \mathsf{y}) & = \pi_1(\mathsf{x}, \mathsf{y}) \vee \pi_2(\mathsf{x}, \mathsf{y}); \\ (\pi_1 \cap \pi_2)(\mathsf{x}, \mathsf{y}) & = \pi_1(\mathsf{x}, \mathsf{y}) \wedge \pi_2(\mathsf{x}, \mathsf{y}); \\ \pi^{-1}(\mathsf{x}, \mathsf{y}) & = \pi(\mathsf{y}, \mathsf{x}). \end{array}$$

The formula $\pi(\mathsf{x}, \mathsf{y})$ expresses that y is a π-successor of x. It should be noticed that formulas $\pi(\mathsf{x}, \mathsf{y})$ are in MFO, although the notation might be misleading. Formally:

Theorem 2. *For all worlds $\mathtt{w}, \mathtt{u}$ and π, $\mathtt{w} \xrightarrow{\pi} \mathtt{u}$ if, and only if, $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models \pi(\mathsf{x}, \mathsf{y})$.*

Proof. By induction on π.

- $\pi = p \leftarrow \beta$:
  $\mathtt{w} \xrightarrow{p \leftarrow \beta} \mathtt{u}$ iff ($p \in \mathtt{u}$ iff $\mathtt{w} \models \beta$) and for all $q \neq p$, ($q \in \mathtt{w}$ iff $q \in \mathtt{u}$).
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models P(\mathsf{y}) \leftrightarrow tr(\beta)(\mathsf{x})$ and for all $q \neq p$, $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models Q(\mathsf{x}) \leftrightarrow Q(\mathsf{y})$.
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models (p \leftarrow \beta)(\mathsf{x}, \mathsf{y})$.
- $\pi = \beta?$:
  $\mathtt{w} \xrightarrow{\beta?} \mathtt{u}$ iff $\mathtt{w} = \mathtt{u}$ and $\mathtt{w} \models \beta$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models (\mathsf{x} = \mathsf{y})$ and $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models tr(\beta)(\mathsf{x})$.
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models \beta?(\mathsf{x}, \mathsf{y})$.
- $\pi = \pi_1; \pi_2$:
  $\mathtt{w} \xrightarrow{\pi_1; \pi_2} \mathtt{u}$ iff there exists $\mathtt{v}$ such that $\mathtt{w} \xrightarrow{\pi_1} \mathtt{v}$ and $\mathtt{v} \xrightarrow{\pi_2} \mathtt{u}$
  iff there exists $\mathtt{v}$ such that $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}, \mathsf{z} \leftarrow \mathtt{v}] \models \pi_1(\mathsf{x}, \mathsf{z}) \wedge \pi_2(\mathsf{z}, \mathsf{y})$.
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models (\pi_1; \pi_2)(\mathsf{x}, \mathsf{y})$.
- $\pi = \pi_1 \cup \pi_2$:
  $\mathtt{w} \xrightarrow{\pi_1 \cup \pi_2} \mathtt{u}$ iff $\mathtt{w} \xrightarrow{\pi_1} \mathtt{u}$ or $\mathtt{w} \xrightarrow{\pi_2} \mathtt{u}$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models \pi_1(\mathsf{x}, \mathsf{y})$ or $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models \pi_2(\mathsf{x}, \mathsf{y})$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models (\pi_1 \cup \pi_2)(\mathsf{x}, \mathsf{y})$.
- $\pi = \pi_1 \cap \pi_2$:
  $\mathtt{w} \xrightarrow{\pi_1 \cap \pi_2} \mathtt{u}$ iff $\mathtt{w} \xrightarrow{\pi_1} \mathtt{u}$ and $\mathtt{w} \xrightarrow{\pi_2} \mathtt{u}$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models \pi_1(\mathsf{x}, \mathsf{y})$ and $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models \pi_2(\mathsf{x}, \mathsf{y})$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models (\pi_1 \cap \pi_2)(\mathsf{x}, \mathsf{y})$.
- $\pi = \pi'^{-1}$:
  $\mathtt{w} \xrightarrow{\pi'^{-1}} \mathtt{u}$ iff $\mathtt{u} \xrightarrow{\pi'} \mathtt{w}$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models \pi(\mathsf{y}, \mathsf{x})$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}] \models \pi'^{-1}(\mathsf{x}, \mathsf{y})$.

4.3 From AGPAL-Formulas to MMSO-Formulas

In the following definition, we define $tr_{\mathsf{X}}(\varphi)(\mathsf{x})$ to be the translation of the AGPAL-formula φ, where x is a first-order variable representing the valuation in which the formula φ is evaluated and X is a second-order variable representing the context (namely, the set of valuations that survived the previous announcements). Both variables x and X are the sole free variables of $tr_{\mathsf{X}}(\varphi)(\mathsf{x})$.

Definition 8. *Let $\mathfrak{M} = \langle AP_M, (\pi_a)_{a \in Agt} \rangle$ be a symbolic model, φ be a* AGPAL-*formula, X be a second-order variable, and x be a first-order variable. We define the* MMSO-*formula $tr_{\mathsf{X}}(\varphi)(\mathsf{x})$ by induction over φ, with the notation $\mathsf{Y} \subseteq \mathsf{X}$ for $\forall \mathsf{x}(\mathsf{Y}(\mathsf{x}) \rightarrow \mathsf{X}(\mathsf{x}))$.*

$$
\begin{array}{ll}
tr_{\mathsf{X}}(p)(\mathsf{x}) & = P(\mathsf{x}); \\
tr_{\mathsf{X}}(\neg\varphi)(\mathsf{x}) & = \neg tr_{\mathsf{X}}(\varphi)(\mathsf{x}); \\
tr_{\mathsf{X}}(\varphi_1 \vee \varphi_2)(\mathsf{x}) & = tr_{\mathsf{X}}(\varphi_1)(\mathsf{x}) \vee tr_{\mathsf{X}}(\varphi_2)(\mathsf{x}); \\
tr_{\mathsf{X}}(K_a\varphi)(\mathsf{x}) & = \forall \mathsf{y}\ [(\mathsf{X}(\mathsf{y}) \wedge \pi_a(\mathsf{x},\mathsf{y})) \rightarrow tr_{\mathsf{X}}(\varphi)(\mathsf{y})]; \\
tr_{\mathsf{X}}(\langle\varphi!\rangle\psi)(\mathsf{x}) & = \exists \mathsf{Y}\ (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \leftrightarrow (\mathsf{X}(\mathsf{y}) \wedge tr_{\mathsf{X}}(\psi)(\mathsf{y}))) \wedge \mathsf{Y}(\mathsf{x}) \wedge tr_{\mathsf{Y}}(\varphi)(\mathsf{x}); \\
tr_{\mathsf{X}}(\langle\bullet!\rangle\varphi)(\mathsf{x}) & = \exists \mathsf{Y}\ \mathsf{Y} \subseteq \mathsf{X} \wedge \mathsf{Y}(\mathsf{x}) \wedge tr_{\mathsf{Y}}(\varphi)(\mathsf{x}); \\
tr_{\mathsf{X}}((\langle\bullet!_G\rangle\varphi))(\mathsf{x}) & = \exists \mathsf{Y}\ \mathsf{Y} \subseteq \mathsf{X} \wedge \mathit{isGroupAnnouncement}_G(\mathsf{Y}) \wedge \mathsf{Y}(\mathsf{x}) \wedge tr_{\mathsf{Y}}(\varphi)(\mathsf{x}).
\end{array}
$$

where $\mathit{isGroupAnnouncement}_G(\mathsf{Y}) = \bigwedge_{a \in G} \forall \mathsf{x}\ (\forall \mathsf{y}\ \pi_a(\mathsf{x},\mathsf{y}) \rightarrow (\exists \mathsf{z}\ \pi_a(\mathsf{z},\mathsf{y}) \wedge \mathsf{Y}(\mathsf{z}))) \rightarrow \mathsf{Y}(\mathsf{x})$.

Formula $tr_{\mathsf{X}}(K_a\varphi)(\mathsf{x})$ mimics the standard translation of modal logic into first-order logic ([11], p. 84), except that we use the MFO-formula $\pi_a(\mathsf{x},\mathsf{y})$ instead of $R_a(\mathsf{x},\mathsf{y})$. In formula $tr_{\mathsf{X}}(\langle\varphi!\rangle\psi)(\mathsf{x})$, we ask for the existence of a context Y that corresponds to the set of valuations in which ψ holds ($\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \leftrightarrow (\mathsf{X}(\mathsf{y}) \wedge tr_{\mathsf{X}}(\psi)(\mathsf{y}))$), that contains x ($\mathsf{Y}(\mathsf{x})$) and where φ holds. Formula $tr_{\mathsf{X}}(\langle\bullet!\rangle\varphi)(\mathsf{x})$ is similar to formula $tr_{\mathsf{X}}(\langle\varphi!\rangle\psi)(\mathsf{x})$, except that, as the announcement is arbitrary, we only impose that the context Y is included in X. Formula $tr_{\mathsf{X}}((\langle\bullet!_G\rangle\varphi))(\mathsf{x})$ is similar to $tr_{\mathsf{X}}(\langle\bullet!\rangle\varphi)(\mathsf{x})$ but we impose that the announcement is a group announcement. This constraint is guaranteed by formula $\mathtt{isGroupAnnouncement}_G(\mathsf{Y})$ that is a characterization of submodels generated by a group announcement.

We now state and prove the correctness of the translation.

Theorem 3. *Let $\mathfrak{M}$ be a symbolic model on A, φ be an* AGPAL-*formula on A and $\mathtt{w} \in \mathfrak{M}$. Let $D_{\mathfrak{M}}$ be the set of valuations of $\mathfrak{M}$. Then $\mathfrak{M}, \mathtt{w} \models \varphi$ iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\varphi)(\mathsf{x})$.*

Proof. By induction on φ.

- $\varphi = p$:
  $\mathfrak{M}, \mathtt{w} \models \varphi$ iff $p \in \mathtt{w}$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models P(\mathsf{x})$
- $\varphi = \neg\psi$:
  $\mathfrak{M}, \mathtt{w} \models \neg\psi$ iff $\mathfrak{M}, \mathtt{w} \not\models \psi$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \not\models tr_{\mathsf{X}}(\varphi)(\mathsf{x})$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \neg tr_{\mathsf{X}}(\varphi)(\mathsf{x})$

- $\varphi = \varphi_1 \vee \varphi_2$:
  $\mathfrak{M}, \mathtt{w} \models \varphi_1 \vee \varphi_2$ iff $\mathfrak{M}, \mathtt{w} \models \varphi_1$ or $\mathfrak{M}, \mathtt{w} \models \varphi_2$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\varphi_1)(\mathsf{x})$ or $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\varphi_2)(\mathsf{x})$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\varphi_1)(\mathsf{x}) \vee tr_{\mathsf{X}}(\varphi_2)(\mathsf{x})$
- $\varphi = (K_a\varphi)$:
  $\mathfrak{M}, \mathtt{w} \models (K_a\psi)$ iff for all $\mathtt{u} \in D_{\mathfrak{M}}$ such that $\mathtt{w} \xrightarrow{\pi_a} \mathtt{u}$, $\mathfrak{M}, \mathtt{u} \models \psi$
  iff for all $\mathtt{u} \in D_{\mathfrak{M}}$ such that $\mathtt{w} \xrightarrow{\pi_a} \mathtt{u}$, $\mathscr{M}_A[\mathsf{y} \leftarrow \mathtt{u}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\psi)(\mathsf{y})$
  iff for all $\mathtt{u} \in D_{\mathfrak{M}}$ such that $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \pi_a(\mathsf{x}, \mathsf{y})$,
  $\mathscr{M}_A[\mathsf{y} \leftarrow \mathtt{u}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\psi)(\mathsf{y})$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \forall \mathsf{y}\ (\mathsf{X}(\mathsf{y}) \wedge \pi_a(\mathsf{x}, \mathsf{y}) \rightarrow tr_{\mathsf{X}}(\varphi)(\mathsf{y}))$
- $\varphi = (\langle \chi ! \rangle \psi)$:
  $\mathfrak{M}, \mathtt{w} \models (\langle \chi ! \rangle \psi)$ iff $\mathfrak{M}, \mathtt{w} \models \chi$ and $\mathfrak{M}^{\chi}, \mathtt{w} \models \psi$
  iff $\mathfrak{M}, \mathtt{w} \models \chi$; for all $\mathtt{u}$, ($\mathtt{u} \in D_{\mathfrak{M}^{\chi}}$ iff $\mathtt{u} \in D_{\mathfrak{M}}$ and $\mathfrak{M}, \mathtt{u} \models \chi$); and $\mathfrak{M}^{\chi}, \mathtt{w} \models \psi$
  iff $\mathtt{w} \in D_{\mathfrak{M}^{\chi}}$; and for all $\mathtt{u}$, ($\mathscr{M}_A[\mathsf{y} \leftarrow \mathtt{u}, \mathsf{Y} \leftarrow D_{\mathfrak{M}^{\chi}}] \models \mathsf{Y}(\mathsf{y})$ iff
  $\mathscr{M}_A[\mathsf{y} \leftarrow \mathtt{u}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \mathsf{X}(\mathsf{y})$ and $\mathscr{M}_A[\mathsf{y} \leftarrow \mathtt{u}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\chi)(\mathsf{y}))$;
  and $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{Y} \leftarrow D_{\mathfrak{M}^{\chi}}] \models tr_{\mathsf{X}}(\psi)(\mathsf{y}))$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D_{\mathfrak{M}^{\chi}}] \models \mathsf{Y}(\mathsf{x}) \wedge (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \leftrightarrow (\mathsf{X}(\mathsf{y}) \wedge tr_{\mathsf{X}}(\psi)(\mathsf{y}))) \wedge tr_{\mathsf{Y}}(\chi)(\mathsf{x})$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \exists \mathsf{Y}\ \mathsf{Y}(\mathsf{x}) \wedge (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \leftrightarrow (\mathsf{X}(\mathsf{y}) \wedge tr_{\mathsf{X}}(\psi)(\mathsf{y}))) \wedge tr_{\mathsf{Y}}(\chi)(\mathsf{x})$
- $\varphi = (\langle \bullet ! \rangle \psi)$:[9]
  $\mathfrak{M}, \mathtt{w} \models (\langle \bullet ! \rangle \psi)$ iff there exists a formula χ such that $\mathfrak{M}, \mathtt{w} \models \langle \chi ! \rangle \psi$.
  iff there exists $D' \subseteq D_{\mathfrak{M}}$ such that $\mathtt{w} \in D'$ and $\mathfrak{M}', \mathtt{w} \models \psi$
  (where $\mathfrak{M}'$ is $\mathfrak{M}$ restricted to D'.) (see footnote 9)
  iff there exists D' such that $\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D'] \models \forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})$ and
  $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{Y} \leftarrow D'] \models \mathsf{Y}(\mathsf{x})$ and $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{Y} \leftarrow D'] \models tr_{\mathsf{Y}}(\psi)(\mathsf{x}))$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \exists \mathsf{Y}\ (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})) \wedge \mathsf{Y}(\mathsf{x}) \wedge tr_{\mathsf{Y}}(\varphi)(\mathsf{x})$
- $\varphi = (\langle \bullet !_G \rangle \psi)$: to prove this case, we first prove the following lemma.

Lemma 1. *Let $\mathfrak{M}$ be a Kripke model on AP, ψ be a formula on AP_M, a an agent. Then for all contexts $D' \subseteq D_{\mathfrak{M}}$, there exists χ such that $D' = D_{\mathfrak{M}^{K_{\pi_a}\chi}}$ iff*

$$\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D'] \models (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})) \wedge \forall \mathsf{x}\ (\forall \mathsf{y}\ \pi_a(\mathsf{x}, \mathsf{y}) \rightarrow (\exists \mathsf{z}\ \pi_a(\mathsf{z}, \mathsf{y}) \wedge \mathsf{Y}(\mathsf{z}))) \rightarrow \mathsf{Y}(x))$$

Proof. $\Rightarrow$ If there exists χ such that $D' = D_{\mathfrak{M}^{K_{\pi_a}\chi}}$ then $\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D'] \models (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y}))$. For the other formula, let $\mathtt{w}$ be a world such that for all $\mathtt{u}$ with $\mathtt{w} \xrightarrow{\pi_a} \mathtt{u}$, there exists a world $\mathtt{v}$ with $\mathtt{v} \xrightarrow{\pi_a} \mathtt{u}$. Then by definition, $\mathfrak{M}, \mathtt{u} \models \chi$ and so $\mathfrak{M}, \mathtt{w} \models K_{\pi_a}\chi$. We conclude that $\mathtt{w} \in D'$, so $\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D'] \models (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})) \wedge \forall \mathsf{x}\ (\forall \mathsf{y}\ \pi_a(\mathsf{x}, \mathsf{y}) \rightarrow (\exists \mathsf{z}\ \pi_a(\mathsf{z}, \mathsf{y}) \wedge \mathsf{Y}(\mathsf{z}))) \rightarrow \mathsf{Y}(x))$.

$\Leftarrow$ If $\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D'] \models (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})) \wedge \forall \mathsf{x}\ (\forall \mathsf{y}\ \pi_a(\mathsf{x}, \mathsf{y}) \rightarrow (\exists \mathsf{z}\ \pi_a(\mathsf{z}, \mathsf{y}) \wedge \mathsf{Y}(\mathsf{z}))) \rightarrow \mathsf{Y}(x))$ then $D' \subseteq D_{\mathfrak{M}}$. Let χ be the formula characterizing $post_{\pi_a}(D') = \{\mathtt{u} \in D_{\mathfrak{M}}$, there exists $\mathtt{v} \in D'$ such that $\mathtt{v} \xrightarrow{\pi_a} \mathtt{u}\}$ (the successors of D' via π_a). Then we obtain $D' \subseteq D_{\mathfrak{M}^{K_{\pi_a}\chi}}$. For the other implication, we observe that any element of $D' \subseteq D_{\mathfrak{M}^{K_{\pi_a}\chi}}$ has all its π_a-successors in $post_{\pi_a}(D')$, so is in D'.

Now back to the proof of the $\varphi = (\langle \bullet !_G \rangle \psi)$ case. Thanks to Lemma 1, we obtain:

[9] The right-to-left implication is proven by considering $\chi = \bigvee_{\mathtt{w} \in D'} \bigwedge_{p \in A, p \in \mathtt{w}} p \wedge \bigwedge_{q \in A, q \notin \mathtt{w}} \neg q$.

$\mathfrak{M}, \mathtt{w} \models (\langle \bullet ! \rangle \psi)$ iff there exists formulas $\{\chi_g, g \in G\}$ such that $\mathfrak{M}, \mathtt{w} \models \langle \bigwedge_{g \in G} K_{\pi_g} \chi_g ! \rangle \psi$.
iff there exists $\{D_g, g \in G\}$ such that for all $g \in G$
$\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D_g] \models (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})) \wedge \forall \mathsf{x}\ (\forall \mathsf{y}\ \pi_a(\mathsf{x}, \mathsf{y}) \rightarrow (\exists \mathsf{z}\ \pi_a(\mathsf{z}, \mathsf{y}) \wedge \mathsf{Y}(\mathsf{z}))) \rightarrow \mathsf{Y}(x))$
and $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{Y} \leftarrow \bigcap_{g \in G} D_g] \models \mathsf{Y}(\mathsf{x}) \wedge tr_{\mathsf{Y}}(\psi)(\mathsf{x})$.
iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}((\langle \bullet !_G \rangle \psi))(\mathsf{x})$.

4.4 Reduction from AGPAL-mc to MMSO-sat

We wrap up our results obtained so far to define the reduction from the symbolic model checking problem against AGPAL to the MMSO-satisfiability problem.

Definition 9 (reduction). *Given a pointed symbolic Kripke model* $(\mathfrak{M}, w)$ *and an* AGPAL*-formula* φ*, we let* $\tau(\mathfrak{M}, w, \varphi)$ *be the* MMSO *formula* $\mathscr{T}_A \wedge tr(\mathtt{w})(\mathsf{x}) \wedge \forall \mathsf{y} \mathsf{X}(\mathsf{y}) \wedge tr_{\mathsf{X}}(\varphi)(\mathsf{x})$ *that is computable in polynomial time in the size of* $\mathfrak{M}$*.*

By Corollary 1 and Theorem 3 we get the following.

Theorem 4. $\mathfrak{M}, w, \models \varphi$ *iff* $\tau(\mathfrak{M}, w, \varphi)$ *is* MMSO*-satisfiable.*

Because the symbolic model checking of AGPAL is $\mathrm{A_{pol}}$EXPTIME-hard [19], we obtain:

Corollary 2. MMSO*-satisfiability problem is* A_{pol}EXPTIME*-hard.*

However, as discussed in the next section, restricting to logic ∃AGPAL yields a reduction to the satisfiability problem of monadic first-order logic MFO.

5 Existential Announcement Logic into Monadic First-Order Logic

If we restrict inputs $\mathfrak{M}, w, \varphi$ of the AGPAL-model checking by letting $\varphi \in \exists$AGPAL, then $\tau(\mathfrak{M}, w, \varphi)$ is an MMSO-formula where all second-order quantifiers are existential and are not under the scope of universal quantifiers. Such second-order quantifiers can be removed from the formula $\tau(\mathfrak{M}, w, \varphi)$ resulting in a MFO-formula.

Since the symbolic model checking against ∃AGPAL is NEXPTIME-hard [19], the icing on the cake is the following already well-known lower-bound.

Corollary 3. MFO*-satisfiability problem is* NEXPTIME*-hard.*

In the next section, we make use of this reduction to solve the symbolic model checking problem against ∃AGPAL.

6 Implementation

We implemented the reduction from ∃AGPAL to MFO in OCaml. We also built benchmarks. The code and a readme file can be found at the following link https://github.com/tcharrie/agpal-mmso

6.1 Description of the Implementation

The input is an ∃AGPAL formula of the type `agpal_formula` in the source code. The type `acc_program` represents accessibility programs, the type `bool_formula` boolean formulas, and the type `fo_formula` MFO-formulas (the output of the code). The function `agpal_formula_to_mfo` defines the translation from ∃AGPAL formulas to MFO formulas (as in Definition 9).

In addition to the algorithm for the reduction, we implemented a function from existential formulas to the TPTP format [1] used by the FO-SAT-solvers, called `agpal_formula_to_tptp`. It first calls the function `agpal_formula_to_mfo`, then calls the function `mfo_formula_to_tptp` that transforms a MFO-formula into its TPTP representation.

6.2 Benchmarks

We provide benchmarks for FO-provers built from the muddy children and the Russian card puzzles in order to tests the combinatorial ability of FO-provers.

