Sujata Ghosh
Sanjiva Prasad (Eds.)

LNCS 10119

Logic and Its Applications

7th Indian Conference, ICLA 2017
Kanpur, India, January 5–7, 2017
Proceedings

Springer

Lecture Notes in Computer Science **10119**

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

FoLLI Publications on Logic, Language and Information

Subline of Lectures Notes in Computer Science

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

Sujata Ghosh · Sanjiva Prasad (Eds.)

Logic and Its Applications

7th Indian Conference, ICLA 2017
Kanpur, India, January 5–7, 2017
Proceedings

Springer

Editors
Sujata Ghosh
Indian Statistical Institute
Chennai, Tamil Nadu
India

Sanjiva Prasad
Indian Institute of Technology Delhi
New Delhi
India

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-662-54068-8 ISBN 978-3-662-54069-5 (eBook)
DOI 10.1007/978-3-662-54069-5

Library of Congress Control Number: 2016959632

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer-Verlag GmbH Germany
The registered company address is: Heidelberger Platz 3, 14197 Berlin, Germany

Preface

The seventh edition of the Indian Conference on Logic and Its Applications (ICLA 2017) was held during January 5–7, 2017 at IIT Kanpur. Co-located with the conference was the ninth edition of the Methods for Modalities Workshop (M4M-9), held during January 8–9, 2017. This volume contains the papers that were accepted for publication and presentation at ICLA 2017.

The ICLA is a biennial conference organized under the aegis of ALI, the Association for Logic in India. The aim of this conference series is to bring together researchers from a wide variety of fields in which formal logic plays a significant role. Areas of interest include mathematical and philosophical logic, computer science logic, foundations and philosophy of mathematics and the sciences, use of formal logic in areas of theoretical computer science and artificial intelligence, logic and linguistics, and the relationship between logic and other branches of knowledge. Of special interest are studies in systems of logic in the Indian tradition, and historical research on logic.

We received 34 submissions this year. Each submission was reviewed by at least three Program Committee members, and by external experts in some cases. We thank all those who submitted papers to ICLA 2017. After going through the detailed reviews and having extensive discussions on each paper, the Program Committee decided to accept 13 papers for publication and presentation. These contributions range over a varied set of themes including proof theory, model theory, automata theory, modal logics, algebraic logics, and Indian systems. In addition, the authors of some other submissions were invited to participate in the conference and to present their ideas for discussion. We would like to extend our gratitude to the Program Committee members for their hard work, patience, and knowledge in putting together an excellent technical program. We also extend our thanks to the external reviewers for their efforts in providing expert opinions and valuable feedback to the authors.

The program also included four invited talks. We are grateful to Nicholas Asher, Natasha Dobrinen, Luke Ong, and Richard Zach for accepting our invitation to speak at ICLA 2017 and for contributing to this proceedings volume.

We would like to express our appreciation of the Department of Mathematics and the Department of Computer Science and Engineering at IIT Kanpur for hosting the conference. Special thanks are due to Anil Seth, Mohua Banerjee, Sunil Simon, and other members of the Organizing Committee for their commitment and effort, and their excellent arrangements in the smooth running of the conference. We also express our appreciation of the tireless efforts of all the volunteers who contributed to making the conference a success.

The putting together of the technical program was immensely facilitated by the EasyChair conference management software, which we used from managing the submissions to producing these proceedings.

We would like to thank the Association for Symbolic Logic for supporting the conference. Finally, we are grateful to the Editorial Board at Springer for publishing this volume in the LNCS series.

November 2016

Sujata Ghosh
Sanjiva Prasad

Organization

Program Committee

Natasha Alechina	University of Nottingham, UK
Maria Aloni	University of Amsterdam, The Netherlands
Steve Awodey	Carnegie Mellon University, Pittsburgh, USA
Mohua Banerjee	Indian Institute of Technology Kanpur, India
Patricia Blanchette	University of Notre Dame, USA
Maria Paola Bonacina	Università degli Studi di Verona, Italy
Lopamudra Choudhury	Jadavpur University, India
Agata Ciabattoni	Technische Universität Wien, Austria
Anuj Dawar	University of Cambridge, UK
Hans van Ditmarsch	LORIA, Nancy, France
Sujata Ghosh	Indian Statistical Institute Chennai, India
Brendan Gillon	McGill University, Montreal, Canada
Roman Kossak	City University of New York, USA
S. Krishna	Indian Institute of Technology Bombay, India
Benedikt Löwe	Universiteit van Amsterdam, The Netherlands and Universität Hamburg, Germany
Gopalan Nadathur	University of Minnesota, USA
Satyadev Nandakumar	Indian Institute of Technology Kanpur, India
Alessandra Palmigiano	Technische Universiteit Delft, The Netherlands
Prakash Panangaden	McGill University, Montreal, Canada
Sanjiva Prasad	Indian Institute of Technology Delhi, India
R. Ramanujam	Institute of Mathematical Sciences, Chennai, India
Christian Retoré	LIRMM University of Montpellier, France
Sunil Simon	Indian Institute of Technology Kanpur, India
Isidora Stojanovic	Jean Nicod Institute, Paris, France
S.P. Suresh	Chennai Mathematical Institute, India
Rineke Verbrugge	University of Groningen, The Netherlands
Yanjing Wang	Peking University, China

Additional Reviewers

Bagchi, Amitabha
Bienvenu, Meghyn
Bilkova, Marta
Fisseni, Bernhard
Freschi, Elisa
Greco, Giuseppe
Gupta, Gopal
Henk, Paula
Ju, Fengkui
Karmakar, Samir
Kurur, Piyush
Kuznets, Roman
Lapenta, Serafina
Lodaya, Kamal
Mukhopadhyay, Partha
Majer, Ondrej
Narayan Kumar, K.
Paris, Jeff
Rafiee Rad, Soroush
Sadrzadeh, Mehrnoosh
Sreejith, A.V.
Turaga, Prathamesh
Velázquez-Quesada, Fernando R.
Woltzenlogel Paleo, Bruno
Zanuttini, Bruno

Contents

Conversation and Games

Nicholas Asher(✉) and Soumya Paul

Institut de Recherche en Informatique de Toulouse, Toulouse, France
nicholas.asher@irit.fr, soumya.paul@gmail.com

Abstract. In this paper we summarize concepts from earlier work and demonstrate how infinite sequential games can be used to model strategic conversations. Such a model allows one to reason about the structure and complexity of various kinds of winning goals that conversationalists might have. We show how to use tools from topology, set-theory and logic to express such goals. We then show how to tie down the notion of a winning condition to specific discourse moves using techniques from Mean Payoff games and discounting. We argue, however, that this still requires another addition from epistemic game theory to define appropriate solution and rationality underlying a conversation.

Keywords: Strategic reasoning · Conversations · Dialogues · Infinite games · Epistemic game theory

1 Introduction

Conversations have a natural analysis as games. They involve typically at least two agents, each with their own interests and goals. These goals may be compatible, or they may conflict; but in either case, one agents' successfully achieving her conversational goals will typically depend upon her taking her interlocutor's goals and interests into account. In cooperative conversations where agents' goals are completely aligned, conversational partners may still need to coordinate actions, even linguistic actions. A strategic or non-cooperative conversation involves (at least) two people (agents) who have opposing interests concerning the outcome of the conversation. A debate between two political candidates is an instance. Each candidate has a certain number of points to convey to the audience, and each wants to promote her own position and damage her opponent's or opponents'. To achieve these goals, each participant typically needs to plan for anticipated responses from the other.

This paper surveys some results from what we feel is an exciting new application of games to language. The core of formal results are summarized from [4,6]; the part on weighted and discounted games draws from [3] but also introduces new material; the last section points to work in progress.

Various game-theoretic models for cooperative conversation have been proposed, most notably the model of signalling games [22]. Another closely related

The authors thank ERC grant 269427 for supporting this research.

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 1–18, 2017.
DOI: 10.1007/978-3-662-54069-5_1

model is that persuasion games [15]. In a signalling game one player with a knowledge of the actual state sends a signal and the other player who has no knowledge of the state chooses an action, usually upon an interpretation of the received signal. The standard setup supposes that both players have common knowledge of each other's preference profiles as well as their own over a set of commonly known set of possible states, actions and signals. However for modeling non-cooperative strategic contexts of sequential dynamic games, signalling games suffer from many drawbacks. We summarise below the difficulties we see (see [6] for a more comprehensive discussion):

- A game that models a non-cooperative setting, that is a setting where the preferences of the players are opposed, must be zero-sum. However, it has been shown [11] that in a zero-sum criterion, in equilibrium, the sending and receiving of any message has no effect on the receiver's decision. Signaling games typically assign a game a finite horizon; backward inductions arguments threaten to conclude that communication should not occur in such situations.
- In order to use games as part of a general theory of meaning, one has to make clear how to construct the game-context, which includes providing an interpretation of the game's ingredients (types, messages, actions). Franke's extension of signalling, games, *interpretation games*, addresses this issue [13]. Such games encode a 'canonical context' for an utterance, in which relevant conversational implicatures may be drawn. The game structure is determined by the set of 'sender types'. Interpretation games model the interpretation of the messages and actions of a signaling game in a co-operative context for 'Gricean agents' quite well. But in the non-cooperative setting, things get very intricate and problems remain.
- Signalling games are one-shot and fail to capture the dynamic nature of a strategic conversation. One can attempt to encode a finite sequence of moves of a particular player as a single message m sent by that player but then one runs into the problem of assigning correct utilities for m because such utilities depend again on the possible set of continuations of m.
- Finally, there is an inherent asymmetry associated with the setting of a signalling game - one player is informed of the state of the world but the other is not; one player sends a message but the other does not. Conversations (like debates), on the other hand, are symmetric - all participants should (and usually do) get equal opportunities to get their messages across.

Strategic conversations are thus special and have characteristics unique to them which, to our knowledge, have not been captured in other frameworks. Here is a short list of these characteristics:

- Conversations are sequential and dynamic and inherently involve a 'turn-structure' which is important in determining the merit of a conversation to the participants. In other words, it is important to keep track of "who said what".
- A 'move' by a player in a linguistic game typically carries more semantic content than usually assumed in game theory. What a player says may have a

set of 'implicatures', may be 'ambiguous', may be 'coherent/incoherent' or 'consistent/inconsistent' with regards to what she had said earlier in the conversation. She may also 'acknowledge' other people's contributions or 'retract' her previous assertions. These features too have important consequences on the existence and complexity of winning strategies.

– Conversations typically have a 'Jury' who evaluates the conversation and determines if one or more of the players have reached their goals. In other words a Jury determines the winners in a conversation, if there is a winner. Players will spin the description of the game to their advantage and so may not present an accurate view of what happened. The Jury can be a concrete or even a hypothetical entity who acts as a 'passive player' in the game. For example, in a courtroom situation there is a physical Jury who gives the verdict, whereas in a political debate the Jury is the audience or the citizenry in general. This means that the winning conditions of the players are affected by the Jury in that, they depend on what they believe that the Jury expects them to achieve.
– Conversations do not have a 'set end'. When two or more people engage in a conversation they do not know at the outset how many turns it will last or how many chances each player will get to speak (if at all). In a more scripted conversation like a political debate or a courtroom debate, there may be a moderator whose job is to ensure that each player receives his or her fair chance to put their points across; but even such a moderator may not know at the outset how the conversation will unfold and how many turns each player will receive. Players thus cannot strategize for a set horizon while starting a conversation. This rules out backward induction reasoning for both the players and analysts of conversation.
– Finally, epistemic elements are a natural component of such games. The players and the Jury have 'types', and players have 'beliefs' about the types of the other players and the Jury. They strategize based on their beliefs and also update their beliefs after each turn.

The first four considerations led [6] to model conversations as infinite games over a countable 'vocabulary' V. They call such games *Message Exchange games* (ME games). The intuitive idea behind an ME game is that a conversation proceeds in turns where in each turn one of the players 'speaks' or plays a string of letters from her own "vocabulary". The two vocabularies are distinguished in order to keep track of who said what, which is crucial to the analysis of a conversation. We will assume that both players use the same expressions in a set V to communicate, but that when 0 uses a symbol $v \in V$, she is actually playing $(v, 0)$, which allows us to see that it was 0 that played v at a certain point in the sequence; and when 1 plays v, he's actually playing $(v, 1)$.

However, a conversationalist does not play just any sequence of arbitrary strings but sentences or sets of sentences that 'make sense'. To ensure this, the vocabulary V should have a built-in, exogenous semantics. [6] identify V with the language of a semantic theory for discourse, SDRT [1]. SDRT's language characterizes the semantics and pragmatics of moves in dialogue. This means

that we can exploit the notion of entailment associated with the language of SDRSs to track commitments of each player in an ME game. In particular, the language of SDRT features variables for dialogue moves that are characterized by contents that the move commits its speaker to. Crucially, some of this content involves predicates that denote rhetorical relations between moves—like the relation of *question answer pair* (qap), in which one move answers a prior move characterized by a question. The vocabulary V of an ME game thus contains a countable set of discourse constituent labels $\mathsf{DU} = \{\pi, \pi_1, \pi_2, \ldots\}$, and a finite set of discourse relation symbols $\mathcal{R} = \{\mathsf{R}, \mathsf{R}_1, \ldots \mathsf{R}_n\}$, and formulas $\phi, \phi_1, ...$ from some fixed language for describing elementary discourse move contents. V consists of formulas of the form $\pi : \phi$, where ϕ is a description of the content of the discourse unit labelled by π in a logical language like the language of higher order logic used, e.g., in Montague Grammar, and $\mathsf{R}(\pi, \pi_1)$, which says that π_1 stands in relation R to π. One such relation R is qap. Thus, each discourse relation represented in V comes with constraints as to when it can be coherently used in a context and when it cannot.

2 Message Exchange Games

We now formally define Message Exchange games, state some of their properties and show how they model strategic conversations, as explored in [6]. For simplicity, we restrict our description to conversations with two participants, whom we denote by Player 0 and Player 1. It is straightforward to generalize ME games to the case where there are more than two players. Thus, in what follows, we let i range over the set of players $\{0, 1\}$. Furthermore, Player $-i$ will always denote Player $(1 - i)$, the opponent of Player i.

We first define the notion of a 'Jury'. As noted in Sect. 1, a Jury is an entity or a group of entities that evaluates a conversation and decides the winner. A Jury thus 'groups' instances of conversations as being winning for Player 0 or Player 1 or both.

For any set A let A^* be the set of all finite sequences over A and let A^ω be the set of all countably infinite sequences over A. Let $A^\infty = A^* \cup A^\omega$ and $A^+ = A^* \backslash \{\epsilon\}$. Now, let V be a vocabulary as defined at the end of Sect. 1 and let $V_i = V \times \{i\}$.

Definition 1. *A* Jury $\mathcal{J}$ *over* $(V_0 \cup V_1)^\omega$ *is a tuple* $\mathcal{J} = (\mathsf{win}_1, \mathsf{win}_2)$ *where* $\mathsf{win}_i \subseteq (V_0 \cup V_1)^\omega$ *is the* winning condition *or* winning set *for Player* i.

Given the definition of a Jury over $(V_0 \cup V_1)^\omega$ we define a Message Excahge game game as:

Definition 2. *A Message Exchange game (ME game)* $\mathcal{G}$ *over* $(V_0 \cup V_1)^\omega$ *is a tuple* $\mathcal{G} = ((V_0 \cup V_1)^\omega, \mathcal{J})$ *where* $\mathcal{J}$ *is a Jury over* $(V_0 \cup V_1)^\omega$.

Formally an ME game $\mathcal{G}$ is played as follows. Player 0 starts the game by playing a non-empty sequence in V_0^+. The turn then moves to Player 1 who plays

a non-empty sequence from V_1^+. The turn then goes back to Player 0 and so on. The game generates a play ρ_n after n (≥ 0) turns, where by convention, $\rho_0 = \epsilon$ (the empty move). A play can potentially go on forever generating an infinite play ρ_ω, or more simply ρ. Player i wins the play ρ iff $\rho \in \mathsf{win}_i$. $\mathcal{G}$ is zero-sum if $\mathsf{win}_i = (V_0 \cup V_1)^\omega \backslash \mathsf{win}_{-i}$ and is non zero-sum otherwise. Note that both player or neither player might win a non zero-sum ME game $\mathcal{G}$. The Jury of a zero-sum ME game can be denoted simply as win where by convention $\mathsf{win} = \mathsf{win}_0$ and $\mathsf{win}_1 = (V_0 \cup V_1)^\omega \backslash \mathsf{win}$.

The basic structure of an ME game means that plays are segmented into *rounds*—a move by Player 0 followed by a move by Player 1. A finite play of an ME game is (also) called a history, and is denoted by ρ. Let Z be the set of all such histories, $Z \subseteq (V_0 \cup V_1)^*$, where $\epsilon \in Z$ is the empty history and where a history of the form $(V_0 \cup V_1)^+ V_0^+$ is a 0-history and one of the form $(V_0 \cup V_1)^+ V_1^+$ is a 1-history. We denote the set of i-histories by Z_i. Thus $Z = Z_0 \cup Z_1$. For $\rho \in Z$, $\mathsf{turns}(\rho)$ denotes the total number of turns (by either player) in ρ. A strategy σ_i of Player i is thus a function from the set of $-i$-histories to V_i^+. That is, $\sigma_i : Z_{-i} \to V_i^+$. A play $\rho = x_0 x_1 \ldots$ of an ME game $\mathcal{G}$ is said to conform to a strategy σ_i of Player i if for every prefix ρ_j of ρ, $j = i(\mod 2)$ implies $\rho_{j+1} = \rho_j \sigma_i(\rho_j)$. A strategy σ_i is called winning for Player i if $\rho \in \mathsf{win}_i$ for every play ρ that conforms to σ_i.

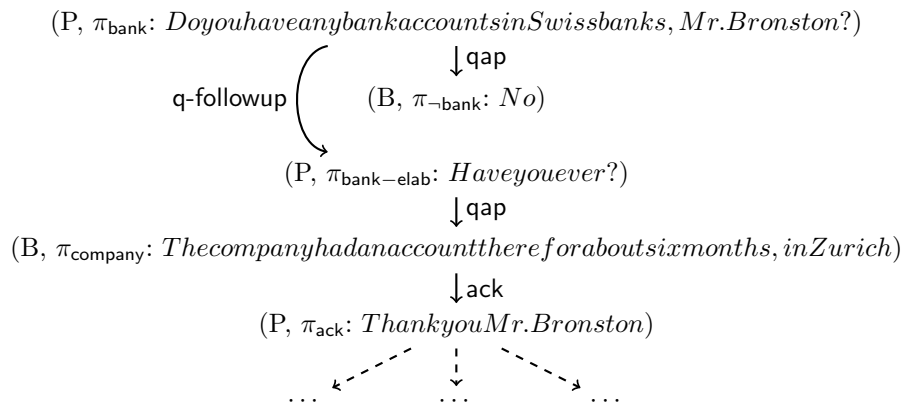
Given how we have characterized the vocabulary $(V_0 \cup V_1)$, we have a fixed meaning assignment function from EDUs to formulas describing their contents. Then, a sequence of conversational moves can be represented as a graph (DU, E, ℓ), where DU is the set of vertices each representing a discourse unit, $E \subseteq \mathsf{DU} \times \mathsf{DU}$ a set of edges representing links between discourse units that are labeled by $\ell : E \to \mathcal{R}$ with discourse relations.[1]

Example 1. To illustrate this structure of conversations, consider the following example taken from [2] from a courtroom proceedings where a prosecutor is querying the defendant. We shall return to this example later on for a strategic analysis.

a. **Prosecutor:** *Do you have any bank accounts in Swiss banks, Mr. Bronston?*
b. **Bronston:** *No, sir.*
c. **Prosecutor:** *Have you ever?*
d. **Bronston:** *The company had an account there for about six months, in Zurich.*
e. **Prosecutor:** *Thank you Mr. Bronston.*

Example 2. We can view the conversation in Eg. 1 as a play of an ME game as follows.

[1] We note that this is a simplification of SDRT which also countenances complex discourse units (CDUs) and another set of edges in the graph representation, linking CDUs to their simpler constituents. These edges represent parthood, not rhetorical relations. We will not, however, appeal to CDUs here.



The picture shows a weakly connected graph with a set of discourse constituent labels

$$\mathsf{DU} = \{\pi_{\mathsf{bank}}, \pi_{\neg\mathsf{bank}}, \pi_{\mathsf{bank-elab}}, \pi_{\mathsf{company}}, \pi_{\mathsf{ack}}, \ldots\}$$

and a set of relations

$$\mathcal{R} = \{\mathsf{qap}, \mathsf{q-followup}, \mathsf{ack}, \ldots\}$$

The arrows depict the individual relation instances between the DUs. A weakly connected graph represents a fully coherent conversation, in which each player's contribution is coherently linked with a preceding one. The graph also reveals that each player responds to a contribution of the other; this is a property that [6] call *responsiveness* (*vide infra*).

ME game messages come with a conventionally associated meaning in virtue of the constraints enforced by the Jury; an agent who asserts a content of a message commits to that content, and it is in virtue of such commitments that other agents respond in kind. While SDRT has a rich language for describing dialogue moves, earlier work did not make explicit how dialogue moves explicitly affect the commitments of the agents who make the moves or those who observe the moves. [24,25] link the semantics of the SDRT language with commitments explicitly (in two different ways). They augment the SDRT language with formulas that describe the commitments of dialogue participants, using a simple propositional modal syntax. Thus for any formula ϕ in the language of dynamic semantics that describes the content of a label $\pi \in \mathsf{DU}$, they add:

$$\neg\phi \mid \phi_1 \vee \phi_2 \mid \mathsf{C}_i\phi,\ i \in \{0,1\} \mid \mathsf{C}^*\phi$$

with the derived operators $\wedge$, $\Longrightarrow$, $\top, \bot$ are defined as usual, providing a propositional logic of commitments over the formulas that describe labels. Of particular interest are the commitment operators C_i and C^*. If ϕ is a formula for describing a content, $\mathsf{C}_i\phi$ is a formula that says that Player i commits to ϕ and $\mathsf{C}^*\phi$ denotes 'common commitment' of ϕ. Commitment is modelled as a Kripke modal

operator via an alternativeness relation in a pointed model with a distinguished (actual) world w_0. This allows them to provide a semantics for discourse moves that links the making of a discourse move by an agent to her commitments: i's assertion of a discourse move ϕ, for instance, we will assume, entails a common commitment that i commits to ϕ, written $\mathsf{C}^*\mathsf{C}_i\phi$. They show how each discourse move ϕ defines an action, a change or update on the model's commitment structure; in the style of public announcement logic viz. [8,9]. For instance, if agent i asserts ϕ, then the commitment structure for the conversational participants is updated such so as to reflect the fact that $\mathsf{C}^*\mathsf{C}_i\phi$. Finally, they define an entailment relation $\models$ that ensures that $\phi \models \mathsf{C}^*\mathsf{C}_i\phi$. This semantics is useful because it allows us to move from sequences of discourse moves to sequences of updates on any model for the discourse language. See [24,25] for a detailed development and discussion.

ME games resemble infinite games like Banach Mazur or Gale-Stewart games that have been used in topology, set theory [18] and computer science [16]. We can leverage some of the results from these areas to talk about the general 'shape' of conversations or to analyse the complexity of the winning conditions of the players in ME games. For instance, [23] shows that ME games, like Banach Mazur games or Gale-Stewart games, are determined. Other features have been extensively explored in [6]. We give a flavor of some of the applications here.

To do that we first need to define an appropriate topology on $(V_0 \cup V_1)^\omega$ which will allow us to characterize the descriptive complexity of the winning sets win_0 and win_1. We proceed as follows. We define the topology on $(V_0 \cup V_1)^\omega$ by defining the open sets to be sets of the form $A(V_0 \cup V_1)^\omega$ where $A \subseteq (V_0 \cup V_1)^*$. Such an open set will be often denoted as $\mathcal{O}(A)$. When A is a singleton set $\{x\}$ (say), we abuse notation and write $\mathcal{O}(\{x\})$ as $\mathcal{O}(x)$. The Borel sets are defined as the sigma-algebra generated by the open sets of this topology. The Borel sets can be arranged in a natural hierarchy called the Borel hierarchy which is defined as follows. Let Σ^0_1 be the set of all open sets. $\Pi^0_1 = \overline{\Sigma^0_1}$, the complement of the set of Σ^0_1 sets, is the set of all closed sets. Then for any $\alpha > 1$ where α is a successor ordinal, define Σ^0_α to be the countable union of all $\Pi^0_{\alpha-1}$ sets and define Π^0_α to be the complement of Σ^0_α. $\Delta^0_\alpha = \Sigma^0_\alpha \cap \Pi^0_\alpha$.

Definition 3 [18]. *A set A is called* complete *for a class Σ^0_α (resp. Π^0_α) if $A \in \Sigma^0_\alpha \backslash \Pi^0_\alpha$ (resp. $\Pi^0_\alpha \backslash \Sigma^0_\alpha$) and $A \notin (\Sigma^0_\beta \cup \Pi^0_\beta)$ for any $\beta < \alpha$.*

The Borel hierarchy represents the descriptive or structural complexity of the Borel sets. A set higher up in the hierarchy is structurally more complex than one that is lower down. Complete sets for a particular class of the hierarchy represent the structurally most complex sets of that class. We can use the Borel hierarchy and the notion of completeness to capture the complexity of winning conditions in conversations. For example, two typical sets in the fist level of the Borel hierarchy are defined as follows. Let $A \subseteq (V_0 \cup V_1)^+$, then

$$\mathsf{reach}(A) = \{\rho \in (V_0 \cup V_1)^\omega | \rho = xy\rho', y \in A\}$$

and

$$\mathsf{safe}(A) = (V_0 \cup V_1)^\omega \backslash \mathsf{reach}(A)$$

A little thought shows that $\mathsf{reach}(A) \in \Sigma_1^0$ and $\mathsf{safe}(A) \in \Pi_1^0$. Let reachability be the class of sets of the form $\mathsf{reach}(A)$ and safety be the class of sets of the form $\mathsf{safe}(A)$.

Example 3. Returning to our example of Bronston and the Prosecutor, let us consider what goals the Jury expects each of them to achieve. The Jury will award its verdict in favor of the Prosecutor: (i) if he can eventually get Bronston to admit that (a) he had an account in Swiss banks, or (b) he never had an account in Swiss banks, or (ii) if Bronston avoids answering the Prosecutor forever. In the case of (i)a, Bronston is incriminated, (i)b, he is charged with perjury and (ii), he is charged with contempt of court. Bronston's goal is the complement of the above, that is to avoid either of the situations (i)a, (i)b and (ii). We thus see that the Jury winning condition for the Prosecutor is a boolean combination of a reachability condition and the complement of a safety condition, which is in the first level of the Borel hierarchy.

Conversations typically must also satisfy certain natural constraints which the Jury might impose throughout the course of a play. Here are some constraints defined in [6]. We will then study the complexity of the sets satisfying them.

Let $\rho = x_0x_1x_2\ldots$ be a play of an ME game $\mathcal{G}$ where $x_0 = \epsilon$ and $x_j \in V^+_{((j-1) \bmod 2)}$ is the sequence played by Player $((j-1) \bmod 2)$ in turn j. For every i define the function $\mathsf{du}_i : V_i^+ \to \wp(\mathsf{DU})$ such that $\mathsf{du}_i(x_j)$ gives the set of contributions (in terms of DUs) of Player i in the jth turn. By convention, $\mathsf{du}_i(x_j) = \emptyset$ for $x_j \in V^+_{-i}$.

Definition 4. *Let $\mathcal{G} = ((V_0 \cup V_1)^\omega, \mathcal{J})$ be an ME game over $(V_0 \cup V_1)^\omega$. Let $\rho = x_0x_1x_2\ldots$ be a play of $\mathcal{G}$. Then*

Consistency: ρ is consistent *for Player i if the set $\{\mathsf{du}_i(x_j)\}_{j>0}$ is consistent. Let CONS_i denote the set of consistent plays for Player i in $\mathcal{G}$.*

Coherence: Player i is coherent *on turn $j > 0$ of play ρ if for all $\pi \in \mathsf{du}_i(x_j)$ there exists $\pi' \in (\mathsf{du}_i(x_k) \cup \mathsf{du}_{-i}(x_{k-1}))$ where $k \leq j$ such that there exits $\mathsf{R} \in \mathcal{R}$ such that $(\pi'\mathsf{R}\pi \vee \pi\mathsf{R}\pi')$ holds. Let COH_i denote the set of all coherent plays for Player i in $\mathcal{G}$.*

Responsiveness: Player i is responsive *on turn $j > 0$ of play ρ if there exists $\pi \in \mathsf{du}_j(x_j)$ such that there exits $\pi' \in \mathsf{du}_{-i}(x_{j-1})$ such that $\pi'\mathsf{R}\pi$ for some $\mathsf{R} \in \mathcal{R}$. Let RES_i denote the set of responsive plays for Player i in $\mathcal{G}$. x_j (or abusing notation, π) will be sometimes called a* response move.

Rhetorical-cooperativity: Player i is rhetorically-cooperative *in ρ if she is both coherent and responsive in every turn of hers in ρ. ρ is rhetorically-cooperative if both the players are rhetorically-cooperative in ρ. Let RC_i denote the set of rhetorically-cooperative plays for Player i in $\mathcal{G}$ and let RC be the set of all rhetorically-cooperative plays.*

To define two more constraints, NEC and CNEC, we need definitions of an 'attack' and a 'response'.

Definition 5. *Let $\mathcal{G} = ((V_0 \cup V_1)^\omega, \mathcal{J})$ be an ME game over $(V_0 \cup V_1)^\omega$. Let $\rho = x_0x_1x_2\ldots$ be a play of $\mathcal{G}$. Then*

Attack: $\mathsf{attack}(\pi', \pi)$ *on Player* $-i$ *holds at turn* j *of Player* i *just in case* $\pi \in \mathsf{du}_i(x_j)$, $\pi' \in \mathsf{du}_{-i}(x_k)$ *for some* $k \leq j$, *there is an* $\mathsf{R} \in \mathcal{R}$ *such that* $\pi' \mathsf{R} \pi$ *and: (i)* π' *entails that* $-i$ *is committed to* ϕ *for some* ϕ, *(ii)* ϕ *entails that* $\neg\phi$ *holds. In such a case, we shall often abuse notation and denote it as* $\mathsf{attack}(k, j)$. *Furthermore,* x_j *or alternatively* π *shall be called an* attack move. *An attack move is* relevant *if it is also a* response *move.* $\mathsf{attack}(k, j)$ *on* $-i$ *is* irrefutable *if there is no move* $x_\ell \in V_{-i}$ *in any turn* $\ell > j$ *such that* $\mathsf{attack}(j, \ell)$ *holds and* $x_0 x_1 \ldots x_\ell$ *is consistent for* $-i$.

Response: $\mathsf{response}(\pi', \pi)$ *on Player* $-i$ *holds at turn* j *of Player* i *if there exits* $\pi'' \in \mathsf{du}_i(x_\ell)$, $\pi' \in \mathsf{du}_{-i}(x_k)$ *and* $\pi \in \mathsf{du}_i(x_j)$ *for some* $\ell \leq k \leq j$, *such that* $\mathsf{attack}(\pi'', \pi')$ *holds at turn* k *of Player* $-i$, *there exists* $\mathsf{R} \in \mathcal{R}$ *such that* $\pi' \mathsf{R} \pi$ *and* π *implies that (i) one of* i*'s commitments* ϕ *attacked in* π' *is true or (ii) one of* $-i$*'s commitments in* π' *that entails that* i *was committed to* $\neg\phi$ *is false. We shall often denote this as* $\mathsf{response}(k, j)$.

Definition 6. *Let* $\mathcal{G} = ((V_0 \cup V_1)^\omega, \mathcal{J})$ *be an ME game over* $(V_0 \cup V_1)^\omega$. *Let* $\rho = x_0 x_1 x_2 \ldots$ *be a play of* $\mathcal{G}$. *Then*

NEC*:* NEC *holds for Player* i *in* ρ *on turn* j *if for all* ℓ, k, $\ell \leq k < j$, *such that* $\mathsf{attack}(\ell, k)$, *there exists* m, $k < m \leq j$, *such that* $\mathsf{response}(k, m)$. NEC *holds for Player* i *for the entire play* ρ *if it holds for her in* ρ *for infinitely many turns. Let* NEC_i *denote the set of plays of* $\mathcal{G}$ *where* NEC *holds for player* i.

CNEC*:* CNEC *holds for Player* i *on turn* j *of* ρ *if there are fewer attacks on* i *with no response in* ρ_j *than for* $-i$. CNEC *holds for Player* i *over a* ρ *if in the limit there are more prefixes of* ρ *where* CNEC *holds for* i *than there are prefixes* ρ *where* CNEC *holds for* $-i$. *Let* CNEC_i *be the set of all plays of* $\mathcal{G}$ *where* CNEC *holds for* i.

For a zero-sum ME game $\mathcal{G}$, the structural complexities of most of the above constraints can be derived from the constraint of rhetorical decomposition sensitivity (RDS), which is a crucial feature of many conversational goals and is defined as follows.

Definition 7. *Given a zero sum ME game* $\mathcal{G} = ((V_0 \cup V_1)^\omega, \mathsf{win})$, win *is rhetorically decomposition sensitive* (RDS) *if for all* $\rho \in \mathsf{win}$ *and for all finite prefixes* ρ_j *of* ρ, $\rho_j \in Z_1$ *implies there exists* $x \in V_0^+$ *such that* $\mathcal{O}(\rho_j x) \cap \mathsf{win} = \emptyset$.

[6] show that if Player 0 has a winning strategy for an RDS winning condition win then win is a Π_2^0 complete set. Formally,

Proposition 1 [6]. *Let* $\mathcal{G} = ((V_0 \cup V_1)^\omega, \mathsf{win})$ *be a zero-sum ME game such that* win *is* RDS. *If Player 0 has a winning strategy in* $\mathcal{G}$ *then* win *is* Π_2^0 *complete for the Borel hierarchy.*

In the zero-sum setting, CONS_0, RES_0, COH_0, NEC_0 are all RDS and it is easy to observe that Player 0 has winning strategies in all these constraints (considered individually). Hence, as an immediate corollary to Proposition 1 we have

Corollary 1. CONS_0, RES_0, COH_0, NEC_0 *are* Π^0_2 *complete for the Borel hierarchy for a zero sum ME game.*

CNEC, on the other hand, is a structurally more complex constraint. This is not surprising because CNEC can be intuitively viewed as a limiting case of NEC. Indeed, this was formally shown in [6].

Proposition 2 [6]. CNEC_i *is* Π^0_3 *complete for the Borel hierarchy for a zero sum ME game.*

The above results have interesting consequences in terms of first-order definability. Note that certain infinite sequences over our vocabulary $(V_0 \cup V_1)$ can be coded up using first-order logic over discrete linear orders $(\mathbb{N}, <)$, where $\mathbb{N}$ is the set of non-negative natural numbers. Indeed, for every i and for every $a \in V_i$, let a^i_0 be a predicate such that given a sequence $x = x_0x_1 \ldots,\ x_j \in (V_0 \cup V_1)$ for all $j \geq 0$, $x \models a^i_0(j)$ iff $x_j = a$. Closing under finite boolean operations and $\forall, \exists$, we obtain the logic $\mathsf{FO}(<)$. Now for any formula $\varphi \in \mathsf{FO}(<)$ and for any play ρ of an ME game $\mathcal{G}$, $\rho \models \varphi$ can be defined in the standard way. Thus every formula $\varphi \in \mathsf{FO}(<)$ gives a set of plays $\rho(\varphi)$ of $\mathcal{G}$ defined as:

$$\rho(\varphi) = \{\rho \in (V_0 \cup V_1)^\omega |\ \rho \models \varphi\}$$

A set $A \subseteq (V_0 \cup V_1)^\omega$ is said to be $\mathsf{FO}(<)$ definable if there exists a $\mathsf{FO}(<)$ formula φ such that $A = \rho(\varphi)$. The following result is well-known.

Theorem 1 [20]. $A \subseteq (V_0 \cup V_1)^\omega$ *is* $\mathsf{FO}(<)$ *definable if and only if* $A \in (\Sigma^0_2 \cup \Pi^0_2)$.

Thus $\mathsf{FO}(<)$ cannot define sets that are higher than the second level of the Borel hierarchy in their structural complexity. Thus as a corollary of Proposition 2 and Corollary 1, we have

Corollary 2. CONS_0, RES_0, COH_0, NEC_0 *are all* $\mathsf{FO}(<)$ *definable but* CNEC_i *is not.*

This agrees with our intuition because as we observed, CNEC_i is a limit constraint and $\mathsf{FO}(<)$, being local [14], lacks the power to capture it. To define CNEC_i one has to go beyond $\mathsf{FO}(<)$ and look at more expressive logics. One such option is to augment $\mathsf{FO}(<)$ with a counting predicate cnt which ranges over $(\mathbb{N} \cup \{\infty\})$ [19]. Call this logic $\mathsf{FO}(<, \mathsf{cnt})$. One can write formulas of the type $\exists^\infty x \varphi(x)$ in $\mathsf{FO}(<, \mathsf{cnt})$ which says that "there are infinitely many x's such that $\varphi(x)$ holds." Note that it is straightforward to write a formula in $\mathsf{FO}(<, \mathsf{cnt})$ that describes CNEC_i. Another option is to consider the logic $\mathscr{L}_{\omega_1\omega}(FO, <)$ which is obtained by closing $\mathsf{FO}(<)$ under infinitary boolean connectives $\bigvee_j$ and $\bigwedge_j$. We can define a strict syntactic subclass of $\mathscr{L}_{\omega_1\omega}(FO, <)$, denoted $\mathscr{L}^*_{\omega_1\omega}(FO, <)$, where every formula is of the form $O_pO_q \ldots O_t\varphi_{pq\ldots t}$, where, for $k \in \{p, q, \ldots, t-1\}$, $O_k = \bigvee_k$ iff $O_{k+1} = \bigwedge_{k+1}$ and each $\varphi_{pq\ldots t}$ is an $(FO, <)$ formula, $p, q, \ldots, t \in \mathbb{N}$. That is, in every formula of $\mathscr{L}^*_{\omega_1\omega}(FO, <)$, the infinitary connnectives are not nested and occur only in the beginning. We can then show that $\mathscr{L}^*_{\omega_1\omega}(FO, <)$ can express sets in any countable level of the Borel hierarchy.

3 Weighted Message Exchange Games

So far we have reviewed how the framework of Message Exchange games models strategic conversations as infinite sequential games and how we can use it to analyze the complexity of certain intuitive, winning goals in such conversations in terms of both their topological and logical complexities. Nevertheless, there are two issues with ME games that still need to be addressed.

- Let's suppose that a conversation at the outset can be potentially infinite. But still in real life, the Jury ends the game after a finite number of turns. By doing so, how can it be sure that it has correctly determined the outcome of the conversation? In other words, how does the Jury, at any point in a conversation gauge how the players are faring and how can it reliably (or even rationally) choose a winner in a finite time?
- How does the Jury determine the winning conditions win_0 and win_1? Surely, it does not come up with a arbitrary subset of $(V_0 \cup V_1)^\omega$ with an arbitrary Borel complexity.

To address the above questions, [3] introduced the model of *weighted ME games* or WME games. A WME game is an ME game where the Jury specifies the winning sets win_i as subsets of $(V_0 \cup V_1)^\omega$ by evaluating each move of every player. It does this by assigning a 'weight' or a 'score' to the moves. The cummulative weight of a conversation ρ is then the discounted sum of these individual weights.

More formally, let $\mathbb{Z}$ be the set of all integers and $\mathbb{Z}_+$ be the set of non-negative integers. For any $n \in \mathbb{Z}_+$ let $[n] = [0, n-1] \cap \mathbb{Z}_+ = \{0, 1, \ldots, n-1\}$. A weight function is a function $w : (Z_0 \times V_1^+ \cup Z_1 \times V_0^+) \to \{0,1,2\} \times \{0,1,2\}$. Intuitively, given a history $\rho \in Z$, w assigns a tuple of integers $(a_0, a_1) = w(\rho, x)$ to the next legal move x of the play ρ. A weight of 0 is intended to denote a 'bad' move, 1 a 'neutral' or 'average' move, and 2 is intended to denote a 'good' or 'strong' move. An example of a 'strong' move is an attack CDU whereas an example of a 'bad' move can be an incoherent CDU, as defined in Sect. 2. Note that the weight function, w depends on the current history of the game in that, given two different histories $\rho_1, \rho_2 \in Z$, it might be the case that $w(\rho_1, x) \neq w(\rho_2, x)$ for the same continuing move x. For notational simplicity, in what follows, given a play $\rho = x_0 x_1 \ldots$, we shall denote by $w_i^j(\rho)$, the weight assigned by w to Player i in the jth turn of ρ ($j \geq 1$). That is, if $w(\rho_{j-1}, x_j) = (a_0, a_1)$, then $w_0^j(\rho) = a_0$ and $w_1^j(\rho) = a_1$.

A discounting factor is a real $\lambda \in (0,1)$. For every play ρ of an ME game $\mathcal{G}$, the Jury, using some discounting factor λ, computes the discounted-weight of ρ for each player i, which is denoted by $w_i(\rho)$ and is defined as:

Definition 8. *Let ρ be a play of $\mathcal{G}$ and let λ be a discounting factor. Then the discounted-weight of ρ for Player i is given by*

$$w_i(\rho) = \sum_{j \geq 1} \lambda^{j-1} w_i^j(\rho)$$

We can now consider the Jury simply as a tuple (w, λ) where w is a weight function and λ is a discounting factor.[2] And formally define WME games as:

Definition 9. *A Weighted Message Exchange game (WME game) is a tuple* $\mathcal{G} = ((V_0 \cup V_1)^\omega, (w, \lambda))$.

We can now use w and λ to implicitly determine the winning sets win_i of the players and turn $\mathcal{G}$ into either a zero-sum or a non zero-sum game.

Definition 10. *Let* $\mathcal{G} = ((V_0 \cup V_1)^\omega, (w, \lambda))$ *be WME game. Then*

i. **Zero-sum:** $\mathsf{win} = \{\rho \in (V_0 \cup V_1)^\omega | w_0(\rho) \geq w_1(\rho)\}$.
ii. **Non-zero sum:** *Fix constants* $\nu_i \in \mathbb{R}$ *called 'thresholds'. Then,*

$$\mathsf{win}_i = \{\rho \in (V_0 \cup V_1)^\omega | w_i(\rho) \geq \nu_i\}.$$

For this exposition, we concentrate on the zero-sum setting. Winning strategies are then defined as in Sect. 2. We can also define the notions of best-response and ϵ-best-response strategies for a given $\epsilon > 0$. This leads to the definition of a Nash-equilibrium and an ϵ-Nash-equilibrium. It can also be shown that ϵ-Nash-equilibia always exist in WME games (see [3] for more details). It was also shown in [3] that given an $\epsilon > 0$ there exists $n_\epsilon \in \mathbb{Z}_+$ such that after n_ϵ turns neither player can gain more than just a 'small amount' than what they have already gained so far. More formally,

Proposition 3 [3]. *Let* $\mathcal{G} = ((V_0 \cup V_1)^\omega, (w, \lambda))$ *be a WME game. Then given* $\epsilon > 0$ *we have for Player* i *and any play* ρ *of* $\mathcal{G}$

$$\sum_{j=1}^{n_\epsilon} \lambda^{j-1} w_i^j(\rho) - \epsilon \leq w_i(\rho) \leq \sum_{j=1}^{n_\epsilon} \lambda^{j-1} w_i^j(\rho) + \epsilon$$

where $n_\epsilon \leq \frac{\ln[\frac{\epsilon}{2}(1-\lambda)]}{\ln \lambda}$.

Thus if the Jury stops the conversation ρ after n_ϵ turns it is guaranteed that no player could have gained more than ϵ from what they have already gained so far. Thus, it may already be able to come to a conclusion after n_ϵ turns of the game - if Player i has already gained much more than 2ϵ than Player $-i$, then i may be declared the winner.

We have thus answered both the questions posed at the beginning of the section.

Let's now consider an application of WME games to the segment of a real-life debate.

[2] Note that [3] considers the discounting as a function of the history rather than a constant factor which, arguably, better reflects real-life situations. We stick to a constant discounting factor here for the simplicity of presentation. The main concepts remain the same.

Example 4. Consider the following excerpt from the 1988 Dan Quayle-Lloyd Bentsen Vice-Presidential debate that has exercised us now for several years. Quayle (Q), a very junior and politically inexperienced Vice-Presidential candidate, was repeatedly questioned about his experience and his qualifications to be President. Till a point in the dbate both of them were going neck to neck. But then to rebut doubts about his qualifications, Quayle compared his experience with that of the young John (Jack) Kennedy. To that, Bentsen (BN) made a discourse move that Quayle apparently did not anticipate. We give the relevant part of the debate below where for the simplicity of the ensuing analysis we have labeled each CDU:

a. **Quayle:** *... the question you're asking is, "What kind of qualifications does Dan Quayle have to be president,"*
b. **Quayle:** *... I have far more experience than many others that sought the office of vice president of this country. I have as much experience in the Congress as Jack Kennedy did when he sought the presidency.*
c. **Bensten:** *Senator, I served with Jack Kennedy. I knew Jack Kennedy. Jack Kennedy was a friend of mine. Senator, you're no Jack Kennedy.*
d. **Quayle:** *That was unfair, sir. Unfair.*
e. **Bensten:** *You brought up Kennedy, I didn't.*

Let us analyze the above exchange from the perspective of a WME game. Without loss of generality suppose Quayle is Player 0 and Bensten is Player 1. Let us denote by ρ all the conversation that took place before the above exchange. Since both of them were neck-neck till then we can assume that both had gained a weight of c (say) that far. Next, Quayle makes moves (a) and (b) which might be considered an average move at that point (the audience applauds but is skeptical). So we can assign $w(\rho, \langle a\rangle\langle b\rangle) = (1, 1)$ - Bensten neither gains nor loses from this move of Quayle. Bensten then makes the brilliant move (b) which does serious damage to Quayle. The audience bursts with applause. Hence, we set $w(\rho\langle a\rangle\langle b\rangle, \langle c\rangle) = (0, 2)$. Quayle is unable to retaliate to (b) and makes another rather timid move (c) which has even a negative impact to his cause on the audience. The audience is still basking in Bensten's previous move and we set $w(\rho\langle a\rangle\langle b\rangle\langle c\rangle, \langle d\rangle) = (1, 1)$. Bensten goes ahead and cements his position further by making another attack move (d) on Quayle. We hence set $w(\rho\langle a\rangle\langle b\rangle\langle c\rangle\langle d\rangle, \langle e\rangle) = (0, 2)$.

Now suppose the Jury (in this case the audience) is using a discount factor λ. The discounted-weights to Quale and Bensten are respectively:

$$w_Q(\rho\langle a\rangle\langle b\rangle\langle c\rangle\langle d\rangle\langle e\rangle) = 1 + \lambda^2$$

and

$$w_{BN}(\rho\langle a\rangle\langle b\rangle\langle c\rangle\langle d\rangle\langle e\rangle) = 1 + 2\lambda + \lambda^2 + 2\lambda^3$$

We thus see that $w_{BN}(\rho\langle a\rangle\langle b\rangle\langle c\rangle\langle d\rangle\langle e\rangle) > w_Q(\rho\langle a\rangle\langle b\rangle\langle c\rangle\langle d\rangle\langle e\rangle)$ for any value of $\lambda \in (0, 1)$. Not just that, even if after the above initial slump, Quayle plays in such a way that every move he makes is a brilliant move and every move

Bensten makes is a disaster, Quayle still cannot recover and gain more than Bensten eventually for values of λ as high as 0.8! Discounting thus reiterates the fact that it is always beneficial to makes ones best moves earlier on in a debate. This also 'colours' the weighting function of the Jury in ones favour.

In passing, we would like to remark that Quayle never recovered from one disastrous move in that debate and lost handily as is rightly predicted by our model.

4 Imperfect Information and Epistemic Considerations

WME games address certain open questions in the theory of ME games, as we have shown in the previous section. But they give rise to other questions as well.

- How does the Jury determine a weighting scheme?
- If the Jury is identified simply with a weighting function and a discount factor, and players know these parameters, they can determine when the Jury will end the game. So don't WME games fall prey to troublesome backwards induction arguments that ME games were designed to avoid?

Concerning the first question, we've shown that the predictions of WME games hold for a wide range of weighting schemes, but indeed it is clear that different Juries will have different weighting schemes. Consider how a partisan audience say of a political candidate c reacts to his discourse moves and how a audience hostile to c's views reacts. The U.S. Presidential primary debates and general debates show that these reactions can vary widely. In particular, Juries may be biased and only "hear what they want to hear," even to the extent that they ignore inconsistencies or incoherences on the part of their preferred player. Concrete Juries adopt the weighting schemes they do, in virtue of their beliefs and desires. Thus, weighting schemes may vary quite widely, and a conversational participant should be as well informed as she can be about the Jury she wants to sway.

The second question needs a negative response. [3] simply assumes that the Jury's characteristics are unknown to the conversational participants. But this is not really realistic, especially in virtue of our response to the first question above. So in this section, we study the exact information structure implicit in the strategic reasoning in conversations by extending framework of ME games with epistemic notions. We use the well-established theory of type-structures, first introduced in [17] and widely studied since. We assume that each player $i \in (\{0,1\} \cup \{\mathcal{J}\})$ has a (possibly infinite) set of types T_i. With each type t_i of Player i is associated a (first-order) belief function $\beta_i(t_i)$ which assigns to t_i a probability distribution over the types of the other players. That is, $\beta_i : T_i \rightarrow \Delta(\prod_{j \neq i} T_j)$. $\beta_i(t_i)$ represents the 'beliefs' of type t_i of Player i about the types of the other players and the Jury. The higher-order beliefs can be defined in a standard way by iterating the functions β_i. We assume that each type t_i of each Player i starts the game with an initial belief $\beta_i(t_i) \in \Delta(\prod_{j \neq i} T_j)$, called the 'prior belief'. The players take turns in making their moves and after every

move, all the players dynamically update their beliefs through Bayesian updates. The notions of 'optimal strategies', 'best-response', 'rationality', 'common belief in rationality' etc. can then be defined in the standard way (see [12]).

Having imposed the above epistemic structure on ME games, we can now reason about the 'rationality' of the players' strategies. In order to justify or predict the outcome of games, many different solution concepts viz., Nash equilibrium, iterated removal of dominated strategies, correlated equilibrium, rationalizability etc. have been proposed [7,10,21]. Most of them have also been characterized in terms of the exact belief structure and strategic behavior of the players (see [12] for an overview). We can borrow results from this rich literature to predict or justify outcomes in strategic conversations. The details of the above is ongoing work and we leave it to an ensuing paper. However, let us apply the above concepts and analyze our original example of Bronston and the Prosecutor.

To illustrate the power of types, let us return to Eg. 1. One conversational goal of the Prosecutor in Eg. 1 is to get Bronston to commit to an answer eventually (and admit to an incriminating fact) or to continue to refuse to answer (in which case he will be charged with contempt of court). Under such a situation, the response 1d of Bronston is clearly a clever strategic move. Bronston's response (1d) was a strategic move aimed to 'misdirect' the Jury $\mathcal{J}$. He believed that $\mathcal{J}$ was of a type that would be convinced by his ambiguous response and neither incriminate him nor charge him with perjury nor of contempt of court. His move was indeed rational, *given his belief* about the Jury type. It turns out that while the jury of a lower court $\mathcal{J}_1$ was not convinced of Bronston's arguments and charged him with perjury, a higher court $\mathcal{J}_2$ overturned the verdict and released him. Thus his belief agreed with $\mathcal{J}_2$ but not $\mathcal{J}_1$.

We now return briefly to the information players have about the Jury in WME games. Intuitively, each Player i is uncertain about: (i) the type of the other player $(1-i)$, (ii) the strategy that $(1-i)$ is employing and (iii) the type of the Jury which is the discounting factor λ and the weight function w. We thus assume that at every history ρ of an ME/WME game $\mathcal{G}$ each type $t_i \in \mathcal{T}_i$ of Player i has beliefs on:

1. the set of types $T_{(1-i)}$ of Player $(1-i)$,
2. the set of strategies $S_{(1-i)}$ of Player $(1-i)$,
3. the weight function w.
4. the discounting factor λ.

Although going into the details of each one of these points would take too long for this exposition, we can show that each of these factors can be modelled precisely preserving our intuitions. This supplies us with a needed answer to our second question. Indeed, the Jury ends the game after a finite number of turns n (say), and from its viewpoint, the game is finite. But note that the players are uncertain about the exact value of n and hold beliefs about it. Hence, from their viewpoint, although the game ends after finitely many turns, they do not know the exact number of turns. Thus, intuitively a rational player is one who strategizes for a wide range of possibilities for the value of n [this will be elaborated presently]. For her, the game is 'potentially infinite'. And hence, we

as analysts, model the situation as an infinite game as well. In [5], we argued that an infinitary approach was needed to handle both technical issues having to do with Backwards Induction arguments as well as to capture the intuition that a conversationalist, to be sure of succeeding in convincing a Jury of a particular position, should be prepared to argue for her position for "as long as it takes" and to answer every possible objection by an opponent. Since the list of possible objections is most likely infinite, the analyst must provide an infinitary game-theoretic framework. These points still hold once we add an epistemic layer to WME games.

5 Conclusion

In this paper we have summarized concepts from earlier work and have demonstrated how infinite sequential games paired with the notion of a Jury, ME games, can be used to model strategic conversations. Such a model allows one to reason about the structure and complexity of various kinds of winning goals that conversationalists might have. We have shown how to use tools from topology, set-theory and logic to express such goals. We then discussed a problem with pure ME games: how can an actual Jury reliably determine a winner or winners in a conversation after only finitely many rounds. We addressed this issue by moving to Weighted ME (WME) games. We showed how to apply elements of WME games to a snippet of a historic moment in American political debates. However, WME games, we also showed, don't furnish a completely satisfactory analysis, because though the Jury can reliably determine a winner or winners of a conversation after a finite moment, this information crucially cannot be common knowledge of the participants without re-introducing the damaging backwards induction arguments that ME games were originally designed to solve. We then demonstrated how we can use ideas from epistemic game theory would in principle solve this problem.

Thus, what we have put forward in this paper is a framework for an epistemic, game-theoretic approach to conversation. As far as we know, this approach is utterly different from any other model proposed for the study of linguistic conversation, though it may have other applications as well. There are many directions into which we would like to delve deeper in the future. One such direction, as we already mentioned, is to work out the epistemic theory of ME games in full detail. That is our current work in progress. Another direction has to do with a more detailed investigation of the Jury, or possible Juries. So far we have considered the Jury as a 'passive' entity; it simply evaluates the play and determines the winner. In real life situations, however, the Jury actively participate in the conversation itself, albeit typically in a limited way. It can applaud or boo moves of the players. Thus, the Jury can be seen as making these moves in the game. Based on what the players observe about the Jury, they may update or change their beliefs and vice-versa. Incorporating this into our ME games requires a modification of the current framework where the Jury is another player making moves from its own set of vocabulary. We plan to explore this in future work.

Finally, in addition to the Jury, debates usually also have a moderator whose job is to conduct the debate and assign turns to the players. The moderator may also actively 'pass comments' about the moves of the players. A fair moderator gives all the players equal opportunity to speak and put their points across. However, if the moderator is unfair, he may 'starve' a particular player by not letting her enough chance to speak, respond to attacks and so on. Exploring the effects a biased moderator can have on conversations is another interesting, future topic of research.

References

1. Asher, N., Lascarides, A.: Logics of Conversation. Cambridge University Press, Cambridge (2003)
2. Asher, N., Lascarides, A.: Strategic conversation. Semantics, Pragmatics **6**(2) (2013). http://dx.doi.org/10.3765/sp.6.2
3. Asher, N., Paul, S., Evaluating conversational success: weighted message exchange games. In: Hunter, J., Stone, M. (eds.) 20th Workshop on the Semantics and Pragmatics of Dialogue (SEMDIAL), New Jersey, USA, July 2016 (2016, to appear)
4. Hintikka, J.: Language-games. In: Saarinen, E. (ed.) Game-Theoretical Semantics. Synthese Language Library, vol. 6790, pp. 1–26. Springer, Netherlands (2011). doi:10.1007/978-1-4020-4108-2_1
5. Asher, N., Paul, S., Venant, A.: Message exchange games in strategic conversation. J. Philos. Log. (2016). doi:10.1007/s10992-016-9402-1
6. Asher, N., Paul, S., Venant, A.: Message exchange games in strategic conversations. J. Philos. Log. (2016, in press)
7. Aumann, R.: Subjectivity and correlation in randomized strategies. J. Math. Econom. **1**, 67–96 (1974)
8. Baltag, A., Moss, L.S.: Logics for epistemic programs. Synthese **139**(2), 165–224 (2004)
9. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge and private suspicions. Technical report SEN-R9922, Centrum voor Wiskunde en Informatica (1999)
10. Bernheim, B.D.: Rationalizable strategic behaviour. Econometrica **52**(4), 1007–1028 (1984)
11. Crawford, V., Sobel, J.: Strategic information transmission. Econometrica **50**(6), 1431–1451 (1982)
12. Dekel, E., Siniscalchi, M.: Epistemic game theory. In: Aumann, R.J., Hart, S. (eds.) Handbook of Game Theory with Economic Applications, vol. 4, chap. 12, pp. 619–702. Elsevier Publications (2015)
13. Franke, M.: Semantic meaning and pragmatic inference in non-cooperative conversation. In: Icard, T., Muskens, R. (eds.) Interfaces: Explorations in Logic. Language and Computation, Lecture Notes in Artificial Intelligence, pp. 13–24. Springer-Verlag, Berlin, Heidelberg (2010)
14. Gaiffman, H.: On local and non-local properties. In: Proceedings of the Herbrand Symposium, Logic Colloquium 1981. North Holland (1982)
15. Glazer, J., Rubinstein, A.: On optimal rules of persuasion. Econometrica **72**(6), 119–123 (2004)
16. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata Logics, and Infinite Games: A Guide to Current Research. LNCS, vol. 2500. Springer, Heidelberg (2002)

17. Harsanyi, J.C.: Games with incomplete information played by bayesian players, parts i-iii. Manag. Sci. **14**, 159–182 (1967)
18. Kechris, A.: Classical Descriptive Set Theory. Springer-Verlag, New York (1995)
19. Libkin, L.: Elements of finite model theory. Springer, Heidelberg (2004)
20. McNaughton, R., Papert, S.: Counter-free automata. In: Research Monograph, vol. 65. MIT Press, Cambridge (1971)
21. Nash, J.: Non-cooperative games. Ann. Math. **54**(2), 286–295 (1951)
22. Spence, A.M.: Job market signaling. J. Econom. **87**(3), 355–374 (1973)
23. Venant, A.: Structures, semantics and games in strategic conversations. Ph.D. thesis, Université Paul Sabatier, Toulouse (2016)
24. Venant, A., Asher, N.: Dynamics of public commitments in dialogue. In: Proceedings of the 11th International Conference on Computational Semantics, pp. 272–282, London, UK, Association for Computational Linguistics, April 2015
25. Asher, N., Venant, A.: Ok or not ok? In: Semantics and Linguistic Theory 25. Cornell University Press, New York (2015)

Ramsey Theory on Trees and Applications

Natasha Dobrinen(✉)

University of Denver, 2280 S Vine Street, Denver, USA
Natasha.Dobrinen@du.edu

Modern Ramsey Theory on infinite structures began with the following seminal result of Ramsey.

Theorem 1 (Ramsey, [14]). *For each positive integer k and each finite coloring of all k-sized subsets of the natural numbers, $\mathbb{N}$, there is an infinite set M of natural numbers such that each k-sized subset of M has the same color.*

This result was motivated by and applied to solve a problem in logic regarding canonical k-ary relations on the natural numbers. Ramsey's Theorem has been extended in a myriad of directions, for instance, varying sizes of sets colored, varying the number of colors allowed, including infinitely many colors, and coloring more complex structures. Progress in Ramsey theory has led to progress in a wide array of mathematical areas, such as model theory, set theory, and logic in general, as well as algebra, analysis, topology and dynamics. In this talk, we concentrate on Ramsey theory on trees and applications to homogeneous structures.

A key result en route to the proof that the Boolean Prime Ideal Theorem is strictly weaker than the Axiom of Choice (see [9]) is the Ramsey-type theorem of Halpern and Läuchli on trees. There are many variations of the Halpern-Läuchli Theorem (see [18]); here we shall state the strong tree version. Let T be a finitely branching tree of height ω with no terminal nodes, and let $T(n)$ denote the nodes on the n-th level of T. A subtree $S \subseteq T$ is called a *strong subtree* of T if for each level of n of S at which some node in S branches, every node in $S(n)$ branches maximally in T. The following is the Strong Tree Version of the Halpern-Läuchli Theorem, proved in another form in [8].

Theorem 2. *Let $d \geq 1$ and let T_i, $i < d$, be finitely branching trees of height ω. Given any finite coloring of $\bigcup_{n<\omega} \prod_{i<d} T_i(n)$, there are strong subtrees $S_i \subseteq T_i$, all with the same infinite set L of branching levels, such that, for all $n \in L$, all members of $\prod_{i<d} S_i(n)$ have the same color.*

For one tree, Milliken strengthened the Halpern-Läuchli Theorem by showing that for any given any finitely branching strong tree T of height ω, given any finite strong tree U and a coloring of all copies of U in T by finitely many colors, there is a strong subtree $S \subseteq T$ of infinite height in which all copies of U have the same color. (See [13].) In the terminology of [18], the collection of strong subtrees of T forms a topological Ramsey space.

Milliken's Theorem has found numerous applications to homogeneous relational structures including the following. In [16], Sauer applied Milliken's Theorem in his proof that the Rado graph $\mathcal{R}$, also known as the infinite random graph,

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 19–22, 2017.
DOI: 10.1007/978-3-662-54069-5_2

and other homogeneous universal binary structures, have finite Ramsey degrees. This means that for each finite graph G, there is a finite number t_G such that given any finite coloring of all copies of G in $\mathcal{R}$, there is a copy $\mathcal{R}'$ of $\mathcal{R}$ in which all copies of G take on at most t_G colors. Moreover, for all graphs G with two or more vertices, the Ramsey number t_G is greater than one. Avilés and Todorcevic applied Milliken's Theorem in [1] to find a finite basis for analytic strong n-gaps. More recently, they developed a new type of Miiliken's Theorem in [2] in order to classify minimal analytic gaps. A dual version of the Halpern-Läuchli Theorem was established by Todorcevic and Tyros in [19].

Building on Sauer's techniques, Dobrinen, Laflamme and Sauer employed Milliken's Theorem to prove in [5] that the Rado graph, and more generally simple binary relational structures, have the rainbow Ramsey property, even though they do not have the Ramsey property. The rainbow Ramsey property states that for each finite k, each finite graph G, and each coloring of the copies of G in $\mathcal{R}$ by ω many colors, where each color appears at most k times, there is a copy $\mathcal{R}'$ inside $\mathcal{R}$ where each color appears at most once.

Extending Sauer's result in another direction, Laflamme, Sauer and Vuksanovic used Milliken's Theorem to obtain canonical partitions for finitary as well as countable colorings of n-tuples in countable homogeneous binary relational structures in [12]. More recently, Vlitas has established a Ramsey-classification theorem for equivalence relations on sets of finite strong subtrees of finitely many countably infinite strong trees in [20].

Turning now to trees on uncountable cardinals, Shelah proved in [17] that it is consistent with ZFC (the standard axioms of set theory) that a version of Milliken's Theorem holds for one strong tree on a measurable cardinal κ. In that theorem, the tree has height κ and less than κ-sized branching on each level, and the coloring is on m-sized subsets of levels of the tree, where m is some fixed positive integer. This result was augmented by Džamonja, Larson and Mitchell in [6] to prove homogeneity for colorings of m-sized antichains in a strong tree on a measurable cardinal. They then applied that result to obtain canonical partitions of m-sized subsets of the κ-rationals in [6] and canonical partitions for colorings of finite subgraphs of the universal graph on κ vertices in [7]. Recently, Dobrinen and Hathaway in [4] proved consistency of the strong subtree version of Milliken's Theorem for finitely many trees on a measurable cardinal, also establishing results for trees on weakly compact cardinals.

All of the proofs of the results mentioned in this paragraph use the set-theoretic method of forcing, using ideas from an unpublished proof of Harrington of the strong tree version of the Halpern-Läuchli Theorem for finitely many finitely branching strong trees of countable height. Recently, Dobrinen has built on these ideas to prove a version of Milliken's Theorem relevant to the universal homogeneous triangle-free graph.

A *triangle-free graph* is a graph which omits triangles. The universal homogeneous triangle-free graph is the Fraïssé limit of the Fraïssé class of all finite triangle-free graphs, which we shall denote by $\mathcal{H}_3$. Each countable triangle-free graph embeds into $\mathcal{H}_3$. A construction of $\mathcal{H}_3$ was given by Henson in [10], where

among other things, he proved that for any coloring of the vertices of $\mathcal{H}_3$ into two colors, there is either a copy of $\mathcal{H}_3$ in the first color, or else there are copies of each finite triangle-free graph in the second color. Later, it was proved by Komjáth and Rödl in [11] that for any coloring of the vertices in $\mathcal{H}_3$ into two colors, there is a copy of $\mathcal{H}_3$ with all vertices having the same color.

The question of colorings of vertices being resolved, interest turned to colorings of copies of finite triangle-free graphs G in $\mathcal{H}_3$. The *Ramsey degree* of a finite triangle-free graph G is the smallest number t_G such that for any coloring of the copies of G in $\mathcal{H}_3$ into finitely many colors, there is always a copy $\mathcal{H}'$ of $\mathcal{H}_3$ in which all copies of G take on at most t_G colors. If there is no such bound, then we write $t_G = \infty$. The *big Ramsey numbers problem* for $\mathcal{H}_3$ is the problem of finding out whether or not each finite triangle-free graph G has $t_G < \infty$.

Sauer proved in [15] that for G being an edge, that is a graph with two vertices with one edge between them, $t_G = 2$. In recent work, the Dobrinen has developed a notion of strong tree coding triangle-free graphs. Using ideas from Harrington's forcing proof of the Halpern-Läuchli Theorem, the author has proved an analogue of Milliken's Theorem for these *strong triangle-free trees*, from which it follows that the spaces of strong triangle-free trees are almost topological Ramsey spaces. Using this plus a new type of so-called subtree envelope, the Dobrinen has recovered the results in [11] for vertices and [15] for edges, as well as other finite graphs. At the time of writing this abstract, it looks like all finite triangle-free graphs have finite Ramsey degrees, though the paper [3] is not yet in final form.

This talk will provide an overview of the various versions of the Halpern-Läuchli Theorem and Milliken Theorem and the applications mentioned in this abstract. The author aims to convey the fascinating confluence of ideas from logic, Ramsey theory and set theory leading to applications to solving problems in model theory/universal relational structures.

Acknowledgments. The author gratefully acknowledges the support of NSF Grants DMS-142470 and DMS-1600781.

References

1. Avilés, A., Todorcevic, S.: Finite basis for analytic strong n-gaps. Combinatorica **33**(4), 375–393 (2013)
2. Avilés, A., Todorcevic, S.: Types in the n-adic tree and minimal analytic gaps. Adv. Math. **292**, 558–600 (2016)
3. Dobrinen, N.: The universal triangle-free graph has finite Ramsey degrees. (2016, in preparation)
4. Dobrinen, N., Hathaway, D.: The Halpern-Läuchli Theorem at a measurable cardinal (2016, submitted). 15 pages
5. Dobrinen, N., Laflamme, C., Sauer, N.: Rainbow Ramsey simple structures. Discrete Math. **339**(11), 2848–2855 (2016)
6. Džamonja, M., Larson, J., Mitchell, W.J.: A partition theorem for a large dense linear order. Israel J. Math. **171**, 237–284 (2009)

7. Džamonja, M., Larson, J., Mitchell, W.J.: Partitions of large Rado graphs. Arch. Math. Logic **48**(6), 579–606 (2009)
8. Halpern, J.D., Läuchli, H.: A partition theorem. Trans. Am. Math. Soc. **124**, 360–367 (1966)
9. Halpern, J.D., Lévy, A.: The Boolean prime ideal theorem does not imply the axiom of choice. In: Axiomatic Set Theory, pp. 83–134. American Mathematical Society (1971). Proceedings of the Symposium on Pure Mathematics, Vol. XIII, Part I, University California, Los Angeles, Calififfornia (1967)
10. Henson, C.W.: A family of countable homogeneous graphs. Pac. J. Math. **38**(1), 69–83 (1971)
11. Komjáth, P., Rödl, V.: Coloring of universal graphs. Graphs Comb. **2**(1), 55–60 (1986)
12. Laflamme, C., Sauer, N., Vuksanovic, V.: Canonical partitions of universal structures. Combinatorica **26**(2), 183–205 (2006)
13. Milliken, K.R.: A partition theorem for the infinite subtrees of a tree. Trans. Am. Math. Soc. **263**(1), 137–148 (1981)
14. Ramsey, F.P.: On a problem of formal logic. Proc. Lon. Math. Soc. **30**, 264–296 (1929)
15. Sauer, N.: Edge partitions of the countable triangle free homogenous graph. Discrete Math. **185**(1–3), 137–181 (1998)
16. Sauer, N.: Coloring subgraphs of the Rado graph. Combinatorica **26**(2), 231–253 (2006)
17. Shelah, S.: Strong partition relations below the power set: consistency - was Sierpinski right? II. In: Sets, Graphs and Numbers, Budapest, vol. 60, pp. 637–688 (1991). Colloq. Math. Soc. János Bolyai, North-Holland
18. Todorcevic, S.: Introduction to Ramsey Spaces. Princeton University Press, Princeton (2010)
19. Todorcevic, S., Tyros, K.: A disjoint unions theorem for threes. Adv. Math. **285**, 1487–1510 (2015)
20. Vlitas, D.: A canonical partition relation for uniform families of finite strong subtrees. Discrete Math. **335**, 45–65 (2014)

Automata, Logic and Games for the λ-Calculus

C.-H. Luke Ong(✉)

University of Oxford, Oxford, UK
Luke.Ong@cs.ox.ac.uk

Automata, logic and games provide the mathematical theory that underpins the model checking of reactive systems:

- *automata* on infinite words and trees as models of computation for state-based systems,
- *logical systems* such as temporal and modal logics for specifying correctness properties, and
- *two-person games* as a mathematical model of the interactions between a system and its environment.

An elegant and fundamental result in the theory of automata, logic and games [3] is the correspondence between alternating parity tree automata (APT), the modal mu-calculus L_μ, and parity games, which are standard formalisms for algorithmic reasoning about trees. On the one hand, the mu-calculus model-checking problem and the PARITY decision problem (Does Verifier have a winning strategy in a given parity game?) are interreducible [17,19]. On the other, modal mu-calculus and alternating parity tree automata are recursively equivalent for defining tree languages [2,8,19].

Research in model checking has traditionally concerned itself with the verification of properties of "ground type objects" such as words or trees. A recent trend in algorithmic verification is *higher-order model checking* [4,9,10], which is the model checking of infinite trees generated by the $\lambda\mathbf{Y}$-calculus (simply-typed lambda calculus extended with fixpoint operators) [13,15] or, equivalently, recursion schemes. Higher-order model checking has been applied with some success to the verification of higher-type functional programs [5,7,11,14]. In this model checking approach, the verification problem is reduced to the model checking of Böhm trees, which are the computation trees of functional programs.

Our work is motivated by the question: *what is the automata-logic-games correspondence for (higher-type) Böhm trees?* Simply-typed Böhm trees are ordered ranked trees extended with binders in the form of lambda-abstractions; which may be viewed as higher-order functions on trees. Indeed, one may well ask if such a correspondence is plausible, since Clairambault and Murawski [1] have considered a monadic second order logic over a class of binding structures and shown the model checking problem to be undecidable. However, we develop just such a correspondence for Böhm trees. In the following we discuss the main ideas.

Abstract of an invited talk presented at ICLA 2017.

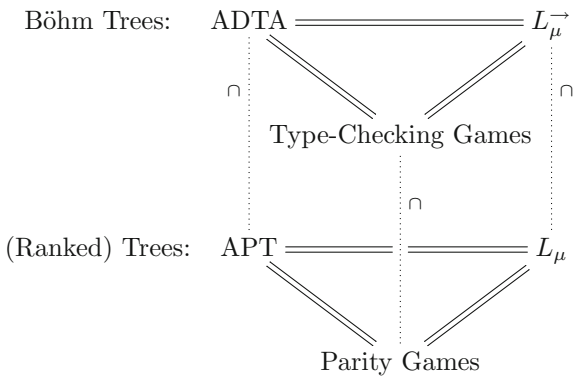
S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 23–26, 2017.
DOI: 10.1007/978-3-662-54069-5_3

The starting point is recent work by Tsukada and Ong [18] on the model checking problem for higher-type (possibly infinite) Böhm trees. They introduced a notion of type [6] for Böhm trees: a Böhm tree u is said to have type σ if Verifier has a winning strategy in the corresponding *type-checking game*. Tsukada and Ong showed that the type checking of $\lambda\mathbf{Y}$-definable Böhm trees is decidable. This type-checking game is the games component of the new trio for higher types.

The automata component of the trio is *alternating dependency tree automata* (ADTA), which was first introduced by Stirling [16] for finite binding trees, in order to characterise solution sets of the Higher-Order Matching Problem. We extend ADTA to infinite binding trees with ω-regular winning conditions. ADTA are closed under union, intersection and complementation. The emptiness problem for nondeterministic dependency tree automata is decidable, but undecidable for alternating dependency tree automata [12]. A key result is that types and ADTA are effectively equivalent[1] for defining languages of Böhm trees. As a corollary, the ADTA acceptance problem for $\lambda\mathbf{Y}$-definable Böhm trees is decidable.

The logic for describing the corresponding correctness properties is *higher-type mu-calculus* $L_\mu^{\rightarrow}$, which extends ordinary modal mu-calculus with predicates for detecting variables, and corresponding constructs for detecting λ-abstractions. There is a characterisation of the set-theoretic semantics of $L_\mu^{\rightarrow}$ by a model checking game. Furthermore $L_\mu^{\rightarrow}$ and ADTA are recursively equivalent for defining languages of Böhm trees.

Thus there is an exact automata-logic-games correspondence for Böhm trees at higher types, which naturally extends the classical correspondence for ordinary trees, as illustrated by the following diagram.



[1] We suppress a delicate distinction between types and a subsystem of *parity permissive types*. There is a corresponding distinction between ADTA and a subclass of *parity permissive ADTA*. The expressive equivalence result holds both generally and when restricted to the parity permissive subsystems.

Acknowledgements. This is based on joint work with Matthew Hague, Steven Ramsay, and Takeshi Tsukada, partially funded by EPSRC UK. Part of the work was done while the authors were visiting the Institute for Mathematical Sciences, National University of Singapore in 2016. The visit was partially supported by the Institute.

References

1. Clairambault, P., Murawski, A.S.: Böhm trees as higher-order recursive schemes. In: Proceedings of IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2013), LIPIcs, vol. 24, pp. 91–102. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2013)
2. Emerson, E.A., Jutla, C.S.: Tree automata, mu-calculus and determinacy (extended abstract). In: 32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, vol. 1–4 , pp. 368–377, October 1991
3. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata, Logics, and Infinite Games: A Guide to Current Research. LNCS, vol. 2500. Springer, Heidelberg (2002). doi:10.1007/3-540-36387-4
4. Knapik, T., Niwiński, D., Urzyczyn, P.: Higher-order pushdown trees are easy. FoSSaCS **2002**, 205–222 (2002)
5. Kobayashi, N.: Model checking higher-order programs. J. ACM **60**(3), 1–62 (2013)
6. Kobayashi, N., Ong, C.-H.L.: A type system equivalent to the modal mu-calculus model checking of higher-order recursion schemes. In: Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, 11–14 August 2009, Los Angeles, CA, USA, pp. 179–188 (2009)
7. Kobayashi, N., Sato, R., Unno, H.: Predicate abstraction and CEGAR for higher-order model checking. In: Hall, M.W., Padua, D.A. (eds.) PLDI, pp. 222–233. ACM (2011)
8. Kupferman, O., Vardi, M.Y., Wolper, P.: An automata-theoretic approach to branching-time model checking. J. ACM **47**(2), 312–360 (2000)
9. Ong, C.-H.L.: On model-checking trees generated by higher-order recursion schemes. In: Proceedings of 21th IEEE Symposium on Logic in Computer Science (LICS 2006), pp. 81–90. IEEE Computer Society (2006)
10. Ong, C.-H.L.: Higher-order model checking: an overview. In: 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, 6–10 July 2015, pp. 1–15 (2015)
11. Ong, C.-H.L., Ramsay, S.J.: Verifying higher-order functional programs with pattern-matching algebraic data types. In: POPL 2011, vol. 46, pp. 587–598, January 2011
12. Ong, C.-H.L., Tzevelekos, N.: Functional reachability. In: 2009 24th Annual IEEE Symposium on Logic in Computer Science (LICS 2009), pp. 286–295, August 2009
13. Platek, R.A.: Foundations of recursion theory. Ph.D. thesis, Standford University (1966)
14. Ramsay, S.J., Neatherway, R.P., Ong, C.-H.L.: A type-directed abstraction refinement approach to higher-order model checking. In: The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2014, San Diego, CA, USA, 20–21 January 2014, pp. 61–72. ACM (2014)
15. Scott, D.S.: A type-theoretical alternative to ISWIM, CUCH, OWHY. Theor. Comput. Sci. **121**(1&2), 411–440 (1993)

16. Stirling, C.: Dependency tree automata. In: Alfaro, L. (ed.) FoSSaCS 2009. LNCS, vol. 5504, pp. 92–106. Springer, Heidelberg (2009). doi:10.1007/978-3-642-00596-1_8
17. Streett, R.S., Emerson, E.A.: An automata theoretic decision procedure for the propositional mu-calculus. Inf. Comput. **81**(3), 249–264 (1989)
18. Tsukada, T., Ong, C.-H.L.: Compositional higher-order model checking via ω-regular games over böhm trees. In: Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS 2014, Vienna, Austria, 14–18 July 2014, pp. 78:1–78:10 (2014)
19. Walukiewicz, I.: Pushdown processes: games and model-checking. Inf. Comput. **164**(2), 234–263 (2001)

Semantics and Proof Theory of the Epsilon Calculus

Richard Zach(✉)

Department of Philosophy, University of Calgary, Calgary, Canada
rzach@ucalgary.ca

Abstract. The epsilon operator is a term-forming operator which replaces quantifiers in ordinary predicate logic. The application of this undervalued formalism has been hampered by the absence of well-behaved proof systems on the one hand, and accessible presentations of its theory on the other. One significant early result for the original axiomatic proof system for the ε-calculus is the first epsilon theorem, for which a proof is sketched. The system itself is discussed, also relative to possible semantic interpretations. The problems facing the development of proof-theoretically well-behaved systems are outlined.

1 Introduction

A formalism for logical choice operators has long been available in the form of Hilbert's epsilon calculus. The epsilon calculus is one of the first formal systems of first-order predicate logic. It was introduced in 1921 by David Hilbert [10], who proposed to use it for the formalization and proof theoretical investigation of mathematical systems. In the epsilon calculus, a term-forming operator ε is used, the intuitive meaning of which is an indefinite choice function: $\varepsilon_x A(x)$ is some x which satisfies $A(x)$ if $A(x)$ is satisfied at all, and arbitrary otherwise. Quantifiers can then be defined, e.g., $(\exists x)A(x)$ as $A(\epsilon_x A(x))$.

The epsilon calculus and proof theoretic methods developed for it, such as the so-called epsilon substitution method, have mainly been applied to the proof theoretic analysis of mathematical systems of arithmetic and analysis (especially in work by Ackermann, Mints, Arai). (See [4] for a survey of the epsilon calculus and its history.) Despite its long history and manifold uses, the epsilon calculus as a logical formalism in general is not thoroughly understood, yet its potential for applications in logic and other areas, especially linguistics and computer science, has by far not been fully explored.

There are various options for definitions of semantics of the epsilon operator. The choice of $\varepsilon_x A(x)$ may be extensional (i.e., depend only on the set of x which satisfy $A(x)$; this definition validates the so-called axiom of ε-extensionality), it may be intensional (i.e., depend also on $A(x)$ itself; ε-extensionality fails), and it may be completely indeterministic (i.e., different occurrences of the same ε-term

Richard Zach—Research supported by the Natural Sciences and Engineering Research Council.

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 27–47, 2017.
DOI: 10.1007/978-3-662-54069-5_4

$\varepsilon_x A(x)$ may select different witnesses for $A(x)$). The first and third versions have been investigated by Blass and Gurevich [6]. These different semantics result in different expressive power (in particular, over finite models), and are characterized by different formalizations. Below we present the first two versions of the semantics of the ε-calculus and sketch completeness results.

The very beginnings of proof theory in the work of Hilbert and his students consisted in the proof theoretic study of axiom systems for the ε-calculus. One of the most significant results in this connection are the epsilon theorems. It plays a role similar to Gentzen's midsequent theorem in the proof theory of the sequent calculus: it yields a version of Herbrand's Theorem. In fact, it was used to give the first correct proof of Herbrand's theorem (Hilbert and Bernays [11]). In a simple formulation, the theorem states that if an existential formula $(\exists x)A(x)$ (not containing ε) is derivable in the epsilon calculus, then there are terms $t_1,\ldots,t_n$ so that a (Herbrand-) disjunction $A(t_1) \vee \ldots \vee A(t_n)$ is derivable in propositional logic. The proof gives a constructive procedure that, given a derivation of $(\exists x)A(x)$, produces the corresponding Herbrand disjunction. An analysis of this proof (see [18]) gives a hyper-exponential bound on the length of the Herbrand disjunction in the number of critical formulas occurring in the proof. The bound is essentially optimal, since it is known from work by Orevkov and Statman that the length of Herbrand disjunctions is hyper-exponential in the length of proofs of the original existential formula (this is the basis for familiar speed-up theorems of systems with cut over cut-free systems). In Sect. 4 we prove the first epsilon theorem with identity, along the lines of Bernays's proof.

A general proof theory of the epsilon calculus requires formal systems that are more amenable to proof-theoretic investigations than the Hilbert-type axiomatic systems studied in the Hilbert school. Although some sequent systems for the epsilon calculus exist, it is not clear that they are the best possible formulations, nor have their proof-theoretic properties been investigated in depth. Maehara's [13] and Leisenring's [12] systems were not cut-free complete. Yasuhara [21] studied a cut-free complete system, but only gave a semantic cut-elimination proof. Section 5 surveys these and other systems, and highlights some of the difficulties in developing a systematic proof theory on the basis of them. Proof-theoretically suitable formalisms for the ε-calculus are still a desideratum for applications of the epsilon calculus.

The classical ε-calculus is usually investigated as a proof-theoretic formalism, and no systematic study of the model theory of epsilon calculi other than Asser's classic [3] exists. However, Abiteboul and Vianu [2], Blass and Gurevich [6], and Otto [19] have studied the model theory of choice operators in the context of finite model theory and database query languages. And applications of choice operators to model definite and indefinite noun phrases in computational linguistics Meyer Viol [15] and von Heusinger [8,9] have led to the definition of indexed epsilon calculus by Mints and Sarenac [16].

With a view to applications, it is especially important to develop the semantics and proof theory of epsilon operators in non-classical logics. Of particular importance in this context is the development of epsilon calculi for intuitionistic logic, not least because this is the context in which the epsilon calculus can and

has been applied in programming language semantics. Some work has been done on intuitionistic ε-calculi (e.g., Bell [5], DeVidi [7], Meyer Viol [15], Mints [17]), but there are still many important open questions. The straightforward extensions of intuitionistic logic by epsilon operators are not conservative and result in intermediate logics related to Gödel logic. Meyer Viol [15] has proposed a conservative extensions of intuitionistic logic by epsilon operators which warrants further study.

2 Syntax and Axiomatic Proof Systems

Definition 1. The language of of the elementary calculus $L^{=}_{\mathrm{EC}}$ contains the usual logical symbols (variables, function and predicate symbols, $=$). A subscript ε will indicate the presence of the symbol ε, and $\forall$ the presence of the quantifiers $\forall$ and $\exists$. The *terms* Trm and *formulas* Frm of $L_{\varepsilon\forall}$ are defined as usual, but simultaneously, to include:

> If A is a formula in which x has a free occurrence but no bound occurrence, then $\varepsilon_x A$ is a term, and all occurrences of x in it are bound.

If E is an expression (term or formula), then $\mathrm{FV}(E)$ is the set of variables which have free occurrences in E.

When E, E' are expressions (terms or formulas), we write $E \equiv E'$ iff E and E' are syntactically identical up to a renaming of bound variables. We say that a term t is *free for x in E* iff x does not occur free in the scope of an ε-operator ε_y or quantifier $\forall y$, $\exists y$ for any $y \in \mathrm{FV}(t)$.

If E is an expression and t is a term, we write $E[x/t]$ for the result of substituting every free occurrence of x in E by t, provided t is free for x in E, and renaming bound variables in t if necessary. We write $E(x)$ to indicate that $x \in \mathrm{FV}(E)$, and $E(t)$ for $E[x/t]$. We write $E\{t/u\}$ for the result of replacing every occurrence of t in E by u.[1]

Definition 2 (ε-Translation). If E is an expression, define E^ε by:

1. $E^\varepsilon = E$ if E is a variable, a constant symbol, or $\bot$.
2. If $E = f_i^n(t_1, \ldots, t_n)$, $E^\varepsilon = f_i^n(t_1^\varepsilon, \ldots, t_n^\varepsilon)$.
3. If $E = P_i^n(t_1, \ldots, t_n)$, $E^\varepsilon = P_i^n(t_1^\varepsilon, \ldots, t_n^\varepsilon)$.
4. If $E = \neg A$, then $E^\varepsilon = \neg A^\varepsilon$.
5. If $E = (A \wedge B)$, $(A \vee B)$, $(A \to B)$, or $(A \leftrightarrow B)$, then $E^\varepsilon = (A^\varepsilon \wedge B^\varepsilon)$, $(A^\varepsilon \vee B^\varepsilon)$, $(A^\varepsilon \to B^\varepsilon)$, or $(A^\varepsilon \leftrightarrow B^\varepsilon)$, respectively.

[1] Skipping details, (a) we want to replace not just every occurrence of t by u, but every occurrence of a term $t' \equiv t$. (b) t may have an occurrence in E where a variable in t is bound by a quantifier or ε outside t, and such occurrences shouldn't be replaced (they are not subterm occurrences). (c) When replacing t by u, bound variables in u might have to be renamed to avoid conflicts with the bound variables in E' and bound variables in E' might have to be renamed to avoid free variables in u being bound.

6. If $E = \exists x\, A(x)$ or $\forall x\, A(x)$, then $E^\varepsilon = A^\varepsilon(\varepsilon_x\, A(x)^\varepsilon)$ or $A^\varepsilon(\varepsilon_x \neg A(x)^\varepsilon)$.
7. If $E = \varepsilon_x\, A(x)$, then $E^\varepsilon = \varepsilon_x\, A(x)^\varepsilon$.

Definition 3. An ε-term $p \equiv \varepsilon_x\, B(x; x_1, \ldots, x_n)$ is a *type of an ε-term* $\varepsilon_x\, A(x)$ iff

1. $p \equiv \varepsilon_x\, A(x)[x_1/t_1] \ldots [x_n/t_n]$ for some terms $t_1, \ldots, t_n$.
2. $\mathrm{FV}(p) = \{x_1, \ldots, x_n\}$.
3. $x_1, \ldots, x_n$ are all immediate subterms of p.
4. Each x_i has exactly one occurrence in p.
5. The occurrence of x_i is left of the occurrence of x_j in p if $i < j$.

We denote the set of types as Typ.

Proposition 4. *The type of an epsilon term $\varepsilon_x\, A(x)$ is unique up to renaming of bound, and disjoint renaming of free variables.*

Definition 5. An ε-term e is *nested in* an ε-term e' if e is a proper subterm of e.

Definition 6. The *degree* $\deg(e)$ of an ε-term e is defined as follows: (1) $\deg(e) = 1$ iff e contains no nested ε-terms. (2) $\deg(e) = \max\{\deg(e_1), \ldots, \deg(e_n)\} + 1$ if $e_1, \ldots, e_n$ are all the ε-terms nested in e. For convenience, let $\deg(t) = 0$ if t is not an ε-term.

Definition 7. An ε-term e is *subordinate to* an ε-term $e' = \varepsilon_x\, A(x)$ if some $e'' \equiv e$ occurs in e' and $x \in \mathrm{FV}(e'')$.

Note that if e is subordinate to e' it is *not* a subterm of e', because x is free in e and so the occurrence of e (really, of the variant e'') in e' is in the scope of ε_x.[2]

Definition 8. The *rank* $\mathrm{rk}(e)$ of an ε-term e is defined as follows: (1) $\mathrm{rk}(e) = 1$ iff e contains no subordinate ε-terms. (2) $\mathrm{rk}(e) = \max\{\mathrm{rk}(e_1), \ldots, \mathrm{rk}(e_n)\} + 1$ if $e_1, \ldots, e_n$ are all the ε-terms subordinate to e.

Proposition 9. *If p is the type of e, then* $\mathrm{rk}(p) = \mathrm{rk}(e)$.

2.1 Axioms and Proofs

Definition 10. The axioms of the *elementary calculus* EC are

$$A \qquad \text{for any tautology } A \qquad \text{(Taut)}$$

[2] One might think that replacing e in $\varepsilon_x\, A(x)$ by a new variable y would result in an ε-term $\varepsilon_x\, A'(y)$ so that $e' \equiv \varepsilon_x\, A'(y)[y/e]$. But (a) $\varepsilon_x\, A'(y)$ is not in general a term, since it is not guaranteed that x is free in $A'(y)$ and (b) e is not free for y in $\varepsilon_x\, A'(y)$.

and its only rule of inference is

$$\frac{A \quad A \to B}{A}\ \text{MP}$$

For $\text{EC}^{=}$, we add

$$t = t \qquad \text{for any term } t \qquad (=_1)$$
$$t = u \to (A[x/t] \leftrightarrow A[x/u]). \qquad (=_2)$$

The axioms and rules of the (intensional) *ε-calculus* EC_ε ($\text{EC}_\varepsilon^{=}$) are those of EC ($\text{EC}^{=}$) plus the *critical formulas*

$$A(t) \to A(\varepsilon_x A(x)). \qquad \text{(crit)}$$

The axioms and rules of the *extensional ε-calculus* $\text{EC}_\varepsilon^{\text{ext}}$ are those of $\text{EC}_\varepsilon^{=}$ plus

$$(\forall x(A(x) \leftrightarrow B(x)))^\varepsilon \to \varepsilon_x A(x) = \varepsilon_x B(x), \quad \text{(ext)}$$

that is,

$$A(\varepsilon_x \neg(A(x) \leftrightarrow B(x))) \leftrightarrow B(\varepsilon_x \neg(A(x) \leftrightarrow B(x))) \to \varepsilon_x A(x) = \varepsilon_x B(x)$$

The axioms and rules of $\text{EC}_\forall$, $\text{EC}_{\varepsilon\forall}$, $\text{EC}_{\varepsilon\forall}^{\text{ext}}$ are those of EC, EC_ε, $\text{EC}_\varepsilon^{\text{ext}}$, respectively, together with the axioms

$$A(t) \to \exists x\, A(x) \qquad (\text{Ax}\exists)$$
$$\forall x\, A(x) \to A(t) \qquad (\text{Ax}\forall)$$

and the rules

$$\frac{A(x) \to B}{\exists x\, A(x) \to B}\ R\exists \qquad \frac{B \to A(x)}{B \to \forall x\, A(x)}\ R\forall$$

Applications of these rules must satisfy the *eigenvariable condition*, viz., the variable x must not appear in the conclusion or anywhere below it in the proof.

Definition 11. If Γ is a set of formulas, a *proof of A from Γ in* $\text{EC}_{\varepsilon\forall}^{\text{ext}}$ is a sequence π of formulas $A_1, \ldots, A_n = A$ where for each $i \leq n$, $A_i \in \Gamma$, A_i is an instance of an axiom, or follows from formulas A_j $(j < i)$ by a rule of inference.

If π only uses the axioms and rules of EC, EC_ε, $\text{EC}_\varepsilon^{\text{ext}}$, etc., then it is a proof of A from Γ in EC, EC_ε, $\text{EC}_\varepsilon^{\text{ext}}$, etc., and we write $\Gamma \vdash^\pi A$, $\Gamma \vdash_\varepsilon^\pi A$, $\Gamma \vdash_{\varepsilon\text{ext}}^\pi A$, etc.

We say that A is provable from Γ in EC, etc. ($\Gamma \vdash A$, etc.), if there is a proof of A from Γ in EC, etc.

Note that our definition of proof, because of its use of $\equiv$, includes a tacit rule for renaming bound variables. Note also that substitution into members of Γ is *not* permitted. However, we can simulate a provability relation in which substitution into members of Γ is allowed by considering Γ^{inst}, the set of all substitution instances of members of Γ. If Γ is a set of sentences, then $\Gamma^{\text{inst}} = \Gamma$.

Proposition 12. *If $\pi = A_1, \ldots, A_n \equiv A$ is a proof of A from Γ and $x \notin \mathrm{FV}(\Gamma)$ is not an eigenvariable in π, then $\pi[x/t] = A_1[x/t], \ldots, A_n[x/t]$ is a proof of $A[x/t]$ from Γ^{inst}.*

Lemma 13. *If π is a proof of B from $\Gamma \cup \{A\}$, then there is a proof $\pi[A]$ of $A \to B$ from Γ, provided A contains no eigenvariables of π free.*

Proof. By induction on the length of π, as in the classical case.

Theorem 14 (Deduction Theorem). *If $\Sigma \cup \{A\}$ is a set of sentences, $\Sigma \vdash A \to B$ iff $\Sigma \cup \{A\} \vdash B$.*

Corollary 15. *If $\Sigma \cup \{A\}$ is a set of sentences, $\Sigma \vdash A$ iff $\Sigma \cup \{\neg A\} \vdash \bot$.*

Lemma 16 (ε-Embedding Lemma). *If $\Gamma \vdash^{\pi}_{\varepsilon\forall} A$, then there is a proof π^{ε} so that $\Gamma^{\varepsilon\, inst} \vdash^{\pi^{\varepsilon}}_{\varepsilon} A^{\varepsilon}$.*

Proof. By induction, see [18].

3 Semantics and Completeness

3.1 Semantics for $\mathbf{EC}^{\mathbf{ext}}_{\boldsymbol{\varepsilon}}$

Definition 17. A *structure* $\mathfrak{M} = \langle |\mathfrak{M}|, (\cdot)^{\mathfrak{M}} \rangle$ consists of a nonempty *domain* $|\mathfrak{M}| \neq \emptyset$ and a mapping $(\cdot)^{\mathfrak{M}}$ on function and predicate symbols where $(f^0_i)^{\mathfrak{M}} \in |\mathfrak{M}|$, $(f^n_i)^M \in |\mathfrak{M}|^{|\mathfrak{M}|^n}$, and $(P^n_i)^{\mathfrak{M}} \subseteq |\mathfrak{M}|^n$.

Definition 18. An *extensional choice function* Φ *on* $\mathfrak{M}$ is a function $\Phi\colon \wp(|\mathfrak{M}|) \to |\mathfrak{M}|$ where $\Phi(X) \in X$ whenever $X \neq \emptyset$.

Note that Φ is total on $\wp(|\mathfrak{M}|)$, and so $\Phi(\emptyset) \in |\mathfrak{M}|$.

Definition 19. An *assignment* s *on* $\mathfrak{M}$ is a function $s\colon \mathrm{Var} \to |\mathfrak{M}|$.

If $x \in \mathrm{Var}$ and $m \in |\mathfrak{M}|$, $s[x/m]$ is the assignment defined by

$$s[x/m](y) = \begin{cases} m & \text{if } y = x \\ s(y) & \text{otherwise} \end{cases}$$

Definition 20. The *value* $\mathrm{val}_{\mathfrak{M},\Phi,s}(t)$ *of a term* and the *satisfaction relation* $\mathfrak{M}, \Phi, s \models A$ are defined as follows:

1. $\mathrm{val}_{\mathfrak{M},\Phi,s}(x) = s(x)$
2. $\mathfrak{M}, \Phi, s \models \top$ and $\mathfrak{M}, \Phi, s \not\models \bot$

3. $\mathrm{val}_{\mathfrak{M},\Phi,s}(f_i^n(t_1,\ldots,t_n)) = (f_i^n)^{\mathfrak{M}}(\mathrm{val}_{\mathfrak{M},\Phi,s}(t_1),\ldots,\mathrm{val}_{\mathfrak{M},\Phi,s}(t_n))$
4. $\mathfrak{M},\Phi,s \models t_1 = t_n$ iff $\mathrm{val}_{\mathfrak{M},\Phi,s}(t_1) = \mathrm{val}_{\mathfrak{M},\Phi,s}(t_2)$
5. $\mathfrak{M},\Phi,s \models P_i^n(t_1,\ldots,t_n)$ iff $\langle \mathrm{val}_{\mathfrak{M},\Phi,s}(t_1),\ldots,\mathrm{val}_{\mathfrak{M},\Phi,s}(t_n)\rangle \in (P_i^n)^{\mathfrak{M}}$
6. $\mathrm{val}_{\mathfrak{M},\Phi,s}(\varepsilon_x A(x)) = \Phi(\mathrm{val}_{\mathfrak{M},\Phi,s}(A(x)))$ where

$$\mathrm{val}_{\mathfrak{M},\Phi,s}(A(x)) = \{m \in |\mathfrak{M}| : \mathfrak{M},\Phi,s[x/m] \models A(x)\}$$

7. $\mathfrak{M},\Phi,s \models \exists x\, A(x)$ iff for some $m \in |\mathfrak{M}|$, $\mathfrak{M},\Phi,s[x/m] \models A(x)$
8. $\mathfrak{M},\Phi,s \models \forall x\, A(x)$ iff for all $m \in |\mathfrak{M}|$, $\mathfrak{M},\Phi,s[x/m] \models A(x)$

Proposition 21. *If $s(x) = s'(x)$ for all $x \notin \mathrm{FV}(t) \cup \mathrm{FV}(A)$, then $\mathrm{val}_{\mathfrak{M},\Phi,s}(t) = \mathrm{val}_{\mathfrak{M},\Phi,s'}(t)$ and $\mathfrak{M},\Phi,s \models A$ iff $\mathfrak{M},\Phi,s' \models A$.*

Proposition 22 (Substitution Lemma). *If $m = \mathrm{val}_{\mathfrak{M},\Phi,s}(u)$, then $\mathrm{val}_{\mathfrak{M},\Phi,s}(t(u)) = \mathrm{val}_{\mathfrak{M},\Phi,s[x/m]}(t(x))$ and $\mathfrak{M},\Phi,s \models A(u)$ iff $\mathfrak{M},\Phi,s[x/m] \models A(x)$.*

Definition 23. 1. A is *locally true* in $\mathfrak{M}$ w.r.t. Φ and s iff $\mathfrak{M},\Phi,s \models A$.
2. A is *true* in $\mathfrak{M}$ with respect to Φ, $\mathfrak{M},\Phi \models A$, iff for all s on $\mathfrak{M}$: $\mathfrak{M},\Phi,s \models A$.
3. A is *generically true* in $\mathfrak{M}$ with respect to s, $\mathfrak{M},s \models^g A$, iff for all choice functions Φ on $\mathfrak{M}$: $\mathfrak{M},\Phi,s \models A$.
4. A is *generically valid* in $\mathfrak{M}$, $\mathfrak{M} \models A$, if for all choice functions Φ and assignments s on $\mathfrak{M}$: $\mathfrak{M},\Phi,s \models A$.

Definition 24. Let $\Gamma \cup \{A\}$ be a set of formulas.

1. A is a *local consequence* of Γ, $\Gamma \models^l A$, iff for all $\mathfrak{M}$, Φ, and s: if $\mathfrak{M},\Phi,s \models \Gamma$ then $\mathfrak{M},\Phi,s \models A$.
2. A is a *truth consequence* of Γ, $\Gamma \models A$, iff for all $\mathfrak{M}$, Φ: if $\mathfrak{M},\Phi \models \Gamma$ then $\mathfrak{M},\Phi \models A$.
3. A is a *generic consequence* of Γ, $\Gamma \models^g A$, iff for all $\mathfrak{M}$ and s: if $\mathfrak{M},s \models^g \Gamma$ then $\mathfrak{M} \models A$.
4. A is a *generic validity consequence* of Γ, $\Gamma \models^v A$, iff for all $\mathfrak{M}$: if $\mathfrak{M} \models^v \Gamma$ then $\mathfrak{M} \models A$.

Proposition 25. *If $\Sigma \cup \{A\}$ is a set of sentences, $\Sigma \models^l A$ iff $\Sigma \models A$*

Proposition 26. *If $\Sigma \cup \{A,B\}$ is a set of sentences, $\Sigma \cup \{A\} \models B$ iff $\Sigma \models A \to B$.*

Corollary 27. *If $\Sigma \cup \{A\}$ is a set of sentences, $\Sigma \models A$ iff for no $\mathfrak{M}$, Φ, $\mathfrak{M} \models \Sigma \cup \{\neg A\}$*

3.2 Soundness and Completeness

Theorem 28. *If $\Gamma \vdash_\varepsilon A$, then $\Gamma \models^l A$.*

Proof. Suppose $\Gamma, \Phi, s \models \Gamma$. We show by induction on the length n of a proof π that $\mathfrak{M}, \Phi, s' \models A$ for all s' which agree with s on $\mathrm{FV}(\Gamma)$. We may assume that no eigenvariable x of π is in $\mathrm{FV}(\Gamma)$ (if it is, let $y \notin \mathrm{FV}(\pi)$ and not occurring in π; consider $\pi[x/y]$ instead of π).

If $n = 0$ there's nothing to prove. Otherwise, we distinguish cases according to the last line A_n in π. The only interesting case is when A_n is a critical formula, i.e., $A_n \equiv A(t) \to A(\varepsilon_x A(x))$. Then either $\mathfrak{M}, \Phi, s \models A(t)$ or not (in which case there's nothing to prove). If yes, $\mathfrak{M}, \Phi, s[x/m] \models A(x)$ for $m = \mathrm{val}_{\mathfrak{M},\Phi,s}(t)$, and so $Y = \mathrm{val}_{\mathfrak{M},\Phi,s}(A(x)) \neq \emptyset$. Consequently, $\Phi(Y) \in Y$, and hence $\mathfrak{M}, \Phi, s \models A(\varepsilon_x A(x))$.

Lemma 29. *If Γ is a set of sentences and $\Gamma \nvdash_\varepsilon \bot$, then there are $\mathfrak{M}$, Φ so that $\mathfrak{M}, \Phi \models \Gamma$.*

Theorem 30 (Completeness). *If $\Gamma \cup \{A\}$ are sentences and $\Gamma \models A$, then $\Gamma \vdash_{\varepsilon\mathrm{ext}} A$.*

Proof. Suppose $\Gamma \not\models A$. Then for some $\mathfrak{M}$, Φ we have $\mathfrak{M}, \Phi \models \Gamma$ but $\mathfrak{M}, \Phi \not\models A$. Hence $\mathfrak{M}, \Phi \models \Gamma \cup \{\neg A\}$. By the Lemma, $\Gamma \cup \{\neg A\} \vdash_\varepsilon \bot$. By Corollary 15, $\Gamma \vdash_\varepsilon A$.

The proof of the Lemma comes in several stages. We have to show that if Γ is consistent, we can construct $\mathfrak{M}$, Φ, and s so that $\mathfrak{M}, \Phi, s \models \Gamma$. Since $\mathrm{FV}(\Gamma) = \emptyset$, we then have $\mathfrak{M}, \Phi \models \Gamma$.

Lemma 31. *If $\Gamma \nvdash_\varepsilon \bot$, there is $\Gamma^* \supseteq \Gamma$ with (1) $\Gamma^* \nvdash_\varepsilon \bot$ and (2) for all formulas A, either $A \in \Gamma^*$ or $\neg A \in \Gamma^*$.*

Proof. Let $A_1, A_2, \ldots$ be an enumeration of Frm_ε. Define $\Gamma_0 = \Gamma$ and

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{A_n\} & \text{if } \Gamma_n \cup \{A_n\} \nvdash_\varepsilon \bot \\ \Gamma_n \cup \{\neg A_n\} & \text{if } \Gamma_n \cup \{\neg A_n\} \nvdash_\varepsilon \bot \text{ otherwise} \end{cases}$$

Let $\Gamma^* = \bigcup_{n \geq 0} \Gamma_n$. Obviously, $\Gamma \subseteq \Gamma^*$. For (1), observe that if $\Gamma^* \vdash_\varepsilon^\pi \bot$, then π contains only finitely many formulas from Γ^*, so for some n, $\Gamma_n \vdash_\varepsilon^\pi \bot$. But Γ_n is consistent by definition.

To verify (2), we have to show that for each n, either $\Gamma_n \cup \{A_n\} \nvdash_\varepsilon \bot$ or $\Gamma_n \cup \{\neg A\} \nvdash_\varepsilon \bot$. For $n = 0$, this is the assumption of the lemma. So suppose the claim holds for $n - 1$. Suppose $\Gamma_n \cup \{A\} \vdash_\varepsilon^\pi \bot$ and $\Gamma_n \cup \{\neg A\} \vdash_\varepsilon^{\pi'} \bot$. Then by the Deduction Theorem, we have $\Gamma_n \vdash_A^{\pi[A]} \to \bot$ and $\Gamma_n \vdash_\neg^{\pi'[A']} A \to \bot$. Since $(A \to \bot) \to ((\neg A \to \bot) \to \bot)$ is a tautology, we have $\Gamma_n \vdash_\varepsilon \bot$, contradicting the induction hypothesis.

Lemma 32. *If $\Gamma^* \vdash_\varepsilon B$, then $B \in \Gamma^*$.*

Proof. If not, then $\neg B \in \Gamma^*$ by maximality, so Γ^* would be inconsistent.

Definition 33. Let $\approx$ be the relation on Trm_ε defined by

$$t \approx u \text{ iff } t = u \in \Gamma^*$$

It is easily seen that $\approx$ is an equivalence relation. Let $\widetilde{t} = \{u : u \approx t\}$ and $\widetilde{\mathrm{Trm}} = \{\widetilde{t} : t \in \mathrm{Trm}\}$.

Definition 34. A set $T \in \widetilde{\mathrm{Trm}}$ is *represented by* $A(x)$ if $T = \{\widetilde{t} : A(t) \in \Gamma^*\}$.

Let Φ_0 be a fixed choice function on $\widetilde{\mathrm{Trm}}$, and define

$$\Phi(T) = \begin{cases} \widetilde{\varepsilon_x A(x)} & \text{if } T \text{ is represented by } A(x) \\ \Phi_0(T) & \text{otherwise.} \end{cases}$$

Proposition 35. *Φ is a well-defined choice function on $\widetilde{\mathrm{Trm}}$.*

Proof. Use (ext) for well-definedness and (crit) for choice function.

Now let $\mathfrak{M} = \langle \widetilde{\mathrm{Trm}}, (\cdot)^{\mathfrak{M}} \rangle$ with $c^{\mathfrak{M}} = \widetilde{c}$, $(P_i^n)^{\mathfrak{M}} = \{\langle \widetilde{t_1}, \ldots, \widetilde{t_1} \rangle : P_i^n(t_1, \ldots, t_n)\}$, and let $s(x) = \widetilde{s}$.

Proposition 36. $\mathfrak{M}, \Phi, s \models \Gamma^*$.

Proof. We show that $\mathrm{val}_{\mathfrak{M},\Phi,s}(t) = \widetilde{t}$ and $\mathfrak{M}, \Phi, s \models A$ iff $A \in \Gamma^*$ by simultaneous induction on the complexity of t and A.

If $t = c$ is a constant, the claim holds by definition of $(\cdot)^{\mathfrak{M}}$. If $A = \bot$ or $= \top$, the claim holds by Lemma 32.

If $A \equiv P^n(t_1, \ldots, t_n)$, then by induction hypothesis, $\mathrm{val}_{\mathfrak{M},\Phi,s}(t)_i = \widetilde{t_i}$. By definition of $(\cdot)^{\mathfrak{M}}$, $\langle \widetilde{t_1}, \ldots, \widetilde{t_n} \rangle \in (P_i^n)(t_1, \ldots, t_n)$ iff $P_i^n(t_1, \ldots, t_n) \in \Gamma^*$.

If $A \equiv \neg B$, $(B \wedge C)$, $(B \vee C)$, $(B \to C)$, $(B \leftrightarrow C)$, the claim follows immediately from the induction hypothesis and the definition of $\models$ and the closure properties of Γ^*. For instance, $\mathfrak{M}, \Phi, s \models (B \wedge C)$ iff $\mathfrak{M}, \Phi, s \models B$ and $\mathfrak{M}, \Phi, s \models C$. By induction hypothesis, this is the case iff $B \in \Gamma^*$ and $C \in \Gamma^*$. But since $B, C \vdash_\varepsilon B \wedge C$ and $B \wedge C \vdash_\varepsilon B$ and $\vdash_\varepsilon C$, this is the case iff $(B \wedge C) \in \Gamma^*$. Remaining cases: Exercise.

If $t \equiv \varepsilon_x A(x)$, then $\mathrm{val}_{\mathfrak{M},\Phi,s}(t) = \Phi(\mathrm{val}_{\mathfrak{M},\Phi,s}(A(x)))$. Since $\widetilde{\mathrm{val}_{\mathfrak{M},\Phi,s}(A(x))}$ is represented by $A(x)$ by induction hypothesis, we have $\mathrm{val}_{\mathfrak{M},\Phi,s}(t) = \widetilde{\varepsilon_x A(x)}$ by definition of Φ.

3.3 Semantics for EC_ε

In order to give a complete semantics for EC_ε, i.e., for the calculus without the extensionality axiom (ext), it is necessary to change the notion of choice function so that two ε-terms $\varepsilon_x A(x)$ and $\varepsilon_x B(x)$ may be assigned different representatives even when $\mathfrak{M}, \Phi, s \models \forall x(A(x) \leftrightarrow B(x))$, since then the negation of (ext) is consistent in the resulting calculus. The idea is to add the ε-term itself as an additional argument to the choice function. However, in order for this semantics to be sound for the calculus—specifically, in order for $(=_2)$ to be valid—we have to use not ε-terms but ε-types.

Definition 37. An *intensional choice operator* is a mapping $\Psi\colon \mathrm{Typ}\times|\mathfrak{M}|^{<\omega} \to |\mathfrak{M}|^{\wp(|\mathfrak{M}|)}$ such that for every type $p = \varepsilon_x A(x; y_1, \ldots, y_n)$ is a type, and m_1, $\ldots$, $m_n \in |\mathfrak{M}|$, $\Psi(p, m_1, \ldots, m_n)$ is a choice function.

Definition 38. If $\mathfrak{M}$ is a structure, Ψ an intensional choice operator, and s an assignment, $\mathrm{val}_{\mathfrak{M},\Psi,s}(t)$ and $\mathfrak{M}, \Psi, s \models A$ is defined as before, except (6) in Definition 20 is replaced by:

(6′) $\mathrm{val}_{\mathfrak{M},\Psi,s}(\varepsilon_x A(x)) = \Psi(p, m_1, \ldots, m_n)(\mathrm{val}_{\mathfrak{M},\Phi,s}(A(x)))$ where

(a) $p = \varepsilon_x A'(x; x_1, \ldots, x_n)$ is the type of $\varepsilon_x A(x)$,
(b) t_1, $\ldots$, t_n are the subterms corresponding to x_1, $\ldots$, x_n, i.e., $\varepsilon_x A(x) \equiv \varepsilon_x A'(x; t_1, \ldots, t_n)$,
(c) $m_i = \mathrm{val}_{\mathfrak{M},\Psi,s}(t)_1$, and
(d) $\mathrm{val}_{\mathfrak{M},\Phi,s}(A(x)) = \{m \in |\mathfrak{M}| : \mathfrak{M}, \Psi, s[x/m] \models A(x)\}$

The soundness and completeness proofs generalize to EC_ε, $\mathrm{EC}_\varepsilon^=$, and $\mathrm{EC}_{\varepsilon\forall}$.

4 The First Epsilon Theorem

4.1 The Case Without Identity

Definition 39. An ε-term e is *critical in* π if $A(t) \to A(e)$ is one of the critical formulas in π. The *rank* $\mathrm{rk}(\pi)$ *of a proof* π is the maximal rank of its critical ε-terms. The *r-degree* $\deg(\pi, r)$ of π is the maximum degree of its critical ε-terms of rank r. The *r-order* $o(\pi, r)$ of π is the number of different (up to renaming of bound variables) critical ε-terms of rank r.

Lemma 40. *If $e = \varepsilon_x A(x)$, $\varepsilon_y B(y)$ are critical in π, $\mathrm{rk}(e) = \mathrm{rk}(\pi)$, and $B^* \equiv B(u) \to B(\varepsilon_y B(y))$ is a critical formula in π. Then, if e is a subterm of B^*, it is a subterm of $B(y)$ or a subterm of u.*

Proof. Suppose not. Since e is a subterm of B^*, we have $B(y) \equiv B'(\varepsilon_x A'(x, y), y)$ and either $e \equiv \varepsilon_x A'(x, u)$ or $e \equiv \varepsilon_x A'(x, \varepsilon_y B(y))$. In each case, we see that $\varepsilon_x A'(x, y)$ and e have the same rank, since the latter is an instance of the former (and so have the same type). On the other hand, in either case, $\varepsilon_y B(y)$ would be

$$\varepsilon_y B'(\varepsilon_x A'(x, y), y)$$

and so would have a higher rank than $\varepsilon_x A'(x, y)$ as that ε-term is subordinate to it. This contradicts $\mathrm{rk}(e) = \mathrm{rk}(\pi)$.

Lemma 41. *Let e, B^* be as in the lemma, and t be any term. Then*

1. *If e is not a subterm of $B(y)$, $B^*\{e/t\} \equiv B(u') \to B(\varepsilon_y B(y))$.*
2. *If e is a subterm of $B(y)$, i.e., $B(y) \equiv B'(e, y)$, $B^*\{e/t\} \equiv B'(t, u') \to B'(t, \varepsilon_y B'(t, y))$.*

Lemma 42. *If* $\vdash_\varepsilon^\pi E$ *and* E *does not contain* ε*, then there is a proof* π' *such that* $\vdash_\varepsilon^{\pi'} E$ *and* $\mathrm{rk}(\pi') \leq \mathrm{rk}(pi) = r$ *and* $o(\pi', r) < o(\pi, r)$*.*

Proof. Let e be an ε-term critical in π and let $A(t_1) \to A(e)$, dots, $A(t_n) \to A(e)$ be all its critical formulas in π.

Consider $\pi\{e/t\}_i$, i.e., π with e replaced by t_i throughout. Each critical formula belonging to e now is of the form $A(t'_j) \to A(t_i)$, since e obviously cannot be a subterm of $A(x)$ (if it were, e would be a subterm of $\varepsilon_x A(x)$, i.e., of itself!). Let $\hat{\pi}_i$ be the sequence of tautologies $A(t_i) \to (A(t'_j) \to A(t_i))$ for $i = 1, \ldots, n$, followed by $\pi\{e/t\}_i$. Each one of the formulas $A(t'_j) \to A(t_i)$ follows from one of these by (MP) from $A(t_i)$. Hence, $A(t_i) \vdash_\varepsilon^{\hat{\pi}_i} E$. Let $\pi_i = \hat{\pi}_i[A_i]$ as in Lemma 13. We have $\vdash_\varepsilon^{\pi_i} A_i \to E$.

The ε-term e is not critical in π_i: Its original critical formulas are replaced by $A(t_i) \to (A(t'_j) \to A(t_i))$, which are tautologies. By (1) of the preceding Lemma, no critical ε-term of rank r was changed at all. By (2) of the preceding Lemma, no critical ε-term of rank $< r$ was replaced by a critical ε-term of rank $\geq r$. Hence, $o(\pi_i, r) = o(\pi) - 1$.

Let π'' be the sequence of tautologies $\neg \bigvee_{i=1}^n A(t_i) \to (A(t_i) \to A(e))$ followed by π. Then $\bigvee_{i=1}^n A(t_i) \vdash_E^{\pi''}$, e is not critical in π'', and otherwise π and π'' have the same critical formulas. The same goes for $\pi''[\neg \bigvee A(t_i)]$, a proof of $\neg \bigvee A(t_i) \to E$.

We now obtain π' as the π_i, $i = 1, \ldots, n$, followed by $\pi[\neg \bigvee_{i=1}^n A(t_i)]$, followed by the tautology

$$(\neg \bigvee A(t_i) \to E) \to (A(t_1) \to E) \to \cdots \to (A(t_n) \to E) \to E) \ldots)$$

from which E follows by $n + 1$ applications of (MP).

Theorem 43 (First Epsilon Theorem for EC_ε**).** *If* E *is a formula not containing any* ε*-terms and* $\vdash_\varepsilon E$*, then* $\vdash_\varepsilon E$*.*

Proof. By induction on $o(\pi, r)$, we have: if $\vdash_\varepsilon^\pi E$, then there is a proof π^* of E with $\mathrm{rk}(\pi^-) < r$. By induction on $\mathrm{rk}(()\pi)$ we have a proof π^{**} of E with $\mathrm{rk}(\pi^{**}) = 0$, i.e., without critical formulas at all.

Corollary 44 (Extended First ε**-Theorem).** *If* $\vdash_\varepsilon E(e_1, \ldots, e_n)$*, then* $\vdash_{\bigvee_{i=1}^m} E(t_1^j, \ldots, t_n^j)$ *for some terms* t_j *(in* EC*).*

Proof. If E contains ε-terms, say, E is $E(e_1, \ldots, e_n)$, then replacement of ε-terms in the construction of π_i may change E—but of course only the ε-terms appearing as subterms in it. In each step we obtain not a proof of E but of some disjunction of instances $E(e'_1, \ldots, e'_n)$. For details, see [18].

4.2 The Case with Identity

In the presence of the identity (=) predicate in the language, things get a bit more complicated. The reason is that instances of the $(=_2)$ axiom schema,

$$t = u \to (A(t) \to A(u))$$

may also contain ε-terms, and the replacement of an ε-term e by a term t_i in the construction of π_i may result in a formula which no longer is an instance of $(=_2)$. For instance, suppose that t is a subterm of $e = e'(t)$ and $A(t)$ is of the form $A'(e'(t))$. Then the original axiom is

$$t = u \to (A'(e'(t)) \to A'(e'(u))$$

which after replacing $e = e'(t)$ by t_i turns into

$$t = u \to (A'(t_i) \to A'(e'(u)).$$

So this must be avoided. In order to do this, we first observe that just as in the case of the predicate calculus, the instances of $(=_2)$ can be derived from restricted instances. In the case of the predicate calculus, the restricted axioms are

$$t = u \to (P^n(s_1, \ldots, t, \ldots s_n) \to P^n(s_1, \ldots, u, \ldots, s_n) \qquad (=_2')$$
$$t = u \to f^n(s_1, \ldots, t, \ldots, s_n) = f^n(s_1, \ldots, u, \ldots, s_n) \qquad (=_2'')$$

to which we have to add the *ε-identity axiom schema:*

$$t = u \to \varepsilon_x A(x; s_1, \ldots, t, \ldots s_n) = \varepsilon_x A(x; s_1, \ldots, u, \ldots s_n) \qquad (=_\varepsilon)$$

where $\varepsilon_x A(x; x_1, \ldots, x_n)$ is an ε-type.

Proposition 45. *Every instance of* $(=_2)$ *can be derived from* $(=_2')$, $(=_2'')$, *and* $(=_\varepsilon)$.

Proof. By induction.

Now replacing every occurrence of e in an instance of $(=_2')$ or $(=_2'')$—where e obviously can only occur inside one of the terms t, u, s_1, …, s_n—results in a (different) instance of $(=_2')$ or $(=_2'')$. The same is true of $(=_\varepsilon)$, *provided that* the e is neither $\varepsilon_x A(x; s_1, \ldots, t, \ldots s_n)$ nor $\varepsilon_x A(x; s_1, \ldots, u, \ldots s_n)$. This would be guaranteed if the type of e is not $\varepsilon_x A(x; x_1, \ldots, x_n)$, in particular, if the rank of e is higher than the rank of $\varepsilon_x A(x; x_1, \ldots, x_n)$. Moreover, the result of replacing e by t_i in any such instance of $(=_\varepsilon)$ results in an instance of $(=_\varepsilon)$ which belongs to the same ε-type. Thus, in order for the proof of the first ε-theorem to work also when $=$ and axioms $(=_1)$, $(=_2')$, $(=_2'')$, and $(=_\varepsilon)$ are present, it suffices to show that the instances of $(=_\varepsilon)$ with ε-terms of rank $\mathrm{rk}(\pi)$ can be removed. Call an ε-term e *special* in π, if π contains an occurrence of $t = u \to e' = e$ as an instance of $(=_\varepsilon)$.

Theorem 46. *If* $\vdash^{\pi}_{\varepsilon=} E$, *then there is a proof* $\pi^=$ *so that* $\vdash^{\pi^=}_{\varepsilon=} E$, $\mathrm{rk}(\pi^=) = \mathrm{rk}(pi)$, *and the rank of the special* ε*-terms in* $\pi^=$ *has rank* $< \mathrm{rk}(\pi)$.

Proof. The basic idea is simple: Suppose $t = u \to e' = e$ is an instance of $(=_\varepsilon)$, with $e' \equiv \varepsilon_x A(x; s_1, \ldots, t, \ldots s_n)$ and $e \equiv \varepsilon_x A(x; s_1, \ldots, u, \ldots s_n)$. Replace e

everywhere in the proof by e'. Then the instance of $(=_\varepsilon)$ under consideration is removed, since it is now provable from $e' = e'$. This potentially interferes with critical formulas belonging to e, but this can also be fixed: we just have to show that by a judicious choice of e it can be done in such a way that the other $(=_\varepsilon)$ axioms are still of the required form.

Let $p = \varepsilon_x A(x; x_1, \ldots, x_n)$ be an ε-type of rank $\mathrm{rk}(\pi)$, and let $e_1, \ldots, e_l$ be all the ε-terms of type p which have a corresponding instance of $(=_\varepsilon)$ in π. Let T_i be the set of all immediate subterms of $e_1, \ldots, e_l$, in the same position as x_i, i.e., the smallest set of terms so that if $e_i \equiv \varepsilon_x A(x; t_1, \ldots, t_n)$, then $t_i \in T$. Now let let T^* be all instances of p with terms from T_i substituted for the x_i. Obviously, T and thus T^* are finite (up to renaming of bound variables). Pick a strict order $\prec$ on T which respects degree, i.e., if $\deg(t) < \deg(u)$ then $t \prec u$. Extend $\prec$ to T^* by

$$\varepsilon_x A(x; t_1, \ldots, t_n) \prec \varepsilon_x A(x; t'_1, \ldots, t'_n)$$

iff

1. $\max\{\deg(t_i) : i = 1, \ldots, n\} < \max\{\deg(t_i) : i = 1, \ldots, n\}$ or
2. $\max\{\deg(t_i) : i = 1, \ldots, n\} = \max\{\deg(t_i) : i = 1, \ldots, n\}$ and
   (a) $t_i \equiv t'_i$ for $i = 1, \ldots, k$.
   (b) $t_{k+1} \prec t'_{k+1}$

Lemma 47. *Suppose $\vdash^{\pi}_{\varepsilon=} E$, e a special ε-term in π with $\mathrm{rk}(e) = \mathrm{rk}(\pi)$, $\deg(e)$ maximal among the special ε-terms of rank $\mathrm{rk}(\pi)$, and e maximal with respect to $\prec$ defined above. Let $t = u \to e' = e$ be an instance of $(=_\varepsilon)$ in π. Then there is a proof π', $\vdash^{\pi'}_{\varepsilon=} E$ such that*

1. $\mathrm{rk}(\pi') = \mathrm{rk}(\pi)$
2. *π' does not contain $t = u \to e' = e$ as an axiom*
3. *Every special ε-term e'' of π' with the same type as e is so that $e'' \prec e$.*

Proof. Let $\pi_0 = \pi\{e/e'\}$ and suppose $t' = u' \to e''' = e''$ is an $(=_\varepsilon)$ axiom in π.

If $\mathrm{rk}(e'') < \mathrm{rk}(e)$, then the replacement of e by e' can only change subterms of e'' and e'''. In this case, the uniform replacement results in another instance of $(=_\varepsilon)$ with ε-terms of the same ε-type, and hence of the same rank $< \mathrm{rk}(\pi)$, as the original.

If $\mathrm{rk}(e'') = \mathrm{rk}(e)$ but has a different type than e, then this axiom is unchanged in π_0: Neither e'' nor e''' can be $\equiv e$, because they have different ε-types, and neither e'' nor e''' (nor t' or u', which are subterms of e'', e''') can contain e as a subterm, since then e wouldn't be degree-maximal among the special ε-terms of π of rank $\mathrm{rk}(\pi)$.

If the type of e'', e''' is the same as that of e, e cannot be a proper subterm of e'' or e''', since otherwise e'' or e''' would again be a special ε-term of rank $\mathrm{rk}(\pi)$ but of higher degree than e. So either $e \equiv e''$ or $e \equiv e'''$, without loss of generality suppose $e \equiv e''$. Then the $(=_\varepsilon)$ axiom in question has the form

$$t' = u' \to \underbrace{\varepsilon_x A(x; s_1, \ldots t', \ldots s_n)}_{e'''} = \underbrace{\varepsilon_x A(x; s_1, \ldots u', \ldots s_n)}_{e'' \equiv e}$$

and with e replaced by e':

$$t' = u' \to \underbrace{\varepsilon_x\, A(x; s_1, \dots t', \dots s_n)}_{e'''} = \underbrace{\varepsilon_x\, A(x; s_1, \dots t, \dots s_n)}_{e'}$$

which is no longer an instance of $(=_\varepsilon)$, but can be proved from new instances of $(=_\varepsilon)$. We have to distinguish two cases according to whether the indicated position of t and t' in e', e''' is the same or not. In the first case, $u \equiv u'$, and the new formula

$$t' = u \to \underbrace{\varepsilon_x\, A(x; s_1, \dots t', \dots s_n)}_{e'''} = \underbrace{\varepsilon_x\, A(x; s_1, \dots t, \dots s_n)}_{e'}$$

can be proved from $t = u$ together with

$$t' = t \to \underbrace{\varepsilon_x\, A(x; s_1, \dots t', \dots s_n)}_{e'''} = \underbrace{\varepsilon_x\, A(x; s_1, \dots t, \dots s_n)}_{e'} \qquad (=_\varepsilon)$$

$$t = u \to (t' = u \to t' = t) \qquad (='_2)$$

Since e' and e''' already occurred in π, by assumption $e', e''' \prec e$.
In the second case, the original formulas read, with terms indicated:

$$t = u \to \underbrace{\varepsilon_x\, A(x; s_1, \dots t, \dots, u', \dots, s_n)}_{e'} = \underbrace{\varepsilon_x\, A(x; s_1, \dots u, \dots, u', \dots, s_n)}_{e}$$

$$t' = u' \to \underbrace{\varepsilon_x\, A(x; s_1, \dots u, \dots, t', \dots, s_n)}_{e'''} = \underbrace{\varepsilon_x\, A(x; s_1, \dots u, \dots, u', \dots, s_n)}_{e'' \equiv e}$$

and with e replaced by e' the latter becomes:

$$t' = u' \to \underbrace{\varepsilon_x\, A(x; s_1, \dots u, \dots, t', \dots s_n)}_{e'''} = \underbrace{\varepsilon_x\, A(x; s_1, \dots t, \dots, u', \dots, s_n)}_{e'}$$

This new formula is provable from $t = u$ together with

$$u = t \to \underbrace{\varepsilon_x\, A(x; s_1, \dots u, \dots, t', \dots s_n)}_{e'''} = \underbrace{\varepsilon_x\, A(x; s_1, \dots t, \dots, t', \dots, s_n)}_{e''''}$$

$$t' = u' \to \underbrace{\varepsilon_x\, A(x; s_1, \dots t, \dots, t', \dots s_n)}_{e''''} = \underbrace{\varepsilon_x\, A(x; s_1, \dots t, \dots, u', \dots, s_n)}_{e'}$$

and some instances of $(='_2)$. Hence, π' contains a (possibly new) special ε-term e''''. However, $e'''' \prec e$.

In the special case where $e = e''$ and $e' = e'''$, i.e., the instance of $(=_\varepsilon)$ we started with, then replacing e by e' results in $t = u \to e' = e'$, which is provable from $e' = e'$, an instance of $(=_1)$.

Let π_1 be π_0 with the necessary new instances of $(=_\varepsilon)$, added. The instances of $(=_\varepsilon)$ in π_1 satisfy the properties required in the statement of the lemma.

However, the results of replacing e by e' may have impacted some of the critical formulas in the original proof. For a critical formula to which $e \equiv \varepsilon_x A(x,u)$ belongs is of the form

$$A(t',u) \to A(\varepsilon_x A(x,u),u) \tag{1}$$

which after replacing e by e' becomes

$$A(t'',u) \to A(\varepsilon_x A(x,t),u) \tag{2}$$

which is no longer a critical formula. This formula, however, can be derived from $t = u$ together with

$$\begin{aligned} &A(t'',u) \to A(\varepsilon_x A(x,t),u) && (\varepsilon)\\ &t = u \to (A(\varepsilon_x A(x,t),t) \to A(\varepsilon_x A(x,t),u)) && (=_2)\\ &u = t \to (A(t'',u) \to A(t'',t)) && (=_2) \end{aligned}$$

Let π_2 be π_1 plus these derivations of (2) with the instances of $(=_2)$ themselves proved from $(=_2')$ and $(=_\varepsilon)$. The rank of the new critical formulas is the same, so the rank of π_2 is the same as that of π. The new instances of $(=_\varepsilon)$ required for the derivation of the last two formulas only contain ε-terms of lower rank that that of e, as can be verified.

π_2 is thus a proof of E from $t = u$ which satisfies the conditions of the lemma. From it, we obtain a proof $\pi_2[t = u]$ of $t = u \to E$ by the deduction theorem. On the other hand, the instance $t = u \to e' = e$ under consideration can also be proved trivially from $t \neq u$. The proof $\pi[t \neq u]$ thus is also a proof, this time of $t \neq u \to E$, which satisfies the conditions of the lemma. We obtain π' by combining the two proofs.

Theorem 48 (First Epsilon Theorem for $\mathrm{EC}_\varepsilon^=$). *If E is a formula not containing any ε-terms and $\vdash_{\varepsilon=} E$, then $\vdash_= E$ (in $\mathrm{EC}^=$).*

Proof. By repeated application of the Lemma, every instance of $(=_\varepsilon)$ involving ε-terms of a given type p can be eliminated from π. The Theorem follows by induction on the number of different types of special ε-terms of rank $\mathrm{rk}(\pi)$ in π.

5 Proof Theory of the Epsilon Calculus

5.1 Sequent Calculi

Leisenring [12] presented a one-sided sequent calculus for the ε-calculus. It operates on sets of formulas (sequents); proofs are trees of sets of formulas each of which is either an axiom (at a leaf of the tree) or follows from the sets of formulas above it by an inference rule. Axioms are $A, \neg A$. The rules are given below:

$$\frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \wedge B}\ \wedge R \qquad \frac{\Gamma, \neg A, \neg B}{\Gamma, \neg(A \wedge B)}\ \wedge L \qquad \frac{\Gamma, A}{\Gamma, \neg\neg A}\ \neg\neg$$

$$\frac{\Gamma, A, B}{\Gamma, A \vee B}\ \vee R \qquad \frac{\Gamma, \neg A \quad \Gamma, \neg B}{\Gamma, \neg(A \vee B)}\ \vee L \qquad \frac{\Pi, A \quad \Lambda, \neg A}{\Pi, \Lambda}\ cut$$

$$\frac{\Gamma, A(t)}{\Gamma, \exists x\, A(x)}\ \exists R \qquad \frac{\Gamma, \neg A(\varepsilon_x\, A(x))}{\Gamma, \neg\exists x\, A(x)}\ \exists L \qquad \frac{\Gamma, A}{\Gamma, A, B}\ w$$

$$\frac{\Gamma, A(\varepsilon_x \neg A(x))}{\Gamma, \forall x\, A(x)}\ \forall R \qquad \frac{\Gamma, \neg A(t)}{\Gamma, \neg\forall x\, A(x)}\ \forall L$$

In contrast to classical sequent systems, there are no eigenvariable conditions!

It is complete, since proofs can easily be translated into derivations in EC_ε; in particular it derives critical formulas:

$$\dfrac{\dfrac{\neg A(t), A(t)}{\neg A(t), \boxed{\exists x\, A(x)}}\ \exists R \qquad \dfrac{\neg A(\varepsilon_x\, A(x)), A(\varepsilon_x\, A(x))}{\boxed{\neg\exists x\, A(x)}, A(\varepsilon_x\, A(x))}\ \exists L}{\neg A(t), A(\varepsilon_x\, A(x))}\ cut$$

This sequent, however, has no cut-free proof.

Maehara [13] instead proposed to simply add axioms corresponding to to critical formulas and leave out quantifier rules. Hence, its axioms are $\neg A, A$ and $\neg A(t), A(\varepsilon_x\, A(x))$. It is complete, since the additional axioms allow derivation of critical formulas. However, it is also not cut-free complete. Converses of critical formulas are derivable using cut:

$$\frac{\boxed{\neg\neg A(t)}, \neg A(\varepsilon_x \neg A(x)) \qquad \boxed{\neg A(t)}, A(t)}{\neg A(\varepsilon_x \neg A(x)), A(t)}\ cut$$

But these obviously have no cut-free proof. Furthermore, addition of these converses as axioms will not result in a cut-free complete system, either. Consider the example given by Wessels: Let $e = \varepsilon_x \neg(A(x) \vee B(x))$.

$$\dfrac{\begin{array}{c}\boxed{\neg\neg(A(t) \vee B(t))}, \neg(A(e) \vee B(e)) \\ \vdots\ cut \\ \boxed{\neg(A(e) \vee B(e))}, A(t) \vee B(t)\end{array} \qquad \dfrac{\neg A(e), A(e)}{\neg A(e), \boxed{A(e) \vee B(e)}}\ \vee R}{\neg A(\varepsilon_x \neg(A(x) \vee B(x))), A(t) \vee B(t)}\ cut$$

Wessels [20] proposed to add instead the following rule to the propositional one-sided sequent calculus:

$$\frac{\Gamma, \Delta(z), \neg A(z) \quad \Gamma, A(t)}{\Gamma, \Delta(\varepsilon_x\, A(x))}\ \varepsilon 0$$

Here, $\Delta(z)$ must be not empty, and z may not occur in the lower sequent. This system also derives critical formulas, and so is complete:

$$\dfrac{\dfrac{A(z), \neg A(z)}{\underbrace{\neg A(t)}_{\Gamma}, \underbrace{A(a)}_{\Delta}, \neg A(z)}\, w \qquad \underbrace{\neg A(t)}_{\Gamma}, A(t)}{\neg A(t), A(\varepsilon_x\, A(x))}\, \varepsilon 0$$

The rule $\varepsilon 0$ is sound.[3]

Wessels offered a cut-elimination proof for her system. However, the proof relied on a false lemma to which Maehara gave a counterexample.

Wessels' Lemma. If $\vdash \Gamma, \Delta(\varepsilon_x\, A(x))$ then $\vdash \Gamma, \Delta(z), \neg A(z)$.

Let $A(x) = P(x, \varepsilon_y\, Q(\varepsilon_u\, P(u, y)))$, $\Delta(z) = Q(z)$, and $\Gamma = \neg Q(\varepsilon_x\, B(x, w))$. Then

$$\underbrace{\neg Q(\varepsilon_x\, P(x, w))}_{\Gamma}, \underbrace{Q(\varepsilon_x\, P(x, \varepsilon_y\, Q(\varepsilon_u\, P(u, y))))}_{\Delta(\varepsilon_x\, A(x))}$$

is derivable, since it is of the form $\neg B(w), B(\varepsilon_y\, B(y))$. However, the corresponding sequent in the consequent of the lemma,

$$\underbrace{\neg Q(\varepsilon_x\, P(x, w))}_{\Gamma}, \underbrace{Q(z)}_{\Delta(z)}, \underbrace{\neg P(z, \varepsilon_y\, Q(\varepsilon_u\, P(u, y)))}_{\neg A(z)}$$

is not derivable, because not valid.[4]

Mints (in a review of Wessels' paper) proposed the following rule instead:

$$\dfrac{\Gamma, \Delta(\varepsilon_x\, A(x)), \neg A(\varepsilon_x\, A(x)) \quad \Gamma, A(t)}{\Gamma, \Delta(\varepsilon_x\, A(x))}\, \varepsilon 1$$

It, too, derives all critical formulas:

$$\dfrac{\dfrac{A(\varepsilon_x\, A(x)), \neg A(\varepsilon_x\, A(x))}{\underbrace{\neg A(t)}_{\Gamma}, \underbrace{A(\varepsilon_x\, A(x))}_{\Delta}, \neg A(\varepsilon_x\, A(x))}\, w \qquad \underbrace{\neg A(t)}_{\Gamma}, A(t)}{\neg A(t), A(\varepsilon_x\, A(x))}\, \varepsilon 1$$

[3] Suppose the upper sequents are valid but the lower sequent is not, i.e., for some $\mathfrak{M}, \Psi, s$, $\mathfrak{M} \not\models \Gamma, \Delta(\varepsilon_x\, A(x))$. In particular, $\mathfrak{M}, \Psi, s \not\models \Gamma$. Hence, $\mathfrak{M}, \Psi, s \models A(t)$, i.e., $\mathfrak{M}, \Psi, s \models A'(t, t_1, \ldots, t_n)$, as the right premise is valid. So $\mathrm{val}_{\mathfrak{M},\Psi,s}(t) \in \mathrm{val}_{\mathfrak{M},\Psi,s}(A(x))$. Now let $s(z) = \mathrm{val}_{M,\Psi,s}(\varepsilon_x\, A'(x, t_1, \ldots, t_n))$. Then $\mathfrak{M}, \Psi, s \models A(z)$ and so $\mathfrak{M}, \Psi, s \not\models \neg A(z)$. Since the left premise is valid, $\mathfrak{M}, \Psi, s \models \Delta(z)$. But also $\mathfrak{M}, \Psi, s \not\models \Delta(z)$ since $\mathfrak{M}, \Psi, s \not\models \Delta(\varepsilon_x\, A(x))$.

[4] Let $|\mathfrak{M}| = \{1, 2\}$, $Q^{\mathfrak{M}} = \{1\}$, $P^{\mathfrak{M}} = \{\langle 1, 2\rangle, \langle 2, 2\rangle\}$, $s(z) = s(w) = 2$. Since $\langle 1, 2\rangle \in P^{\mathfrak{M}}$, we can choose Ψ so that $\mathrm{val}_{\mathfrak{M},\Psi,s}(\varepsilon_x\, P(x, 2)) = 1$. So $\mathfrak{M}, \Psi, s \not\models \neg Q(\varepsilon_x\, P(x, w))$. Also, $\mathfrak{M}, \Psi, s \not\models Q(z)$. As $\mathrm{val}_{\mathfrak{M},\Psi,s}(\varepsilon_u\, P(u, 2)) = 1$ and $1 \in Q^{\mathfrak{M}}$, we can also fix Ψ so that $\mathrm{val}_{\mathfrak{M},\Psi,s}(\varepsilon_y\, Q(\varepsilon_u\, P(u, y))) = 2$. But then $\mathfrak{M}, \Psi, s \not\models \neg P(z, \varepsilon_y\, Q(\varepsilon_u\, P(u, y)))$.).

The system was developed in detail by Yasuhara [21]. The Mints-Yasuhara system is cut-free complete. However, it is not known if the sequent has a cut-elimination theorem that transforms a proof with cuts successively into one without cuts. Both Gentzen's and Tait's approach to cut-elimination do not seem to work. In a Gentzen-style proof, the main induction is on on cut length, i.e., the height of the proof tree above an uppermost cut. In the induction step, a cut is permuted upward to reduce the cut length. For instance, we replace the subproof proof ending in a cut

$$\dfrac{\dfrac{\vdots\ \pi}{\Pi, \boxed{A}} \quad \dfrac{\vdots\ \pi'}{\neg A, \Lambda, B(t)}}{\dfrac{\Pi, \boxed{\neg A}, \Lambda, \exists x\, B(x)}{\Pi, \Lambda, \exists x\, B(x)}\ \exists R}\ cut \qquad \text{by} \qquad \dfrac{\dfrac{\vdots\ \pi}{\Pi, \boxed{A}} \quad \dfrac{\vdots\ \pi'}{\boxed{\neg A}, \Lambda, B(t)}}{\dfrac{\Pi, \Lambda, B(t)}{\Pi, \Lambda, \exists x\, B(x)}\ \exists R}\ cut$$

To permute a cut across the $\varepsilon 1$ rule:

$$\dfrac{\dfrac{\vdots\ \pi}{\Pi, \boxed{A}} \quad \dfrac{\vdots\ \pi' \qquad\qquad \vdots\ \pi''}{\neg A, \Gamma, \Delta(\varepsilon_x\, B(x)), \neg B(\varepsilon_x\, B(x)) \qquad \Gamma, B(t)}\ \varepsilon 1}{\dfrac{\Pi, \boxed{\neg A}, \Gamma, \Delta(\varepsilon_x\, B(x))}{\Pi, \Gamma, \Delta(\varepsilon_x\, B(x))}\ cut}$$

one might try to replace the proof tree with

$$\dfrac{\dfrac{\dfrac{\vdots\ \pi}{\Pi, \boxed{A}} \quad \dfrac{\vdots\ \pi'}{\boxed{\neg A}, \Gamma, \Delta(\varepsilon_x\, B(x)), \neg B(\varepsilon_x\, B(x))}}{\Pi, \Gamma, \Delta(\varepsilon_x\, B(x)), \neg B(\varepsilon_x\, B(x))}\ cut \qquad \dfrac{\vdots\ \pi''}{\Gamma, B(t)}}{\Gamma, \Delta(\varepsilon_x\, B(x))}\ \varepsilon 1$$

However, here the condition on $\varepsilon 1$ is violated if $\neg A$ is in Δ.