Muddy children. We consider the following true properties:

- $\varphi^{muddy}_{standard} = \langle \bigvee_{a\in \mathtt{Agt}} p_a! \rangle \langle \bigwedge_{a\in \mathtt{Agt}} \neg(K_a p_a \wedge \neg K_a \neg p_a)! \rangle \ldots \langle \bigwedge_{a\in \mathtt{Agt}} \neg(K_a p_a \wedge \neg K_a \neg p_a)! \rangle \bigvee_{a\in \mathtt{Agt}} (K_a p_a \vee K_a \neg p_a)$: standard formalization of the muddy children.
- $\varphi^{muddy}_{arbitrary} = \langle \bigvee_{a\in \mathtt{Agt}} p_a! \rangle \langle \bullet ! \rangle \bigwedge_{a\in \mathtt{Agt}} (K_a p_a \vee K_a \neg p_a)$: variant with an arbitrary announcement.
- $\varphi^{muddy}_{group} = \langle \bigvee_{a\in \mathtt{Agt}} p_a! \rangle \langle \bullet !_{\mathtt{Agt}} \rangle \bigwedge_{a\in \mathtt{Agt}} (K_a p_a \vee K_a \neg p_a)$: variant with a group announcement.

where $\mathtt{Agt} = \{1, \ldots, n\}$.

Russian cards. For this example, agents a and b holds the same number of cards n. For instance, the classical Russian cards problem corresponds to $n = 3$. Let $\varphi^{Russian}_{goal} = \bigwedge_{i=1}^{2n+1} (K_a p_{i,b} \vee K_a \neg p_{i,b}) \wedge (K_b p_{i,a} \vee K_b \neg p_{i,a}) \wedge \neg K_c p_{i,a} \wedge \neg K_c \neg p_{i,a} \wedge \neg K_c p_{i,b} \wedge \neg K_c \neg p_{i,b}$. We consider three types of properties:

- $\varphi^{Russian}_{arbitrary} = \langle \varphi_R ! \rangle \langle \bullet ! \rangle \varphi^{Russian}_{goal}$: formalization of the Russian cards with a unique arbitrary announcement.
- $\varphi^{Russian}_{group_1} = \langle \varphi_R ! \rangle \langle \bullet !_a \rangle \varphi^{Russian}_{goal}$: formalization with only one announcement from a. This formula is not satisfiable.
- $\varphi^{Russian}_{group_2} = \langle \varphi_R ! \rangle \langle \bullet !_a \rangle \langle \bullet !_b \rangle \varphi^{Russian}_{goal}$: normal formalization of the Russian cards problem.

6.3 Experiments

To perform the tests, we used the FO-solver Iprover [30] on a HP EliteBook 840 G2. The prover Iprover enabled us to test whether a FO-formula is satisfiable or not. The results are summarized in Fig. 3.

We now briefly comment on the experiments.

$n =$	$\varphi^{muddy}_{arbitrary}$	$n =$	$\varphi^{muddy}_{standard}$	φ^{muddy}_{group}	$n =$	$\varphi^{Russian}_{arbitrary}$	$\varphi^{Russian}_{group_1}$	$\varphi^{Russian}_{group_2}$
3	0.03s	3	0.07s	0.04s	2	0.18s	0.32s	0.45s
10	0.20s	4	0.09s	0.08s	3	0.44s	0.85s	0.92s
25	1.32s	5	0.19s	0.22s	4	3.80s	3.51s	3.32s
40	3.23s	6	0.24s	0.25s	5	23.48s	26.80s	24.20s
55	9.405s	7	$>$ 10min	$>$ 10min	6	$>$ 10min	$>$ 10min	$>$ 10min

Fig. 3. Results for the implementation of the reduction from ∃AGPAL to MFO, using the FO-SAT-solver Iprover.

Muddy children. For $\varphi^{muddy}_{arbitrary}$, the FO-SAT solver seems to perform well in all cases, as arbitrary announcements only require the new context to be included in the previous one. Hence, in this example, it is sufficient to restrict the model to the current world in order to satisfy the goal of $\varphi^{muddy}_{arbitrary}$. However, for the other tests, namely $\varphi^{muddy}_{standard}$ and φ^{muddy}_{group}, the FO-SAT-solver is able to test up to $n = 6$ agents. This can be explained by the fact public announcements and group announcements add significant combinatorial constraints to the specification.

Russian cards. For the three properties, the tests cannot exceed $n = 6$ cards, the main reason being that the rules of the game are very combinatorial, as for the muddy children.

Notice that the problems we have considered are puzzles, thus highly combinatorial. For the muddy children puzzle, the existential second-order quantification ranges over 2^{2^n} subsets. For $n = 7$, we have $2^{2^7} = 2^{128} \sim 10^{38}$, that is, about the number of positions $1.15868.. \times 10^{42}$ of a chess board.

Still, our implementation is promising and provides some interesting benchmarks for FO-provers.

7 Conclusion

We have reduced the problem of model checking symbolic Kripke models against AGPAL formulas to the satisfiability problem of MMSO, and shown that for the fragment ∃AGPAL, the reduction yields a satisfiability problem of some MFO formulas, which is known to be decidable [4,33]. We then have conducted experiments with FO provers. Our experiments show that the symbolic model checking problem against ∃AGPAL is difficult. As this problem is equivalent[10] to the MFO-satisfiability problem (they are both NEXPTIME-complete), we claim that efforts to obtain efficient algorithms are alike.

An interesting future work would be to effectively synthesize announcements. To this aim, we would like to generate the most simple formula to be announced so that a given property holds. This is close to the problem of generating a first-order model for a given MFO-formula.

[10] A reversed reduction can be proved.

We believe that our work is important since it would give efficient algorithms for several symbolic models in epistemic logic [7,18,26,27]. We also believe that the work done can improve epistemic planning specifications: in epistemic planning instances [13], the set of available actions is finite and described explicitly. Arbitrary announcement is a way to describe them implicitly. One can think of them as an action type while a specific announcement is an action token. Having efficient algorithms in this context would be very relevant.

Besides, we strongly believe that efficient data structures as in [37] for representing sets of sets of valuations are useful. Indeed, as Boolean formulas correspond to a set of valuations (and thus to binary decision diagrams [22]), an AGPAL-formula corresponds to a set of pair context/world, that, in a nutshell, could be represented by a set of sets of valuations.

On a more theoretical side, we would like to investigate on the relationship between announcement logics and MSO. Indeed, in MSO, second-order quantifications range over arbitrary sets (or over finite sets in weak-MSO) while announcements restrict the model to sets that are bisimulation-closed. We are not aware of any results regarding such second-order quantifiers.

Acknowledgments. We thank Konstantin Korovin who helped us to use iprover. We thank Ocan Sankur for pin-pointing us the article [12] where the authors reduce the model checking of safety properties into FO.

References

1. The tptp (thousands of problems for theorem provers) library. http://www.cs.miami.edu/~tptp/
2. Ågotnes, T., Balbiani, P., van Ditmarsch, H., Seban, P.: Group announcement logic. J. Appl. Logic **8**(1), 62–81 (2010)
3. Ågotnes, T., van Ditmarsch, H., French, T.: The undecidability of group announcements. In: International Conference on Autonomous Agents and Multi-agent Systems, AAMAS 2014, Paris, France, 5–9 May 2014, pp. 893–900 (2014)
4. Bachmair, L., Ganzinger, H., Waldmann, U.: Set constraints are the monadic class. In: Proceedings of the Eighth Annual Symposium on Logic in Computer Science (LICS 1993), Montreal, Canada, 19–23 June 1993, pp. 75–83 (1993)
5. Baier, C., Katoen, J.-P.: Principles of Model Checking. MIT Press, Cambridge (2008)
6. Balbiani, P., Baltag, A., van Ditmarsch, H.P., Herzig, A., Hoshi, T., De Lima, T.: What can we achieve by arbitrary announcements?: a dynamic take on fitch's knowability. In: TARK, pp. 42–51 (2007)
7. Balbiani, P., Gasquet, O., Schwarzentruber, F.: Agents that look at one another. Logic J. IGPL **21**(3), 438–467 (2013)
8. Balbiani, P., Herzig, A., Troquard, N.: Dynamic logic of propositional assignments: a well-behaved variant of PDL. In: LICS, pp. 143–152 (2013)
9. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In: Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge, pp. 43–56. Morgan Kaufmann Publishers Inc. (1998)

10. Barrett, C., Conway, C.L., Deters, M., Hadarean, L., Jovanović, D., King, T., Reynolds, A., Tinelli, C.: CVC4. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 171–177. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22110-1_14
11. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, New York (2001)
12. Bloem, R., Könighofer, R., Seidl, M.: SAT-based synthesis methods for safety specs. In: McMillan, K.L., Rival, X. (eds.) VMCAI 2014. LNCS, vol. 8318, pp. 1–20. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54013-4_1
13. Bolander, T., Andersen, M.B.: Epistemic planning for single and multi-agent systems. J. Appl. Non-Class. Logics **21**(1), 9–34 (2011)
14. Bozzelli, L., van Ditmarsch, H., Pinchinat, S.: The complexity of one-agent refinement modal logic. In: del Cerro, L.F., Herzig, A., Mengin, J. (eds.) JELIA 2012. LNCS (LNAI), vol. 7519, pp. 120–133. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33353-8_10
15. Bozzelli, L., van Ditmarsch, H.P., Pinchinat, S.: The complexity of one-agent refinement modal logic. In: IJCAI (2013)
16. Chandra, A.K., Stockmeyer, L.J.: Alternation. In: Proceedings of FOCS 1976, pp. 98–108 (1976)
17. Chareton, C., van Ditmarsch, H.: Strategic knowledge of the past in quantum cryptography. In: Baltag, A., Seligman, J., Yamada, T. (eds.) LORI 2017. LNCS, vol. 10455, pp. 347–361. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-55665-8_24
18. Charrier, T., Herzig, A., Lorini, E., Maffre, F., Schwarzentruber, F.: Building epistemic logic from observations and public announcements. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, 25–29 April 2016, pp. 268–277 (2016)
19. Charrier, T., Schwarzentruber, F.: Arbitrary public announcement logic with mental programs. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, 4–8 May 2015, pp. 1471–1479 (2015)
20. Charrier, T., Schwarzentruber, F.: A succinct language for dynamic epistemic logic. In: Proceedings of the 16th Conference on Autonomous Agents and Multi-agent Systems, AAMAS 2017, São Paulo, Brazil, 8–12 May 2017, pp. 123–131 (2017)
21. de Moura, L., Bjørner, N.: Z3: an efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 337–340. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78800-3_24
22. Drechsler, R., Becker, B.: Binary Decision Diagrams - Theory and Implementation. Springer, Berlin (1998). https://doi.org/10.1007/978-1-4757-2892-7
23. Fagin, R., Moses, Y., Halpern, J.Y., Vardi, M.Y.: Reasoning About Knowledge. MIT Press, Cambridge (2003)
24. Fischer, M.J., Ladner, R.E.: Propositional dynamic logic of regular programs. J. Comput. Syst. Sci. **18**(2), 194–211 (1979)
25. French, T., van Ditmarsch, H.P.: Undecidability for arbitrary public announcement logic. In: Advances in Modal Logic, pp. 23–42 (2008)
26. Gasquet, O., Goranko, V., Schwarzentruber, F.: Big brother logic: visual-epistemic reasoning in stationary multi-agent systems. Auton. Agent. Multi-Agent Syst. **30**(5), 793–825 (2016)

27. Herzig, A., Lorini, E., Maffre, F.: A poor man's epistemic logic based on propositional assignment and higher-order observation. In: van der Hoek, W., Holliday, W.H., Wang, W. (eds.) LORI 2015. LNCS, vol. 9394, pp. 156–168. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48561-3_13
28. Herzig, A., Maffre, F.: How to share knowledge by gossiping. In: Rovatsos, M., Vouros, G., Julian, V. (eds.) EUMAS/AT -2015. LNCS (LNAI), vol. 9571, pp. 249–263. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33509-4_20
29. Johnson, D.S.: A catalog of complexity classes. In: Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A), pp. 67–161. Elsevier (1990)
30. Korovin, K.: iProver – an instantiation-based theorem prover for first-order logic (system description). In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS (LNAI), vol. 5195, pp. 292–298. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-71070-7_24
31. Lemaignan, S., Ros, R., Mosenlechner, L., Alami, R., Beetz, M.: ORO, a knowledge management platform for cognitive architectures in robotics. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3548–3553. IEEE (2010)
32. Lemaignan, S., Warnier, M., Sisbot, A.E., Alami, R.: Human-robot interaction: tackling the AI challenges. Artif. Intell. (2014)
33. Lewis, H.R.: Complexity results for classes of quantificational formulas. J. Comput. Syst. Sci. **21**(3), 317–353 (1980)
34. Löwe, B., et al.: Logic and the simulation of interaction and reasoning: introductory remarks (2008)
35. Löwenheim, L.: Über möglichkeiten im relativkalkül. Math. Ann. **76**, 447–470 (1915)
36. Miller, J.S., Moss, L.S.: The undecidability of iterated modal relativization. Stud. Logica. **79**(3), 373–407 (2005)
37. Niveau, A., Zanuttini, B.: Efficient representations for the modal logic S5. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016, pp. 1223–1229 (2016)
38. Plaza, J.: Logics of public communications. Synthese **158**(2), 165–179 (2007)
39. Sutcliffe, G.: The CADE ATP system competition - CASC. AI Mag. **37**(2), 99–101 (2016)
40. van Benthem, J., van Eijck, J., Gattinger, M., Su, K.: Symbolic model checking for dynamic epistemic logic. In: van der Hoek, W., Holliday, W.H., Wang, W. (eds.) LORI 2015. LNCS, vol. 9394, pp. 366–378. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48561-3_30
41. van Ditmarsch, H., Grossi, D., Herzig, A., van der Hoek, W., Kuijer, L.B.: Parameters for epistemic gossip problems. In: Proceedings of LOFT 2016 (2016)
42. van Ditmarsch, H., Kooi, B.: One Hundred Prisoners and a Light Bulb. Springer, Switzerland (2015). https://doi.org/10.1007/978-3-319-16694-0
43. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer, Dordecht (2008). https://doi.org/10.1007/978-1-4020-5839-4
44. van Ditmarsch, H.P.: The Russian cards problem. Stud. Logica. **75**(1), 31–62 (2003)

Dynamic Logic: A Personal Perspective

Vaughan Pratt(✉)

Stanford University, Stanford, USA
pratt@cs.stanford.edu

Abstract. We review a few of the developments of dynamic logic from the author's perspective. As implied by the title the review is not intended as a survey of the field as a whole but rather as how the author's outlook on imperative programs and their logics evolved during the four decades up to the start of this millennium.

1 Pre-DL

If DaLí 2017 is any indication, dynamic logic has arguably stood the test of time. Since I was neither trained as a logician *per se* nor had envisaged a career in any area remotely like logic, the circumstances under which dynamic logic came to pass may therefore be of interest.

I was not a complete stranger to logic. Our high school swim team's coach Barry Blackwell thought I might enjoy the small logic booklet he'd studied in college, which focused on Aristotle's syllogistic. This was an entirely new concept for me, and I read it through several times. At Sydney University the following year I enrolled in science with the eventual goal of becoming a theoretical physicist. Having one elective I chose philosophy in order to learn more about logic. After that year however I went on to obtain a double honours degree in pure mathematics and physics, a five-year program, 1962–1966, without taking any further interest in logic.

At the time USyd's School of Physics had seven departments thanks to the not inconsiderable entrepreneurship of Canadian-born Harry Messel. In 1967 I started a physics masters degree in one of them, the Basser Computing Department. My attention soon turned to natural language. For my master's thesis it was suggested that I program a computer to solve Lewis Carroll's syllogisms, which eventually led to my thesis "Translation of Lewis Carroll's syllogisms into logic" [33]. I supported myself during that time with work on an implementation of Tarski's decision method for real closed fields for Charles Hamblin, chair of UNSW's philosophy department, then the next year for John Cannon at USyd implementing a precursor to Cayley [4,5] along with the Todd-Coxeter algorithm for enumerating cosets of a group.

For my Ph.D. I had been planning to extend the program to find the conclusion of each syllogism. However Margot and I had just married and as a US citizen she wanted to pursue a Ph.D. in the US. We applied to six universities,

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 153–170, 2018.
https://doi.org/10.1007/978-3-319-73579-5_10

of which three accepted us both. We chose Berkeley, which offered us both RA support, in my case from automata theorist Mike Harrison.

During our year at Berkeley I took courses from Dick Karp on resolution theorem proving, Steve Cook on complexity of algorithms, and Manuel Blum on recursion theory and Blum-style complexity theory, as well as various core computer science courses. I also audited a few lectures of Julia Robinson's graduate model theory course but found the prerequisites beyond me. Sadly there was no AI at all at Berkeley, nor could I find any linguist interested in looking at my masters thesis.

Within weeks of starting at Berkeley I began a correspondence with Stanford's Don Knuth, all of it dealing with his interest in analysis of algorithms. At mid-year Knuth suggested Margot and I transfer to Stanford, which we did.

During the first quarter of 1970/71 at Stanford I took a graduate model theory course from Paul Eklof which I found much better paced for me than Julia Robinson's course had been. I proposed to continue working in AI on an approach to natural language acquisition addressing Mark Gold's pessimism [12] but was persuaded by Knuth and John Hopcroft (on sabbatical from Cornell) that I could graduate considerably faster with an algorithms thesis, an argument I couldn't refute since by then I had more than enough material for that. My subsequent postdoc under Knuth during 1971/72 and my early years as an MIT faculty member largely continued my algorithms work with natural language processing as a side interest.

2 An MIT Course in Logics of Programs

MIT offered me a position as an assistant professor starting in 1972. At that time Oxford's Joe Stoy was visiting and teaching Dana Scott's lattice-based treatment of functional programming, which I learned from Joe's notes, lattice theory being a branch of algebra that I had no previous exposure to at all.

When Joe returned to Oxford I decided to start a course on semantics of programming languages, starting in 1973 with functional programming. But it soon struck me that most programs were written in the imperative style catered for by Floyd's [10] and Hoare's [17] axiomatic semantics based on respectively flowcharts and block-structured programs, so I added those two logics for that programming style to the material of the course.

I was next struck by how differently Floyd and Hoare viewed logic compared with Julia Robinson, Eklof, Scott, Stoy and other logicians I'd met. There was no semantics, only axioms! Both Floyd and Hoare took the position that their respective axiomatizations defined the meaning of the programs, though Floyd turned around near the end of his paper, "Assigning meanings to programs", and proved a completeness result for assignment that obviously depended on an unstated model-theoretic semantics that he must have had in the back of his mind.

Feeling uncomfortable about this lack of semantics, in 1974 I adopted the relational semantics then being advocated in CWI Amsterdam by Jaco de Bakker and Willem de Roever [6], a single-sorted system of relations based on regular expressions, and used it to give Hoare's $P\{a\}Q$ language a semantics for a two-sorted system of programs and sentences of first order logic (or three-sorted when counting terms). Relations were between states defined as interpretations of symbols. I forget where I first encountered the test concept but I found that it permitted extending Hoare's logic in a natural way to handle nondeterministic programs by breaking up conditionals and loops. Thinking of first order logic as a static logic, I regarded $P\{a\}Q$ furnished with these semantics as a dynamic logic.

Some of the students in the class, in particular Bob Moore, Martin Brooks, and Jerry Ginsparg, were familiar with modal logic and pointed out that my notion of state seemed to correspond with Kripke's notion of possible world from his paper "Semantical Considerations on Modal Logic" [24]. So over the weekend I took home a copy of Hughes and Cresswell to learn modal logic, worked out how to formulate Hoare's partial correctness assertions and beyond in modal form, and wrote it up as class notes for the next lecture.

3 The Resulting Paper

After teaching both imperative and functional semantics for two years, in 1976 I decided it was time to run my semantics for logics of imperative programs past the STOC-FOCS community I was then part of, which I began by writing it up as an MIT Technical Report before submitting it to FOCS'76 [34].

Section 1.1 gives the usual Tarskian or homomorphic notion of an interpretation of an expression, including the (referentially opaque) interpretation $I(\langle a\rangle P)$ defined as the disjunction over all $J(P)$ for which the relation IaJ holds, with dual modality $[a]P$ definable as $\neg\langle a\rangle\neg P$. A state is then defined as an interpretation, a transition as a pair of states, and a program as a set of transitions, making a program the same thing as the interpretation a of a modality $\langle a\rangle$.

Section 1.2 concerns Hoare's notion of partial correctness assertion or pca $P\{a\}Q$. Satisfaction of an antecedent-consequent pair (P, Q) by a transition (I, J) is defined as $I(P)$ implies $J(Q)$, and truth of $P\{a\}Q$ as satisfaction of (P, Q) by all transitions in a. I pointed out a duality principle, namely equivalence of $P\{a\}Q$ and $\neg Q\{a^-\}\neg P$ where a^- is the converse of a.

I also pointed out that $[a]Q$ is the weakest antecedent P for the consequent Q such that $P\{a\}Q$ is true, and made a connection with a corresponding notion in Dijktra's book that had just come out [7]. However I neglected to state the correct connection, which was not with "weakest precondition" (which entails termination) but rather with "weakest *liberal* precondition". I also was completely unaware at the time of the considerably earlier and extensive work by Andrzej Salwicki and his wife Grazyna Mirkowska on Algorithmic Logic [53], which had the same content as $[a]P$, albeit only for deterministic programs, and therefore was a major oversight.[1]

[1] By way of partial excuse, the Berlin Wall still existed and duplication of effort between East and West was not uncommon, for example Marshall Stone's discovery [56] that Boolean algebras were rings without being aware that Zhegalkin had pointed it out nine years earlier [58].

Section 2 introduces and proves properties of particular first-order programs including array assignments, quantifiers (as random assignments), tests, and their combinations with the three regular operations. Section 3.1 generalizes binary relations to several other semantics of programs. This included what I called $*$-ary relations: whereas a binary relation is a set of pairs of states a $*$-ary relation is a set of sequences of states. Section 3.2 expanded Hoare's language $P\{a\}Q$ to the modal logic formulas $[a]P$ and $\langle a\rangle P$ that had been introduced in the third last paragraph of Sect. 1.1. $P\{a\}Q$ is subsumed with $P \rightarrow [a]Q$, while termination of a is expressible with $\langle a\rangle T$.

I presented the paper at FOCS'76 and the response was positive, which I took to mean that those attendees steeped in logic recognized that the previous axiomatic treatments of imperative program verification had no semantic basis at all, let alone the clean one provided by de Bakker's relations between states, also found in contemporary work by Hoare and Lauer [18].

In hindsight it might seem that I had defined multimodal logic in Sect. 1.1 and hence first order dynamic logic in Sect. 1.2. However introduction of a new language was not the main goal of the paper, whence the deferral to Sect. 3.2 of what was later identified with my (then) broader concept of a dynamic logic. Rather I wanted to furnish Hoare's logic with a semantics based on interpretations-as-states, much as Kripke furnished modal logic with a semantics based on possible worlds. I therefore chose the title "Semantical considerations on Floyd-Hoare logic" to convey this parallel with Kripke's "Semantical considerations on modal logic" [24]. Other authors such as Burstall, Schwarz, and Kroeger had proposed modal logic for program verification but without semantics of any kind let alone Kripke's possible worlds semantics, and without having gained any evident interest or traction, so the need for a modal language seemed less urgent than the need for a semantics for Hoare's logic. I mentioned the modalities early on in Sect. 1.1 only because they seemed to help in clarifying the concepts on which my semantics for $P\{a\}Q$ rested.

When I submitted the paper for journal publication the sole referee's report seemed to bear out my concerns. It was by far the most negative report I'd received in my career to date. It complained that there didn't seem to be any point to the paper since Hoare's partial correctness assertions and their associated axioms were exactly what was needed for program verification. Moreover the hope I'd expressed in the paper that it could be used as class notes was a lost cause because the paper was too unreadable to serve as such. The referee did not even mention the possible worlds semantics, presumably because Hoare's axioms and Floyd's verification conditions sufficed for the purposes of both program verification and "assigning meanings to programs" (the title of Floyd's paper).

Having no prior experience of negative referee's reports I took the criticisms to heart and abandoned that paper in favor of coming up with a much better one when I understood the issues better. In hindsight that was very naive of me, but I was just coming up for tenure then and the report opened my mind to the possibility that my case could be judged equally harshly. That concern was heightened when I was told, I forget by whom, that switching subjects, in this case from algorithms to logics of programs, just before tenure could be taken as a bad sign.

As it turned out the modal language in Sect. 3.2 was accepted enthusiastically by the more theoretically inclined computer scientists, rendering moot what the more practical consumers of program verification technology thought of it. This acceptance started earlier on than one might have guessed. When I was starting to develop the modal version in 1974 I was carpooling to work with Mike Fischer, then a tenured associate professor at MIT, on an almost daily basis, giving us nearly an hour each carpool day to discuss our common interests, including the ideas that ended up two years later in my FOCS'76 paper.

Not long after, Mike accepted the University of Washington's offer of a full professorship, taking with him our common understanding of many theoretical ideas in both logic and algorithms. Richard Ladner at UW had proved some complexity results about classical modal logic [25], so it was very natural for the two of them to start a collaboration on a propositional version of dynamic logic, asking the same types of complexity questions about it that Richard had previously addressed. From time to time we heard from Mike and Richard about their progress in getting the definitions sorted out and the upper and lower bounds on the computational complexity of PDL tightened up.

4 Early Collaborations

Shortly after my FOCS'76 paper several people at and near MIT joined me in working on first order dynamic logic, most notably my incoming research assistant David Harel who had just finished his masters under Amir Pnueli on program verification and had started his Ph.D. at MIT as my research assistant funded by a new NSF grant I'd been awarded. He read my FOCS paper in the course of his first week at MIT, and participated in my class teaching the material, I forget whether in the role of the class's TA or my RA. Not long after, he began writing "First Order Dynamic Logic", which eventually turned into his thesis, with which he graduated in 1978, two years after entering MIT.

Albert Meyer also joined us, resulting in a three-author paper "Computability and Completeness in logics of programs" presented at STOC'77 [14]. Rohit Parikh from nearby Boston University soon became a regular visitor to our group, while Mike Paterson from Warwick, who had been a frequent visitor to our theoretical computer science group well before DL existed, also became interested in DL; both Rohit and Mike found the propositional version of particular interest.

5 Program Verification

Floyd-Hoare logic being applicable to program verification, and dynamic logic being (hopefully) an elegant formulation of Floyd-Hoare logic, I wanted to see how well DL fared in practice. To this end I assigned my student Steve

Litvintchouk the task of implementing a dynamic logic proof checker. We applied it to the verification of the Knuth-Morris-Pratt pattern matcher [20] and presented it at IJCAI'77 [26].

On the one hand this was surely the first application of dynamic logic to any kind of software engineering problem. On the other it attracted negligible attention: according to Google Scholar it accounted for only 0.1% of citations of papers coauthored by me. Furthermore neither Steve nor I had it anywhere near the top of our respective priorities and I paid no further attention to the construction of program verifiers until my planned 1980 sabbatical at Stanford.

6 Propositional Dynamic Logic: Complexity and Axiomatization

At STOC'77 Fischer and Ladner presented "Propositional Modal logic of programs" [9] which introduced propositional dynamic logic, PDL, to the community. They obtained upper and lower bounds on the complexity of their logic of respectively nondeterministic and deterministic exponential time. Unlike space, for which the gap between determinism and nondeterminism is quadratic [54], the corresponding gap for time is not known to be less than exponential, raising the question of whether this gap in the case of PDL could be narrowed. Liking computational complexity questions like that, I began to think about it seriously.

A little later, in the summer of 1977, Krister Segerberg visited UW, proposed an axiomatization of PDL, and claimed it was complete. I invited him to visit us at MIT to speak on his proof and to stay at my house. When he arrived he told me he no longer believed his completeness proof was sound. This was an unexpected development since his axioms had the intuitive feel of a complete axiomatization of a modal logic. It was immediately clear that any incompleteness would have to involve PDL's star operator, since without it completeness of his axiomatization could easily be proved by standard methods of modal logic of the kind first developed by Kripke. Krister's talk at MIT on PDL and the prospects for its complete axiomatization were nonetheless well received.

This started a search in 1977 for a completeness proof of Krister's axiomatization, which was joined by Dov Gabbay at King's College, London, Rohit Parikh at Boston University, myself, and possibly others, all working independently. Dov wrote me a brief note (which I still have) sketching a proof of about a dozen lines seemingly on the basis of nothing more than that standard modal logic techniques applied here, which seemed inconsistent with Segerberg's experience in discovering a bug in what he originally took to be a sound proof of completeness of his system. At the same time Rohit developed a proof based on a nonstandard semantics of PDL, which he presented as part of an invited talk on dynamic logic at the Mathematical Foundations of CS (MFCS) conference in September 1978 at Zakopane, Poland [30]. Simultaneously I attempted to convince myself of the completeness of Krister's system but my arguments by induction on the height of the terms became so long that I was unable to satisfy myself that I hadn't overlooked some important detail, though I published my best effort in that regard as [41] as a sort of "hail-mary" that it might be bug-free (I certainly couldn't find any bugs). However I was doing no better trying to convince myself of the soundness of Rohit's proof either and therefore regarded the problem as still open. Neither Fischer nor Ladner had accepted any of these proofs, and Richard would ask me from time to time about progress on the problem.

7 Decision Methods for PDL

In 1978 I took the decidability of PDL as a good reason to focus on the propositional fragment of the Litvintchouk-Pratt proof checker [26]. At that time the fastest deterministic decision method for PDL appeared to require time two exponentials in the input length. I was therefore willing to settle for what I considered a heuristically efficient method [35], namely the method of semantic or analytic tableaux, for which I referenced Gentzen [11], Hintikka [16], Beth [2], and Smullyan [55]. I found the method well suited to dynamic logic; what I did not realize at the time (pointed out to me recently by Rajeev Goré) was that nearly two decades earlier Kripke had found essentially the same method suitable for showing the completeness of first order S5 modal logic with equality [23]. As far as I know however my decision method was the first application of tableaux to logics of programs.

At MFCS'79 in Olomouc, Czechoslovakia, Valiev [57] kindly pointed out two errors in my Gentzen type axiomatization of PDL [35, p. 335]. The first antecedent of the rule $\neg[*$ for induction should have been $\Gamma \vdash p, \Delta$; my omission of Δ was obviously in this case just a typo, whether mine or the typist's I'll never know. However my first rule, the axiom $\neg P$ asserting $P \vdash P$, should have been $\neg p$ asserting $p \vdash p$ because in PDL not all compound p's can be decomposed to atoms P the way they can in propositional logic: stars get in the way. At the conference where Valiev was pointing this out I was sitting next to Juris Hartmanis, who turned to me in wonder that anyone would consider my using the wrong case a significant error. I had to explain *sotto voce* that Valiev had identified a genuine and serious error in my axiomatization. Were there more sharp-eyed people like Valiev checking code for bugs, 143,000 customers of Equifax might be breathing easier today!

In the same paper I also followed up on Sect. 3.1 of [34] by applying the method to $*$-ary relations as sets of *sequences* of states together with additional modalities *throughout*, *during*, and *preserves*. Here *throughout*(a, p) means that p holds at all states, *during*(a, p) means that p holds at some state, and *preserves*(a, p) means that if p becomes true at some state then it remains true at all subsequent states; in each case these are required to hold for *all* trajectories of a. They all have evident duals in which p respectively holds at some state, holds at every state, and becomes true (has a 0–1 transition) at some state, with all three required to hold for *some* trajectory of a. I extended the tableau and decision method to cater for *throughout*, leaving open how to handle *during* and *preserves*, mainly to avoid an absurdly large set of rules.

In 1978 I was discussing something unrelated to PDL with Albert Meyer when a closure property of PDL occurred to me that immediately convinced me it that PDL must be in EXPTIME, thereby closing the aforementioned gap. I interrupted our conversation to tell this to Albert, who was understandably dubious. While solutions to problems do occasionally pop suddently into my head, this was the first and perhaps only time in my life when it happened in the course of an unrelated conversation! I incorporated the method into the journal version of the heuristic STOC'78 paper [35] and replaced "Practical" by "Near-optimal" (in the sense of "to within a polynomial") in the title [41].

8 Going Algebraic

I felt that a simpler completeness proof might be possible if there were some way of making the syntax of PDL more abstract, in particular by somehow identifying those terms that were obviously equivalent semantically so as to significantly reduce the number of paths in the proof. At STOC'79 in Atlanta, Georgia I asked Jonas Makowsky for advice on ways to make proofs more abstract and he referred me to a recent (1977) tutorial paper by Leon Henkin in the American Math Monthly, "The logic of equality" [15], showing how to use a theorem of Birkhoff to prove completeness of equationally axiomatized theories by purely algebraic methods. As this looked exactly like what I wanted, as soon as I returned to MIT I borrowed Rasiowa and Sikorsky's book "The Mathematics of Metamathematics" [51] and read the first few chapters over the weekend in order to ground myself in the basis for Henkin's approach.

Somewhere about that time, either at POPL'79 or STOC'79, perhaps the latter, Dexter Kozen told me something about *-free PDL that had something to do with Stone duality. Not knowing what Stone duality was and being focused on star as the main problem with PDL, I was unable to follow it. But then in May 1979 Dexter sent me a short manuscript titled "A representation theorem for *-free PDL" [21] based on an equational axiomatization of PDL without star, which clarified what he'd been telling me.

Except for not catering for star, which simplified PDL to something we already knew how to axiomatize without consulting Segerberg's axioms, Dexter's equational axiomatization fitted perfectly with Henkin's approach, and it was immediately clear how one could use that approach to prove a completeness result for an axiomatization of *-free PDL.

In this equational algebraic framework it was natural to use juxtaposition to denote $a;b$ as ab and $\langle a\rangle p$ as ap, and to form Boolean combinations of them in the usual manner of Boolean algebra. As Dexter had pointed out, PDL equations like $(a \cup b)p = ap \vee bp$ and $(ab)p = a(bp)$ then resembled the axioms for vector spaces and more generally noncommutative R-modules, with programs playing the role of elements from a ring R and propositions as vectors (points) in the R-module.[2]

[2] An R-module is a vector space just when the ring R is a field.

The question now became, was there a corresponding finite axiomatization of PDL *with* star that would allow Henkin's method to be applied to Segerberg's axiomatization of PDL, thereby giving the desired completeness proof?

I axiomatized PDL with star by translating Segerberg's induction axiom into algebraic notation as follows, noting that $p \leq q$ can be expressed equationally as $p \vee q = q$.

$$a^*p \leq p \vee a^*(p' \wedge ap)$$

The meaning of this is that if iterating a can eventually make p true, then either p is already true or it is possible by iterating a to reach a state where p is not yet true but one more iteration of a can make p true.

Along with the other Segerberg axioms translated into equations I was then able to show that in any model of these equations a^*p was the least Boolean value for q satisfying $p \vee aq \leq q$. This made a^* essentially reflexive transitive closure, not in the sense of an infinite union but in the sense that a is defined axiomatically as reflexive when $p \leq ap$, and transitive when $aap \leq ap$, for all p. With the further nonequational condition of separability for a dynamic algebra, namely that if $ap = bp$ for all p then $a = b$, one can then say that in any ***separable*** dynamic algebra a is reflexive when $1 \leq a$ and transitive when $aa \leq a$. In the following I'll assume separability is always an ambient condition on dynamic algebras.

Whether this finitary notion of dynamic algebra in order to meet the requirements of an axiomatization of PDL with star was better or worse than Kozen's infinitary one seemed to me like an irrelevant value judgement. The point was that both of us contributed to the development of the notion that eventually led to my Henkin-like proof of completeness of Segerberg's axiomatization of PDL.

It would have been nice to show that all dynamic algebras in my sense were isomorphic to Kripke structures. However Dexter had previously shown the existence of dynamic algebras in his sense in which star as infinite union yielded a dynamic algebra that was "nonstandard", i.e. not isomorphic to any Kripke structure. While I was not sure whether this would also be the case for my notion, in any event it was not necessary to decide this as it was only necessary to show representability of *free* dynamic algebras.

My main theorem was that every free dynamic algebra (still with this ambient assumption of separability) was ***residually finite***, meaning that it was isomorphic to a subdirect product of finite dynamic algebras. This by itself established completeness of Segerberg's axioms. Feeling that the problem was solved at last, I wrote all this up in a 33-page MIT technical memo, TM#138, "Dynamic Algebras: Examples, Constructions, Applications" [37] which appeared in July, 1979. Its abstract reads as follows. (Note the distinction between "basic" and "main" results, both of which I considered important).

Dynamic algebras combine the classes of Boolean ($B\ \vee\ '\ 0$) *and regular* ($R\ \cup\ ;\ \star$) *algebras into a single finitely axiomatized variety* ($\mathcal{B}\ \mathcal{R}\ \diamond$) *resembling an R-module with "scalar" multiplication* $\diamond$. *The basic result is that* $\star$ *is reflexive transitive closure, contrary to the intuition that this concept should* *require quantifiers for its definition. Using this result we give several examples of dynamic algebras arising naturally in connection with additive functions, binary relations, state trajectories, languages, and flowcharts. The main result is that free dynamic algebras are residually finite (i.e. factor as a subdirect product of finite dynamic algebras), important because finite separable dynamic algebras are isomorphic to Kripke structures. Applications include a new completeness proof for the Segerberg axiomatization of propositional dynamic logic, and yet another notion of regular algebra.*

My 33-page technical report was clearly too long for a STOC submission (STOC and FOCS being my preferred publication vehicles rather than journals) so I submitted a much shorter version to STOC'80 as "Dynamic algebras and the nature of induction" [40], focusing on the "basic" and "main" results.

Following the above reasoning as to who contributed what I simply wrote "The class of dynamic algebras consists of all models of the Segerberg axioms for PDL. It was first studied as a class by D. Kozen and the author." Had I been more sensitive back then to proper attribution it might have been more precise to say that the equationally defined class of all models of *-free PDL was introduced by D. Kozen and subsequently conservatively extended by me to the class of all models of PDL with star. But even that isn't quite right because Dexter defined a different conservative extension in which star was defined externally as an infinite union making PDL with star a noncompact theory that could not be used to prove the desired completeness result. The possibility of consulting Dexter as to the best possible attribution did not occur to me.

9 A Conflict

That year I was invited to speak at two immediately back-to-back conferences in Europe. The first was the 6th International Conference on Logic, Philosophy and Methodology of Science held in Hanover, Germany in August 22–29, 1979. The second, three days later, was the 8th MFCS conference in Olomouc, Czechoslovakia (now Slovakia), September 3–7, 1979, entailing a long train trip from Prague to Olomouc behind the Iron Curtain, the one I mentioned earlier in connection with M.K. Valiev.

For the first I provided a paper titled "Dynamic Logic" [38]. For the latter I provided a paper titled "Axioms or Algorithms" [36] proposing an approach to replacing traditional axioms and inference rules of proof systems by axioms for decidable fragments of logic intended to allow many fewer steps in a computer-checkable proof. Although this latter paper made no significant reference to dynamic logic beyond noting that PDL was decidable in exponential time, I had in mind using the idea as the basis for a proofchecker that I was planning to build in collaboration with Derek Oppen on my sabbatical at Stanford in 1980/81.

At Hanover I met many algebraic logicians including Istan Nemeti, Hajnal Andreka, Larissa Maximova, Mike Dunn from Indiana U., and Richard Routley from ANU, Australia. I explained Dexter's model of *-free PDL to Mike Dunn, who reprimanded me sharply for attributing to Dexter what had been developed by Jonsson and Tarski in 1951 [19]. This unnerved me as the attribution to Dexter appeared in my invited paper for the conference. However this was before the internet and I had no way of researching this until I returned to MIT. So I decided, unwisely as it turned out, to not bring up attribution unless asked, on the assumption that if Mike was right then I would simply be using 30-year-old mathematics familiar to modal logicians. This turned out to be unwise because I had underestimated the importance of being the first to apply one area of mathematics to another.

This issue did not come up at Hanover but it did at the MFCS meeting in Olomouc the next week. I gave my invited paper, "Axioms or Algorithms" [36], as scheduled but then was asked to give an evening talk about my completeness proof for the Segerberg axioms. On short notice I hastily prepared slides sketching the proof, which of course involved Henkin's algebraic techniques based on the first few chapters of Rasiowa and Sikorski, and presented the material, which was well received. What I didn't allow for however was many people there were very familiar with the techniques in that book but had never seen it applied to a completeness proof for a logic of programs.

It turned out that Dexter had been invited to talk to a group that intersected with the MFCS attendees, in the week immediately following MFCS. They very excitedly informed him of the techniques I'd used in proving the result, so naturally he asked whether I'd mentioned him. When he learned that I hadn't, it wasn't long before I heard from him about that. I gathered that he felt I was trying to take sole credit for the development of dynamic algebra. Even though all my writings on the subject have always been careful to credit him, this incident led to a long period of unhappiness between us.

When the STOC'80 deadline came up I tightened my long memo of July 79 to just the "basic result" (* is reflexive transitive closure in the algebraic sense) and the algebraic completeness proof and submitted it to STOC'80 under the title "Dynamic algebras and the nature of induction".

I assumed that people would satisfy themselves as to the soundness of my completeness proof, which by now I understood thoroughly. Istvan Nemeti did so. But he also spotted an improvement I'd overlooked. Every finite dynamic algebra is representable as a (finite) Kripke structure, and every Kripke structure of any size is representable as a subdirect product of finite Kripke structures, by a straightforward argument. It is therefore an easy consequence of my exotic-sounding result, that free dynamic algebras are residually finite, that every free dynamic algebra is isomorphic to a Kripke structure. He also pointed out that freee dynamic algebras were automatically separable so that nonequational condition could be dropped, making the result even cleaner, leaving freeness itself as the only nonequational condition. I was the referee for his paper making that point [29], and since the result was obvious and yet an improvement on my clumsier formulation it took only a minute to check it and judge it acceptable for publication. This was easily my easiest refereeing job ever! I was however very sad that I'd overlooked it, which I put down to my inexperience in the brave new world of algebraic logic.

Much later I realized that no one else was acknowledging that I'd found a clearly correct completeness proof. The problem was that almost no one interested in the problem understood or cared about algebraic techniques: I might as well have written the paper in Latin!

A decade later Istvan Nemeti contacted me about my much longer July 1979 memo that included the results in my STOC'80 paper to say that he was sorry I had never published it, since he felt the examples would be of interest to many. So I added a few historical remarks and reflections at the end, updated one or two of the citations to their latest versions, and it appeared in Studia Logica in 1991 [49].

10 Onwards to Applications!

Having satisfied myself after STOC'80 as to the completeness of the Segerberg axioms, I turned my attention to a quite different project: basing a program verification system on my deterministic decision method for PDL combined with the several decision methods for certain equational theories of datatypes commonly encountered in programs, developed by Nelson and Oppen during the preceding several years [28], using the decision method approach described in my invited paper "Axioms or Algorithms" that I'd presented at MFCS. This was a natural extension of the aforementioned work in 1977 with my student Steve Litvintchouk on a dynamic logic proof checker [26].

To this end I proposed to Derek Oppen, newly appointed to the Stanford CS faculty, that I spend a sabbatical year at Stanford collaborating with him on the implementation of such a system. I applied for and was granted a sabbatical from MIT for 1980/81, while Derek found a house I could rent in Los Altos Hills for the year, a 25-min bike ride from the CS department. Renting our Weston, Massachusetts house to a family for the year, my family of four settled into the Los Altos house.

Once settled I showed up at Stanford's CS department to get a desk and get to work with Derek. I got the desk, but was informed that Derek had left Stanford the previous week to do an MBA in order to become a businessman.

Well, so much for my sabbatical. I went home and spent two days learning how to solve Instant Insanity and then Rubik's cube quickly while mulling over what should be done about this revolting development. I could work on my own, but that would have been a waste of a sabbatical since I'd been doing that very successfully for eight years at MIT and felt it was time to get involved in more collaborative work.

So I bicycled back to the department and looked around to see what interesting projects other people were up to. One that I found particularly appealing was Forest Baskett's project to develop what he was describing at the time as a smart terminal but which he cheerfully agreed had enough power to be easily turned into a self-contained personal computer, what later came to be called an engineering workstation to distinguish it from the smaller personal computers that were then springing up like Daisies. I found myself quickly sucked into Forest's project.

Within a month however Forest informed me that he was leaving Stanford, just like Derek, I thought. I wondered if I was jinxing the place—shades of my thesis adviser Mike Harrison at Berkeley in 1969, who had spent that year on sabbatical in Israel and whom I first met in person in 1973! This left me as the only faculty member with any interest in the Sun project, though Jim Clark was very keen to build his graphics engine chip, and the two us shared a wide desk in Margaret Jacks 433 for a few weeks as we typed away on our terminals (Jim's official office was over in ERL).