In a Tait-style cut elimination proof, the main induction is on cut rank, i.e., complexity of the cut formula. In the induction step, the complexity of the cut formula is reduced. For instance, if a subproof ends in a cut

$$\dfrac{\dfrac{\vdots\ \pi}{\Pi, \boxed{\neg(A \wedge B)}} \quad \dfrac{\vdots\ \pi'}{\Lambda, \boxed{A \wedge B}}}{\Pi, \Lambda}\ cut$$

we replace it with

$$\dfrac{\dfrac{\dfrac{\vdots\ \pi_1}{\Pi, \boxed{\neg A}, \neg B} \quad \dfrac{\vdots\ \pi_1'}{\Lambda, \boxed{A}}}{\Pi, \Lambda, \boxed{\neg B}}\ cut \qquad \dfrac{\vdots\ \pi_2'}{\Lambda, \boxed{B}}}{\Pi, \Lambda}\ cut$$

This approach requires *inversion lemmas*. A typical case is: If $\pi' \vdash \Pi, A \wedge B$ then there is a $\pi'_1 \vdash \Pi, A$ of cut rank and length $\leq$ that of π'. In the proof of the inversion lemma, one replaces all ancestors of $A \wedge B$ in π' by A and "fixes" those rules that are no longer valid. For instance, replace

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Gamma, A & \Gamma, B\end{array}}{\Gamma, A}\ \wedge R \quad \text{by} \quad \begin{array}{c}\vdots \\ \Gamma, A\end{array}$$

But now consider a derivation π' which contains the $\varepsilon 1$ rule:[5]

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Pi, A \wedge B(\varepsilon_x\, C(x)), \neg C(\varepsilon_x\, C(x)) & \Pi, C(t)\end{array}}{\Pi, A \wedge B(\varepsilon_x\, C(x))}\ \varepsilon 1$$

The inversion lemma produces

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Pi, A, \neg C(\varepsilon_x\, C(x)) & \Pi, C(t)\end{array}}{\Pi, A}\ \varepsilon 1$$

This, again, no longer satisfies the condition of ε_1.

Open Problem 49. Prove cut-elimination for the Mints-Yasuhara system, or give a similarly simple sequent calculus for which it can be proved.

5.2 Natural Deduction

In Gentzen's classical natural deduction system NK, the quantifier rules are given by

$$\dfrac{A(z)}{\forall x\, A(x)}\ \forall I \qquad \dfrac{\forall x\, A(x)}{A(t)}\ \forall E$$

$$\dfrac{A(t)}{\exists x\, A(x)}\ \exists I \qquad \dfrac{\exists x\, A(x) \qquad \begin{array}{c}[A(z)] \\ \vdots \\ C\end{array}}{B}\ \exists E$$

where z must not appear in any undischarged assumptions (nor in $A(x)$ or B). Meyer Viol [15] has proposed a system in which the $\exists E$ rule is replaced by

$$\dfrac{\exists x\, A(x)}{A(\varepsilon_x\, A(x))}\ \exists E_\varepsilon$$

and the following term rule is added

$$\dfrac{A(t)}{A(\varepsilon_x\, A(x))}\ I\varepsilon$$

[5] $(A \wedge B(\varepsilon_x\, C(x))$ is $\Delta(\varepsilon_x\, C(x))$ in this case).

Open Problem 50. Does Meyer Viol's system have a normal form theorem?

Adding $\exists E_\varepsilon$ and $I\varepsilon$ to the intuitionistic system NJ results in a system that is not conservative over intuitionistic logic. For instance, *Plato's principle*, the formula

$$\exists x(\exists y\, A(y) \to A(x))$$

becomes derivable:

$$\dfrac{\dfrac{\dfrac{[\exists y\, A(\varepsilon_x\, A(x))]}{A(\varepsilon_x\, A(x))}\,\exists E_\varepsilon}{\exists y\, A(y) \to A(\varepsilon_x\, A(x))}\to I}{\exists x(\exists y\, A(y) \to A(x))}\,\exists I$$

However, the system also does not collapse to classical logic: it is conservative for propositional formulas.

Intuitionistic natural deduction systems are especially intriguing, as Abadi, Gonthier and Werner [1] have shown that a system of quantified propositional intuitionistic logic with a choice operator ε_X can be given a Curry-Howard correspondence via a type system which $\varepsilon_X\, A(X)$ is a type such that the type $A(X)$ is inhabited. System $\mathcal{E}$ is paired with a simply typed λ-calculus that, in addition to λ-abstraction and application, features *implementation*: $\langle t\colon A \text{ with } X = T\rangle$ of type $A(\epsilon_X A/X)$. If $A(X)$ is a type specification of an interface with variable type X, then $A(T)$ for some type T is an implementation of that interface.

References

1. Abadi, M., Gonthier, G., Werner, B.: Choice in dynamic linking. In: Walukiewicz, I. (ed.) FoSSaCS 2004. LNCS, vol. 2987, pp. 12–26. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24727-2_3
2. Abiteboul, S., Vianu, V.: Non-determinism in logic-based languages. Ann. Math. Artif. Intell. **3**(2–4), 151–186 (1991)
3. Asser, G.: Theorie der logischen Auswahlfunktionen. Z. Math. Logik Grundlag. Math. **3**, 30–68 (1957)
4. Avigad, J., Zach, R.: The epsilon calculus. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy (Summer 2016th edn. (2016). http://plato.stanford.edu/entries/epsilon-calculus/
5. Bell, J.L.: Hilbert's epsilon-operator and classical logic. J. Philos. Logic **22**, 1–18 (1993)
6. Blass, A., Gurevich, Y.: The logic of choice. J. Symbolic Logic **65**, 1264–1310 (2000)
7. DeVidi, D.: Intuitionistic epsilon- and tau-calculi. Math. Logic Q. **41**, 523–546 (1995)
8. von Heusinger, K.: The reference of indefinites. In: von Heusinger, K., Egli, U. (eds.) Reference and Anaphoric Relations, pp. 247–265. Kluwer, Dordrecht (2000)
9. von Heusinger, K.: Choice functions and the anaphoric semantics of definite NPs. Res. Lang. Comput. **2**, 309–329 (2004)
10. Hilbert, D.: Neubegründung der Mathematik: Erste Mitteilung. Abhandlungen aus dem Seminar der Hamburgischen Universität 1, 157–77 , series of talks given at the University of Hamburg, July 25–27, 1921. English in [14], pp. 198–214 (1922)

11. Hilbert, D., Bernays, P.: Grundlagen der Mathematik. Springer, Berlin (1939)
12. Leisenring, A.: Mathematical Logic and Hilbert's ϵ-symbol. MacDonald Technical and Scientific, London (1969)
13. Maehara, S.: The predicate calculus with ϵ-symbol. J. Math. Soc. Japan **7**, 323–344 (1955)
14. Mancosu, P. (ed.): From Brouwer to Hilbert. The Debate on the Foundations of Mathematics in the 1920s. Oxford University Press, New York (1998)
15. Meyer Viol, W.P.M.: Instantial Logic. An Investigation into Reasoning with Instances. ILLC Dissertation Series 1995–11. ILLC, Amsterdam (1995)
16. Mints, G., Sarenac, D.: Completeness of indexed epsilon-calculus. Arch. Math. Logic **42**, 617–625 (2003)
17. Mints, G.: Heyting predicate calculus with epsilon symbol. J. Soviet Math. **8**, 317–323 (1977)
18. Moser, G., Zach, R.: The epsilon calculus and herbrand complexity. Stud. Logica. **82**(1), 133–155 (2006)
19. Otto, M.: Epsilon-logic is more expressive than first-order logic over finite structures. J. Symbolic Logic **65**(4), 1749–1757 (2000)
20. Wessels, L.: Cut elimination in a Gentzen-style ε-calculus without identity. Z. Math. Logik Grundlag. Math. **23**, 527–538 (1977)
21. Yashahura, M.: Cut elimination in ϵ-calculi. Z. Math. Logik Grundlag. Math. **28**, 311–316 (1982)

Neighbourhood Contingency Bisimulation

Zeinab Bakhtiari[1], Hans van Ditmarsch[1,2], and Helle Hvid Hansen[3,4(✉)]

[1] LORIA, CNRS — Université de Lorraine, Nancy, France
[2] Institute for Mathematical Sciences, Chennai, India
[3] Delft University of Technology, Delft, The Netherlands
h.h.hansen@tudelft.nl
[4] CWI, Amsterdam, The Netherlands

Abstract. We introduce a notion of bisimulation for contingency logic interpreted on neighbourhood structures, characterise this logic as bisimulation-invariant fragment of modal logic and of first-order logic, and compare it with existing notions in the literature.

1 Introduction

A proposition is non-contingent if it is necessarily true or necessarily false, and otherwise it contingent. The notion of (non-)contingency goes back to Aristotle [1]. The modal logic of contingency goes back to Montgomery and Routley [14]. They captured non-contingency by an operator Δ such that $\Delta\varphi$ means that formula φ is non-contingent (and where $\nabla\varphi$ means that φ is contingent). In an epistemic modal logic, 'φ is non-contingent' means that you know whether φ, and 'φ is contingent' means that you are ignorant about φ [8,10,18]. Contingency is definable with necessity: $\Delta\varphi$ is definable as $\Box\varphi \vee \Box\neg\varphi$. But necessity cannot always be defined with non-contingency. The definability of $\Box$ with Δ has been explored in various studies [7,14,16]. In [7] the *almost-definability* schema $\nabla\psi \to (\Box\varphi \leftrightarrow (\Delta\varphi \wedge \Delta(\psi \to \varphi)))$ is proposed — as long as there is a contingent proposition ψ, $\Box$ is definable with Δ; which inspired a matching notion of *contingency bisimulation*: back and forth only apply when non-bisimilar accessible worlds exist.

Schemas such as $\Delta(\varphi \wedge \psi) \to (\Delta\varphi \wedge \Delta\psi)$ are invalid for the non-contingency operator. The operator Δ is therefore not monotone, and the logic of contingency is not a normal modal logic. Non-normal logics are standardly interpreted on neighbourhood models [2,13,17]. Fan and Van Ditmarsch proposed in [6] to interpret the contingency operator on neighbourhood models. They left as an open question what a suitable notion of contingency bisimulation would be over neighbourhood models. We answer this question here.

We introduce a notion of *neighbourhood Δ-bisimilarity*, inspired by the semantics of the Δ-modality and [9], where different notions of structural invariance among neighbourhood models were studied. By way of augmented neighbourhood models and their correspondence to Kripke models we can provide a detailed comparison to the bisimulations of [7]. We show that the two notions

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 48–63, 2017.
DOI: 10.1007/978-3-662-54069-5_5

differ at the level of relations, but the ensuing bisimilarity notions coincide. Furthermore, we investigate the notions of Δ-morphisms and Δ-quotients and prove some analogues of results from [9]. These are instrumental in proving our two characterisation theorems (similar to [7, Theorems 4.4 and 4.5]): neighbourhood contingency logic is the Δ-bisimulation invariant fragment of classical modal logic, and of first-order logic.

Section 2 provides preliminaries. Section 3 recalls contingency logic over Kripke models and introduces different perspectives on relational contingency bisimulation. Section 4 introduces neighbourhood contingency bisimulation and studies its properties, and it is followed by the characterisation results in Sect. 5. The concluding Sect. 6 reflects on the relevance of our work and indicates future directions. Due to space limitations, some proofs have been omitted. They will be included in the extended version.

2 Coherence

We assume that the reader is familiar with the standard notions of sets, functions and relations. The following is merely to recall notation and to introduce the crucial notion of coherence. Given $U \subseteq X$, we denote by U^c the complement of U in X. The disjoint union of two sets X_1 and X_2 is denoted by $X_1 + X_2$ and the inclusion maps by $\iota_i \colon X_i \to X_1 + X_2$, $i = 1, 2$. Given a function $f \colon X \to Y$, the f-image of $U \subseteq X$ is $f[U] = \{f(x) \in Y \mid x \in U\}$, and the inverse f-image of $V \subseteq Y$ is $f^{-1}[V] = \{x \in X \mid f(x) \in V\}$. The *graph of* f is the relation $Gr(f) = \{(x, f(x)) \subseteq X \times Y \mid x \in X\}$. The *kernel of* f is the relation $\ker(f) = \{(x, y) \in X \times X \mid f(x) = f(y)\}$. Let $R \subseteq X \times Y$ be a relation. The R-image of $U \subseteq X$ is the set $R[U] = \{y \in Y \mid \exists x \in U : (x, y) \in R\}$, and the inverse R-image of $V \subseteq Y$ is $R^{-1}[V] = \{x \in X \mid \exists y \in V : (x, y) \in R\}$.

Given a relation $R \subseteq X \times Y$, the converse of R is written $R^{-1} \subseteq Y \times X$, the composition of R and $S \subseteq Y \times Z$ is $R; S \subseteq X \times Z$. For the reflexive, symmetric, and transitive closure we employ, respectively, R^r, R^s, and R^+ such that the equivalence closure can be defined as $R^e = ((R^r)^s)^+$. If R is an equivalence relation, we often write $[x]_R$ (or simply $[x]$) instead of $R(x)$.

Definition 1 (R-coherent pairs). *Let $R \subseteq X \times Y$ be a relation, $U \subseteq X$ and $V \subseteq Y$. The pair (U, V) is* R-coherent *if $R[U] \subseteq V$ and $R^{-1}[V] \subseteq U$, or equivalently, for all $(x, y) \in R$, $x \in U$ iff $y \in V$. Given a relation $R \subseteq X \times X$, we say that $U \subseteq X$ is* R-closed *if (U, U) is R-coherent.*

Note that if R is reflexive and (U, U') is R-coherent, then $U = U'$.

3 Contingency Logic

In this section we introduce basic modal logic and contingency logic on Kripke models, and contingency bisimulation following [7,8]. We also compare that to a novel notion of relational contingency bisimulation in terms of coherence.

Definition 2 (Languages). *Let* AtProp *be a set of atomic propositions. The languages* $\mathcal{L}_\Box$ *and* $\mathcal{L}_\Delta$ *are generated by the following grammars:*

$$\mathcal{L}_\Box \ni \varphi \quad ::= \; p \in \mathsf{AtProp} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box\varphi$$
$$\mathcal{L}_\Delta \ni \varphi \quad ::= \; p \in \mathsf{AtProp} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Delta\varphi$$

The other Boolean connectives $\bot, \top$, $\vee$ *and* $\leftrightarrow$ *are defined in the usual way.*

The formula $\Box\varphi$ should be read as "φ is necessarily true", and the formula $\Delta\varphi$ as "φ is non-contingent". The language $\mathcal{L}_\Delta$ can be viewed as a fragment of $\mathcal{L}_\Box$ via an inductively defined translation $(-)^t : \mathcal{L}_\Delta \to \mathcal{L}_\Box$ with only non-trivial clause $(\Delta\varphi)^t = \Box\varphi^t \vee \Box\neg\varphi^t$.

Definition 3 (Kripke models). *A* (Kripke) frame *is a pair* $F = (S, R)$ *where* S *is a set (of states), and* $R \subseteq S \times S$ *is an accessibility relation. A* Kripke model *is a triple* $M = (S, R, V)$ *where* (S, R) *is a frame and where* $V\colon \mathsf{AtProp} \to \mathcal{P}(S)$ *is a* valuation. *Given* $, \in S$*, a pair* (M, s) *is a* pointed model.

Definition 4. *Let* $M = (S, R, V)$ *be a Kripke model, and* $s \in S$*. The interpretation of formulas from* $\mathcal{L}_\Box$ *and* $\mathcal{L}_\Delta$ *is defined inductively in the usual manner:*

$$\begin{array}{ll} M, s \models p & \textit{iff } s \in V(p) \\ M, s \models \varphi \wedge \psi & \textit{iff } M, s \models \varphi \textit{and} M, s \models \psi \\ M, s \models \neg\varphi & \textit{iff } M, s \not\models \varphi \\ M, s \models \Box\varphi & \textit{iff for all } t \in R(s) : M, t \models \varphi \\ M, s \models \Delta\varphi & \textit{iff for all } t_1, t_2 \in R(s) : (M, t_1 \models \varphi \Leftrightarrow M, t_2 \models \varphi). \end{array}$$

where $p \in \mathsf{AtProp}$*. We say that* (M, s) *and* (M', s') *are* modally $\mathcal{L}_\Delta$*-equivalent (notation:* $(M, s) \equiv_\Delta (M', s')$*) if for all* $\varphi \in \mathcal{L}_\Delta$*,* $M, s \models \varphi$ *iff* $M', s' \models \varphi$*.*

For all Kripke models M, states s in M, and all $\varphi \in \mathcal{L}_\Delta$, $M, s \models \varphi$ iff $M, s \models \varphi^t$.

We assume the reader is familiar with standard relational bisimulations (for $\Box$). In [7], Fan, Wang and Van Ditmarsch defined a weaker notion (for Δ) which we refer to as *o-Δ-bisimulation* for "original Δ-bisimulation".

Definition 5 (o-Δ-bisimulation [7]). *Let* $M = (S, R, V)$ *be a Kripke model. A relation* $Z \subseteq S \times S$ *is an* o-Δ-bisimulation *on* M*, if whenever* $(s, s') \in Z$*:*

(**Atoms**) *s and s' satisfy the same propositional variables;*
(Δ**−Zig**) *for all* $t \in R(s)$*, if there are* $t_1, t_2 \in R(s)$ *such that* $(t_1, t_2) \notin Z$*, then there is a* $t' \in R(s')$ *such that* $(t, t') \in Z$*;*
(Δ**−Zag**) *for all* $t' \in R(s')$*, if there are* $t'_1, t'_2 \in R(s')$ *such that* $(t'_1, t'_2) \notin Z$*, then there is a* $t \in R(s)$ *such that* $(t, t') \in Z$*.*

We write $(M, s) \approx^{\mathsf{on}}_\Delta (M, s')$*, if there is an o-$\Delta$-bisimulation on M that contains* (s, s')*. Two pointed models* (M, s) *and* (M', s') *are* o-Δ-bisimilar*, written* $(M, s) \approx_\Delta (M', s')$*, if* $(M + M', \iota_1(s)) \approx^{\mathsf{on}}_\Delta (M + M', \iota_2(s'))$*, i.e., there is an o-$\Delta$-bisimulation on the disjoint union of M and M' linking (the injection images of) s and s'.*

Note that $(M, s) \approx_{\Delta} (M', s')$ is not witnessed by a relation $Z \subseteq S \times S'$ since, by definition, o-Δ-bisimulation relations always live on a single model.

We introduced the notation $\approx^{\mathsf{on}}_{\Delta}$, since, a priori, it is not clear whether $(M, s) \approx^{\mathsf{on}}_{\Delta} (M, s')$ iff $(M, s) \approx_{\Delta} (M, s')$. At the end of this section (Proposition 4), we will see that, in fact, this is true, and hence we could dispense with the notation $\approx^{\mathsf{on}}_{\Delta}$, but for now we keep writing $\approx^{\mathsf{on}}_{\Delta}$ for clarity.

Given a model M, we will also view $\approx^{\mathsf{on}}_{\Delta}$ as the relation on the state space of M that contains all pairs (s, s') such that $(M, s) \approx^{\mathsf{on}}_{\Delta} (M, s')$. In order to compare o-$\Delta$-bisimilarity with our later notion (in Definition 6), we need the following result.

Proposition 1. *For all Kripke models M, the relation $\approx^{\mathsf{on}}_{\Delta}$ on M is an equivalence relation, and itself an o-Δ-bisimulation on M.*

Proposition 1 follows from the stronger result that o-Δ-bisimilarity is an equivalence relation over the class of all pointed Kripke models [4,5]. This is quite non-trivial to prove, since o-Δ-bisimulations are not closed under composition (Example 1). Our proof relies on a number of closure properties. We must omit details due to space limitations.

Lemma 1. *The set of o-Δ-bisimulation relations on a Kripke model M is closed under taking unions, converse, and transitive symmetric closure.*

It is now easy to prove Proposition 1 using the closure properties of Lemma 1. *Proof of Proposition 1.* By definition, the relation $\approx^{\mathsf{on}}_{\Delta}$ on M is the union of all o-Δ-bisimulations on M, and hence the largest one. Reflexivity of $\approx^{\mathsf{on}}_{\Delta}$ follows since the identity relation is an (o-Δ-)bisimulation. Symmetry follows from closure under converse. For transitivity, we use that the composition of two bisimulations is contained in the transitive symmetric closure of their union, which is again a bisimulation.

On (relational) $\Box$-bisimilarity and o-Δ-bisimilarity it is known that $\Box$-bisimilarity implies o-Δ-bisimilarity, but not vice versa [7, Proposition 3.4]; o-Δ-bisimilarity implies $\mathcal{L}_{\Delta}$-equivalence [7, Proposition 3.5], whereas the converse only holds over saturated Kripke models [7, Proposition 3.9]; An $\mathcal{L}_{\Box}$-formula is equivalent to an $\mathcal{L}_{\Delta}$-formula iff it is invariant under o-Δ-bisimulation [7, Theorem 4.4]. A first-order formula is equivalent to an $\mathcal{L}_{\Delta}$-formula iff it is invariant under o-Δ-bisimulation [7, Theorem 4.5]. o-Δ-bisimilarity is an o-Δ-bisimulation [7, Proposition 3.13].

The notion of contingency bisimulation for neighbourhood models using coherent sets, introduced later in Definition 9, has a natural analogue for Kripke models. The definition is derived from the semantics of the Δ-modality.

Definition 6 (rel-Δ-bisimulation). *Let $M = (S, R, V)$ and $M' = (S', R', V')$ be Kripke models. A relation $Z \subseteq S \times S'$ is a* rel-Δ-bisimulation *(for* relational Δ-bisimulation*) between M and M', if whenever $(s, s') \in Z$:*

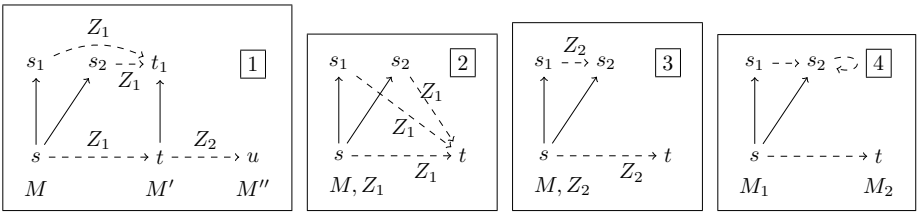
(**Atoms**) *s and s' satisfy the same propositional variables;*
(**Coherence**) *for all Z-coherent pairs (U, U'):*

$$(R(s) \subseteq U \text{ or } R(s) \subseteq U^c) \quad \textit{iff} \quad (R'(s') \subseteq U' \text{ or } R'(s') \subseteq U'^c)$$

We write $(M, s) \sim_{\Delta}^{\mathsf{betw}} (M', s')$*, if there is a rel-*$\Delta$*-bisimulation between* M *and* M' *that contains* (s, s')*. A rel-*Δ*-bisimulation* on a model M *is a rel-*Δ*-bisimulation between* M *and* M*. We define the notion of rel-*Δ*-bisimilarity between states in potentially different models via the disjoint union (analogously to the notion of o-*Δ*-bisimilarity): Two pointed models* (M, s) *and* (M', s') *are* rel-Δ-bisimilar*, written* $(M, s) \sim_{\Delta} (M', s')$*, if* $(M + M', \iota_1(s)) \sim_{\Delta}^{\mathsf{betw}} (M + M', \iota_2(s'))$*, i.e.,. if there is a rel-*Δ*-bisimulation on* $M + M'$ *that contains* $(\iota_1(s), \iota_2(s'))$*.*

In Proposition 4 we will see that over a single model $\sim_{\Delta}^{\mathsf{betw}}$ and $\sim_{\Delta}$ coincide, but in general they differ. At first it would seem more natural to define rel-Δ-bisimilarity between pointed models as $\sim_{\Delta}^{\mathsf{betw}}$. However, the following Example 1 (item 4) shows that this notion is too restrictive. The example also shows that, in general, rel-Δ-bisimulations are different from o-Δ-bisimulations.

Example 1. Consider the four figures (and matching items below) where we assume a single variable p to be false in all states of all models, except in figure 4 where p is true at s and t.



1 *The composition of two o-*Δ*-bisimulations may not be an o-*Δ*-bisimulation.* Z_1 and Z_2 are o-Δ-bisimulations, but not $Z_1; Z_2 = \{(s, u)\}$, as Δ-Zig fails.

2 *A rel-*Δ*-bisimulation may not be an o-*Δ*-bisimulation.* Z_1 is not an o-Δ-bisimulation, since Δ-Zig fails for $(s, t) \in Z_1$. However, Z_1 is a rel-Δ-bisimulation on M. The Z_1-coherent pairs are: $(\{s, s_1, s_2\}, U')$ and (S, U') for all U' with $t \in U'$, $(\{t\}, U')$ for all U' with $t \notin U'$, and $(\emptyset, \emptyset)$. Since $R(s_1) = R(s_2) = R(t) = \emptyset$, **(Coherence)** for (s_1, t) and (s_2, t) is satisfied. For (s, t), e.g., for $(\{s, s_1, s_2\}, \{t\})$: $R(s) = \{s_1, s_2\} \subseteq \{s, s_1, s_2\}$ and $R(t) = \emptyset \subseteq \{t\}$, and for $(\{t\}, \{s_1\})$: $R(s) = \{s_1, s_2\} \subseteq \{t\}^c$ and $R(t) = \emptyset \subseteq \{s_1\}$.

3 *An o-*Δ*-bisimulation may not be a rel-*Δ*-bisimulation.* Z_2 is an o-Δ-bisimulation, but not a rel-Δ-bisimulation, since $(\{s_1\}, \{s_2\})$ is Z_2-coherent, $(s, t) \in Z_2$, and $\emptyset = R(t) \subseteq \{s_2\}$, but $R(s) \not\subseteq \{s_1\}$ and $R(s) \not\subseteq \{s_1\}^c$.

4 *A rel-*Δ*-bisimulation on a disjoint union, but not between disjoints.* The pictured relation is a rel-Δ-bisimulation on $M_1 + M_2$, but there is no rel-Δ-bisimulation between M_1 and M_2 linking s and t. The only candidate is $\{(s, t)\}$, but the coherent pair $(\{s, s_1\}, \{t\})$ does not satisfy **(Coherence)**. So $(M_1 + M_2, \iota_1(s)) \sim_{\Delta}^{\mathsf{betw}} (M_1 + M_2, \iota_2(t))$, but not $(M_1, s) \sim_{\Delta}^{\mathsf{betw}} (M_2, t)$.

Although the two notions of contingency bisimulations differ at the level of relations, we can show that rel-Δ-bisimilarity coincides with o-Δ-bisimilarity. We will need the following lemma.

Lemma 2. *Let $M = (S, R, V)$ be a Kripke model, and assume that $Z \subseteq S \times S$ is an equivalence relation. Z is an o-Δ-bisimulation iff Z is a rel-Δ-bisimulation.*

Proof. First, suppose Z is an o-Δ-bisimulation and $(s, s') \in Z$. Since Z is an equivalence relation, we need to show that for all Z-closed subsets U,

$$(R(s) \subseteq U \text{ or } R(s) \subseteq U^c) \quad \text{iff} \quad (R(s') \subseteq U \text{ or } R(s') \subseteq U^c) \tag{1}$$

To see that (1) holds, let $R(s) \subseteq U$ *or* $R(s) \subseteq U^c$, where U is Z-closed. Suppose towards a contradiction that $R(s') \cap U \neq \emptyset$ and $R(s') \cap U^c \neq \emptyset$. Then, there are $t_1, t_2 \in R(s')$ such that $t_1 \in U$ and $t_2 \in U^c$. Since U is Z-closed, $(t_1, t_2) \notin Z$. By applying Δ-Zag, there are $s_1, s_2 \in R(s)$ such that $(s_1, t_1), (s_2, t_2) \in Z$. From $R(s) \subseteq U$ *or* $R(s) \subseteq U^c$, we obtain $t_1, t_2 \in U$ or $t_1, t_2 \in U^c$, which is a contradiction. Therefore, $R(s') \subseteq U$ or $R(s') \subseteq U^c$. The other direction of (1) may be checked in a similar way.

Now, assume that Z is a rel-Δ-bisimulation, and let $(s, s') \in Z$. (**Atoms**) is immediate. For Δ-Zig, assume $t, t_1, t_2 \in R(s)$ such that $(t_1, t_2) \notin Z$. Suppose towards a contradiction that there is no $t' \in R(s')$ such that $(t, t') \in Z$, then $Z(t) \cap R(s') = \emptyset$ and hence $R(s') \subseteq (Z(t))^c$. As $Z(t)$ is Z-closed and Z is a rel-Δ-bisimulation we get by **(Coherence)** that $R(s) \subseteq Z(t)$ or $R(s) \subseteq (Z(t))^c$. But $R(s) \subseteq Z(t)$ is false since $(t_1, t_2) \notin Z$, and $R(s) \subseteq (Z(t))^c$ is also false since $t \in R(s) \cap Z(t)$. Hence we have a contradiction and conclude that Z satisfies the Δ-Zig condition. By a similar argument Z satisfies Δ-Zig.

We have the following analogue of Proposition 1, and it can be proved in a similar way (via closure properties). We omit a proof due to space limitations.

Proposition 2. *For all Kripke models M, the relation $\sim_\Delta^{\mathsf{betw}}$ on M is the largest rel-Δ-bisimulation on M, and it is is an equivalence relation on S.*

It follows from Propositions 1 and 2, and Lemma 2 that the two notions of contingency bisimilarity coincide.

Proposition 3. *Let M and M' be Kripke models.*

1. *For all s, t in M: $(M, s) \approx_\Delta^{\mathsf{on}} (M, t)$ iff $(M, s) \sim_\Delta^{\mathsf{betw}} (M, t)$.*
2. *For all s in M and s' in M': $(M, s) \approx_\Delta (M', s')$ iff $(M, s) \sim_\Delta (M', s')$.*

Recall that [4,5] proved that over the class of all pointed Kripke models, o-Δ-bisimilarity $\approx_\Delta$ is an equivalence. Due to Proposition 3(2), it follows that also rel-Δ-bisimilarity $\sim_\Delta$ an equivalence.

Finally, we show that we could dispense with the notation $\approx_\Delta^{\mathsf{on}}$ as item 1 of the next proposition ensures that no ambiguity can arise when writing $(M, s) \approx_\Delta (M, s')$. We also clarify the similar question regarding $\sim_\Delta^{\mathsf{betw}}$ and $\sim_\Delta$.

Proposition 4. *For all Kripke models M and M':*
1. $(M,s) \approx^{\mathsf{on}}_{\Delta} (M,s')$ *iff* $(M,s) \approx_{\Delta} (M,s')$.
2. $(M,s) \sim^{\mathsf{betw}}_{\Delta} (M,s')$ *iff* $(M,s) \sim_{\Delta} (M,s')$.
3. $(M,s) \sim^{\mathsf{betw}}_{\Delta} (M',s')$ *implies* $(M,s) \sim_{\Delta} (M',s')$. *The implication is strict.*

Proof. Item 1. ($\Rightarrow$): If Z is an o-Δ-bisimulation on M, then it is easy to prove that $Y := \{(\iota_1(s), \iota_2(t)) \mid (s,t) \in Z\}$ is a o-Δ-bisimulation on $M+M$.

($\Leftarrow$): Let Y be an o-Δ-bisimulation on $M+M$. Define $Z := \{(s,s') \in S \times S \mid \exists i,j \in \{1,2\} : (\iota_i(s), \iota_j(s')) \in Y\}$. To prove Δ-Zig for Z, suppose $(s,s') \in Z$ and $t, t_1, t_2 \in R(s)$ such that $(t_1,t_2) \notin Z$. This implies that $\iota_i(t), \iota_i(t_1), \iota_i(t_2) \in R_i(\iota_i(s))$, and there are $i,j \in \{1,2\}$ such that $(\iota_i(s), \iota_j(s')) \in Y$, and by definition of Z, $(\iota_i(t_1), \iota_i(t_2)) \notin Y$. By Δ-Zig for Y, there are $\iota_j(t'), \iota_j(t'_1), \iota_j(t'_2) \in R_j(\iota_j(s'))$ such that $(\iota_i(t), \iota_j(t')), (\iota_i(t_1), \iota_j(t'_1)), (\iota_i(t_2), \iota_j(t'_2)) \in Y$. Hence $t' \in R(s')$ and $(t,t'), (t_1,t'_1), (t_2,t'_2) \in Z$, which proves Δ-Zig. Δ-Zag can be proved in a similar manner.

Item 2.
$$\begin{aligned}(M,s) \sim^{\mathsf{betw}}_{\Delta} (M,s') &\iff (M,s) \approx^{\mathsf{on}}_{\Delta} (M,s') \text{ Proposition 3(2)}\\ &\iff (M,s) \approx_{\Delta} (M,s') \text{ (Item 1)}\\ &\iff (M,s) \sim_{\Delta} (M,s') \text{ Proposition 3(1)}\end{aligned}$$

Item 3. The implication can be proved using Lemma 3 and Proposition 5 of the next section. The converse fails since item 4 of Example 1 shows models (M_1,s) and (M_2,t) such that $(M_1,s) \sim_{\Delta} (M_2,t)$, however, we do not have $(M_1,s) \sim^{\mathsf{betw}}_{\Delta} (M_2,t)$.

4 Neighbourhood Semantics of Contingency Logic

In this section we recall the neighbourhood semantics of $\mathcal{L}_{\Delta}$ from [8], and then we proceed to introduce the notion of Δ-bisimulation between neighbourhood models, and investigate its properties.

Definition 7 (Neighbourhood models). *A neighbourhood frame is a pair (S,ν) where S is a set of states and $\nu : S \to \mathcal{P}(\mathcal{P}(S))$ is a neighbourhood function which assigns to each $s \in S$ its collection $\nu(s)$ of neighbourhoods. A neighbourhood model is a triple $M = (S,\nu,V)$ where (S,ν) is a neighbourhood frame and $V : \mathsf{AtProp} \to \mathcal{P}(S)$ is a valuation. A* neighbourhood morphism *between $M = (S,\nu,V)$ and $M' = (S',\nu',V')$ is a function $f : S \to S'$ such that (i) for all $p \in \mathsf{AtProp}$, $s \in V(p)$ iff $f(s) \in V'(p)$, and (ii) for all subsets $U \subseteq S'$, $f^{-1}(U) \in \nu(s)$ iff $U \in \nu'(f(s))$.*

Neighbourhood morphisms are the neighbourhood analogue of bounded morphisms, and they indeed preserve truth of $\mathcal{L}_{\Box}$-formulas [9, Lemma 2.6], and hence also of $\mathcal{L}_{\Delta}$-formulas. The semantics of $\mathcal{L}_{\Box} \cup \mathcal{L}_{\Delta}$-formulas is given below in Definition 8.

In what follows, we will also use disjoint unions (or coproducts) of neighbourhood models. We recall the definition from [9, Definition 2.9]. Let $M_1 = (S_1,\nu_1,V_1)$ and $M_2 = (S_2,\nu_2,V_2)$ be two neighbourhood models. Their *disjoint union* $M_1 + M_2$ is the model $M = (S,\nu,V)$ where $S = S_1 + S_2$,

$V(p) = \iota_1[V_1(p)] \cup \iota_2[V_2(p)]$, and for all $U \subseteq S_1 + S_2$, all $i = 1, 2$, and all $s_i \in S_i$: $U \in \nu(\iota_i(s_i))$ iff $\iota_i^{-1}[S_i] \in \nu_i(s_i)$. Being a bit sloppy and omitting explicit use of inclusion maps, this condition can be stated as: $U \in \nu(s_i)$ iff $U \cap S_i \in \nu_i(s_i)$. The definition of ν ensures that the inclusion maps $\iota_i \colon S_i \to S_1 + S_2$ are neighbourhood morphisms, and hence preserve truth of $\mathcal{L}_\Box \cup \mathcal{L}_\Delta$-formulas.

Definition 8 (Neighbourhood Semantics of Contingency Logic). *Given a neighbourhood model $M = (S, \nu, V)$. The interpretation of formulas from $\mathcal{L}_\Box$ and $\mathcal{L}_\Delta$ in M is defined inductively for atomic propositions and Boolean connectives as usual. Truth of modal formulas is given by,*

$$M, s \models \Box\varphi \text{ iff } \quad [\![\varphi]\!]_M \in \nu(s)$$
$$M, s \models \Delta\varphi \text{ iff } \quad [\![\varphi]\!]_M \in \nu(s) \text{ or } [\![\varphi]\!]^c_M \in \nu(s).$$

where $[\![\varphi]\!]_M = \{s \in S \mid M, s \models \varphi\}$ denotes the truth set of φ in M. We write $(M, s) \equiv_\Delta (M', s')$ if (M, s) and (M', s') satisfy the same $\mathcal{L}_\Delta$-formulas.

Again, it is clear that over neighbourhood models we can view $\mathcal{L}_\Delta$ as a fragment of $\mathcal{L}_\Box$, since for all neighbourhood models M, all states s in M, and all $\varphi \in \mathcal{L}_\Delta$, $M, s \models \varphi$ iff $M, s \models \varphi^t$.

Augmented Neighbourhood Models. Neighbourhood semantics can be seen as a generalization of Kripke semantics, since every Kripke model can be turned into a pointwise equivalent neighbourhood model, cf. [2, Theorem 7.9]. For a Kripke model $K = (S, R, V)$, define $nbh(K) = (S, \nu_R, V)$ where $\nu_R(s) = \{X \subseteq S \mid R(s) \subseteq X\}$. It is straightforward to check that for all $\varphi \in \mathcal{L}_\Box \cup \mathcal{L}_\Delta$,

$$K, s \models \varphi \quad \text{iff} \quad nbh(K), s \models \varphi. \tag{2}$$

A neighbourhood model (S, ν, V) is *augmented* (cf. [2]) if all neighbourhood collections are closed under supersets and under arbitrary intersections, that is, for all $s \in S$, if $U \in \nu(s)$ and $U \subseteq U' \subseteq S$, then $U' \in \nu(s)$; and $\bigcap \nu_R(s) \in \nu_R(s)$. For an augmented $M = (S, \nu, V)$, define a Kripke model $krp(M) = (S, R, V)$ by taking $R(s) = \bigcap \nu(s)$. Again, M and $krp(M)$ are pointwise equivalent, and we have $nbh(krp(M)) = M$ and $krp(nbh(K)) = K$. Thus, Kripke models are in 1-1 correspondence with augmented neighbourhood models.

In [8, Theorem 19], the logic CL was shown to be sound and strongly complete with respect to the class of Kripke frames. From Eq. (2) it follows immediately that CL is sound and strongly complete with respect to the class of augmented neighbourhood frames. This question was left open in [6].

We now define the notion of Δ-bisimulation between neighbourhood models. The idea of this definition was inspired by the definition of *precocongruences* in [9] and the neighbourhood semantics of the Δ-modality.

Definition 9 (nbh-Δ-bisimulation). *Let $M = (S, \nu, V)$ and $M' = (S', \nu', V')$ be neighbourhood models. A relation $Z \subseteq S \times S'$ is a* nbh-Δ-bisimulation *(for "neighbourhood Δ-bisimulation") if for all $(s, s') \in Z$, the following hold:*

(**Atoms**) *s and s' satisfy the same atomic propositions.*
(**Coherence**) *for all Z-coherent pairs (U, U'):*

$$U \in \nu(s) \text{ or } U^c \in \nu(s) \quad \text{iff} \quad U' \in \nu'(s') \text{ or } U'^c \in \nu'(s').$$

We write $(M, s) \sim_{\Delta}^{\mathsf{betw}} (M', s')$, if there is a nbh-$\Delta$-bisimulation between M and M' that contains (s, s'). A nbh-Δ-bisimulation on a model *M is a nbh-Δ-bisimulation between M and M. Two pointed models (M, s) and (M', s') are* nbh-Δ-bisimilar*, written $(M, s) \sim_{\Delta} (M', s')$, if $(M{+}M', \iota_1(s)) \sim_{\Delta}^{\mathsf{betw}} (M{+}M', \iota_2(s'))$, i.e., if there is a nbh-Δ-bisimulation on $M + M'$ that contains $(\iota_1(s), \iota_2(s'))$.*

The following proposition shows that there is no conflict between the notions of nbh-Δ-bisimulations and rel-Δ-bisimulations for augmented models. This allows us to simply speak of Δ-bisimulations, and it justifies the overloading of the notation $\sim_{\Delta}$.

Proposition 5. *A relation Z is a rel-Δ-bisimulation between Kripke models M and M' if and only if Z is a nbh-Δ-bisimulation between $nbh(M)$ and $nbh(M')$. Consequently,*

1. *$(M, s) \sim_{\Delta}^{\mathsf{betw}} (M', s')$ iff $(nbh(M), s) \sim_{\Delta}^{\mathsf{betw}} (nbh(M'), s')$.*
2. *$(M, s) \sim_{\Delta} (M', s')$ iff $(nbh(M), s) \sim_{\Delta} (nbh(M'), s')$.*

Proof. Item 1 is straigtforward to prove using the correspondence between Kripke models and augmented neighbourhood models. Item 2 can be proved using item 1 and the isomorphism $nbh(M + M') \cong nbh(M) + nbh(M')$, which is easy to verify.

Over arbitrary pointed neighbourhood models, $\sim_{\Delta}^{\mathsf{betw}}$ is strictly contained in $\sim_{\Delta}$, but on a single neighbourhood model they coincide.

Lemma 3. *For all pointed neighbourhood models (M, s) and (M', s'):*

1. *$(M, s) \sim_{\Delta}^{\mathsf{betw}} (M', s')$ implies $(M, s) \sim_{\Delta} (M', s')$. The implication is strict.*
2. *$(M, s) \sim_{\Delta}^{\mathsf{betw}} (M, s')$ iff $(M, s) \sim_{\Delta} (M, s')$.*

Proof. Item 1. One can show that if Z is a nbh-Δ-bisimulation between M_1 and M_2, then the embedding $\iota(Z) = \{(\iota_1(s_1), \iota_2(s_2)) \mid (s_1, s_2) \in Z\}$ is a nbh-Δ-bisimulation on $M_1 + M_2 = (S, \nu, V)$. The implication is strict due to Example 1 (item 4) and Proposition 5.

Item 2. ($\Rightarrow$) follows from item 1. To prove ($\Leftarrow$), assume that Y is a nbh-Δ-bisimulation on $M + M$. We show that $Z := \{(s, t) \in S \times S \mid \exists i, j \in \{1, 2\} : (\iota_i(s), \iota_j(t)) \in Y\}$ is a nbh-Δ-bisimulation on M. First, note that for all $s \in S$, $U \subseteq S$, and all $i \in \{1, 2\}$: $\iota_i(s) \in \iota_1[U] \cup \iota_2[U]$ iff $s \in U$.

(Atoms): Let $(s, t) \in Z$ witnessed by $(\iota_i(s), \iota_j(t)) \in Y$ where $i, j \in \{1, 2\}$. Since Y satisfies **(Atoms)**, we have $\iota_i(s) \in \iota_2[V(p)] \cup \iota_1[V(p)]$ iff $\iota_j(t) \in \iota_1[V(p)] \cup \iota_2[V(p)]$, and hence $s \in V(p)$ iff $t \in V(p)$.

(Coherence): We first note that if the pair (U, V) is Z-coherent, then $(\iota_1[U] \cup \iota_2[U], \iota_1[V] \cup \iota_2[V])$ is Y-coherent. Namely, take any pair $(\iota_i(s), \iota_j(t)) \in Y$.

By definition of Z, it follows that $(s,t) \in Z$. We now have $\iota_i(s) \in \iota_1[U] \cup \iota_2[U]$ iff $s \in U$ iff (by Z-coherence) $t \in V$ iff $\iota_j(t) \in \iota_1[V] \cup \iota_2[V]$. Furthermore, it is straightforward to show that for all $s \in S$, all $U \subseteq S$, and all $i \in \{1,2\}$:

$$U \in \nu(s) \iff (\iota_1[U] \cup \iota_2[U]) \in \nu'(\iota_i(s)) \tag{3}$$

$$U^c \in \nu(s) \iff (\iota_1[U] \cup \iota_2[U])^c \in \nu'(\iota_i(s)) \tag{4}$$

Coherence for Z now follows easily from (3), (4) and coherence for Y.

We state another basic fact about Δ-bisimilarity which can be proved using closure properties as for Proposition 1.

Proposition 6. *For all neighbourhood models M, the Δ-bisimilarity relation $\sim_\Delta$ on M is itself a Δ-bisimulation and an equivalence relation.*

As desired, Δ-bisimilar states cannot be distinguished with the $\mathcal{L}_\Delta$-language.

Proposition 7. *For all pointed neighbourhood models (M_1, s_1) and (M_2, s_2), if $(M_1, s_1) \sim_\Delta (M_2, s_2)$ then $(M_1, s_1) \equiv_\Delta (M_2, s_2)$.*

Proof. $(M_1, s_1) \sim_\Delta (M_2, s_2)$ iff $(M_1 + M_2, \iota_1(s_1)) \sim_\Delta^{\mathsf{betw}} (M_1 + M_2, \iota_2(s_2))$. Since the inclusion morphisms preserve truth, we have for all $\mathcal{L}_\Delta$-formulas φ that $M_1, s_1 \models \varphi$ iff $(M_1 + M_2), \iota_1(s_1) \models \varphi$, and similarly for M_2, s_2. Hence it suffices to prove that for all models M, $(M,s) \sim_\Delta^{\mathsf{betw}} (M,s')$ implies $(M,s) \equiv_\Delta (M,s')$.

So assume that Z is a Δ-bisimulation on a model M. We prove that for all formulas $\varphi \in \mathcal{L}_\Delta$ and all $(s,s') \in Z$, $M, s \models \varphi$ iff $M, s' \models \varphi$, by induction on φ. The base case $\varphi = p$ holds by (**Atoms**). The Boolean cases are routine, so lets turn to the case where $\varphi = \Delta\psi$. By induction hypothesis, we have for all $(x,y) \in Z$, $x \in [\![\psi]\!]_M$ iff $y \in [\![\psi]\!]_M$. That is, the pair $([\![\psi]\!]_M, [\![\psi]\!]_M)$ is Z-coherent. As Z is a Δ-bisimulation, it follows that for all $(s,s') \in Z$, ($[\![\psi]\!]_M \in \nu(s)$ or $[\![\psi]\!]_M^c \in \nu(s)$) iff ($[\![\psi]\!]_M \in \nu'(s')$ or $[\![\psi]\!]_M^c \in \nu'(s')$), that is, $M, s \models \Delta\psi$ iff $M, s' \models \Delta\psi$.

As with the standard notions of Kripke and neighbourhood bisimulations, $\mathcal{L}_\Delta$-equivalence does not always imply Δ-bisimilarity. Neither does $\mathcal{L}_\Delta$-equivalence imply o-Δ-bisimilarity as shown in [7, Example 3.10]. The same example shows that also $\mathcal{L}_\Delta$-equivalence, does not imply nbh-Δ-bisimilarity due to Propositions 3(2) and 5(2). However, a converse to Proposition 7 can be proved for an appropriate notion of saturated models following a similar line of reasoning as in [9, Sect. 4.1]. To this end, we introduce Δ-morphisms and Δ-congruences. They will play the part of neighbourhood morphisms and congruences from [9].

Definition 10 (Δ-morphisms and Δ-congruences). *Let $M = (S, \nu, V)$ and $M' = (S', \nu', V')$ be neighbourhood models. A function $f\colon S \to S'$ is a* Δ-morphism from M to M' *if its graph $Gr(f)$ is a Δ-bisimulation. A relation is a* Δ-congruence *if it is the kernel of some Δ-morphism.*

It is natural to ask whether Δ-morphisms are a generalisation of neighbourhood morphisms (cf. Definition 7). This is indeed the case.

Lemma 4. *Every neighbourhood morphism is a Δ-morphism.*

As a step towards showing that Δ-congruences are Δ-bisimulations, we show that we can take quotients with respect to Δ-bisimulations that are also equivalence relations.

Proposition 8 (Δ-quotient). *Let $M = (S, \nu, V)$ be a neighbourhood model and let Z be a Δ-bisimulation on M which is also an equivalence relation, i.e., for all Z-closed $U \subseteq S$ and all $(s,t) \in Z$,*

$$(U \in \nu(s) \text{ or } U^c \in \nu(s)) \iff (U \in \nu(t) \text{ or } U^c \in \nu(t)). \qquad (\dagger)$$

*We define the Δ-*quotient of M by Z *as the model $M_Z = (S_Z, \nu_Z, V_Z)$ where $S_Z = \{[s] \mid s \in S\}$ is the set of Z-equivalence classes, $V_Z(p) = \{[s] \mid s \in V(p)\}$, and*

$$\nu_Z([s]) = \{U_Z \subseteq S_Z \mid q^{-1}[U_Z] \in \nu(s) \text{ or } q^{-1}[U_Z]^c \in \nu(s)\}.$$

The quotient map $q\colon S \to S_Z$ given by $q(s) = [s]$ is a Δ-morphism, and $Z = \ker(q)$. Consequently, $(M,s) \sim_\Delta^{\mathsf{betw}} (M_Z, [s])$.

We can now show that Δ-congruences are indeed a special kind of Δ-bisimulations. This will be used to prove the Hennessy-Milner theorem in a moment.

Proposition 9. *Let $M = (S, \nu, V)$ be a neighbourhood model and Z a relation on S. Z is a Δ-congruence iff Z is an equivalence relation and a Δ-bisimulation.*

Proof. Assume $Z = \ker(f)$ for some Δ-morphism f from M to M'. Note that if U is Z-closed then $(U, f[U])$ is $Gr(f)$-coherent. Equation ($\dagger$) now easily follows from f being a Δ-morphism, and $Z = \ker(f)$. Conversely, if Z is an equivalence relation and a Δ-bisimulation on M, then we can form the Δ-quotient M_Z, and it follows that Z is a Δ-congruence.

Proposition 9 allows us to show a neighbourhood analogue of the fact that Kripke bisimilarity implies o-Δ-bisimilarity [7]. For neighbourhood models, the equivalence notion that matches the expressiveness of the language $\mathcal{L}_\Box$ is called behavioural equivalence [9]: Two pointed neighbourhood models (M,s) and (M',s') are *behaviourally equivalent* if there exists a neighbourhood model N and neighbourhood morphisms $f\colon M \to N$ and $f'\colon M' \to N$ such that $f(s) = f'(s')$.

Proposition 10. *Let M be a neighbourhood model, and s, t two states in M. If (M,s) and (M,t) are behaviourally equivalent then they are Δ-bisimilar.*

Proof. If (M,s) and (M,t) are behaviourally equivalent, then by [9, Proposition 3.20] the pair (s,t) is contained in a congruence, i.e. in the kernel of a neighbourhood morphism f. By Lemma 4, $\ker(f)$ is a Δ-congruence, which by Proposition 9, is a Δ-bisimulation on M, hence $(M,s) \sim_\Delta^{\mathsf{betw}} (M,t)$. Finally, it follows from Lemma 3 that $(M,s) \sim_\Delta (M,t)$.

Finally, we prove a Hennessy-Milner style theorem for an appropriate notion of saturated models which essentially comes from [9, Sect. 4.1].

Definition 11 ($\mathcal{L}_\Delta$-saturated model). *Let $M = (S, \nu, V)$ be a neighbourhood model. A subset $X \subseteq S$ is* $\mathcal{L}_\Delta$-compact *if for all sets Φ of $\mathcal{L}_\Delta$-formulas, if any finite subset $\Phi' \subseteq \Phi$ is satisfiable in X, then Φ is satisfiable in X. M is* $\mathcal{L}_\Delta$-saturated*, if for all $s \in S$ and all $\equiv_\Delta$-closed neighbourhoods $X \in \nu(s)$, both X and X^c are $\mathcal{L}_\Delta$-compact.*

Theorem 1 (Hennessy-Milner).

1. *For all $\mathcal{L}_\Delta$-saturated neighbourhood models M, and all states s, t in M: $(M, s) \equiv_\Delta (M, t)$ iff $(M, s) \sim_\Delta^{\mathsf{betw}} (M, t)$.*
2. *If $\mathbf{N}$ is a class of neighbourhood models in which the disjoint union of any two models is $\mathcal{L}_\Delta$-saturated, then for all M, M' in $\mathbf{N}$,*

$$(M, s) \equiv_\Delta (M', s') \quad \textit{iff} \quad (M, s) \sim_\Delta (M', s').$$

Proof. Due to space limitations we only provide an outline. *Item 1:* Can be proved using the same line of argumentation as in the proofs of Lemma 4.3, Lemma 4.5 and Proposition 4.6 of [9]. More precisely, we can show for any neighbourhood model $M = (S, \nu, V)$: (i) If all $\equiv_\Delta$-coherent neighbourhoods $X \in \nu(s)$ are $\mathcal{L}_\Delta$-definable then $\equiv_\Delta$ is a Δ-congruence. (ii) If M is $\mathcal{L}_\Delta$-saturated then for all $X \subseteq S$, X is $\equiv_\Delta$-coherent iff X is $\mathcal{L}_\Delta$-definable. The theorem follows from items (i) and (ii) together with Proposition 9.

Item 2: $(M, s) \equiv_\Delta (M', s')$ implies $(M + M', s) \equiv_\Delta (M + M', s')$ since the inclusion morphisms are Δ-bisimulations. By item 1, $(M + M', s) \sim_\Delta^{\mathsf{betw}} (M + M', s')$, hence by definition, $(M, s) \sim_\Delta (M', s')$.

As finite neighbourhood models are clearly $\mathcal{L}_\Delta$-saturated, we have an immediate corollary.

Corollary 1. *Over the class of finite neighbourhood models, $\mathcal{L}_\Delta$-equivalence implies Δ-bisimilarity.*

Frame Class (un)definability. We now use Δ-bisimulations to demonstrate that $\mathcal{L}_\Delta$ is too weak to define some well-known frame classes. These results were already proved in [6, Proposition 7], but without the use of a bisimulation argument.

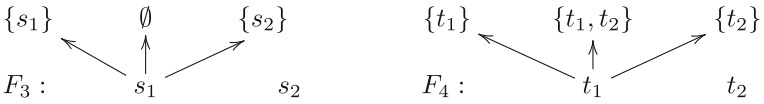
A frame class $\mathbf{F}$ *is $\mathcal{L}_\Delta$-definable* if there is a set $\Phi \subseteq \mathcal{L}_\Delta$ such that for all frames F, $F \in \mathbf{F}$ iff $F \models \Phi$.

Let $\mathbf{M}$ be the class of (monotone) neighbourhood frames (S, ν) in which $\nu(s)$ is closed under supersets, for all $s \in S$. Let $\mathbf{C}$ be the class of neighbourhood frames (S, ν) in which $\nu(s)$ is closed under intersections, for all $s \in S$.

Example 2. Consider the neighbourhood frames shown here:

in particular, $\nu_1(s_2) = \nu_2(t_2) = \emptyset$. It can easily be checked that $Z = \{(s_1, t_1), (s_2, t_2)\}$ is a Δ-bisimulation. Note that $F_1 \in \mathbf{M}$, but $F_2 \notin \mathbf{M}$.

Example 3. Consider the following neighbourhood frames:



in particular, $\nu_3(s_2) = \nu_4(t_2) = \emptyset$. It can easily be checked that $Z = \{(s_1, t_1), (s_2, t_2)\}$ is a Δ-bisimulation. Note that $F_3 \in \mathbf{C}$, but $F_4 \notin \mathbf{C}$.

Proposition 11. *The frame classes* $\mathbf{M}$ *and* $\mathbf{C}$ *are not definable in* $\mathcal{L}_\Delta$.

Proof. Example 2 shows that $\mathbf{M}$ is not $\mathcal{L}_\Delta$-definable, since suppose towards a contradiction that $\Phi \subseteq \mathcal{L}_\Delta$ defines $\mathbf{M}$. Then $F_1 \models \Phi$ and $F_2 \not\models \Phi$. Hence there is a valuation V_2 on F_2, a state t_j in F_2 and a $\varphi \in \Phi$ such that $(F_2, V_2), t_j \not\models \varphi$. We define a valuation V_1 on F_1 by $s_i \in V_1(p)$ iff $t_i \in V_2(p)$ for $i = 1, 2$ and all $p \in \mathsf{AtProp}$. It follows that $((F_1, V_1), s_i) \sim_\Delta ((F_2, V_2), t_i)$ for $i = 1, 2$, and hence that $(F_1, V_1), s_j \not\models \varphi$, which implies that $F_1 \not\models \Phi$, a contradiction.

Similarly, Example 3 can be used to show that $\mathbf{C}$ is not $\mathcal{L}_\Delta$-definable.

5 Characterisation Results

We first recall the basic definition of an ultrafilter. Let S be a non-empty set. An *ultrafilter* over S is a collection of sets $\mathfrak{u} \subseteq \mathcal{P}(S)$ satisfying (i) $S \in \mathfrak{u}$ and $\emptyset \notin \mathfrak{u}$; (ii) $U_1, U_2 \in \mathfrak{u}$ implies $U_1 \cap U_2 \in \mathfrak{u}$; (iii) $U_1 \in \mathfrak{u}$ and $U_1 \subseteq U_2 \subseteq S$ implies $U_2 \in \mathfrak{u}$; and (iv) for all $U \subseteq S$ we have $U \in \mathfrak{u}$ or $U^c \in \mathfrak{u}$.

The collection of all ultrafilters over S will be denoted by $Ult(S)$. For $s \in S$, the principal ultrafilter generated by s is $\mathfrak{u}_s = \{U \subseteq S \mid s \in U\}$.

Definition 12 (Ultrafilter extension [9]). *Let* $M = (S, \nu, V)$ *be a neighbourhood model. The* ultrafilter extension of M *is the triple* $M^{ue} = (Ult(S), \nu^{ue}, V^{ue})$ *where* $V^{ue}(p) = \{\mathfrak{u} \in Ult(S) \mid V(p) \in \mathfrak{u}\}$ *and* $\nu^{ue} : Ult(S) \to \mathcal{P}(\mathcal{P}(Ult(S)))$ *is defined by*

$$\nu^{ue}(\mathfrak{u}) = \{\hat{U} \subseteq Ult(S) \mid U \subseteq S, \Box(U) \in \mathfrak{u}\}$$

where $\Box(U) = \{s \in S \mid U \in \nu(s)\}$ *and* $\hat{U} = \{\mathfrak{v} \in Ult(S) \mid U \in \mathfrak{v}\}$.

Lemma 5. *Let* (M, s) *be a pointed neighbourhood model. Then,* M^{ue} *is an* $\mathcal{L}_\Delta$*-saturated model and* $(M, s) \equiv_\Delta (M^{ue}, \mathfrak{u}_s)$.

Proof. Since $\mathcal{L}_{\Delta}$ can be seen as a fragment of $\mathcal{L}_{\Box}$, [9, Lemma 4.24] ensures that $(M, s) \equiv_{\Delta} (M^{ue}, \mathfrak{u}_s)$ and [9, Proposition 4.25] ensures that M^{ue} is $\mathcal{L}_{\Delta}$-saturated.

As in the $\mathcal{L}_{\Box}$ case, modal $\mathcal{L}_{\Delta}$-equivalence in a model implies Δ-bisimilarity in the ultrafilter extension. (Apply Lemma 5 and Theorem 1.)

Proposition 12. *Let M be a neighbourhood model and s, t states in M. Then, $(M, s) \equiv_{\Delta} (M, t)$ implies $(M^{ue}, \mathfrak{u}_s) \sim_{\Delta} (M^{ue}, \mathfrak{u}_t)$.*

We are now ready to prove the characterisation theorems.

Theorem 2. *An $\mathcal{L}_{\Box}$-formula is equivalent to an $\mathcal{L}_{\Delta}$-formula over the class of neighbourhood models iff it is invariant under Δ-bisimulation.*

Proof. This can be proved analogously to the characterisation result [7, Theorem 4.4] using the above notions of $\mathcal{L}_{\Delta}$-saturation and ultrafilter extensions for neighbourhood models, together with the compactness of classical modal logic (via strong completeness), cf. [2, Sect. 9.2]. The only minor difference is that we must first take disjoint unions before taking the ultrafilter extension.

In [9], a Van Benthem style characterisation theorem was given for classical modal logic with respect to a two-sorted first-order correspondence language $\mathcal{L}_1$. The two sorts $\mathbf{s}$ and $\mathbf{n}$ correspond to states and to neighbourhoods, respectively, and the basic idea of viewing a neighbourhood model as a first-order $\mathcal{L}_1$-structure is to encode the neighbourhood function ν as a relation $R_\nu \subseteq \mathbf{s} \times \mathbf{n}$ between states and neighbourhoods, and encode subsets via the (inverse) element-of relation $R_\ni \subseteq \mathbf{n} \times \mathbf{s}$ between neighbourhoods and states. The language $\mathcal{L}_1$ is a first-order language with equality which contains a unary predicate symbol P (of sort $\mathbf{s}$) for each $p \in \mathsf{AtProp}$, a binary relation symbol N (interpreted by R_ν), and a binary relation symbol E (interpreted by $R_\ni$). A translation $(-)^\sharp \colon \mathcal{L}_{\Box} \to \mathcal{L}_1$ is defined recursively over the Boolean connectives and atomic propositions, and by $(\Box\varphi)^\sharp = \exists u \, (\, x\mathsf{N}u \ \wedge \ \forall y \, (\, u\mathsf{E}y \leftrightarrow \varphi^\sharp \,))$. We refer to [9, Sect. 5] for further details.

Theorem 3. *A first-order $\mathcal{L}_1$-formula is equivalent to an $\mathcal{L}_{\Delta}$-formula over the class of neighbourhood models iff it is invariant under Δ-bisimulation.*

Proof. Let $\alpha \in \mathcal{L}_1$ be invariant under Δ-bisimulations. It follows from Lemma 4 that α is invariant under neighbourhood morphisms, and hence under behavioural equivalence. From the characterisation theorem [9, Theorem 5.5] it follows that α is equivalent to $\varphi^\sharp$ for some formula $\varphi \in \mathcal{L}_{\Box}$ which is necessarily also invariant under Δ-bisimulations. Hence by our Theorem 2, φ is equivalent to ψ^t for some $\psi \in \mathcal{L}_{\Delta}$.

6 Discussion and Future Work

We proposed a notion of contingency bisimulation on neighbourhood models, we related it to an existing notion of contingency bisimulation on Kripke models,

and also provided the characterization of (neighbourhood) contingency logic as a fragment of the modal logic of necessity, and of first-order logic. Our work contributes to a research program aiming at generalizing *knowing that* to *knowing whether, knowing how, knowing value*, etc. [19], including weaker modal notions than knowledge.

In [8], the $\mathcal{L}_{\Delta}$-theory of all Kripke frames was axiomatized by the logic CL (going back to [10–12,20]). We observed (below (2)) that CL is sound and complete with respect to the class of augmented neighbourhood frames (which answers an open question in [7]). In [7] an axiomatization CCL of classical contingency logic (i.e., the $\mathcal{L}_{\Delta}$-theory of all neighbourhood frames) is also given. This raises the questions of what the axiomatizations are of monotone contingency logic and regular contingency logic. Proposition 11 means that one cannot fill these gaps with the axioms $\Delta\varphi \to \Delta(\varphi \to \psi) \vee \Delta(\neg\varphi \to \chi)$ and $\Delta(\psi \to \varphi) \wedge \Delta(\neg\psi \to \varphi) \to \Delta\varphi$ that are in CL but not in CCL. So these questions remain open.

The **(Coherence)** condition in our definition of Δ-bisimulation is a non-local property, since one needs to check all Z-coherent pairs, so over large Kripke models the Δ-Zig and Δ-Zag conditions of o-Δ-bisimulations will be easier to check. As we proved that Δ-bisimilarity coincides with o-Δ-bisimilarity, one can view the Δ-Zig and Δ-Zag conditions as a back-forth characterisation of Δ-bisimilarity over Kripke models. We would like to find local zig-zag conditions also for Δ-bisimilarity over neighbourhood models.

The notion of Δ-bisimulation was based on the semantics of the modality Δ. It has a natural generalisation to the framework of coalgebraic modal logic [3,15]. Many of our results hold at this general coalgebraic level. We are preparing a separate paper in which the coalgebraic perspective will be worked out.

Acknowledgments. Zeinab Bakhtiari and Hans van Ditmarsch gratefully acknowledge support from European Research Council grant EPS 313360. We thank Jie Fan, Yanjing Wang and the anonymous referees for their comments which helped improve the paper substantially.

References

1. Brogan, A.P.: Aristotle's logic of statements about contingency. Mind **76**(301), 49–61 (1967)
2. Chellas, B.F.: Modal Logic, An Intoduction. Cambridge University Press, Cambridge (1980)
3. Cirstea, C., Kurz, A., Pattinson, D., Schröder, L., Venema, Y.: Modal logics are coalgebraic. Comput. J. **54**(1), 31–41 (2008)
4. Fan, J.: Logical studies for non-contingency operator. Ph.D. thesis, Peking University (2015). (in Chinese)
5. Fan, J.: A note on non-contingency logic (manuscript) (2016). https://www.researchgate.net/publication/305091939
6. Fan, J., van Ditmarsch, H.: Neighborhood contingency logic. In: Banerjee, M., Krishna, S.N. (eds.) ICLA 2015. LNCS, vol. 8923, pp. 88–99. Springer, Heidelberg (2015). doi:10.1007/978-3-662-45824-2_6

7. Fan, J., Wang, Y., van Ditmarsch, H.: Almost necessary. In: Proceedings of 10th Advances in Modal Logic (AiML), pp. 178–196 (2014)
8. Fan, J., Wang, Y., van Ditmarsch, H.: Contingency and knowing whether. Rev. Symbolic Logic **8**(1), 75–107 (2015)
9. Hansen, H.H., Kupke, C., Pacuit, E.: Neighbourhood structures: bisimilarity and basic model theory. Logical Methods Comput. Sci. **5**(2) (2009). (paper 2)
10. van der Hoek, W., Lomuscio, A.: A logic for ignorance. Electron. Notes Theor. Comput. Sci. **85**(2), 117–133 (2004)
11. Humberstone, L.: The logic of non-contingency. Notre Dame J. Formal Logic **36**(2), 214–229 (1995)
12. Kuhn, S.: Minimal non-contingency logic. Notre Dame J. Formal Logic **36**(2), 230–234 (1995)
13. Montague, R.: Universal grammar. Theoria **36**, 373–398 (1970)
14. Montgomery, H., Routley, R.: Contingency and non-contingency bases for normal modal logics. Logique et Analyse **9**, 318–328 (1966)
15. Pattinson, D.: Coalgebraic modal logic: soundness, completeness and decidability of local consequence. Theor. Comput. Sci. **309**(1–3), 177–193 (2003)
16. Pizzi, C.: Contingency logics and propositional quantification. Manuscrito **22**(2), 283 (1999)
17. Scott, D.: Advice on modal logic. In: Lambert, K. (ed.) Philosophical Problems in Logic: Some Recent Developments, pp. 143–173. Kluwer, Dordrecht (1970)
18. Steinsvold, C.: A note on logics of ignorance and borders. Notre Dame J. Formal Logic **49**(4), 385–392 (2008)
19. Wang, Y.: Beyond knowing that: a new generation of epistemic logics. In: van Ditmarsch, H., Sandu, G. (eds.) Jaakko Hintikka on Knowledge and Game Theoretical Semantics. Outstanding Contributions to Logic. Springer (2016, to appear)
20. Zolin, E.: Completeness and definability in the logic of noncontingency. Notre Dame J. Formal Logic **40**(4), 533–547 (1999)

The Complexity of Finding Read-Once NAE-Resolution Refutations

Hans Kleine Büning[1], Piotr Wojciechowski[2(✉)], and K. Subramani[2]

[1] Universität Paderborn, Paderborn, Germany
kbcsl@uni-paderborn.de
[2] LCSEE, West Virginia University, Morgantown, WV, USA
pwojciec@mix.wvu.edu, k.subramani@mail.wvu.edu

Abstract. In this paper, we analyze boolean formulas in conjunctive normal form (CNF) from the perspective of read-once resolution (ROR) refutation. A read-once (resolution) refutation is one in which each input clause is used at most once. It is well-known that read-once resolution is not **complete**, i.e., there exist unsatisfiable formulas for which no read-once resolution exists. Likewise, the problem of checking if a 3CNF formula has a read-once refutation is **NP-complete**. This paper is concerned with a variant of satisfiability called Not-All-Equal Satisfiability (NAE-Satisfiability). NAE-Satisfiability is the problem of checking whether an arbitrary CNF formula has a satisfying assignment in which at least one literal in each clause is set to **false**. It is well-known that NAE-satisfiability is **NP-complete**. Clearly, the class of CNF formulas which are NAE-satisfiable is a proper subset of the class of satisfiable CNF formulas. It follows that traditional resolution cannot always find a proof of NAE-unsatisfiability. Thus, traditional resolution is not a **sound** procedure for checking NAE-satisfiability. In this paper, we introduce a variant of resolution called NAE-resolution, which is a sound and complete procedure for checking NAE-satisfiability in CNF formulas. We focus on a variant of NAE-resolution called read-once NAE-resolution, in which each input clause can be part of at most one NAE-resolution step. Our principal result is that read-once NAE-resolution is a sound and complete procedure for checking the NAE-satisfiability of 2CNF formulas; we also provide a polynomial time algorithm to determine the shortest read-once NAE-resolution of a 2CNF formula. Finally, we establish that the problem of checking whether a 3CNF formula has a read-once NAE-resolution is **NP-complete**.