During that period Stanford offered me a full professorship, which was an improvement on my associate professorship at MIT. MIT refused to be bullied into matching their offer, which I therefore accepted. Meanwhile I stepped into Forest's shoes as director of the project, which sucked me into it even deeper, with Forest's student Andy Bechtolsheim now mine.

The rest is more or less history that is irrelevant to dynamic logic. The work with Sun, first as a Stanford project known as the Stanford University Network (SUN) terminal and 18 months later as a Stanford spinoff named Sun Microsystems, which I worked with initially as a consultant and then for two years, first as an employee on leave from Stanford working on computer graphics, digital typography, etc. and then as Sun's first director of research. This adventure lasted sufficiently long to cure me of my itch to implement a proof checker based on decision methods.

11 Multimodal Logic, Flowgraphs, Minimization, and Modal Mu-Calculi

Late in 1980, as a sideline to my focus at the time on the Sun workstation, I found myself wondering whether the flowcharts on which Floyd's program logic [10] was based were fully subsumed by the regular expressions of PDL. Floyd interpreted each edge e of a flowchart as a proposition $I(e)$ that he called the *tag* of e. As usual a flowchart is made more like a finite state automaton when the edges are represented as vertices and the command boxes as edges, call this the flowgraph form of the flowchart. Each tag $I(e)$ on a flowchart edge e then becomes a tag q_x on the corresponding flowgraph vertex x.

In practice there exist flowgraphs whose least equivalent regular expression is exponentially larger, suitably measured [8]. Hence if one tested satisfiability of a proposition about a flowgraph by first translating the flowgraph into a regular expression, the exponential blowup possible with the translation, when composed with my deterministic exponential time decision method for PDL, would appear to require doubly exponential time. This raised the question of whether there was a faster decision method for the propositional version of Floyd's logic as understood in the framework of PDL.

At a conference at IBM in May 1981 organized by Dexter Kozen I defined Propositional Flowgraph Logic, PFL, as multimodal[3] logic together with all finitary flowgraph operations [43]. The three regular operations of PDL arise as special cases defined by the obvious small flowgraphs, for example star a^* is expressed as a one-vertex flowgraph with a self-loop a.

I showed that PFL was in EXPTIME using a decision method for a proposition r based on my method for PDL. The approach conceptually was to start with the free Boolean algebra $\mathcal{B}$ generated by the propositional variables along with an additional variable Q_{Ap} for each action A labeling an edge of a flowgraph and each subformula p of r. Each subformula p is interpreted as an element $h(p)$ of $\mathcal{B}$, while each action A is interpreted as the maximal strict finitely additive function $h(A)$ on $\mathcal{B}$ satisfying $h(A)(h(p)) \leq h(Q_{Ap})$, which provably exists. $\mathcal{B}$ is then shrunk keeping the $h(A)$'s maximal until every inequality $h(A)(h(p)) \leq h(Q_{Ap})$ becomes an equality. Then r is satisfied just when $h(r)$ is nonzero. The shrinking of $\mathcal{B}$ is accomplished by a tag minimization procedure analogous to transitive closure applied to matrices over $\mathcal{B}$.

The maximization and minimization operations in this algorithm soon led me to the idea that the flowgraphs in PFL could be replaced by their corresponding inequalities along with a minimization operation, constituting the language of what in retrospect was the first modal mu-calculus. I wrote this up for FOCS'81 and presented it in October 1981, six months after the IBM conference [42]. Unfortunately most of my time by then had been absorbed by the Sun workstation project and I did a very poor job of defining and explaining my version of the modal mu-calculus. Two years later Dexter did a much better job of defining the modal mu-calculus [22], which is the version that has stuck.

12 CS353: Algebraic Logic

Although the Sun project had cured me of my itch to build a verifier based on dynamic logic and the Nelson-Oppen methodology, what I was not cured of was a new appreciation for algebra acquired after its success (in my mind) with the proof of completeness of Segerberg's axiomation of PDL. I started teaching an annual course on algebraic logic in the CS department, initially limited to lattice theory and universal algebra. But after a couple of years teaching it I began to wonder how it related to category theory, which I had originally learned at Sydney University from Max Kelly, Australia's top category theorist, but had since forgotten all about. Eventually I made the connection to my satisfaction and incorporated it into the course so as to segue smoothly between universal algebra and category theory. I taught the course, CS353, Algebraic Logic, each year until about 2002, averaging about ten students each year. The course's popularity remained on a steady keel throughout until I decided, two years after my retirement in 2000, to call it quits.

[3] I had not used the term "multimodal" explicitly in any of my earlier papers. Rennie's n-multiply modal calculus based on constants $M_1, M_2, \ldots, M_n$ [52] is a much earlier related concept.

13 Concurrency

Early on in the development of dynamic logic I had wondered what sort of a program connective could cater for concurrency. My initially foray there was via various notions of process logic starting in 1979 [39]. However the so-called Brock-Ackerman anomaly for dataflow machines [3] soon came to my attention and started to bother me. This led in due course to my abandonment of the states on which I founded the semantics of dynamic logic and the adoption instead of an event based semantics [44–46], which seemed intuitively to lend itself better to a form of concurrency not vulnerable to the Brock-Ackerman anomaly. This pursuit led to my formulation of pomsets, a name that has stuck even though Grabowski had invented the concept a year earlier under the less catchy rubric of "partial languages" [13]. The more serious competitors at that time were Petri nets, developed in the 1960s [31], Mazurkiewicz traces [27], and Pnueli's temporal logic [32], both developed in the 1970s.

At a conference organized by Jack Dennis in New Brunswick in 1977 Pnueli characterized temporal logic as an endogenous logic to distinguish it from exogenous logics like Hoare logic and dynamic logic. Whereas dynamic logic treated programs compositionally temporal logic regarded the universe as under the control of one huge program stepping forward in time. A single proposition could refer to multiple parts of a state separated in space though not time. In this logic the natural concurrency connective was simply conjunction: the conjunction of two propositions about different parts of the state were simply true simultaneously. An element of time was introduced by modal operators such as Next and Eventually.

Because dynamic logic could express everything temporal logic could, and moreover initially the latter had no semantics, just axioms like Hoare logic, some of us in the dynamic logic camp didn't take temporal logic seriously. I felt however that two reasons why it caught on in due course were that it was a simpler logic than dynamic logic making it somewhat easier to treat theoretically, and that it catered for concurrency with no hassle at all, whereas all proposals for concurrency of programs built compositionally seemed to suffer from one defect or another.

But my involvement with Sun dragged me away from that issue for several years. Nevertheless I always had it in the back of my mind as something to look into at some point.

Eventually I found myself asking why events seemed better than states. It didn't take long to see why: states existed in automata having only one-dimensional transitions. In order for two events to behave independently their corresponding transitions in the state view would need to do so as well. This called for a two-dimension transition, and soon I had a paper "Modelling concurrency with geometry" [47] that I presented at POPL'90. This gave rise to a whole cottage industry of such semantics for concurrency, with a series of conferences organized by Eric Goubault, one of its earliest fans.

In a "parallel" development my student Vineet Gupta and I studied the Chu spaces of Mike Barr and his masters student Po-Hsiang (Peter) Chu [1].

Chu spaces created a natural framework in which to embed the duality between state-based and event-based semantics of concurrency, which I had previously identified as the Duality of Time and Information [48]. This led to a considerable body of work that is more than will fit here. My current favorite paper in that quite long series is "Transition and Cancellation in Concurrency and Branching Time" [50] which ties together a number of threads including the connection between geometric models and Chu spaces.

While I would love to continue in this vein, for the purposes of DaLí this is going to have to suffice.

References

1. Barr, M.: ∗-Autonomous Categories. Lecture Notes in Mathematics, vol. 752. Springer, Berlin (1979)
2. Beth, E.W.: The Foundations of Mathematics. North Holland, Amsterdam (1959)
3. Brock, J.D., Ackerman, W.B.: An anomaly in the specifications of nondeterministic packet systems. Technical report Computation Structures Group Note CSG-33, MIT Lab. for Computer Science, November 1977
4. Cannon, J.J.: A general purpose group theory program. In: Proceedings of the Second International Conference Theory of Groups, Canberra, pp. 204–217 (1973)
5. Cannon, J.J.: A draft description of the group theory language cayley. In: Proceedings of the Third ACM Symposium on Symbolic and Algebraic Computation, SYMSAC 1976, pp. 66–84. ACM, New York (1976)
6. de Bakker, J.W., de Roever, W.P.: A calculus for recursive program schemes. In: Nivat, M. (ed.) Automata, Languages and Programming, pp. 167–196. North Holland (1972)
7. Dijkstra, E.W.: A Discipline of Programming. Prentice-Hall, Englewood Cliffs (1976)
8. Ehrenfeucht, A., Zeiger, P.: Complexity measures for regular expressions. J. Comput. Syst. Sci. **12**(2), 134–146 (1976)
9. Fischer, M.J., Ladner, R.E.: Propositional modal logic of programs. In: Proceedings of the 9th ACM Symposium on Theory of Computing, pp. 194–211, Boulder, May 1977. Journal version: Propositional dynamic logic of regular programs, JCSS 18:2 (1979)
10. Floyd, R.W.: Assigning meanings to programs. In: Schwartz, J.T. (ed.) Mathematical Aspects of Computer Science, pp. 19–32 (1967)
11. Gentzen, G.: Investigations into logical deductions. In: Szabo, M.E. (ed.) The Collected Papers of Gerhard Gentzen, pp. 68–131. North-Holland, Amsterdam (1934)
12. Gold, E.M.: Language identification in the limit. Inf. Control **10**(5), 447–474 (1967)
13. Grabowski, J.: On partial languages. Fundam. Inform. **IV**(2), 427–498 (1981)
14. Harel, D., Meyer, A.R., Pratt, V.R.: Computability and completeness in logics of programs. In: Proceedings of the 9th Annual ACM Symposium on Theory of Computation, pp. 261–268 (1977)
15. Henkin, L.: The logic of equality. Amer. Math. Mon. **84**(8), 597–612 (1977)
16. Hintikka, K.J.J.: Form and content ni quantification theory. Acta Philos. Fenni. **8**, 7–55 (1955)
17. Hoare, C.A.R.: An axiomatic basis for computer programming. Commun. ACM **12**, 576–580 (1969)

18. Hoare, C.A.R., Lauer, P.E.: Consistent and complementary formal theories of the semantics of programming languages. Acta Inform. **3**, 135–153 (1974)
19. Jónsson, B., Tarski, A.: Boolean algebras with operators. Part I. Amer. J. Math. **73**, 891–939 (1951)
20. Knuth, D.E., Morris, J., Pratt, V.R.: Fast pattern matching in strings. SIAM J. Comput. **6**(2), 323–350 (1977)
21. Kozen, D.: A representation theorem for models of $*$-free PDL. Technical report RC7864, IBM, September 1979
22. Kozen, D.: Results on the propositional mu-calculus. Theor. Comput. Sci. **23** (1983)
23. Kripke, S.: A completeness theorem in modal logic. J. Symb. Logic **24**(1), 1–14 (1959)
24. Kripke, S.: Semantical considerations on modal logic. Acta Philos. Fenn. **16**, 83–94 (1963)
25. Ladner, R.E.: The computational complexity of provability in systems of modal propositional logic. SIAM J. Comput. **6**(3), 467–480 (1977)
26. Litvintchouk, S.D., Pratt, V.R.: A proof checker for dynamic logic. In: 5th International Joint Conference on A.I., pp. 552–558, August 1977
27. Mazurkiewicz, A.: Concurrent program schemes and their interpretations. Technical report DAIMI Report PB-78, Aarhus University, Aarhus (1977)
28. Nelson, G., Oppen, D.C.: Fast decision algorithms based on union and find. In: 18th IEEE Symposium on Foundations of Computer Science, October 1977
29. Németi, I.: Every free algebra in the variety generated by the representable dynamic algebras is separable and representable. Theoret. Comput. Sci. **17**, 343–347 (1982)
30. Parikh, R.: A completeness result for a propositional dynamic logic. In: Winkowski, J. (ed.) MFCS 1978. LNCS, vol. 64, pp. 403–415. Springer, Heidelberg (1978)
31. Petri, C.A.: Fundamentals of a theory of asynchronous information flow. In: Proceedings of the IFIP Congress 62, Munich, pp. 386–390 (1962). North-Holland, Amsterdam
32. Pnueli, A.: The temporal logic of programs. In: 18th IEEE Symposium on Foundations of Computer Science, pp. 46–57, October 1977
33. Pratt, V.R.: Translation of lewis carroll's syllogisms into logic. Masters Thesis, August 1969
34. Pratt, V.R.: Semantical considerations on Floyd-Hoare logic. In: Proceedings of the 17th Annual IEEE Symposium on Foundations of Computer Science, pp. 109–121, October 1976
35. Pratt, V.R.: A practical decision method for propositional dynamic logic. In: Proceedings of the 10th Annual ACM Symposium on Theory of Computing, San Diego, pp. 326–337, May 1978
36. Pratt, V.R.: Axioms or algorithms. In: Proceedings of the 6th Symposium on Mathematical Foundations of Computer Science, Olomouc, Czech. (1979)
37. Pratt, V.R.: Dynamic algebras: Examples, constructions, applications. Technical report MIT/LCS/TM-138, M.I.T. Laboratory for Computer Science, July 1979
38. Pratt, V.R.: Dynamic logic. In: Proceedings of the 6th Conference on Logic, Methodology, and Philosophy of Science, Hanover, West Germany, pp. 251–261 (1979)
39. Pratt, V.R.: Process logic. In: Proceedings of the 6th Annual ACM Symposium on Principles of Programming Languages, San Antonio, pp. 93–100, January 1979
40. Pratt, V.R.: Dynamic algebras and the nature of induction. In: 12th ACM Symposium on Theory of Computation, Los Angeles, April 1980

41. Pratt, V.R.: A near optimal method for reasoning about action. J. Comput. Syst. Sci. **2**, 231–254 (1980). Also MIT/LCS/TM-113, M.I.T., Sept. 1978
42. Pratt, V.R.: A decidable mu-calculus. In: Proceedings of the 22nd IEEE Conference on Foundations of Computer Science, pp. 421–427, October 1981
43. Pratt, V.R.: Using graphs to understand PDL. In: Kozen, D. (ed.) Logic of Programs 1981. LNCS, vol. 131, pp. 387–396. Springer, Heidelberg (1982). https://doi.org/10.1007/BFb0025792
44. Pratt, V.R.: Position statement. Circulated at the Panel on Mathematics of Parallel Processes, chair A.R.G. Milner, IFIP-83, September 1983
45. Pratt, V.: The pomset model of parallel processes: unifying the temporal and the spatial. In: Brookes, S.D., Roscoe, A.W., Winskel, G. (eds.) CONCURRENCY 1984. LNCS, vol. 197, pp. 180–196. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-15670-4_9
46. Pratt, V.: Two-way channel with disconnect. In: Denvir, B.T., Harwood, W.T., Jackson, M.I., Wray, M.J. (eds.) The Analysis of Concurrent Systems. LNCS, vol. 207, pp. 110–114. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-16047-7_39
47. Pratt, V.R.: Modeling concurrency with geometry. In: Proceedings of the 18th Annual ACM Symposium on Principles of Programming Languages, pp. 311–322, January 1991
48. Pratt, V.R.: The duality of time and information. In: Cleaveland, W.R. (ed.) CONCUR 1992. LNCS, vol. 630, pp. 237–253. Springer, Heidelberg (1992). https://doi.org/10.1007/BFb0084795
49. Pratt, V.R.: Dynamic algebras: examples, constructions, applications. Stud. Logica **50**(3/4), 571–605 (1992)
50. Pratt, V.R.: Transition and cancellation in concurrency and branching time. Math. Struct. Comp. Sci. **13**(4), 485–529 (2003). Special issue on the difference between sequentiality and concurrency
51. Rasiowa, H., Sikorski, R.: The Mathematics of Metamathematics. Polska Akademia Nauk. Monografie matematyczne, vol. 41. Drukarnia Uniwersytetu, Warsaw (1963)
52. Rennie, M.K.: Models for multiply modal systems. Zeitschr. j. math. Logik und Grundlagen d. Math. **16**, 175–186 (1970)
53. Salwicki, A.: Formalized algorithmic languages. Bull. Acad. Pol. Sci., Ser. Sci. Math. Astr. Phys. **18**(5), 227–232 (1970)
54. Savitch, W.J.: Relationships between nondeterministic and deterministic tape complexities. J. Comput. Syst. Sci. **4**, 177–192 (1970)
55. Smullyan, R.: First Order Logic. Springer, Berlin (1968)
56. Stone, M.: The theory of representations for Boolean algebras. Trans. Amer. Math. Soc. **40**, 37–111 (1936)
57. Valiev, M.K.: On axiomatization of deterministic propositional dynamic logic. In: Proceedings of the 6th Symposium on Mathematical Foundations of Computer Science, Olomouc, Czech. (1979)
58. Zhegalkin, I.I.: On the technique of calculating propositions in symbolic logic. Matematicheskii Sbornik **43**, 9–28 (1927)

The Creation and Change of Social Networks: A Logical Study Based on Group Size

Sonja Smets and Fernando R. Velázquez-Quesada(✉)

Institute for Logic, Language and Computation, Universiteit van Amsterdam, Amsterdam, The Netherlands
{S.J.L.Smets,F.R.VelazquezQuesada}@uva.nl

Abstract. This paper is part of an on-going programme in which we provide a logical study of social network formations. In the proposed setting, agent a will consider agent b as part of her network if the number of features (properties) on which they differ is small enough, given the constraints on the size of agent a's 'social space'. We import this idea about a limit on one's social space from the cognitive science literature. In this context we study the creation of new networks and use the tools of Dynamic Epistemic Logic to model the updates of the networks. By providing a set of reduction axioms we are able to provide sound and complete axiomatizations for the logics studied in this paper.

1 Introduction

While the study of social interactions has received a lot of attention in logic and AI, the existence of a specific social group or network on which these studies are based is typically taken for granted. So what is left mostly unexplored is the way a social group is formed or the way in which a social network is created. This is exactly the topic we address in this paper. As such, this proposal complements our previous work in [1] which provides a *threshold* based approach to social network formation. In the threshold setting, an agent a considers agent b as part of her social network if and only if the number of features in which they differ is smaller or equal than a given threshold θ. This paper follows a different approach by using an idea that arises from the cognitive science literature: focus not on a similarity threshold, but rather on the size of the agent's 'social space'. In real life, agents may be willing to keep expanding their social network with people who are decreasingly less similar from them, as long as there is still 'enough space' in their social environment.[1] This is famously known as the *Dunbar's number*: a suggested cognitive limit to the number of people with whom one can maintain stable social relationships (see, e.g., [2]).

In the next section we first introduce the social network models as a context in which we can specify a distance between agents in a network. This distance

[1] Think, for example, how we establish conversations with relatively 'distant' acquaintances mostly only when our close friends are not around.

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 171–184, 2018.
https://doi.org/10.1007/978-3-319-73579-5_11

is then used to create a layered structure of an agent's possible social contacts, which is an essential ingredient in the mechanism that allows agents to form a new social network or to extend a given one when they are asked to take into account the bound on their 'social space'. We study a logical system that can express such network creations, giving a sound and complete axiomatisation for it. Finally we focus on the representation of more refined scenarios in which not all features play the same important role in the network creation/formation process. We conclude with a series of ideas for possible generalizations and/or alternative settings that can be explored in future work.

2 Modelling Social Networks

Similar to [1], our starting point is the basic setting of [3] in which we work with a relational 'Kripke' model in which the domain is interpreted as the set of agents, the accessibility relation represents a social connection from one agent to another, and the atomic valuation describes the features (behavior/opinions) that each agent has. Let A denote a countable set of agents, and P (with $\mathsf{A} \cap \mathsf{P} = \varnothing$) a countable set of features that agents might or might not have:

Definition 2.1 (Social Network Model). *A* social network model *(SNM) is a tuple $M = \langle \mathsf{A}, S, V \rangle$ where $S \subseteq \mathsf{A} \times \mathsf{A}$ is the* social *relation (Sab indicates that agent a is socially connected to agent b) and $V : \mathsf{A} \to \wp(\mathsf{P})$ is a* feature *function ($p \in V(a)$ indicates that agent a has feature p).*

Note how the social relation S does not need to satisfy any specific property (in particular, it is not required to be irreflexive, and neither symmetric), and thus it differs from the *friendship* relation of other approaches (e.g., [3–6]). Given a social network model, we define a notion of 'distance' between agents based on the number of features in which they differ.

Definition 2.2 (Distance). *Let $M = \langle \mathsf{A}, S, V \rangle$ be a SNM. Let $\textsc{msmtch}_M(a, b)$ be the set of features distinguishing agents $a, b \in \mathsf{A}$ in M:*

$$\textsc{msmtch}_M(a, b) := \mathsf{P} \setminus \{p \in \mathsf{P} : p \in V(a) \textit{ iff } p \in V(b)\}$$

Then, the distance *between a and b in M is given by*

$$\textsc{dist}_M(a, b) := |\textsc{msmtch}_M(a, b)|$$

As discussed in [1], $\textsc{dist}$ is a mathematical distance: for any agents $a, b \in \mathsf{A}$ and any SNM, ***(i)*** the distance from a to b is non-negative (non-negativity: $\textsc{dist}_M(a, b) \geq 0$), ***(ii)*** the distance from a to b is equal to that from b to a (symmetry: $\textsc{dist}_M(a, b) = \textsc{dist}_M(b, a)$), and ***(iii)*** the distance from an agent to herself is 0 (reflexivity: $\textsc{dist}_M(a, a) = 0$). Moreover, $\textsc{dist}$ is a *semi-metric*, as it also satisfies subadditivity: 'going directly' from a to c is 'faster' than 'going' via another agent ($\textsc{dist}_M(a, c) \leq \textsc{dist}_M(a, b) + \textsc{dist}_M(b, c)$). Still, $\textsc{dist}$ is not a *metric*, as it does not satisfy *identity of indiscernibles*: $\text{DIST}_M(a, b) = 0$ does not imply $a = b$, as two different agents may have exactly the same features.[2]

Static Language $\mathcal{L}$. Following [3], social network models are described by a *propositional* language $\mathcal{L}$, with special atoms describing the agents' features and their social relationship:

Definition 2.3 (Language $\mathcal{L}$). *Formulas φ, ψ of the language $\mathcal{L}$ are given by*

$$\varphi, \psi ::= p_a \mid \mathrm{S}_{ab} \mid \neg\varphi \mid \varphi \wedge \psi$$

with $p \in \mathsf{P}$ and $a, b \in \mathsf{A}$. We read p_a as "agent a has feature p" *and S_{ab} as* "agent a is socially connected to b"*. Boolean constants ($\top, \bot$) and other Boolean operators ($\vee, \rightarrow, \leftrightarrow, \veebar$, the latter representing the exclusive disjunction) are defined as usual. Given a SNM $M = \langle \mathsf{A}, S, V\rangle$, the semantic interpretation of $\mathcal{L}$-formulas in M is given by:*

$$M \Vdash p_a \quad \textit{iff}_{def} \quad p \in V(a), \qquad\qquad M \Vdash \neg\varphi \quad \textit{iff}_{def} \quad M \not\Vdash \varphi,$$
$$M \Vdash \mathrm{S}_{ab} \quad \textit{iff}_{def} \quad Sab, \qquad\qquad M \Vdash \varphi \wedge \psi \quad \textit{iff}_{def} \quad M \Vdash \varphi \textit{ and } M \Vdash \psi.$$

A formula $\varphi \in \mathcal{L}$ is valid (notation: $\Vdash \varphi$) when $M \Vdash \varphi$ holds for all models M.