Keywords: Read-once · NAE-SAT · Refutation · Optimal length refutation

P. Wojciechowski—This research is supported in part by the National Science Foundation under Award CCF-0827397.

K. Subramani—This work was supported by the Air Force Research Laboratory under US Air Force contract FA8750-16-3-6003. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 64–76, 2017.
DOI: 10.1007/978-3-662-54069-5_6

1 Introduction

This paper is concerned with techniques for checking Not-All-Equal (NAE) satisfiability of propositional formulas in Conjunctive Normal Form (CNF). Briefly, the NAE-SAT problem is concerned with checking if a CNF formula has a satisfying assignment in which each clause has at least one literal set to **false**. It is well-known that the NAE-satisfiability problem for 3CNF formulas (also called NAE3SAT) is **NP-complete** [10]. Indeed, the problem remains **NP-complete**, even when all the literals in each clause are positive. The problem can be solved in polynomial time, when there are at most two literals per clause [7,8].

It is not hard to see that the class of CNF formulas which are NAE-satisfiable is a proper subset of CNF formulas which are satisfiable in the ordinary sense. Therefore, proof systems for satisfiability may not be **sound** for checking NAE-satisfiability. Indeed, this is the case with resolution refutation [9], which is **complete** for NAE-satisfiability but not sound. In other words, if a resolution refutation exists for a CNF formula, then the formula is definitely NAE-unsatisfiable (since it is unsatisfiable). However, if a refutation does not exist for a formula, then it may still be NAE-unsatisfiable. In this paper, we design a new resolution scheme called NAE-resolution which is simultaneously **sound** and **complete** for the problem of checking NAE-satisfiability in CNF formulas.

Propositional proof complexity is concerned with lengths of proofs (alternatively refutations) in propositional logic [1]. In order to discuss lengths of proofs, it is vital that we have a concrete proof system in mind [11]. Several proof systems have been discussed in the literature including Frege Systems, Extended Frege Systems, Resolution and so on. The notion of proof length in various proof systems is discussed in [2]. Observe that if it can be established that the length of *any* proof (refutation) of a contradiction must be exponential in the length of the input formula, then we have in fact separated the class **NP** from the class **coNP** [3].

Even if we focus on a particular proof system there exist several variants with different computational complexities. For instance, in case of resolution refutations, the commonly studied variants are tree-like proofs, dag-like proofs and read-once proofs [4]. Read-once refutations are the simplest from the conceptual perspective, since each clause (original or derived) can be used exactly once.

This paper focuses on a weak and **incomplete** proof system called read-once resolution. It is well-known that read-once resolution is an incomplete proof system [5]. Furthermore, even asking if an arbitrary unsatisfiable CNF formula has a read-once refutation is **NP-complete**. We design a variant of read-once resolution called read-once NAE-resolution which is sound but not complete.

The investigations of this paper are concerned with properties of read-once NAE-resolutions when applied to the problem of checking NAE-satisfiability in CNF formulas.

The principal contributions of this paper are as follows:

1. A proof of existence of read-once NAE-resolution refutations for every NAE-unsatisfiable 2CNF formula.
2. The design and analysis of a polynomial time algorithm for finding a read-once NAE-resolution refutation for a NAE-unsatisfiable 2CNF formula.
3. The design and analysis of a polynomial time algorithm for finding the shortest read-once NAE-resolution refutation for a NAE-unsatisfiable 2CNF formula.
4. The design and analysis of a polynomial time algorithm for finding the minimum-weight read-once NAE-resolution refutation of a NAE-unsatisfiable 2CNF formula.
5. A proof that the problem of checking for the presence of read-once NAE-resolution refutations in a 3CNF formula is **NP-complete**.

The rest of this paper proceeds as follows: In Sect. 2, we cover some of the basic concepts necessary for our results. Section 3 formally defines the problems studied in this paper. Sections 4 and 5 describe the results that we have obtained. Finally, in Sect. 6 we summarize our results and describe avenues of future research.

2 NOT-ALL-EQUAL Satisfiability

We assume that the reader is familiar with the basic concepts and terminology of propositional logic. A formula α in CNF (conjunctive normal form) is a conjunction of clauses. Each clause in α is disjunction of literals written as $(L_1 \vee \ldots \vee L_n)$ or $(L_1, \ldots, L_n)$. A literal is a propositional variable x or its negation, $\neg x$. Let $\phi = (L_1, \ldots, L_n)$ be a clause, then ϕ^c is the clause $(\neg L_1, \ldots, \neg L_n)$.

We now recall some definitions with respect to NAE-satisfiability.

Definition 1. *A clause is* NAE-satisfied *by a truth assignment* $\mathbf{v}$*, if at least one literal in the clause is assigned a value of* **false** *and one literal is assigned a value of* **true**.

Definition 2. *A CNF formula* ϕ*, is* NAE-satisfiable *if there exists a truth assignment* $\mathbf{v}$ *such that every clause of* ϕ *is NAE-satisfied.*

The class of NAE-satisfiable formulas is denoted as NAE-SAT.

Note that, if a truth assignment $\mathbf{v}$ NAE-satisfies a formula in CNF, then so does $\neg\mathbf{v}$. We now compare NAE-satisfiability to regular satisfiability.

Lemma 1. *Let* ϕ *be a formula in CNF. We have that* $\phi \in$ *NAE-SAT if and only if* $\phi \cup \phi^c \in$ *SAT, where* $\phi^c := \{(\neg L_1, \ldots, \neg L_t) : (L_1, \ldots, L_t) \in \phi\}$.

Proof. Let ϕ be NAE-satisfied by the truth assignment $\mathbf{v}$. Thus, every clause ϕ_i of ϕ contains a literal L_i and a literal K_i for which $v(L_i) = \mathbf{true}$ and $v(K_i) = \mathbf{false}$. Hence, $\mathbf{v}$ satisfies both ϕ_i and ϕ_i^c.

Let $\phi \wedge \phi^c$ be satisfied by the truth assignment, $\mathbf{v}$. Thus, the clause ϕ_i contains a literal L_i such that $v(L_i) = \mathbf{true}$. Similarly, ϕ_i^c contains a literal K_i such that $v(K_i) = \mathbf{true}$. By construction, ϕ_i contains the literal $\neg K_i$. Thus, under truth assignment $\mathbf{v}$, ϕ_i contains both a **true** and a **false** literal. This means that $\mathbf{v}$ NAE-satisfies ϕ. □

Lemma 1 immediately leads to the observation that deciding whether a formula ϕ is in NAE-SAT can be performed by means of resolution on $\phi \cup \phi^c$. Instead of adding the complementary clause ϕ_i^c in the beginning we extend the resolution calculus with a new rule. This rule generates complementary clauses on demand.

We now define the inference rules for NAE-resolution.

Definition 3. *Let L_i and K_j be literals, and let x be a variable.* NAE-resolution *consists of the following inference rules:*

1. *Resolution:*
$$\frac{(L_1, \ldots, L_t, x) \qquad (\neg x, K_1, \ldots, K_r)}{(L_1, \ldots, L_t, K_1, \ldots, K_r)}.$$
We denote this NAE-resolution step as:
$$(L_1, \ldots, L_t, x), (\neg x, K_1, \ldots, K_r) \mid\!\!\!\frac{1}{RES} (L_1, \ldots, L_t, K_1, \ldots, K_r).$$
2. *NAE-extension:*
$$\frac{(L_1, \ldots, L_t)}{(\neg L_1, \ldots, \neg L_t)}.$$
We denote this NAE-resolution step as:
$$(L_1, \ldots, L_t) \mid\!\!\!\frac{}{NAE\text{-}ext} (\neg L_1, \ldots, \neg L_t).$$

We write $\phi \mid\!\!\!\frac{}{NAE\text{-}Res} \pi$ to indicate that the clause π can be derived from ϕ by NAE-resolution. Similarly, we write $\phi \mid\!\!\!\frac{}{RES} \pi$ to indicate that the clause π can be derived from ϕ by regular resolution.

Note that the NAE-extension rule is what allows us to simultaneously negate all the literals of a clause.

It can easily be seen that NAE-resolution preserves NAE-satisfiability. That is, if the original formula is NAE-satisfiable, then any formula we get by adding the resolvents and the clauses introduced by the NAE-extension rule is NAE-satisfiable. The following theorem summarizes the relationship between the various calculi.

Theorem 1. *Let ϕ be a formula in CNF. We have that the following propositions are equivalent:*

1. *$\phi \notin$ NAE-SAT.*
2. *$\phi \cup \phi^c \notin$ SAT.*

3. $\phi \cup \phi^c \vdash_{RES} \sqcup$.
4. there exists some literal $L : \phi \vdash_{NAE\text{-}Res} L$.
5. $\phi \vdash_{NAE\text{-}Res} \sqcup$.

Proof. The proof of this is broken up as follows:

1. $\phi \notin$ NAE-SAT if and only if $\phi \cup \phi^c \notin$ SAT:

   This was already proved in Lemma 1.
2. $\phi \cup \phi^c \notin$ SAT if and only if $\phi \cup \phi^c \vdash_{RES} \sqcup$:
   This is a trivial consequence of the definition of SAT and the completeness of resolution.
3. $\phi \notin$ NAE-SAT if and only if there exists some literal $L : \phi \vdash_{NAE\text{-}Res} L$:
   See Theorem 4 for proof.
4. There exists some literal $L : \phi \vdash_{NAE\text{-}Res} L$ if and only if $\phi \vdash_{NAE\text{-}Res} \sqcup$:
   If $\phi \vdash_{NAE\text{-}Res} L$, then $\phi \vdash_{NAE\text{-}Res} \neg L$ since $L \vdash_{NAE\text{-}ext} \neg L$. Thus, we have that $\phi \vdash_{NAE\text{-}Res} \sqcup$ since $L, \neg L \vdash^{1}_{RES} \sqcup$.
   If $\phi \vdash_{NAE\text{-}Res} \sqcup$, then the final resolution step must be $L, \neg L \vdash^{1}_{RES} \sqcup$ for some literal L. Thus, $\phi \vdash_{NAE\text{-}Res} L$. □

3 Read-Once Proofs and NAE-SAT

Let ϕ be a CNF formula and let π be a clause. A read-once resolution derivation of π, $\phi \vdash_{ROR} \pi$, is a resolution derivation, such that in each resolution step we remove the parent clauses from the current set of clauses and add the resolvent. Let *ROR* be the set of CNF formulas for which a read-once resolution refutation exists. It has been shown [5] that the problem of determining if a CNF formula is in ROR is **NP-complete**.

Definition 4. *Let* ϕ *be a CNF formula and* π *a clause. A* read-once NAE-resolution derivation *of* π*,* $\phi \vdash_{RO\text{-}NAE\text{-}Res} \pi$*, is a derivation using the resolution rule and/or NAE-extension rule. In the case of resolution, we delete the parent clauses and add the resolvent. In the case of the extension rule,* $\sigma \vdash_{NAE\text{-}ext} \sigma^c$*, we remove the clause* σ *and add* σ^c*.*

There are two ways to check for the existence of a read-once proof of NAE-unsatisfiability. Given a CNF formula ϕ, we can check if $\phi \cup \phi^c$ has a read-once refutation. Alternatively, we can ask whether ϕ has a read-once NAE-resolution refutation (under the resolution and NAE-extension rules).

These methods of checking for read-once refutations correspond to the following sets of CNF formulas:

1. ROR-NAE := $\{\phi \in CNF \mid \ \phi \wedge \phi^c \vdash_{ROR} \sqcup\}$.
2. RO-NAE-RES := $\{\phi \in CNF \mid \ \exists$ literal $L : \phi \vdash_{RO\text{-}NAE\text{-}Res} L\}$.

In this paper, we study the problems of determining if certain forms of CNF formulas, specifically 2CNF and 3CNF, are subsets of to these classes.

We now define the length of a read-once NAE-resolution refutation.

Definition 5. *The* length *of a read-once NAE-resolution refutation is the number of NAE-resolution steps in that refutation.*

This lets us define the concept of a shortest read-once NAE-resolution refutation.

Definition 6. *The* shortest *read-once NAE-resolution refutation of a system ϕ, is the read-once NAE-resolution resolution with the fewest steps.*

We now show that a formula in 2CNF, ϕ, has a read-once NAE-resolution refutation if and only if $\phi \cup \phi^c$ has a read-once resolution refutation.

Theorem 2. *$\phi \in$ ROR-NAE if and only if $\phi \in$ RO-NAE-RES.*

Proof. Let ϕ be in ROR-NAE. There exists a read-once resolution refutation $\phi \cup \phi^c \mid\!\!\frac{}{}_{ROR}\ \sqcup$. The final step of this refutation must be resolving a pair of one literal clauses to derive the empty clause. Thus, we must have that, for some literal L, $\phi \cup \phi^c \mid\!\!\frac{}{}_{ROR}\ L$. Let D be the shortest such resolution derivation. Thus, there is no literal L' which can be derived by a shorter read-once resolution derivation.

By construction, every resolution step (except the last one) of D results in a two literal clause. Thus, we can restructure D so that each resolution step is of the form:

$$(L, x_i), (\neg x_i, x_j) \mid\!\!\frac{1}{}_{RES}\ (L, x_j).$$

Let π be a clause such that both π and π^c are used in D. Without loss of generality, we can assume that $\pi = (x_i, x_j)$, and that the restructured derivation uses π before it uses π^c. There are four cases we need to consider:

1. The resolution step involving π is $(L, \neg x_i), \pi \mid\!\!\frac{1}{}_{RES}\ (L, x_j)$ and the resolution step involving π^c is $(L, x_i), \pi^c \mid\!\!\frac{1}{}_{RES}\ (L, \neg x_j)$. Thus, there must be a sequence of resolution steps which produced (L, x_i) from $(L, \neg x_i)$. However, this means that the set of clauses used in these resolution steps can derive x_i. This contradicts our construction of D.
2. The resolution step involving π is $(L, \neg x_j), \pi \mid\!\!\frac{1}{}_{RES}\ (L, x_i)$ and the resolution step involving π^c is $(L, x_j), \pi^c \mid\!\!\frac{1}{}_{RES}\ (L, \neg x_i)$. Thus, there must be a sequence of resolution steps which produced (L, x_j) from $(L, \neg x_j)$. However, this means that the set of clauses used in these resolution steps can derive x_j. This contradicts our construction of D.
3. The resolution step involving π is $(L, \neg x_j), \pi \mid\!\!\frac{1}{}_{RES}\ (L, x_i)$ and the resolution step involving π^c is $(L, x_i), \pi^c \mid\!\!\frac{1}{}_{RES}\ (L, \neg x_j)$. Thus, D derives (L, x_i) twice. By removing the sequence of resolution steps between these two derivations of (L, x_i), we produce a shorter derivation of L. This contradicts our construction of D.

4. The resolution step involving π is $(L, \neg x_i), \pi \mid\!\!\frac{1}{RES}\; (L, x_j)$ and the resolution step involving π^c is $(L, x_j), \pi^c \mid\!\!\frac{1}{RES}\; (L, \neg x_i)$. Thus, D derives (L, x_j) twice. By removing the sequence of resolution steps between these two derivations of (L, x_j), we produce a shorter derivation of L. This contradicts our construction of D.

Thus, π and π^c cannot be both used in D.

We now have $\phi \mid\!\!\frac{}{NAE\text{-}Res}\; L$ as follows:

1. Apply the NAE-extension rule to every clause $\pi \in \phi$ such that π^c is used in D. Note, we are guaranteed that π is not used in D.
2. Derive L using the same resolution steps as D.

Now let ϕ be in RO-NAE-RES. For some literal L, there exists a read-once derivation $\phi \mid\!\!\frac{}{NAE\text{-}Res}\; L$. Let $\pi_1, \ldots, \pi_t$ be the set of clauses used in this read-once derivation. We can use the dual clauses $\pi_1^c, \ldots, \pi_t^c$ to derive $(\neg L)$. As a final step, we can use (L) and $(\neg L)$ to derive the empty clause. This forms a read-once resolution refutation of $\phi \cup \phi^c$. □

4 NAE-2SAT

In this section, we show that read-once NAE-resolution refutation is both sound and complete for formulas in 2CNF. We also show that the problem of finding the shortest read-once NAE-resolution refutation is in **P**.

Theorem 3. *Let ϕ be a formula in 2CNF. We have $\phi \notin$ NAE-SAT if and only if $\phi \in$ ROR-NAE, and a refutation can be found in quadratic time.*

Proof. Let ϕ be a 2CNF formula that is not in NAE-SAT. If ϕ contains a unit clause, say (x), then $\{(x), (\neg x)\} \subseteq \phi \cup \phi^c$. We have that $(x), (\neg x) \mid\!\!\frac{1}{RES}\; \sqcup$. This is clearly a ROR-NAE-SAT refutation. Thus, we assume that ϕ contains no unit clause.

From $\phi \cup \phi^c$, we create an implication graph, G, as follows:

1. For every variable x_i, we create the verticies x_i and $\bar{x}_i$.
2. For every clause $(L \vee K)$, we create the edges $\bar{L} \to K$ and $\bar{K} \to L$.

G contains a strongly connected component, say G_1, with a pair of complementary literals if and only if $\phi \cup \phi^c$ is unsatisfiable. Moreover, the computation of the strongly connected components and finding a complementary pair of literals can be performed in linear time. A formula ϕ is not in NAE-SAT if and only if $\phi \cup \phi^c$ is unsatisfiable. Thus, there exists a strongly connected component C in G that contains complementary literals.

Let $\bar{L}_0 \to L_1 \to L_2 \ldots L_m \to L_0$ be a shortest path in C between the complementary pair of literals L_0 and $\neg L_0$. For $i \neq j$ we have $L_i \neq L_j$ and $L_i \neq \neg L_j$, otherwise there would be a shorter path in C.

Thus, there is a read-once resolution derivation

$$(L_0 \vee L_1), (\neg L_1 \vee L_2), \ldots, (\neg L_m \vee L_0) \vdash_{ROR} L_0.$$

Since we are dealing with $\phi \cup \phi^c$, there are clauses $(\neg L_0 \vee \neg L_1), (L_1 \vee \neg L_2), \ldots,$ $(L_m \vee \neg L_0)$ in $\phi \cup \phi^c$. These clauses form a read-once resolution of $(\neg L_0)$. Moreover, the two sets of clauses have no clause in common, because the literals L_i for $i \neq 0$ are pairwise disjoint. Finally, we can resolve (L_0) and $(\neg L_0)$. Thus, we have a read-once resolution refutation for $\phi \cup \phi^c$. By Theorem 2, this corresponds to a ROR-NAE-SAT refutation of ϕ.

Since the computation of the strongly connected components includes deciding whether a complementary pair of literals exists costs linear time and finding a complementary pair with a shortest path costs for each variable again takes linear time, to construct a read-once resolution proof requires no more than quadratic time. □

4.1 Finding Shortest Proofs

Earlier in Sect. 4, we described an implication graph for checking the satisfiability of 2CNF formulas. We can construct a similar implication graph for checking the NAE-satisfiability of 2CNF formulas. We refer to this as the NAE-implication graph. The NAE-implication graph of a formula ϕ is equivalent to the implication graph of $\phi \cup \phi^c$.

Example 1. Consider the 2CNF formula.

$$(x_1, x_2)\ (x_2, x_3)\ (\neg x_3, x_4)$$

From this formula we can generate the NAE-implication graph in Fig. 1.

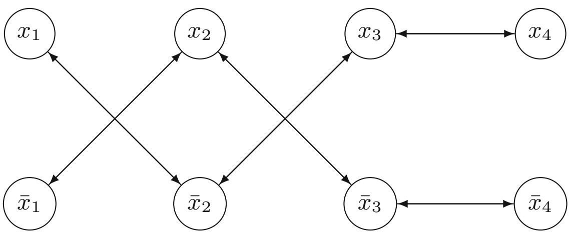


Fig. 1. Example NAE-implication graph

Theorem 4. *A CNF formula ϕ is* **not** *NAE-satisfiable if and only if the clause $\phi \vdash_{NAE\text{-}Res} (x_i)$ for some variable x_i.*

Proof. Assume that $\phi \vdash_{\overline{NAE\text{-}Res}} (x_i)$ for some variable x_i. We know that any assignment, $\mathbf{x}$, that NAE-satisfies ϕ must NAE-satisfy (x_i). However, the clause (x_i) has only one literal. Thus, it cannot be NAE-satisfied. This means that ϕ is not NAE-satisfiable.

Let ϕ be a CNF formula that is not NAE-satisfiable. We can construct the unsatisfiable formula $\phi' = \phi \cup \phi^c$ of CNF clauses.

Since ϕ' is unsatisfiable we can derive the clauses (x_i) and $(\neg x_i)$ for some variable x_i.

Let $(x_{j1}, \ldots, x_{jm}, x_k) \wedge (\neg x_k, x_{l1}, \ldots, x_{lm}) \vdash^1_{\overline{RES}} (x_{j1}, \ldots, x_{jm}, x_{l1}, \ldots, x_{lm})$ be the first step in the derivation of (x_i) from the clauses in ϕ'. We have four possibilities for the original clauses in ϕ.

1. $(x_{j1}, \ldots, x_{jm}, x_k), (\neg x_k, x_{l1}, \ldots, x_{lm}) \in \phi$:
   From the NAE-resolution rules we get:

$$(x_{j1}, \ldots, x_{jm}, x_k), (\neg x_k, x_{l1}, \ldots, x_{lm}) \vdash^1_{\overline{RES}} (x_{j1}, \ldots, x_{jm}, x_{l1}, \ldots, x_{lm}).$$

2. $(x_{j1}, \ldots, x_{jm}, x_k), (x_k, \neg x_{l1}, \ldots, \neg x_{lm}) \in \phi$:
   From the NAE-resolution rules we get:

$$(x_k, \neg x_{l1}, \ldots, \neg x_{lm}) \vdash_{\overline{NAE\text{-}ext}} (\neg x_k, x_{l1}, \ldots, x_{lm}).$$

$$(x_{j1}, \ldots, x_{jm}, x_k) \wedge (\neg x_k, x_{l1}, \ldots, x_{lm}) \vdash^1_{\overline{RES}} (x_{j1}, \ldots, x_{jm}, x_{l1}, \ldots, x_{lm}).$$

3. $(\neg x_{j1}, \ldots, \neg x_{jm}, \neg x_k), (\neg x_k, x_{l1}, \ldots, x_{lm}) \in \phi$:
   From the NAE-resolution rules we get:

$$(\neg x_{j1}, \ldots, \neg x_{jm}, \neg x_k) \vdash_{\overline{NAE\text{-}ext}} (x_{j1}, \ldots, x_{jm}, x_k).$$

$$(x_{j1}, \ldots, x_{jm}, x_k) \wedge (\neg x_k, x_{l1}, \ldots, x_{lm}) \vdash^1_{\overline{RES}} (x_{j1}, \ldots, x_{jm}, x_{l1}, \ldots, x_{lm}).$$

4. $(\neg x_{j1}, \ldots, \neg x_{jm}, \neg x_k), (x_k, \neg x_{l1}, \ldots, \neg x_{lm}) \in \phi$:
   From the NAE-resolution rules we get:

$$(x_k, \neg x_{l1}, \ldots, \neg x_{lm}) \vdash_{\overline{NAE\text{-}ext}} (\neg x_k, x_{l1}, \ldots, x_{lm}).$$

$$(\neg x_{j1}, \ldots, \neg x_{jm}, \neg x_k) \vdash_{\overline{NAE\text{-}ext}} (x_{j1}, \ldots, x_{jm}, x_k).$$

$$(x_{j1}, \ldots, x_{jm}, x_k) \wedge (\neg x_k, x_{l1}, \ldots, x_{lm}) \vdash^1_{\overline{RES}} (x_{j1}, \ldots, x_{jm}, x_{l1}, \ldots, x_{lm}).$$

In all four cases, $\phi \vdash_{\overline{NAE\text{-}Res}} (x_{j1}, \ldots, x_{jm}, x_{l1}, \ldots, x_{lm})$.

This same argument can be repeated for each subsequent derivation step. Thus, $\phi \vdash_{\overline{NAE\text{-}Res}} (x_i)$. □

Let ϕ be a CNF formula such that there is a read-once resolution refutation $\phi \cup \phi^c \vdash_{\overline{ROR}} \sqcup$. Starting with ϕ, we apply the NAE-extension rule and generate $\phi \cup \phi^c$. Next, we apply the resolution rule according to the read-once resolution refutation for $\phi \cup \phi^c$.

Now, suppose there is a derivation $\phi \vdash_{\overline{NAE\text{-}Res}} \sqcup$ in which the resolution operation is read-once and the extension rule is used at most once on either ϕ or ϕ^c.

We rearrange the derivation such that we first apply the extension rule and then the resolution rule.

Let $(\alpha \vee x), (\neg x \vee \beta) \vdash_{\overline{RES}}^{1} (\alpha \vee \beta)$ and $(\alpha \vee \beta) \vdash_{\overline{NAE\text{-}ext}} (\alpha^c \vee \beta^c)$ be an instance where the extension rule is used on a derived clause. We can replace these derivation steps with:

$(\alpha \vee x) \vdash_{\overline{NAE\text{-}ext}} (\alpha^c \vee \neg x)$, $(\neg x \vee \beta) \vdash_{\overline{NAE\text{-}ext}} (x \vee \beta^c)$, and $(\alpha^c \vee \neg x)(x \vee \beta^c) \vdash_{\overline{RES}}^{1} (\alpha^c \vee \beta^c)$.

This can be done repeatedly until the NAE-extension rule is applied to only the original clauses of the formula. Since the desired refutation starts with $\phi \cup \phi^c$, we can remove the instances of the NAE-extension rule to generate a proof of $\phi \cup \phi^c \vdash_{\overline{ROR}} \sqcup$.

To prove NAE-unsatisfiability we need to derive the clause (x_i). Thus, we need to find a path from $\bar{x}_i$ to x_i in the NAE-implication graph. Note that we do not need to also find a path from x_i to $\bar{x}_i$. Thus, we have the following theorem.

Theorem 5. *Let ϕ be a formula in 2-CNF without unit clauses. The following statements are equivalent:*

1. *ϕ is not in NAE-SAT.*
2. *$\phi \cup \{(\neg L_1, \neg L_2) : (L_1, L_2) \in \phi\} \vdash_{\overline{RES}} L$ for some literal L, and there is a derivation in which at most one of the clauses (L_1, L_2) or $(\neg L_1, \neg L_2)$ occurs.*

A decision procedure based on the representation as a graph solves the problem in linear time.

We show that, in the case of NAE-unsatisfiable 2CNF formulas, we always have a Read-Once NAE-resolution refutation.

Theorem 6. *If a 2CNF formula, ϕ, has a NAE-resolution derivation of (x_i), then it has a NAE-resolution derivation of (x_i) using only one literal more than once.*

Proof. Let G be the NAE-implication graph corresponding to ϕ. We know that $\phi \vdash_{\overline{NAE\text{-}Res}} (x_i)$ if and only if there exists a path from $\bar{x}_i$ to x_i in G. Let p denote this path. Let x_j be the first variable such that both x_j and $\bar{x}_j$ appear on p. We are guaranteed for this x_j to exist since both x_i and $\bar{x}_i$ appear on p. We can assume without loss of generality that x_j appears before $\bar{x}_j$. Thus, we can break p up as follows:

1. a path, p_1, from $\bar{x}_i$ to x_j,
2. a path, p_2, from x_j to $\bar{x}_j$,
3. and a path, p_3, from $\bar{x}_k$ to x_i.

This can be seen in Fig. 2.

By our choice of x_j, we know that for $k \neq j$, p_1 and p_2 together do not contain both x_k and $\bar{x}_k$. As a consequence of this no two edges in p_1 or p_2 correspond to the same constraint. Thus, p_2 corresponds to a read-once NAE-resolution derivation of $(\neg x_j)$ in which only the literal $\neg x_j$ appears twice.

We also have that p_1 is a literal once NAE-resolution derivation of (x_i, x_j) which has no literals in common with the NAE-resolution derivation corresponding to p_2. Combining these two yields a read-once NAE-resolution derivation of (x_i) in which only the literal $\neg x_j$ is used twice. □

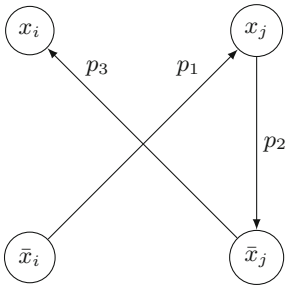


Fig. 2. Example of path p

Note that, in this NAE-resolution derivation the subpath p_2 from x_j to $\bar{x}_j$ is a proof of NAE-unsatisfiability by itself since it shows $(\neg x_j)$ which already enough to force x_j to be both **true** and **false**. Thus, we have the following corollary.

Corollary 1. *If a 2CNF formula, ϕ, has a NAE-resolution derivation of (x_i), then, for some x_j, there is a NAE-resolution derivation of (x_j) (or $(\neg x_j)$) using only the literal x_j (or $\neg x_j$) more than once.*

Corollary 1 provides us with a polynomial time algorithm to find the shortest read-once NAE-refutation of a 2CNF formula.

Algorithm 1. FIND-MINIMUM-NAE-REFUTATION

Function FIND-MINIMUM-NAE-REFUTATION (NAE-unsatisfiable 2CNF formula ϕ)

1: From ϕ, construct the NAE-implication graph G.
2: **for** (Each $i = 1 \ldots n$) **do**
3: Find the shortest path from $\bar{x}_i$ to x_i in G.
4: **end for**
5: **return** (The shortest of the located paths.)

Note that, we do not need to consider the paths from x_i to $\bar{x}_i$ since the existence of such a path means that there is a path of equal length from $\bar{x}_i$ to x_i.

This algorithm can be easily modified to solve the following problem:

Definition 7. *In the* minimum-weight *read-once NAE-resolution refutation problem each clause of ϕ is assigned a non-negative weight. The goal is to find a read-once NAE-resolution refutation with minimum total weight.*

To find the minimum-weight read-once NAE-resolution refutation for a 2CNF formula, we construct a weighted NAE-implication graph. In the weighted graph each edge is assigned the same weight as the corresponding 2CNF clause. We then run a modified version of Algorithm 1 on this weighted graph to find the minimum-weight path from $\bar{x_i}$ to x_i.

5 Read-Once NAE-resolution Refutation for 3CNF

Now we focus on applying NAE-resolution to formulas in 3CNF and show that the problem whether for a formula ϕ the formula $\phi \cup \phi^c$ has a read-once resolution refutation is **NP-complete**. Since ROR - the set of formulas in CNF for which a read-once resolution exists - is **NP-complete**, we see that ROR-NAE-3SAT is in **NP**. Therefore, we only have to show **NP-hardness**. This is done by a reduction to the problem whether a formula in 2CNF has a read-once resolution refutation (ROR-2CNF). The ROR-2CNF problem is **NP-complete** [6].

Theorem 7. *ROR-NAE-3SAT is* **NP-complete**.

Proof. Let ϕ be a 2CNF formula. We construct the 3CNF formula ϕ^* as follows:

1. For each variable x_i of ϕ, create the variable x_i for ϕ^*.
2. Create the variable x_0 for ϕ^*.
3. For each clause $\pi \in \phi$, create the clause $(\pi \vee x_0) \in \phi^*$.

We show that $\phi \in$ ROR-2CNF if and only if $\phi^* \in$ ROR-NAE-3SAT.

Assume that $\phi \in$ ROR-2CNF. A read-once resolution refutation $\phi \vdash_{ROR} \sqcup$ can easily be extended to the read-once NAE-resolution derivation $\phi^* \vdash_{ROR} x_0$. Thus, by Theorem 4, ϕ^* is in ROR-NAE-3SAT.

Now suppose that ϕ^* is in ROR-NAE-3SAT. We must show that ϕ has a read-once resolution refutation. We do this by showing that every resolution step done on the 3CNF clauses corresponds to a valid derivation on the 2CNF clauses.

We have the following cases:

1. $(x_i, x_j, x_0) \vdash_{NAE\text{-}ext} (\neg x_i, \neg x_j, \neg x_0)$: Both of these clauses correspond to the 2CNF clause (x_i, x_j). If (x_i, x_j) is satisfied, then both (x_i, x_j, x_0) and $(\neg x_i, \neg x_j, \neg x_0)$ are NAE-satisfied by setting x_0 to **false**.
2. $(x_i, x_j, x_0), (\neg x_k, \neg x_l, \neg x_0) \vdash^1_{RES} (x_i, x_j, \neg x_k, \neg x_l)$: This corresponds to the two CNF clauses $(x_i, x_j, \neg x_k, \neg x_l)$ and $(\neg x_i, \neg x_j, x_k, x_l)$. However, these are made redundant by the 2CNF clauses (x_i, x_j) and (x_k, x_l) which are already derivable from ϕ. Thus, no NAE-resolution refutation of ϕ^* performs a resolution step centered on x_0.
3. $(x_i, x_j, x_0), (\neg x_j, \neg x_k, x_0) \vdash^1_{RES} (x_i, \neg x_k, x_0)$: This corresponds to the resolution step $(x_i, x_j), (\neg x_j, \neg x_k) \vdash^1_{RES} (x_i, \neg x_k)$. Since $\phi \vdash_{RES} (x_i, x_j)$ and $\phi \vdash_{RES} (\neg x_j, \neg x_k)$, this is a valid derivation from ϕ.

Thus, all steps in the NAE-resolution refutation of the 3CNF formula correspond to steps used in the resolution refutation of the original 2CNF formula. Thus, ϕ^* has a read-once NAE-resolution refutation if and only if ϕ the has a read-once resolution refutation. ◻

6 Conclusion

In this paper, we introduced the notion of NAE-resolutions and show how they can be applied to the problem of checking NAE-satisfiability in CNF formulas. Prior to our work, the standard approach in the literature was to convert the NAE-satisfiability problem to simple satisfiability. Our principal contribution is showing that the problem of checking whether a 2CNF formula has a read-once NAE-resolution is in **P**. Furthermore, we showed that the problem of finding the optimal length read-once NAE-resolution is also in **P**.

References

1. Beame, P., Pitassi, T.: Propositional proof complexity: past, present, future. Bull. EATCS **65**, 66–89 (1998)
2. Buss, S.R.: Propositional proof complexity: an introduction. http://www.math.ucsd.edu/~sbuss/ResearchWeb/marktoberdorf97/paper.pdf
3. Cook, S.A., Reckhow, R.A.: On the lengths of proofs in the propositional calculus (preliminary version). In: Proceedings of the 6th Annual ACM Symposium on Theory of Computing, Seattle, Washington, USA, 30 April – 2 May 1974, pp. 135–148 (1974)
4. Harrison, J.: Handbook of Practical Logic and Automated Reasoning, 1st edn. Cambridge University Press, Cambridge (2009)
5. Iwama, K., Miyano, E.: Intractability of read-once resolution. In: Proceedings of the 10th Annual Conference on Structure in Complexity Theory (SCTC 1995), CA, USA, pp. 29–36. IEEE Computer Society Press, Los Alamitos, June 1995
6. Kleine Büning, H., Wojciechowski, P., Subramani, K.: On the computational complexity of read once resolution decidability in 2CNF formulas. https://arxiv.org/abs/1610.04523
7. Moore, C., Mertens, S.: The Nature of Computation, 1st edn. Oxford University Press, Oxford (2011)
8. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, New York (1994)
9. Robinson, J.A.: A machine-oriented logic based on the resolution principle. J. ACM **12**(1), 23–41 (1965)
10. Schaefer, T.: The complexity of satisfiability problems. In: Aho, A. (ed.) Proceedings of the 10th Annual ACM Symposium on Theory of Computing, pp. 216–226. ACM Press, New York City (1978)
11. Urquhart, A.: The complexity of propositional proofs. Bull. Symbolic Logic **1**(4), 425–467 (1995)

Knowing Values and Public Inspection

Jan van Eijck[1,2], Malvin Gattinger[1(✉)], and Yanjing Wang[3]

[1] ILLC, University of Amsterdam, Amsterdam, The Netherlands
malvin@w4eg.eu
[2] SEN1, CWI, Amsterdam, The Netherlands
[3] Department of Philosophy, Peking University, Beijing, China

Abstract. We present a basic dynamic epistemic logic of "knowing the value". Analogous to public announcement in standard DEL, we study "public inspection", a new dynamic operator which updates the agents' knowledge about the values of constants. We provide a sound and strongly complete axiomatization for the single and multi-agent case, making use of the well-known Armstrong axioms for dependencies in databases.

Keywords: Knowing what · Bisimulation · Public announcement logic

1 Introduction

Standard epistemic logic studies propositional knowledge expressed by "knowing that". However, in everyday life we talk about knowledge in many other ways, such as "knowing what the password is", "knowing how to swim", "knowing why he was late" and so on. Recently the epistemic logics of such expressions are drawing more and more attention (see [1] for a survey).

Merely reasoning about static knowledge is important but it is also interesting to study the changes of knowledge. Dynamic Epistemic Logic (DEL) is an important tool for this, which handles how knowledge (and belief) is updated by events or actions [2]. For example, extending standard epistemic logic, one can update the propositional knowledge of agents by making propositional announcements. They are nicely studied by public announcement logic [3] which includes reduction axioms to completely describe the interplay of "knowing that" and "announcing that". Given this, we can also ask: What are natural dynamic counterparts the knowledge expressed by other expressions such as knowing what, knowing how etc.? How do we formalize "announcing what"?

In this paper, we study a basic dynamic operation which updates the knowledge of the values of certain constants.[1] The action of *public inspection* is the knowing value counterpart of public announcement and we will see that it fits

[1] In this paper, by *constant* we mean something which has a single value given the actual situation. The range of possible values of a constant may be infinite. This terminology is motivated by first-order modal logic as it will become more clear later.

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 77–90, 2017.
DOI: 10.1007/978-3-662-54069-5_7

well with the logic of knowing value. As an example, we may use a sensor to measure the current temperature of the room. It is reasonable to say that after using the sensor you will know the temperature of the room. Note that it is not reasonable to encode this by standard public announcement since it may result in a possibly infinite formula: $[t = 27.1\,°\mathrm{C}]K(t = 27.1\,°\mathrm{C}) \wedge [t = 27.2\,°\mathrm{C}]K(t = 27.2\,°\mathrm{C}) \wedge \ldots$, and the inspection action itself may require an infinite action model in the standard DEL framework of [4] with a separate event for each possible value. Hence public inspection can be viewed as a public announcement of the actual value, but new techniques are required to express it formally. In our simple framework we define knowing and inspecting values as primitive operators, leaving the actual values out of our logical language.

The notions of knowing and inspecting values have a natural connection with dependencies in databases. This will also play a crucial role in the later technical development of the paper. In particular, our completeness proofs employ the famous set of axioms from [5]. For now, consider the following example.

Example 1. Suppose a university course was evaluated using anonymous questionnaires which besides an assessment for the teacher also asked the students for their main subject. See Table 1 for the results. Now suppose a student tells you, the teacher, that his major is Computer Science. Then clearly you know how that student assessed the course, since there is some dependency between the two columns. More precisely, in the cases of students 3 and 4, telling you the value of "Subject" effectively also tells you the value of "Assessment". In practice, a better questionnaire would only ask for combinations of questions that do not allow the identification of students.

Table 1. Evaluation results

Student	Subject	Assessment
1	Mathematics	Good
2	Mathematics	Very good
3	Logic	Good
4	Computer Science	Bad

Other examples abound: The author of [6] gives an account of how easily so-called 'de-identified data' produced from medical records could be 're-identified', by matching patient names to publicly available health data.

These examples illustrate that reasoning about knowledge of values in isolation, i.e. separated from knowledge *that*, is both possible and informative. It is such knowledge and its dynamics that we will study here.

2 Existing Work

Our work relates to a collection of papers on epistemic logics with other operators than the standard "knowing that" $K\varphi$. In particular we are interested in the Kv

operator expressing that an agent knows a value of a variable or constant. This operator is already mentioned in the seminal work [3] which introduced public announcement logic (PAL). However, a complete axiomatization of PAL together with Kv was only given in [7,8] using the relativized operator $Kv(\varphi, c)$ for the single and multi-agent cases. Moreover, it has been shown in [9] that by treating the negation of Kv as a primitive diamond-like operator, the logic can be seen as a normal modal logic in disguise with binary modalities.

Inspired by a talk partly based on an earlier version of this paper, Baltag proposed the very expressive Logic of Epistemic Dependency (LED) [10], where knowing that, knowing value, announcing that, announcing value can all be encoded in a general language which also includes equalities like $c = 4$ to facilitate the axiomatization.

In this paper we go in the other direction: Instead of extending the standard PAL framework with Kv, we study it in isolation together with its dynamic counterpart $[c]$ for public inspection. In general, the motto of our work here is to see how far one can get in formalizing knowledge and inspection of values without going all the way to or even beyond PAL. In particular we do not include values in the syntax and we do not have any nested epistemic modalities.

As one would expect, our simple language is accompanied by simpler models and also the proofs are less complicated than existing methods. Still we consider our Public Inspection Logic (PIL) more than a toy logic. Our completeness proof includes a novel construction which we call "canonical dependency graph" (Definition 6). We also establish the precise connection between our axioms and the Armstrong axioms widely used in database theory [5].

Table 2 shows how PIL fits into the family of existing languages. Note that [10] is the most expressive language in which all operators are encoded using $K_i^{t_1,\ldots,t_n} t$ which expresses that given the current values of t_1 to t_n, agent i knows the value of t. Moreover, to obtain a complete proof system for LED one also needs to include equality and rigid constants in the language. It is thus an open question to find axiomatizations for a language between PIL and LED without equality.

Table 2. Comparison of languages

PAL	p	$K\varphi$				$[!\varphi]\varphi$		[3]
PAL + Kv	p	$K\varphi$	$Kv(c)$			$[!\varphi]\varphi$		[3]
PAL + Kv^r	p	$K\varphi$	$Kv(c)$	$Kv(\varphi, c)$		$[!\varphi]\varphi$		[7–9]
PIL			$Kv(c)$		$[c]\varphi$			This paper
PIL + K		$K\varphi$	$Kv(c)$		$[c]\varphi$			Future work
LED	p	$K\varphi$	$Kv(c)$	$Kv(\varphi, c)$	$[c]\varphi$	$[!\varphi]\varphi$	$c = c$	[10]

All languages include the standard boolean operators $\top$, $\neg$ and $\wedge$ which we do not list in Table 2.

We also discuss other related works not in this line at the end of the paper.

3 Single-Agent PIL

We first consider a simple single-agent language to talk about knowing and inspecting values. Throughout the paper we assume a fixed set of constants $\mathbb{C}$.

Definition 1 (Syntax). *Let c range over $\mathbb{C}$. The language $\mathcal{L}_1$ is given by:*

$$\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid Kv(c) \mid [c]\varphi$$

Besides standard interpretations of the boolean connectives, the intended meanings are as follows: $Kv(c)$ reads "the agent knows the value of c" and the formula $[c]\varphi$ is meant to say "after revealing the actual value of c, φ is the case". We also use the standard abbreviations $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$ and $\varphi \rightarrow \psi := \neg\varphi \vee \psi$.

Definition 2 (Models and Semantics). *A model for $\mathcal{L}_1$ is a tuple $\mathcal{M} = \langle S, \mathcal{D}, V\rangle$ where S is a non-empty set of worlds (also called states), $\mathcal{D}$ is a non-empty domain and V is a valuation $V : (S \times \mathbb{C}) \rightarrow \mathcal{D}$. To denote $V(s,c) = V(t,c)$, i.e. that c has the same value at s and t according to V, we write $s =_c t$. If this holds for all $c \in C \subseteq \mathbb{C}$ we write $s =_C t$. The semantics are as follows:*

$\mathcal{M}, s \vDash \top$	*always*
$\mathcal{M}, s \vDash \neg\varphi$	$\Leftrightarrow \mathcal{M}, s \nvDash \varphi$
$\mathcal{M}, s \vDash \varphi \wedge \psi$	$\Leftrightarrow \mathcal{M}, s \vDash \varphi$ *and* $\mathcal{M}, s \vDash \psi$
$\mathcal{M}, s \vDash Kv(c)$	$\Leftrightarrow$ *for all* $t \in S : s =_c t$
$\mathcal{M}, s \vDash [c]\varphi$	$\Leftrightarrow \mathcal{M}|^s_c, s \vDash \varphi$

where $\mathcal{M}|^s_c$ is $\langle S', \mathcal{D}, V|_{S' \times \mathbb{C}}\rangle$ with $S' = \{t \in S \mid s =_c t\}$. If for a set of formulas Γ and a formula φ we have that whenever a model $\mathcal{M}$ and a state s satisfy $\mathcal{M}, s \vDash \Gamma$ then they also satisfy $\mathcal{M}, s \vDash \varphi$, then we say that φ follows semantically from Γ and write $\Gamma \vDash \varphi$. If this hold for $\Gamma = \varnothing$ we say that φ is semantically valid and write $\vDash \varphi$.

Note that the actual state s plays an important role in the last clause of our semantics: Public inspection of c at s reveals the *local actual* value of c to the agent. The model is restricted to those worlds which agree on c with s. This is different from PAL and other DEL variants based on action models, where updates are usually defined on models directly and not on pointed models.

We employ the usual abbreviation $\langle c\rangle\varphi$ as $\neg[c]\neg\varphi$. Note however, that public inspection of c can always take place and is deterministic. Hence the determinacy axiom $\langle c\rangle\varphi \leftrightarrow [c]\varphi$ is semantically valid and we include it in the following system.

Definition 3. *The proof system $\mathbb{SPIL}_1$ for PIL in the language $\mathcal{L}_1$ consists of the following axiom schemata and rules. If a formula φ is provable from a set of premises Γ we write $\Gamma \vdash \varphi$. If this holds for $\Gamma = \varnothing$ we also write $\vdash \varphi$.*

Axiom Schemata		***Rules***
TAUT	*all instances of propositional tautologies*	MP $\dfrac{\varphi, \varphi \to \psi}{\psi}$
DIST	$[c](\varphi \to \psi) \to ([c]\varphi \to [c]\psi)$	
LEARN	$[c]Kv(c)$	
NF	$Kv(c) \to [d]Kv(c)$	NEC $\dfrac{\varphi}{[c]\varphi}$
DET	$\langle c\rangle\varphi \leftrightarrow [c]\varphi$	
COMM	$[c][d]\varphi \leftrightarrow [d][c]\varphi$	
IR	$Kv(c) \to ([c]\varphi \to \varphi)$	

Intuitively, LEARN captures the effect of the inspection; NF says that the agent does not forget; DET says that inspection is deterministic; COMM says that inspections commute; finally, IR expresses that inspection does not bring any new information if the value is known already. Note that DET says that $[c]$ is a function. It also implies seriality which we list in the following Lemma.

Lemma 1. *The following schemes are provable in* $\mathbb{SPIL}_1$*:*

- $\langle c\rangle\top$ *(seriality)*
- $Kv(c) \to (\varphi \to [c]\varphi)$ *(IR')*
- $[c](\varphi \wedge \psi) \leftrightarrow [c]\varphi \wedge [c]\psi$ *(DIST')*
- $[c_1]\dots[c_n](\varphi \to \psi) \to ([c_1]\dots[c_n]\varphi \to [c_1]\dots[c_n]\psi)$ *(multi-DIST)*
- $[c_1]\dots[c_n](\varphi \wedge \psi) \leftrightarrow [c_1]\dots[c_n]\varphi \wedge [c_1]\dots[c_n]\psi$ *(multi-DIST')*
- $[c_1]\dots[c_n](Kv(c_1) \wedge \dots Kv(c_n))$ *(multi-LEARN)*
- $(Kv(c_1) \wedge \dots \wedge Kv(c_n)) \to [d_1]\dots[d_n](Kv(c_1) \wedge \dots \wedge Kv(c_n))$ *(multi-NF)*
- $(Kv(c_1) \wedge \dots \wedge Kv(c_n)) \to ([c_1]\dots[c_n]\varphi \to \varphi)$ *(multi-IR)*

Moreover, the multi-NEC rule is admissible: If $\vdash \varphi$*, then* $\vdash [c_1]\dots[c_n]\varphi$*.*

Proof. For reasons of space we only prove three of the items and leave the others as an exercise for the reader. For IR', we use DET and TAUT:

$$\dfrac{\dfrac{\dfrac{}{Kv(c) \to ([c]\neg\varphi \to \neg\varphi)}\,(\texttt{IR})}{Kv(c) \to (\neg[c]\varphi \to \neg\varphi)}\,(\texttt{DET})}{Kv(c) \to (\varphi \to [c]\varphi)}\,(\texttt{TAUT})$$

To show multi-NEC, we use DIST, NEC and TAUT. For simplicity, consider the case where $C = \{c_1, c_2\}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{[c_2](\varphi \to \psi) \to ([c_2]\varphi \to [c_2]\psi)}\,(\texttt{DIST})}{[c_1]([c_2](\varphi \to \psi) \to ([c_2]\varphi \to [c_2]\psi))}\,(\texttt{NEC})}{[c_1][c_2](\varphi \to \psi) \to [c_1]([c_2]\varphi \to [c_2]\psi)}\,(\texttt{DIST}, \texttt{TAUT})}{[c_1][c_2](\varphi \to \psi) \to ([c_1][c_2]\varphi \to [c_1][c_2]\psi)}\,(\texttt{DIST}, \texttt{TAUT})$$

For multi-LEARN, we use LEARN, NEC, COMM, DIST' and TAUT:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{[c_1]Kv(c_1)}\,(\texttt{LEARN})}{[c_2][c_1]Kv(c_1)}\,(\texttt{NEC})}{[c_1][c_2]Kv(c_1)}\,(\texttt{COMM}) \qquad \dfrac{\dfrac{}{[c_2]Kv(c_2)}\,(\texttt{LEARN})}{[c_1][c_2]Kv(c_2)}\,(\texttt{NEC})}{[c_1]([c_2]Kv(c_1) \wedge [c_2]Kv(c_2))}\,(\texttt{DIST'}, \texttt{TAUT})}{[c_1][c_2](Kv(c_1) \wedge Kv(c_2))}\,(\texttt{DIST'}, \texttt{TAUT})$$

Definition 4. *We use the following abbreviations for any two finite sets of constants* $C = \{c_1, \ldots, c_m\}$ *and* $D = \{d_1, \ldots, d_n\}$.

- $Kv(C) := Kv(c_1) \wedge \cdots \wedge Kv(c_m)$
- $[C]\varphi := [c_1] \ldots [c_m]\varphi$
- $Kv(C, D) := [C]Kv(D)$.

Note that by multi-DIST' and COMM the exact enumeration of C and D in Definition 4 do not matter modulo logical equivalence.

In particular, these abbreviations allow us to shorten the "multi" items from Lemma 1 to $Kv(C, C)$, $Kv(C) \to Kv(D, C)$ and $Kv(C) \to ([C]\varphi \to \varphi)$. The abbreviation $Kv(C, D)$ allows us to define dependencies and it will be crucial in our completeness proof. We have that:

$$\mathcal{M}, s \vDash Kv(C, D) \Leftrightarrow \text{for all } t \in S : \text{if } s =_C t \text{ then } s =_D t$$

Definition 5. *Let* $\mathcal{L}_2$ *be the language given by* $\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid Kv(C, C)$.

Note that this language is essentially a fragment of $\mathcal{L}_1$ due to the above abbreviation, where (possibly multiple) $[c]$ operators only occur in front of Kv operators (or conjunctions thereof). Moreover, the next Lemma might count as a small surprise.

Lemma 2. $\mathcal{L}_1$ *and* $\mathcal{L}_2$ *are equally expressive.*

Proof. As $Kv(\cdot, \cdot)$ was just defined as an abbreviation, we already know that $\mathcal{L}_1$ is at least as expressive as $\mathcal{L}_2$: we have $\mathcal{L}_2 \subseteq \mathcal{L}_1$. We can also translate in the other direction by pushing all sensing operators through negations and conjunctions. Formally, let $t : \mathcal{L}_1 \to \mathcal{L}_2$ be defined by

$$\begin{array}{ll} Kv(d) \mapsto Kv(\varnothing, \{d\}) & [c]\neg\varphi \mapsto \neg t([c]\varphi) \\ \neg\varphi \mapsto \neg t(\varphi) & [c](\varphi \wedge \psi) \mapsto t([c]\varphi) \wedge t([c]\psi) \\ \varphi \wedge \psi \mapsto t(\varphi) \wedge t(\psi) & [c]\top \mapsto \top \\ & [c_1] \ldots [c_n]Kv(d) \mapsto Kv(\{c_1, \ldots, c_n\}, \{d\}) \end{array}$$

Note that this translation preserves and reflects truth because determinacy and distribution are valid (determinacy allows us to push $[c]$ through negations, distribution to push $[c]$ through conjunctions). At this stage we have not yet established completeness, but determinacy is also an axiom. Hence we can note separately that $\varphi \leftrightarrow t(\varphi)$ is provable and that t preserves and reflects provability and consistency.

Example 2. Note that the translation of $[c]\varphi$ formulas also depends on the top connective within φ. For example we have

$$\begin{aligned} t([c](\neg Kv(d) \wedge [e]Kv(f))) &= t([c]\neg Kv(d)) \wedge t([c][e]Kv(f)) \\ &= \neg Kv(\{c\},\{d\}) \wedge Kv(\{c,e\},\{f\}) \end{aligned}$$

The language $\mathcal{L}_2$ allows us to connect PIL to the maybe most famous axioms about database theory and dependence logic from [5].

Lemma 3. *Armstrong's axioms are semantically valid and derivable in* $\mathbb{SPIL}_1$*:*

- $Kv(C,D)$ *for any* $D \subseteq C$ *(projectivity)*
- $Kv(C,D) \wedge Kv(D,E) \to Kv(C,E)$ *(transitivity)*
- $Kv(C,D) \wedge Kv(C,E) \to Kv(C,D \cup E)$ *(additivity)*

Proof. The semantic validity is easy to check, hence we focus on the derivations.

For projectivity, take any two finite sets $D \subseteq C$. If $D = C$, then we only need a derivation like the following which basically generalizes learning to finite sets.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{[c_1]Kv(c_1)}\,\texttt{(LEARN)}}{[c_2][c_1]Kv(c_1)}\,\texttt{(NEC)}}{[c_1][c_2]Kv(c_1)}\,\texttt{(COMM)} \qquad \dfrac{\dfrac{}{[c_2]Kv(c_2)}\,\texttt{(LEARN)}}{[c_1][c_2]Kv(c_1)}\,\texttt{(NEC)}}{[c_1]([c_2]Kv(c_1) \wedge [c_2]Kv(c_2))}\,\texttt{(DIST)}}{[c_1][c_2](Kv(c_1) \wedge Kv(c_2))}\,\texttt{(DIST)}$$

If $D \subsetneq C$, then continue by applying NEC for all elements of $C \setminus D$ to get $Kv(C,D)$.

Transitivity follows from IR and NF as follows. For simplicity, first we only consider the case where C, D and E are singletons.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Kv(e) \to [c]Kv(e)}\,\texttt{(NF)}}{[d](Kv(e) \to [c]Kv(e))}\,\texttt{(NEC)}}{[d]Kv(e) \to [d][c]Kv(e)}\,\texttt{(DIST)}}{[d]Kv(e) \to [c][d]Kv(e)}\,\texttt{(COMM)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{Kv(d) \to ([d]Kv(e) \to Kv(e))}\,\texttt{(IR)}}{[c](Kv(d) \to ([d]Kv(e) \to Kv(e)))}\,\texttt{(NEC)}}{[c]Kv(d) \to [c]([d]Kv(e) \to Kv(e))}\,\texttt{(DIST)}}{[c]Kv(d) \to ([c][d]Kv(e) \to [c]Kv(e))}\,\texttt{(DIST)}}{[c]Kv(d) \to ([d]Kv(e) \to [c]Kv(e))}\,\texttt{(TAUT)}$$

Now consider any three finite sets of constants $C = \{c_1, \ldots, c_l\}$. Using the abbreviations from Definition 4 and the "multi" rules given in Lemma 1 it is easy to generalize the proof. In fact, the proof is exactly the same with capital letters.

Similarly, additivity follows immediately from multi-DIST'.

We can now use Armstrong's axioms to prove completeness of our logic. The crucial idea is a new definition of a canonical dependency graph.

Theorem 1 (Strong Completeness). *For all sets of formulas* $\Delta \subseteq \mathcal{L}_1$ *and all formulas* $\varphi \in \mathcal{L}_1$*, if* $\Delta \vDash \varphi$*, then also* $\Delta \vdash \varphi$*.*

Proof. By contraposition using a canonical model. Suppose $\Delta \nvdash \varphi$. Then $\Delta \cup \{\neg\varphi\}$ is consistent and there is a maximally consistent set $\Gamma \subseteq \mathcal{L}_1$ such that $\Gamma \supseteq \Delta \cup \{\neg\varphi\}$. We will now build a model $\mathcal{M}_\Gamma$ such that for the world $\mathbb{C}$ in that model we have $\mathcal{M}_\Gamma, \mathbb{C} \vDash \Gamma$ which implies $\Delta \nvDash \varphi$.

Definition 6 (Canonical Graph and Model). *Let the graph* $G_\Gamma := (\mathcal{P}(\mathbb{C}), \rightarrow)$ *be given by* $A \rightarrow B$ *iff* $Kv(A, B) \in \Gamma$. *By Lemma 3 this graph has properties corresponding to the Armstrong axioms: projectivity, transitivity and additivity. We call a set of variables* $s \subseteq \mathbb{C}$ closed *under* G_Γ *iff whenever* $A \subseteq s$ *and* $A \rightarrow B$ *in* G_Γ, *then also* $B \subseteq s$. *Then let the canonical model be* $\mathcal{M}_\Gamma := (S, \mathcal{D}, V)$ *where*

$$S := \{s \subseteq \mathbb{C} \mid s \text{ is closed under } G_\Gamma\}, \mathcal{D} := \{0, 1\} \text{ and } V(s, c) = \begin{cases} 0 \text{ if } c \in s \\ 1 \text{ otherwise} \end{cases}$$

Note that our domain is just $\{0, 1\}$. This is possible because we do not have to find a model where the dependencies hold globally. Instead, $Kv(C, d)$ only says that given the C-values at the actual world, also the d values are the same at the other worlds. The dependency does not need to hold between two non-actual worlds. This distinguishes our models from relationships as discussed in [5] where no actual world or state is used, see Example 4 below.

Given the definition of a canonical model we can now show:

Lemma 4 (Truth Lemma). $\mathcal{M}_\Gamma, \mathbb{C} \vDash \varphi$ *iff* $\varphi \in \Gamma$.

Before going into the proof, let us emphasize two peculiarities of our truth lemma: First, the states in our canonical model are not maximally consistent sets of formulas but sets of constants. Second, we only claim the truth Lemma at one specific state, namely $\mathbb{C}$ where all constants have value 0. As our language does not include nested epistemic modalities, we actually never evaluate formulas at other states of our canonical model.

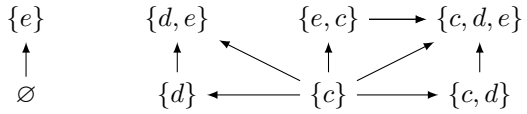
Proof (Truth Lemma). Note that it suffices to show this for all φ in $\mathcal{L}_2$: Given some $\varphi \in \mathcal{L}_1$, by Lemma 2 we have that $\mathcal{M}_\Gamma, \mathbb{C} \vDash \varphi \iff \mathcal{M}_\Gamma, \mathbb{C} \vDash t(\varphi)$ because the translation preserves and reflects truth. Moreover, we have $\varphi \in \Gamma \iff t(\varphi) \in \Gamma$, because $\varphi \leftrightarrow t(\varphi)$ is provable in $\mathbb{SPIL}_1$. Hence it suffices to show that $\mathcal{M}_\Gamma, \mathbb{C} \vDash t(\varphi)$ iff $t(\varphi) \in \Gamma$, i.e. to show the Truth Lemma for $\mathcal{L}_2$. Again, negation and conjunction are standard, the crucial case are dependencies.

Suppose $Kv(C, D) \in \Gamma$. By definition $C \rightarrow D$ in G_Γ. To show $\mathcal{M}_\Gamma, \mathbb{C} \vDash Kv(C, D)$, take any t such that $\mathbb{C} =_C t$ in $\mathcal{M}_\Gamma$. Then by definition of V we have $C \subseteq t$. As t is closed under G_Γ, this implies $D \subseteq t$. Now by definition of V we have $\mathbb{C} =_D t$.

For the converse, suppose $Kv(C, D) \notin \Gamma$. Then by definition $C \not\rightarrow D$ in G_Γ. Now, let $t := \{c' \in \mathbb{C} \mid C \rightarrow \{c'\} \text{ in } G_\Gamma\}$. This gives us $C \subseteq t$. But we also have $D \not\subseteq t$ because otherwise additivity would imply $C \rightarrow D$ in G_Γ. Moreover, because G_Γ is transitive it is enough to "go one step" in G_Γ to get a set that is closed under G_Γ. This means that t is closed under G_Γ and therefore a state in our model, i.e. we have $t \in S$. Now by definition of V and projectivity, we have $\mathbb{C} =_C t$ but $\mathbb{C} \neq_D t$. Thus t is a witness for $\mathcal{M}_\Gamma, \mathbb{C} \nvDash Kv(C, D)$.

This also finishes the completeness proof. Note that we used all three properties corresponding to the Armstrong axioms.

Example 3. To illustrate the idea of the canonical dependency graph, let us study a concrete example of what the graph and model look like. Consider the maximally consistent set $\Gamma = \{\neg Kv(c), \neg Kv(d), Kv(e), Kv(c,d), \dots\}$. The interesting part of the canonical graph G_Γ then looks as follows, where the nodes are subsets of $\{c,d,e\}$. For clarity we only draw $\to \cap \not\subseteq$, i.e. we omit edges given by inclusions. For example all nodes will also have an edge going to the $\varnothing$ node.



To get a model out of this graph, note that there are exactly three subsets of $\mathbb{C}$ closed under following the edges. Namely, let $S = \{s : \{e\}, t : \{d,e\}, u : \{c,d,e\}\}$ and use the binary valuation which says that a constant has value 0 iff it is an element of the state. It is then easy to check that $\mathcal{M}, u \vDash \Gamma$.

	s	t	u
c	1	1	0
d	1	0	0
e	0	0	0

It is also straightforward to define an appropriate notion of bisimulation.

Definition 7. *Two pointed models* $((S, \mathcal{D}, V), s)$ *and* $((S', \mathcal{D}', V'), s')$*, are* bisimilar *iff (i) For all finite* $C \subseteq \mathbb{C}$ *and all* $d \in \mathbb{C}$*: If there is a* $t \in S$ *such that* $s =_C t$ *and* $s \neq_d t$*, then there is a* $t' \in S'$ *such that* $s' =_C t'$ *and* $s' \neq_d t'$*; and (ii) Vice versa.*

Note that we do not need the bisimulation to also link non-actual worlds. This is because all formulas are evaluated at the same world. In fact it would be too strong for the following characterization.

Theorem 2. *Two pointed models satisfy the same formulas iff they are bisimilar.*

Proof. By Lemma 2 we only have to consider formulas of $\mathcal{L}_2$. Moreover, it suffices to consider formulas $Kv(C,d)$ with a singleton in the second set because $Kv(C,D)$ is equivalent to $\bigwedge_{d \in D} Kv(C,d)$. Then it is straightforward to show that if $\mathcal{M}, s$ and $\mathcal{M}', s'$ are bisimilar then $\mathcal{M}, s \vDash \neg Kv(C,d) \iff \mathcal{M}', s' \vDash \neg Kv(C,d)$ by definition of our bisimulation. The other way around is also obvious since the two conditions for bisimulation are based on the semantics of $\neg Kv(C,d)$.

Note that a bisimulation characterization for a language without the dynamic operator can be obtained by restricting Definition 7 to $C = \varnothing$. We leave it as an exercise for the reader to use this and Theorem 2 to show that $[c]$ is not reducible, which distinguishes it from the public announcement $[\varphi]$ in PAL.

Example 4 (Pointed Models Make a Difference). It seems that the following theorem of our logic does not translate to Armstrong's system from [5].

$$[c](Kv(d) \vee Kv(e)) \leftrightarrow ([c]Kv(d) \vee [c]Kv(e))$$

First, to see that this is provable, note that it follows from determinacy and seriality. Second, it is valid because we consider pointed models which convey more information than a simple list of possible values. Consider the following table which represents 4 possible worlds.

c	d	e
1	1	3
1	1	2
2	2	1
2	3	1

Here we would say that "After learning c we know d or we know e.", i.e. the antecedent of above formula holds. However, the consequent only holds if we evaluate formulas while pointing at a specific world/row: It is globally true that given c we will learn d or that given c we will learn e. But none of the two disjuncts holds globally which would be needed for a dependency in Armstrong's sense. Note that this is more a matter of expressiveness than of logical strength. In Armstrong's system there is just no way to express $[c](Kv(d) \vee Kv(e))$.

4 Multi-agent PIL

We now generalize the Public Inspection Logic to multiple agents. In the language we use Kv_i to say that agent i knows the value of c and in the models an accessibility relation for each agent is added to describe their knowledge. To obtain a complete proof system we can leave most axioms as above but have to restrict the irrelevance axiom. Again the completeness +proof uses a canonical model construction and a truth lemma for a +restricted but equally expressive syntax. The only change is that we now define a dependency graph for each agent in order to define accessibility relations instead of restricted sets of worlds.

Definition 8 (Multi-Agent PIL). *We fix a non-empty set of agents I. The language $\mathcal{L}_1^I$ of multi-agent Public Inspection Logic is given by*

$$\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid Kv_i c \mid [c]\varphi$$

where $i \in I$. We interpret it on models $\langle S, \mathcal{D}, V, R\rangle$ where S, $\mathcal{D}$ and V are as before and R assigns to each agent i an equivalence relation $\sim_i$ over S. The semantics are standard for the booleans and as follows:

$$\begin{array}{l} \mathcal{M}, s \vDash Kv_i c \iff \forall t \in S : s \sim_i t \Rightarrow s =_c t \\ \mathcal{M}, s \vDash [c]\varphi \iff \mathcal{M}|_c^s, s \vDash \varphi \end{array}$$

where $\mathcal{M}|_c^s$ is $\langle S', \mathcal{D}, V|_{S' \times \mathbb{C}}, R|_{S' \times S'}\rangle$ with $S' = \{t \in S \mid s =_c t\}$.

Analogous to Definition 4 we define the following abbreviation to express dependencies known by agent i and note its semantics:

$$Kv_i(C, D) := [c_1] \dots [c_n](Kv_i(d_1) \wedge \dots \wedge Kv_i(d_m))$$

$$\mathcal{M}, s \vDash Kv_i(C, D) \Leftrightarrow \textit{for all } t \in S : \textit{if } s \sim_i t \textit{ and } s =_C t \textit{ then } s =_D t$$

The proof system $\mathbb{SPIL}$ for PIL in the language $\mathcal{L}_1^I$ is obtained by replacing each Kv in the axioms of $\mathbb{SPIL}_1$ by Kv_i, and replacing IR by the following restricted version:

RIR $\quad Kv_i c \to ([c]\varphi \to \varphi)$ *where φ does not mention any agent besides i*

Before summarizing the completeness proof for the multi-agent setting, let us highlight some details of this definition.

As be fore the actual state s plays an important role in the semantics of $[c]$. However, we could also use an alternative but equivalent definition: Instead of deleting states, only delete the $\sim_i$ links between states that disagree on the value of c. Then the update no longer depends on the actual state.

For traditional reasons we define $\sim_i$ to be an equivalence relation. This is not strictly necessary, because our language can not tell whether the relation is reflexive, transitive or symmetric. Removing this constraint and extending the class of models would thus not make any difference in terms of validities.

For the proof system, note that the original irrelevance axiom IR is *not* valid in the multi-agent setting because φ might talk about other agents for which the inspection of c does matter.

Theorem 3 (Strong Completeness for $\mathbb{SPIL}$). *For all sets of formulas $\Delta \subseteq \mathcal{L}_1^I$ and all formulas $\varphi \in \mathcal{L}_1^I$, if $\Delta \vDash \varphi$, then also $\Delta \vdash \varphi$.*

Proof. By the same methods as for Theorem 1. Given a maximally consistent set $\Gamma \subseteq \mathcal{L}_1^I$ we want to build a model $\mathcal{M}_\Gamma$ such that for the world $\mathbb{C}$ in that model we have $\mathcal{M}_\Gamma, \mathbb{C} \vDash \Gamma$.

First, for each agent $i \in I$, let G_Γ^i be the graph given by $A \to_i B \;\; :\Longleftrightarrow \Gamma \vdash Kv_i(A, B)$. Given that the proof system $\mathbb{SPIL}$ was obtained by indexing the axioms of $\mathbb{SPIL}_1$, it is easy to check that indexed versions of the Armstrong axioms are provable and therefore all the graphs G_Γ^i for $i \in I$ will have the corresponding properties. In particular RIR suffices for this.

Second, define the canonical model $\mathcal{M}_\Gamma := (S, \mathcal{D}, V, R)$ where $S := \mathcal{P}(\mathbb{C})$, $\mathcal{D} := \{0, 1\}$, $V(s, c) := 0$ if $c \in s$ and $V(s, c) := 1$ otherwise, and $s \sim_i t$ iff s and t are both closed or both not closed under G_Γ^i.

Lemma 5 (Multi-Agent Truth Lemma). $\mathcal{M}_\Gamma, \mathbb{C} \vDash \varphi$ *iff* $\varphi \in \Gamma$.

Proof. Again it suffices to show the Truth Lemma for a restricted language and we only consider the state $\mathbb{C}$. We proceed by induction on φ. The crucial case is when φ is of form $Kv_i(C, D)$.

Suppose $Kv_i(C, D) \in \Gamma$. Then by definition $C \to D$ in G_Γ^i. To show $\mathcal{M}_\Gamma, \mathbb{C} \vDash Kv_i(C, D)$, take any t such that $\mathbb{C} \sim_i t$ and $\mathbb{C} =_C t$ in $\mathcal{M}_\Gamma$. Then by definition

of V we have $C \subseteq t$. Moreover, $\mathbb{C}$ is closed under G^i_Γ. Hence by definition of $\sim_i$ also t must be closed under G^i_Γ which implies $D \subseteq t$. Now by definition of V we have $\mathbb{C} =_D t$.

For the converse, suppose $Kv_i(C, D) \notin \Gamma$. Then by definition $C \not\to D$ in G^i_Γ. Now, let $t := \{c' \in \mathbb{C} \mid C \to \{c'\} \text{ in } G^i_\Gamma\}$. This gives us $C \subseteq t$. But we also have $D \not\subseteq t$ because otherwise additivity would imply $C \to D$ in G^i_Γ. Moreover, because G^i_Γ is transitive it is enough to "go one step" in G^i_Γ to get a set that is closed under G^i_Γ. This means that t is closed under G^i_Γ and therefore by definition of $\sim_i$ we have $\mathbb{C} \sim_i t$. Now by definition of V and projectivity, we have $\mathbb{C} =_C t$ but $\mathbb{C} \neq_D t$. Thus t is a witness for $\mathcal{M}_\Gamma, \mathbb{C} \not\models Kv_i(C, D)$.

Again the Truth Lemma also finishes the completeness proof.

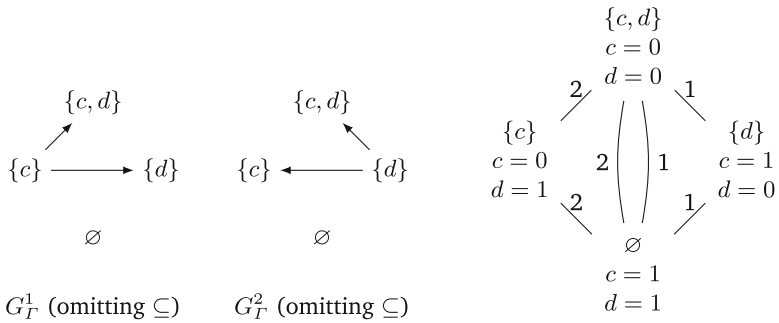


Fig. 1. Two canonical dependency graphs and the resulting canonical model.

Example 5. Analogous to Example 3, the following illustrates the multi-agent version of our canonical construction. Consider the maximally consistent set $\Gamma = \{\neg Kv_1(d), Kv_1(c, d), \neg Kv_1(d, c), \neg Kv_2(c), \neg Kv_2(c, d), Kv_1(d, c), \dots\}$. Note that agents 1 and 2 do not differ in which values they know right now but there is a difference in what they will learn from inspections of c and d. The two canonical dependency graphs generated from Γ are shown in Fig. 1. Again for clarity we only draw the non-inclusion arrows. The subsets of $\mathbb{C} = \{c, d\}$ closed under the graphs are thus $\{\{c, d\}, \{d\}, \varnothing\}$ and $\{\{c, d\}, \{c\}, \varnothing\}$ for agent 1 and 2 respectively, inducing the equivalence relations as shown in Fig. 1.

It is also not hard to find the right notion of bisimulation for $\mathbb{SPIL}$.

Definition 9. *Given two models $(S, \mathcal{D}, V, R)$ and $(S', \mathcal{D}', V', R')$, a relation $Z \subseteq S \times S'$ is a* multi-agent bisimulation *iff for all sZs' we have (i) For all finite $C \subseteq \mathbb{C}$, all $d \in \mathbb{C}$ and all agents i: If there is a $t \in S$ such that $s \sim_i t$ and $s =_C t$ and $s \neq_d t$, then there is a $t' \in S'$ such that tZt' and $s' \sim_i t'$ and $s =_C t$ and $s' \neq_d t'$; and (ii) Vice versa.*

Theorem 4. *Two pointed models satisfy the same formulas of the multi-agent language $\mathcal{L}^I_1$ iff there is a multi-agent bisimulation linking them.*

As it is very similar to the one of Theorem 2, we omit the proof here.

5 Future Work

Between our specific approach and the general language of [10], a lot can still be explored. An advantage of having a weaker language with explicit operators, instead of encoding them in a more general language, is that we can clearly see the properties of those operators showing up as intuitive axioms.

The framework can be extended in different directions. We could for example add equalities $c = d$ to the language, together with knowledge $K(c = d)$ and announcement $[c = d]$. No changes to the models are needed, but axiomatizing these operators seems not straightforward. Alternatively, just like Plaza added Kv to PAL, we can also add K to PIL. Another next language to be studied is thus $\mathsf{PIL} + K$ from Table 2 above and given by

$$\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid Kv_i c \mid K_i\varphi \mid [c]\varphi.$$

Note that in this language, we can also express *knowledge of* dependency in contrast to *de facto* dependency. For example, $K_i[c]Kv_i d$ expresses that agent i knows that d functionally depends on c, while $[c]Kv_i d$ express that the value of d (given the information state of i) is determined by the *actual value* of c *de facto*. In particular the latter does not imply that i knows this. The agent can still consider other values of c possible that would not determine the value of d. To see the difference technically, we can spell out the truth condition for $K_i[c]Kv_i(d)$ under standard Kripke semantics for K_i on S5 models:

$$\mathcal{M}, s \vDash K_i[c]Kv_i(d) \Leftrightarrow \text{ for all } t_1 \sim_i s, t_2 \sim_i s : t_1 =_c t_2 \implies t_1 =_d t_2$$

Now consider Example 4: $[c]Kv(d)$ holds in the first row, but $K[c]Kv(d)$ does not hold since the semantics of K require $[c]Kv(d)$ to hold at *all* worlds considered possible by the agent. This also shows that $[c]Kv(d)$ is not positively introspective (i.e. the formula $[c]Kv(d) \to K_i[c]Kv(d)$ is not valid), and it is essentially not a subjective epistemic formula.

In this way, $K[c]Kv(d)$ can also be viewed as the atomic formula $= (c, d)$ in *dependence logic* (DL) from [11]. A *team model* of DL can be viewed as the set of epistemically accessible worlds, i.e., a single-agent model in our case. The connection with dependence logic also brings PIL closer to the first-order variant of *epistemic inquisitive logic* by [12], where knowledge of entailment of interrogatives can also be viewed as the knowledge of dependency. For a detailed comparison with our approach, see [13, Sect. 6.7.4].

Another approach is to make the dependency more explicit and include functions in the syntax. In [14] a functional dependency operator $\mathcal{K}f_i$ is added to the epistemic language with Kv_i operators: $\mathcal{K}f_i(c, d) := \exists f K_i(d = f(c))$ where f ranges over a pool of functions.

Finally, there is an independent but related line of work on (in)dependency of variables using predicates, see for example [15–18]. In particular, [17] also uses a notion of dependency as an epistemic implication "Knowing c implies knowing d.", similar to our formula $Kv(c, d)$. In [18] also a "dependency graph" is used to describe how different variables, in this case payoff functions in strategic games,

may depend on each other. Note however, that these graphs are not the same as our canonical dependency graphs from Definition 6. Our graphs are directed and describe determination between sets of variables. In contrast, the graphs in [18] are undirected and consist of singleton nodes for each player in a game. We leave a more detailed comparison for a future occasion.

Acknowledgements. We thank the following people for useful comments on this work: Alexandru Baltag, Peter van Emde Boas, Hans van Ditmarsch, Jie Fan, Kai Li and our anonymous reviewers.

This research cooperation was made possible by travel grant 040.11.490 from NWO for Yanjing Wang, which is herewith gratefully acknowledged.

References

1. Wang, Y.: Beyond knowing that: a new generation of epistemic logics. In: van Ditmarsch, H., Sandu, G. (eds.) Jaakko Hintikka on Knowledge and Game Theoretical Semantics. Springer (2016, forthcoming)
2. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic, vol. 1. Springer, Heidelberg (2007)
3. Plaza, J.: Logics of public communications. Synthese **158**(2), 165–179 (2007)
4. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In: Bilboa, I. (ed.) TARK 1998, pp. 43–56 (1998)
5. Armstrong, W.W.: Dependency structures of data base relationships. In: IFIP Congress, Geneva, Switzerland, vol. 74, pp. 580–583 (1974)
6. Sweeney, L.: Only you, your doctor, and many others may know. Technology Science (2015). http://techscience.org/a/2015092903/
7. Wang, Y., Fan, J.: Knowing that, knowing what, and public communication: public announcement logic with KV operators. In: IJCAI 2013, pp. 1147–1154 (2013)
8. Wang, Y., Fan, J.: Conditionally knowing what. In: Advances in Modal Logic, vol. 10, pp. 569–587 (2014)
9. Gu, T., Wang, Y.: "Knowing value" logic as a normal modal logic. In: Advances in Modal Logic, vol. 11, pp. 362–381 (2016)
10. Baltag, A.: To know is to know the value of a variable. In: Advances in Modal Logic, vol. 11, pp. 135–155 (2016)
11. Väänänen, J.: Dependence Logic: A New Approach to Independence Friendly Logic. Cambridge University Press, New York (2007)
12. Ciardelli, I., Roelofsen, F.: Inquisitive dynamic epistemic logic. Synthese **192**(6), 1643–1687 (2015)
13. Ciardelli, I.: Questions in logic. Ph.D. thesis, University of Amsterdam (2016)
14. Ding, Y.: Epistemic logic with functional dependency operator. Bachelor's thesis (in Chinese), Peking University (2015)
15. More, S.M., Naumov, P.: An independence relation for sets of secrets. Stud. Logica **94**(1), 73–85 (2010)
16. Naumov, P.: Independence in information spaces. Stud. Logica **100**(5), 953–973 (2012)
17. Naumov, P., Nicholls, B.: Rationally functional dependence. J. Philos. Logic **43**(2–3), 603–616 (2014)
18. Harjes, K., Naumov, P.: Functional dependence in strategic games. Notre Dame J. Formal Logic **57**(3), 341–353 (2016)

Random Models for Evaluating Efficient Büchi Universality Checking

Corey Fisher[1(✉)], Seth Fogarty[2], and Moshe Vardi[1]

[1] Rice University, Houston, USA
corey.s.fisher@gmail.com
[2] Trinity University, San Antonio, USA

Abstract. Automata-theoretic formal verification approaches the problem of guaranteeing that a program conforms to its specification by reducing conformance to language containment. We can prove conformance by representing both programs and specifications as automata and proving that the specification contains the program. This connection to the theory of automata on infinite words motivated an extensive research program into the algorithmic theory of automata on infinite words, with a focus on algorithms that perform well in practice. The focus on practical performance is important because of the large gap between worst-case complexity and practice for many automata-theoretic algorithms. Unfortunately, there are few benchmark instances of automata in industrial verification. To overcome this challenge, Tabakov and Vardi proposed a model for generating random automata as test cases.

The Tabakov-Vardi (T-V) model, however, is just one random model, based on a specific, rather simple model of random graphs. Other models of random graphs have been studied over the years. While the T-V model has the advantage of simplicity, it is not clear that performance analysis conducted on this model is robust, and an analogous analysis over other random models might yield different conclusions. To address this problem, we introduce three novel models of random automata, yielding automata that are richer in structure than the automata generated by the T-V model. By generating large corpora of random automata and using them to evaluate the performance of universality-checking algorithms, we show that the T-V model is a robust random model for evaluating performance of universality-checking algorithms.

1 Introduction

Automata-theoretic formal verification is an approach to the problem of guaranteeing that a program (in software or hardware) conforms to its specification, in which conformance is reduced to the problem to language containment. By representing both programs and specifications as automata and proving that the specification contains the program, we can prove conformance [19]. This connection to automata theory, and, in particular, to the theory of automata on infinite

We recommend viewing the plots in this paper online. For a longer technical report, see http://www.cs.rice.edu/~vardi.

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 91–105, 2017.
DOI: 10.1007/978-3-662-54069-5_8

words [21], motivated an extensive research program into the algorithmic theory of automata on infinite words, cf. [20], and the focus of this program is often on algorithms that perform well in practice, cf. [12].

We focus here on the Büchi *universality-checking* problem, which is a simplified case of containment checking, the canonical verification problem [19]. An automaton A is *universal* if it accepts all input words; equivalently A is universal if its complement $\overline{A}$ is *empty*, that is it accepts no input words. A simplistic way to check universality of A is to check emptiness of $\overline{A}$, which can be reduced to reachability analysis of $\overline{A}$'s state-transition graphs. Such an approach would have to deal with the blow-up of Büchi complementation, so extant algorithms for universality use a variety of heuristics to check emptiness of $\overline{A}$ without constructing it in full, cf. [4].

The focus on performance in practice is important because of the large gap between worst-case complexity and performance in practice for many automata-theoretic algorithms. For example, the best upper bound for the complementation of Büchi automata is $2^{O(n \log n)}$ [15] (realized, for example, by the *rank-based* construction in [9]), which matches the known lower bound [13]. This bound is significantly lower that the earlier upper bound of $2^{O(n^2)}$ [16], which uses Büchi 's *Ramsey-based* construction [1]. Yet a comparison of the rank-based construction with the Ramsey-based construction on real-life instances showed that the Ramsey-based construction can be quite competitive in practice with the rank-based construction – each outperforms the other on different problem instances [5].

Nevertheless, the quest for automata-theoretic algorithms that perform well in practice is hampered by the fact that there is a shortage of benchmark instances of automata that arise in industrial verification (see discussion below). To overcome this challenge, Tabakov and Vardi proposed a model for generating random automata on which different algorithms can be evaluated and compared [17,18]. The model has three parameters: (1) the size (number of states) of the automaton, (2) the *density* of transitions (ratio of transitions to states), and (3) The *density* of accepting states (ratio of accepting to total number of states). Subject to these parameters, the model generates automata randomly. The Tabakov-Vardi (T-V, for short) model is attractive for two reasons [17]: First, the model gives rise to an interesting *universality terrain*, which describes the relationship between the probability of automaton universality (which means that all input words are accepted) and the density parameters. Second, the model gives rise to an interesting *performance terrain*, which describes the relationship between algorithmic performance and the density parameters. (We discuss these two terrains in detail in the body of the paper.) In subsequent years, this model has become the standard model for the evaluation of Büchi -complementation tools, cf. [2,4,11,14].

The T-V model, however, is just one specific random model, based on a specific, and quite simple model of random graphs [8]. As we show in this paper, several other models of random graphs have been studied over the years. While the T-V model has the advantage of simplicity, it is not a priori clear that performance analyses conducted on this model are robust, as it is entirely possible that

analogous analyses over other random models would yield different conclusions. Since performance analyses over random models are used in this context as a substitute to such analyses over a benchmark suite of real-life problem instances, it is desirable at least to know whether analyses over random models yield robust conclusions.

To address this problem, we introduce three[1] novel models of structured random automata, based on existing random graph models – the *vertex-copying* model [7], the *Frank-Strauss* model [6], and the *co-accessible* model [10]. These models are based on different models that have been proposed for random graphs. While the T-V model is uniformly random, generating unstructured automata, these new models constrain randomness in some way to provide structural guarantees about the resulting automata: The vertex-copying model guarantees a power-law degree distribution, the Frank-Strauss model restricts which transitions are valid, and the co-accessible model guarantees that each state in the resulting automaton can reach an accepting state.

These structural properties help the models represent a wide variety of possible types of problem instances that might be encountered in the real world. Furthermore, these model generate problem instances that are quite unlikely to be generated by the T-V model. Our goal is to compare performance analysis on the T-V model against performance analysis on the three new models. If performance analyses on the a variety of different models all reach similar conclusions, then we can conclude that these conclusions are likely robust. If, on the other hand, performance analyses on different models reach different conclusions, then we would gain a deeper understanding of how structure affects algorithmic performance and learn that the choice of algorithm should depend on the structure of the problem instance being solved.

By generating large corpora of random automata and using them to evaluate the performance of universality-checking algorithms we first show that the new models possess the same useful properties for universality as the T-V model. We then replicate results of Fogarty and Vardi [4] for universality checking, using all four random models. We show that the finding reached in [4], concluding that the two tools compared are competitive, is robust across the four models. Finally, we compare Fogarty and Vardi's Rank tool [4], the most recent implementation of the rank-based algorithm, with a modern Ramsey-based tool, RABIT 2.3[2], and show that the Ramsey-based tool strongly outperforms the rank-based tool, again over all four models. We conclude, therefore, that the T-V model, in spite of its simplicity, is an adequate random model for evaluating performance of universality-checking algorithms.

[1] The full version of the paper, with more models, can be found in the technical report [3].

[2] http://languageinclusion.org/doku.php?id=tools.

2 Background

Automata Theory. A *Büchi automaton* is a tuple $A = (\Sigma, Q, Q_0, \delta, F)$, where Σ is a finite alphabet, Q is the finite set of states, $Q_0 \subseteq Q$ is the set of initial states, $\delta \subseteq Q \times \Sigma \times Q$ is the transition relation, and $F \subseteq Q$ is the set of accepting states. Büchi automata take infinite words from Σ^ω as input. A *run* of a Büchi automaton on a word $w_0, w_1, ... \in \Sigma^\omega$ is any infinite sequence $q_0, q_1, ... \in Q^\omega S$ such that $q_0 \in Q_0$, and $(q_i, w_i, q_{i+1}) \in \delta$. A run is *accepting* if some accepting state $q_i \in F$ occurs infinitely often in the run. The Büchi automaton accepts a word w if there is some run of w that is accepting. The set of all words an automaton A accepts is called the *language of* A, or $L(A)$. A *complement* $\overline{A}$ of an automaton A is an automaton whose language is $\Sigma^\omega \backslash L(A)$. Finding the complement of an automaton is called *complementation.*

An automaton A is *contained* in an automaton B when $L(A) \subseteq L(B)$. In automata-theoretic verification [19], we prove that a program satisfies a specification by modeling the program as a Büchi automaton A and the specification as a Büchi automaton B, and then proving that A is contained in B. To check this containment, we check that the intersection of $L(A)$ with $L(\overline{B})$ is empty. If it is not empty, then a word in the intersection is a trace of A that violates the specification B. In practice, efficient containment algorithms do not explicitly construct the complement $\overline{B}$, using instead various strategies for on-the-fly complementation and symbolic construction, cf. [4]. Nevertheless, because these strategies are still fundamentally based on complementation, there is a close link between the efficiency of complementation and the efficiency of containment. The two complementation constructions that have been studied in the context of containment checking are the Ramsey-based construction of [16] and the *rank-based* construction of [9]. While the rank-based construction has a better worst-case complexity, the Ramsey-based approach is quite competitive in the context of containment checking [4]. Since the hard step in containment checking is the need to construct (at least implicitly) $\overline{B}$, papers on the subject, e.g. [4,17,18], usually focus on *universality checking*, where $L(A) = \Sigma^\omega$ – that is, checking if $L(B)$ contains the set of all words.

Evaluating Automata-Theoretic Algorithms. The quest for automata-theoretic algorithms that perform well in practice is hampered by a shortage of benchmark instances of automata that arise in industrial verification. The automaton B above corresponds to a formal specification of intended design functionality. Industrial specifications are typically proprietary and not openly available. To overcome this challenge, Tabakov and Vardi (T-V) proposed a model for generating random automata on which different algorithms can be evaluated and compared [17,18]. In subsequent years, this model has become the standard model for the evaluation of automata-theoretic tools, cf. [2,4,11,14]. Specifically, the T-V model was used in [4] to show that despite the worst-case-complexity gap between the Ramsey-based and the rank-based approaches, the two approaches are co-competitive in practice – that is, they each can outperform the other in non-trivial cases, depending on the properties of the automata being checked.

The T-V model generates automata using the *uniformly random* choice of elements from a set. The T-V model takes three parameters - an integral *size* n, a positive real *transition density* r, and a real *accepting-state density* f between 0 and 1. The transition density is the average out-degree of each state in the result automaton per input symbol. The accepting-state density is the percentage of states in the result automaton that are accepting states. Formally, a (n, r, f) T-V random automaton is defined as follows. Each random automaton $A = (\Sigma, Q, Q_0, \delta, F)$ has the alphabet $\Sigma = \{1, 0\}$ and set of states $Q = \{0, \ldots, n-1\}$. The set Q_0 of initial states is $\{0\}$. For each $\sigma \in \Sigma$, the model generates a digraph (directed graph) D_σ over the nodes $\{0, \ldots, n-1\}$ with $n * r$ edges chosen uniformly at random from the set of all possible edges $(u, v) \in Q \times Q$. The transition relation δ is then defined as $\{(u, \sigma, v) \mid (u, v) \in D_\sigma\}$. The accepting states F comprise $\lfloor n * f \rfloor$ states selected uniformly at random from Q. Note that each element of D_σ is a random digraph - specifically, a Karp [8] random digraph. Thus, we say that the T-V model *lifts* the Karp model of random digraphs into automata.

The T-V model is attractive for performance evaluation for two reasons [17,18]: First, the useful properties of its *universality terrain*, which describes the relationship between the probability of automaton universality (which means that all input words are accepted) and the density parameters. When transition and accepting-state densities are low, the probability for universality is low, while at higher densities the probability steadily increases. Thus, the model provides a way to evaluate the performance of universality-checking algorithms on both universal and non-universal automata. We call a model "*interesting*" when its universality probabilities vary with the input parameters and increase from low to high probability. Second, the model gives rise to an interesting *performance terrain*, which describes the relationship between algorithmic performance and the density parameters. Specifically, at low and high densities universality checking is easier than at intermediate densities. Thus, the model provides a way to evaluate the performance of universality-checking tools on both easy and hard problems. We take these two features, universality terrain and performance terrain to be desiderata that we expect to have in other models of random automata.

3 Random Models

Our goal in this work is to compare the T-V model to other models of random automata as a framework for evaluating the performance of universality-checking algorithms. We take advantage of the fact that the Tabakov-Vardi technique of *lifting* digraphs into automata is not limited to Karp random digraphs. By substituting other random-digraph models, we can generate new models of random automata.

The Tabakov-Vardi lifting is as follows. A random automata model that lifts a random digraph model has all of the parameters of the digraph model, plus an accepting-state density parameter f. Each random automaton is a tuple $(\Sigma, Q, Q_0, \delta, F)$, with the elements defined as follows. We take the alphabet

$\Sigma = \{0, 1\}$ for all models. For each character $\sigma \in \Sigma$, create a random digraph D_σ using the digraph parameter values of the automaton model. The set Q of states of the random automaton is equivalent to the set N of D_σ's nodes, usually $N = \{0, \ldots, n-1\}$, where n is the size parameter. The initial state set $Q_0 \subseteq Q$ is a singleton set containing one state from Q, usually 0. The transition relation δ is the union of all sets $\{(q, \sigma, r) \mid (q, r) \in D_\sigma\}$ for $\sigma \in \Sigma$. Finally, the set $F \subseteq Q$ of accepting states consists of $\lfloor |N| * f \rfloor$ elements of Q chosen uniformly at random (without repetition). Not all models we study use the Tabakov-Vardi lifting; see details below.

In the rest of this section, we introduce three[3] new models based on this lifting - the *vertex-copying* model, the *Frank-Strauss* model, and the *co-accessible* model. The first two models are based on existing models of structured random digraphs which have found common use in other disciplines, and the co-accessible model guarantees a particular automaton property. While the lack of existing benchmarks makes it difficult to compare these models directly to industrial problem instances, we can use a variety of structured random models to more fully explore the problem space. If these models disagree with the Tabakov-Vardi model, then the T-V model is not rich enough to fully represent the space on its own – if they agree, then it is likely that the conclusions of the T-V model are quite robust.

We show each of the models to have a universality terrain that is somewhat similar but not identical to that of the T-V model, using experiments run on the DAVinCI cluster[4] at Rice University. To show that each model has an interesting universality terrain, we present with each model a terrain plot showing how likely the automata generated by the model are to be universal when made with certain parameters. We generated and tested 100 automata using the parameters at each point on the plot. The universality terrains show that the random models we introduce generate automata whose likelihood of being universal ranges from 0 to 1, just as in the T-V model.

Vertex-Copying Automata. The random vertex-copying model presented here is a simplification of the model defined by Kleinberg *et al.* [7]. A vertex-copying digraph starts out as an empty set of nodes, and adds edges over time. By sometimes choosing edges at random, and at other times copying edges from one node to another, it creates a heavy-tailed distribution – a "rich get richer" effect as nodes with many edges steadily gain more and more edges. This copying is intended to model hyperlinks on the Web – links are often created when someone discovers a link to a site they're interested in on another site, then adds a link to it on their own website, thus "copying" the link from one site to another. This approach may also model code reuse - when a code block is reused, then calls to functions are duplicated.

An (n, b, r) vertex-copying random digraph takes as parameters the *size* n, the *copying probability* b, and the *transition density* r. The vertices are $\{0, \ldots, n-1\}$. The model begins with no edges and adds edges (u, v) to the

[3] Other models can be found in the technical report [3].
[4] http://www.rcsg.rice.edu/sharecore/davinci/.

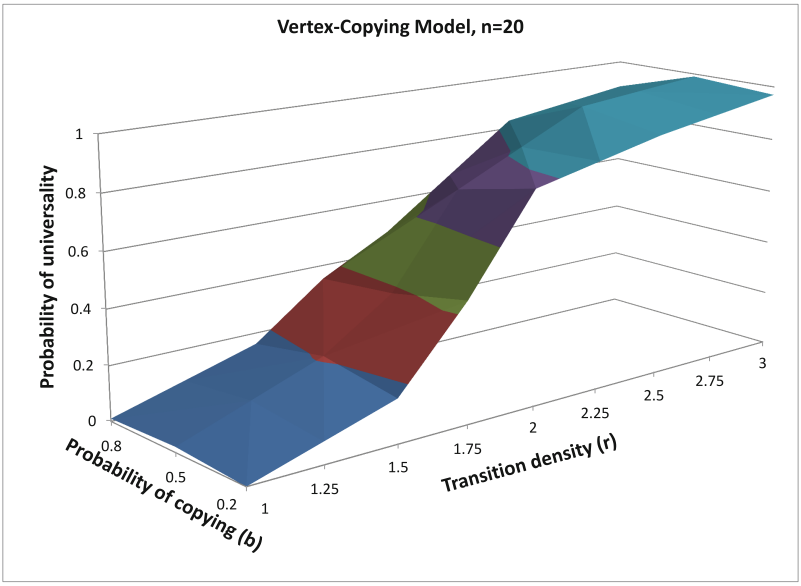


Fig. 1. A vertex-copying universality terrain for $n = 20$. The transition density r ranges from 1 to 3, and the copying probability b ranges from 0.2 to 0.8. The accepting-state density f was set to 0.3. The universality probability is comparable to that of the T-V model for most values of r. Note that increasing b does not monotonically increase universality probability – after a certain point it actually reduces it. This may be because all transitions go to a small number of states, with few transitions leaving them, increasing the likelihood of rejection.

graph one at a time until there are $\lfloor n * r \rfloor$ edges. Each time it does so, it has a probability b of copying an edge from one node to another, and a probability $1 - b$ of simply generating an edge uniformly at random. If it copies, then it chooses an edge $(u, v) \in E$ and a node $u' \in V \backslash u$ uniformly at random. It then adds (u', v) to E. If it generates the edge at random, it acts as in the T-V model. This digraph model extends to automata by directly using the standard lifting. Its universality terrain is given in Fig. 1.

Frank-Strauss Automata. The Frank-Strauss random graph model, based on an approach by Frank and Strauss[5] [6], limits the space of possible edges. Instead of the vertices being integers, vertices are unordered pairs of integers. The Frank-Strauss model permits edges only between vertices that share an element – the vertex $(0, 1)$ can connect to $(0, 3)$ and $(1, 3)$, but not to $(2, 3)$. Within this space, edges are generated uniformly at random. The Frank-Strauss model can represent systems that require some relationship between actors. For example, it can be used to represent binary relationships between individuals in a social setting. Alternatively, we may have a program such that if one module calls another,

[5] Referred to in their paper as a "Markov graph".

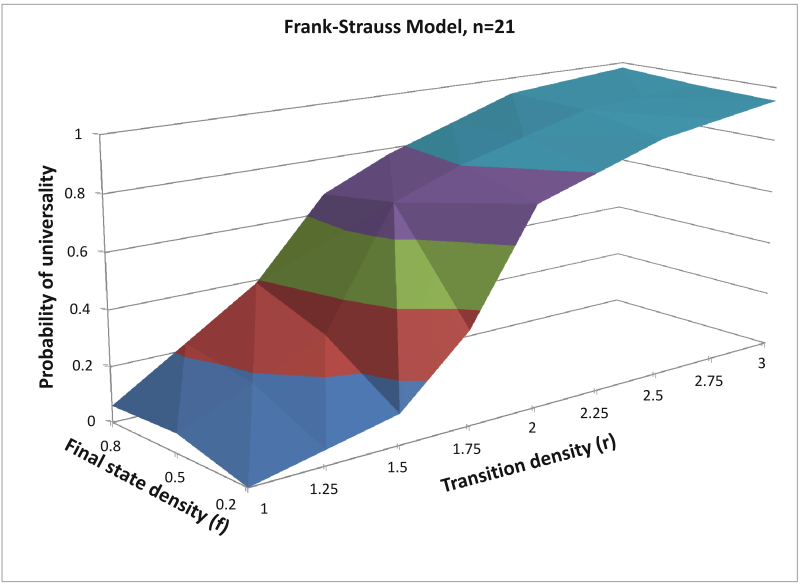


Fig. 2. A Frank-Strauss universality terrain for $l = 21$. r ranges from 1 to 3 and f ranges from 0.2 to 0.8. While the universality probably scales more quickly with r than in the T-V model, there are still a number of points where universality is neither nearly guaranteed nor always absent.

then there must be some relation between them – for example, operating on shared data.

An (l, r) Frank-Strauss random graph takes as parameters a *label size* l and a *transition density* r. The set V of vertices is the set $\{(i, j) \mid i, j \in 0, \ldots, l-1\}$ of unordered pairs of elements. Since we allow the case where $i = j$, there are $\binom{l+1}{2} = \frac{l(l+1)}{2}$ such vertices. We generate $\lfloor |V| * r \rfloor$ edges. To generate each edge, first choose a vertex (u_1, u_2) uniformly at random as the source, and then choose a vertex $(v_1, v_2) \in \{u_1, u_2\} \times \{0, \ldots, l\}$ uniformly at random as the destination. This digraph model extends to automata directly by using the standard lifting. The universality terrain is presented in Fig. 2.

Co-accessible Automata. The co-accessible model of random automata is so named because it guarantees that the resulting automata are co-accessible, where an automaton is *co-accessible* if all states $q \in Q$ are co-accessible, that is, can reach an accepting state. Because this property is meaningful only for automata, the co-accessible model cannot be based on lifting a model of random digraphs. It is loosely based on Leslie's generation of connected automata [10]. Automata possessing this property correspond to useful program properties – for example, a co-accessible automaton may specify that the program can recover and perform its intended function from every state.

The co-accessible model takes as parameters a *size* n, a *transition density* r, and an *accepting state density* f. The co-accessible model does not define the transition relation based on an underlying digraph. Instead, we start with a set $Q = \{0, \ldots, n-1\}$ of states and initial and accepting state sets Q_0 and F as in the T-V model. The transition relation δ is initially empty.

To fill in δ, we construct a random spanning inverted forest over Q. This is a set of trees over the automaton which contains every state, each rooted at an accepting state, and where edges go from children to parents instead of parents to children. A forest can be found as follows: make a set of co-accessible states $C = F$ and states that are not yet co-accessible $U = Q \backslash F$, then select some $u \in U$, $c \in C$ and $\sigma \in \Sigma$ uniformly at random. Add (u, σ, c) to δ, then remove u from U and add it to C, repeating until U is empty.

Once the spanning forest has been constructed, the model must fill in the rest of the transition relation. It then ensures that each character $\sigma \in \Sigma$ is associated with exactly $\lfloor n * r \rfloor$ edges. If some σ_0 has more than $\lfloor n * r \rfloor$ transitions, replace random transitions (u, σ_0, v) with (u, σ_1, v) for $\sigma_0 \neq \sigma_1$ and $\sigma_1 \in \Sigma$. Then generate new edges uniformly at random, as in the T-V model, for each character with fewer than $\lfloor n * r \rfloor$ transitions. We assume $r \geq 1$. The universality terrain is given in Fig. 3.

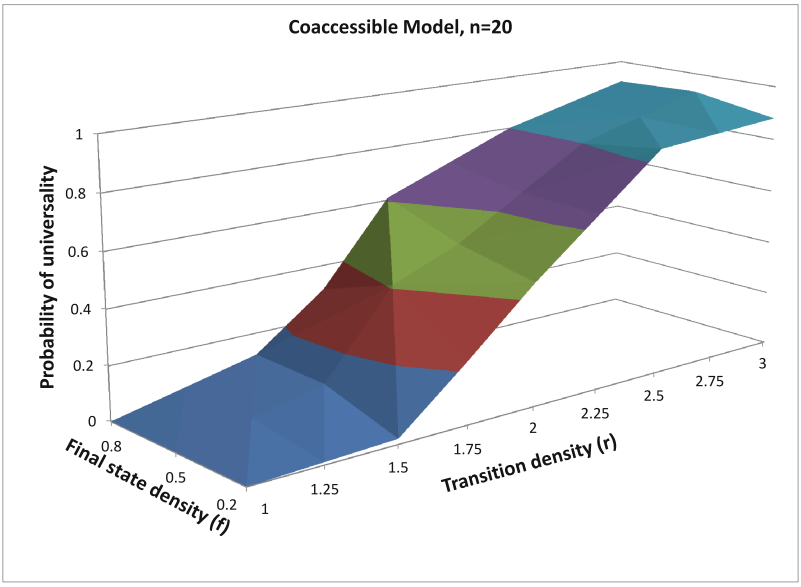


Fig. 3. A co-accessible universality terrain for $n = 20$. The transition density r ranges from 1 to 3, and f ranges from 0.2 to 0.8. Notice that the slope is much shallower than in previous models. This gives us an extremely wide range of useful configurations for testing.

4 Experiments

Methodology. Having defined three new random models and, via universality testing, proven them to be interesting for performance evaluation, we then used these models to run timing experiments for three universality checkers. We first compared the Rank and Ramsey tools[6] from [4]. To acquire a more recent picture of the comparison between algorithms, we also compared these tools with the RABIT 2.3 tool[7], a more recent Ramsey-based containment checker. As in the previous section, experiments were run on the DAVinCI cluster at Rice University, which consists of many Westmere nodes with 2.83 GHz processors and 48 GB of memory per node. We limited each job to 30 GB of memory and one hour of time. Jobs that did not finish were marked as timeouts.

We ran two types of experiments: terrain experiments and scaling experiments. In terrain experiments, the size of the automata is held constant, and two other parameters are changed to see the effects on running time. In scaling experiments, all parameters are held constant except those affecting the size of the automaton, and we steadily increase the size to see how the implementations respond to larger problems. We conduct scaling experiments with parameters that are particularly difficult for at least one tool to handle, as determined by the terrain experiments, to test practical worst-case performance. We generated 100 automata using each combination of parameter values in both kinds of experiments, and report median running time.

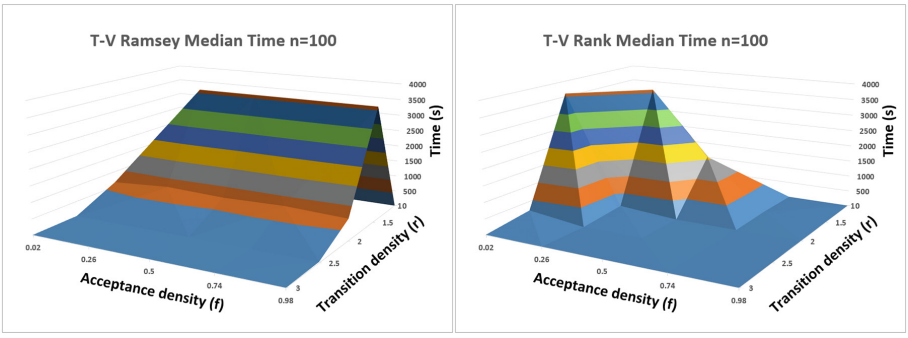


Fig. 4. For terrain experiments on the Tabakov-Vardi model, we tested parameter values of $n = 100$, or $l = 14$, $r \in \{1, 1.5, 2, 2.5, 3\}$, and $f \in \{0.02, 0.26, 0.5, 0.74, 0.98\}$. These graphs show results for the Rank and Ramsey tools. Note that Rank and Ramsey are not directly comparable - Ramsey tends to be slower at points where $r = 1.5$ and $r = 2$, while Rank tends to be slower at $f = 0.02$ and $f = 0.26$. This agrees with previous results [4] using the Tabakov-Vardi model.

[6] https://www.cs.rice.edu/CS/Verification/Software/software.html.
[7] http://www.languageinclusion.org/doku.php?id=tools.

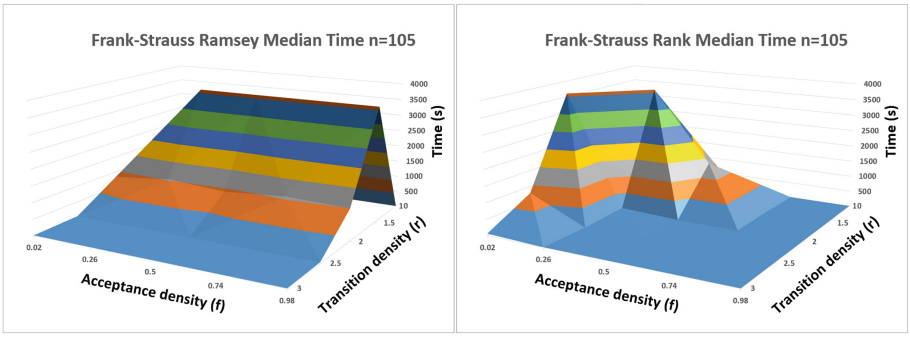


Fig. 5. For terrain experiments on the Frank-Strauss model, we tested parameter values of $n = 105$, or $l = 14$, $r \in \{1, 1.5, 2, 2.5, 3\}$, and $f \in \{0.02, 0.26, 0.5, 0.74, 0.98\}$. These graphs show results for the Rank and Ramsey tools. Again, the Rank model tends to perform the slowest at low f and low r, while Ramsey is slowest at $r = 2$. This agrees with our results on the Tabakov-Vardi model, as do the terrains of other models found in the technical report [3].

Results. We find both that choice of model does not seriously impact tool comparisons, and that RABIT noticeably outperforms Rank and Ramsey.

In both terrain (Figs. 4, 5 and 6) and scaling (Fig. 7) experiments, we find that the relative efficiency of tools is very similar across models. All models show that, as in the Tabakov-Vardi model in [4], the Rank and Ramsey are not directly comparable – which parameters are used to generate an automaton determine which tool solves it most efficiently, as seen in the terrain experiments in Fig. 5. Since all models agree with T-V here, it is reasonable to use the T-V model to compare tools. Nevertheless, while models agree on the comparison between tools, they do not have the same running time. For example, in Fig. 7, we see on a log scale that there is a factor 10 difference between the running time of Ramsey on the Tabakov-Vardi and co-accessible models. Thus, the T-V model should be relied on for relative comparisons, but not for predicting runtimes.

Since there was little difference in comparison between models, Rank and Ramsey compare similarly to their results in [4]. Yet, when we compare Rank to RABIT, we saw a massive speedup at all difficult points – sometimes thousands of times faster. At $n = 100$, the terrain was flat, with most cases terminating in just over a tenth of a second. Therefore, the improved modern Ramsey tools are more suited for practical use than Rank-based ones. However, as seen in Fig. 6, random models can still provide interesting performance terrain on the more efficient tools by scaling up the size of the problems.

There is one noticeable difference between algorithms not shown – both Ramsey-based algorithms used much more memory than Rank did. When provided with 5 GB of memory, the Rank tool performed acceptably, but Ramsey and RABIT crashed regularly. 30 GB of memory provided was necessary to avoid crashes due to running out of memory.

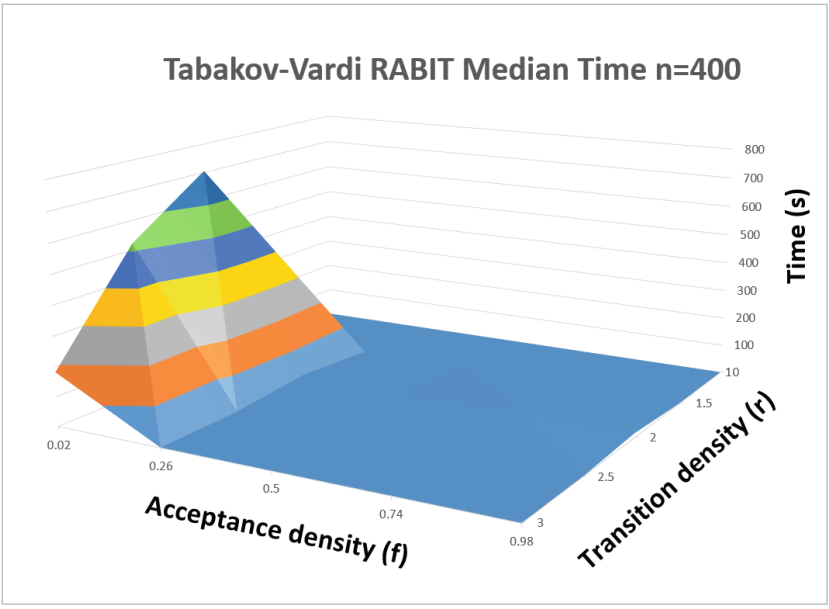


Fig. 6. For all terrain experiments at $n = 100$ for RABIT, we found that the terrain was entirely flat - very few problems took more than one second to terminate. Therefore we show results for RABIT on $n = 400$, instead, with parameter values $r \in \{1, 1.5, 2, 2.5, 3\}$, and $f \in \{0.02, 0.26, 0.5, 0.74, 0.98\}$. Note that the maximum Y-axis value is only 800 seconds, because at no point was the median result a timeout. RABIT has the most difficulty at high transition density and extremely low acceptance densities, with orders of magnitude slower performance on $f = 0.02$. While it does not appear on this graph, we also find that RABIT takes about two orders of magnitude more time at $r = 2.0$ and high f than other areas, and one order of magnitude less than the extremely difficult areas. Also, we find that at $r = 1.5$, we consistently had a small (5 %) chance of timeouts at all values of n tested with few to no timeouts elsewhere, though the median time taken was no higher.

5 Concluding Remarks

While formal verification provides important software tools, it has been unclear whether these tools are efficient enough to be used in practice. Thus, the T-V model is a powerful tool for automata-theoretic formal verification, allowing us to test the efficiency of algorithms for determining conformance to a specification. Due to concerns about whether the model accurately reflected real-world performance, we tested other models to see if the structure of a problem would influence the results; we found that it did not. Future work in the area can proceed to test algorithms and tools on the T-V model, more confident that it is robust and that its results are widely applicable.

This work gives reason to believe that the Tabakov-Vardi model is a robust model with results that are likely to be close to the real-world. Complementation,

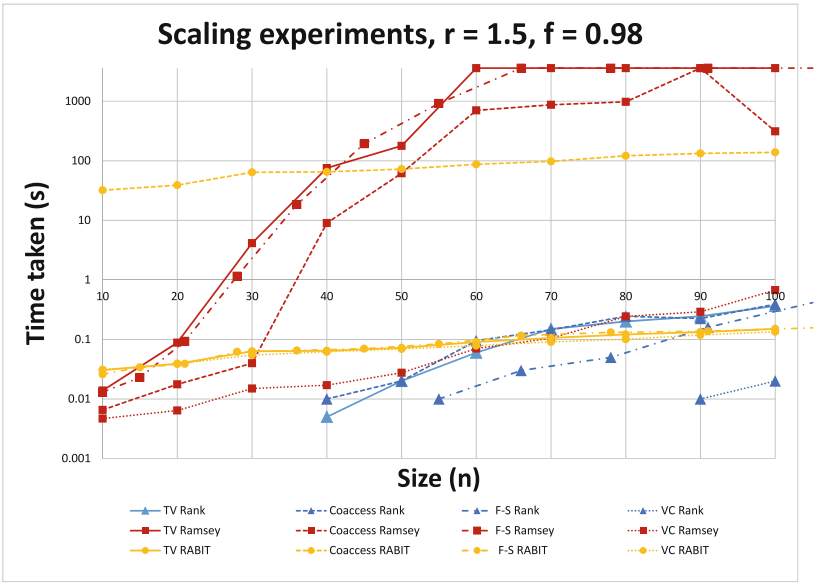


Fig. 7. For this set of scaling experiments, we set $r = 1.5$ and $f = 0.98$, and scale n from 10 to 100. In the Frank-Strauss model, l scales from 4 to 14. This point was chosen for scaling because it is particularly difficult for Ramsey. On this log-scale plot, different tools (indicated by shared color and marker shape) tend to have similar slopes regardless of model (indicated by shared line style). Notably, an obvious exponential gap exists between other models and Ramsey at these parameters for every model except the trivally-easy vertex-copying model. Since f is high, this is an easy point for Rank. The relationship between tools found by T-V is also reflected in the other random models shown here.

and thus containment checking, should be practical on real-world problems. We also discovered an improvement of many orders of magnitude in modern containment checkers using a Ramsey-based approach. RABIT outperformed both older Ramsey and rank-based tools significantly, and can scale up much higher. Since little work has been done on rank-based solvers since 2010, current heuristics-driven Ramsey-based approaches are the best available options for containment checking for Büchi automata.

Acknowledgements. Work supported in part by NSF grants CCF-1319459 and IIS-1527668, by NSF Expeditions in Computing project "ExCAPE: Expeditions in Computer Augmented Program Engineering", as well as the Data Analysis and Visualization Cyberinfrastructure funded by NSF grant OCI-0959097 and Rice University.

References

1. Büchi, J.R.: Turing-machines and the Entscheidungsproblem. Math. Ann. **148**(3), 201–213 (1962)
2. Doyen, L., Raskin, J.: Antichains for the automata-based approach to model-checking. arXiv preprint arXiv:0902.3958 (2009)
3. Fisher, C., Fogarty, S., Vardi, M.: Random models for efficient Büchi universality checking. Technical report. Department of Computer Science, Rice University, Houston, TX, October 2016. http://www.cs.rice.edu/~vardi
4. Fogarty, S., Vardi, M.Y.: Efficient Büchi Universality Checking. In: Esparza, J., Majumdar, R. (eds.) TACAS 2010. LNCS, vol. 6015, pp. 205–220. Springer, Heidelberg (2010). doi:10.1007/978-3-642-12002-2_17
5. Fogarty, S., Vardi, M.Y.: Büchi complementation and size-change termination. In: Kowalewski, S., Philippou, A. (eds.) TACAS 2009. LNCS, vol. 5505, pp. 16–30. Springer, Heidelberg (2009). doi:10.1007/978-3-642-00768-2_2
6. Frank, O., Strauss, D.: Markov graphs. J. Am. Stat. Assoc. **81**(395), 832–842 (1986)
7. Kleinberg, J., Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A.: The web as a graph: measurements, models, and methods. In: Asano, T., Imai, H., Lee, D.T., Nakano, S., Tokuyama, T. (eds.) COCOON 1999. LNCS, vol. 1627, pp. 1–17. Springer, Heidelberg (1999). doi:10.1007/3-540-48686-0_1
8. Karp, R.M.: The transitive closure of a random digraph. Random Struct. Alg. **1**(1), 73–93 (1990)
9. Kupferman, O., Vardi, M.Y.: Weak alternating automata are not that weak. ACM Trans. Comput. Logic (TOCL) **2**(3), 408–429 (2001)
10. Leslie, T.: Efficient approaches to subset construction. Technical report. University of Waterloo, Canada (1995)
11. de Wulf, M., Doyen, L., Henzinger, T.A., Raskin, J.-F.: Antichains: a new algorithm for checking universality of finite automata. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 17–30. Springer, Heidelberg (2006). doi:10.1007/11817963_5
12. Tsai, M.-H., Fogarty, S., Vardi, M.Y., Tsay, Y.-K.: State of Büchi complementation. In: Domaratzki, M., Salomaa, K. (eds.) CIAA 2010. LNCS, vol. 6482, pp. 261–271. Springer, Heidelberg (2011). doi:10.1007/978-3-642-18098-9_28
13. Michel, M.: Complementation is more difficult with automata on infinite words. CNET, Paris (1988). 15
14. Abdulla, P.A., Chen, Y.-F., Clemente, L., Holík, L., Hong, C.-D., Mayr, R., Vojnar, T.: Advanced ramsey-based Büchi automata inclusion testing. In: Katoen, J.-P., König, B. (eds.) CONCUR 2011. LNCS, vol. 6901, pp. 187–202. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23217-6_13
15. Safra, S.: On the complexity of ω-automata. In: 29th Annual Symposium on Foundations of Computer Science, pp. 319–327. IEEE (1988)
16. Sistla, A.P., Vardi, M.Y., Wolper, P.: The complementation problem for Büchi automata with applications to temporal logic. Theor. Comput. Sci. **49**(2), 217–237 (1987)
17. Tabakov, D., Vardi, M.Y.: Experimental evaluation of classical automata constructions. In: Sutcliffe, G., Voronkov, A. (eds.) LPAR 2005. LNCS, vol. 3835, pp. 396–411. Springer, Heidelberg (2005). doi:10.1007/11591191_28
18. Tabakov, D., Vardi, M.Y.: Model checking Büchi specifications. In: Proceedings of 1st International Conference on Language and Automata Theory and Applications, pp. 565–576 (2007)

19. Vardi, M., Wolper, P.: An automata-theoretic approach to automatic program verification. In: Proceedings of the First Symposium on Logic in Computer Science, pp. 322–331. IEEE Computer Society (1986)
20. Vardi, M.Y.: The Büchi complementation saga. In: Thomas, W., Weil, P. (eds.) STACS 2007. LNCS, vol. 4393, pp. 12–22. Springer, Heidelberg (2007). doi:10.1007/978-3-540-70918-3_2
21. Vardi, M.Y., Wolper, P.: Reasoning about infinite computations. Inf. Comput. **115**(1), 1–37 (1994)

A Substructural Epistemic Resource Logic

Didier Galmiche[1], Pierre Kimmel[1], and David Pym[2(✉)]

[1] Université de Lorraine, LORIA, Nancy, France
[2] University College London, London, UK
d.pym@ucl.ac.uk

Abstract. We present a substructural epistemic logic, based on Boolean BI, in which the epistemic modalities are parametrized on agents' local resources. The new modalities can be seen as generalizations of the usual epistemic modalities. The logic combines Boolean BI's resource semantics with epistemic agency. We give a labelled tableaux calculus and establish soundness and completeness with respect to the resource semantics. We illustrate the use of the logic by discussing an example of side-channels in access control using resource tokens.

1 Introduction

The concept of resource is important in many fields including, among others, computer science, economics, and security. For example, in operating systems, processes access system resources such as memory, files, processor time, and bandwidth, with correct resource usage being essential for the robust function of the system. The internet can be regarded as a giant, dynamic net of resources, in which Uniform Resource Locators refer to located data and code. In recent years, the concept of resource has been studied and analysed in computer science through the bunched logic, BI, [14] and its variants, such as Boolean BI (BBI) [15] and applications, such as Separation Logic [15,21]. The *resource semantics* — i.e., the interpretation of BI's semantics in terms of resources — that underpins these logics is mainly concerned sharing and separation, corresponding to additive, such as $\wedge$, and multiplicative connectives, such as $*$, respectively. These logics are the logical kernels of the separating, or separation, logics, with resources being interpreted in various ways, such as memory regions, [15,21] or elements of other particular monoids of resources [3].

The logic BI of bunched implications — see, for example, [11,14,20] — freely combines intuitionistic propositional additives with intuitionistic propositional multiplicatives. In Boolean BI (BBI) [15], the additives are classical. The key feature of BI as a modelling tool, and hence of its specific model Separation Logic, is its control of the representation and handling of resources provided by the resource semantics and the associated proof systems. BI's basic propositional connectives come in two groups. The additives, which can be handled either classically or intuitionistically, are familiar disjunction, conjunction, and implication. For example,

$$r \models \phi \wedge \psi \text{ iff } r \models \phi \text{ and } r \models \psi.$$

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 106–122, 2017.
DOI: 10.1007/978-3-662-54069-5_9

The key point here is that the resource r is *shared* between the two components of the disjunction.

In contrast, the multiplicative conjunction, $*$, divides the resource between its propositional components, using a partial commutative monoidal operation, $\circ$,

$$r \models \phi * \psi \text{ iff there are } s \text{ and } t \text{ such that } r = s \circ t \text{ and } s \models \phi \text{ and } t \models \psi.$$

That is, the monoid specifies a *separation* of the resources between the components of the conjunction. In Separation Logic, where the semantics is built out of sets of memory locations, the two resource components are required to be disjoint. Details may be found in the references given above.

BI's sequent proof systems employ *bunches*, with two context-building operations: one for the additives (characterized by $\wedge$, which admits weakening and contraction) and one for the multiplicatives (characterized by $*$, which admits neither weakening nor contraction), leading to the following rules for the corresponding implications, $\rightarrow$ and $-\!\!*$:

$$\frac{\Gamma\,;\,\phi \vdash \psi}{\Gamma \vdash \phi \rightarrow \psi} \quad \text{and} \quad \frac{\Gamma\,,\,\phi \vdash \psi}{\Gamma \vdash \phi -\!\!* \psi}.$$

Again, details may be found in the references given above.

The soundness and completeness of BI for the semantics given above is established in [20] and via labelled tableaux in [11], and the completeness of BBI for the partial monoid semantics described above is established in [16].

Modal extensions of BI, such as MBI [1,3], DBI, and DMBI [6], have been proposed to introduce dynamics into resource semantics. In recent work, the idea of introducing agents, together with their knowledge, into the resource semantics has led to an Epistemic Separation Logic, called ESL, in which epistemic possible worlds are considered as resources [7]. This logic corresponds to an extension of Boolean BI with a knowledge modality, K_a, such that $K_a\,\phi$ means that the agent a knows that ϕ holds.

Various previous works on epistemic logics consider the concept of resource, using a variety of approaches. They include [2,13,17]. Here we aim to explore more deeply the idea of epistemic [9] reasoning in the context of resource semantics, and its associated logic, by taking the basic epistemic modality K_a and parametrizing it with a resource s, with the associated introduction of relations not only between resources, according to an agent, but also between composition of resources in different ways. The parametrizing resource may be thought of as being associated with, or local to, the agent. This approach leads to the definition of three new modalities $\mathbf{L}^s_a$, $\mathbf{M}^s_a$, and $\mathbf{N}^s_a$ and, consequently, to a new logic in which, as a leading example, we can obtain an account of access to resources and its control, whether they be pieces of knowledge, locations, or other entities. We call this logic *Epistemic Resource Logic* or ERL.

In Sect. 2, we set up the logic ERL by a semantic definition and, in Sect. 3, we give the key conservative extension properties of the logic. In Sect. 4, we explain, how to use the logic to model and reason about the relationship between a security policy — in the context of access control — and the system to which it

is applied (cf. [22]). Our application to systems security policy stands in contrast to other work (e.g., [19]) in which epistemic logic has been applied to the analysis of cryptographic protocols. In Sect. 5, we set up a labelled tableaux calculus for ERL, and establish soundness with respect to ERL's semantic definition and also completeness from a countermodel extraction method. Details of the arguments are provided in [12].

2 An Epistemic Resource Logic

The language $\mathcal{L}$ of the epistemic resource logic, or ERL, is obtained by adding two new modal operators $\mathbf{L}$ and $\mathbf{M}$ to the BI language. In order to define the language of ERL, we introduce the following structures: a finite set of agents A; a finite set of resources Res, with a particular element, e; an internal composition operator $\cdot$ on Res ($\cdot : Res \times Res \rightharpoonup Res$); a countable set of propositional symbols Prop. The language $\mathcal{L}$ of ERL is defined as follows:

$$\phi ::= \mathrm{p} \mid \bot \mid \top \mid \neg\phi \mid \mathrm{I} \mid \phi \vee \psi \mid \phi \wedge \psi \mid \phi \rightarrow \phi \mid \phi * \phi \mid \phi \mathrel{-\!\!*} \phi \mid \mathbf{L}^s_a \phi \mid \mathbf{M}^s_a \phi,$$

where $\mathrm{p} \in \mathrm{Prop}$, $a \in A$ and $s \in Res$. We also define the following operators: $\mathbf{N}^s_a \phi \equiv \mathbf{L}^s_a(\mathbf{M}^s_a \phi)$, $\widetilde{\mathbf{M}}^s_a \phi \equiv \neg\mathbf{M}^s_a\neg\phi$, $\widetilde{\mathbf{L}}^s_a \phi \equiv \neg\mathbf{L}^s_a\neg\phi$, $\widetilde{\mathbf{N}}^s_a \phi \equiv \neg\mathbf{N}^s_a\neg\phi$. The meanings of these connectives are defined in the sequence of definitions that follow below. For simplicity, we write rs instead of $r \cdot s$ and so write $\mathbf{L}^{rs}_a \phi$ instead of $\mathbf{L}^{r\cdot s}_a \phi$.

Note that we introduce modalities that depend on agents and resources, and compare them with previous work on an epistemic extension of Boolean BI [7]. With a slight abuse of notation, we have explicit resources in the language syntax: just as in [8], we must assume that the resource elements present in the syntax of the modalities have counterparts in the partial resource monoid semantics. This design choice has consequences both for the expressivity of the logic and for the formulation of the tableaux calculus.

Definition 1 (Partial resource monoid). *A* partial resource monoid *(PRM) is a structure $\mathcal{R} = (R, \bullet)$ such that*

- *R is a set of* resources *such that $Res \subseteq R$ (which notably means that $e \in R$), and*
- *$\bullet : R \times R \rightharpoonup R$ is an operator on R such that, for all $r_1, r_2, r_3 \in R$,*
  - *$\bullet$ is an extension of $\cdot$: if $r_1, r_2, r_3 \in Res$, then $r_1 = r_2 \cdot r_3$ iff $r_1 = r_2 \bullet r_3$,*
  - *e is a neutral element: $r_1 \bullet e \downarrow$ and $r_1 \bullet e = r_1$,*
  - *$\bullet$ is commutative: if $r_1 \bullet r_2 \downarrow$, then $r_2 \bullet r_1 \downarrow$ and $r_2 \bullet r_1 = r_1 \bullet r_2$, and*
  - *$\bullet$ is associative: if $r_1 \bullet (r_2 \bullet r_3) \downarrow$, then $(r_1 \bullet r_2) \bullet r_3 \downarrow$ and $(r_1 \bullet r_2 \bullet) r_3 = r_1 \bullet (r_2 \bullet r_3)$.*

Here $r \bullet r' \downarrow$ means $r \bullet r'$ is defined. We call e the *unit resource* and $\bullet$ the *resource composition*. Henceforth, $\wp(S)$ denotes the powerset of S.

Definition 2 (Model). *A* model *is a triple* $\mathcal{M} = (\mathcal{R}, \{\sim_a\}_{a\in A}, V)$ *such that*

- $\mathcal{R} = (R, \bullet)$ *is a PRM,*
- *for all* $a \in A$, $\sim_a \subseteq R \times R$ *is an equivalence relation, and*
- $V : \text{Prop} \to \wp(R)$ *is a valuation function.*

We can place this logic in the context of our previous work on modal [3,4] and epistemic extensions of (Boolean) BI [6,7]. In [7], an epistemic extension of Boolean BI, called ESL, is introduced. In this logic, there is just one epistemic modality, K_a, which allows the knowledge of an agent a to be expressed. More formally, the semantics of this modality is defined by $r \models_{\mathcal{W}} K_a\,\phi$ if and only if, for all r' such that $r \sim_a r'$, $r' \models_{\mathcal{W}} \phi$, where r and r' are semantic worlds (or resources) and $\sim_a$ is a relation between worlds that expresses that they are equivalent from the point of view of the agent a. This parametrization of modality on resource derives from ideas that are conveniently expressed in, for example, [3,4].

In this paper, we aim to develop the idea in order to consider a modality like K_a and to parametrize it on a resource s, requiring the world relation to be of the form $r \bullet s \sim_a r'$ or $r \sim_a r' \bullet s$ or even $r \bullet s \sim_a r' \bullet s$. Then, in the spirit of ESL, we define a new logic from Boolean BI that allows us to model not only relations between resources according to an agent, but also how those relations are restricted by resources. We can also consider the resources upon which the agent's relation are parametrized to be local to the agent.

In this spirit, we define three new modalities $\mathbf{L}^s_a\,\phi$, $\mathbf{M}^s_a\,\phi$, and $\mathbf{N}^s_a\,\phi$, for which we have the following semantics expressing the evident three forms of the agent's contingency for truth in the presence of composable resources:

1. $\mathbf{L}^s_a\,\phi$ expresses that the agent, a, can establish the truth of ϕ using a given resource whenever the ambient resource, r, can be combined with the agent's local resource, s, to yield a resource that a judges to be equivalent to that given resource:

$$r \models_{\mathcal{W}} \mathbf{L}^s_a\,\phi \text{ iff for all } r' \text{ such that } r' \sim_a r \bullet s,\ r' \models_{\mathcal{W}} \phi.$$

2. $\mathbf{M}^s_a\,\phi$ expresses that the agent, a, can establish the truth of ϕ using a resource that is the combination of its local resource, s, with any resource such that a judges the combined resource to be equivalent to the ambient resource, r:

$$r \models_{\mathcal{W}} \mathbf{M}^s_a\,\phi \text{ iff for all } r' \text{ such that } r' \bullet s \sim_a r,\ r' \bullet s \models_{\mathcal{W}} \phi.$$

3. $\mathbf{N}^s_a\,\phi$ expresses that the agent, a, can establish the truth of ϕ using any resource combined with its local resource, s, provided a judges that combination to be equivalent to the combination of that resource with the ambient resource, r:

$$r \models_{\mathcal{W}} \mathbf{N}^s_a\,\phi \text{ iff for all } r' \text{ such that } r' \bullet s \sim_a r \bullet s,\ r' \bullet s \models_{\mathcal{W}} \phi.$$

ERL can thus be seen as a particular epistemic logic that provides new modalities which model access to resources, whether they are interpreted as pieces of knowledge, locations, or otherwise.

Definition 3 (Satisfaction and validity). *Let $\mathcal{M} = (\mathcal{R}, \{\sim_a\}_{a\in A}, V)$ be a model. The satisfaction relation $\models_w \subseteq R \times \mathcal{L}$ is defined, for all $r \in R$, as follows:*

$$\begin{array}{ll} r \models_w \mathrm{p} & \text{iff} \quad r \in V(\mathrm{p}) \\ r \models_w \bot & \text{never} \\ r \models_w \top & \text{always} \\ r \models_w \neg\phi & \text{iff} \quad r \not\models_w \phi \end{array} \qquad \begin{array}{l} r \models_w \phi \vee \psi \text{ iff } r \models_w \phi \text{ or } r \models_w \psi \\ r \models_w \phi \wedge \psi \text{ iff } r \models_w \phi \text{ and } r \models_w \psi \\ r \models_w \phi \to \psi \text{ iff if } r \models_w \phi, \text{ then } r \models_w \psi \end{array}$$

$$\begin{array}{l} r \models_w \mathrm{I} \text{ iff } r = e \\ r \models_w \phi * \psi \text{ iff there exist } r_1, r_2 \in R \text{ s.t. } r_1 \bullet r_2 \downarrow,\ r_1 \bullet r_2 = r, \text{ and } r_1 \models_w \phi \text{ and } r_2 \models_w \psi \\ r \models_w \phi \mathrel{-\!\!*} \psi \text{ iff for all } r' \in R, \text{ if } r \bullet r' \downarrow \text{ and } r' \models_w \phi, \text{ then } r \bullet r' \models_w \psi \end{array}$$

$$\begin{array}{l} r \models_w \mathbf{L}^s_a \phi \text{ iff for all } r' \in R, \text{ if } r \bullet s \sim_a r', \text{ then } r' \models_w \phi \\ r \models_w \mathbf{M}^s_a \phi \text{ iff for all } r' \in R, \text{ if } r \sim_a r' \bullet s, \text{ then } r' \bullet s \models_w \phi \\ r \models_w \mathbf{N}^s_a \phi \text{ iff } \textit{for all } r' \textit{ such that } r' \bullet s \sim_a r \bullet s, r' \bullet s \models_w \phi\,. \end{array}$$

A formula ϕ is valid, *denoted $\vDash \phi$, if and only if, for all $\mathcal{M}$ and all r, $r \models_w \phi$.*

Note that $\mathbf{N}^s_a \phi \equiv \mathbf{L}^s_a(\mathbf{M}^s_a \phi)$. To see this, consider that $r \models_w \mathbf{L}^s_a(\mathbf{M}^s_a \phi)$ iff, for all $r' \in R$, if $r \bullet s \sim_a r'$, then $r' \models_w \mathbf{M}^s_a \phi$ iff, for all $r' \in R$, if $r \bullet s \sim_a r'$, then, for all $r'' \in R$, if $r' \sim_a r'' \bullet s$, then $r'' \bullet s \models_w \phi$ iff, for all $r', r'' \in R$, if $r \bullet s \sim_a r'$ and $r' \sim_a r'' \bullet s$, then $r'' \bullet s \models_w \phi$ iff (by the transitivity of $\sim_a$), for all $r'' \in R$, if $r \bullet s \sim_a r'' \bullet s$, then $r'' \bullet s \models_w \phi$ iff $r \models_w \mathbf{N}^s_a \phi$.

Proposition 1 (Satisfaction for the secondary modalities). *Let $\mathcal{M} = (\mathcal{R}, \{\sim_a\}_{a\in A}, V)$ be a model, and let $r \in R$. The following statements hold:*

1. $r \models_w \widetilde{\mathbf{L}}^s_a \phi$ *iff there exists $r' \in R$ such that $r \bullet s \sim_a r'$ and $r' \models_w \phi$;*
2. $r \models_w \widetilde{\mathbf{M}}^s_a \phi$ *iff there exists $r' \in R$ such that $r \sim_a r' \bullet s$ and $r' \bullet s \models_w \phi$;*
3. $r \models_w \widetilde{\mathbf{N}}^s_a \phi$ *iff there exists $r' \in R$ such that $r \bullet s \sim_a r' \bullet s$ and $r' \bullet s \models_w \phi$.*

Proof. For example, consider the first part. $\widetilde{\mathbf{L}}^s_a \phi \equiv \neg\mathbf{L}^s_a\neg\phi$, so $r \models_w \widetilde{\mathbf{L}}^s_a \phi$ iff $r \models_w \neg\mathbf{L}^s_a\neg\phi$ iff $r \not\models_w \mathbf{L}^s_a\neg\phi$ iff there exists $r' \in R$ s.t. $r \bullet s \sim_a r'$ and $r' \not\models_w \neg\phi$ iff there exists $r' \in R$ s.t. $r \bullet s \sim_a r'$ and $r' \models_w \phi$. Parts 2 and 3 are similar.

Note that the first point of the definition of $\bullet$, in Definition 1, implies that the three other definitions (neutral element, commutativity, and associativity) extend to $\cdot$, so that the following are semantically equivalent (i.e., every valid formula in the one is valid in the other) for any agent a and any resources r, s, and t: $\mathbf{L}^{re}_a \phi \equiv \mathbf{L}^r_a \phi$, $\mathbf{L}^{rs}_a \equiv \mathbf{L}^{sr}_a$, and $\mathbf{L}^{r(st)}_a \equiv \mathbf{L}^{(rs)t}_a$. Of course, these equivalences also hold for $\mathbf{M}\,\phi$, $\mathbf{N}\,\phi$, $\widetilde{\mathbf{L}}\,\phi$, $\widetilde{\mathbf{M}}\,\phi$, and $\widetilde{\mathbf{N}}\,\phi$.

3 Some Properties of ERL

Consider two fragments of ERL. First, $\mathrm{ERL}_{\mathrm{BBI}}$ — corresponding to BBI [15] — with $A = \emptyset$ on the language $\mathcal{L}_{|BBI}$ defined as $\mathcal{L}$ excluding the $\mathbf{L}$, $\mathbf{M}$, and $\mathbf{N}$ operators. Second, $\mathrm{ERL}_{\mathrm{EL}}$ — corresponding to the epistemic logic EL consisting

of classical propositional additives and the basic epistemic operator $\mathbf{K}_a$ [9] — with $Res = \{e\}$, on the language $\mathcal{L}_{|EL}$ defined as $\mathcal{L}$ excluding I, $*$, and $-\!\!*$ and with $\mathbf{L}$, $\mathbf{M}$, and $\mathbf{N}$ replaced by the operator $\mathbf{K}_a$, which is defined, for all agents a, by $\mathbf{K}_a\,\phi = \mathbf{L}^e_a\,\phi = \mathbf{M}^e_a\,\phi$.

Proposition 2 (ERL is a conservative extension of BBI and EL). *If, in every model of BBI, the neutral element of the composition is the element e of Res, then* $\mathrm{ERL}_{\mathrm{BBI}}$ *is semantically equivalent to Boolean BI (BBI). If the agent sets are the same for the two languages,* $\mathrm{ERL}_{\mathrm{EL}}$ *is semantically equivalent to the epistemic logic EL.*

Definition 4. *The logic* ERL^* *is defined as ERL but with the addition of the two following properties to the partial resource monoid (Definition 1): 1. $\bullet$ has the right-composition property, namely, if $r_1 = r_2$ and $r_1 \bullet r_3 \downarrow$, then $r_2 \bullet r_3 \downarrow$ and $r_2 \bullet r_3 = r_1 \bullet r_3$; 2. $\bullet$ has the right-cancellation property, namely, if $r_1 \bullet r_3 = r_2 \bullet r_3$, then $r_1 = r_2$.*

Note that left-cancellation and left-composition also hold trivially, as $\bullet$ is commutative.

Lemma 1. *Let $a \in A$ be an agent, $r, s \in Res$ be resources and ϕ be a formula of* ERL^**. We have the following equalities:*

1. $\mathbf{L}^t_a(\mathbf{L}^s_a\,\phi) \equiv \mathbf{L}^{ts}_a\,\phi$
2. $\mathbf{M}^t_a(\mathbf{M}^s_a\,\phi) \equiv \mathbf{M}^s_a\,\phi$
3. $\mathbf{M}^t_a(\mathbf{L}^s_a\,\phi) \equiv \mathbf{L}^s_a\,\phi$
4. $\mathbf{N}^t_a(\mathbf{N}^s_a\,\phi) \equiv \mathbf{L}^t_a(\mathbf{N}^s_a\,\phi)$
5. $\mathbf{L}^e_a\,\phi \equiv \mathbf{M}^e_a\,\phi \equiv \mathbf{N}^e_a\,\phi$
6. $\widetilde{\mathbf{L}}^t_a(\widetilde{\mathbf{L}}^s_a\,\phi) \equiv \widetilde{\mathbf{L}}^{ts}_a\,\phi$
7. $\widetilde{\mathbf{M}}^t_a(\widetilde{\mathbf{M}}^s_a\,\phi) \equiv \widetilde{\mathbf{M}}^s_a\,\phi$
8. $\widetilde{\mathbf{M}}^t_a(\widetilde{\mathbf{L}}^s_a\,\phi) \equiv \widetilde{\mathbf{L}}^s_a\,\phi$
9. $\widetilde{\mathbf{N}}^t_a(\widetilde{\mathbf{N}}^s_a\,\phi) \equiv \widetilde{\mathbf{L}}^t_a(\widetilde{\mathbf{N}}^s_a\,\phi)$.

Proof. Straightforward calculations using the semantic definitions of the modalities.

4 Modelling with the Logic ERL

Using a very simple, well-known example, we illustrate how to use ERL, and its special fragment ERL^*, in modelling access control situations. There is often a gap between theory and practice when dealing with security matters. Specifically, when a particular security *policy* is applied to a particular *system*, the behaviour of the system may not be as intended.