Since there are no restrictions on the social relation nor on the feature function, any axiom system of classical propositional logic is fit to characterize syntactically the validities of $\mathcal{L}$ over the class of social network models.

3 Group-Size-Based Social Network Creation

As mentioned before, [1] approaches social network creation by considering a similarity threshold θ, then defining each agent's new social space as all those agents that differ from her in at most $\theta \in \mathbb{N}$ features. This proposal follows a different strategy. Borrowing an idea from cognitive science [2], it considers a maximum group-size $\lambda \in \mathbb{N}$, then defining each agent's new social space as the λ agents that are closer to her, according to the above defined distance.

This section implements this idea of agents having a size-bounded social space; the following tools are used to make this idea precise.

Definition 3.1. *Given a social network model $M = \langle \mathsf{A}, S, V\rangle$ and an agent $a \in \mathsf{A}$, the quantitative notion of distance* DIST *induces a qualitative (total, reflexive, transitive and well-founded) relation $\preccurlyeq_a^M \subseteq \mathsf{A} \times \mathsf{A}$ of* distance from a. *Such relation is given by*

$$\preccurlyeq_a^M := \{(b_1, b_2) \in \mathsf{A} \times \mathsf{A} : \text{DIST}_M(a, b_1) \leq \text{DIST}_M(a, b_2)\},$$

and thus $b_1 \preccurlyeq_a^M b_2$ indicates that, in model M, agent b_1 is at least as close to agent a as agent b_2. By defining the notion of $\preccurlyeq_a^M$-minimum in the standard way (for $\mathsf{B} \subseteq \mathsf{A}$, take $\text{MIN}_a(\mathsf{B}) := \{b \in \mathsf{B} : b \preccurlyeq_a^M b'$ for all $b' \in \mathsf{B}\}$),

[2] See [7, Chap. 1] for more details on mathematical distances.

this relation induces a sequence of layers (i.e., an ordered list of subsets) on A *(*$\mathrm{A}_a(-1), \mathrm{A}_a(0), \ldots, \mathrm{A}_a(n), \ldots$*, for* $n \geq 0$*), with each set containing agents equally distant from* a*:*

$$\mathrm{A}_a(-1) := \varnothing, \quad \mathrm{A}_a(0) := \textsc{min}_a(\mathsf{A}), \quad \mathrm{A}_a(n+1) := \textsc{min}_a(\mathsf{A} \setminus \bigcup_{k=-1}^{n} \mathrm{A}_a(k)).$$

Different agents might be 'equally distant' from a, and thus $\preccurlyeq_a^M$ is not anti-symmetric: layers might have more than one element.[3] Moreover: while an initial empty layer $\mathrm{A}_a(-1)$ has been defined (its usefulness will be clear below), the layer $\mathrm{A}_a(0)$ always contains those agents that are feature-wise identical to a (including a herself). Note also how the layers are collectively exhaustive and pairwise disjoint: every agent appears in exactly one of them. Finally, when A is finite, at some point a 'first' empty layer $\mathrm{A}_a(k)$ will appear (for some $k > 0$), and from that moment on all layers will be empty too.

The layered structure of an agent's social contacts will be a helpful tool to model how agents can form a new social network or even extend a given one by performing updates on their social relations. Such agents are asked to establish new connections to agents that are close enough to them given the bound on their 'social space'. To model this we introduce the idea of a bounded similarity update operation on models, using the tools of Dynamic Epistemic Logic on how one can model such transformations on models [8–10].

Definition 3.2 (Bounded similarity update). *Let* $M = \langle \mathsf{A}, R, V \rangle$ *be a* SNM*; take* $\lambda \in \mathbb{N}$*. Denote by* $\ell_a(\lambda)$ *the 'last' layer of contacts an agent* $a \in \mathsf{A}$ *can add to her network without going above the maximum group size* λ*, i.e.,*

$$\ell_a(\lambda) := \max\{n \in \mathbb{N} \cup \{-1\} : \ |\bigcup_{k=-1}^{n} \mathrm{A}_a(k)| \leq \lambda\}$$

The bounded similarity update *on* M *produces the* SNM $M_{\bowtie_\lambda} = \langle \mathsf{A}, S_{\bowtie_\lambda}, V \rangle$*, with its social relation given by*

$$S_{\bowtie_\lambda} := \{(a,b) \in \mathsf{A} \times \mathsf{A} : \ b \in \bigcup_{k=-1}^{\ell_a(\lambda)} \mathrm{A}_a(k)\}$$

Since layers might have more than one element, each agent could reach a point where she should decide whether to add the next layer of friends and go above the limit λ, or stop and stay strictly below it. The definition provided above chooses the second possibility: agents will always stay below the limit, even if that means leaving some 'memory slots' empty. The extra empty layer $\mathrm{A}_a(-1)$ makes this definition work in cases in which the first layer $\mathrm{A}_a(0)$ contains already

[3] In such case, and if no additional criteria is used to distinguish agents in the same layer, all of them should 'stand together': the decision of whether they will become part of a's social network should be of a 'either all or else none' nature.

too many agents. In such situations, $\ell(\lambda) = -1$ and hence $\bigcup_{k=-1}^{\ell(\lambda)} \mathrm{A}_a(k) = \mathrm{A}_a(-1) = \varnothing$; thus, after the bounded similarity update operation, the agent will be friendless.

Properties and Variations. The social network created by the threshold approach of [1] is reflexive (hence serial) and symmetric, though it might not be neither transitive nor Euclidean. In contrast, a social network created by the group-size bounded similarity update does not guarantee any of such properties. First, for reflexivity,

Proposition 3.1. *Let $M = \langle \mathsf{A}, S, V \rangle$ be a* SNM *and $a \in \mathsf{A}$ be an agent; take $M_{\bowtie_\lambda} = \langle \mathsf{A}, S_{\bowtie_\lambda}, V \rangle$ (Definition 3.2). Then, a considers herself as part of her new social network ($S_{\bowtie_\lambda} aa$) if and only if the amount of people that are feature-wise identical to her is at most the limit λ ($|\mathrm{A}_a(0)| \leq \lambda$).*

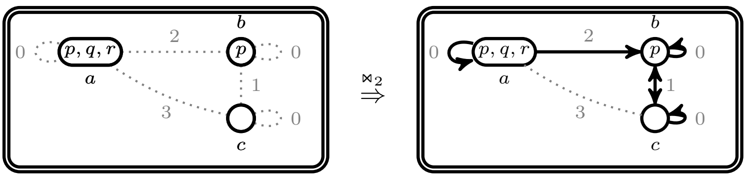
Note how $|\mathrm{A}_a(0)| > \lambda$ implies not only that $S_{\bowtie_\lambda} aa$ will fail, but also that $S_{\bowtie_\lambda}[a] = \varnothing$ (so a will be friendless after the operation).

For symmetry, transitivity and Euclideanity,

Fact 3.1. Let $M = \langle \mathsf{A}, S, V \rangle$ be a SNM; take $M_{\bowtie_\lambda} = \langle \mathsf{A}, S_{\bowtie_\lambda}, V \rangle$. Then, $S_{\bowtie_\lambda}$ might not be neither symmetric, nor transitive nor Euclidean.

Proof. Here are counterexamples to each one of these properties.

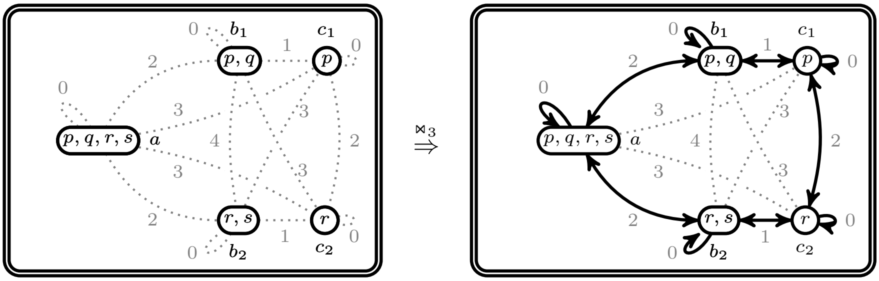
- Symmetry fails for a and b if, despite a having 'enough space' for b, there is some c that is both closer to b than a ($\mathrm{DIST}_M(b, c) \leq \mathrm{DIST}_M(b, a)$, so b would pick c over a), and farther away from a than b ($\mathrm{DIST}_M(a, b) \leq \mathrm{DIST}_M(a, c)$, so a would chose b over c). By taking $\lambda = 2$, the SNM below on the left shows such situation, with the SNM on the right being the result of the update.[4]



  More generally, symmetry fails if a high occurrence of similar agents produces a fully connected cluster, leaving dissimilar ones with asymmetric edges.
- The failure of transitivity also relies on b being close enough to a (so $S_{\bowtie_\lambda} ab$ holds) and c being both close enough to b (so $S_{\bowtie_\lambda} bc$ holds) and further away from a 'in b's direction' (so $S_{\bowtie_\lambda} ac$ fails). The models above showing the failure of symmetry also show how transitivity might fail.

[4] Numbers over edges indicate distance. Edges in black are actual pairs in the social network relation, and dotted grey edges are shown only for distance information.

- Finally, the relation is not Euclidean if, even though b_1 and b_2 are both close enough to a for the latter to call them her friends, they are different enough from each other to allow somebody else to take their supposed place by being more similar to each one of them (while also being very different from a). Such slightly convoluted situations are described better graphically, and the SNM below on the left is an example (take $\lambda = 3$).



Characterising those situations in which the group-size approach produces symmetric, transitive or Euclidean social networks is not straightforward. Obviously, a λ larger or equal than $|\mathsf{A}|$ will produce fully connected (hence symmetric, transitive and Euclidean) relations; still, these properties might be achieved under other circumstances. For example, symmetry can be achieved also when the agents are 'similarly dissimilar' (i.e., their differences are 'uniformly distributed'), as the update might yield ring-like structures with symmetric edges (see the above 'Euclideanity' counterexample).

These results might suggest that the networks created by Definition 3.2 are relatively 'arbitrary': compared with the threshold approach of [1], which guarantees reflexivity and symmetry, the group-size approach might seem to produce random social networks. This is actually not the case. In the threshold approach, what matters for deciding whether b will become part of a's social network (besides the threshold itself) is only the distance between a and b. However, in the group-size approach, what matters for deciding whether b will become part of a's social network (besides the group-size itself) is the distance between a and *all agents*. Indeed, the distance between a and b is, by itself, not enough: it is possible for a and b to be extremely similar (say, $\text{DIST}_M(a, b) = 1$), and still b will not be in a's social network if the number of agents feature-wise identical to a is high. Even more: a and b might be feature-wise identical, and yet they will not be socially connected if the number of agents feature-wise identical to them is larger than the group-size.

The group-size approach is context-sensitive: the new social networks are built not in terms of how fit is each candidate individually, but rather on how fit is each candidate *compared with the rest*. In other words, it is not about similarity, but rather about *relative* similarity. A detailed study of the group-conditions that guarantee the social network will have specific properties is left for future work.

Still, the operation might be defined in slightly alternative ways. As mentioned before, in some cases the agent will have empty 'memory slots' because the next layer would have put her social network above the size limit. One could make it possible for an agent to take on exactly λ contacts by asking for additional criteria to distinguish agents in the same layer (e.g., in an appropriate setting, considering not only the agents' features but also their preferences/beliefs). Still, one can also assume that λ is a loose limit, allowing the agent to go above it when she cannot tell the members of a group apart. For this, a small change in the definition of the upper limit ℓ_a (Definition 3.2) is enough:

$$\ell_a(\lambda) := \max\{n+1 \in \mathbb{N} : |\bigcup_{k=-1}^{n} \mathrm{A}_a(k)| \leq \lambda\}.$$

Readers interested in irreflexive friendship relations (as those in [3–6]) can achieve this property by defining agent a's sequence of layers not in terms of the full set of agents A, but rather in terms of $\mathsf{A}^{-a} := \mathsf{A} \setminus \{a\}$.

Finally, a further variation is to allow for each agent to have a *personal group-size* [11,12]. This can be represented by a function $\Lambda : \mathsf{A} \to \mathbb{N}$ indicating how many friends each agent can handle, which can be then used to define the new updated relation as $S_{\bowtie_\lambda} := \{(a,b) \in \mathsf{A} \times \mathsf{A} : b \in \bigcup_{k=-1}^{\ell_a(\Lambda(a))} \mathrm{A}_a(k)\}$, the only difference being the use of $\Lambda(a)$ instead of λ when defining the agents who will join a's social group.

Dynamic Language $\mathcal{L}_{\bowtie\lambda}$. To express how the bounded update changes a social network, we define the language $\mathcal{L}_{\bowtie_\lambda}$.

Definition 3.3 (Language $\mathcal{L}_{\bowtie\lambda}$). *The language $\mathcal{L}_{\bowtie_\lambda}$ extends $\mathcal{L}$ with a modality $[\bowtie_\lambda]$ to build formulas of the form $[\bowtie_\lambda]\,\varphi$ (*"after a bounded similarity update, φ is the case"*). The semantic interpretation of this modality refers to the bounded similarity updated model of Definition 3.2 as follows. Let M be a* SNM*; then,*

$$M \Vdash [\bowtie_\lambda]\,\varphi \quad \textit{iff}_{def} \quad M_{\bowtie_\lambda} \Vdash \varphi.$$

Note that no precondition is required for a bounded similarity update. Because of this and the functionality of the model operation, the dual modality $\langle \bowtie_\lambda \rangle\,\varphi := \neg\,[\bowtie_\lambda]\,\neg\varphi$ is such that $\Vdash [\bowtie_\lambda]\,\varphi \leftrightarrow \langle \bowtie_\lambda \rangle\,\varphi$.

The axiom system characterising validities of $\mathcal{L}_{\bowtie_\lambda}$ in SNM is built via the *DEL* technique of *recursion axioms*. As such, it makes crucial use of the fact that the basic 'static' language $\mathcal{L}$ is already expressive enough to characterise the changes that the bounded similarity update operation brings about. The crucial axiom, the one characterising the way in which the social network relation changes, will be built up step by step.

First, note that, when P is *finite*, the following $\mathcal{L}$-formula is true in a model M if and only if agents a and b differ in exactly $t \in \mathbb{N}$ features:[5]

$$\mathrm{Dist}^{t}_{a \cdot b} := \bigvee_{\{\mathsf{P}' \subseteq \mathsf{P} : |\mathsf{P}'| = t\}} \Big(\bigwedge_{p \in \mathsf{P}'} (p_a \veebar p_b) \wedge \bigwedge_{p \in \mathsf{P} \setminus \mathsf{P}'} (p_a \leftrightarrow p_b) \Big)$$

The second step consists in defining the $\mathcal{L}$-formula $\mathrm{Closer}_{a \cdot b_1 \cdot b_2}$, which is true in a model M if and only if agent b_2 is at most as close to agent a as agent b_1 (i.e., $\textsc{dist}_M(a, b_1) \leq \textsc{dist}_M(a, b_2)$):[6]

$$\mathrm{Closer}_{a \cdot b_1 \cdot b_2} := \bigvee_{j_1 = 0}^{|\mathsf{P}|} \bigvee_{j_2 = j_1}^{|\mathsf{P}|} \Big(\mathrm{Dist}^{j_1}_{a \cdot b_1} \wedge \mathrm{Dist}^{j_2}_{a \cdot b_2} \Big)$$

By using the $\mathrm{Closer}_{a \cdot b_1 \cdot b_2}$ formula, and in those cases in which A is finite, it is possible to provide further $\mathcal{L}$-formulas characterising the agents in each one of the layers induced by the qualitative '*distance from a*' relation $\preccurlyeq^M_a$: for $n \geq 0$,

$$\mathrm{InLay}_{a,-1}(b) := \bot, \qquad \mathrm{InLay}_{a,0}(b) := \bigwedge_{p \in \mathsf{P}} (p_a \leftrightarrow p_b),$$

$$\mathrm{InLay}_{a,n+1}(b) := \bigwedge_{k=0}^{n} \neg\, \mathrm{InLay}_{a,k}(b) \;\wedge\; \bigwedge_{c \in \mathsf{A}} \left(\bigwedge_{k=0}^{n} \neg\, \mathrm{InLay}_{a,k}(c) \;\rightarrow\; \mathrm{Closer}_{a \cdot b \cdot c} \right).$$

It is not hard to see that each formula $\mathrm{InLay}_{a,k}(b)$ indeed characterises each layer $\mathrm{A}_a(k)$, i.e., for every SNM M, $a \in \mathsf{A}$ and $k \in \mathbb{N} \cup \{-1\}$,

$$\mathrm{A}_a(k) = \{ b \in \mathsf{A} : \; M \Vdash \mathrm{InLay}_{a,k}(b) \}$$

The cases for $k = -1$ and $k = 0$ are straightforward: $\mathrm{A}_a(-1)$ is always empty, and $\mathrm{A}_a(0)$ always contains those agents that are feature-wise identical to agent a. The remaining (inductive) case is also straightforward, as a given agent b is in $\mathrm{A}_a(n+1)$ (formula: $\mathrm{InLay}_{a,n+1}(b)$) if and only if it is not in any 'lower' layer (formula: $\bigwedge_{k=0}^{n} \neg\, \mathrm{InLay}_{a,k}(b)$) and every agent that is not in a 'lower' layer is at most as close to a than b herself (formula: $\bigwedge_{c \in \mathsf{A}} \big(\bigwedge_{k=0}^{n} \neg\, \mathrm{InLay}_{a,k}(c) \;\rightarrow\; \mathrm{Closer}_{a \cdot b \cdot c} \big)$).

Finally, given Definition 3.2, it follows that the following $\mathcal{L}_{\bowtie_\lambda}$-validity characterizes the way the social relation changes:

$$\Vdash [\bowtie_\lambda]\, \mathrm{S}_{ab} \;\leftrightarrow\; \bigvee_{k=0}^{\ell_a(\lambda)} \mathrm{InLay}_{a,k}(b)$$

[5] More precisely, the formula states that there is at least one set of features P', of size t, such that a and b differ in all features in P' and coincide in all features in $\mathsf{P} \setminus \mathsf{P}'$. There can be a most one such set; therefore the formula is true exactly when a and b differ in exactly t features.

[6] More precisely, the formula states that there are $j_1, j_2 \in \{0, \ldots, |\mathsf{P}|\}$, with $j_1 \leq j_2$, such that j_1 is the distance from a to b_1, and j_2 is the distance from a to b_2.

In words, after a bounded similarity update agent a will have agent b in her social network, $[\bowtie_\lambda]\,\mathrm{S}_{ab}$, if and only if, before the update, agent b was in some of the layers whose agents will be part of a's social network, $\bigvee_{k=0}^{\ell_a(\lambda)} \mathrm{InLay}_{a,k}(b)$.

As only the social relation changes in the new model, we have the following.

Theorem 3.1 *The reduction axioms and the rule on Table 1 provide, together with a propositional axiom system schema, a sound and strongly complete axiom system characterising the validities of the dynamic language $\mathcal{L}_{\bowtie_\lambda}$ (for a finite set of features and a finite set of agents).*

Table 1. Axiom system for $\mathcal{L}_{\bowtie_\lambda}$ over social network models.

$\vdash [\bowtie_\lambda]\, p_a \leftrightarrow p_a$	for $a \in \mathsf{A}$	From $\vdash \varphi$ infer $\vdash [\bowtie_\lambda]\,\varphi$
$\vdash [\bowtie_\lambda]\,\mathrm{S}_{ab} \leftrightarrow \bigvee_{k=0}^{\ell_a(\lambda)} \mathrm{InLay}_{a,k}(b)$	for $a, b \in \mathsf{A}$	From $\vdash \psi_1 \leftrightarrow \psi_2$ infer $\vdash \varphi \leftrightarrow \varphi\,[\psi_2/\psi_1]$ (with $\varphi\,[\psi_2/\psi_1]$ any formula obtained by replacing one or more occurrences of ψ_1 in φ with ψ_2).
$\vdash [\bowtie_\lambda]\,\neg\varphi \leftrightarrow \neg\,[\bowtie_\lambda]\,\varphi$		
$\vdash [\bowtie_\lambda](\varphi \wedge \psi) \leftrightarrow ([\bowtie_\lambda]\,\varphi \wedge [\bowtie_\lambda]\,\psi)$		

If the relation $S_{\bowtie_\lambda}$ is forced to be irreflexive following the suggestion above, it is enough to restrict the current axiom characterizing the new social relation to cases in which a and b are different agents, and then add an additional axiom expressing that S_{aa} is never the case after the update operation.

$$\vdash [\bowtie_\lambda]\,\mathrm{S}_{ab} \leftrightarrow \bigvee_{k=0}^{\ell_a(\lambda)} \mathrm{InLay}_{a,k}(b) \quad \text{for } a \neq b, \qquad\qquad \vdash [\bowtie_\lambda]\,\mathrm{S}_{aa} \leftrightarrow \bot$$

For the variation in which the new social relation relies on personal group-size restrictions, the only needed change is the upper limit of the social network axiom: instead of the 'general' $\ell_a(\lambda)$, the 'personal' $\ell_a(\Lambda(a))$ should be used.

4 A Restriction to *relevant* Features

Any social-network-creation operation, such as the threshold update of [1] or the bounded update of Definition 3.2, can be seen as a 'public conversation' where all agents 'discuss' their features. Then, as the 'conversation' continues, agents will form subgroups of people sharing prior common interests.

When looking at social network creation from this perspective, it becomes clear that not all features can be 'discussed' at once: just some of them will be relevant at each stage of the discussion. This is not a novel idea; in [13], the authors use a game theoretic setting to define the agreement and disagreement of agents on a specific feature (or issue), which yields a way for them to update the social relation of agents with respect to one specific feature at a time.