Consider the example of Schneier's gate, [22], wherein a security system is ineffective because of the existence of a side-channel that allows a control to be circumvented. Here a facility that is intended to be secured is protected by a barrier that prevents cars from entering into the facility. The barrier may be controlled by a token — such as a card, a remote, or a code — the holding of which distinguishes authorized personnel from intruders. If, however, the barrier itself is surrounded by ground that can be traversed by a vehicle, without any kind of fence or wall, then any car can drive around it (whether it's with a malicious intent or just by laziness of getting through the security procedure)

and the access control policy, as implemented by the barrier and the tokens, is undermined. So, the access control policy — that only authorized personnel, in possession of a token, may take vehicles into the facility — is undermined by the architecture of the system to which it is applied.

We show how ERL* can be used to model, and so reason about, the situation described above (following [22]), illustrating how such situations can be identified by logical analysis. Related analyses, employing logical models of layered graphs, can be found in [5]. We start with a simple model, depicted in Fig. 1, and gradually refine it. We model just a facility protected by an access barrier. A vehicle having the appropriate access token should be able to get inside. Here we use resources to represent various entities in the model and the atomic formulae characterize properties on those entities. A substantive explanation of systems modelling using locations, resources, processes, and associated substructural modal logics may be found in [1,3]. We consider the following sets of resources, agents, and properties: $Res = \{e, t, b\}$, $A = \{a\}$, $\text{Prop} = \{O, J\}$. O and J respectively express being *outside* and *inside* the facility (we use J instead of I to avoid confusion with I, the unit operator). If a resource $c \in R$ represents a vehicle, $c \models_{\mathcal{W}} O$ means that c is outside the facility, and $c \models_{\mathcal{W}} J$ means it is inside. The agent a is a generic one that represents a user of the system. The resources b and t represent tokens that stand respectively for the barrier and the access token of the users.

From the modelling perspective, the resources we have exposed here are diverse in nature: t is a material token (key or card for instance), c represents a car, while b seems to be just a marker for the presence or well-functioning of the barrier. This diversity raises the question of the meaning and value of the neutral resource e. We elide that problem by accepting that resources encompass a variety of different objects, but we can also employ the epistemic nature of our logic and consider that resources represent not objects as such but rather the knowledge that a given object is in our system. For example, c can be viewed as an abstract token marking the presence of a car, and t the presence of the required access device in this car. Thus resources act as an abstraction layer of our system. In that view, it's easy to see e as the absence of information (we know nothing of our system).

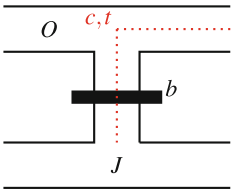


Fig. 1. Barrier problem, base case

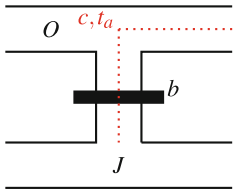


Fig. 2. Barrier problem with agents

We have the following property: $O \rightarrow \mathbf{L}_a^{bt} J$. According to the semantics, based on a resource monoid R, $c \models_{\mathcal{W}} O \rightarrow \mathbf{L}_a^{bt} J$ just in case if $c \models_{\mathcal{W}} O$, then,

for every $c' \in R$ such that $c \bullet b \bullet t \sim_a c'$, $c' \models_{\mathcal{W}} J$. Thus the combination of the two tokens grants access to the inside. The use of the token b for the presence of the barrier helps in modelling a situation in which the barrier is completely shut or is broken (in which case entering wouldn't be possible). Note that the formulae $O \to \mathbf{L}_a^t J$, $O \to \mathbf{L}_a^b J$, and $O \to \mathbf{L}_a^e J$ are not valid because we cannot enter if the barrier is shut, if we have no access token, or both.

The use of the operator $\mathbf{L}$ in this situation is illustrative. First, consider what differences the use of one of the other two operators would make. If we were to state $O \to \mathbf{M}_a^{bt} J$, then it would mean that anyone outside can get (without condition) inside and acquire the two access tokens. This is of course not what we expect. On the other hand, using $\mathbf{N}$ has an interesting effect. $O \to \mathbf{N}_a^{bt} J$ requires not only that an entering agent have the expected tokens, but also that those tokens remain active once they are inside. This is slightly different from our first approach: we don't know if the tokens are still active once the agent is inside.

We can also consider which of the additive implication, $\to$, and the multiplicative, $-\!\!*$, would be the better modelling choice in this example. For a first approach, $\to$ seems quite sufficient. Indeed, if we assert $O \to \mathbf{L}_a^{bt} J$ as valid, then any resource satisfies it. So, if we have a car c such that $c \models_{\mathcal{W}} O$, we also have $c \models_{\mathcal{W}} O \to \mathbf{L}_a^{bt} J$, and then we get the expected $c \models_{\mathcal{W}} \mathbf{L}_a^{bt} J$.

However, if we consider more complex properties, the situation is different. Imagine, for example, an environment that is composed not only of the car c, but also other information o. Our epistemic world is thus $o \bullet c$. So, even if we have $c \models_{\mathcal{W}} O$, we cannot use the property $O \to \mathbf{L}_a^{bt} J$ as we don't have $o \bullet c \models_{\mathcal{W}} O$. On the contrary, if we state the property $O -\!\!* \mathbf{L}_a^{bt} J$ as valid instead, then we have, in particular, $o \models_{\mathcal{W}} O -\!\!* \mathbf{L}_a^{bt} J$ and, together with $c \models_{\mathcal{W}} O$, this gives $o \bullet c \models_{\mathcal{W}} \mathbf{L}_a^{bt} J$, as desired. So, the use of $-\!\!*$ instead of $\to$ is much more useful in more complex systems, as it allows us to set aside, as with Separation Logic's Frame Rule, some of the entities of our system and still apply the property.

Now we introduce agents to the model (see Fig. 2). The first model may seem crude, because a single resource is used to model the access of any agent. So, we seek to benefit from the logic that allows us to take agents into account. We change the model by defining a detailed set of agents, $A = \{\alpha, \beta, \gamma\}$ and now take three users, α, β, and γ. Each user should have its own access token, and the resource set is modified accordingly: $Res = \{e, b, t_\alpha, t_\beta, t_\gamma\}$. Now the slightly different formula $O \to \mathbf{L}_a^{bt_a} J$ is valid for any agent $a \in A$. So, for example, $O \to \mathbf{L}_\alpha^{bt_\alpha} J$ is valid, which means that α can get inside with his own token, but $O \to \mathbf{L}_\alpha^{bt_\beta} J$ is not, which means α cannot use β's token.

Now consider that the access is controlled and the agents are supposed to cross the barrier only if they have the appropriate access device. We want to capture the fact that the system can actually be flawed (as mentioned in the problem presentation). It is actually quite easy to do, because being able to circumvent the barrier just means being able to access inside of the complex without any token. We could be a little more specific by imagining that some agents know the shortcut (or dare to use it) and others don't (See Fig. 3). In

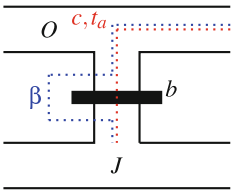


Fig. 3. Barrier problem with a shortcut

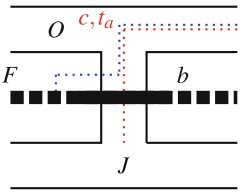


Fig. 4. Barrier problem with a fence

the previous setting, suppose that the agent β is aware of the shortcut and is disposed to use it. Our new set of properties should now be the following:

$$\left\{ O \to \mathbf{L}_a^{bt_a} J \text{ (for every } a \in A),\ O \to \mathbf{L}_\beta^e J \right\}.$$

The unit resource e expresses a direct access (with no resource needed). Note how the use of agents can help us to express different security policies in the same model.

We can reasonably suppose that such a flawed system would be quickly dealt with; for example, by installing a fence that would prevent going around the barrier (See Fig. 4). We could, of course, just model that by removing our last addition and get back to the intended policy, but it is more interesting to encode it by a formula. For example, we might then also describe a fault in the fence (or its removal). To do so, we can simply add a propositional formula F that is valid for any resource provided there is a fence preventing the passage of 'rogue' agents. Our system then becomes

$$\left\{ O \to \mathbf{L}_a^{bt_a} J \text{ (for every } a \in A),\ O \wedge \neg F \to \mathbf{L}_\beta^e J \right\}.$$

Having established a system of formulae that describes our modelling situation quite clearly, we can seek to some properties of the model. The idea is to establish a property of the system that goes beyond its basic definition. For example, we may want to check that every agent inside the facility has passed the barrier and has in its possession its access token. This means that we must prove that, for every agent $a \in A$, $J \to \widetilde{\mathbf{M}}_a^{bt_a} J$.

Indeed, if $c \models_{\mathcal{W}} J \to \widetilde{\mathbf{M}}_a^{bt_a} J$, this means that if $c \models_{\mathcal{W}} J$, then there exists $c' \in R$ such that $c \sim_a c' \bullet b \bullet t_a$ and $c' \bullet b \bullet t_a \models_{\mathcal{W}} J$, which expresses that every resource representing a car that is inside must in fact be equivalent, for a certain agent $a \in A$, to a resource that is inside *and* is composed with both the appropriate token t_a and the barrier token b. This is exactly what we wanted to capture. Notice that this particular property is not verified by the system we stated in the last paragraph. Indeed, as we noticed before, specifying entrance with $r \models_{\mathcal{W}} O \to \mathbf{L}_a^{bt_a} J$ makes J being satisfied by any resource r' such that $r \bullet b \bullet t_a \sim_a r'$. We see that r' does not contain b and t_a. The use of $\mathbf{N}$ instead solves this problem: we then have $r \bullet b \bullet t_a \sim_a r' \bullet b \bullet t_a$ and $r' \bullet b \bullet t_a \models_{\mathcal{W}} J$, as required.

So far, we have consider only simple situations, mainly one car crossing the barrier in various situations. Of course, we may wish to consider more complex models and establish similar properties. For example, we may want to see what happen if several cars are modelled together in the system. We have the sets of properties in the form of implications stated before. To state there is a car in the system, we just assert that the formula O is valid. Then, by looking at the semantics of our formulae, we create a resource c to satisfy that formula. In order to have several cars, we are first tempted to state something like $O \wedge O \wedge O$ (for three cars). However, given our semantics, we have trivially that $O \wedge O \wedge O \equiv O$, which is annoying for our modelling. It is better to state $O * O * O$, using the multiplicative conjunction, instead. Then, to satisfy this formula, we need indeed three resources c_1, c_2, c_3 and we have $c_1 \bullet c_2 \bullet c_3 \models_{\mathcal{W}} O * O * O$. Then, using $-\!\!*$ as described above, we can see the system evolve as cars are allowed inside. Thus, the use of $*$ is particularly relevant to model several instances of a same object.

Although we have shown how ERL is sufficiently expressive to describe a security problem and check some of its behavioural properties, the modelling approach described so far quite limited to capturing specific situations in a more-or-less ad hoc manner. One approach to analysing the relationship between policy and system architecture is to reason in terms of *layers*, as developed in [4,5,10], using logics that are similar to, but weaker than, BI. In this set-up, a policy architecture is layered over a system architecture. Another way to think of this that we design first a general model with very few details, and then to design several others that *refine* one another by inheriting the last model's designs while adding some new and more precise details.

5 A Tableaux Calculus for ERL

We define a labelled tableaux calculus for ERL in the spirit of previous work for BI [11], BBI [16], ESL [7], and ILGL [10]. First, we introduce labels and constraints that correspond, respectively, to resources and to the equality and equivalence relations on resources and agents. We consider a finite set of constants Λ_r such that $|\Lambda_r| = |Res| - 1$. On it we build an infinite countable set of (resource) constants γ_r such that $\Lambda_r \subset \gamma_r$, and then $\gamma_r = \Lambda_r \cup \{c_1, c_2, \ldots\}$. Concatenation of lists is denoted by $\oplus$; $[\![]\!]$ denotes the empty list. A *resource label* is a word built on γ_r, where the order of letters is not taken into account; that is, a finite multiset γ_r and by ε the empty word. For example, xy is the composition of the resource labels x and y. We say that x is a *resource sublabel* of y if and only if there exists z such that $xz = y$. The set of resource sublabels of x is denoted $\mathfrak{L}(x)$.

We define a function $\lambda : Res \mapsto L_r$ such that: 1. $\lambda(e) = \varepsilon$; 2. for all $r \in Res \backslash \{e\}$, $\lambda(r) \in \Lambda_r$; 3. λ is injective (if $\lambda(r) = \lambda(r')$, then $r = r'$). Note that λ is trivially a bijection between Res and $\Lambda_r \cup \{\varepsilon\}$.

Definition 5 (Constraints). *A* resource constraint *is an expression of the form $x \simeq y$, where x and y are resource labels. An* agent constraint *is an*

Rules for resource constraints:

$$\frac{}{\varepsilon \simeq \varepsilon}\ \langle \varepsilon \rangle \qquad \frac{x \simeq y}{y \simeq x}\ \langle s_r \rangle \qquad \frac{xy \simeq xy}{x \simeq x}\ \langle d_r \rangle \qquad \frac{x \simeq y \qquad y \simeq z}{x \simeq z}\ \langle t_r \rangle$$

$$\frac{x \simeq y \qquad yk \simeq yk}{xk \simeq yk}\ \langle c_r \rangle \qquad \frac{x =\!\!=_u y}{x \simeq x}\ \langle k_r \rangle$$

Rules for agent constraints:

$$\frac{x \simeq x}{x =\!\!=_v x}\ \langle r_a \rangle \qquad \frac{x =\!\!=_u y}{y =\!\!=_u x}\ \langle s_a \rangle \qquad \frac{x =\!\!=_u y \qquad y =\!\!=_u z}{x =\!\!=_u z}\ \langle t_a \rangle \qquad \frac{x =\!\!=_u y \qquad x \simeq k}{k =\!\!=_u y}\ \langle k_a \rangle$$

Fig. 5. Rules for constraint closure (for any $u \in A$)

expression of the form $x =\!\!=_u y$, where x and y are resource labels and u belongs to the set of agents A.

A *set of constraints* is any set $\mathcal{C}$ that contains resource constraints and agent constraints. Let $\mathcal{C}$ be a set of constraints. The (resource) *domain* of $\mathcal{C}$ is the set of all resource sublabels that appear in $\mathcal{C}$; that is,

$$\mathcal{D}_r(\mathcal{C}) = \bigcup_{x \simeq y \in \mathcal{C}} (\mathcal{E}(x) \cup \mathcal{E}(y)) \ \cup \bigcup_{x =\!\!=_u y \in \mathcal{C}} (\mathcal{E}(x) \cup \mathcal{E}(y)).$$

Let $\mathcal{C}$ be a set of constraints. The (resource) *alphabet* $\mathcal{A}_r(\mathcal{C})$ of $\mathcal{C}$ is the set of resource constants that appear in $\mathcal{C}$. In particular, $\mathcal{A}_r(\mathcal{C}) = \gamma_r \cap \mathcal{D}_r(\mathcal{C})$. Now we introduce, in Fig. 5, the rules for constraint closure that allow us to capture the properties of the models into the calculus.

Definition 6 (Closure of constraints). *Let $\mathcal{C}$ be a set of constraints. The closure of $\mathcal{C}$, denoted $\overline{\mathcal{C}}$, is the least relation closed under the rules of Fig. 5 such that $\mathcal{C} \subseteq \overline{\mathcal{C}}$.*

There are six rules ($\langle \varepsilon \rangle$, $\langle s_r \rangle$, $\langle d_r \rangle$, $\langle t_r \rangle$, $\langle c_r \rangle$, and $\langle k_r \rangle$) that produce resource constraints and four rules ($\langle r_a \rangle$, $\langle s_a \rangle$, $\langle t_a \rangle$, and $\langle k_a \rangle$) that produce agent constraints. We note that v, introduced in the rule $\langle r_a \rangle$, must belong to the set of agents A.

Proposition 3. *The following rules can be derived from the rules of constraint closure:*

$$\frac{xk \simeq y}{x \simeq x}\ \langle p_l \rangle \qquad \frac{x \simeq yk}{y \simeq y}\ \langle p_r \rangle \qquad \frac{xk =\!\!=_u y}{x \simeq x}\ \langle q_l \rangle$$

$$\frac{x =\!\!=_u yk}{y \simeq y}\ \langle q_r \rangle \qquad \frac{x =\!\!=_u y \qquad x \simeq x' \qquad y \simeq y'}{x' =\!\!=_u y'}\ \langle w_a \rangle \ .$$

Corollary 1. *Let $\mathcal{C}$ be a set of constraints and $u \in A$ be an agent.*

1. *$x \in \mathcal{D}_r(\overline{\mathcal{C}})$ iff $x \simeq x \in \overline{\mathcal{C}}$ iff $x =_u x \in \overline{\mathcal{C}}$.*
2. *If $xy \in \mathcal{D}_r(\overline{\mathcal{C}})$, $x' \simeq x \in \overline{\mathcal{C}}$, and $y' \simeq y \in \overline{\mathcal{C}}$, then $xy \simeq x'y' \in \overline{\mathcal{C}}$.*

Proposition 4. *Let $\mathcal{C}$ be a set of constraints. We have $\mathcal{A}_r(\mathcal{C}) = \mathcal{A}_r(\overline{\mathcal{C}})$.*

Lemma 2 (Compactness). *Let $\mathcal{C}$ be a (possibly infinite) set of constraints.*

1. *If $x \simeq y \in \overline{\mathcal{C}}$, then there is a finite set $\mathcal{C}_f$ such that $\mathcal{C}_f \subseteq \mathcal{C}$ and $x \simeq y \in \overline{\mathcal{C}_f}$.*
2. *If $x =_u y \in \overline{\mathcal{C}}$, then there is a finite set $\mathcal{C}_f$ such that $\mathcal{C}_f \subseteq \mathcal{C}$ and $x =_u y \in \overline{\mathcal{C}_f}$.*

We define a labelled tableaux calculus for ERL in the spirit of previous work for BI [11], BBI [16], ESL [7], and ILGL [10] by using similar definitions and results.

Definition 7. *A* labelled formula *is a 3-tuple of the form $(\mathbb{S}\,\phi : x)$ such that $S \in \{\mathbb{T}, \mathbb{F}\}$, $\phi \in \mathcal{L}$ is a formula and $x \in L_r$ is a resource label. A* constrained set of statements *(CSS) is a pair $\langle \mathcal{F}, \mathcal{C} \rangle$, where $\mathcal{F}$ is a set of labelled formulae and $\mathcal{C}$ is a set of constraints, satisfying the property: if $(\mathbb{S}\,\phi : x) \in \mathcal{F}$, then $x \simeq x \in \overline{\mathcal{C}}$ (call this property P_{css}). A CSS $\langle \mathcal{F}, \mathcal{C} \rangle$ is* finite *if $\mathcal{F}$ and $\mathcal{C}$ are finite. The relation $\preccurlyeq$ is defined by $\langle \mathcal{F}, \mathcal{C} \rangle \preccurlyeq \langle \mathcal{F}', \mathcal{C}' \rangle$ iff $\mathcal{F} \subseteq \mathcal{F}'$ and $\mathcal{C} \subseteq \mathcal{C}'$. We write $\langle \mathcal{F}_f, \mathcal{C}_f \rangle \preccurlyeq_f \langle \mathcal{F}, \mathcal{C} \rangle$ when $\langle \mathcal{F}_f, \mathcal{C}_f \rangle \preccurlyeq \langle \mathcal{F}, \mathcal{C} \rangle$ holds and $\langle \mathcal{F}_f, \mathcal{C}_f \rangle$ is finite, meaning that $\mathcal{F}_f$ and $\mathcal{C}_f$ are both finite.*

Proposition 5. *For any CSS $\langle \mathcal{F}_f, \mathcal{C} \rangle$, where $\mathcal{F}_f$ is finite, there exists $\mathcal{C}_f \subseteq \mathcal{C}$ such that $\mathcal{C}_f$ is finite and $\langle \mathcal{F}_f, \mathcal{C}_f \rangle$ is a CSS.*

Proof. By induction on the number of labelled formulae of $\mathcal{F}_f$ and by Lemma 2.

Figure 6 presents the rules of tableaux calculus for ERL. Note that 'c_i and c_j are new label constants' means $c_i \neq c_j \in \gamma_r \setminus (\mathcal{A}_r(\mathcal{C}) \cup \Lambda_r)$.

Definition 8 (Tableau). *Let $\langle \mathcal{F}_0, \mathcal{C}_0 \rangle$ be a finite CSS. A* tableau *for $\langle \mathcal{F}_0, \mathcal{C}_0 \rangle$ is a list of CSS, called* branches, *inductively built according the following rules:*

1. *The one branch list $[\langle \mathcal{F}_0, \mathcal{C}_0 \rangle]$ is a tableau for $\langle \mathcal{F}_0, \mathcal{C}_0 \rangle$;*
2. *If the list $\mathcal{T}_m \oplus [\langle \mathcal{F}, \mathcal{C} \rangle] \oplus \mathcal{T}_n$ is a tableau for $\langle \mathcal{F}_0, \mathcal{C}_0 \rangle$ and*

$$\frac{cond\langle \mathcal{F}, \mathcal{C} \rangle}{\langle \mathcal{F}_1, \mathcal{C}_1 \rangle \mid \ldots \mid \langle \mathcal{F}_k, \mathcal{C}_k \rangle}$$

is an instance of a rule of Fig. 6 for which $cond\langle \mathcal{F}, \mathcal{C} \rangle$ is fulfilled, then the list $\mathcal{T}_m \oplus [\langle \mathcal{F} \cup \mathcal{F}_1, \mathcal{C} \cup \mathcal{C}_1 \rangle; \ldots; \langle \mathcal{F} \cup \mathcal{F}_k, \mathcal{C} \cup \mathcal{C}_k \rangle] \oplus \mathcal{T}_n$ is a tableau for $\langle \mathcal{F}_0, \mathcal{C}_0 \rangle$.

A tableau *for the formula ϕ is a* tableau *for $\langle \{(\mathbb{F}\,\phi : c_1)\}, \{c_1 \simeq c_1\} \rangle$.*

$$\frac{(\mathbb{T}\mathbf{I} : x) \in \mathcal{F}}{\langle \emptyset, \{x \simeq \varepsilon\} \rangle} \ \langle \mathbb{T}\mathbf{I} \rangle$$

$$\frac{(\mathbb{T}\neg\phi : x) \in \mathcal{F}}{\langle \{(\mathbb{F}\phi : x)\}, \emptyset \rangle} \ \langle \mathbb{T}\neg \rangle \qquad \frac{(\mathbb{F}\neg\phi : x) \in \mathcal{F}}{\langle \{(\mathbb{T}\phi : x)\}, \emptyset \rangle} \ \langle \mathbb{F}\neg \rangle$$

$$\frac{(\mathbb{T}\phi \wedge \psi : x) \in \mathcal{F}}{\langle \{(\mathbb{T}\phi : x), (\mathbb{T}\psi : x)\}, \emptyset \rangle} \ \langle \mathbb{T}\wedge \rangle \qquad \frac{(\mathbb{F}\phi \wedge \psi : x) \in \mathcal{F}}{\langle \{(\mathbb{F}\phi : x)\}, \emptyset \rangle \mid \langle \{(\mathbb{F}\psi : x)\}, \emptyset \rangle} \ \langle \mathbb{F}\wedge \rangle$$

$$\frac{(\mathbb{T}\phi \vee \psi : x) \in \mathcal{F}}{\langle \{(\mathbb{T}\phi : x)\}, \emptyset \rangle \mid \langle \{(\mathbb{T}\psi : x)\}, \emptyset \rangle} \ \langle \mathbb{T}\vee \rangle \qquad \frac{(\mathbb{F}\phi \vee \psi : x) \in \mathcal{F}}{\langle \{(\mathbb{F}\phi : x), (\mathbb{F}\psi : x)\}, \emptyset \rangle} \ \langle \mathbb{F}\vee \rangle$$

$$\frac{(\mathbb{T}\phi \rightarrow \psi : x) \in \mathcal{F}}{\langle \{(\mathbb{F}\phi : x)\}, \emptyset \rangle \mid \langle \{(\mathbb{T}\psi : x)\}, \emptyset \rangle} \ \langle \mathbb{T}\rightarrow \rangle \qquad \frac{(\mathbb{F}\phi \rightarrow \psi : x) \in \mathcal{F}}{\langle \{(\mathbb{T}\phi : x), (\mathbb{F}\psi : x)\}, \emptyset \rangle} \ \langle \mathbb{F}\rightarrow \rangle$$

$$\frac{(\mathbb{T}\phi * \psi : x) \in \mathcal{F}}{\langle \{(\mathbb{T}\phi : c_i), (\mathbb{T}\psi : c_j)\}, \{x \simeq c_i c_j\} \rangle} \ \langle \mathbb{T}* \rangle \qquad \frac{(\mathbb{F}\phi * \psi : x) \in \mathcal{F} \text{ and } x \simeq yz \in \overline{\mathcal{C}}}{\langle \{(\mathbb{F}\phi : y)\}, \emptyset \rangle \mid \langle \{(\mathbb{F}\psi : z)\}, \emptyset \rangle} \ \langle \mathbb{F}* \rangle$$

$$\frac{(\mathbb{T}\phi \mathrel{-\!\!*} \psi : x) \in \mathcal{F} \text{ and } xy \simeq xy \in \overline{\mathcal{C}}}{\langle \{(\mathbb{F}\phi : y)\}, \emptyset \rangle \mid \langle \{(\mathbb{T}\psi : xy)\}, \emptyset \rangle} \ \langle \mathbb{T}\mathrel{-\!\!*} \rangle \qquad \frac{(\mathbb{F}\phi \mathrel{-\!\!*} \psi : x) \in \mathcal{F}}{\langle \{(\mathbb{T}\phi : c_i), (\mathbb{F}\psi : xc_i)\}, \{xc_i \simeq xc_i\} \rangle} \ \langle \mathbb{F}\mathrel{-\!\!*} \rangle$$

$$\frac{(\mathbb{T}\mathbf{L}^r_u\phi : x) \in \mathcal{F} \text{ and } x\lambda(r) =_u y \in \overline{\mathcal{C}}}{\langle \{(\mathbb{T}\phi : y)\}, \emptyset \rangle} \ \langle \mathbb{T}\mathbf{L} \rangle \qquad \frac{(\mathbb{F}\mathbf{L}^r_u\phi : x) \in \mathcal{F}}{\langle \{(\mathbb{F}\phi : c_i)\}, \{x\lambda(r) =_u c_i\} \rangle} \ \langle \mathbb{F}\mathbf{L} \rangle$$

$$\frac{(\mathbb{T}\mathbf{M}^r_u\phi : x) \in \mathcal{F} \text{ and } x =_u y\lambda(r) \in \overline{\mathcal{C}}}{\langle \{(\mathbb{T}\phi : y\lambda(r))\}, \emptyset \rangle} \ \langle \mathbb{T}\mathbf{M} \rangle \qquad \frac{(\mathbb{F}\mathbf{M}^r_u\phi : x) \in \mathcal{F}}{\langle \{(\mathbb{F}\phi : c_i\lambda(r))\}, \{x =_u c_i\lambda(r)\} \rangle} \ \langle \mathbb{F}\mathbf{M} \rangle$$

$$\frac{(\mathbb{T}\mathbf{N}^r_u\phi : x) \in \mathcal{F} \text{ and } x\lambda(r) =_u y\lambda(r) \in \overline{\mathcal{C}}}{\langle \{(\mathbb{T}\phi : y\lambda(r))\}, \emptyset \rangle} \ \langle \mathbb{T}\mathbf{N} \rangle \qquad \frac{(\mathbb{F}\mathbf{N}^r_u\phi : x) \in \mathcal{F}}{\langle \{(\mathbb{F}\phi : c_i\lambda(r))\}, \{x\lambda(r) =_u c_i\lambda(r)\} \rangle} \ \langle \mathbb{F}\mathbf{N} \rangle$$

$$\frac{(\mathbb{T}\widetilde{\mathbf{L}}^r_u\phi : x) \in \mathcal{F}}{\langle \{(\mathbb{T}\phi : c_i)\}, \{x\lambda(r) =_u c_i\} \rangle} \ \langle \mathbb{T}\widetilde{\mathbf{L}} \rangle \qquad \frac{(\mathbb{F}\widetilde{\mathbf{L}}^r_u\phi : x) \in \mathcal{F} \text{ and } x\lambda(r) =_u y \in \overline{\mathcal{C}}}{\langle \{(\mathbb{F}\phi : y)\}, \emptyset \rangle} \ \langle \mathbb{F}\widetilde{\mathbf{L}} \rangle$$

$$\frac{(\mathbb{T}\widetilde{\mathbf{M}}^r_u\phi : x) \in \mathcal{F}}{\langle \{(\mathbb{T}\phi : c_i\lambda(r))\}, \{x =_u c_i\lambda(r)\} \rangle} \ \langle \mathbb{T}\widetilde{\mathbf{M}} \rangle \qquad \frac{(\mathbb{F}\widetilde{\mathbf{M}}^r_u\phi : x) \in \mathcal{F} \text{ and } x =_u y\lambda(r) \in \overline{\mathcal{C}}}{\langle \{(\mathbb{F}\phi : y\lambda(r))\}, \emptyset \rangle} \ \langle \mathbb{F}\widetilde{\mathbf{M}} \rangle$$

$$\frac{(\mathbb{T}\widetilde{\mathbf{N}}^r_u\phi : x) \in \mathcal{F}}{\langle \{(\mathbb{T}\phi : c_i\lambda(r))\}, \{x\lambda(r) =_u c_i\lambda(r)\} \rangle} \ \langle \mathbb{T}\widetilde{\mathbf{N}} \rangle \qquad \frac{(\mathbb{F}\widetilde{\mathbf{N}}^r_u\phi : x) \in \mathcal{F} \text{ and } x\lambda(r) =_u y\lambda(r) \in \overline{\mathcal{C}}}{\langle \{(\mathbb{F}\phi : y\lambda(r))\}, \emptyset \rangle} \ \langle \mathbb{F}\widetilde{\mathbf{N}} \rangle$$

Note: c_i and c_j are new label constants, with $c_i, c_j \notin \Lambda_r$

Fig. 6. Rules of the tableaux calculus for ERL

We remark that a tableau for a formula ϕ verifies the property (P_{css}) of Definition 7 (by the rule $\langle r_a \rangle$) and any application of a rule of Fig. 6 provide also a tableau that verifies the property (P_{css}) (in particular by Corollary 1).

In this calculus, we have two particular set of rules. The first set is composed by the rules $\langle \mathbb{T}\mathrm{I} \rangle$, $\langle \mathbb{T}* \rangle$, $\langle \mathbb{F}{-\!\!*} \rangle$, $\langle \mathbb{F}\mathbf{L} \rangle$, $\langle \mathbb{F}\mathbf{M} \rangle$, $\langle \mathbb{F}\mathbf{N} \rangle$, $\langle \mathbb{T}\widetilde{\mathbf{L}} \rangle$, $\langle \mathbb{T}\widetilde{\mathbf{M}} \rangle$, and $\langle \mathbb{T}\widetilde{\mathbf{N}} \rangle$, that introduce new label constants (c_i and c_j) and new constraints, except for $\langle \mathbb{T}\mathrm{I} \rangle$ that only introduces a new constraint. The second set is composed of the rules $\langle \mathbb{F}* \rangle$, $\langle \mathbb{T}{-\!\!*} \rangle$, $\langle \mathbb{T}\mathbf{L} \rangle$, $\langle \mathbb{T}\mathbf{M} \rangle$, $\langle \mathbb{T}\mathbf{N} \rangle$, $\langle \mathbb{F}\widetilde{\mathbf{L}} \rangle$, $\langle \mathbb{F}\widetilde{\mathbf{M}} \rangle$, and$\langle \mathbb{F}\widetilde{\mathbf{N}} \rangle$, that have a condition on the closure of constraints. To apply one of these rules we choose a label which satisfies the condition and then apply the corresponding rule. Otherwise, we cannot apply the rule.

Definition 9 (Closure condition). *A CSS $\langle \mathcal{F}, \mathcal{C} \rangle$ is* closed *if one of the following conditions holds, where $\phi \in \mathcal{L}$: 1. $(\mathbb{T}\,\phi : x) \in \mathcal{F}$, $(\mathbb{F}\,\phi : y) \in \mathcal{F}$ and $x \simeq y \in \overline{\mathcal{C}}$; 2. $(\mathbb{F}\mathrm{I} : x) \in \mathcal{F}$ and $x \simeq \varepsilon \in \overline{\mathcal{C}}$; 3. $(\mathbb{F}\top : x) \in \mathcal{F}$; and 4. $(\mathbb{T}\bot : x) \in \mathcal{F}$. A CSS is* open *if it is not closed. A tableau for ϕ is* closed *if all its branches are closed and a* tableaux proof *for ϕ is a closed tableau for ϕ.*

To illustrate the construction of tableaux, we consider $\mathbf{M}_a^s\,\phi \to \mathbf{M}_a^r(\mathbf{M}_a^s\,\phi)$. To build the corresponding tableau, we start with the CCS $\langle \{(\mathbb{F}\mathbf{M}_a^s\,\phi \to \mathbf{M}_a^r(\mathbf{M}_a^s\,\phi) : c_1)\}, \{c_1 \simeq c_1\} \rangle$ and with the following representation of the formula set $\mathcal{F}$ and the constraints set $\mathcal{C}$:

$$\begin{array}{cc} [\mathcal{F}] & [\mathcal{C}] \\ \surd_1(\mathbb{F}\mathbf{M}_a^s\,\phi \to \mathbf{M}_a^r(\mathbf{M}_a^s\,\phi) : c_1) & c_1 \simeq c_1. \end{array}$$

We then apply the rules of our tableaux method, respecting the priority order, and we obtain the tableau of Fig. 7. We omit the λ and write r for $\lambda(r)$, for any resource.

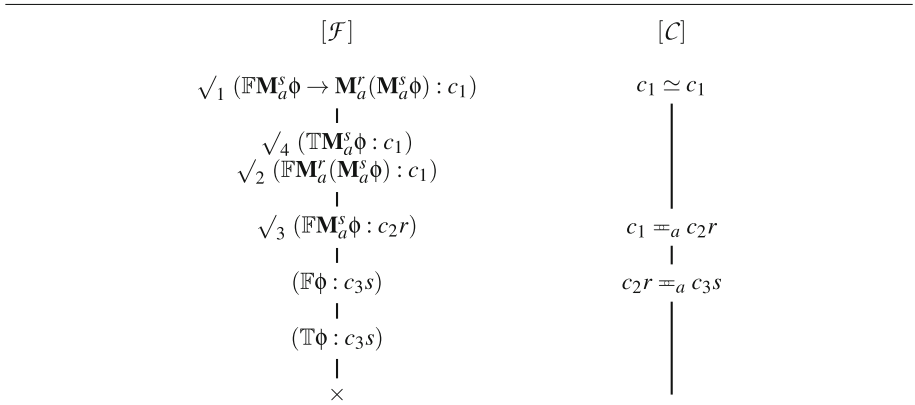


Fig. 7. Tableau for $\mathbf{M}_a^s\,\phi \to \mathbf{M}_a^r(\mathbf{M}_a^s\,\phi)$

Note that we mark with $\surd$ the steps of the tableau construction. The main steps are the following: first apply the rule $\langle \mathbb{F} \rightarrow \rangle$ and then obtain two formulae both with $\mathbf{M}$ as operator. According to the priority rules, first apply the $\langle \mathbb{F}\mathbf{M} \rangle$ rule, which generates a new formula, a new resource label c_2, and the constraint $c_1 \eqsim_a c_2 r$. Then apply the $\langle \mathbb{F}\mathbf{M} \rangle$ rule again, which generates a new formula, a new resource label c_3, and the constraint $c_2 r \eqsim_a c_3 s$. We must now apply the $\langle \mathbb{T}\mathbf{M} \rangle$ rule and then we need a resource label z such that $c_1 \eqsim_a zs \in \overline{\mathcal{C}}$. Now, having closure by rule $\langle t_a \rangle$ with agent a, we generate the constraint $c_1 \eqsim_a c_3 s$, and thus apply the rule with $z = c_1$ and generate $(\mathbb{T}\,\phi : c_3 s)$. As we also have $(\mathbb{F}\,\phi : c_3 s)$, we have a closed branch and thus a closed tableau.

Theorem 1 (Soundness). *Let ϕ be a formula of ERL. If there exists a tableaux proof for ϕ, then ϕ is valid.*

Proof. The proof is similar to the soundness proof of BI tableaux [11] and its recent extensions [6,7,10]. The main point is the notion of *realizability* of a CSS $\langle \mathcal{F}, \mathcal{C} \rangle$, meaning that there exists a model $\mathcal{M}$ and an embedding $(|.|)$ from the resource labels to the resource set of $\mathcal{M}$ such that if $(\mathbb{T}\,\phi : x) \in \mathcal{F}$, then $|x| \vDash_{\mathcal{M}} \phi$,and if $(\mathbb{F}\,\phi : x) \in \mathcal{F}$, then $|x| \nvDash_{\mathcal{M}} \phi$. More details are given in [12].

We propose a countermodel extraction method, adapted from [16], that transforms the sets of resource and agent constraints of a branch $\langle \mathcal{F}, \mathcal{C} \rangle$ into a model $\mathcal{M}$ such that if $(\mathbb{T}\,\phi : x) \in \mathcal{F}$, then $\rho_x \vDash_{\mathcal{M}} \phi$, and if $(\mathbb{F}\,\phi : x) \in \mathcal{F}$, then $\rho_x \nvDash_{\mathcal{M}} \phi$, where ρ_x is the representative of the equivalence class of x.

More details are given in [12] and examples of countermodels with a similar method are given in [6–8,10,11].

Theorem 2 (Completeness). *Let ϕ be an ERL formula. If ϕ is valid, then there exists a tableaux proof for ϕ.*

Proof. The proof consists in building, using a fair strategy, a Hintikka CSS from a formula for which there is no tableaux proof that is a sequence of labelled formulae in which all labelled formulae occur infinitely many times, and an oracle that is a set of non-closed CSS with some specific properties. Then, assuming there is no tableaux proof for ϕ, we build a Hintikka CSS, and deduce from it that ϕ is not valid. More details are given in [12].

6 Conclusions

We have presented a substructural epistemic logic, based on Boolean BI, in which the epistemic modalities, which extend the usual epistemic modalities, are parametrized on the agent's local resource. The logic represents a first step in developing an epistemic resource semantics. This step is illustrated through an example that explores the gap between policy and implementation in access control. We have provided a system of labelled tableaux for the logic, and established soundness and completeness.

Much further work is suggested. First, the theory, pragmatics, and interpretation of the epistemic modalities with resource semantics, including aspects of local reasoning for resource-carrying agents [15,21], concurrency [18]. Second, logical theory, including proof systems, model-theoretic properties, and complexity. Connections with other approaches to modelling the relationship between policy and implementation in system management, such as those discussed in [23] and approaches involving logics for layered graphs [1,4], should be explored.

References

1. Anderson, G., Pym, D.: A calculus and logic of bunched resources and processes. Theor. Comput. Sci. **614**, 63–96 (2016)
2. Baltag, A., Coecke, B., Sadrzadeh, M.: Epistemic actions as resources. J. Logic Comput. **17**(3), 555–585 (2006)
3. Collinson, M., Monahan, B., Pym, D.: A Discipline of Mathematical Systems Modelling. College Publications (2012)
4. Collinson, M., McDonald, K., Pym, D.: Layered graph logic as an assertion language for access control policy models. J. Logic Comput. (2015). doi:10.1093/logcom/exv020
5. Collinson, M., McDonald, K., Pym, D.: A substructural logic for layered graphs. J. Logic Comput. **24**(4), 953–988 (2014)
6. Courtault, J.-R., Galmiche, D.: A modal separation logic for resource dynamics. J. Logic Comput., 46 pages (2015). doi:10.1093/logcom/exv031
7. Courtault, J.-R., Ditmarsch, H., Galmiche, D.: An epistemic separation logic. In: Paiva, V., Queiroz, R., Moss, L.S., Leivant, D., Oliveira, A.G. (eds.) WoLLIC 2015. LNCS, vol. 9160, pp. 156–173. Springer, Heidelberg (2015). doi:10.1007/978-3-662-47709-0_12
8. Courtault, J.-R., Galmiche, D., Pym, D.: A logic of separating modalities. Theor. Comput. Sci. **637**, 30–58 (2016). doi:10.1016/j.tcs.2016.04.040
9. van Ditmarsch, H., Halpern, J.Y., van der Hoek, W., Kooi, B. (eds.): Handbook of Epistemic Logic. College Publications (2015)
10. Docherty, S., Pym, D.: Intuitionistic layered graph logic. In: Olivetti, N., Tiwari, A. (eds.) IJCAR 2016. LNCS (LNAI), vol. 9706, pp. 469–486. Springer, Heidelberg (2016). doi:10.1007/978-3-319-40229-1_32
11. Galmiche, D., Méry, D., Pym, D.: The semantics of BI and resource tableaux. Math. Struct. Comp. Sci. **15**(6), 1033–1088 (2005)
12. Galmiche, D., Kimmel, P., Pym, D.: A substructural epistemic resource logic (extended version). UCL research note RN/16/08 (2016). http://www.cs.ucl.ac.uk/fileadmin/UCL-CS/research/Research_Notes/RN_16_08.pdf
13. Halpern, J.Y., Pucella, R.: Modeling adversaries in a logic for security protocol analysis. In: Abdallah, A.E., Ryan, P., Schneider, S. (eds.) FASec 2002. LNCS, vol. 2629, pp. 115–132. Springer, Heidelberg (2003). doi:10.1007/978-3-540-40981-6_11
14. O'Hearn, P., Pym, D.: The logic of bunched implications. Bull. Symbolic Logic **5**(2), 215–244 (1999)
15. Ishtiaq, S., O'Hearn, P.: BI as an assertion language for mutable data structures. In: 28th ACM Symposium on Principles of Programming Languages (POPL), London, pp. 14–26 (2001)
16. Larchey-Wendling, D.: The formal strong completeness of partial monoidal Boolean BI. J. Logic Comput. **26**(2), 605–640 (2014). doi:10.1093/logcom/exu031

17. Naumov, P., Tao, J.: Budget-constrained knowledge in multiagent systems. Proc. AAMAS **219–226**, 2015 (2015)
18. O'Hearn, P.W.: Resources, concurrency and local reasoning. Theor. Comput. Sci. **375**(1–3), 271–307 (2007)
19. Pucella, R.: Knowledge and security. Chap. 12 of [9], pp. 591–655
20. Pym, D., O'Hearn, P., Yang, H.: Possible worlds, resources: the semantics of BI. Theor. Comput. Sci. **315**(1), 257–305. Erratum: p. 22, l. 22 (preprint), p. 285, 1.-12 (TCS): ', for some P', $Q \equiv P; P'$' should be '$P \vdash Q$'
21. Reynolds, J.: Separation logic: a logic for shared mutable data structures. IEEE Symposium on Logic in Computer Science, LICS 2002, pp. 55–74, Denmark, Copenhagen (July 2002)
22. Schneier, B.: The weakest link (2005). https://www.schneier.com/blog/archives/2005/02/the_weakest_lin.html. Schneier on security, https://www.schneier.com
23. Toninho, B., Caires, L.: A spatial-epistemic logic for reasoning about security protocols. In: 8th International Workshop on Security Issues in Concurrency, SecCo 2010 (2010)

Deriving Natural Deduction Rules from Truth Tables

Herman Geuvers[1,2(✉)] and Tonny Hurkens[1,2]

[1] Radboud University, Nijmegen, The Netherlands
herman@cs.ru.nl
[2] Technical University Eindhoven, Eindhoven, The Netherlands

Abstract. We develop a general method for deriving natural deduction rules from the truth table for a connective. The method applies to both constructive and classical logic. This implies we can derive "constructively valid" rules for any classical connective. We show this constructive validity by giving a general Kripke semantics, that is shown to be sound and complete for the constructive rules. For the well-known connectives ($\vee$, $\wedge$, $\rightarrow$, $\neg$) the constructive rules we derive are equivalent to the natural deduction rules we know from Gentzen and Prawitz. However, they have a different shape, because we want all our rules to have a standard "format", to make it easier to define the notions of cut and to study proof reductions. In style they are close to the "general elimination rules" studied by Von Plato [13] and others. The rules also shed some new light on the classical connectives: e.g. the classical rules we derive for $\rightarrow$ allow to prove Peirce's law. Our method also allows to derive rules for connectives that are usually not treated in natural deduction textbooks, like the "if-then-else", whose truth table is clear but whose constructive deduction rules are not. We prove that "if-then-else", in combination with $\bot$ and $\top$, is functionally complete (all other constructive connectives can be defined from it). We define the notion of cut, generally for any constructive connective and we describe the process of "cut-elimination".

1 Introduction

Natural deduction rules come in various forms, where one either uses formulas A, or sequents $\Gamma \vdash A$ (where Γ is a sequence or a finite set of formulas). Other formalisms use a linear format, using flags or boxes to explicitly manage the open and discharged assumptions.

We use a tree format with sequents, where all rules have a special form:

$$\frac{\Gamma \vdash A_1 \quad \ldots \quad \Gamma \vdash A_n \qquad \Gamma, B_1 \vdash D \quad \ldots \quad \Gamma, B_m \vdash D}{\Gamma \vdash D}$$

So if the conclusion of a rule is $\Gamma \vdash D$, then the hypotheses of the rule can be of one of two forms:

1. $\Gamma \vdash A$: instead of proving D from Γ, we now need to prove A from Γ. We call A a Lemma.

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 123–138, 2017.
DOI: 10.1007/978-3-662-54069-5_10

2. $\Gamma, B \vdash D$: we still need to prove D from Γ, but we are now also allowed to use B as additional assumption. We call B a Casus.

One obvious advantage is that we don't have to give the Γ explicitly, as it can be retrieved from the other information in a deduction. So, we will present the deduction rules without the Γ in the format

$$\frac{\vdash A_1 \quad \ldots \quad \vdash A_n \qquad B_1 \vdash D \quad \ldots \quad B_m \vdash D}{\vdash D}$$

For every connective we have elimination rules and introduction rules. The elimination rules have the following form, where φ is the formula that is eliminated and A_i, B_j are direct subformulas of φ.

$$\frac{\vdash \varphi \quad \vdash A_1 \quad \ldots \quad \vdash A_n \qquad B_1 \vdash D \quad \ldots \quad B_m \vdash D}{\vdash D}\ \mathrm{el}$$

The introduction rules have a classical and an intuitionistic form; the following form is the classical one. (φ is the formula that is "introduced" and A_i, B_j are direct subformulas of φ.) The classical duality between elimination and introduction is clearly visible from these rules.

$$\frac{\varphi \vdash D \quad \vdash A_1 \quad \ldots \quad \vdash A_n \qquad B_1 \vdash D \quad \ldots \quad B_m \vdash D}{\vdash D}\ \mathrm{in}^c$$

The intuitionistic introduction rules have the following form

$$\frac{\vdash A_1 \quad \ldots \quad \vdash A_n \qquad B_1 \vdash \varphi \quad \ldots \quad B_m \vdash \varphi}{\vdash \varphi}\ \mathrm{in}^i$$

We see that, compared to the classical rule, the D has been replaced by φ, the formula we introduce, and we have omitted the first premise, which is $\varphi \vdash \varphi$, because it is trivial. For each connective, we extract the introduction and elimination rules from a truth table as described in the following Definition.

Definition 1. *Suppose we have an n-ary connective c with a truth table t_c (with 2^n rows). We write $\varphi = c(A_1, \ldots, A_n)$ for a formula with c as main connective and $A_1, \ldots, A_n$ as immediate subformulas. Each row of t_c gives rise to an elimination rule or an introduction rule for c in the following way.*

$$\frac{\begin{array}{c|c} A_1 \ \ldots\ A_n & \varphi \\ \hline p_1\ \ldots\ p_n & 0 \end{array}}{} \mapsto \frac{\vdash \varphi \quad \ldots \vdash A_j \ (\textit{if } p_j = 1) \ldots \qquad \ldots A_i \vdash D \ (\textit{if } p_i = 0) \ldots}{\vdash D}\ el$$

$$\frac{\begin{array}{c|c} A_1 \ \ldots\ A_n & \varphi \\ \hline q_1\ \ldots\ q_n & 1 \end{array}}{} \mapsto \frac{\ldots \vdash A_j \ (\textit{if } q_j = 1) \ldots \qquad \ldots A_i \vdash \varphi \ (\textit{if } q_i = 0) \ldots}{\vdash \varphi}\ in^i$$

$$\frac{\begin{array}{c|c} A_1 \ \ldots\ A_n & \varphi \\ \hline r_1\ \ldots\ r_n & 1 \end{array}}{} \mapsto \frac{\varphi \vdash D \quad \ldots \vdash A_j \ (\textit{if } r_j = 1) \ldots \qquad \ldots A_i \vdash D \ (\textit{if } r_i = 0) \ldots}{\vdash D}\ in^c$$

If $p_j = 1$ in t_c, then A_j occurs as a Lemma *in the rule; if $p_i = 0$ in t_c, then A_i occurs as a* Casus. *The rules are given in abbreviated form and it should be*

understood that all judgments can be used with an extended hypotheses set Γ. So the elimination rule in full reads as follows (where Γ is a set of formulas).

$$\frac{\Gamma \vdash \varphi \quad \dots \Gamma \vdash A_j \ (\text{if } p_j = 1) \dots \quad \dots \Gamma, A_i \vdash D \ (\text{if } p_i = 0) \dots}{\Gamma \vdash D}\ el$$

Definition 2. *Given a set of connectives $\mathcal{C} := \{c_1, \dots, c_n\}$, we define the* intuitionistic *and* classical *natural deduction systems for $\mathcal{C}$, $\mathsf{IPC}_\mathcal{C}$ and $\mathsf{CPC}_\mathcal{C}$ as follows.*

- *Both $\mathsf{IPC}_\mathcal{C}$ and $\mathsf{CPC}_\mathcal{C}$ have an* axiom rule

$$\frac{}{\Gamma \vdash A}\ axiom(\ \text{if } A \in \Gamma)$$

- *$\mathsf{IPC}_\mathcal{C}$ has the elimination rules for the connectives in $\mathcal{C}$ and the intuitionistic introduction rules for the connectives in $\mathcal{C}$, as defined in Definition 1.*
- *$\mathsf{CPC}_\mathcal{C}$ has the elimination rules for the connectives in $\mathcal{C}$ and the classical introduction rules for the connectives in $\mathcal{C}$, as defined in Definition 1.*

Example 3. *From the truth table we derive the following intuitionistic rules for $\wedge$, 3 elimination rules and one introduction rule:*

$$\frac{\vdash A \wedge B \quad A \vdash D \quad B \vdash D}{\vdash D}\ \wedge\text{-}el_a \qquad \frac{\vdash A \wedge B \quad A \vdash D \quad \vdash B}{\vdash D}\ \wedge\text{-}el_b$$

$$\frac{\vdash A \wedge B \quad \vdash A \quad B \vdash D}{\vdash D}\ \wedge\text{-}el_c \qquad \frac{\vdash A \quad \vdash B}{\vdash A \wedge B}\ \wedge\text{-}in$$

These rules are all intuitionistically correct, as one can observe by inspection. We will show that these are equivalent to the well-known intuitionistic rules. We will also show how these rules can be optimized and be reduced to 2 elimination rules and 1 introduction rule.

From the truth table we also derive the following rules for $\neg$, 1 elimination rule and 1 introduction rule, a classical and an intuitionistic one.

$$\frac{\vdash \neg A \quad \vdash A}{\vdash D}\ \neg\text{-}el \qquad \frac{A \vdash \neg A}{\vdash \neg A}\ \neg\text{-}in^i \qquad \frac{\neg A \vdash D \quad A \vdash D}{\vdash D}\ \neg\text{-}in^c$$

As an example of the classical derivation rules we show that $\neg\neg A \vdash A$ is derivable:

$$\dfrac{\dfrac{\dfrac{\neg\neg A, \neg A \vdash \neg\neg A \quad \neg\neg A, \neg A \vdash \neg A}{\neg\neg A, \neg A \vdash A}\ \neg\text{-}el \qquad \neg\neg A, A \vdash A}{\neg\neg A \vdash A}\ \neg\text{-}in^c$$

It can be proven that $\neg\neg A \vdash A$ is not derivable with the intuitionistic rules. As an example of the intuitionistic derivation rules we show that $A \vdash \neg\neg A$ is derivable:

$$\dfrac{\dfrac{A, \neg A \vdash \neg A \quad A, \neg A \vdash A}{A, \neg A \vdash \neg\neg A}\ \neg\text{-}el}{A \vdash \neg\neg A}\ \neg\text{-}in^i$$

In the intuitionistic case, there is an obvious notion of *cut*: an intro of φ immediately followed by an elimination of φ. In such case there is at least one k for which $p_k \neq q_k$. In case $p_k = 0, q_k = 1$, we have a sub-derivation Σ of $\vdash A_k$ and a sub-derivation Θ of $A_k \vdash D$ and we can "plug" Σ on top of Θ to obtain a derivation of $\vdash D$. In case $p_k = 1, q_k = 0$, we have a sub-derivation Σ of $A_k \vdash \varphi$ and a sub-derivation Θ of $\vdash A_k$ and we can "plug" Θ on top of Σ to obtain a derivation of $\vdash \varphi$. This is then used as a hypothesis for the elimination rule (that remains in this case) instead of the original one that was a consequence of the introduction rule (that now disappears). Note that in general there are more such k, so the cut-elimination procedure is non-deterministic. We view this non-determinism as a natural feature in natural deduction; the fact that for some connectives (or combination of connectives), cut-elimination is deterministic is an "emerging" property.

1.1 Contribution of the Paper and Related Work

The main contributions of the paper are:

- A general construction of natural deduction rules for a logical connective from its truth table semantics, yielding natural deduction rules in a fixed structured format.
- The method applies to both a classical and a constructive (!) reading of the connectives, and applies to connectives of any arity.
- Soundness and completeness of the constructive connectives with respect to a general Kripke semantics that we define.
- Example of the if-then-else connective, which is shown to be constructively functionally complete, once the constants $\top$ and $\bot$ have been added.
- A general definition of "direct cut" and "elimination of a direct cut" for the generalized constructive connectives.

Natural deduction has been studied extensively, since the original work by Gentzen, both for classical and intuitionistic logic. Overviews can be found in [7, 12]. Also the generalization of natural deduction to include other connectives or allow different derivation rules has been studied by various researchers. Notably, there is the work of Schroeder-Heister [10], Von Plato [13], Milne [6] and Francez and Dyckhoff [3,4] is related to ours. Schroeder-Heister studies general formats of natural deduction where also rules may be discharged (as opposed to the normal situation where only formulas may be discharged). He also studies a general rule format for intuitionistic logic and shows that the connectives $\wedge, \vee, \rightarrow, \bot$ are complete for it. Von Plato, Milne, Francez and Dyckhoff study "generalized elimination rules", where the idea is that elimination rules arise naturally from the introduction rules, following Prawitz's [9] inversion principle: "the conclusion obtained by an elimination does not state anything more than what must have already been obtained if the major premiss of the elimination was inferred by an introduction".

A difference is that we focus not so much on the rules but on the fact that we can define different and new connectives constructively. In our work, we do

not take the introduction rules as primary, with the elimination rules defined from them, but we derive elimination and introduction rules directly from the truth table. Then we optimize them, which can be done in various ways, where we adhere to a fixed format for the rules. Many of the generalized elimination rules, for example for $\wedge$, appear naturally as a consequence of our approach of deriving the rules from the truth table.

The idea of deriving deduction rules from the truth table also occurs in the work of Milne [6], for the classical case: from the introduction rules, a truth table is derived and then the elimination rules are derived from the truth table. For the if-then-else connective, this amounts to classical rules equivalent to ours (see Sect. 2.1), but less optimized. We start from the truth table and also derive rules for constructive logic.

In Sect. 3 we give a complete Kripke semantics for the constructive connectives. This is reminiscent of the *Lindenbaum construction* used in [6] to prove classical completeness. The Kripke semantics also allows us to prove some meta properties about the rules. For example, we give a generalization of the *disjunction property* in intuitionistic logic. In Sect. 4 we define cuts precisely, for the intuitionistic case.

2 Simple Properties and Examples

We first define precisely how the "plugging one derivation in another" works.

Lemma 4. *If $\Gamma \vdash \varphi$ and $\Delta, \varphi \vdash \psi$, then $\Gamma, \Delta \vdash \psi$*

Proof. By a simple induction on the derivation of $\Delta, \varphi \vdash \psi$, using the fact that, in general (for all Γ, Γ' and φ): If $\Gamma \vdash \varphi$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash \varphi$. □

We can be a bit more precise about what is happening in the proof of Lemma 4. If Π is the derivation of $\Delta, \varphi \vdash \psi$, due to the format of our rules, the only place in Π where the hypothesis φ can be used is at a leaf of Π, in an instance of the (axiom) rule. These leaves are of the shape $\Delta', \varphi \vdash \varphi$ for some $\Delta' \supseteq \Delta$.

If Σ is the derivation of $\Gamma \vdash \varphi$, then Σ is also a derivation of $\Delta', \Gamma \vdash \varphi$ (for any Δ). So, we can replace each leaf of Π that is an instance of an axiom $\Delta', \varphi \vdash \varphi$ by a derivation Σ of $\Delta', \Gamma \vdash \varphi$, to obtain a derivation of $\Gamma, \Delta \vdash \psi$. We introduce some notation to support this.

Notation 5. *If Σ is a derivation of $\Gamma \vdash \varphi$ and Π is a derivation of $\Delta, \varphi \vdash \psi$, then we have a derivation of $\Gamma, \Delta \vdash \psi$ that looks like this:*

$$\begin{array}{c} \vdots\,\Sigma \\ \Gamma, \Delta_1 \vdash \varphi \end{array} \quad \ldots \quad \begin{array}{c} \vdots\,\Sigma \\ \Gamma, \Delta_n \vdash \varphi \end{array}$$
$$\vdots\,\Pi$$
$$\Gamma, \Delta \vdash \psi$$

So in Π, every application of an (axiom) rule at a leaf, deriving $\Delta' \vdash \varphi$ for some $\Delta' \supseteq \Delta$ is replaced by a copy of a derivation Σ, which is also a derivation of $\Delta', \Gamma \vdash \varphi$.

In Definitions 1 and 2, we have given the precise rules for our logic, in intuitionistic and classical format. We can freely reuse formulas and weaken the context, so the structural rules of contraction and weakening are wired into the system. To reduce the number of rules, we can take a number of rules together and drop one or more hypotheses. We illustrate this by again looking at the example of the rules for $\wedge$ (Example 3).

Example 6. *From the truth table we have derived the* 3 *intuitionistic elimination rules of Example 3. These rules can be reduced to the following* 2 *equivalent elimination rules:*

$$\frac{\vdash A \wedge B \quad A \vdash D}{\vdash D}\,\wedge\text{-}el_1 \qquad \frac{\vdash A \wedge B \quad B \vdash D}{\vdash D}\,\wedge\text{-}el_2$$

The general method is that we can replace two rules that only differ in one hypothesis, which in one rule occurs as a Lemma and in the other as a Casus, by one rule where the hypothesis is removed. It will be clear that the Γ's above are not relevant for the argument, so we will not write these.

Lemma 7. *A system with two derivation rules of the form*

$$\frac{\vdash A_1 \ldots \vdash A_n \quad B_1 \vdash D \ldots B_m \vdash D \quad C \vdash D}{\vdash D} \qquad \frac{\vdash A_1 \ldots \vdash A_n \quad \vdash C \quad B_1 \vdash D \ldots B_m \vdash D}{\vdash D}$$

is equivalent to the system with these two rules replaced by

$$\frac{\vdash A_1 \ \ldots \ \vdash A_n \quad B_1 \vdash D \ \ldots \ B_m \vdash D}{\vdash D}$$

Proof. The implication from bottom to top is immediate. From top to bottom, suppose we have the two given rules. We now derive the bottom one. Assume we have derivations of $\vdash A_1, \ldots, \vdash A_n, B_1 \vdash D, \ldots, B_m \vdash D$. We now have the following derivation of $\vdash D$.

$$\frac{\vdash A_1 \ldots \vdash A_n \quad B_1 \vdash D \ldots B_m \vdash D \qquad \dfrac{C \vdash A_1 \ldots C \vdash A_n \quad C \vdash C \quad C, B_1 \vdash D \ldots C, B_m \vdash D}{C \vdash D}}{\vdash D}$$

Similarly, we can replace a rule which has only one Casus by a rule where the Casus is the conclusion. To illustrate this: the simplified elimination rules for $\wedge$, $\wedge$-el$_1$ and $\wedge$-el$_2$ have only one Casus. The rule $\wedge$-el$_1$ (left) can thus be replaced by the rule $\wedge$-el$_1'$ (right), which is the usual projection rule.

$$\frac{\vdash A \wedge B \quad A \vdash D}{\vdash D}\,\wedge\text{-el}_1 \qquad \frac{\vdash A \wedge B}{\vdash A}\,\wedge\text{-el}_1'$$

There is a general Lemma stating this simplification is correct. The proof is similar to the proof of Lemma 4.

Lemma 8. *A system with a derivation rule of the form to the left is equivalent to the system with this rule replaced by the rule on the right.*

$$\frac{\vdash A_1 \ \ldots \ \vdash A_n \quad \psi \vdash D}{\vdash D} \qquad \frac{\vdash A_1 \ \ldots \ \vdash A_n}{\vdash \psi}$$

Definition 9. *The derivation rules for the standard intuitionistic connectives are the following. These rules are derived from the truth tables and optimized following Lemmas 7 and 8. The rules for $\wedge$ are the intro rule of Example 3 and the elimination rules of Example 6. The rules for $\neg$ are given in Example 3. The rules for $\vee$ and $\rightarrow$ and $\top$ and $\bot$ are:*

$$\frac{\vdash A \vee B \quad A \vdash D \quad B \vdash D}{\vdash D} \vee\text{-}el \qquad \frac{\vdash A}{\vdash A \vee B} \vee\text{-}in_1 \qquad \frac{\vdash B}{\vdash A \vee B} \vee\text{-}in_2 \qquad \frac{}{\vdash \top} \top\text{-}in$$

$$\frac{\vdash A \rightarrow B \quad \vdash A}{\vdash B} \rightarrow\text{-}el \qquad \frac{\vdash B}{\vdash A \rightarrow B} \rightarrow\text{-}in_1 \qquad \frac{A \vdash A \rightarrow B}{\vdash A \rightarrow B} \rightarrow\text{-}in_2 \qquad \frac{\vdash \bot}{\vdash D} \bot\text{-}el$$

Example 10. *As our only example for classical logic, we give the classical rules for implication. The elimination rule is the same ($\rightarrow$ -el above) and we also have the first introduction rule $\rightarrow$ -in_1, but in addition we have the rule $\rightarrow$ -in_2^c. We observe that this rule is classical in the sense that one can derive Peirce's law, without using negation. See the derivation below, of Peirce's law.*

$$\dfrac{\dfrac{\boxed{\dfrac{A \vdash D \quad A \rightarrow B \vdash D}{\vdash D} \rightarrow\text{-}in_2^c} \qquad \dfrac{(A \rightarrow B) \rightarrow A \vdash (A \rightarrow B) \rightarrow A \quad A \rightarrow B \vdash A \rightarrow B}{A \rightarrow B, (A \rightarrow B) \rightarrow A \vdash A}}{A \rightarrow B, (A \rightarrow B) \rightarrow A \vdash ((A \rightarrow B) \rightarrow A) \rightarrow A} \qquad \dfrac{A \vdash A}{A \vdash ((A \rightarrow B) \rightarrow A) \rightarrow A}}{A \rightarrow B \vdash ((A \rightarrow B) \rightarrow A) \rightarrow A} \rightarrow\text{-}in_2^c$$

$$\vdash ((A \rightarrow B) \rightarrow A) \rightarrow A$$

2.1 If Then Else

We now give two examples of ternary connectives that we can treat by our method: if-then-else and most, which have the obvious (classical) truth table semantics given below. We look into if-then-else in further detail and we will say something about most in Sect. 3.

A	B	C	most(A,B,C)	$A \rightarrow B/C$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

Example 11. *The constructive rules for* if-then-else *we obtain, after optimization are the following four.*

$$\frac{\vdash A{\to}B/C \quad \vdash A}{\vdash B}\ \textit{then-el} \qquad \frac{\vdash A{\to}B/C \quad A \vdash D \quad C \vdash D}{\vdash D}\ \textit{else-el}$$

$$\frac{\vdash A \quad \vdash B}{\vdash A{\to}B/C}\ \textit{then-in} \qquad \frac{A \vdash A{\to}B/C \quad \vdash C}{\vdash A{\to}B/C}\ \textit{else-in}$$

We now show in some detail that we can obtain these four optimized rules. (NB. other optimizations are possible, yielding a different set of rules.) From the lines in the truth table of $A{\to}B/C$ *with a* 0 *we get the following four elimination rules:*

$$\frac{\vdash A{\to}B/C \quad A \vdash D \quad B \vdash D \quad C \vdash D}{\vdash D} \qquad \frac{\vdash A{\to}B/C \quad A \vdash D \quad \vdash B \quad C \vdash D}{\vdash D}$$

$$\frac{\vdash A{\to}B/C \quad \vdash A \quad B \vdash D \quad C \vdash D}{\vdash D} \qquad \frac{\vdash A{\to}B/C \quad \vdash A \quad B \vdash D \quad \vdash C}{\vdash D}$$

Using Lemmas 7 and 8, these can be reduced. The two rules on the first line reduce to else-el, the two rules on the second line reduce to then-el.

Similarly, from the lines in the truth table of $A{\to}B/C$ *with a* 1 *we get four introduction rules, which can consequently be reduced to else-in and then-in.*

Example 12. *From the lines in the truth table of* $A{\to}B/C$ *with a* 1 *we get the following four classical introduction rules:*

$$\frac{A{\to}B/C \vdash D \quad A \vdash D \quad B \vdash D \quad \vdash C}{\vdash D} \qquad \frac{A{\to}B/C \vdash D \quad A \vdash D \quad \vdash B \quad \vdash C}{\vdash D}$$

$$\frac{A{\to}B/C \vdash D \quad \vdash A \quad \vdash B \quad C \vdash D}{\vdash D} \qquad \frac{A{\to}B/C \vdash D \quad \vdash A \quad \vdash B \quad \vdash C}{\vdash D}$$

Using Lemmas 7 and 8 these can be reduced to the following two. (The two rules on the first line reduce to else-in, the two rules on the second line reduce to then-in.)

$$\frac{A{\to}B/C \vdash D \quad A \vdash D \quad \vdash C}{\vdash D}\ \textit{else-in}^c \qquad \frac{\vdash A \quad \vdash B}{\vdash A{\to}B/C}\ \textit{then-in}$$

These are the classical rules for if-then-else. *Only the rule else-in*^{c} *is different from the constructive one, as given in Example 11.*

Constructively, $A{\rightarrow}B/C$ is equivalent to $(A \rightarrow B) \wedge (A \vee C)$. It can be shown that $A{\rightarrow}B/C$ is "in between" other constructive renderings of if-then-else:

$$(A \wedge B) \vee (\neg A \wedge C) \quad \overset{\not}{\vdash} \quad A{\rightarrow}B/C \quad \overset{\not}{\vdash} \quad (A \rightarrow B) \wedge (\neg A \rightarrow C)$$

The left-to-right can easily be derived, for the non-derivability of the reverse, we need a Kripke model (see Sect. 3).

If we compare with well-known classical rules for if-then-else, we observe that two of them hold, while the other fails.

Fact 13. *1.* $\mathsf{if}\ A\ \mathsf{then}\ B\ \mathsf{else}\ B \vdash B$ *and* $B \vdash \mathsf{if}\ A\ \mathsf{then}\ B\ \mathsf{else}\ B$*,*

2. $\mathsf{if}\,(\mathsf{if}\ A\ \mathsf{then}\ B\ \mathsf{else}\ C)\ \mathsf{then}\ D\ \mathsf{else}\ E \not\vdash \mathsf{if}\ A\ \mathsf{then}\,(\mathsf{if}\ B\ \mathsf{then}\ D\ \mathsf{else}\ E)\ \mathsf{else}\,(\mathsf{if}\ C\ \mathsf{then}\ D\ \mathsf{else}\ E)$

3. $\mathsf{if}\ A\ \mathsf{then}\,(\mathsf{if}\ B\ \mathsf{then}\ D\ \mathsf{else}\ E)\ \mathsf{else}\,(\mathsf{if}\ C\ \mathsf{then}\ D\ \mathsf{else}\ E) \not\vdash \mathsf{if}\,(\mathsf{if}\ A\ \mathsf{then}\ \mathsf{else}\ BC)\ \mathsf{then}\ D\ \mathsf{else}\ E$*.*

As a matter of fact, either one of the last two rules renders the connective if-then-else *classical.*

An important property is that (just as in classical logic), the constructive if-then-else, together with $\top$ and $\bot$ is *functionally complete*: all other connectives can be defined in terms of it. We prove this for $\wedge$, $\vee$, $\rightarrow$ and $\neg$. A result from Schroeder-Heister [10] implies that all constructive connectives can be defined in terms of if-then-else.

Definition 14. *We define the usual intuitionistic connectives in terms of* $\top$*,* $\bot$ *and* if-then-else*, as follows:* $A \dot{\vee} B := A{\rightarrow}A/B$*,* $A \dot{\wedge} B := A{\rightarrow}B/A$*,* $A \dot{\rightarrow} B := A{\rightarrow}B/\top$*,* $\dot{\neg}A := A{\rightarrow}\bot/\top$*.*

Lemma 15. *The defined connectives in Definition 14 satisfy the derivation rules for these same connectives as given in Definition 9. As an immediate consequence, the intuitionistic connective* if-then-else*, together with* $\top$ *and* $\bot$*, is functionally complete.*

Proof. Lemma 15 shows that the well-known intuitionistic connectives can all be defined in terms of if-then-else, $\top$ and $\bot$. In [10], it is shown that all connectives can be defined in terms of $\vee$, $\wedge$, $\rightarrow$ and $\neg$. □

3 Kripke Semantics

We now define a Kripke semantics for the intuitionistic rules and prove that it is complete. We follow standard methods, given e.g. in [11,12], which we generalize to arbitrary connectives. Formulas are built from atoms using existing or defined connectives of any arity, so for each n-ary connective c, we assume a truth table $t_c : \{0,1\}^n \rightarrow \{0,1\}$ and we have inductively defined derivability $\vdash$ as a relation between a sets of formulas and a formula above.

Definition 16. *We define a* Kripke model *as a triple* $(W, \leq, \mathsf{at})$ *where* W *is a set of worlds with a reflexive, transitive relation* $\leq$ *on it and a function* $\mathsf{at} : W \to \wp(\mathsf{At})$ *satisfying* $w \leq w' \Rightarrow \mathsf{at}(w) \subseteq \mathsf{at}(w')$.

In a Kripke model we want to define the relation $w \Vdash \varphi$ between worlds and formulas (φ *is true in world* w). We do this by defining $[\![\varphi]\!]_w \in \{0, 1\}$, with the meaning that $[\![\varphi]\!]_w = 1$ if $w \Vdash \varphi$ and $[\![\varphi]\!]_w = 0$ if $w \nVdash \varphi$.

Definition 17. *Given a Kripke model* $(W, \leq, \mathsf{at})$ *we define* $[\![\varphi]\!]_w \in \{0, 1\}$, *by induction on* φ *as follows.*

- *If* φ *is atomic, we define* $[\![\varphi]\!]_w := 1$ *if* $\varphi \in \mathsf{at}(w)$.
- *If* $\varphi = c(\varphi_1, \ldots, \varphi_n)$, *we define* $[\![\varphi]\!]_w := 1$ *if* $t_c([\![\varphi_1]\!]_{w'}, \ldots, [\![\varphi_n]\!]_{w'}) = 1$ *for each* $w' \geq w$, *where* t_c *is the truth table of* c.

We define $\Gamma \models \psi$ (ψ *is a consequence of* Γ) *as: for each Kripke model and each world* w, *if for each* φ *in* Γ, $[\![\varphi]\!]_w = 1$, *then* $[\![\psi]\!]_w = 1$.

An immediate consequence of this definition is that for all worlds w, w', if $[\![\varphi]\!]_w = 1$ and $w' \geq w$, then $[\![\varphi]\!]_{w'} = 1$.

Lemma 18 (Soundness). *If* $\Gamma \vdash \psi$, *then* $\Gamma \models \psi$

Proof. By induction on the derivation of $\Gamma \vdash \psi$. We treat the case for the last rule being an introduction: $\psi = c(\psi_1, \ldots, \psi_n)$ and we have a line $p_1, \ldots, p_n | 1$ in the truth table for c. The introduction rule then is as follows.

$$\frac{\Gamma \vdash \psi_j \text{ (for } \psi_j \text{ with } p_j = 1) \ldots \quad \ldots \Gamma, \psi_i \vdash \psi \text{ (for } \psi_i \text{ with } p_i = 0) \ldots}{\Gamma \vdash \psi} \text{ in}$$

Given a Kripke model and a world w in this model with $[\![\varphi]\!]_w = 1$ for all $\varphi \in \Gamma$, we need to prove that $[\![\psi]\!]_w = 1$. The induction hypothesis says that $[\![\psi_j]\!]_w = 1$ for all j with $p_j = 1$. Let $w' \geq w$. There are two cases: (1) $[\![\psi_i]\!]_{w'} = 1$ for some i with $p_i = 0$. Then by induction hypothesis: $[\![\psi]\!]_{w'} = 1$, so $t_c([\![\psi_1]\!]_{w'}, \ldots, [\![\psi_n]\!]_{w'}) = 1$. (2) $[\![\psi_i]\!]_{w'} = 0$ for all i with $p_i = 0$. Then $t_c([\![\psi_1]\!]_{w'}, \ldots, [\![\psi_n]\!]_{w'}) = 1$. So, for all $w' \geq w$: $t_c([\![\psi_1]\!]_{w'}, \ldots, [\![\psi_n]\!]_{w'}) = 1$. So $[\![\psi]\!]_w = 1$. □

Now we prove completeness: if $\Gamma \models \psi$, then $\Gamma \vdash \psi$. We prove this by constructing a special, universal Kripke model.

Definition 19. *For* ψ *a formula and* Γ *a set of formulas, we say that* Γ is ψ-maximal *if* $\Gamma \nvdash \psi$ *and for every formula* $\varphi \notin \Gamma$ *we have:* $\Gamma, \varphi \vdash \psi$.

If $\Gamma \nvdash \psi$, we can extend Γ to a ψ-maximal set Γ' that contains Γ as follows. Take an enumeration of the formulas as $\varphi_1, \varphi_2, \ldots$ and define recursively $\Gamma_0 := \Gamma$ and $\Gamma_{i+1} := \Gamma_i$ if $\Gamma_i, \varphi_{i+1} \vdash \psi$ and $\Gamma_{i+1} := \Gamma_i, \varphi_{i+1}$ if $\Gamma_i, \varphi_{i+1} \nvdash \psi$. Then take $\Gamma' := \bigcup_{i \in \mathbf{N}} \Gamma_i$. (NB. as always, Γ_i, φ_{i+1} denotes $\Gamma_i \cup \{\varphi_{i+1}\}$.)

Fact 20. *We list a couple of simple important facts about ψ-maximal sets Γ.*

1. *For every φ, we have $\varphi \in \Gamma$ or $\Gamma, \varphi \vdash \psi$.*
2. *So, for every φ, if $\varphi \notin \Gamma$ then $\Gamma, \varphi \vdash \psi$.*
3. *For every φ, if $\Gamma \vdash \varphi$, then $\varphi \in \Gamma$.*

Definition 21. *We define the Kripke model $U = (W, \leq, \mathsf{at})$ as follows:*

- *A world $w \in W$ is a pair (Γ, ψ) where Γ is a ψ-maximal set of formulas.*
- $(\Gamma, \psi) \leq (\Gamma', \psi') := \Gamma \subseteq \Gamma'$.
- $\mathsf{at}(\Gamma, \psi) := \Gamma \cap \mathsf{At}$.

Lemma 22. *In the model U we have, for all worlds $(\Gamma, \psi) \in W$:*

$$\forall \varphi, \varphi \in \Gamma \Leftrightarrow [\![\varphi]\!]_{(\Gamma,\psi)} = 1.$$

Proof. The proof is by induction on φ. If $\varphi \in \mathsf{At}$, the result is immediate, so suppose that $\varphi = c(\varphi_1, \ldots, \varphi_n)$ where c has truth table t_c. We prove the two directions separately.

($\Rightarrow$): Assume $\varphi \in \Gamma$.

We have $[\![\varphi]\!]_{(\Gamma,\psi)} = 1$ iff for all $\Gamma' \supseteq \Gamma$ and for all ψ', writing $w' = (\Gamma', \psi')$, we have $t_c([\![\varphi_1]\!]_{w'}, \ldots, [\![\varphi_n]\!]_{w'}) = 1$.

So let $\Gamma' \supseteq \Gamma$ and let ψ' be a formula such that Γ' is ψ'-maximal. For the sub-formulas of φ we have the following possibilities

- $[\![\varphi_j]\!]_{w'} = 1$, and then by induction hypothesis: $\varphi_j \in \Gamma'$ and so $\Gamma' \vdash \varphi_j$.
- $[\![\varphi_i]\!]_{w'} = 0$, and then by induction hypothesis: $\varphi_i \notin \Gamma'$ and so $\Gamma', \varphi_i \vdash \psi'$.

This corresponds to an entry in the truth table t_c for the connective c.

Suppose $t_c([\![\varphi_1]\!]_{w'}, \ldots, [\![\varphi_n]\!]_{w'}) = 0$. Then this row in the truth table yields an elimination rule that allows us to prove ψ':

$$\frac{\Gamma' \vdash \varphi \quad \ldots \Gamma' \vdash \varphi_j \text{ (for } \varphi_j \text{ with } [\![\varphi_j]\!]_{w'} = 1) \ldots \quad \ldots \Gamma', \varphi_i \vdash \psi' \text{ (for } \varphi_i \text{ with } [\![\varphi_i]\!]_{w'} = 0) \ldots}{\Gamma' \vdash \psi'} \text{ el}$$

Note that all hypotheses of the rule are derivable, because $\varphi \in \Gamma'$ and the other hypotheses are derivable by induction. So we have $\Gamma' \vdash \psi'$. Contradiction! So: $t_c([\![\varphi_1]\!]_{w'}, \ldots, [\![\varphi_n]\!]_{w'}) = 1$, what we needed to prove.

($\Leftarrow$): Assume $[\![\varphi]\!]_{(\Gamma,\psi)} = 1$ and suppose (towards a contradiction) $\varphi \notin \Gamma$.

Then $\Gamma \nvdash \varphi$ (because if $\Gamma \vdash \varphi$, then $\varphi \in \Gamma$ by the facts we remarked about Kripke model U.) So there is a φ-maximal theory $\Gamma' \supseteq \Gamma$ with $\Gamma' \nvdash \varphi$. So $(\Gamma,' \varphi)$ is a world in U with $(\Gamma, \psi) \leq (\Gamma', \varphi)$. We write $w' := (\Gamma', \varphi)$ and we have

$$t_c([\![\varphi_1]\!]_{w'}, \ldots, [\![\varphi_n]\!]_{w'}) = 1.$$

We consider the different sub-formulas of φ:

- the φ_j with $[\![\varphi_j]\!]_{w'} = 1$, and so (by induction hypothesis) $\varphi_j \in \Gamma'$ and so $\Gamma' \vdash \varphi_j$;
- the φ_i with $[\![\varphi_i]\!]_{w'} = 0$, and so (by induction hypothesis) $\varphi_i \notin \Gamma'$ and so $\Gamma', \varphi_i \vdash \varphi$.

Now, using an introduction rule for connective c, we can derive φ:

$$\frac{\Gamma' \vdash \varphi_j \text{ (for } \varphi_j \text{ with } [\![\varphi_j]\!]_{w'} = 1) \ldots \quad \ldots \Gamma', \varphi_i \vdash \varphi \text{ (for } \varphi_i \text{ with } [\![\varphi_i]\!]_{w'} = 0) \ldots}{\Gamma' \vdash \varphi} \text{ in}$$

So we have $\Gamma' \vdash \varphi$, because the hypotheses of the rule are all derivable. Contradiction! So $\varphi \in \Gamma'$. □

Theorem 23. *If $\Gamma \models \psi$, then $\Gamma \vdash \psi$.*

Proof. Suppose $\Gamma \models \psi$ and $\Gamma \not\vdash \psi$. We can find a ψ-maximal superset Γ' of Γ such that $\Gamma' \not\vdash \psi$. In particular: $\psi \notin \Gamma'$. So (Γ', ψ) is a world in the Kripke model U in which each member of Γ is true: $[\![\varphi]\!]_{(\Gamma',\psi)} = 1$ for $\varphi \in \Gamma$, by Lemma 22. However, ψ is not true in (Γ', ψ): $[\![\psi]\!]_{(\Gamma',\psi)} = 0$. So $\Gamma \not\models \psi$. Contradiction, so $\Gamma \vdash \psi$. □

In intuitionistic logic, the connective $\vee$ has a special property that does not hold for classical logic, called the *disjunction property*: If $\vdash A \vee B$, then $\vdash A$ or $\vdash B$. This implies that the disjunction is "strong": if one has a proof of a disjunction, one has a proof of one of the disjoints. (Which is classically not the case, viz. $\vdash A \vee \neg A$.) The disjunction property can easily be proved using Kripke semantics, relying on the completeness theorem. We want to generalize this to other connectives and we introduce the notion of a *splitting connective*.

Definition 24. *Let c be an n-ary connective, $1 \leq i, j \leq n$. We say that c is* i,j-splitting *in case the truth table for c has the following shape*

A_1	…	A_i	…	A_j	…	A_n	$c(A_1, \ldots, A_n)$
–	…	0	…	0	…	–	0
–	…	0	…	0	…	–	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
–	…	0	…	0	…	–	0
–	…	0	…	0	…	–	0

So, in all rows where $p_i = p_j = 0$ we have $c(p_1, \ldots, p_n) = 0$. Phrased purely in terms of t_c, that is: $t_c(p_1, \ldots, p_{i-1}, 0, p_{i+1}, \ldots, p_{j-1}, 0, p_{j+1}, \ldots, p_n) = 0$ for all $p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_{j-1}, p_{j+1}, \ldots, p_n \in \{0, 1\}$.

A connective can be i,j-splitting for more than one i,j-pair. Examples are the ternary connectives most and if-then-else. We now state and prove our generalization of the disjunction property.

Lemma 25. *Let c be an i,j-splitting connective and suppose $\vdash c(A_1, \ldots, A_n)$. Then $\vdash A_i$ or $\vdash A_j$.*

Proof. Let c be an i,j-splitting connective and let $\varphi = c(A_1,\ldots,A_n)$ be a formula with $\vdash \varphi$.

Suppose $\nvdash A_i$ and $\nvdash A_j$. Then there are Kripke models K_1 and K_2 such that $K_1 \Vdash A_i$ and $K_2 \Vdash A_j$. We may assume that the sets of worlds of K_1 and K_2 are disjoint so we can construct a Kripke model K as the union of K_1 and K_2 where we add a special "root world" w_0 that is below all worlds of K_1 and K_2, with $\mathsf{at}(w_0) = \emptyset$. It is easily verified that K is a Kripke model and we have $w_0 \nVdash A_i$, because w_0 is below some world w in K_1 with $w \nVdash A_i$; similarly $w_0 \nVdash A_j$. So, $[\![A_i]\!]_{w_0} = [\![A_j]\!]_{w_0} = 0$. But then $w_0 \nVdash \varphi$, because $[\![\varphi]\!]_{w_0} = [\![c(A_1,\ldots,A_n)]\!]_{w_0} = 1$ iff for all $w \geq w_0$: $t_c([\![A_1]\!]_w,\ldots,[\![A_n]\!]_w) = 1$. However, for $w := w_0$, whatever the values of $[\![A_k]\!]_w$ are for $k \neq i,j$, $t_c([\![A_1]\!]_w,\ldots,[\![A_n]\!]_w) = 0$. On the other hand, $w_0 \Vdash \varphi$, because $\vdash \varphi$, so we have a contradiction. We conclude that $\vdash A_i$ or $\vdash A_j$. □

Example 26. *Looking at the truth tables in Sect. 2.1, we see that* most *is i,j-splitting for every i,j. Indeed, if $\vdash \mathsf{most}(A,B,C)$, we can derive $\vdash A$ or $\vdash B$ but also $\vdash A$ or $\vdash C$ and also $\vdash B$ or $\vdash C$.*

The connective if-then-else *is not $1,2$-splitting but it is $1,3$-splitting and $2,3$-splitting: if $\vdash A{\to}B/C$, then we have $\vdash A$ or $\vdash C$ and also $\vdash B$ or $\vdash C$.*

4 Cuts and Cut-Elimination

The idea of a cut in intuitionistic logic is an introduction of a formula φ immediately followed by an elimination of φ. We will call this a *direct intuitionistic cut*. In general in between the intro rule for φ and the elim rule for φ, there may be other auxiliary rules, so occasionally we may have to first permute the elim rule with these auxiliary rules to obtain a direct cut that can be contracted. We leave that for future research and now just define the notion of direct cut and contraction of a direct cut.

Definition 27. *Let c be a connective of arity n, with an elim rule and an intuitionistic intro rule derived from the truth table, as in Definition 1. So suppose we have the following rules in the truth table t_c.*

$A_1 \ \ldots \ A_n$	$c(A_1,\ldots,A_n)$
$p_1 \ \ldots \ p_n$	0
$q_1 \ \ldots \ q_n$	1

An intuitionistic direct cut *in a derivation is a pattern of the following form, where $\varphi = c(A_1,\ldots,A_n)$ and: (1) A_j ranges over all formulas where $q_j = 1$, A_i ranges over all formulas where $q_i = 0$; (2) A_k ranges over all formulas where $p_k = 1$, A_ℓ over all formulas where $p_\ell = 0$,*

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\Sigma_j\\ \ldots\ \Gamma \vdash A_j\ \ldots\end{array} \qquad \begin{array}{c}\vdots\,\Sigma_i\\ \ldots\ \Gamma, A_i \vdash \varphi\ \ldots\end{array}}{\Gamma \vdash \varphi} \qquad \begin{array}{c}\vdots\,\Pi_k\\ \ldots\ \Gamma \vdash A_k\ \ldots\end{array} \qquad \begin{array}{c}\vdots\,\Pi_\ell\\ \ldots\ \Gamma, A_\ell \vdash D\ \ldots\end{array}}{\Gamma \vdash D}$$

The elimination of a direct cut *is defined by replacing the derivation pattern above by*

1. *If* $\ell = j$ *(for some* ℓ, j*):*

$$\dfrac{\dfrac{\dfrac{\vdots\,\Sigma_j}{\Gamma \vdash A_j} \quad \ldots \quad \dfrac{\vdots\,\Sigma_j}{\Gamma \vdash A_j}}{\vdots\,\Pi_\ell}}{\Gamma \vdash D}$$

2. *If* $k = i$ *(for some* k, i*):*

$$\dfrac{\dfrac{\dfrac{\vdots\,\Pi_k}{\Gamma \vdash A_i} \quad \ldots \quad \dfrac{\vdots\,\Pi_k}{\Gamma \vdash A_i}}{\vdots\,\Sigma_i}}{\Gamma \vdash \varphi} \qquad \dfrac{\ldots \quad \dfrac{\vdots\,\Pi_k}{\Gamma \vdash A_i} \quad \ldots \quad \ldots \quad \dfrac{\vdots\,\Pi_\ell}{\Gamma, A_\ell \vdash D} \quad \ldots}{\Gamma \vdash D}$$

There may be several choices for the i and j in the previous definition, so cut-elimination is non-deterministic in general. As an example, we give the cut-elimination rules for if-then-else with optimized deduction rules.

Example 28. *The intuitionistic cut-elimination rules for* if-then-else *are the following.*

(then-then)

$$\dfrac{\dfrac{\Gamma \vdash A \qquad \dfrac{\vdots\,\Sigma}{\Gamma \vdash B}}{\Gamma \vdash A{\to}B/C} \qquad \Gamma \vdash A}{\Gamma \vdash B} \quad \mapsto \quad \dfrac{\vdots\,\Sigma}{\Gamma \vdash B}$$

(else-then)

$$\dfrac{\dfrac{\dfrac{\vdots\,\Sigma}{\Gamma, A \vdash A{\to}B/C} \qquad \Gamma \vdash C}{\Gamma \vdash A{\to}B/C} \qquad \dfrac{\vdots\,\Pi}{\Gamma \vdash A}}{\Gamma \vdash B} \quad \mapsto \quad \dfrac{\dfrac{\dfrac{\vdots\,\Pi}{\Gamma \vdash A} \quad \ldots \quad \dfrac{\vdots\,\Pi}{\Gamma \vdash A}}{\vdots\,\Sigma} \qquad \dfrac{\vdots\,\Pi}{\Gamma \vdash A}}{\Gamma \vdash A{\to}B/C} \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

(then-else)

$$\dfrac{\dfrac{\vdots\,\Sigma}{\Gamma \vdash A} \quad \Gamma \vdash B}{\Gamma \vdash A{\rightarrow}B/C} \qquad \vdots\,\Pi \qquad \Gamma, A \vdash D \qquad \Gamma, C \vdash D}{\Gamma \vdash D}$$

$$\begin{array}{c} \vdots\,\Sigma \qquad\qquad \vdots\,\Sigma \\ \mapsto \Gamma \vdash A \quad \ldots \quad \Gamma \vdash A \\ \vdots\,\Pi \\ \Gamma \vdash D \end{array}$$

(else-else)

$$\dfrac{\Gamma, A \vdash A{\rightarrow}B/C \qquad \dfrac{\vdots\,\Sigma}{\Gamma \vdash C}}{\Gamma \vdash A{\rightarrow}B/C} \qquad \vdots\,\Pi \qquad \Gamma, A \vdash D \qquad \Gamma, C \vdash D}{\Gamma \vdash D}$$

$$\begin{array}{c} \vdots\,\Sigma \qquad\qquad \vdots\,\Sigma \\ \mapsto \Gamma \vdash C \quad \ldots \quad \Gamma \vdash C \\ \vdots\,\Pi \\ \Gamma \vdash D \end{array}$$

5 Conclusion and Further Work

We have introduced a general procedure for deriving natural deduction rules from truth tables that applies both to classical and intuitionistic logic. Our deduction rules obey a specific format, making it easier to study. To show that the intuitionistic rules are truly constructive we have defined a complete Kripke semantics for the intuitionistic rules. We have defined cut-elimination for intuitionistic logic in general. In an extended version of the paper [5] we have described a Curry-Howard proofs-as-terms isomorphism for the derivations in constructive logic. We have studied it in more detail for if-then-else.

The work described here raises many new research questions that we will pursue further: Is cut-elimination normalizing in general for an arbitrary set of connectives? How to define cut-elimination for the classical case, and what is its connection with a term calculus for classical logic as studied e.g. in [1,2,8]?

Another issue is the possibility of "hidden cuts" that need to be made explicit via a permuting conversion operation on the derivation (or on the proof-term). These already occur in the fragment with just if-then-else and we describe these permuting conversions in [5]. The question is if we can describe and study these permuting conversions in general.

References

1. Ariola, Z.M., Herbelin, H.: Minimal classical logic and control operators. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 871–885. Springer, Heidelberg (2003). doi:10.1007/3-540-45061-0_68
2. Curien, P.-L., Herbelin, H.: The duality of computation. In: ICFP, pp. 233–243 (2000)
3. Dyckhoff, R.: Some remarks on proof-theoretic semantics. In: Piecha, T., Schroeder-Heister, P. (eds.) Advances in Proof-Theoretic Semantics, vol. 43, pp. 79–93. Springer, Heidelberg (2016)

4. Francez, N., Dyckhoff, R.: A note on harmony. J. Philos. Logic **41**(3), 613–628 (2012)
5. Geuvers, H., Hurkens, T.: Deriving natural deduction rules from truth tables (Extended version). Technical report (2016). http://www.cs.ru.nl/~herman/PUBS/NatDedTruthTables_Extended.pdf
6. Milne, P.: Inversion principles and introduction rules. In: Dag Prawitz on Proofs and Meaning, Outstanding Contributions to Logic, vol. 7, pp. 189–224 (2015)
7. Negri, S., von Plato, J.: Structural Proof Theory. Cambridge University Press, Cambridge (2001)
8. Parigot, M.: $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In: Voronkov, A. (ed.) LPAR 1992. LNCS, vol. 624, pp. 190–201. Springer, Heidelberg (1992). doi:10.1007/BFb0013061
9. Prawitz, D.: Ideas and results in proof theory. In: Fenstad, J., (ed.) 2nd Scandinavian Logic Symposium, North-Holland, pp. 237–309 (1971)
10. Schroeder-Heister, P.: A natural extension of natural deduction. J. Symb. Log. **49**(4), 1284–1300 (1984)
11. Troelstra, A.S., van Dalen, D.: Constructivism in Mathematics, vol. 1. Elsevier, Amsterdam (1988)
12. van Dalen, D.: Logic and Structure. Universitext, 3rd edn. Springer, London (1994)
13. von Plato, J.: Natural deduction with general elimination rules. Arch. Math. Log. **40**(7), 541–567 (2001)

A Semantic Analysis of Stone and Dual Stone Negations with Regularity

Arun Kumar[1(✉)] and Mohua Banerjee[2]

[1] Department of Mathematics, Institute of Science,
Banaras Hindu University, Varanasi 221005, India
arunk2956@gmail.com
[2] Department of Mathematics and Statistics,
Indian Institute of Technology, Kanpur 208016, India
mohua@iitk.ac.in

Abstract. This article investigates whether a few well-known 'negation' operators may be termed as negations, using Dunn's approach. The semantics of the Stone negation is investigated in perp frames, that of dual Stone negation in exhaustive frames, and that of Stone and dual Stone negations with the regularity property, in K_{-} frames. The study leads to new semantics for the logics corresponding to the classes of Stone algebras, dual Stone algebras and regular double Stone algebras.

Keywords: Perp semantics · Regular double stone algebras

1 Introduction

In classical logic, the interpretation of negation ($\neg$) is such that a proposition $\neg\alpha$ is true at a state if and only if α is false at that state. However, there are different semantics of negation as we move to non-classical logics. In the well-known Kripke semantics for intuitionistic logic [1], $\neg\alpha$ is true at a state (present time point or evidential situation) w if and only if α is false (not verified) not only at w, but as well at every other accessible ('later') state. The 'perp' semantics introduced by Dunn in [2,3] provides another framework for studying various negations as modal operators. The intended interpretation of negation in this Kripke-type semantics is that of *impossibility* or *unnecessity*. As mentioned by Dunn in [4], the motivation of perp semantics lies in the Birkhoff-von Neumann-Goldblatt definition of ortho-negation in quantum logic, where ortho-negation is described using a relation of 'incompatibility' (or orthogonality or perp) between states. $\neg\alpha$ is true at a given state w if and only if w is incompatible with every state at which α is true. Dunn in his model of negations, uses a 'compatibility' relation to define the perp semantics. Propositions are interpreted in frames containing a compatibility relation, and $\neg\alpha$ is true at a given state w if and only if α is false at all states which are compatible with w.

In this article, we study a few well-known 'negation' operators, investigating whether they qualify to be negations in accordance with the (perp) semantic

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 139–153, 2017.
DOI: 10.1007/978-3-662-54069-5_11

analysis due to Dunn. Our motivation for selecting the particular operators under study here, comes from our interest and work in classical Pawlakian rough set theory [5]. Rough sets have been shown to form various algebraic structures (cf. [6]), such as Stone, regular double Stone, Nelson, topological quasi Boolean, pre-rough, rough algebras, or Kleene algebras [7]. For many of these classes of algebras, representation theorems in terms of rough set algebras have been obtained as well. All the afore-mentioned algebras involve some unary operators as algebraic 'negations'. One is a De Morgan as well as a Kleene negation, another a Stone negation, and a third one is a dual Stone negation. Moreover, the last two together satisfy the *regularity* property. We pose the question about the kind of semantics these operators induce, in line with Dunn's approach to the study of negations.

The characterization of the De Morgan laws in the perp semantic framework is given by Dunn [8] and Restall [9]. In [7], we obtained a characterization for Kleene negation in perp frames. In this work, our focus is on Stone and dual Stone negations satisfying the regularity property.

The basic logic involved in the study is the bounded distributive lattice logic $BDLL$. Distributive lattices are algebraic models of the logic DLL introduced by Dunn [10]. The language of DLL consists of propositional variables $p, q, r, \ldots$, and the logical connectives $\vee, \wedge$. The well-formed formulas of the logic are then given by the scheme: $p \mid \alpha \vee \beta \mid \alpha \wedge \beta$. Let us denote the set of propositional variables by $\mathcal{P}$, and that of well-formed formulas by $\mathcal{F}$. Consequents $\alpha \vdash \beta$ are used to define the system through the following postulates and rules – for this, we refer to [4,8].

Definition 1 (DLL- postulates)**.**

1. $\alpha \vdash \alpha$ (Reflexivity)
2. $\dfrac{\alpha \vdash \beta \quad \beta \vdash \gamma}{\alpha \vdash \gamma}$ (Transitivity)
3. $\alpha \wedge \beta \vdash \alpha,\ \alpha \wedge \beta \vdash \beta$ (Conjunction Elimination)
4. $\dfrac{\alpha \vdash \beta \quad \alpha \vdash \gamma}{\alpha \vdash \beta \wedge \gamma}$ (Conjunction Introduction)
5. $\alpha \vdash \alpha \vee \beta,\ \beta \vdash \alpha \vee \beta$ (Disjunction Introduction)
6. $\dfrac{\alpha \vdash \gamma \quad \beta \vdash \gamma}{\alpha \vee \beta \vdash \gamma}$ (Disjunction Elimination)
7. $\alpha \wedge (\beta \vee \gamma) \vdash (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ (Distributivity)

In [10], DLL was extended by adding the propositional constants $\top, \bot$ and the postulates below to give the logic $BDLL$, whose algebraic models are bounded distributive lattices:

– $\alpha \vdash \top$ (Top);
– $\bot \vdash \alpha$ (Bottom).

In this article, we study logics $\mathcal{L}$ that are all extensions of $BDLL$. By $\alpha \vdash_{\mathcal{L}} \beta$, we shall mean that the consequent $\alpha \vdash \beta$ is derivable in $\mathcal{L}$. The algebraic semantics

of $\mathcal{L}$ is defined in the standard way. If $\mathcal{A}_{\mathcal{L}}$ denotes the class of all algebras corresponding to the logic $\mathcal{L}$, validity of $\alpha \vdash \beta$ in $\mathcal{A}_{\mathcal{L}}$ will be denoted as $\alpha \vDash_{\mathcal{A}_{\mathcal{L}}} \beta$.

In Sect. 2, we present the preliminaries of perp semantics, and then study the perp semantics for the logic of Stone algebras. The study of the semantics for the logic of dual Stone algebras in 'exhaustive' frames is given in Sect. 3. In Sect. 4, semantics for the logic of regular double Stone algebras is studied in a 'united' framework. Section 5 concludes the article.

2 The Stone Negation in the Kite of Negations

Let us present the basic notions in perp semantics. The language of the extensions of $BDLL$ considered here contain an additional unary connective $\sim$, to stand for negation.

Definition 2. *A* compatibility frame *is a triple* $(W, C, \leq)$ *with the following properties:*

1. $(W, \leq)$ *is a partially ordered set;*
2. C *is a binary relation on* W *such that for* $x, y, x', y' \in W$, *if* $x' \leq x$, $y' \leq y$ *and* xCy *then* $x'Cy'$.

C *is called a* compatibility relation *on* W.

A perp frame *is a tuple* $(W, \perp, \leq)$, *where* $\perp$, *the* perp relation *on* W, *is the complement of the compatibility relation* C.

As in [8], we do not distinguish between compatibility and perp frames, and present the results in terms of the compatibility relation.
A relation $\vDash$ between points of W and propositional variables in $\mathcal{P}$ is called an *evaluation*, if it satisfies the *hereditary* condition:

$$\text{if } x \vDash p \text{ and } x \leq y \text{ then } y \vDash p, \text{ for any } x, y \in W.$$

Recursively, an evaluation $\vDash$ can be extended to $\mathcal{F}$; in particular, the definition for the $\sim$ case is given for any $x \in W$ as:

$$x \vDash \sim \alpha \text{ if and only if for all } y \in W,\ xCy \text{ implies that } y \nvDash \alpha.$$

For a compatibility frame $\mathbf{F} := (W, C, \leq)$ and an evaluation $\vDash$, the pair $(\mathbf{F}, \vDash)$ is called a *model*. The notion of validity is given in the usual manner. A consequent $\alpha \vdash \beta$ is *valid in a model* $(\mathbf{F}, \vDash)$, denoted as $\alpha \vDash_{(\mathbf{F},\vDash)} \beta$, if and only if, if $x \vDash \alpha$ then $x \vDash \beta$, for each $x \in W$. If $\mathbb{F}$ denotes a class of compatibility frames, $\alpha \vdash \beta$ is *valid in* $\mathbb{F}$, denoted as $\alpha \vDash_{\mathbb{F}} \beta$, if and only if $\alpha \vDash_{(\mathbf{F},\vDash)} \beta$ for all $\mathbf{F} \in \mathbb{F}$.

In [8] it has been proved that the following logic K_i is the minimal logic which is sound and complete with respect to the class of all compatibility frames. K_i is built upon the logic $BDLL$, by adding the following rules and postulates.

1. $\dfrac{\alpha \vdash \beta}{\sim \beta \vdash \sim \alpha}$ (Contraposition).

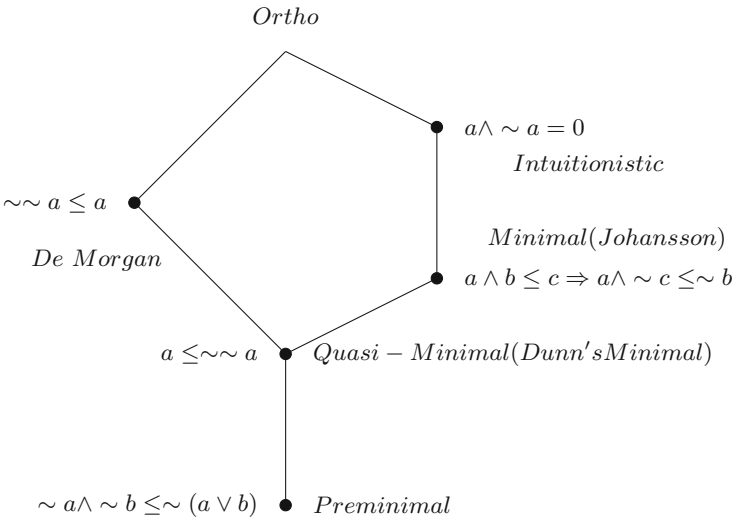


Fig. 1. Dunn's Lopsided Kite of Negations

2. $\sim \alpha \wedge \sim \beta \vdash \sim (\alpha \vee \beta)$ ($\vee$-linearity).
3. $\top \vdash \sim \bot$ (Nor).

Further, Dunn [2,3,8] established correspondence and completeness results about various negations with respect to perp semantics and arrived at the *lopsided kite of negations* (Fig. 1).

Frame completeness results for various logics with negation have been proved using the canonical frames for the logics. Let Λ denote any extension of K_i. The definitions for the canonical frame are as follows. A set P of sentences in Λ is called a *prime theory* if

1. $\alpha \vdash \beta$ holds and $\alpha \in P$, then $\beta \in P$,
2. $\alpha, \beta \in P$ then $\alpha \wedge \beta \in P$,
3. $\top \in P$ and $\bot \notin P$,
4. $\alpha \vee \beta \in P$ implies $\alpha \in P$ or $\beta \in P$.

Let W_c be the collection of all prime theories of Λ. Define a relation C_c on W_c as $P_1 C_c P_2$ if and only if, for all sentences α of $\mathcal{F}$, $\sim \alpha \in P_1$ implies $\alpha \notin P_2$. The tuple $(W_c, C_c, \subseteq)$ is the *canonical frame* for Λ. Λ is called *canonical*, if its canonical frame is a frame for Λ.

Let us note that the algebraic structures corresponding to the logic K_i, the *K_i-algebras*, are structures of the form $(K, \vee, \wedge, \sim, 0, 1)$ where

1. $(K, \vee, \wedge, 0, 1)$ is a bounded distributive lattice,
2. $\forall a, b \in K$, $a \leq b$ implies $\sim b \leq \sim a$,
3. $\sim a \wedge \sim b \leq \sim (a \vee b)$, and
4. $\sim 0 = 1$.

2.1 Stone Property

In [8], an intuitionistic negation is defined as one having the (1) *Absurd* ($a\wedge \sim a = 0$) and (2) *minimal (Johansson)* ($a \wedge b \leq c \Rightarrow a\wedge \sim c \leq \sim b$) properties. The usual definition of pseudo complement in a bounded distributive lattice L is:

$$\sim a := max\{c \in L : a \wedge c = 0\},\ a \in L. \qquad (*)$$

One can show that it satisfies (1) and (2); on the other hand, if $\sim$ in a K_i-algebra satisfies (1) and (2), then it coincides with the pseudo complement defined in (*). Hence a *Stone algebra* can be defined as a K_i-algebra with $\sim$ satisfying (1), (2) and the *Stone property*: $\sim a \vee \sim\sim a = 1$. $\sim$ is a *Stone negation*.

We prove a correspondence result for the Stone property.

Theorem 1. *$\top \vdash \sim \alpha \vee \sim\sim \alpha$ is valid in a compatibility frame $(W, C, \leq)$ if and only if C satisfies the following first order property:*

$$\forall x \forall y_1 \forall y_2 (xCy_1 \wedge xCy_2 \rightarrow (y_1Cy_2 \wedge y_2Cy_1)). \qquad (*)$$

The extension of the logic K_i having the axiom $\top \vdash \sim \alpha \vee \sim\sim \alpha$, is canonical.

Proof. Let (*) hold in any compatibility frame $(W, C, \leq)$, and let us assume $x \not\models \sim \alpha \vee \sim\sim \alpha$, i.e., $x \not\models \sim \alpha$ and $x \not\models \sim\sim \alpha$.

$x \not\models \sim \alpha$ implies there exists y_1 such that xCy_1 and $y_1 \models \alpha$. $x \not\models \sim\sim \alpha$ implies there exists y_2 such that xCy_2 and $y_2 \models \sim \alpha$. As, (*) holds, we have y_1Cy_2 and y_2Cy_1. But y_2Cy_1 and $y_1 \models \alpha$ imply $y_2 \not\models \sim \alpha$ which is a contradiction. Hence, $x \models \sim \alpha \vee \sim\sim \alpha$.

Now suppose (*) does not hold. This means $\exists x \exists y_1 \exists y_2((xCy_1 \wedge xCy_2) \wedge (not(y_1Cy_2) \vee not(y_2Cy_1)))$. Assume $not(y_1Cy_2)$ is true and define, $z \models p$ if and only if $y_2 \leq z$ and $not(y_1Cz)$. Let us first show that the relation $\models$ is well defined, i.e. hereditary. So let $z \models p$ and $z \leq z'$, hence, $y_2 \leq z'$. If y_1Cz' then using the frame condition we have y_1Cz, which is a contradiction.

We have $x \not\models \sim p$ as xCy_2 and $y_2 \models p$ (as $not(y_1Cy_2)$). Also, $x \not\models \sim\sim p$ as xCy_1 and y_1Cz imply $z \not\models p$. Hence we have $x \not\models \sim p \vee \sim\sim p$.

Canonicity: First observe that for any prime theory P and any formula α of this logic, $\sim \alpha \vee \sim\sim \alpha \in P$, which implies either $\sim \alpha \in P$ or $\sim\sim \alpha \in P$.
Now let PC_cQ_1 and PC_cQ_2, for any prime theories P, Q_1 and Q_2. We want to show that $Q_1C_cQ_2$ and $Q_2C_cQ_1$. Let us show $Q_1C_cQ_2$.
Let $\sim \alpha \in Q_1$ but we have PC_cQ_1 hence $\sim\sim \alpha \notin P$. This means $\sim \alpha \in P$. We also have PC_cQ_2 which will give us $\alpha \notin Q_2$. Hence $Q_1C_cQ_2$.
Similarly one can show $Q_2C_cQ_1$. □

The Absurd and Minimal (Johansson) properties are characterized in [2,8].

Theorem 2 *([2,8]).*

1. *The rule* $\dfrac{\alpha \wedge \beta \vdash \gamma}{\alpha \wedge \sim \gamma \vdash \sim \beta}$ *is valid in a compatibility frame if and only if the following frame condition holds:*

$$\forall x \forall y (xCy \rightarrow \exists z (x \leq z \wedge y \leq z \wedge xCz)).$$

2. $\alpha \wedge \sim \alpha \vdash \bot$ *is valid, precisely in the class of all compatibility frames satisfying the frame condition:* $\forall x (xCx)$.

Moreover, canonicity holds in the respective cases.

Hence Stone negation can be visualized as an 'impossibility' as well as a modal operator. Let $\mathcal{L}_S$ denote K_i enhanced with the following rules and postulates.

(i) $\dfrac{\alpha \wedge \beta \vdash \gamma}{\alpha \wedge \sim \gamma \vdash \sim \beta}$
(ii) $\alpha \wedge \sim \alpha \vdash \bot$
(iii) $\top \vdash \sim \alpha \vee \sim\sim \alpha$.

Let us call a compatibility frame $(W, C, \leq)$ a *Stone frame* if it satisfies the frame conditions:

1. $\forall x \forall y (xCy \rightarrow \exists z (x \leq z \wedge y \leq z \wedge xCz))$,
2. $\forall x (xCx)$ and
3. $\forall x \forall y_1 \forall y_2 (xCy_1 \wedge xCy_2 \rightarrow (y_1 Cy_2 \wedge y_2 Cy_1))$.

Let $\mathbb{F}_S$ denote the class of all Stone frames, and $\mathcal{A}_S$, the class of all Stone algebras. Then we can conclude the following theorem, and that the Stone negation can be positioned in Dunn's Lopsided Kite of Negations (Fig. 2).

Theorem 3. *For any* $\alpha, \beta \in \mathcal{F}$. *The following are equivalent.*

(i) $\alpha \vdash_{\mathcal{L}_S} \beta$.
(ii) $\alpha \models_{\mathcal{A}_S} \beta$.
(iii) $\alpha \models_{\mathbb{F}_S} \beta$.

3 The Dual Stone Negation in the Dual Kite

Dunn shows in [8] that the dual of the negations in the kite of negations can be studied via the 'dual' of compatibility frames, namely *exhaustive frames* (defined below). Through the semantics in exhaustive frames, it has been shown that the negations in the *dual lopsided kite of negations* can be treated as modal operators. But in this case, modalities are interpreted as 'unnecessity'. To study the dual Stone negation, we make use of this semantics. The additional unary connective for negation in the language of the extensions of $BDLL$ considered in this case, is denoted $\neg$.

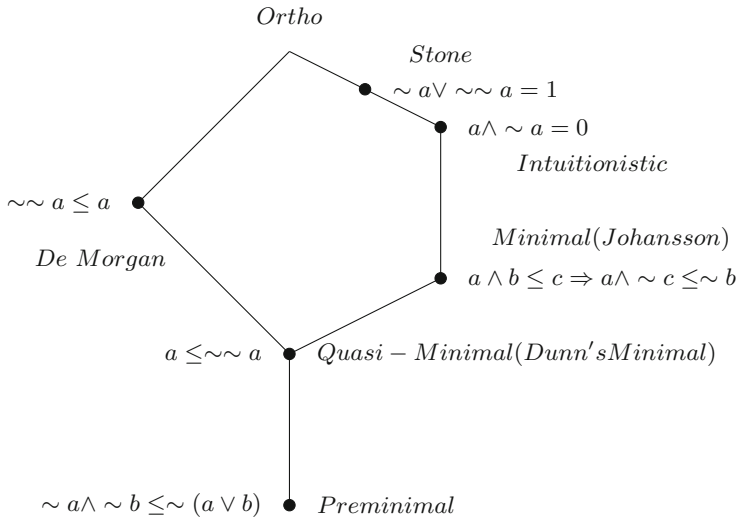


Fig. 2. Enhanced Lopsided Kite of Negations

Definition 3. *An* exhaustive frame *is a triple* $(W, R, \leq)$ *such that*

1. $(W, \leq)$ *is a partially ordered set, and*
2. $\leq \circ R \circ \leq^{-1} \subseteq R$.

It is observed in [8] that exhaustive frames are the 'dual' of compatibility frames. In fact, any compatibility frame is an exhaustive frame, but the interpretations for negation are different in these frames. Negation is interpreted as 'impossibility' in compatibility frames, while in exhaustive frames negation is interpreted as 'unnecessity'.
A relation $\vDash$ between points of W and propositional variables in $\mathcal{P}$ is called an *evaluation* here, if it satisfies the *backward hereditary* condition:

$$\text{if } x \vDash p \text{ and } y \leq x \text{ then } y \vDash p, x, y \in W.$$

$\vDash$ can be recursively extended to the set $\mathcal{F}$, with the evaluation of the formula $\neg\alpha$ at $x \in W$ given as:

$$x \vDash \neg\alpha \text{ if and only if } \exists y(xRy \wedge y \nvDash \alpha).$$

The notion of validity is as in the previous section. The minimal logic which is sound and complete with respect to the class of all exhaustive frames, is the logic K_u [8]. It is $BDLL$ enhanced with:

1. $\dfrac{\alpha \vdash \beta}{\neg\beta \vdash \neg\alpha}$ (Contraposition).
2. $\neg(\alpha \wedge \beta) \vdash \neg\alpha \vee \neg\beta$ ($\wedge$-linearity).
3. $\neg\top \vdash \bot$ (dual-Nor).

The completeness results in this case, are also proved using the canonical model. The definition is as follows. For any extension Λ of the logic K_u, let W_c be the collection of all the prime theories. The canonical relation R_c on W_c is defined as: PR_cQ if and only if, for all sentences α, $\neg\alpha \notin P$ implies $\alpha \in Q$. The tuple $(W_c, R_c, \supseteq)$ is the *canonical frame* for Λ.

The algebras corresponding to the logic K_u are the $K_u - algebras$, which are structures of the form $(K, \vee, \wedge, \neg, 0, 1)$, where

1. $(K, \vee, \wedge, 0, 1)$ is a bounded distributive lattice,
2. $\forall a, b \in K$, $\neg(a \wedge b) = \neg a \vee \neg b$,
3. $a \leq b \Rightarrow \neg b \leq \neg a$, and
4. $\neg 1 = 0$.

3.1 Dual Stone Property

The dual pseudo complement in a distributive lattice L is defined as:

$$\neg a := min\{c \in L : a \vee c = 1\},\ a \in L.$$

It can be shown that a K_u-algebra with $\neg$ satisfying the (1) *dual intuitionistic property* ($a \vee \sim a = 0$) and (2) *dual minimal (Johansson) property* ($c \leq a \vee b \Rightarrow \neg b \leq a \vee \neg c$), coincides with a dual pseudo complemented lattice. Thus a *dual Stone algebra* is a K_u-algebra with negation satisfying (1), (2) and the *dual Stone* property: $\neg a \wedge \neg\neg a = 0$. $\neg$ is a *dual Stone negation*.

Theorem 4 *(Dunn [8])*.

1. $\top \vdash \alpha \vee \neg\alpha$ *is valid in an exhaustive frame if and only if the frame satisfies the following first order condition:* $\forall x(xRx)$.
2. *The rule* $\dfrac{\gamma \vdash \alpha \vee \beta}{\neg\beta \vdash \alpha \vee \neg\gamma}$ *is valid in an exhaustive frame if and only if the frame satisfies the following first order condition:*

$$\forall x \forall y(xRy \rightarrow \exists z(x \leq z \wedge y \leq z \wedge xRz)).$$

We characterize the dual Stone property in an exhaustive frame.

Theorem 5. $\neg\alpha \wedge \neg\neg\alpha \vdash \bot$ *is valid in an exhaustive frame* $(W, R, \leq)$ *if and only if* R *satisfies the following first order property:*

$$\forall x \forall y_1 \forall y_2(xRy_1 \wedge xRy_2 \rightarrow (y_1Ry_2 \wedge y_2Ry_1)). \qquad (*)$$

The extension of K_u *containing the axiom* $\neg\alpha \wedge \neg\neg\alpha \vdash \bot$, *is canonical.*

Proof. Let $(^*)$ hold in an exhaustive frame $(W, R, \leq)$ and let $x \in W$. Suppose $x \vDash \neg\alpha \wedge \neg\neg\alpha$, this implies $x \vDash \neg\alpha$ and $x \vDash \neg\neg\alpha$. Hence there exist y_1, y_2 such that xRy_1, xRy_2 and $y_1 \nvDash \alpha$, $y_2 \nvDash \neg\alpha$. As $(^*)$ holds, y_2Ry_1. But then $y_2 \vDash \neg\alpha$, which is a contradiction. Hence, $x \nvDash \neg\alpha \wedge \neg\neg\alpha$.

Suppose (*) does not hold. This means $\exists x \exists y_1 \exists y_2((xRy_1 \wedge xRy_2) \wedge (not(y_1Ry_2) \vee not(y_2Ry_1)))$. Assume $not(y_1Ry_2)$ and define, $z \vDash p$ if and only if y_1Rz.

Then $\vDash$ is well-defined: let $z \vDash p$ and $z' \leq z$, then using the property of exhaustive frames we have, $y_1 \leq y_1$, y_1Rz and $z \geq z'$ imply y_1Rz'. Hence $z' \vDash p$.

Now, clearly we have $x \vDash \neg p$, as, xRy_2 and $y_2 \nvDash p$. Also, $x \vDash \neg\neg p$ as, xRy_1 and $y_1 \nvDash \neg p$. Hence $x \vDash \neg p \wedge \neg\neg p$. So, $\neg p \wedge \neg\neg p \vdash \bot$ is not valid in $(W, R, \leq)$.

Canonicity: First observe that for any prime theory P and any formula α, $\neg\alpha \wedge \neg\neg\alpha \notin P$, as we have assumed our logic contains $\neg\alpha \wedge \neg\neg\alpha \vdash \bot$. This implies $\neg\alpha \notin P$ or $\neg\neg\alpha \notin P$. Now let PR_cQ_1 and PR_cQ_2. We want to show that $Q_1R_cQ_2$ and $Q_2R_cQ_1$. Let us show $Q_1R_cQ_2$. So, let $\neg\alpha \notin Q_1$ but we have PR_cQ_1 hence $\neg\neg\alpha \in P$. This means $\neg\alpha \notin P$. We also have PR_cQ_2 which will give us $\alpha \in Q_2$. Hence $Q_1R_cQ_2$.

Similarly one can show $Q_2R_cQ_1$. □

Let $\mathcal{L}_{DS}$ denote the logic K_u with the following additional rules and postulates.

$$\text{(i)}\ \frac{\gamma \vdash \alpha \vee \beta}{\neg\beta \vdash \alpha \vee \neg\gamma}.$$

(ii) $\top \vdash \alpha \vee \neg\alpha$.

(iii) $\neg\alpha \wedge \neg\neg\alpha \vdash \bot$.

Let us call an exhaustive frame $(W, R, \leq)$ a *dual Stone frame* if it satisfies the following frame conditions.

1. $\forall x \forall y(xRy \rightarrow \exists z(x \leq z \wedge y \leq z \wedge xRz))$.
2. $\forall x(xRx)$.
3. $\forall x \forall y_1 \forall y_2(xRy_1 \wedge xRy_2 \rightarrow (y_1Ry_2 \wedge y_2Ry_1))$.

Denote by $\mathbb{F}_{DS}$, the class of all dual Stone frames, and let $\mathcal{A}_{DS}$ denote the class of dual Stone algebras. The following theorem results, and we conclude that the dual Stone negation can be positioned in Dunn's dual (Lopsided) Kite of Negation (Fig. 3).

Theorem 6. *For any $\alpha, \beta \in \mathcal{F}$, the following are equivalent.*

(i) $\alpha \vdash_{\mathcal{L}_{DS}} \beta$.
(ii) $\alpha \vDash_{\mathcal{A}_{DS}} \beta$.
(iii) $\alpha \vDash_{\mathbb{F}_{DS}} \beta$.

4 Stone and Dual Stone Negations with Regularity in the United Kite

Dunn further provided a 'uniform' semantics for combining both the kite of negations and its dual to give the *united kite of negations*. The minimal logic in this context is K_{-}, an extension of $BDLL$ containing unary connectives $\sim, \neg$ to stand for two negations, and the following postulates and rules.

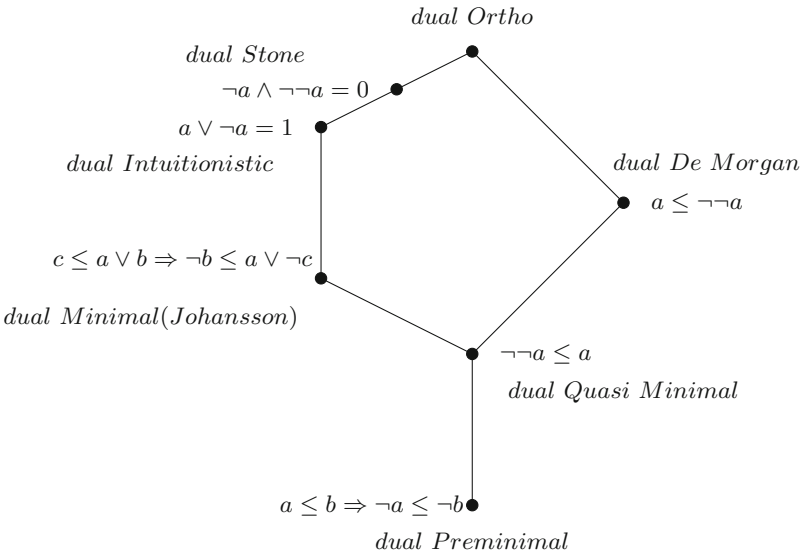


Fig. 3. Enhanced Dual Lopsided Kite of Negations

1. $\sim \alpha \wedge \sim \beta \vdash \sim (\alpha \vee \beta)$.
2. $\neg(\alpha \wedge \beta) \vdash \neg\alpha \vee \neg\beta$.
3. $\top \vdash \sim \bot$.
4. $\neg\top \vdash \bot$.
5. $\dfrac{\alpha \vdash \beta}{\sim \beta \vdash \sim \alpha}$.
6. $\dfrac{\alpha \vdash \beta}{\neg\beta \vdash \neg\alpha}$.
7. $\sim \alpha \wedge \neg\beta \vdash \neg(\alpha \vee \beta)$.
8. $\sim (\alpha \wedge \beta) \vdash \sim \alpha \vee \neg\beta$.

The semantics of the logic $K_{\sim}$ is defined in a $K_{\sim}$ *frame*, a triple $(W, R, \leq)$ such that

1. $(W, \leq)$ is a partially ordered set,
2. $\leq^{-1} \circ R \subseteq R \circ \leq$, and
3. $\leq \circ R \subseteq R \circ \leq^{-1}$.

The semantic clauses are as in the previous sections. The evaluations are defined as in Sect. 2; the extension to the cases for the two negations are given for any $x \in W$ as:

$x \vDash \sim \alpha$ if and only if $\forall y(xRy \rightarrow y \nvDash \alpha)$, and
$x \vDash \neg\alpha$ if and only if $\exists y(xRy \wedge y \nvDash \alpha)$.

Let $R_{\neg} := R \circ \leq$ and $R_{\sim} := R \circ \leq^{-1}$. The semantic clauses for $\sim$ and $\neg$ can be re-defined in terms of $R_{\neg}$ and $R_{\sim}$.

Lemma 1. *[8] For any $x \in W$,*
$x \vDash \sim \alpha$ if and only if $\forall y(xR_{\sim}y \rightarrow y \nvDash \alpha)$,
$x \vDash \neg\alpha$ if and only if $\exists y(xR_{\neg}y \wedge y \nvDash \alpha)$.

The canonical model for K_{-} is defined as follows. Let P be a prime theory, and consider the two sets $P_{\neg} := \{\alpha : \neg\alpha \notin P\}$ and $P_{\sim} := \{\alpha : \sim \alpha \notin P\}$. $(W_c, R_c, \subseteq_c)$ can be shown [8] to be a K_{-} frame and is called the canonical model for K_{-}, where W_c is the collection of all prime theories, $\subseteq_c$ is the inclusion relation and R_c is defined as follows: PR_cQ if and only if $P_{\neg} \subseteq Q \subseteq P_{\sim}$. [8] also proves

Lemma 2. *If $P_{\neg} \subseteq Q$, then $PR_{C\neg}Q$; if $Q \subseteq P_{\sim}$, $PR_{C\sim}Q$.*

K_{-} is sound and complete with respect to the class of all K_{-} *− algebras*, which are defined to be structures of the form $(K, \vee, \wedge, \sim, \neg, 0, 1)$, where

1. $(K, \vee, \wedge, \sim, 0, 1)$ is a K_i- algebra,
2. $(K, \vee, \wedge, \neg, 0, 1)$ is a K_u- algebra,
3. $\forall a, b \in K$, $(\sim a \wedge \neg b) \leq \neg(a \vee b)$,
4. $\forall a, b \in K$, $\sim (a \wedge b) \leq (\sim a \vee \neg b)$.

4.1 Stone and Dual Stone Negations with Regularity

A *regular double Stone algebra* $(K, \vee, \wedge, \sim, \neg, 0, 1)$ is a bounded distributive lattice such that (1) $\sim$ defines a Stone negation, (2) $\neg$ defines a dual Stone negation and (3) $a \wedge \neg a \leq b \vee \sim b$, for all $a, b \in K$. Note that regularity can also be characterized [11] as: $\sim a = \sim b$ and $\neg a = \neg b$ imply $a = b$, $a, b \in K$. A regular double Stone algebra is a K_{-}-algebra with the negations satisfying (1)–(3). Observe that a negation that is a Stone as well as dual Stone negation, is just an Ortho (Boolean) negation.

A sequent calculus for the logic of regular double Stone algebras and a rough set semantics for it was provided in [12,13]. In this section, we present another semantics for this logic. For that, we characterize the Stone, dual Stone and regularity properties in a K_{-} frame.

Theorem 7.

1. *$\top \vdash \sim \alpha \vee \sim\sim \alpha$ is valid in a K_{-} frame $(W, R, \leq)$ if and only if the frame satisfies the following first order property:*

$$\forall x \forall y_1 \forall y_2 (xR_{\sim}y_1 \wedge xR_{\sim}y_2 \rightarrow (y_1R_{\sim}y_2 \wedge y_2R_{\sim}y_1)).$$

2. *$\neg\alpha \wedge \neg\neg\alpha \vdash \bot$ is valid in a K_{-} frame $(W, R, \leq)$ if and only if the frame satisfies the following first order condition:*

$$\forall x \forall y_1 \forall y_2 (xR_{\neg}y_1 \wedge xR_{\neg}y_2 \rightarrow (y_1R_{\neg}y_2 \wedge y_2R_{\neg}y_1)).$$

3. *$\alpha \wedge \neg\alpha \vdash \beta \vee \sim \beta$ is valid in a K_{-} frame $(W, R, \leq)$ if and only if the frame satisfies the following first order property:*

$$\forall x((\forall y(xR_{\neg}y \rightarrow x \leq y)) \vee (\forall z(xR_{\sim}z \rightarrow z \leq x))). \qquad (*)$$

Moreover, canonicity of the enhanced logics holds in all the cases.

Proof. We only prove the canonicity parts for items 1 and 2.
1. *Canonicity:*
For any prime theories P, Q_1 and Q_2, let $PR_{c\sim}Q_1$ and $PR_{c\sim}Q_2$. Our claim is $Q_1R_{c\sim}Q_2$ and $Q_2R_{c\sim}Q_1$. Let us show $Q_1R_{c\sim}Q_2$, the other will follow similarly.

$PR_{c\sim}Q_1$, i.e., $PR_c\circ \subseteq^{-1} Q_1$ implies there exists P_1 such that PR_cP_1 and $P_1 \supseteq Q_1$. By definition of R_c, we have PR_cP_1 implies $P_\neg \subseteq P_1 \subseteq P_\sim$.

$PR_{c\sim}Q_2$ implies that there exists P_2 such that PR_cP_2 and $P_2 \supseteq Q_2$. By definition of R_c, PR_cP_2 implies $P_\neg \subseteq P_2 \subseteq P_\sim$. Let us show that $Q_2 \subseteq Q_{1\sim}$: let $\alpha \notin Q_{1\sim}$. Then $\sim\alpha \in Q_1$, which implies $\sim\alpha \in P_1$. So $\sim\alpha \in P_\sim$, whence $\sim\sim\alpha \notin P$. Then $\sim\alpha \in P$, and then $\alpha \notin P_\sim$, i.e. $\alpha \notin P_2$, implying $\alpha \notin Q_2$. Hence $Q_2 \subseteq Q_{1\sim}$. Using Lemma 2, we have $Q_1R_{c\sim}Q_2$.
2. *Canonicity:*
Let $P, Q_1, Q_2 \in W_c$ such that $PR_{c\neg}Q_1$ and $PR_{c\neg}Q_2$. $PR_{c\neg}Q_1$ implies that there exists a prime theory P_1 such that PR_cP_1 and $P_1 \subseteq Q_1$. Let us show that $Q_1R_{c\neg}Q_2$. In other words, in view of Lemma 2, we have to show that $Q_{1\neg} \subseteq Q_2$. Let $\alpha \in Q_{1\neg}$. Then $\neg\alpha \notin Q_1$, which implies $\neg\alpha \notin P_1$. So $\neg\alpha \notin P_\neg = \{\beta : \neg\beta \notin P\}$, whence $\neg\neg\alpha \in P$. But we have for any $\beta \in \mathcal{F}$ either $\neg\beta \notin P$ or $\neg\neg\beta \notin P$. Hence we have $\neg\alpha \notin P$. So $\alpha \in P_\neg$ and $PR_{c\neg}Q_2$, hence $\alpha \in Q_2$. Hence we have $Q_{1\neg} \subseteq Q_2$.
3. Let (*) hold in any K_- frame $(W, R, \leq)$, and let $x \in W$. Assume $\forall y(xR_\neg y \rightarrow x \leq y)$ is true. Let us show that $x \nvDash \alpha \wedge \neg\alpha$. Assume $x \vDash \alpha$. Let $xR_\neg y$, then by our assumption $x \leq y$. Hence using hereditary property of $\vDash$, $y \vDash \alpha$. So $x \nvDash \neg\alpha$, whereby $x \nvDash \alpha \wedge \neg\alpha$. Now let $\forall z(xR_\sim y \rightarrow z \leq x)$ be true. Let us show that $x \vDash \beta \vee \sim\beta$. Let $x \nvDash \beta$, and $xR_\sim z$. Then by our assumption $z \leq x$. Using hereditary property of $\vDash$ again, we have $z \nvDash \beta$, hence $x \vDash \sim\beta$.

In either case, if $x \vDash \alpha \wedge \sim\alpha$ then $x \vDash \beta \vee \sim\beta$ holds.

Now, let (*) not hold. Then $\exists x((\exists y_1(xR_\neg y_1 \wedge x \nleq y_1)) \wedge (\exists y_2(xR_\sim y_2 \wedge y_2 \nleq x)))$. Let us define $\vDash$ as: (i) $y \vDash p$ if and only if $x \leq y$, and (ii) $z \vDash q$ if and only if $z \nleq x$. One can show that $\vDash$ is a well defined consequence relation: (1) let $y \vDash p$ and $y \leq y'$. Then $x \leq y \leq y'$. Hence $y' \vDash p$. (2) Let $z \vDash q$ and $z \leq z'$. If $z' \nvDash q$ then by definition of $\vDash$, $z' \leq x$. Hence $z \leq z' \leq x$, which implies $z \nvDash q$ contradicting our assumption.

Now $x \vDash p \wedge \neg p$: $x \vDash p$, using the definition of $\vDash$. As (*) does not hold, we have y_1 in W such that $xR_\neg y_1$ and $x \nleq y_1$. By definition $y_1 \nvDash p$. Hence $x \vDash \neg p$.

Also $x \nvDash q \vee \sim q$: as $x \leq x$, hence $x \nvDash q$. We also have an element y_2 in W such that $xR_\sim y_2$ and $y_2 \nleq x$. Hence $x \nvDash \sim q$. So $x \nvDash q \vee \sim q$.
Canonicity:
We show that the canonical frame $(W_c, R_c, \subseteq)$ satisfies (*).

So let $P \in W_c$ and suppose there exists a prime theory Q such that $(PR_{c\neg}Q \wedge P \nsubseteq Q)$. Let us show that $\forall Q'(PR_{c\sim}Q' \rightarrow Q' \subseteq P)$.

$P \nsubseteq Q$ means that there is α such that $\alpha \in P$ and $\alpha \notin Q$. $PR_{c\neg}Q$ implies the existence of a prime theory Q_1 such that PR_cQ_1 and $Q_1 \subseteq Q$. Hence $\alpha \notin Q_1$. So, $\alpha \notin P_\neg$, but then by definition of $P_\neg$, $\neg\alpha \in P$. Hence $\alpha \wedge \neg\alpha \in P$. By our assumption $\alpha \wedge \neg\alpha \vdash \beta \vee \sim\beta$. Hence for any formula β we have either $\beta \in P$ or

$\sim \beta \in P$. Now let $PR_{c\sim}Q'$, and let $\gamma \in Q'$. By definition, $PR_{c\sim}Q'$ implies that there is a prime theory Q'_1 such that $PR_cQ'_1$ and $Q'_1 \supseteq Q'$. Hence $\gamma \in Q'_1$. But $Q'_1 \subseteq P_\sim$. Then $\gamma \in P_\sim$ implies $\sim \gamma \notin P$. Hence $\gamma \in P$, and we get $Q' \subseteq P$. □

Let $\mathcal{L}_{RDSA}$ denote the logic which contains all axioms and postulates of the logic K_- along with the following.

(i) $\alpha \wedge \sim \alpha \vdash \bot$.
(ii) $\alpha \vee \neg\alpha \vdash \bot$.
(iii) $\dfrac{\alpha \wedge \beta \vdash \gamma}{\alpha \wedge \sim \gamma \leq \sim \beta}$.
(iv) $\dfrac{\gamma \vdash \alpha \vee \beta}{\neg\beta \vdash \alpha \vee \neg\gamma}$.
(v) $\top \vdash \sim \alpha \vee \sim\sim \alpha$.
(vi) $\neg\alpha \wedge \neg\neg\alpha \vdash \bot$.
(vii) $\alpha \wedge \neg\alpha \vdash \beta \vee \sim \beta$ (Regularity).

We call a K_- frame $(W, R, \leq)$ a *regular double Stone frame* if it satisfies the following first order conditions.

1. $\forall x \forall y(xR_\sim y \rightarrow yR_\sim x)$.
2. $\forall x \forall y(xR_\neg y \rightarrow yR_\neg x)$.
3. $\forall x(xR_\sim x)$.
4. $\forall x(xR_\neg x)$.
5. $\forall x \forall y_1 \forall y_2(xR_\sim y_1 \wedge xR_\sim y_2 \rightarrow (y_1R_\sim y_2 \wedge y_2R_\sim y_1))$.
6. $\forall x \forall y_1 \forall y_2(xR_\neg y_1 \wedge xR_\neg y_2 \rightarrow (y_1R_\neg y_2 \wedge y_2R_\neg y_1))$.
7. $\forall x((\forall y(xR_\neg y \rightarrow x \leq y)) \vee (\forall z(xR_\sim z \rightarrow z \leq x)))$.

Let us denote by $\mathbb{F}_{RDSA}$, the class of all regular double Stone frames, by $\mathcal{A}_{RDSA}$, the class of all regular double Stone algebras, and by $\mathcal{RS}_{RDSA}$, the class of regular double Stone algebras formed by rough sets over Pawlakian approximation spaces. Comer [14] has proved the representation result that given any regular double Stone algebra K, there is a regular double Stone algebra R formed by rough sets over some approximation space such that K can be embedded in R. Making use of this result, we obtain

Theorem 8. *For any $\alpha, \beta \in \mathcal{F}$, the following are equivalent.*

1. $\alpha \vdash_{\mathcal{L}_{RDSA}} \beta$.
2. $\alpha \models_{\mathcal{A}_{RDSA}} \beta$.
3. $\alpha \models_{\mathcal{RS}_{RDSA}} \beta$.
4. $\alpha \models_{\mathbb{F}_{RDSA}} \beta$.

The enhanced united kite of negations with Stone and dual Stone negations can be seen in Fig. 4.

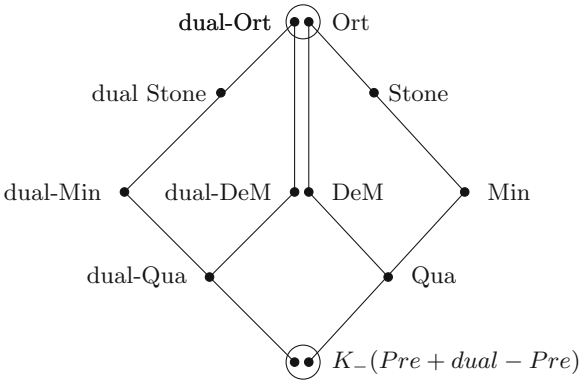


Fig. 4. Enhanced United Kite of Negations

5 Conclusions

We have investigated the semantics of some negations that appear in classical rough set-theoretic structures. It is shown that Stone (dual Stone) negation can be treated as an 'impossibility' ('unnecessity') operator, via the semantics in compatibility (exhaustive) frames. Further, a semantics for the logic of regular double Stone algebras in K_{-} frames is presented, which is equivalent to the already established rough set semantics for the logic.

Similar to the Jónsson-Tarski duality of modal logic, duality results can be proved between classes of (compatibility) frames and classes of various lattices with negation [2,3,8]. Such duality results for the classes of frames and algebras presented here can also be obtained [7]. Our next aim is to carry out a comprehensive semantic analysis as above, of negations appearing in generalized rough set theory. A preliminary study in [7] has shown that some of the operators that arise in the generalized framework occupy new positions in Dunn's kites of negations. So the investigation appears worth pursuing further.

References

1. Kripke, S.: Semantic analysis of intuitionistic logic I. In: Crossley, J., Dummett, M. (eds.) Formal Systems and Recursive Functions, pp. 92–129. North-Holland, Amsterdam (1963)
2. Dunn, J.: Star and Perp: two treatments of negation. In: Tomberlin, J. (ed.) Philosophical Perspectives, vol. 7, pp. 331–357. Ridgeview Publishing Company, Atascadero (1994)
3. Dunn, J.: Generalised ortho negation. In: Wansing, H. (ed.) Negation: A Notion in Focus, pp. 3–26. Walter de Gruyter, Berlin (1996)
4. Dunn, J.: A comparative study of various model-theoretic treatments of negation: a history of formal negations. In: Gabbay, D., Wansing, H. (eds.) What is Negation? pp. 23–51. Kluwer Academic Publishers, Netherlands (1999)

5. Pawlak, Z.: Rough sets. Int. J. Comput. Inf. Sci. **11**, 341–356 (1982)
6. Banerjee, M., Chakraborty, M.K.: Algebras from rough sets. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-Neuro Computing: Techniques for Computing with Words. Cognitive Technologies, pp. 157–184. Springer, Berlin (2004)
7. Kumar, A.: A study of algebras and logics of rough sets based on classical and generalized approximation spaces. Doctoral dissertation, Indian Institute of Technology, Kanpur (2016)
8. Dunn, J.: Negation in the context of gaggle theory. Stud. Logica **80**, 235–264 (2005)
9. Restall, G.: Defining double negation elimination. L. J. IGPL **8**(6), 853–860 (2000)
10. Dunn, J.: Positive modal logic. Stud. Logica **55**, 301–317 (1995)
11. Varlet, J.: A regular variety of type (2,2,1,1,0,0). Algebra Univ. **2**, 218–223 (1972)
12. Dai, J.-H.: Logic for rough sets with rough double stone algebraic semantics. In: Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 141–148. Springer, Heidelberg (2005). doi:10.1007/11548669_15
13. Banerjee, M., Khan, M.A.: Propositional logics from rough set theory. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 1–25. Springer, Heidelberg (2007). doi:10.1007/978-3-540-71200-8_1
14. Comer, S.: Perfect extensions of regular double Stone algebras. Algebra Univ. **34**(1), 96–109 (1995)

Achieving While Maintaining:
A Logic of Knowing How with Intermediate Constraints

Yanjun Li[1(✉)] and Yanjing Wang[2]

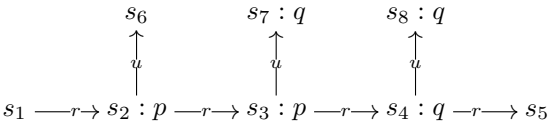
[1] Faculty of Philosophy, University of Groningen, Groningen, The Netherlands
Y.J.Li@rug.nl
[2] Department of Philosophy, Peking University, Beijing, China

Abstract. In this paper, we propose a ternary knowing how operator to express that the agent knows how to achieve φ given ψ while maintaining χ in-between. It generalizes the logic of goal-directed knowing how proposed by Wang in [10]. We give a sound and complete axiomatization of this logic.