This section explores this idea within the bounded update operation of the previous section: only a subset of all features will be relevant for each update. The resulting setting will allow us to describe more realistic scenarios, such as the step-by-step interaction in real dialogues (when personal features are slowly revealed as the conversation goes on), or cases in which agents control when one of their features becomes visible to other agents (e.g. when agents choose to expose some 'private' information only in specific circumstances).

The crucial step in this generalisation is the definition of a notion of distance that is relative only to a subset of features $\mathsf{Q} \subseteq \mathsf{P}$.

Definition 4.1 (Q-Distance). *Let $M = \langle \mathsf{A}, S, V \rangle$ be a SNM, and let $\mathsf{Q} \subseteq \mathsf{P}$ be a set of features. The* Q-distance *between a and b in M (that is, the distance between a and b in M* relative to features in Q*) is given by*

$$\text{DIST}_M^{\mathsf{Q}}(a, b) := |\text{MSMTCH}_M(a, b) \cap \mathsf{Q}|$$

Thus, $\text{DIST}_M^{\mathsf{Q}}(a, b)$ returns the number of atoms in Q on which a and b differ. Then, while some agent b_1 might be strictly closer to agent a than another agent b_2 with respect to all features, agent b_2 might be strictly closer to a than b_1 with respect to some strict subset of them.[7]

With this notion of Q-distance (still a semi-metric, as it satisfies non-negativity, symmetry, reflexivity and subadditivity, but not a metric, as it fails to satisfy the identity of indiscernibles), one can define 'relative to Q' variants of the qualitative '*distance from a*' relation and the sequence of layers it induces.

Definition 4.2 *Given a social network model $M = \langle \mathsf{A}, S, V \rangle$, an agent $a \in \mathsf{A}$ and a subset of features $\mathsf{Q} \subseteq \mathsf{P}$, the quantitative notion of Q-distance DIST^{Q} induces a qualitative (total, reflexive, transitive and well-founded) relation $\preccurlyeq_a^{\mathsf{Q},M} \subseteq \mathsf{A} \times \mathsf{A}$ of* Q-distance from *a. Such a relation is given by*

$$\preccurlyeq_a^{\mathsf{Q},M} := \{(b_1, b_2) \in \mathsf{A} \times \mathsf{A} : \ \text{DIST}_M^{\mathsf{Q}}(a, b_1) \leq \text{DIST}_M^{\mathsf{Q}}(a, b_2)\}$$

and thus $b_1 \preccurlyeq_a^{\mathsf{Q},M} b_2$ indicates that, with respect to the features in Q, agent b_1 is at least as close to agent a as agent b_2 in model M. By defining the notion of $\preccurlyeq_a^{\mathsf{Q},M}$-minimum in the standard way (for $\mathsf{B} \subseteq \mathsf{A}$, take $\text{MIN}_a^{\mathsf{Q}}(\mathsf{B}) := \{b \in \mathsf{B} : b \preccurlyeq_a^{\mathsf{Q},M} b'$ for all $b' \in \mathsf{B}\}$), this relation induces the following sequence of layers on A, each one containing agents equally distant from a:

$$\mathrm{A}_{-1}^{\mathsf{Q}}(a) := \varnothing, \qquad \mathrm{A}_0^{\mathsf{Q}}(a) := \text{MIN}_a^{\mathsf{Q}}(\mathsf{A}), \qquad \mathrm{A}_{n+1}^{\mathsf{Q}}(a) := \text{MIN}_a^{\mathsf{Q}}(\mathsf{A} \setminus \bigcup_{k=-1}^{n} \mathrm{A}_k^{\mathsf{Q}}(a)).$$

Similarly we now restrict the definition of the bounded similarity update to a version that is set to be relative to a given subset of features Q.

Definition 4.3 (Bounded Q-similarity update). *Let $M = \langle \mathsf{A}, R, V \rangle$ be a SNM and $\mathsf{Q} \subseteq \mathsf{P}$ a subset of features; take $\lambda \in \mathbb{N}$. Denote by $\ell_a^{\mathsf{Q}}(\lambda)$ the 'last'*

[7] For an example, take a model with $V(a) = \{p, q, r\}$, $V(b_1) = \{q, r\}$ and $V(b_2) = \{p\}$. Then, $\text{DIST}_M^{\{p,q,r\}}(a, b_1) = 1 < 2 = \text{DIST}_M^{\{p,q,r\}}(a, b_2)$, but nevertheless $\text{DIST}_M^{\{p\}}(a, b_2) = 0 < 1 = \text{DIST}_M^{\{p\}}(a, b_1)$.

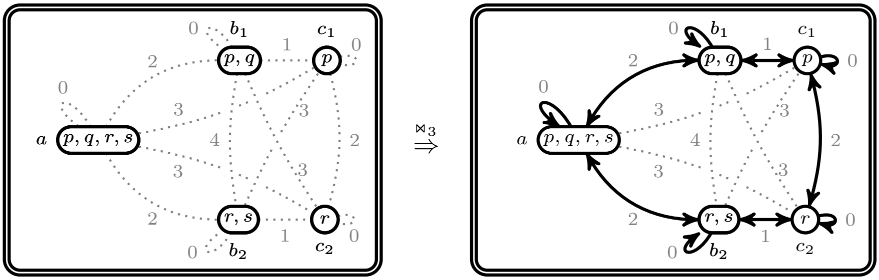
layer of contacts an agent $a \in \mathsf{A}$ *can add to her network without going above the maximum group size* λ*, i.e.,*

$$\ell^{\mathsf{Q}}_{a}(\lambda) := \max\{n \in \mathbb{N} \cup \{-1\} \,:\, |\bigcup_{k=-1}^{n} \mathrm{A}^{\mathsf{Q}}_{k}(a)| \leq \lambda\}$$

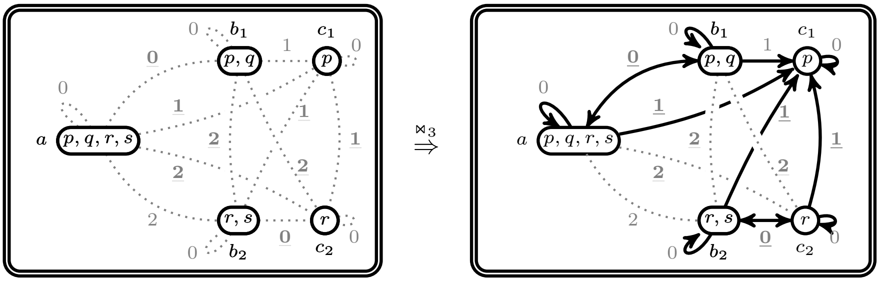
The bounded Q-similarity update *on* M *produces the* SNM $M_{\bowtie^{\mathsf{Q}}_{\lambda}} = \langle \mathsf{A}, S_{\bowtie^{\mathsf{Q}}_{\lambda}}, V \rangle$, *with its social relation given by*

$$S_{\bowtie^{\mathsf{Q}}_{\lambda}} := \{(a,b) \in \mathsf{A} \times \mathsf{A} \,:\, b \in \bigcup_{k=-1}^{\ell^{\mathsf{Q}}_{a}(\lambda)} \mathrm{A}^{\mathsf{Q}}_{k}(a)\}$$

Example 4.1 Consider the SNM models of the counterexample for Euclideanity (Fact 3.1 on page 5), drawn again below.



The SNM above on the left shows the resulting social network when all features $\{p, q, r, s\}$ are 'put on the table'. But suppose that this is not the case; then, the operation produces different social networks. For example, if the agents only 'talk' about features in $\{p, q\}$, then the distances are as shown in the model below on the left (underlined numbers emphasising distances that differ from the original 'fully open' situation):



The resulting SNM is shown above on the right. As expected, the social network relation is different. Note, in particular, how while the relation is still reflexive, it is not symmetric anymore; however, it is now transitive. Also interesting is the fact that, although c_1 is considered 'a friend' by everybody, she is the only member of her own social network: all other agents are at a distance of 1, and thus adding all of them would have taken her above the limit $\lambda = 3$. In fact, by restricting the conversation to the issues in $\{p, q\}$, the resulting network can be seen as three fully connected clusters, $\{a, b_1\}$, $\{b_2, c_2\}$ and $\{c_1\}$, with the members of the firsts pointing asymmetrically to the lone member of the last.

Dynamic Language $\mathcal{L}_{\bowtie_\lambda^\mathsf{Q}}$. The language $\mathcal{L}_{\bowtie_\lambda^\mathsf{Q}}$ is similar to $\mathcal{L}_{\bowtie_\lambda}$; the only difference lies in the semantic interpretation of its 'dynamic' operator, $\bowtie_\lambda^\mathsf{Q}$.

Definition 4.4 (Language $\mathcal{L}_{\bowtie_\lambda}$). *The language $\mathcal{L}_{\bowtie_\lambda^\mathsf{Q}}$ extends $\mathcal{L}$ with a modality $[\bowtie_\lambda^\mathsf{Q}]$ to build formulas of the form $[\bowtie_\lambda^\mathsf{Q}]\,\varphi$ (*"after a bounded Q-similarity update, φ is the case"*). The semantic interpretation of this modality refers to the relativised bounded updated model of Definition 4.3 as follows. Let M be a SNM; then,*

$$M \Vdash [\bowtie_\lambda^\mathsf{Q}]\,\varphi \quad \textit{iff}_{def} \quad M_{\bowtie_\lambda^\mathsf{Q}} \Vdash \varphi.$$

With respect to an axiom system, the system presented in Table 1 can be used almost verbatim. The only change refers to the axiom characterising the new social network relation, which should now be relativised to the subset of features Q; for this, it is enough to replace P with Q in the definitions for formulas $\text{Dist}^t_{a\cdot b}$ and $\text{Closer}_{a\cdot b_1\cdot b_2}$ (page 7), thus obtaining formulas $\text{Dist}^{\mathsf{Q},t}_{a\cdot b}$, $\text{Closer}^{\mathsf{Q}}_{a\cdot b_1\cdot b_2}$ and $\text{InLay}^{\mathsf{Q}}_{a,k}(b)$.

5 Conclusions and Future Work

Following the cognitive science literature (in particular, [2]), we have examined social networks (Sect. 2) by studying a *group-size* approach to social network creation based on the initial idea (see Sect. 3) in which *"agent a will consider agent b to be part of her social network if and only if b is within the λ closest agents to a"*. This proposal can be seen as an alternative to the approach of [1], which uses a *threshold* to establish how similar an agent should be to be incorporated to someone's social environment (i.e. *"agent a will consider agent b to be part of her social network if and only if b's distance from a is at most θ"*). Moreover, the relativised version studied in Sect. 4 allows for the representation of more refined scenarios where not all features play a role during the agents' interaction.

The work presented here and in [1] form the initial steps in the study of the logical structure behind social network creation, and they already suggest interesting alternatives. While both the threshold and the group-size approaches relate agents when they are similar enough in their features, behavior, etc., one can think of an alternative scenario in which one considers the dual situation so that agents connect when they *complement each other*. In order to deal formally with this *complementary* idea, a more fine-grained setting is needed that takes into account not only the agents' features/behaviors, as in this paper, but also their doxastic state and their preferences (e.g., [14,15]).

Another straightforward generalization would be to consider not a single social network, but rather a collection of them. A slightly more realistic approach in this direction is to understand each feature not as a simple choice between *"yes"* and *"no"*, but rather as a choice among a finite range of values. Then the model can support a social network for each feature $p \in \mathsf{P}$, and agents can be grouped according to the value they assign to each such p. After all, someone who chooses football as her favourite sport and Lady Gaga as her favourite musician is bound to have different social environments in each one of these contexts.

A further route will lead us into a combined social network and epistemic study. This is another natural next step, as what matters most when establishing friendship is maybe not the agents' features and differences, but rather what one knows about them. Our work in [1] provides an initial exploration in this direction, using the threshold update approach.

In a related track, one can explore cases in which certain features are taken to be more important than others in such a way that this 'priority ordering' among features differs from agent to agent. This allows for the representation of interesting situations: the number of differences between agents a and b_1 might be very large, and yet they may agree on the feature a cares about the most. Then, a might consider that b_1 is 'closer to her' than some b_2 with whom she shares all but this most important feature. The combination of this (the explored setting in which the update is relative to only a subset of features) and an epistemic setting would allow us to describe situations where strategic behaviour plays an important role. For example, if agent a knows that she and agent b differ in some feature $p \in \mathsf{P}$, and she also knows that p is the most important feature for b, then she (a) might want to keep this topic out of the conversation, at least until it has been commonly established (i.e., it is common knowledge between a and b) that they are similar with respect to several other features. The setting becomes even more interesting when the new social network is defined not only in terms of the agents' similarities, but also in terms of existing social connections (cf. the middleman cases in [1]). In such cases, the features discussed at the beginning will define the social connections that will be available in further stages.

Finally, we observe the importance of the interplay between the social network changing operations of this proposal, and the operations that change the features (or behaviour/beliefs) in [3]. Both ideas deserve to be studied in tandem, as indeed the dynamics studied in one can affect the dynamics studied in the other.

References

1. Smets, S., Velázquez-Quesada, F.R.: How to make friends: a logical approach to social group creation. In: Baltag, A., Seligman, J., Yamada, T. (eds.) LORI 2017. LNCS, vol. 10455, pp. 377–390. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-55665-8_26
2. Dunbar, R.I.M.: Neocortex size as a constraint on group size in primates. J. Hum. Evol. **22**(6), 469–493 (1992)

3. Baltag, A., Christoff, Z., Rendsvig, R.K., Smets, S.: Dynamic epistemic logics of diffusion and prediction in social networks (extended abstract). In: Bonanno, G., van der Hoek, W., Perea, A. (eds.) Proceedings of LOFT 2016 (2016)
4. Seligman, J., Liu, F., Girard, P.: Logic in the community. In: Banerjee, M., Seth, A. (eds.) ICLA 2011. LNCS (LNAI), vol. 6521, pp. 178–188. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-18026-2_15
5. Liu, F., Seligman, J., Girard, P.: Logical dynamics of belief change in the community. Synthese **191**(11), 2403–2431 (2014)
6. Christoff, Z., Hansen, J.U., Proietti, C.: Reflecting on social influence in networks. J. Logic Lang. Inf. **25**(3–4), 299–333 (2016)
7. Deza, M.M., Deza, E.: Encyclopedia of Distances. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00234-2
8. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In: Gilboa, I. (ed.) Proceedings of TARK 1998, pp. 43–56. Kaufmann, San Franscisco (1998)
9. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Synthese Library Series, vol. 337. Springer, The Netherlands (2008). https://doi.org/10.1007/978-1-4020-5839-4
10. van Benthem, J.: Logical Dynamics of Information and Interaction. Cambridge University Press, Cambridge (2011)
11. Roberts, S.G.B., Wilson, R., Fedurek, P., Dunbar, R.I.M.: Individual differences and personal social network size and structure. Pers. Individ. Differ. **44**(4), 954–964 (2008)
12. Roberts, S.G.B., Dunbar, R.I.M., Pollet, T.V., Kuppens, T.: Exploring variation in active network size: constraints and ego characteristics. Social Netw. **31**(2), 138–146 (2009)
13. Solaki, A., Terzopoulou, Z., Zhao, B.: Logic of closeness revision: challenging relations in social networks. In: Köllner, M., Ziai, R. (eds.) Proceedings of the ESSLLI 2016 Student Session, pp. 123–134 (2016)
14. Baltag, A., Smets, S.: A qualitative theory of dynamic interactive belief revision. In: Bonanno, G., van der Hoek, W., Wooldridge, M. (eds.) Logic and the Foundations of Game and Decision Theory (LOFT7). Texts in Logic and Games, vol. 3, pp. 13–60. Amsterdam University Press, Amsterdam (2008)
15. Ghosh, S., Velázquez-Quesada, F.R.: Agreeing to agree: reaching unanimity via preference dynamics based on reliable agents. In: Weiss, G., Yolum, P., Bordini, R.H., Elkind, E. (eds.) Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, 4–8 May 2015, pp. 1491–1499. ACM (2015)

Dynamic Preference Logic as a Logic of Belief Change

Marlo Souza[1(✉)], Álvaro Moreira[2], and Renata Vieira[3]

[1] Institute of Mathematics and Statistics, Federal University of Bahia - UFBA, Av. Adhemar de Barros, S/N, Ondina, Salvador, BA, Brazil
marlo@dcc.ufba.br

[2] Institute of Informatics, Federal University of Rio Grande do Sul - UFRGS, Av. Bento Gonçalves, 9500, Porto Alegre, RS, Brazil
alvaro.moreira@inf.ufrgs.br

[3] Faculty of Informatics, Pontifical Catholic University of Rio Grande do Sul - PUCRS, Av. Ipiranga, 6681, Porto Alegre, RS, Brazil
renata.vieira@pucrs.br

Abstract. AGM's belief revision is one of the main paradigms in the study of belief change operations. Recently, several logics for belief and information change have been proposed in the literature, which were used to encode belief change operations in a rich and expressive framework. While the connection of AGM-like operations and their encoding in dynamic doxastic logics have been studied before, by the exceptional work of Segerberg, most work on the area of Dynamic Epistemic Logics (DEL) have not attempted to characterize belief change operators by means of their logical properties. This work investigates how Dynamic Preference Logic, a logic in the DEL family, can be used to characterise properties of dynamic belief change operators, focusing on well-known postulates of iterated belief change.

Keywords: Dynamic Epistemic Logic · Dynamic Preference Logic Belief revision

1 Introduction

Belief Change is the multidisciplinary area that studies how a doxastic agent comes to change her mind after acquiring new information. Currently, the most influential model for belief change is the so-called AGM paradigm, named after the authors of its seminal work [1]: C. Alchourrón, P. Gädenfors and D. Makinson. Although AGM's approach and hypothesis have been questioned in the literature [7,19], it has brought profound developments for the problem of belief dynamics, influencing areas such as Computer Science, Artificial Intelligence, and Philosophy.

While changes in mental attitudes have been a well studied topic in the literature, the integration of such operations within the logics of beliefs, obligations, desires and others is a somewhat recent development. This shift from extra-logical characterisation of changes in the agents attitudes to their integration

A. Madeira and M. Benevides (Eds.): DALI 2017, LNCS 10669, pp. 185–200, 2018.
https://doi.org/10.1007/978-3-319-73579-5_12

within the representation language has important expressibility consequences. It allows, for example, the study the dynamics of introspective beliefs, which is not representable in axiomatic approach of the AGM framework.

Recently, inspired by the Dutch School, several dynamic logics for information change have been proposed [4,10,14,24]. Particularly, Girard [9,10] proposes Dynamic Preference Logic (DPL) which has been applied to study generalization of belief revision *a la* AGM [9,10]. Following this trend, many different works have encoded well-known belief change operators within this logic[1] [9,14,23,24] and used them to study dynamic behaviour of attitudes such as Preferences, Beliefs, Intentions, etc.

It is not clear in these works, however, how can one use the dynamic logic to investigate the properties of classes of change operators, or even whether a given operator satisfies some well-established desirable properties. As such, given a dynamic logic with a program π, it is not clear how one can use the logic to establish which postulates (from the area of belief change) are satisfied by the operation described by the program π or whether this operation is part of a given class of well-known operators - e.g. Darwiche and Pearl's [7] iterated belief revision operators.

In this work, we study the relationship between the postulates satisfied by belief change operators and the axioms valid in Dynamic Preference Logic using these operators. This study differs from previews research on the connection of the results of Belief Change following AGM tradition and Dynamic Epistemic Logics by employing the validities of the logic to characterize its dynamic operators, instead of encoding in the logic those operators which are known beforehand to satisfy certain desirable properties. As such, we will use the proof theory of Dynamic Preference Logic to investigate which postulates a given dynamic operator satisfies. We wish to point out that, while the focus of our work will be single-agent belief change, our results can be trivially extended to *private changes* in the multi-agent case.

As a result, we also establish a method for deriving axiomatisations of dynamic operators in Dynamic Preference Logic. Our method differs from that proposed by Van Benthem and Liu [28], or to that of Aucher [2], which is based on Propositional Dynamic Logic without iteration, by using the extensively studied postulates from belief change to derive an axiomatisation of the logic with a given operation. As such, our method is applicable to a wider class of dynamic belief change operators, including those which cannot be encoded using Propositional Dynamic Logic programs, e.g. lexicographic contraction [22].

The structure of this work is as follows: in Sect. 2, we present the main approaches in the area of iterated belief change and its postulates; in Sect. 3, we introduce Dynamic Preference Logic, a logic in the tradition of Dynamic Epistemic Logic recently applied to study belief change; in Sect. 4 we show how the

[1] Similarly, works such as that of Baltag and Smets' [4] concern themselves with how to encode some types of belief change using the framework of Dynamic Epistemic Logic, not clearly delineating, however, how the properties of a given operator are reflected in the resulting logic.

postulates satisfied by a dynamic operator imply certain axioms in the Dynamic Preference Logic, and how we can use these postulates to derive an axiomatisation for this logic. In Sect. 5, we discuss some of the related work, and, finally, in Sect. 6, we present our final considerations.

2 Dynamic and Iterated Belief Change

AGM's initial work [1] focused on defining the requirements for rational changes of the agent's beliefs. It is based on three different belief changing operations: expansion, contraction and revision. Belief expansion is the operation of blindly integrating a new piece of information into the agents beliefs. Belief contraction is the operation of removing a currently believed sentence from the agent's set of beliefs, with minimal alterations. Finally, belief revision is the operation of consistently integrating new information into an agent's beliefs. These three operations are interconnected by the properties known as Levi and Harper identities [1].

The AGM approach has primarily studied the structural properties regarding belief change. These structural properties encode the rational or desired conditions to guarantee the minimization of changes in the belief system - modelled in this paradigm as a closed set of formulas, called the belief set. Among the three basic operations studied by AGM, only expansion can be univocally defined. The other two operations are defined by a set a rational constraints, or postulates, usually referred to as the AGM postulates or the Gärdenfors postulates.

While AGM theory is independent of the syntax of the supporting logic, it lacks a clear semantic interpretation for its operations. One such interpretation was provided by Grove [11] using a possible-world semantics, based on Lewis' semantics for counterfactuals. Grove's model for the operation of belief revision has not only clarified the meaning of belief change operations but, also, has become a necessary tool for the development of new methods and operations in the area, such as the iterated belief operations we will discuss in Subsect. 2.1.

A Grove system of spheres for a belief set B is a pair $S_B = \langle W, \leq \rangle$ where W is the set of all models for the logic $\mathcal{L}$ and $\leq \subseteq W \times W$, s.t. the relation $\leq$ is connected and transitive and for any $S \subseteq W$, if $S \neq \emptyset$, then exists $x \in S$ minimal (in respect to $\leq$) in S, and, x is minimal in W iff $x \vDash B$.

Grove shows that for any belief revision operator $*$ satisfying the AGM postulates (R-1)- (R-8), there is a system of spheres $S_B = \langle W, \leq \rangle$ such that $[\![B * \varphi]\!]_{S_B} = Min_{\leq}[\![\varphi]\!]_{S_B}$, where $[\![\varphi]\!]_{S_B} = \{w \in W \mid w \vDash \varphi\}$ is the set of all worlds satisfying a formula (set of formulas) φ and

$$Min_{\leq} S = \{w \in S \mid \nexists w' \in S \text{ s.t. } w' \leq w \wedge w \nleq w'\}$$

is the set of minimal elements of S, according to the pre-order $\leq$.

There are numerous other proposals of belief change operators in the literature. In this work, however, we limit our analysis to those operators possessing a semantic characterization based on systems of spheres. As such, our aim is to encode the postulates discussed further in this section within the Dynamic Preference Logic, a dynamic epistemic logic which has been applied in the literature to the study of belief change [9,10,14,22,23].

2.1 Iterated Belief Revision

AGM belief revision says very little about how to change one agent's beliefs repeatedly. In fact, it has been observed that the AGM approach allows some counter-intuitive behaviour in the iterated case[2].

To remedy this deficiency of AGM's postulates, Darwiche and Pearl [7] (DP) propose a set of additional postulates that further constrain the behaviour of revision operators.

(DP1) If $\vdash \beta \rightarrow \alpha$ then $(B * \alpha) * \beta = B * \beta$
(DP2) If $\vdash \beta \rightarrow \neg\alpha$ then $(B * \alpha) * \beta = B * \beta$
(DP3) If $\alpha \in B * \beta$ then $\alpha \in (B * \alpha) * \beta$
(DP4) If $\neg\alpha \notin B * \beta$ then $\neg\alpha \notin (B * \alpha) * \beta$

Nayak et al. [17] show that DP postulates are incompatible with AGM's. To solve this problem, they propose the notion of dynamic revision operator, in which a belief revision changes not only the belief set of the agent but the operation itself, i.e. the agent belief state. This distinction between *static* and *dynamic* operators has been observed to be relevant in works such as that of Van Benthem [24] and Baltag and Smets [4], or that of Lindström and Rabinowicz [13], in which AGM-like static revision can be seen as a counterfactual reasoning while dynamic revision is modelled as a doxastic action changing the agent's doxastic state.

From a conceptual perspective, the change from static to dynamic belief change has the implication that the result of some change operation is not simply changing the agent's beliefs, but changing the agent's belief state itself. As such, semantically, we ought to describe belief changing operations by the changes they perform in the agent's belief state, such as in a system of spheres.

In the context of dynamic revision operators, Nayak et al. [17] show that the DP postulates are equivalent to requiring that a belief revision of a Grove system of spheres with plausibility relation $\leq$ corresponds to change $\leq$ into $\leq_{*\varphi}$ satisfying the following postulates:

(DP-1) If $w, w' \in [\![\varphi]\!]$, then $w \leq_{*\varphi} w'$ iff $w \leq w'$
(DP-2) If $w, w' \notin [\![\varphi]\!]$, then $w \leq_{*\varphi} w'$ iff $w \leq w'$
(DP-3) If $w' \notin [\![\varphi]\!]$ and $w' \in [\![\varphi]\!]$, then $w <_{*\varphi} w'$ only if $w < w'$
(DP-4) If $w' \notin [\![\varphi]\!]$ and $w' \in [\![\varphi]\!]$, then $w \leq_{*\varphi} w'$ only if $w \leq w'$

The authors show, further, that DP postulates are overpermissible, in the sense that they allow revision operators which they claim to possess undesirable properties. Based on this properties, Nayak et al. [17] propose the operation

[2] Some classical examples were proposed by Darwich and Pearl [7].

of simple lexicography which can be characterized by the properties (DP-1), (DP-2) and (Rec) below. The axiom of recalcitrance states that if two pieces of information φ and ψ are consistent with each other, then if we obtain the information φ and, later, the information ψ, there is no ground to discard φ.

(Rec) If $w \in [\![\varphi]\!]$ and $w' \notin [\![\varphi]\!]$, then $w <_{*\varphi} w'$.

Jin and Thielscher [12], on the other hand, provide a different analysis of the shortcomings of DP postulates based on an analysis of conditional beliefs in systems of spheres. The authors propose a notion of conditional beliefs and show that an DP-compliant operator may create arbitrary conditional beliefs in the revised belief set. The authors argue that both the DP postulates and Recalcitrance allow operations that discard too much information from the belief set. Aiming to provide a condition that guarantees the minimization of the loss of information from a belief set, the authors propose the postulate of Independence (Ind), which state that if two pieces of information φ and ψ are independent from each other, there is no base to discard φ upon discovering ψ. This postulate can be stated as:

(Ind) If $w \in [\![\varphi]\!]$ and $w' \notin [\![\varphi]\!]$, then $w \leq w'$ implies $w <_{*\varphi} w'$.

2.2 Iterated Belief Contraction

Based on a generalization of the Levi and Harper identities [1], Nayak et al. [16] propose three iterated contraction operators, namely Natural Contraction, Moderate Contraction and Lexicographic Contraction[3], and analyse their properties. Later, Ramachandran et al. [18] provided the following set of postulates to characterise these operators, where contracting φ from a system of spheres with relation $\leq$, corresponds to change $\leq$ into $\leq_{-\varphi}$. Here (GR) is the minimal property a contraction should satisfy to be AGM-compatible, as showed by Grove [11].

(GR) If $w \in Min_{\leq}W$ or $w \in Min_{\leq}[\![\neg\varphi]\!]$, then $w \leq_{-\varphi} w'$ for any $w' \in W$.
(NC) If $w \notin Min_{\leq}W$ and $w \notin Min_{\leq}[\![\neg\varphi]\!]$, then for any $w' \in W$, $w \leq_{-\varphi} w'$ iff $w \leq w'$.
(MC) If $w \notin [\![\varphi]\!]$, $w' \in [\![\varphi]\!]$ and $w' \notin Min_{\leq}W$, then $w \leq_{-\varphi} w'$.

In this work we will explore how the properties (or postulates) discussed in this section can be encoded inside Dynamic Preference Logic, i.e. how we can guarantee that a given dynamic operator of the logic satisfies one of these postulates. For that, in the following section, we introduce Dynamic Preference Logic based on the work of Girard [9] and of Souza [22].

[3] In this work we will not investigate lexicographic contraction due to space constraints.

3 Dynamic Preference Logic

Preference Logic (or Order Logic as named by Girard [9]) is a modal logic complete for the class of transitive and reflexive frames. It has been applied to model a plethora of phenomena in Deontic Logic [27], Logics of Preference [26], Logics of Belief [4], etc. Dynamic Preference Logic (DPL) [9] is the result of "dynamifying" Preference Logic, i.e. extending it with dynamic modalities. This logic is one example among several proposed Dynamic Epistemic Logics in the field used to study the dynamics of mental attitudes and it is particularly interesting for its expressibility, allowing the study of dynamic phenomena of attitudes such as Beliefs, Obligations, Preferences etc.

3.1 Preference Logic

We begin our presentation with the language and semantics of (static) Preference Logic, which we will later "dynamify" it.

Definition 1. *Let P be a set of propositional letters. We define the language $\mathcal{L}_{\leq}(P)$ by the following grammar (where $p \in P$):*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid A\varphi \mid [\leq]\varphi \mid [<]\varphi$$

We will often refer to the language $\mathcal{L}_{\leq}(P)$ simply as $\mathcal{L}_{\leq}$, by supposing the set P is fixed. Also, we will denote the language of propositional formulas, i.e. the language removing all modal formulas from $\mathcal{L}_{\leq}(P)$, by $\mathcal{L}_0(P)$ or simply $\mathcal{L}_0$.

Definition 2. *A preference model*[4] *is a tuple $M = \langle W, \leq, v \rangle$ where W is a set of possible worlds, $\leq$ is a reflexive, transitive relation over W, with a well-founded strict part $<$, and $v : P \to 2^W$ a valuation function.*

In such a model, the accessibility relation $\leq$ represents an ordering of the possible worlds according to the preferences of a certain agent. As such, given two possible worlds $w, w' \in W$, we say that w is at least as preferred as w' if, and only if, $w \leq w'$. While we will commonly use the term 'preference relation', the interpretation for that relation depends on the application of the logic. As such, when using preference logic to encode beliefs, the accessibility relation $\leq$ is commonly referred as a 'plausibility relation' among worlds, denoting which state of affairs the agent beliefs to be more plausible. In this context, DPL can be viewed as an epistemic doxastic logic, related to that of Baltag and Smets [4], which has been applied to the study of AGM-like belief change in the literature [9,10,14,22].

The interpretation of the formulas over these models is defined as usual. We will only present the interpretations for the modalities, since the semantics of

[4] Also called order model in [9] and modal betterness model in [14].

the propositional connectives is clear. The A modality is an universal modality[5], while $[\leq]$ modality is a box modality on the accessibility order $\leq$. The $[<]$ modality is the strict variant of $[\leq]$. They are interpreted as:

$$\begin{array}{lll} M, w \vDash A\varphi & \text{iff} & \forall w' \in W : M, w' \vDash \varphi \\ M, w \vDash [\leq]\varphi & \text{iff} & \forall w' \in W : w' \leq w \Rightarrow M, w' \vDash \varphi \\ M, w \vDash [<]\varphi & \text{iff} & \forall w' \in W : w' < w \Rightarrow M, w' \vDash \varphi \end{array}$$

We will refer as $\langle\leq\rangle\varphi$ to the formula $\neg[\leq]\neg\varphi$, as commonly done in modal logic (and similar to the formula $\langle<\rangle\varphi$).

Notice that the inclusion of the universal modality is necessary to represent important notions and belief changing operations, such as conditional modalities, or greatly simplify their representation in the logic.

As usual, given a model M and a formula φ, we use the notation $[\![\varphi]\!]_M$ to denote the set of all the worlds in M satisfying φ. When it is clear to which model we are referring to, we will denote the same set by $[\![\varphi]\!]$. Also, we will denote the minimal elements of $[\![\varphi]\!]$, according to the relation $\leq$, by the notation $Min_{\leq}[\![\varphi]\!]$. This corresponds to the notion of 'most preferred worlds satisfying φ' in the model.

Notice that preference models, as defined in [9], need not to possess well-founded preference relations. It has been observed, however, e.g. by [10], that some important belief change operators are only well-defined if the plausibility relation $\leq$ satisfies the Lewis Limit Condition [11]. Since Lewis Limit condition is intrinsically dependent on the object language, some authors [4,10] propose the adoption of well-foundedness as a purely semantic codification of this property. Recently, Souza et al. [22,23] have provided complete axiomatizations for preference models satisfying the Lewis Limit Condition and with well-founded preference relations. Since we are interested in such belief change operators, e.g. iterated contractions, we limit our Preference Logic to consider only well-founded preference models in Definition 2.

3.2 Dynamifying Preference Logic

In this section, we study the dynamification of Preference logic by the introduction of dynamic modals, i.e. modalities representing operations on a model. In Sect. 4, we will study the relationship between the properties satisfied by a given dynamic operation and the axioms satisfied by dynamic logic defined by this operation.

We define a dynamic operation on a preference model as any operation that takes a preference model and a formula and changes the relation of the model. Let $Mod(\mathcal{L}_{\leq})$ denote the class of preference models.

[5] Notice that we are interpreting possible worlds in our model as *epistemically* possible worlds. As such, the universal modality, in our encoding, encodes epistemic necessity. Notice that we could introduce the modality $\sim$ as epistemic indistinguishably instead of epistemic necessity, as done by Baltag and Smets [4], obtaining the same results presented here.

Definition 3. *Let* $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_{\leq} \rightarrow Mod(\mathcal{L}_{\leq})$*, we say* $\star$ *is a dynamic operator on preference models if for any* $M = \langle W, \leq, v\rangle$ *and formula* $\varphi \in \mathcal{L}_{\leq}$*,* $\star(M, \varphi) = \langle W, \leq_{\star}, v\rangle$*. In other words, an operation on preference models is called a dynamic operator iff it only changes the relation of the preference model. We will use* $M_{\star\varphi}$ *to denote the model* $\star(M, \varphi)$*.*

In the definition above we limited our dynamic operators to not change the set of possible worlds. This limitation is justified by the fact that we are considering belief changing operators, which change the plausibility the agent attributes to each epistemically possible world, not creating nor revoking any epistemic certainties (i.e. knowledge).

Given a dynamic operator $\star$, we extend the language $\mathcal{L}_{\leq}$ with formulas $[\star\varphi]\psi$, obtaining the language $\mathcal{L}_{\leq}(\star)$. Here, we point out some abuse of notation, since the we use $\star$ as both a dynamic operator defined as a function and as a symbol in the object language to define the modality $[\star\varphi]$ - which will correspond to the application of this operator $\star$ to the model.

Definition 4. *Let* $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_{\leq} \rightarrow Mod(\mathcal{L}_{\leq})$ *be a dynamic operator. We define the language* $\mathcal{L}_{\leq}(\star)$ *as the smallest set containing* $\mathcal{L}_{\leq}$ *and all formulas* $[\star\varphi]\psi$*, with* $\varphi, \psi \in \mathcal{L}_{\leq}(\star)$*. Given a preference model* $M = \langle W, \leq, v\rangle$ *and a world* $w \in W$*, then* $M, w \vDash [\star\varphi]\psi$ *iff* $M_{\star\varphi}, w \vDash \psi$*.*

4 Iterated Belief Change and Dynamic Preference Logic

In this section, we will investigate the relationship between the postulates satisfied by iterated belief change operators discussed in Sect. 4 and the axioms satisfied in Dynamic Preference Logic using these operators.

$$
\begin{array}{ll}
[\star\varphi]p & \leftrightarrow p \\
[\star\varphi](\xi \wedge \psi) & \leftrightarrow [\star\varphi]\xi \wedge [\star\varphi]\psi \\
[\star\varphi]\neg\xi & \leftrightarrow \neg[\star\varphi]\xi \\
[\star\varphi]A\xi & \leftrightarrow A[\star\varphi]\xi
\end{array}
$$

Fig. 1. Basic axiom schemata for dynamic operators.

Proposition 5. *Let* $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_0 \rightarrow Mod(\mathcal{L}_{\leq})$ *be a dynamic operator. The operator* $\star$ *satisfies (DP-1) only if the axiom schemata in Fig. 1 and below are valid in* $\mathcal{L}_{\leq}(\star)$*.*

$$
\begin{array}{ll}
[\star\varphi][\leq]\xi & \rightarrow (\varphi \rightarrow [\leq](\varphi \rightarrow [\star\varphi]\xi)) \\
[\star\varphi][<]\xi & \rightarrow (\varphi \rightarrow [<](\varphi \rightarrow [\star\varphi]\xi)) \\
[\leq]\xi & \rightarrow (\varphi \rightarrow [\star\varphi][\leq](\varphi \rightarrow \xi)) \textit{ for } \xi \in \mathcal{L}_0 \\
[<]\xi & \rightarrow (\varphi \rightarrow [\star\varphi][<](\varphi \rightarrow \xi)) \textit{ for } \xi \in \mathcal{L}_0
\end{array}
$$

Proof. We will only show the case for the axiom $[\star\varphi][\leq]\xi \to (\varphi \to [\leq](\varphi \to [\star\varphi]\xi))$, since the others are either trivial or analogous to this case.

Let $M = \langle W, \leq, v\rangle$ be a preference model and $\star$ an operation on preference models that satisfy DP-1. Take $w \in [\![\varphi]\!]$ s.t. $M, w \vDash [\star\varphi][\leq]\xi$ for some ξ, then $M_{\star\varphi}, w \vDash [\leq]\xi$, i.e. for any $w' \in W$ s.t. $w' \leq_{\star\varphi} w$, it holds that $M_{\star\varphi}, w' \vDash \xi$. By definition, it holds that $M, w' \vDash [\star\varphi]\xi$. As such, take $w' \in W$ s.t. $w' \leq_{\star\varphi} w$. Suppose $M, w' \vDash \varphi$, then $M, w' \vDash [\star\varphi]\xi$ and, by DP-1, $w' \leq w$. As such, $M, w \vDash \varphi \to [\leq](\varphi \to [\star\varphi]\xi)$. □

The inverse implication requires an extra condition that for any set S of possible worlds, there is a propositional formula ξ_S that uniquely characterises it, i.e. for any world $w \in W$, $M, w \vDash \xi_S$ iff $w \in S$.

Proposition 6. *Let $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_0 \to Mod(\mathcal{L}_{\leq})$ be a dynamic operator. Let $\mathcal{M} \subseteq Mod(\mathcal{L}_{\leq})$ be a class of preference models s.t. for all $M = \langle W, \leq, v\rangle \in \mathcal{M}$ and for any set $S \subseteq W$ there is a propositional formula ξ_S characterising S. The operator $\star$ satisfies (DP1) if, and only if, the axiom schemata in Proposition 5 is $\mathcal{M}$-valid.*

Proof. Let $M = \langle W, \leq, v\rangle \in \mathcal{M}$ be a preference model and $\star$ a dynamic operator on preference models. Take $w \in W$ s.t. $w \in [\![\varphi]\!]$. (**If**) Let $S = \{w' \in W \mid w' \leq_{\star\varphi} w\}$, by hypothesis there is ξ_S s.t. all $M_{\star\varphi}, w' \vDash \xi_S$ iff $w' \in S$. Clearly, $M, w \vDash [\star\varphi][\leq]\xi_S$. By hypothesis the model satisfies the axiom schemata above, as such $M, w \vDash (\varphi \to [\leq](\varphi \to [\star\varphi]\xi_S))$. Since $w \in [\![\varphi]\!]$, then $M, w \vDash [\leq](\varphi \to [\star\varphi]\xi_S)$. As such, for any $w' \in [\![\varphi]\!]$, if $w' \leq w$ then $M, w' \vDash [\star\varphi]\xi_S$, i.e. $M_{\star\varphi}, w' \vDash \xi_S$. Since ξ_S characterises the set S, we conclude that $w' \in S$, which means that $w' \leq_{\star\varphi} w$ by construction of S. (**Only If**) Let $S = \{w' \in W \mid w' \leq w\}$, by hypothesis there is propositional ξ_S s.t. $M, w' \vDash \xi_S$ iff $w' \in S$. Clearly, $M, w \vDash [\leq]\xi_S$. Since the model satisfies the axiom schemata above, $M, w \vDash (\varphi \to [\star\varphi][\leq](\varphi \to \xi_S))$. Then, for $w' \in W$ s.t. $w' \leq_{\star\varphi} w$, it holds that $M_{\star\varphi}, w' \vDash \varphi \to \xi_S$. Take $w' \in [\![\varphi]\!]$ s.t. $w' \leq_{\star\varphi} w$, then $M_{\star\varphi}, w' \vDash \xi_S$, i.e. $M, w' \vDash [\star\varphi]\xi_S$. Since ξ_S is propositional formula, it holds (by simple induction on the structure of ξ_S) that $M, w' \vDash \xi_S$, i.e. $w' \leq w$. □

Notice that while the existence of *propositional* characterising formulas required in Proposition 6 is a rather strong requirement, it appears naturally in some models, as Grove's systems of spheres [11] and Souza's broad models [22]. While characteristic formulas, in general, can be crafted for finite models [15], we cannot guarantee that, after the application of a dynamic operator to the model, this formula can still characterize the desired set of worlds. It is not clear yet if the result can be generalized to discard the requirement of propositional characteristic formulas in DPL.

More yet, the condition of being able to identify worlds in a model seems to be necessary for such a characterisation. The reason for this is that, without the ability to characterise a world, the logic cannot distinguish between two modally equivalent worlds [5] and, as such, cannot guarantee (DP-1).

Fact 7. *If there are two modally equivalent worlds in a model M, which are not equally preferred to each other, there is a dynamic operator $\star$ s.t. $\mathcal{L}_{\leq}(\star)$ satisfies the axiom schemata in Proposition 5 but $\star$ does not satisfies (DP-1).*

Proof. Let $M = \langle W, \leq, v\rangle$ be a preference model and $w_1, w_2 \in W$ modally equivalent. Take the dynamic operator $\star_w$ s.t. $\star(M', \varphi) = M'$ if $M' \neq M$ or $\varphi \neq \top$ and $\star(M, \top) = \langle W, \preceq, v\rangle$, in which $w \preceq w'$ iff either (i) $w, w' \notin \{w_1, w_2\}$ and $w \leq w'$, (ii) $w = w' = w_1$, (iii) $w = w' = w_2$, (iv) $w = w_1$ and $w_2 \leq w'$, (v) $w = w_2$ and $w_1 \leq w'$, (vi) $w' = w_1$ and $w \leq w_2$, or (vii) $w' = w_2$ and $w \leq w_1$. Essentially, this operation switch the places of w_1 and w_2 in the order and, as such, does not satisfies (DP-1). Since w_1 and w_2 are modally equivalent, however, the axiom schemata of Proposition 5 holds for this operation. □

While Fact 7 concerns the necessity of characterising formulas for (DP-1), it is not difficult to see that it can be adapted for the other postulates as well. We point out that for all postulates presented in this work, results similar to Proposition 6 and Fact 7 can be proved and, as such, we will not present them. These results show that the ability to distinguish worlds in the model is a necessary condition for the characterization of iterated belief change postulates in DPL. As for (DP-1), we can provide characterisations of the other postulates by means of Dynamic Preference Logic axioms.

Proposition 8. *Let $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_0 \rightarrow Mod(\mathcal{L}_{\leq})$ be a dynamic operator on preference models. The operator $\star$ satisfies (DP2) only if the axiom schemata in Fig. 1 and below are valid in $\mathcal{L}_{\leq}(\star)$.*

$$\begin{array}{ll}
[\star\varphi][\leq]\xi & \rightarrow (\neg\varphi \rightarrow [\leq](\neg\varphi \rightarrow [\star\varphi]\xi)) \\
[\star\varphi][<]\xi & \rightarrow (\neg\varphi \rightarrow [<](\neg\varphi \rightarrow [\star\varphi]\xi)) \\
[\leq]\xi & \rightarrow (\neg\varphi \rightarrow [\star\varphi][\leq](\neg\varphi \rightarrow \xi)) \textit{ for } \xi \in \mathcal{L}_0 \\
[<]\xi & \rightarrow (\neg\varphi \rightarrow [\star\varphi][<](\neg\varphi \rightarrow \xi)) \textit{ for } \xi \in \mathcal{L}_0
\end{array}$$

The proof of Proposition 8 is analogous to the proof of Propositions 5 and 6 for (DP-1).

Proposition 9. *Let $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_0 \rightarrow Mod(\mathcal{L}_{\leq})$ be a dynamic operator on preference models. The operator $\star$ satisfies (DP-3) only if the axiom schemata in Fig. 1 and below are valid in $\mathcal{L}_{\leq}(\star)$.*

$$[\star\varphi]\langle<\rangle(\neg\varphi \wedge \xi) \rightarrow (\varphi \rightarrow \langle<\rangle(\neg\varphi \wedge [\star\varphi]\xi))$$

Proof. As before, let's only show it holds for the interesting case, since the others are trivial.

Let $M = \langle W, \leq, v\rangle$ be a preference model and $\star$ an operation on preference models that satisfy (DP-3). Take $w \in [\![\varphi]\!]$ s.t. $M, w \vDash [\star\varphi]\langle<\rangle(\neg\varphi \wedge \xi)$ for some ξ, then $M_{\star\varphi}, w \vDash \langle<\rangle(\neg\varphi \wedge \xi)$. From this, we conclude that there is a $w' \in W$ s.t. $w' <_{\star\varphi} w$ and $M_\varphi, w' \vDash \neg\varphi \wedge \xi$. As such, $M, w' \vDash [\star\varphi](\neg\varphi)$ and $M, w' \vDash [\star\varphi]\xi$. Since $\varphi \in \mathcal{L}_0$, then $M, w' \vDash \neg\varphi$ and, by (DP-3), $w' < w$. As such $M, w \vDash \varphi \rightarrow \langle<\rangle(\neg\varphi \wedge [\star\varphi](\neg\varphi))$. □

Similarly, for (DP-4), we have the following characterisation.

Proposition 10. *Let* $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_{\leq} \to Mod(\mathcal{L}_{\leq})$ *be a dynamic operator on preference models. The operator* $\star$ *satisfies (DP4) only if the axiom schemata in Fig. 1 and below are valid in* $\mathcal{L}_{\leq}(\star)$*.*

$$[\star\varphi]\langle\leq\rangle(\neg\varphi \wedge \xi) \to (\varphi \to \langle\leq\rangle(\neg\varphi \wedge [\star\varphi]\xi))$$

The proof is analogous to the case for (DP-3). Not only DP postulates can be characterised in Dynamic Preference Logic, but also Recalcitrance (Rec) and Independence (Ind).

Proposition 11. *Let* $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_{\leq} \to Mod(\mathcal{L}_{\leq})$ *be a dynamic operator on preference models. The operator* $\star$ *satisfies (Rec) only if the axiom schemata in Fig. 1 and below are valid in* $\mathcal{L}_{\leq}(\star)$*.*

$$[\star\varphi][\leq]\xi \to (\neg\varphi \to A(\varphi \to [\star\varphi]\xi))$$
$$[\star\varphi][<]\xi \to (\neg\varphi \to A(\varphi \to [\star\varphi]\xi))$$

Proof. Let's show it holds for the interesting case.

Let $M = \langle W, \leq, v\rangle$. Take $w' \notin [\![\varphi]\!]$ with $M, w \vDash [\star\varphi][\leq]\xi$ for some ξ. Then for any $w'' \in W$, if $w'' \leq_{\star\varphi} w'$ then $M_{\star\varphi}, w'' \vDash \xi$. Take $w \notin [\![\varphi]\!]$, then by (Rec), $w <_{\star\varphi} w'$. As such $M_{\star\varphi}, w \vDash \xi$, i.e. $M, w \vDash [\star\varphi]\xi$. Since it holds for any world $w \notin [\![\varphi]\!]$, then $M, w' \vDash A(\varphi \to [\star\varphi]\xi)$. Since $w' \notin [\![\varphi]\!]$, we conclude that $M, w' \vDash \neg\varphi \to A(\varphi \to [\star\varphi]\xi)$. □

For Independence, we have the following.

Proposition 12. *Let* $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_{\leq} \to Mod(\mathcal{L}_{\leq})$ *be a dynamic operator on preference models. The operator* $\star$ *satisfies (Ind) only if the axiom schemata in Fig. 1 and below are valid in* $\mathcal{L}_{\leq}(\star)$*.*

$$[\star\varphi][<](\varphi \to \xi) \to (\neg\varphi \to [\leq](\varphi \wedge [\star\varphi]\xi))$$

Proof. Let $M = \langle W, \leq, v\rangle$ be a preference model. Take $w \notin [\![\varphi]\!]$ s.t. it holds that $M, w \vDash [\star\varphi][\leq](\varphi \to \xi)$ for some φ and ξ. For any $w' \in [\![\varphi]\!]$, if $w' <_{\star\varphi} w$, then $M_{\star\varphi}, w' \vDash \xi$ and $M, w' \vDash [\star\varphi]\xi$. Take $w' \in [\![\varphi]\!]$, s.t. $w' \leq w$. By (Ind), $w' <_{\star\varphi} w$, thus $M, w' \vDash \varphi \to [\star\varphi]\xi$. As such, $M, w \vDash \neg\varphi \to [\leq](\varphi \to [\star\varphi]\xi)$. □

Regarding the postulates for iterated belief contraction, similar characterisations in DPL can be obtained. For space constraints, we will omit the proof of the following results, but we point out that the proofs are straight forward.

As the concept of 'most preferred worlds satisfying φ' will be necessary to provide the axiomatizations for contraction, we define the formula $\mu\varphi \equiv \varphi \wedge \neg\langle<\rangle\varphi$ encompassing this exact concept. With that we provide the following result.

Proposition 13. *Let $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_{\leq} \rightarrow Mod(\mathcal{L}_{\leq})$ be a dynamic operator on preference models. The operator $\star$ satisfies (GR) only if the axiom schemata in Fig. 1 and below are valid in $\mathcal{L}_{\leq}(\star)$.*

$$\begin{array}{ll}
[\star\varphi][\leq]\xi & \rightarrow A(\mu\top \rightarrow [\star\varphi]\xi) \\
[\star\varphi][\leq]\xi & \rightarrow A(\mu\neg\varphi \rightarrow [\star\varphi]\xi) \\
[\star\varphi][<]\xi & \rightarrow (\neg\mu\neg\varphi \wedge \neg\mu\top) \rightarrow [<]([\star\varphi]\xi) \\
[\star\varphi][<]\xi & \rightarrow (\neg\mu\neg\varphi \wedge \neg\mu\top) \rightarrow A(\mu\neg\varphi \rightarrow [<][\star\varphi]\xi) \\
(\mu\neg\varphi \vee \mu\top) \rightarrow [\star][<]\xi, \ \textit{for any } \xi &
\end{array}$$

For natural contraction, we have the following characterisation.

Proposition 14. *Let $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_{\leq} \rightarrow Mod(\mathcal{L}_{\leq})$ be a dynamic operator on preference models. The operator $\star$ satisfies (NC) only if the axiom schemata in Fig. 1 and below are valid in $\mathcal{L}_{\leq}(\star)$.*

$$\begin{array}{l}
[\star\varphi][\leq]\xi \rightarrow (\neg\mu\neg\varphi) \rightarrow [\leq]([\star\varphi]\psi) \\
[\star\varphi][<]\xi \rightarrow (\neg\mu\neg\varphi) \rightarrow [<]([\star\varphi]\psi)
\end{array}$$

Finally, we can characterise moderate contraction by the following axioms.

Proposition 15. *Let $\star : Mod(\mathcal{L}_{\leq}) \times \mathcal{L}_{\leq} \rightarrow Mod(\mathcal{L}_{\leq})$ be a dynamic operator on preference models. The operator $\star$ satisfies (MC) only if the axiom schemata in Fig. 1 and below are valid in $\mathcal{L}_{\leq}(\star)$.*

$$\begin{array}{l}
[\star\varphi][\leq]\xi \rightarrow ((\varphi \wedge \neg\mu\top) \rightarrow A(\neg\varphi \rightarrow [\star\varphi]\xi)) \\
[\star\varphi][<]\xi \rightarrow ((\varphi \wedge \neg\mu\top) \rightarrow A(\neg\varphi \rightarrow [\star\varphi]\xi))
\end{array}$$

In a sense, the results above generalize the study of iterated belief change by exposing some fundamental properties of the models adopted in the area. Namely, the postulates of iterated belief revision are intrinsically linked to the structure of the model in which they are based - i.e. Grove's system of spheres - in which each world can be distinguished from the others.

4.1 Deriving Axiomatisations for DPL by Iterated Belief Change Postulates

In this section, we investigate how we can derive correct axiomatisations for the extended logic $\mathcal{L}_{\leq}(\star)$, given the postulates satisfied by $\star$. In the results below, we use the axiomatisation $L_{\leq}$ for Preference Logic with well-founded models, provided by Souza [22].

Theorem 16. *Let $\mathcal{C}$ be a class of dynamic operators and let Γ be a set of axioms s.t. if for $\star \in \mathcal{C}$, then Γ is valid in the logic of $\mathcal{L}_{\leq}(\star)$. If $L_{\leq} \cup \Gamma \vdash [\star\varphi]\xi$ then for any model $M = \langle W, \leq, v\rangle \in Mod(\mathcal{L}_{\leq})$, $w \in W$ and all $\star \in \mathcal{C}$ then $M_{\star\varphi}, w \vDash \xi$.*

Proof. Since we have that the formulas of Γ are valid in the logic of $\mathcal{L}_{\leq}(\star)$, for any $\star \in \mathcal{C}$. By monotony of the logic, for any propositional formula $\varphi \in \mathcal{L}_0$, if $L_{\leq} \cup \bigcup \Gamma \vdash [\star\varphi]\xi$, then for any model $M \in Mod(\mathcal{L}_{\leq})$, $w \in W$ it holds that $M_{\star\varphi}, w \vDash \xi$. $\square$

Theorem 16 says that we can take the union of the axiom schematas provided in Sect. 4 for each postulate and the resulting axiomatization is sound in regard to the class of all dynamic operators satisfying all these postulates. It is not clear whether we can obtain a completeness result for the general case.

Let's apply this result to obtain axiomatizations for the well-known operation of Radical Upgrade [9,24], which corresponds to Nayak et al. simple lexicographic revision [17].

Definition 17. *Let $M = \langle W, \leq, v\rangle$ be a preference model and $\varphi \in \mathcal{L}_0$ a propositional formula. The radical upgrade of φ in M is the model $M_{\Uparrow\varphi} = \langle W, \leq_{\Uparrow\varphi}, v\rangle$ where $\leq_{\Uparrow\varphi} = (\leq \setminus [\![\neg\varphi]\!] \times [\![\varphi]\!]) \cup [\![\varphi]\!] \times [\![\neg\varphi]\!]$.*

As proved by Nayak et al. [17], this operation is completely characterized by the postulates (DP-1), (DP-2) and (Rec). From this, we obtain an axiomatization for the logic extended with Radical Upgrade. In the following result, notice that the proposed axiomatisation is a simplification of the union of the axioms proposed in Propositions 5 and 8, and they are in fact equivalent.

Proposition 18. *Preference Logic extended with Radical Upgrade $\mathcal{L}_{\leq}(\Uparrow)$ [9] is correctly axiomatised by $L_{\leq}$ extended with the following axioms and the modus ponens and necessitation rules for all modalities.*

$$
\begin{array}{ll}
[\Uparrow \varphi]p & \leftrightarrow p \\
[\Uparrow \varphi]\neg\psi & \leftrightarrow \neg[\Uparrow \varphi]\psi \\
[\Uparrow \varphi](\psi \wedge \xi) & \leftrightarrow [\Uparrow \varphi]\psi \wedge [\Uparrow \varphi]\xi \\
[\Uparrow \varphi]A\psi & \leftrightarrow A([\Uparrow \varphi]\psi) \\
[\Uparrow \varphi][\leq]\psi & \leftrightarrow \varphi \rightarrow [\leq](\varphi \rightarrow [\Uparrow \varphi]\psi) \wedge \neg\varphi \rightarrow A(\varphi \rightarrow [\Uparrow \varphi]\psi) \wedge \\
& \quad \neg\varphi \rightarrow [\leq](\neg\varphi \rightarrow [\Uparrow \varphi]\psi)) \\
[\Uparrow \varphi][<]\psi & \leftrightarrow \varphi \rightarrow [<](\varphi \rightarrow [\Uparrow \varphi]\psi) \wedge \neg\varphi \rightarrow A(\varphi \rightarrow [\Uparrow \varphi]\psi) \wedge \\
& \quad \neg\varphi \rightarrow [<](\neg\varphi \rightarrow [\Uparrow \varphi]\psi))
\end{array}
$$

This axiomatisation is, in fact, complete as proved by Girard [9].

5 Related Work

To our knowledge, the work of Segerberg [21] is the first to propose the integration of belief revision operations within an epistemic logic, with his proposal of Dynamic Doxastic Logic (DDL). That integration is important because it allows one to analyse the effects of introspection, and other related phenomena, in the logic of belief change. A famous example of such interaction is the analysis of Moore sentences in the logic of belief change, which shows that AGM's postulates are incompatible in the face of introspection [13]. In this work, Segerberg provides a set axioms which corresponds to encodings within his logic of AGM's postualtes for belief change. In a sense, our work is linked to that approach by investigating these correspondences for dynamic belief change, based on iterated belief change postulates.

In the conext of DDL, Cantwell [6] define some iterated belief revision operators as change operations in hypertheories [13] and show hows these operations can be axiomatically characterized in DDL. Our work differs from his by the fact that we analyse how some well known-postulates can be characterized in our logic and not how to encode specific constructions. Our logic has also the advantage to being more expressive, since it can encode some notion of degrees of belief [22], which cannot be expressed in DDL.

Inspired by Rott's [20], Van Benthem [24] proposed the codification of some iterated belief revision operators within a Dynamic Epistemic Logic (DEL). This work was further extended by Girard [9] and Liu [14] and Souza et al. [23] that studied the use of DPL to encode several (relational) belief revision policies.

Baltag and Smets [4], on the other hand, used a logic similar to DPL to encode different notions for knowledge and belief. These authors show how different iterated belief revision operators can be simulated using DEL action models and product update. Later, in [3], they propose a variation of Segerberg's Full DDL and prove that it is possible to model DEL-style epistemic actions within DDL.

Studying DPL, Van Benthem and Liu [28] showed how one can use Propositional Dynamic Logic to not only encode belief change operators, but how to derive complete axiomatisations for the logic extended with them using Propositional Dynamic Logic. Based on this approach, Girard and Rott [10] propose a Dynamic Preference Logic for studying belief revision. The authors encode several iterated belief revision policies using General Dynamic Dynamic Logic [8] and show that reduction axioms can be obtained for them, in the same fashion of [28].

In the related literature, all the works following the DEL tradition define operations semantically in their logic and either provide axiomatizations by means of crafting an axiomatization for the extended logic or by encoding these operations using a variation of dynamic logic to obtain reduction axioms. In a sense these works employ the well-known results in the area of belief change to choose appropriate operations and then encode these operations in their logics.

From the work following this tradition, perhaps, the one that come most close to ours, in questioning how to characterize classes of operations by means of logic is that of van Benthem [25]. Van Benthem shows that any belief changing operator that satisfies the axiomatization provided must coincide with radical upgrade. As such, the author investigates how to use a dynamic logic to study dynamic operators, instead of just applying the results in belief change in this logic.

Our work, on the other hand, investigates how Dynamic Preference Logic can be used to characterise properties of dynamic belief change operators. To our knowledge, our work is the first to do so for dynamic belief change operators.

6 Final Considerations

In this work we investigated the connection between the well-known characterisation of dynamic operators as studied in the area of Iterated Belief Change and the axiomatisation of Dynamic Preference Logic with these operators. We provide a method for computing correct axiomatisations of the logic $\mathcal{L}(\star)$ given a (complete) characterisation of the operation $\star$ by means of postulates in the area of Iterated Belief Change.

Notice that our method to derive axiomatizations is more general than the use of Propositional Dynamic Logic programs to define dynamic operators, proposed by van Benthem and Liu [28] or, equivalently, Aucher's reductions [2], since there is no expressibility restriction in the dynamic operators which can be investigated by our approach. This is the case, for example, of Nayak et al. [16] lexicographic contraction which can only be expressed as a Propositional Dynamic Logic program if we restrict the analysis to models with limited longest chains of worlds [22]. We can apply our approach to show an infinite set of axioms that define the operation in DPL - taking the axioms of Souza [22] for models of each size.

On the other hand, different than the reduction axioms obtained using Propositional Dynamic Logic, the axiomatizations derived using iterated belief change postulates are not complete, in general. We do point out that, for the special case in which we consider only those classes of models with characterizing formulas, we can obtain completeness proofs trivially for the dynamic operators satisfying a subset of the postulates investigated in this work[6]. This is the case for some important examples, such as Grove's models for belief change in classical propositional logic or Souza's broad models for dynamic preference logic.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: partial meet contraction and revision functions. J. Symb. Logic **50**(2), 510–530 (1985)
2. Aucher, G.: Characterizing updates in dynamic epistemic logic. In: Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (2010)
3. Baltag, A., Fiutek, V., Smets, S.: DDL as an "internalization" of dynamic belief revision. In: Trypuz, R. (ed.) Krister Segerberg on Logic of Actions. Outstanding Contributions to Logic, vol. 1, pp. 253–280. Springer, Dordrecht (2014). https://doi.org/10.1007/978-94-007-7046-1_12
4. Baltag, A., Smets, S.: A qualitative theory of dynamic interactive belief revision. In: Texts in Logic and Games, vol. 3, pp. 9–58 (2008)
5. Blackburn, P., van Benthem, J.F., Wolter, F.: Handbook of Modal Logic, vol. 3. Elsevier, Amsterdam (2006)
6. Cantwell, J.: Some logics of iterated belief change. Stud. Logica. **63**(1), 49–84 (1999)
7. Darwiche, A., Pearl, J.: On the logic of iterated belief revision. Artif. Intell. **89**(1), 1–29 (1997)

[6] Notice that, as commented before, Proposition 6 can be replicated for any postulate investigated in this work. As such, given a class of models $\mathcal{M}$ satisfying the characterizing formula restriction, the class $\mathcal{C}$ in Theorem 16 can be characterized as the intersection of all classes C_i, where C_i is the class of all dynamic operators satisfying each axiom P_i.

8. Girard, P., Seligman, J., Liu, F.: General dynamic dynamic logic. In: Bolander, T., Brauner, T., Ghilardi, S., Moss, L. (eds.) Advances in Modal Logic, vol. 9, pp. 239–260. College Publications, London (2012)
9. Girard, P.: Modal logic for belief and preference change. Ph.D. thesis, Stanford University (2008)
10. Girard, P., Rott, H.: Belief revision and dynamic logic. In: Baltag, A., Smets, S. (eds.) Johan van Benthem on Logic and Information Dynamics. OCL, vol. 5, pp. 203–233. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06025-5_8
11. Grove, A.: Two modelings for theory change. J. Philos. Logic **17**(2), 157–170 (1988)
12. Jin, Y., Thielscher, M.: Iterated belief revision, revised. Artif. Intell. **171**(1), 1–18 (2007)
13. Lindström, S., Rabinowicz, W.: DDL unlimited: dynamic doxastic logic for introspective agents. Erkenntnis **50**(2), 353–385 (1999)
14. Liu, F.: Reasoning About Preference Dynamics, vol. 354. Springer, Dordrecht (2011). https://doi.org/10.1007/978-94-007-1344-4
15. Moss, L.S.: Finite models constructed from canonical formulas. J. Philos. Logic **36**(6), 605–640 (2007)
16. Nayak, A., Goebel, R., Orgun, M., Pham, T.: Taking Levi Identity seriously: a plea for iterated belief contraction. In: Lang, J., Lin, F., Wang, J. (eds.) KSEM 2006. LNCS (LNAI), vol. 4092, pp. 305–317. Springer, Heidelberg (2006). https://doi.org/10.1007/11811220_26
17. Nayak, A.C., Pagnucco, M., Peppas, P.: Dynamic belief revision operators. Artif. Intell. **146**(2), 193–228 (2003)
18. Ramachandran, R., Nayak, A.C., Orgun, M.A.: Three approaches to iterated belief contraction. J. Philos. Logic **41**(1), 115–142 (2012)
19. Rott, H.: Two dogmas of belief revision. J. Philos. **97**(9), 503–522 (2000)
20. Rott, H.: Shifting priorities: simple representations for twenty-seven iterated theory change operators. In: Makinson, D., Malinowski, J., Wansing, H. (eds.) Towards Mathematical Philosophy. Trends in Logic, vol. 28, pp. 269–296. Springer, Dordrecht (2009). https://doi.org/10.1007/978-1-4020-9084-4_14
21. Segerberg, K.: Two traditions in the logic of belief: bringing them together. In: Ohlbach, H.J., Reyle, U. (eds.) Logic, Language and Reasoning. Trends in Logic (Studia Logica Library), vol. 5, pp. 135–147. Springer, Dordrecht (1999). https://doi.org/10.1007/978-94-011-4574-9_8
22. Souza, M.: Choices that make you change your mind: a dynamic epistemic logic approach to the semantics of BDI agent programming languages. Ph.D. thesis, Universidade Federal do Rio Grande do Sul (2016)
23. Souza, M., Moreira, A., Vieira, R., Meyer, J.J.C.: Preference and priorities: a study based on contraction. In: KR 2016, pp. 155–164. AAAI Press (2016)
24. Van Benthem, J.: Dynamic logic for belief revision. J. Appl. Non-Classical Logics **17**(2), 129–155 (2007)
25. Van Benthem, J.: Two logical faces of belief revision. In: Trypuz, R. (ed.) Krister Segerberg on Logic of Actions. Outstanding Contributions to Logic, vol. 1, pp. 281–300. Springer, Dordrecht (2014). https://doi.org/10.1007/978-94-007-7046-1_13
26. Van Benthem, J., Girard, P., Roy, O.: Everything else being equal: a modal logic for ceteris paribus preferences. J. Philos. Logic **38**(1), 83–125 (2009)
27. Van Benthem, J., Grossi, D., Liu, F.: Priority structures in deontic logic. Theoria **80**(2), 116–152 (2014)
28. Van Benthem, J., Liu, F.: Dynamic logic of preference upgrade. J. Appl. Non-Classical Logics **17**(2), 157–182 (2007)

Author Index