1 Introduction

Standard epistemic logic proposed by von Wright and Hintikka studies propositional knowledge expressed by "knowing that φ" [6,9]. However, there are very natural knowledge expressions beyond "knowing that", such as "knowing what your password is", "knowing why he came late", "knowing how to go to Beijing", and so on. In recent years, there have been attempts to capture the logics of such different kinds of knowledge expressions by taking each "knowing X" as a single modality [2,3,5,10,13,14].[1]

In particular, Wang proposed a logical language of goal-directed knowing how [10], which includes formulas $\mathcal{K}h(\psi, \varphi)$ to express that the agent knows how to achieve φ given the precondition ψ.[2] The models are labeled transition systems which represent the agent's abilities, inspired by [11]. Borrowing the idea from conformant planning in AI (cf. e.g., [8,15]), $\mathcal{K}h(\psi, \varphi)$ holds globally in a labeled transition system, if there is an uniform plan such that from all the ψ-states this plan can always be successfully executed to reach some φ-states. As an example, in the following model $\mathcal{K}h(p, q)$ holds, since there is a plan ru which can always work to reach a q-state from any p-state.



[1] See [12] for a survey.
[2] See [1,4,10] for detailed discussions on related work in AI and Philosophy.

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 154–167, 2017.
DOI: 10.1007/978-3-662-54069-5_12

In [10], a sound and complete proof system is given, featuring a crucial axiom capturing the compositionality of plans:

$$\texttt{COMPKh} \qquad \mathcal{K}h(p,r) \wedge \mathcal{K}h(r,q) \rightarrow \mathcal{K}h(p,q)$$

However, as observed in [7], constraints on how we achieve the goal often matter. For example, the ways for me to go to New York are constrained by the money I have; we want to know how to win the game by playing fairly; people want to know how to be rich without breaking the law. Generally speaking, actions have costs, both financially and morally, we need to stay within our "budget" in reaching our goals. Clearly such intermediate constraints cannot be expressed by $\mathcal{K}h(\psi,\varphi)$ since it only cares about the starting and ending states. This motivates us to introduce a ternary modality $\mathcal{K}h(\psi,\chi,\varphi)$ where χ constrains the intermediate states.[3]

In the rest of the paper, we first introduce the language, semantics, and a proof system of our logic in Sect. 2. In Sect. 3 we give the highly non-trivial completeness proof of our system, which is much more complicated than the one for the standard knowing how logic. In the last section we conclude with future directions.

2 The Logic

Definition 1 (Language). *Given a set of proposition letters* $\mathbf{P}$*, the language* $\mathbf{L_{Khm}}$ *is defined as follows:*

$$\varphi := \top \mid p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \mathcal{K}hm(\varphi,\varphi,\varphi)$$

where $p \in \mathbf{P}$*.* $\mathcal{K}hm(\psi,\chi,\varphi)$ *expresses that the agent knows how to guarantee* φ *given* ψ *while maintaining* χ *in-between (excluding the start and the end). Note that* $\mathcal{K}hm(\psi\wedge\chi,\chi,\varphi\wedge\chi)$ *expresses knowing how with inclusive intermediate constraints. We use the standard abbreviations* $\bot, \varphi\vee\psi$ *and* $\varphi\rightarrow\psi$*, and define* $\mathcal{U}\varphi$ *as* $\mathcal{K}hm(\neg\varphi,\top,\bot)$*.* $\mathcal{U}$ *is intended to be an universal modality, and it will become more clear after defining the semantics. Note that the binary know-how operator in [11] can be defined as* $\mathcal{K}h(\psi,\varphi) := \mathcal{K}hm(\psi,\top,\varphi)$*.*

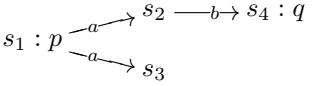
Definition 2 (Model). *Given a countable set of proposition letters* $\mathbf{P}$ *and a countable non-empty set of action symbols* $\mathbf{\Sigma}$*. A model (also called an ability map) is essentially a labelled transition system* $(\mathcal{S},\mathcal{R},\mathcal{V})$ *where:*

- $\mathcal{S}$ *is a non-empty set of states;*
- $\mathcal{R}:\mathbf{\Sigma}\rightarrow 2^{\mathcal{S}\times\mathcal{S}}$ *is a collection of transitions labelled by actions in* $\mathbf{\Sigma}$*;*
- $\mathcal{V}: S\rightarrow 2^{\mathbf{P}}$ *is a valuation function.*

[3] This ternary modality is first proposed and discussed briefly in the full version of [10], which is under submission.

We write $s \xrightarrow{a} t$ *if* $(s,t) \in \mathcal{R}(a)$. *For a sequence* $\sigma = a_1 \dots a_n \in \mathbf{\Sigma}^*$, *we write* $s \xrightarrow{\sigma} t$ *if there exist* $s_2 \dots s_n$ *such that* $s \xrightarrow{a_1} s_2 \xrightarrow{a_2} \cdots \xrightarrow{a_{n-1}} s_n \xrightarrow{a_n} t$. *Note that* σ *can be the empty sequence* ϵ *(when* $n = 0$*), and we set* $s \xrightarrow{\epsilon} s$ *for any* s. *Let* σ_k *be the initial segment of* σ *up to* a_k *for* $k \leq |\sigma|$. *In particular let* $\sigma_0 = \epsilon$. *We say* $\sigma = a_1 \cdots a_n$ *is strongly executable at* s' *if for each* $0 \leq k < n$: $s' \xrightarrow{\sigma_k} t$ *implies that* t *has at least one* a_{k+1}*-successor.*

Intuitively, σ is *strongly executable* at s if you can always successfully finish the whole σ after executing any initial segment of σ from s. For example, ab is not strongly executable at s_1 in the model below, though it is executable at s_1.



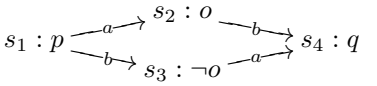
Definition 3 (Semantics). *Suppose* s *is a state in a model* $\mathcal{M} = (\mathcal{S}, \mathcal{R}, \mathcal{V})$. *Then we inductively define the notion of a formula* φ *being satisfied (or true) in* $\mathcal{M}$ *at state* s *as follows:*

$\mathcal{M}, s \vDash \top$		*always*
$\mathcal{M}, s \vDash p$	$\iff$	$s \in \mathcal{V}(p)$
$\mathcal{M}, s \vDash \neg\varphi$	$\iff$	$\mathcal{M}, s \nvDash \varphi$
$\mathcal{M}, s \vDash \varphi \wedge \psi$	$\iff$	$\mathcal{M}, s \vDash \varphi$ *and* $\mathcal{M}, s \vDash \psi$
$\mathcal{M}, s \vDash \mathcal{K}hm(\psi, \chi, \varphi)$	$\iff$	*there exists* $\sigma \in \mathbf{\Sigma}^*$ *such that for each* s' *with* $\mathcal{M}, s' \vDash \psi$ *we have* σ *is strongly* χ*-executable at* s' *and* $\mathcal{M}, t \vDash \varphi$ *for all* t *with* $s' \xrightarrow{\sigma} t$.

where we say $\sigma = a_1 \cdots a_n$ *is strongly* χ*-executable at* s' *if:*

- σ *is strongly executable at* s', *and*
- $s' \xrightarrow{\sigma_k} t$ *implies* $\mathcal{M}, t \vDash \chi$ *for all* $0 < k < n$.

It is obvious that ϵ is strongly χ-executable at each state s for each formula χ. Note that $\mathcal{K}hm(\psi, \bot, \varphi)$ expresses that there is $\sigma \in \mathbf{\Sigma} \cup \{\epsilon\}$ such that the agent knows doing σ on ψ-states can guarantee φ, namely the witness plan σ is at most one-step. As an example, $\mathcal{K}h(p, \bot, o)$ and $\mathcal{K}h(p, o, q)$ hold in the following model for the witness plans a and ab respectively. Note that the truth value of $\mathcal{K}h(\psi, \chi, \varphi)$ does not depend on the designated state.



Now we can also check that the operator $\mathcal{U}$ defined by $\mathcal{K}hm(\neg\psi, \top, \bot)$ is indeed an *universal modality*:

$$\mathcal{M}, s \vDash \mathcal{U}\varphi \Leftrightarrow \text{ for all } t \in \mathcal{S}, \mathcal{M}, t \vDash \varphi$$

The following formulas are valid on all models.

Proposition 1. $\vDash \mathcal{U}(p \to q) \to \mathcal{K}hm(p, \bot, q)$

Proof. Assuming that $\mathcal{M}, s \vDash \mathcal{U}(p \to q)$, it means that $\mathcal{M}, t \vDash p \to q$ for all $t \in \mathcal{S}$. Given $\mathcal{M}, t \vDash p$, it follows that $\mathcal{M}, t \vDash q$. Thus, we have ϵ is strongly $\bot$-executable at t. Therefore, we have $\mathcal{M}, s \vDash \mathcal{K}hm(p, \bot, q)$. □

Proposition 2. $\vDash \mathcal{K}hm(p, o, r) \wedge \mathcal{K}hm(r, o, q) \wedge \mathcal{U}(r \to o) \to \mathcal{K}hm(p, o, q)$

Proof. Assuming $\mathcal{M}, s \vDash \mathcal{K}hm(p, o, r) \wedge \mathcal{K}hm(r, o, q) \wedge \mathcal{U}(r \to o)$, we will show that $\mathcal{M}, s \vDash \mathcal{K}hm(p, o, q)$. Since $\mathcal{M}, s \vDash \mathcal{K}hm(p, o, r)$, it follows that there exists $\sigma \in \mathbf{\Sigma}^*$ such that for each $\mathcal{M}, u \vDash p$, σ is strongly o-executable at u and that $\mathcal{M}, v \vDash r$ for each v with $u \xrightarrow{\sigma} v$. Since $\mathcal{M}, s \vDash \mathcal{K}hm(r, o, q)$, it follows that there exists $\sigma' \in \mathbf{\Sigma}^*$ such that for each $\mathcal{M}, v' \vDash r$, σ' is strongly o-executable at v' and that $\mathcal{M}, t \vDash q$ for each t with $v' \xrightarrow{\sigma} t$. In order to show $\mathcal{M}, s \vDash \mathcal{K}hm(p, o, q)$, we only need to show that $\sigma\sigma'$ is strongly o-executable at u and that $\mathcal{M}, t' \vDash q$ for each t' with $u \xrightarrow{\sigma\sigma'} t'$, where u is a state with $\mathcal{M}, u \vDash p$.

By assumption, we know that σ is strongly o-executable at u, and for each v with $u \xrightarrow{\sigma} v$, it follows by assumption that $\mathcal{M}, v \vDash r$ and σ' is strongly o-executable at v. Moreover, since $\mathcal{M}, s \vDash \mathcal{U}(r \to o)$, it follows that $\mathcal{M}, v \vDash o$ for each v with $u \xrightarrow{\sigma} v$. Thus, $\sigma\sigma'$ is strongly o-executable at u. What is more, for each t' with $u \xrightarrow{\sigma\sigma'} t'$, there is v such that $u \xrightarrow{\sigma} v \xrightarrow{\sigma'} t'$ and $\mathcal{M}, v \vDash r$, it follows by assumption that $\mathcal{M}, t' \vDash q$. Therefore, we have $\mathcal{M}, s \vDash \mathcal{K}hm(p, o, q)$. □

Proposition 3. $\vDash \mathcal{K}hm(p, o, q) \wedge \neg\mathcal{K}hm(p, \bot, q) \to \mathcal{K}hm(p, \bot, o)$

Proof. Assuming $\mathcal{M}, s \vDash \mathcal{K}hm(p, o, q) \wedge \neg\mathcal{K}hm(p, \bot, q)$, we will show that $\mathcal{M}, s \vDash \mathcal{K}hm(p, \bot, o)$. Since $\mathcal{M}, s \vDash \mathcal{K}hm(p, o, q)$, it follows that there exists $\sigma \in \mathbf{\Sigma}^*$ such that for each $\mathcal{M}, u \vDash p$, σ is strongly o-executable at u and $\mathcal{M}, v \vDash q$ for all v with $u \xrightarrow{\sigma} v$. If $\sigma \in \mathbf{\Sigma} \cup \{\epsilon\}$, it follows that $\mathcal{M}, s \vDash \mathcal{K}hm(p, \bot, q)$. Since $\mathcal{M}, s \vDash \neg\mathcal{K}hm(p, \bot, q)$, it follows that $\sigma \notin \mathbf{\Sigma} \cup \{\epsilon\}$. Thus, $\sigma = a_1 \cdots a_n$ where $n \geq 2$. Let u be a state such that $\mathcal{M}, u \vDash p$. Since $\sigma = a_1 \cdots a_n$ is strongly o-executable at u, it follows that a_1 is executable at u. Moreover, since $n \geq 2$, we have $\mathcal{M}, v \vDash o$ for each v with $u \xrightarrow{a_1} v$. Therefore, we have $\mathcal{M}, s \vDash \mathcal{K}hm(p, \bot, o)$. □

Proposition 4. $\vDash \mathcal{U}(p' \to p) \wedge \mathcal{U}(o \to o') \wedge \mathcal{U}(q \to q') \wedge \mathcal{K}hm(p, o, q) \to \mathcal{K}hm(p', o', q')$

Proof. Assuming $\mathcal{M}, s \vDash \mathcal{U}(p' \to p) \wedge \mathcal{U}(o \to o') \wedge \mathcal{U}(q \to q') \wedge \mathcal{K}hm(p, o, q)$, we will show $\mathcal{M}, s \vDash \mathcal{K}hm(p', o', q')$. Since $\mathcal{M}, s \vDash \mathcal{K}hm(p, o, q)$, it follows that there exists $\sigma \in \mathbf{\Sigma}^*$ such that for each $\mathcal{M}, u \vDash p$: σ is strongly o-executable at u and $\mathcal{M}, v \vDash q$ for each v with $u \xrightarrow{\sigma} v$. Let s' be a state with $\mathcal{M}, s' \vDash p'$. Next we will show that σ is strongly o'-executable at s' and $\mathcal{M}, v' \vDash q'$ for all v' with $s' \xrightarrow{\sigma} v'$.

Since $\mathcal{M}, s \vDash \mathcal{U}(p' \to p)$, it follows that $\mathcal{M}, s' \vDash p$. Thus, σ is strongly o-executable at s' and $\mathcal{M}, v' \vDash q$ for each v' with $s' \xrightarrow{\sigma} v'$. Since $\mathcal{M}, s \vDash \mathcal{U}(o \to o')$, it follows that σ is strongly o'-executable at s'. Since $\mathcal{M}, s \vDash \mathcal{U}(q \to q')$, it follows that $\mathcal{M}, v' \vDash q'$ for each v' with $s' \xrightarrow{\sigma} v'$. □

Definition 4 (Deductive System $\mathbb{SKHM}$). *The axioms and rules shown in Table 1 constitutes the proof system* $\mathbb{SKHM}$.

Note that DISTU, NECU, TU are standard for the universal modality $\mathcal{U}$. 4KhmU and 4KhmU are introspection axioms reflecting that $\mathcal{K}hm$ formulas are global. EMPKhm captures the interaction between $\mathcal{U}$ and $\mathcal{K}hm$ via the empty plan. COMPKhm is the new composition axiom for $\mathcal{K}hm$. UKhm shows how we can weaken the knowing how claims. ONEKhm is the characteristic axiom for $\mathbb{SKHM}$ compared to the system for binary $\mathcal{K}h$, and it expresses the condition for the necessity of the intermediate steps.

Table 1. System $\mathbb{SKHM}$

Axioms	
TAUT	all tautologies of propositional logic
DISTU	$\mathcal{U}p \wedge \mathcal{U}(p \to q) \to \mathcal{U}q$
TU	$\mathcal{U}p \to p$
4KhmU	$\mathcal{K}hm(p,o,q) \to \mathcal{U}\mathcal{K}hm(p,o,q)$
5KhmU	$\neg\mathcal{K}hm(p,o,q) \to \mathcal{U}\neg\mathcal{K}hm(p,o,q)$
EMPKhm	$\mathcal{U}(p \to q) \to \mathcal{K}hm(p,\bot,q)$
COMPKhm	$\mathcal{K}hm(p,o,r) \wedge \mathcal{K}hm(r,o,q) \wedge \mathcal{U}(r \to o) \to \mathcal{K}hm(p,o,q)$
ONEKhm	$\mathcal{K}hm(p,o,q) \wedge \neg\mathcal{K}hm(p,\bot,q) \to \mathcal{K}hm(p,\bot,o)$
UKhm	$\mathcal{U}(p' \to p) \wedge \mathcal{U}(o \to o') \wedge \mathcal{U}(q \to q') \wedge \mathcal{K}hm(p,o,q) \to \mathcal{K}hm(p',o',q')$
Rules	
	MP $\dfrac{\varphi, \varphi \to \psi}{\psi}$ NECU $\dfrac{\varphi}{\mathcal{U}\varphi}$ SUB $\dfrac{\varphi(p)}{\varphi[\psi/p]}$

Remark 1. Note that the corresponding axioms for COMPKhm, EMPKhm and UKhm in the setting of binary $\mathcal{K}h$ are the following:[4]

COMPKh	$\mathcal{K}h(p,q) \wedge \mathcal{K}h(q,r) \to \mathcal{K}h(p,r)$
EMPKh	$\mathcal{U}(p \to q) \to \mathcal{K}h(p,q)$
UKh	$\mathcal{U}(p' \to p) \wedge \mathcal{U}(q \to q') \wedge \mathcal{K}h(p,q) \to \mathcal{K}h(p',q')$

In the system $\mathbb{SKH}$ of [10] UKh can be derived using COMPKh and EMPKh. However, UKhm cannot be derived using COMPKhm and EMPKhm. In particular, $\mathcal{K}hm(p',\bot,p) \wedge \mathcal{K}hm(p,o,q) \to \mathcal{K}hm(p',o,q)$ is not valid due to the lack of $\mathcal{U}(p \to o)$, in contrast with the $\mathbb{SKH}$-derivable $\mathcal{K}h(p',p) \wedge \mathcal{K}h(p,q) \to \mathcal{K}h(p',q)$ which is crucial in the derivation of UKh in $\mathbb{SKH}$.

[4] We can obtain the corresponding axioms by taking the intermediate constraint as $\top$. Note that in [10], we use the name WKKh for UKh.

Since $\mathcal{U}$ is an universal modality, DISTU and TU are obviously valid. Due to the fact that the modality $\mathcal{K}hm$ is not local, it is easy to show that 4KhmU and 5KhmU are valid. Moreover, by Propositions 1–4, we have that all axioms are valid. Due to a standard argument in modal logic, we know that the rules MP, NECU and SUB preserve formula's validity. The soundness of $\mathbb{SKHM}$ follows immediately.

Theorem 1. $\mathbb{SKHM}$ *is sound w.r.t. the class of all models.*

Below we derive some theorems and rules that are useful in the later proofs.

Proposition 5. *We can derive the following in* $\mathbb{SKHM}$*:*

4U	$\mathcal{U}p \to \mathcal{U}\mathcal{U}p$
5U	$\neg\mathcal{U}p \to \mathcal{U}\neg\mathcal{U}p$
ULKhm	$\mathcal{U}(p' \to p) \wedge \mathcal{K}hm(p, o, q) \to \mathcal{K}hm(p', o, q)$
UMKhm	$\mathcal{U}(o \to o') \wedge \mathcal{K}hm(p, o, q) \to \mathcal{K}hm(p, o', q)$
URKhm	$\mathcal{U}(q \to q') \wedge \mathcal{K}hm(p, o, q') \to \mathcal{K}hm(p, o, q')$
UNIV	$\mathcal{U}\neg p \to \mathcal{K}hm(p, \bot, \bot)$
REU	*from* $\varphi \leftrightarrow \psi$ *prove* $\mathcal{U}\varphi \leftrightarrow \mathcal{U}\psi$
RE	*from* $\varphi \leftrightarrow \psi$ *prove* $\chi \leftrightarrow \chi'$
	where χ' *is obtained by replacing some occurrences of* φ *in* χ *by* ψ*.*

Proof. REU is immediate given DISTU and NECU. 4U and 5U are special cases of 4KhmU and 5KhmU respectively. ULKhm, UMKhm, URKhm are the special cases of UKhm. To prove UNIV, first note that $\mathcal{U}\neg p \leftrightarrow \mathcal{U}(p \to \bot)$ due to REU. Then due to EMPKhm, we have $\mathcal{U}\neg p \to \mathcal{K}hm(p, \bot, \bot)$. RE can be obtained by an inductive proof on the shape of χ, which uses UKhm and NECU for the case of $\mathcal{K}hm(\cdot, \cdot, \cdot)$. □

3 Completeness

This section will prove that $\mathbb{SKHM}$ is complete w.r.t. the class of all models. The key is to build a canonical model based on a fixed maximal consistent set, just as in [10]. However, the canonical model here is much more complicated. Firstly, the state of the canonical model is a pair consisting of a maximal consistent set and a marker which will play an important role in defining the witness plan for $\mathcal{K}hm$-formulas. Secondly, different from the canonical model in [10] where each formula of the form $\mathcal{K}h(\psi, \varphi)$ is realized by an one-step witness plan, some $\mathcal{K}hm(\psi, \chi, \varphi)$ formulas here have to be realized by a two-step witness plan, and the intermediate states need to satisfy χ.

Here are some notions before we prove the completeness. Given a set of $\mathbf{L_{Khm}}$ formulas Δ, let $\Delta|_{\mathcal{K}hm}$ and $\Delta|_{\neg\mathcal{K}hm}$ be the collections of its positive and negative $\mathcal{K}hm$ formulas:

$$\Delta|_{\mathcal{K}hm} = \{\theta \mid \theta = \mathcal{K}hm(\psi, \chi, \varphi) \in \Delta\};$$
$$\Delta|_{\neg\mathcal{K}hm} = \{\theta \mid \theta = \neg\mathcal{K}hm(\psi, \chi, \varphi) \in \Delta\}.$$

In the following, let Γ be a maximal consistent set (MCS) of $\mathbf{L_{Khm}}$ formulas. We first prepare ourselves with some handy propositions.

Definition 5. *Let Φ_Γ be the set of all MCS Δ such that $\Delta|_{\mathcal{K}hm} = \Gamma|_{\mathcal{K}hm}$.*

Since every $\Delta \in \Phi_\Gamma$ is maximal consistent it follows immediately that:

Proposition 6. *For each $\Delta \in \Phi_\Gamma$, we have $\mathcal{K}hm(\psi, \chi, \varphi) \in \Gamma$ if and only if $\mathcal{K}hm(\psi, \chi, \varphi) \in \Delta$ for all $\mathcal{K}hm(\psi, \chi, \varphi) \in \mathbf{L_{Khm}}$.*

Proposition 7. *If $\varphi \in \Delta$ for all $\Delta \in \Phi_\Gamma$ then $\mathcal{U}\varphi \in \Delta$ for all $\Delta \in \Phi_\Gamma$.*

Proof. Suppose $\varphi \in \Delta$ for all $\Delta \in \Phi_\Gamma$, then by the definition of Φ_Γ, $\neg\varphi$ is not consistent with $\Gamma|_{\mathcal{K}hm} \cup \Gamma|_{\neg\mathcal{K}hm}$, for otherwise $\Gamma|_{\mathcal{K}hm} \cup \Gamma|_{\neg\mathcal{K}hm} \cup \{\neg\varphi\}$ can be extended into a maximal consistent set in Φ_Γ due to a standard Lindenbaum-like argument. Thus there are $\mathcal{K}hm(\psi_1, \chi_1, \varphi_1), \ldots, \mathcal{K}hm(\psi_k, \chi_k, \varphi_k) \in \Gamma|_{\mathcal{K}hm}$ and $\neg\mathcal{K}hm(\psi'_1, \chi'_1, \varphi'_1), \ldots, \neg\mathcal{K}hm(\psi'_l, \chi'_l, \varphi'_l) \in \Gamma|_{\neg\mathcal{K}hm}$ such that

$$\vdash \bigwedge_{1\le i\le k} \mathcal{K}hm(\psi_i, \chi_i, \varphi_i) \wedge \bigwedge_{1\le j\le l} \neg\mathcal{K}hm(\psi'_j, \chi'_j, \varphi'_j) \to \varphi.$$

By `NECU`,

$$\vdash \mathcal{U}(\bigwedge_{1\le i\le k} \mathcal{K}hm(\psi_i, \chi_i, \varphi_i) \wedge \bigwedge_{1\le j\le l} \neg\mathcal{K}hm(\psi'_j, \chi'_j, \varphi'_j) \to \varphi).$$

By `DISTU` we have:

$$\vdash \mathcal{U}(\bigwedge_{1\le i\le k} \mathcal{K}hm(\psi_i, \chi_i, \varphi_i) \wedge \bigwedge_{1\le j\le l} \neg\mathcal{K}hm(\psi'_j, \chi'_j, \varphi'_j)) \to \mathcal{U}\varphi.$$

Since $\mathcal{K}hm(\psi_1, \chi_1, \varphi_1), \ldots, \mathcal{K}hm(\psi_k, \chi_k, \varphi_k) \in \Gamma$, we have $\mathcal{U}\mathcal{K}hm(\psi_1, \chi_1, \varphi_1), \ldots, \mathcal{U}\mathcal{K}hm(\psi_k, \chi_k, \varphi_k) \in \Gamma$ due to `4KhmU` and the fact that Γ is a maximal consistent set. Similarly, we have $\mathcal{U}\neg\mathcal{K}hm(\psi'_1, \chi'_1, \varphi'_1), \ldots, \mathcal{U}\neg\mathcal{K}hm(\psi'_l, \chi'_l, \varphi'_l) \in \Gamma$ due to `5KhmU`. By `DISTU` and `NECU`, it is easy to show that $\vdash \mathcal{U}(p \wedge q) \leftrightarrow \mathcal{U}p \wedge \mathcal{U}q$. Then due to a slight generalization, we have:

$$\mathcal{U}(\bigwedge_{1\le i\le k} \mathcal{K}hm(\psi_i, \chi_i, \varphi_i) \wedge \bigwedge_{1\le j\le l} \neg\mathcal{K}hm(\psi'_j, \chi'_j, \varphi'_j)) \in \Gamma.$$

Now it is immediate that $\mathcal{U}\varphi \in \Gamma$. Due to Proposition 6, $\mathcal{U}\varphi \in \Delta$ for all $\Delta \in \Phi_\Gamma$. □

Proposition 8. *Given $\mathcal{K}hm(\psi, \top, \varphi) \in \Gamma$ and $\Delta \in \Phi_\Gamma$, if $\psi \in \Delta$ then there exists $\Delta' \in \Phi_\Gamma$ such that $\varphi \in \Delta'$.*

Proof. Assuming $\mathcal{K}hm(\psi, \top, \varphi) \in \Gamma$ and $\psi \in \Delta \in \Phi_\Gamma$, if there does not exist $\Delta' \in \Phi_\Gamma$ such that $\varphi \in \Delta'$, it means that $\neg\varphi \in \Delta'$ for all $\Delta' \in \Phi_\Gamma$. It follows by Proposition 7 that $\mathcal{U}\neg\varphi \in \Gamma$, namely $\mathcal{K}hm(\varphi, \top, \bot) \in \Gamma$. Since $\mathcal{U}(\varphi \to \bot)$ and $\mathcal{K}hm(\psi, \top, \varphi) \in \Gamma$, it follows by `COMPKhm` that $\mathcal{K}hm(\psi, \top, \bot) \in \Gamma$ namely, $\mathcal{U}\neg\psi \in \Gamma$. By Proposition 6, we have that $\mathcal{U}\neg\psi \in \Delta$. It follows by `TU` that $\neg\psi \in \Delta$. This is contradictory with $\psi \in \Delta$. Therefore, there exists $\Delta' \in \Phi_\Gamma$ such that $\varphi \in \Delta'$. □

Definition 6. *Let the set of action symbols* $\mathbf{\Sigma}_\Gamma$ *be defined as* $\mathbf{\Sigma}_\Gamma = \{\langle \psi, \bot, \varphi\rangle \mid \mathcal{K}hm(\psi, \bot, \varphi) \in \Gamma\} \cup \{\langle \chi^\psi, \varphi\rangle \mid \mathcal{K}hm(\psi, \chi, \varphi), \neg\mathcal{K}hm(\psi, \bot, \varphi) \in \Gamma\}$.

The later part of $\mathbf{\Sigma}_\Gamma$ is to handle the cases where the intermediate state is indeed necessary: $\neg\mathcal{K}hm(\psi, \bot, \varphi)$ makes sure that you cannot have a plan to guarantee φ in less than two steps.

In the following we build a separate canonical model for each MCS Γ, for it is not possible to satisfy all of $\mathcal{K}hm$ formulas simultaneously in a single model since they are global. Because the later proofs are quite technical, it is very important to first understand the ideas behind the canonical model construction. Note that to satisfy a $\mathcal{K}hm(\psi, \chi, \varphi)$ formula, there are two cases to be considered:

(1) $\mathcal{K}hm(\psi, \bot, \varphi)$ holds and we just need an one-step witness plan, which can be handled similarly using the techniques developed in [10];

(2) $\mathcal{K}hm(\psi, \bot, \varphi)$ does not hold, and we need to have a witness plan which at least involves an intermediate χ-stage. By $\texttt{ONEKhm}$, $\mathcal{K}hm(\psi, \bot, \chi)$ holds. It is then tempting to reduce $\mathcal{K}hm(\psi, \chi, \varphi)$ to $\mathcal{K}hm(\psi, \bot, \chi) \wedge \mathcal{K}hm(\chi, \chi, \varphi)$. However, it is not correct since we may not have a strongly χ-executable plan to make sure φ from *any* χ-state. Note that $\mathcal{K}hm(\psi, \chi, \varphi)$ and $\mathcal{K}hm(\psi, \bot, \chi)$ only make sure we can start from *certain* χ-states that result from the witness plan for $\mathcal{K}hm(\psi, \bot, \chi)$. However, we cannot refer to such χ-states in the language of $\mathbf{L_{Khm}}$. This is why we include χ^ψ markers in the building blocks of the canonical model besides maximal consistent set. χ^ψ roughly tells us where does this state "comes from".[5]

Definition 7 (Canonical Model). *The canonical model for* Γ *is a tuple* $\mathcal{M}^c_\Gamma = \langle \mathcal{S}^c, \mathcal{R}^c, \mathcal{V}^c\rangle$ *where:*

- $\mathcal{S}^c = \{(\Delta, \chi^\psi) \mid \chi \in \Delta \in \Phi_\Gamma$, *and* $\langle \chi^\psi, \varphi\rangle \in \mathbf{\Sigma}_\Gamma$ *for some* φ *or* $\langle \psi, \bot, \chi\rangle \in \mathbf{\Sigma}_\Gamma\}$. *We write the pair in* $\mathcal{S}$ *as* $w, v, \cdots$, *and refer to the first entry of* $w \in \mathcal{S}$ *as* $\mathtt{L}(w)$, *to the second entry as* $\mathtt{R}(w)$;
- $w \xrightarrow{\langle \psi, \bot, \varphi\rangle}_c w'$ *iff* $\psi \in \mathtt{L}(w)$ *and* $\mathtt{R}(w') = \varphi^\psi$;
- $w \xrightarrow{\langle \chi^\psi, \varphi\rangle}_c w'$ *iff* $\mathtt{R}(w) = \chi^\psi$ *and* $\varphi \in \mathtt{L}(w')$;
- $p \in \mathcal{V}^c(w)$ *iff* $p \in \mathtt{L}(w)$.

For each $w \in \mathcal{S}$, *we also call* w *a* ψ*-state if* $\psi \in \mathtt{L}(w)$.

In the above definition, $\mathtt{R}(w)$ marks the use of w as an intermediate state. The same maximal consistent set Δ may have different uses depending on different $\mathtt{R}(w)$. We will make use of the transitions $w \xrightarrow{\langle \psi, \bot, \chi\rangle}_c v \xrightarrow{\langle \chi^\psi, \varphi\rangle}_c w'$ where $\mathtt{R}(v) = \chi^\psi$. Note that if $\mathtt{R}(w) = \chi^\psi$ then $w \xrightarrow{\langle \chi^\psi, \varphi\rangle}_c v$ for each φ-state v. The highly non-trivial part of the later proof of the truth lemma is to show adding such transitions and making them to be composed arbitrarily will not cause some $\mathcal{K}hm(\psi, \chi, \varphi) \notin \mathtt{L}(w)$ to hold at w.

We first show that each $\Delta \in \Phi_\Gamma$ appears as $\mathtt{L}(w)$ for some $w \in \mathcal{S}^c$.

[5] In [10], the canonical models are much simpler: we just need MCSs and the canonical relations are simply labeled by $\langle \psi, \varphi\rangle$ for $\mathcal{K}h(\psi, \varphi) \in \Gamma$.

Proposition 9. *For each $\Delta \in \Phi_\Gamma$, there exists $w \in \mathcal{S}^c$ such that $\mathtt{L}(w) = \Delta$.*

Proof. Since $\vdash \top \to \top$, it follows by NECU that $\vdash \mathcal{U}(\top \to \top)$. Thus, we have $\mathcal{U}(\top \to \top) \in \Gamma$. It follows by EMPKhm that $\mathcal{K}hm(\top, \bot, \top) \in \Gamma$. It follows that $a = \langle \top, \bot, \top \rangle \in \mathbf{\Sigma}_\Gamma$. Since $\top \in \Delta$, it follows that $(\Delta, \top^\top) \in \mathcal{S}^c$. □

Since $\Gamma \in \Phi_\Gamma$, it follows by Proposition 9 that $\mathcal{S}^c \neq \emptyset$.

Proposition 7 helps us to prove the following two handy propositions which will play crucial roles in the completeness proof. Note that according to Proposition 7, to obtain that $\mathcal{U}\varphi$ in all the $\Delta \in \Phi_\Gamma$, we just need to show that φ is in all the $\Delta \in \Phi_\Gamma$, not necessarily in all the $w \in \mathcal{S}^c$.

Proposition 10. *Given $a = \langle \psi', \bot, \varphi' \rangle \in \mathbf{\Sigma}_\Gamma$, If for each ψ-state $w \in \mathcal{S}^c$ we have that a is executable at w, then $\mathcal{U}(\psi \to \psi') \in \Gamma$.*

Proof. Suppose that every ψ-state has an outgoing a-transition, then by the definition of $\mathcal{R}^c$, ψ' is in all the ψ-states. For each $\Delta \in \Phi_\Gamma$, either $\psi \notin \Delta$, or $\psi \in \Delta$ thus $\psi' \in \Delta$. Now by the fact that Δ is maximally consistent it is not hard to show $\psi \to \psi' \in \Delta$ in both cases. By Proposition 7, $\mathcal{U}(\psi \to \psi') \in \Delta$ for all $\Delta \in \Phi_\Gamma$. It follows by $\Gamma \in \Phi_\Gamma$ that $\mathcal{U}(\psi \to \psi') \in \Gamma$. □

Proposition 11. *Given $w \in \mathcal{S}^c$ and $a = \langle \psi, \bot, \varphi' \rangle$ or $\langle \chi^\psi, \varphi' \rangle \in \mathbf{\Sigma}_\Gamma$ such that a is executable at w, if $\varphi \in \mathtt{L}(w')$ for each w' with $w \xrightarrow{a} w'$ then $\mathcal{U}(\varphi' \to \varphi) \in \Gamma$.*

Proof. Firstly, we focus on the case of $a = \langle \psi, \bot, \varphi' \rangle$. For each $\Delta \in \Phi_\Gamma$ with $\varphi' \in \Delta$, we have $v = (\Delta, \varphi'^\psi) \in \mathcal{S}^c$. Since $\langle \psi, \bot, \varphi' \rangle$ is executable at w, it means that $\psi \in \mathtt{L}(w)$. By the definition, it follows that $w \xrightarrow{a} v$. Since $\varphi \in \mathtt{L}(w')$ for each w' with $w \xrightarrow{a} w'$, it follows that $\varphi \in \mathtt{L}(v)$. Therefore, we have $\varphi \in \Delta$ for each $\Delta \in \Phi_\Gamma$ with $\varphi' \in \Delta$, namely $\varphi' \to \varphi \in \Delta$ for all $\Delta \in \Phi_\Gamma$. It follows by Proposition 7 that $\mathcal{U}(\varphi' \to \varphi) \in \Gamma$.

Secondly, we focus on the case of $a = \langle \chi^\psi, \varphi' \rangle$. For each $\Delta \in \Phi_\Gamma$ with $\varphi' \in \Delta$, it follows by Proposition 9 that there exists $v \in \mathcal{S}^c$ such that $\mathtt{L}(v) = \Delta$. Since a is executable at w, it follows that $w \xrightarrow{a} v$. Since $\varphi \in \mathtt{L}(w')$ for each w' with $w \xrightarrow{a} w'$, it follows that $\varphi \in \mathtt{L}(v)$. Therefore, we have shown that $\varphi' \in \Delta$ implies $\varphi \in \Delta$ for all $\Delta \in \Phi_\Gamma$. It follows by Proposition 7 that $\mathcal{U}(\varphi' \to \varphi) \in \Gamma$. □

Before proving the truth lemma, we first need a handy result.

Proposition 12. *Given a non-empty sequence $\sigma = a_1 \cdots a_n \in \mathbf{\Sigma}_\Gamma^*$ where $a_i = \langle \psi_i, \bot, \varphi_i \rangle$ or $a_i = \langle \chi_i^{\psi_i}, \varphi_i \rangle$ for each $1 \leq i \leq n$, we have $\mathcal{K}hm(\psi, \chi, \varphi_i) \in \Gamma$ for all $1 \leq i \leq n$ if for each ψ-state $w \in \mathcal{S}^c$:*

- *σ is strongly executable at w;*
- *$w \xrightarrow{\sigma_j} t'$ implies $\chi \in \mathtt{L}(t')$ for all $1 \leq j < n$.*

Proof. If there is no ψ-state in $\mathcal{S}^c$, it follows that $\neg\psi \in \mathtt{L}(w')$ for each $w' \in \mathcal{S}^c$. It follows by Proposition 9 that $\neg\psi \in \Delta$ for all $\Delta \in \Phi_\Gamma$. By Proposition 7, we have $\mathcal{U}\neg\psi \in \Gamma$. By UNIV, $\mathcal{K}hm(\psi, \bot, \bot) \in \Gamma$. Since $\vdash \bot \to \chi$ and $\vdash \bot \to \varphi$.

Then by NECU, we have $\vdash \mathcal{U}(\bot \to \chi)$ and $\vdash \mathcal{U}(\bot \to \varphi)$. By UMKhm and URKhm, it is obvious that $\mathcal{K}hm(\psi, \chi, \varphi) \in \Gamma$.

Next, assuming $v \in \mathcal{S}^c$ is a ψ-state, we will show $\mathcal{K}hm(\psi, \chi, \varphi) \in \Gamma$. There are two cases: $n = 1$ or $n \geq 2$. For the case of $n = 1$, we will prove it directly; for the case of $n \geq 2$, we will prove it by induction on i.

- $n = 1$. If a_1 is in the form of $\langle \chi_1^{\psi_1}, \varphi_1 \rangle$, by the definition of $\xrightarrow{\langle \chi_1^{\psi_1}, \varphi_1 \rangle}$ it follows that $\mathtt{R}(w) = \chi_1^{\psi_1}$ for each ψ-state w. Let χ_0 be a formula satisfying that $\vdash \chi_0 \leftrightarrow \chi_1$ and $\chi_0 \neq \chi_1$. By the rule of Replacement of Equals RE, it follows that $\langle \chi_0^{\psi_1}, \varphi_1 \rangle \in \mathbf{\Sigma}_\Gamma$. Let $w' = (\mathtt{L}(v), \chi_0^{\psi_1})$ then it follows that $w' \in \mathcal{S}^c$. Since $\psi \in \mathtt{L}(v)$, then we have $\psi \in \mathtt{L}(w')$. However, since $\mathtt{R}(w') = \chi_1^{\psi_1} \neq \chi_0^{\psi_1}$, $\sigma = \langle \chi_1^{\psi_1}, \varphi_1 \rangle$ is not executable at the ψ-state w', contradicting the assumption that σ is strongly executable at all ψ-states. Therefore, we know that a_1 cannot be in the form of $\langle \chi_1^{\psi_1}, \varphi_1 \rangle$.

  If $a_1 = \langle \psi_1, \bot, \varphi_1 \rangle$, it follows that $\mathcal{K}hm(\psi_1, \bot, \varphi_1) \in \Gamma$. Since a_1 is executable at each ψ-state, it follows by Proposition 10 that $\mathcal{U}(\psi \to \psi_1) \in \Gamma$. Since $\mathcal{K}hm(\psi_1, \bot, \varphi_1) \in \Gamma$, it follows by ULKhm that $\mathcal{K}hm(\psi, \bot, \varphi_1) \in \Gamma$. By NECU and UMKhm, it is clear that $\mathcal{K}hm(\psi, \chi, \varphi_1) \in \Gamma$.
- $n \geq 2$. By induction on i, next we will show that $\mathcal{K}hm(\psi, \chi, \varphi_i) \in \Gamma$ for each $1 \leq i \leq n$. For the case of $i = 1$, with the similar proof as in the case of $n = 1$, we can show that a_1 can only be $\langle \psi_1, \bot, \varphi_1 \rangle$ and $\mathcal{U}(\psi \to \psi_1) \in \Gamma$. Therefore by UKhm we have $\mathcal{K}hm(\psi, \chi, \varphi_1) \in \Gamma$. Under the induction hypothesis (IH) that $\mathcal{K}hm(\psi, \chi, \varphi_i) \in \Gamma$ for each $1 \leq i \leq k$, we will show that $\mathcal{K}hm(\psi, \chi, \varphi_{k+1}) \in \Gamma$, where $1 \leq k \leq n-1$. Because σ is strongly executable at v, it follows that there are $w', v' \in \mathcal{S}^c$ such that

$$v \xrightarrow{a_1} \cdots \xrightarrow{a_{k-1}} w' \xrightarrow{a_k} v' \xrightarrow{a_{k+1}} \cdots \xrightarrow{a_n} t.$$

  Moreover, for each t' with $w' \xrightarrow{a_k} t'$ we have $\chi \in \mathtt{L}(t')$. It follows by Proposition 11 that $\mathcal{U}(\varphi_k \to \chi) \in \Gamma$ (▲). Proceeding, there are two cases of a_{k+1}:

  - $a_{k+1} = \langle \psi_{k+1}, \bot, \varphi_{k+1} \rangle$. Since σ is strongly executable at v, it follows that for each t' with $w' \xrightarrow{a_k} t'$ we know that a_{k+1} is executable at each t'. It follows by the definition of $\xrightarrow{\langle \psi_{k+1}, \bot, \varphi_{k+1} \rangle}$ that $\psi_{k+1} \in \mathtt{L}(t')$. Moreover, since a_k is executable at w', it follows by Proposition 11 that $\mathcal{U}(\varphi_k \to \psi_{k+1}) \in \Gamma$. Since $a_{k+1} \in \mathbf{\Sigma}_\Gamma$, it then follows that $\mathcal{K}hm(\psi_{k+1}, \bot, \varphi_{k+1}) \in \Gamma$. It then follows by ULKhm that $\mathcal{K}hm(\varphi_k, \bot, \varphi_{k+1}) \in \Gamma$. Since $\vdash \mathcal{U}(\bot \to \chi)$, it follows by UMKhm that $\mathcal{K}hm(\varphi_k, \chi, \varphi_{k+1}) \in \Gamma$. Since by IH we have that $\mathcal{K}hm(\psi, \chi, \varphi_k) \in \Gamma$, It follows from (▲) and COMPKhm that $\mathcal{K}hm(\psi, \chi, \varphi_{k+1}) \in \Gamma$.
  - $a_{k+1} = \langle \chi_{k+1}^{\psi_{k+1}}, \varphi_{k+1} \rangle$. Since σ is strongly executable at v, it follows that for each t' with $w' \xrightarrow{a_k} t'$ we know that a_{k+1} is executable at t'. Then we have that $\mathtt{R}(t') = \chi_{k+1}^{\psi_{k+1}}$ for each t' with $w' \xrightarrow{a_k} t'$.

Note that the action a_k cannot be in the form of $\langle \chi_k^{\psi_k}, \varphi_k \rangle$. Suppose it can be, let $v'' = (\mathtt{L}(v'), \chi_0^{\psi_{k+1}})$ where $\vdash \chi_0 \leftrightarrow \chi_{k+1}$ and $\chi_0 \neq \chi_{k+1}$. Since $w' \xrightarrow{a_k} v'$, it follows that $\varphi_k \in \mathtt{L}(v')$. Then it follows by the definition of transitions that $w' \xrightarrow{a_k} v''$. However, we know that $\mathtt{R}(v'') \neq \chi_{k+1}^{\psi_{k+1}}$ thus $a_{k+1} = \langle \chi_{k+1}^{\psi_{k+1}}, \varphi_{k+1} \rangle$ is not executable at v'', contradicting the strong executability. Therefore, we know that a_k cannot be in the form of $\langle \chi_k^{\psi_k}, \varphi_k \rangle$.

Now $a_k = \langle \psi_k, \bot, \varphi_k \rangle$. Since $w' \xrightarrow{a_k} v'$ and $a_{k+1} = \langle \chi_{k+1}^{\psi_{k+1}}, \varphi_{k+1} \rangle$ is executable at v', we have $\mathtt{R}(v') = \varphi_k^{\psi_k} = \chi_{k+1}^{\psi_{k+1}}$ by definition of transitions. It follows that $\psi_k = \psi_{k+1}$ and $\varphi_k = \chi_{k+1}$. Since $a_{k+1} \in \mathbf{\Sigma}_\Gamma$, it follows that $\mathcal{K}hm(\psi_{k+1}, \chi_{k+1}, \varphi_{k+1}) \in \Gamma$. Thus, we have $\mathcal{K}hm(\psi_k, \varphi_k, \varphi_{k+1}) \in \Gamma$. By ($\blacktriangle$) and `UMKhm` we then have that $\mathcal{K}hm(\psi_k, \chi, \varphi_{k+1}) \in \Gamma$ ($\blacktriangledown$). If $k = 1$, by Proposition 10 it is easy to show that $\mathcal{U}(\psi \to \psi_1) \in \Gamma$. Then by `ULKhm` we have $\mathcal{K}hm(\psi, \chi, \varphi_{k+1}) \in \Gamma$. If $k > 1$, there is a state w'' such that

$$v \xrightarrow{a_1} \cdots \xrightarrow{a_{k-2}} w'' \xrightarrow{a_{k-1}} w' \xrightarrow{a_k} v' \xrightarrow{a_{k+1}} \cdots \xrightarrow{a_n} t.$$

Since σ is strongly executable at v, it follows that for each t' with $w'' \xrightarrow{a_{k-1}} t'$ we have a_k is executable at t'. It follows by the definition of $\xrightarrow{\langle \psi_k, \bot, \varphi_k \rangle}$, it follows that $\psi_k \in \mathtt{L}(t')$ for each t' with $w'' \xrightarrow{a_{k-1}} t'$. Since a_{k-1} is executable at w'', it follows by Proposition 11 that $\mathcal{U}(\varphi_{k-1} \to \psi_k) \in \Gamma$. Moreover, since $v \xrightarrow{\sigma_{k-1}} t'$ for each t' with $w'' \xrightarrow{a_{k-1}} t'$, it follows that $\chi \in \mathtt{L}(t')$. Thus by Proposition 11 again, we have $\mathcal{U}(\varphi_{k-1} \to \chi) \in \Gamma$. Since we have proved ($\blacktriangledown$), it follows by `ULKhm` that $\mathcal{K}hm(\varphi_{k-1}, \chi, \varphi_{k+1}) \in \Gamma$. Since by IH we have $\mathcal{K}hm(\psi, \chi, \varphi_{k-1}) \in \Gamma$, it follows by `COMPKhm` that $\mathcal{K}hm(\psi, \chi, \varphi_{k+1}) \in \Gamma$.

□

Now we are ready to prove the truth lemma.

Lemma 1. *For each φ, we have $\mathcal{M}_\Gamma^c, w \vDash \varphi$ iff $\varphi \in \mathtt{L}(w)$.*

Proof. Boolean cases are trivial, and we only focus on the case of $\mathcal{K}hm(\psi, \chi, \varphi)$.

Left to Right: If there is no state w' such that $\mathcal{M}_\Gamma^c, w' \vDash \psi$, it follows by induction that $\neg\psi \in \mathtt{L}(w')$ for each $w' \in \mathcal{S}^c$. It follows by Proposition 9 that $\neg\psi \in \Delta$ for all $\Delta \in \Phi_\Gamma$. By Proposition 7, we have $\mathcal{U}\neg\psi \in \mathtt{L}(w)$. By `UNIV`, $\mathcal{K}hm(\psi, \bot, \bot) \in \mathtt{L}(w)$. Since $\vdash \bot \to \chi$ and $\vdash \bot \to \varphi$. Then by `NECU`, we have $\vdash \mathcal{U}(\bot \to \chi)$ and $\vdash \mathcal{U}(\bot \to \varphi)$. By `UMKhm` and `URKhm`, it is obvious that $\mathcal{K}hm(\psi, \chi, \varphi) \in \mathtt{L}(w)$.

Next, assuming $\mathcal{M}_\Gamma^c, v \vDash \psi$ for some $v \in \mathcal{S}^c$, we will show $\mathcal{K}hm(\psi, \chi, \varphi) \in \mathtt{L}(w)$. Since $\mathcal{M}_\Gamma^c, w \vDash \mathcal{K}hm(\psi, \chi, \varphi)$, it follows that there exists $\sigma \in \mathbf{\Sigma}^*$ such that for each $\mathcal{M}_\Gamma^c, w' \vDash \psi$: σ is strongly χ-executable at w' and $\mathcal{M}_\Gamma^c, v' \vDash \varphi$ for all v' with $w' \xrightarrow{\sigma} v'$. There are two cases: σ is empty or not.

- $\sigma = \epsilon$. This means that $\mathcal{M}^c_\Gamma, w' \vDash \varphi$ for each $\mathcal{M}^c_\Gamma, w' \vDash \psi$. It follows by induction that $\psi \in \mathtt{L}(w')$ implies $\varphi \in \mathtt{L}(w')$. Thus, we have $\psi \to \varphi \in \mathtt{L}(w')$ for all $w' \in \mathcal{S}^c$. By Proposition 9, we have $\psi \to \varphi \in \Delta$ for all $\Delta \in \Phi_\Gamma$. It follows by Proposition 7 that $\mathcal{U}(\psi \to \varphi) \in \mathtt{L}(w)$. It then follows by `EMPKhm` that $\mathcal{K}hm(\psi, \bot, \varphi) \in \mathtt{L}(w)$. By `NECU` and `UMKhm`, it is easy to show that $\mathcal{K}hm(\psi, \chi, \varphi) \in \mathtt{L}(w)$.
- $\sigma = a_1 \cdots a_n$ where for each $1 \leq i \leq n$, $a_i = \langle \psi_i, \bot, \varphi_i \rangle$ or $a_i = \langle \chi_i^{\psi_i}, \varphi_i \rangle$. Since σ is strongly χ-executable at each w' with $\mathcal{M}^c_\Gamma, w' \vDash \psi$, it follows by IH that for each ψ-state w': σ is strongly executable at w' and $w' \xrightarrow{\sigma_j} t'$ implies $\chi \in \mathtt{L}(t')$ for all $1 \leq j < n$. By Proposition 12, we have that $\mathcal{K}hm(\psi, \chi, \varphi_n) \in \mathtt{L}(v)$. Since $\mathcal{M}^c_\Gamma, v \vDash \psi$ and σ is strongly χ-executable at v and $\mathcal{M}^c_\Gamma, v'' \vDash \varphi$ for each v'' with $v \xrightarrow{\sigma} v''$, it follows that there exists v' such that a_n is executable at v' and $\mathcal{M}^c_\Gamma, v'' \vDash \varphi$ for each v'' with $v' \xrightarrow{a_n} v''$. (Please note that $v' = v$ if $n = 1$.) Note that a_n is either $\langle \psi_n, \bot, \varphi_n \rangle$ or $\langle \chi_n^{\psi_n}, \varphi_n \rangle$. It follows by Proposition 11 and IH that $\mathcal{U}(\varphi_n \to \varphi) \in \Gamma$, then we have $\mathcal{U}(\varphi_n \to \varphi) \in \mathtt{L}(v)$. It follows by `URKhm` and Proposition 6 that $\mathcal{K}hm(\psi, \chi, \varphi) \in \mathtt{L}(w)$.

This completes the proof for $w \vDash \mathcal{K}hm(\psi, \chi, \varphi)$ implies $\mathcal{K}hm(\psi, \chi, \varphi) \in \mathtt{L}(w)$.

Right to Left: Suppose that $\mathcal{K}hm(\psi, \chi, \varphi) \in \mathtt{L}(w)$, we need to show that $\mathcal{M}^c_\Gamma, w \vDash \mathcal{K}hm(\psi, \chi, \varphi)$. There are two cases: there is a state $w' \in \mathcal{S}^c$ such that $\mathcal{M}^c_\Gamma, w' \vDash \psi$ or not. If there is no such state, it follows $\mathcal{M}^c_\Gamma, w \vDash \mathcal{K}hm(\psi, \chi, \varphi)$.

For the second case, let w' be a state such that $\mathcal{M}^c_\Gamma, w' \vDash \psi$. It follows by IH that $\psi \in \mathtt{L}(w')$. Since we already have $\mathcal{K}hm(\psi, \chi, \varphi) \in \mathtt{L}(w)$, it follows by Proposition 6 that $\mathcal{K}hm(\psi, \chi, \varphi) \in \Gamma$. Since $\vdash \mathcal{U}(\chi \to \top)$, it follows by `UMKhm` that $\mathcal{K}hm(\psi, \top, \varphi) \in \Gamma$. It follows by Proposition 8 that there exists $\Delta' \in \Phi_\Gamma$ such that $\varphi \in \Delta'$. There are two cases: $\mathcal{K}hm(\psi, \bot, \varphi) \in \Gamma$ or not.

- $\mathcal{K}hm(\psi, \bot, \varphi) \in \Gamma$. It follows that $a = \langle \psi, \bot, \varphi \rangle \in \mathbf{\Sigma}_\Gamma$. Therefore, we have $v = (\Delta', \varphi^\psi) \in \mathcal{S}^c$. Since $\psi \in \mathtt{L}(w')$, it follows that $w' \xrightarrow{a} v$. Thus, a is strongly χ-executable at w'. What is more, $\varphi \in \mathtt{L}(v')$ for each v' with $w' \xrightarrow{a} v'$ by the definition of the transition. It follows by IH that $\mathcal{M}^c_\Gamma, v' \vDash \varphi$ for all v' with $w' \xrightarrow{a} v'$. Therefore, we have $\mathcal{M}^c_\Gamma, w \vDash \mathcal{K}hm(\psi, \chi, \varphi)$ witnessed by a single step σ.
- $\neg\mathcal{K}hm(\psi, \bot, \varphi) \in \Gamma$. It follows by `ONEKhm` that $\mathcal{K}hm(\psi, \bot, \chi) \in \Gamma$. We then have $a = \langle \psi, \bot, \chi \rangle \in \mathbf{\Sigma}_\Gamma$ and $b = \langle \chi^\psi, \varphi \rangle \in \mathbf{\Sigma}_\Gamma$. Since $\mathcal{K}hm(\psi, \bot, \chi) \in \Gamma$ and $\vdash \mathcal{U}(\bot \to \top)$, it follows by `UMKhm` that $\mathcal{K}hm(\psi, \top, \chi) \in \Gamma$. It follows by Proposition 8 that there exists $\Delta'' \in \Phi_\Gamma$ such that $\chi \in \Delta''$. Therefore, we have $t = (\Delta'', \chi^\psi) \in \mathcal{S}^c$. Since there exists $\Delta' \in \Phi_\Gamma$ with $\varphi \in \Delta'$, it follows by Proposition 7 that there is $t' \in \mathcal{S}^c$ such that $\mathtt{L}(t') = \Delta'$. Now, starting with any ψ-state, a is clearly executable and it will lead to a χ-state, and then by a b step we will reach all the φ states. Therefore, by IH, we have that ab is strongly χ-executable at w', and that for all v' with $w' \xrightarrow{ab} v'$ we have $\mathcal{M}^c_\Gamma, v' \vDash \varphi$. Therefore, we have $\mathcal{M}^c_\Gamma, w \vDash \mathcal{K}hm(\psi, \chi, \varphi)$. Note that we do need a 2-step σ in this case.

□

Now due to a standard Lindenbaum-like argument, each $\mathbb{SKHM}$-consistent set of formulas can be extended to a maximal consistent set Γ. Due to the truth lemma, $\mathcal{M}^c_\Gamma, (\Gamma, \top^\top) \vDash \Gamma$. The completeness of $\mathbb{SKHM}$ follows immediately.

Theorem 2. $\mathbb{SKHM}$ *is strongly complete w.r.t. the class of all models.*

4 Conclusions

This paper generalizes the knowing how logic presented in [10] and proposes a ternary modal operator $\mathcal{K}hm(\psi, \chi, \varphi)$ to express that the agent knows how to achieve φ given ψ while maintaining χ in-between. This paper also presents a sound and complete axiomatization of this logic. Compared to the completeness proof in [10], the proof here is much more complicated. The essential difference is that, in order to handle the intermediate constraints, a state of the canonical model here is a pair consisting of a maximal consistent set and a marker of the form χ^ψ which indicates that this state has a $\langle \psi, \bot, \chi \rangle$-predecessor.

For future research, besides the obvious questions of decidability and model theory of the logic, we may give some alternative semantics to the same language by relaxing the strong executability. Intuitively, strongly executable plans may be too strong for knowledge-how in some cases. For example, if there is an action sequence σ in the agent's ability map such that doing σ at a ψ-state will always make the agent *stop* on φ states, we can probably also say the agent knows how to achieve φ given ψ, e.g., I know how to start the engine in that old car, just turn the key several times until it starts, and five times should suffice at most. Please note that there are two kinds of states on which the agent might stop: either states that the agent achieves after doing σ successfully, or states on which the agent is unable to continue executing the remaining actions.

Another interesting topic is extending this logic with the public announcement operator. Intuitively, $[\theta]\varphi$ says that φ holds after the information θ is provided. The update of the new information amounts to the change of the background knowledge throughout the model, and this may affect the knowledge-how. For example, a doctor may not know how to treat a patient with the disease p since he is worried that the only available medicine might cause some very bad side-effect r, which can be expressed as $\neg\mathcal{K}hm(p, \neg r, \neg p)$. Suppose a new scientific discovery shows that the side-effect is not possible under the relevant circumstance, then the doctor should know how to treat the patient, which can be expresses as $[\neg r]\mathcal{K}hm(p, \neg r, \neg p)$.[6]

Moreover, we can consider contingent plans which involve conditions based on the knowledge of the agent. A *contingent* plan is a partial function on the agent's belief space. Such plans make more sense when the agent has the ability of observations during the execution of the plan. To consider contingent plan, we need to extend the model (ability map) with an epistemic relation. We then can express knowledge-that and knowledge-how at the same time, and discuss their interactions in one unified logical framework.

[6] However, the announcement operator $[\varphi]$ is not reducible in $\mathbf{L_{Khm}}$ as discussed in the full version of [10] which is under submission.

References

1. Ågotnes, T., Goranko, V., Jamroga, W., Wooldridge, M.: Knowledge and ability. In: van Ditmarsch, H., Halpern, J., van der Hoek, W., Kooi, B. (eds.), Handbook of Epistemic Logic, chapter 11, pp. 543–589. College Publications (2015)
2. Fan, J., Wang, Y., van Ditmarsch, H.: Almost necessary. Adv. Modal Logic **10**, 178–196 (2014)
3. Fan, J., Wang, Y., van Ditmarsch, H.: Contingency and knowing whether. Rev. Symbolic Logic **8**, 75–107 (2015)
4. Gochet, P.: An open problem in the logic of knowing how. In: Hintikka, J. (ed.), Open Problems in Epistemology. The Philosophical Society of Finland (2013)
5. Gu, T., Wang, Y.: Knowing value logic as a normal modal logic. In: Beklemishev, L., Demri, S., Máté, A. (eds.) Advances in Modal Logic, vol. 11, pp. 362–381 (2016)
6. Hintikka, J.: Knowledge and Belief: An Introduction to the Logic of the Two Notions. Cornell University Press, Ithaca (1962)
7. Lau, T., Wang, Y.: Knowing your ability. The Philosophical Forum (2016, forthcoming)
8. Smith, D.E., Weld, D.S.: Conformant graphplan. In: AAAI, vol. 98, pp. 889–896 (1998)
9. von Wright, G.H.: An Essay in Modal Logic. North Holland, Amsterdam (1951)
10. Wang, Y.: A logic of knowing how. In: van der Hoek, W., Holliday, W.H., Wang, W. (eds.) LORI 2015. LNCS, vol. 9394, pp. 392–405. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48561-3_32
11. Wang, Y.: Representing imperfect information of procedures with hyper models. In: Banerjee, M., Krishna, S.N. (eds.) ICLA 2015. LNCS, vol. 8923, pp. 218–231. Springer, Heidelberg (2015). doi:10.1007/978-3-662-45824-2_16
12. Wang, Y.: Beyond knowing that: a new generation of epistemic logics. In: van Ditmarsch, H., Sandu, G. (eds.), Jaakko Hintikka on knowledge and game theoretical semantics. Springer, Heidelberg (2016, forthcoming)
13. Wang, Y., Fan, J.: Knowing that, knowing what, and public communication: public announcement logic with Kv operators. Proc. IJCAI **13**, 1147–1154 (2013)
14. Wang, Y., Fan, J.: Conditionally knowing what. Adv. Modal Logic **10**, 569–587 (2014)
15. Yu, Q., Li, Y., Wang, Y.: A dynamic epistemic framework for conformant planning. In: Proceedings of TARK 2015, pp. 298–318 (2015). EPTCS 2016

Peirce's Sequent Proofs of Distributivity

Minghui Ma[1(✉)] and Ahti-Veikko Pietarinen[2]

[1] Institute of Logic and Cognition, Sun Yat-Sen University, Guangzhou, China
mmh.thu@gmail.com
[2] Chair of Philosophy, Tallinn University of Technology, Tallinn, Estonia
ahti.pietarinen@gmail.com

Abstract. Peirce's 1880 work on the algebra of logic resulted in a successful calculus (**PC**) for Boolean algebra. Its leading principle (Peirce's Rule) is that of residuation. We show how the law of distributivity, which Peirce states but does not prove in 1880, can be proved using Peirce's Rule in **PC**. The system **PC** is here presented as a sequent calculus, which was also Peirce's preferred method. We then give a shorter proof in his 1896 graphical alpha system, and remark on the main findings also of historical importance.

Keywords: Peirce's rule · Distributivity · Sequent calculus · Alpha graphs

1 Introduction

Charles Peirce produced several versions of algebraic and logical calculi when working on the improvements on Boole's work since the late 1860s [10–15,17]. These calculi have in the previous literature been taken to exhibit an important transition from algebraically considered systems to proper logical languages [1, 4,5,7,9,26,31]. They also were destined to lead to the graphical method of logic, namely the theory of existential graphs [3,23,25].

It is widely believed that Peirce's calculus can be understood as natural deduction [4,27,30]. However, we show that Peirce's calculus is a calculus of *sequents*. In the modern terminology, a sequent calculus is a theory about *consequence relation*, and unlike natural deduction does not appeal to assumptions that need to be discharged.

We present Peirce's calculi of 1880 on the algebra of logic [11], in which he asserted, but did not produce, the proof of the distributivity laws. We explain Peirce's sequent calculus for Boolean algebras, and show the centrality of its leading principle (Peirce's Rule) by which the full law of distributivity can then

M. Ma—The work is supported by the National Foundation for Social Sciences and Humanities (grant no. 16CZX049).

A. Pietarinen—The work is supported by the Academy of Finland (project 1270335) and the Estonian Research Council (project PUT 1305) (Principle Investigator A.-V. Pietarinen).

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 168–182, 2017.
DOI: 10.1007/978-3-662-54069-5_13

be proven. We also show that in his 1896 invention of the system of alpha graphs and its calculus [17], which is the graphical version of propositional logic, the proof of distributivity becomes very short. We conclude that Peirce was exactly developing such a sequent calculus for Boolean algebras. The main historical observations are also provided.

2 Peirce's Sequent Calculus for Boolean Algebras

In 1880 Peirce presents a calculus for Boolean algebras [11]. Using modern Lindenbaum–Tarski construction, one can prove its soundness and completeness with respect to the class of all Boolean algebras. It is in this way, by improving upon Boole's work, that he came to develop various calculi for classical propositional logic. Next, we analyse Peirce's calculus and show how the distributivity laws are proved in it.

2.1 The Leading Principle

In Sect. 2, Peirce began with the treatment of illation (deduction). He described the general form of inference as follows:

> The general type of inference is
>
> $$\begin{array}{ll} & P \\ \therefore & C, \end{array}$$
>
> where $\therefore$ is the sign of illation. [11, p. 17]

P is the premiss (or a set of premises), and C is the conclusion obtained by using rules of inference. A general rule of inference is also called a "habit" by Peirce. He then introduced the vital *leading principle*:

> A habit of inference may be formulated in a proposition which shall state that every proposition c, related in a given general way to any true proposition p, is true. Such a proposition is called the *leading principle* of the class of inferences whose validity it implies.

Peirce then introduced a sign $\prec$ of the copula to express this leading principle. The form $P \ \therefore \ C$ expresses an argument, and $P_i \prec C_i$ expresses *the truth of its leading principle*. Peirce presents the meaning of the copula in a modern, model-theoretic fashion:

> The symbol $\prec$ is the copula, and signifies primarily that every state of things in which a proposition of the class P_i is true is a state of things in which the corresponding propositions of the class C_i is true. [11, p. 18]

It follows that the copula $\prec$ is the sign that stands for logical consequence. The calculus that he develops is indeed about the copula and its properties.

Peirce emphasizes the significance of the leading principle or the copula. He identifies the copula of the form $A \prec B$ with a compound proposition built from

A (the premiss) and B (the conclusion) by the sentential operation of material implication. In Sect. 3 of the 1880 paper on forms of propositions, he stated the following:

> The forms $A \prec B$, or A implies B, and $A \overline{\prec} B$, or A does not imply B, *embrace* both hypothetical and categorial propositions. ... To say, 'if A, then B' is obviously the same as to say that from A, B follows, logically or extralogically. By thus identifying the relation expressed by the copula with that of illation, we identify the proposition with the inference, and the term with the proposition. This identification, by means of which all that is found true of term, proposition, or inference is at once known to be true of all three, is a most important engine of reasoning, which we have gained by beginning with a consideration of the genesis of logic. ([11, pp. 21–22], added emphasis)

It has been believed ever since [28] that Peirce (and also Schröder) confused the metalogical consequence relation with the material implication. However, we show that the upshot of Peirce's important identification of the two actually marks a vital discovery in the history of logic, and that it is this identification that justifies Peirce's calculus as a calculus for Boolean algebras.

2.2 The Algebra of the Copula

For the sake of clarity, we separate the two meanings of the copula $\prec$ using two symbols: (1) the consequence relation (the sign of illation) $\Rightarrow$, and (2) the material implication $\rightarrow$. Peirce introduced the algebra of the copula in Sect. 4 of the 1880 paper. His algebra of the copula is a calculus of the consequence relation. An expression of the form $x \Rightarrow y$ is called a *sequent*.

Defintion 1. *The calculus of copula consists of the following axiom and rules:*

(1) Identity: (Id) $x \Rightarrow x$

(2) Peirce's Rule:

$$\frac{x \wedge y \Rightarrow z}{x \Rightarrow y \rightarrow z}\,(\mathrm{PR})$$

(3) Rule of Transitivity:

$$\frac{x \Rightarrow y \quad y \Rightarrow z}{x \Rightarrow z}\,(\mathrm{Tr})$$

The double line in (PR) *means that the lower sequent can be derived from the upper sequent and vice versa.*

The axiom of identity is easy to understand. Every proposition follows from itself. Peirce explained it in terms of the memory or monotonicity of belief: what we have hitherto believed we continue to believe, in the absence of any reason to the contrary. We name the second rule *Peirce's Rule*, because it is probably the first formulation of the *law of residuation*: that the material implication is a right residual of conjunction. We remark on the nature and significance of Peirce's Rule after having presented his 1880 calculus.

The meaning of the rule (Tr) is the transitivity of the consequence relation. If y follows from x and z follows from y, then z must follow from x. Peirce mentions that the transitivity of the copula derives from De Morgan's work. He also states that "the same principle may be algebraically conceived as a rule for the elimination of y from the two propositions $x \prec y$ and $y \prec z$" [11, p. 25]. After Gentzen's 1934 work, the rule (Tr) became called a *cut rule* in proof theory. It concerns the elimination of the middle, or the cut term.

2.3 Peirce's Calculus for Boolean Algebras

After the introduction of the algebra of the copula, Peirce continued his 1880 exposition to introduce the logic of non-relative terms. The non-relative terms are constructed from propositions using logical multiplication $\times$ and addition $+$. Here we change the notation into $\wedge$ for conjunction and $\vee$ for disjunction.

First of all, Peirce commented on the rule (PR) when the negation sign is introduced. For any term x, let $\overline{x}$ be the negation of x. Then the proposition $x \to y$ is equivalent with $\overline{x} \vee y$. Hence by (PR) we can derive:

$$\frac{x \wedge y \Rightarrow z}{x \Rightarrow \overline{y} \vee z}$$

Moreover, Peirce stated the following derived variant of the rule (PR):

$$\frac{x \wedge \overline{y} \Rightarrow z}{x \Rightarrow y \vee z}$$

Two important further variants of (PR) can be stated as follows:

$$\frac{x \Rightarrow y}{(\text{The possible}) \Rightarrow \overline{x} \vee y} \qquad \frac{x \Rightarrow y}{x \wedge \overline{y} \Rightarrow (\text{The impossible})}$$

Peirce proceeded to introduce two notations: 0 for the impossible, and ∞ for the possibility. We replace 0 and ∞ with $\bot$ and $\top$ respectively. The following axioms were given by Peirce:

$$(\top) \quad x \Rightarrow \top \quad (\bot) \quad \bot \Rightarrow x$$

Moreover, from the axiom (Id), and using the two variants of (PR), one can derive the law of excluded middle and the law of contradiction:

$$\top \Rightarrow x \vee \overline{x} \quad \text{and} \quad \overline{x} \wedge x \Rightarrow \bot.$$

The negation sign can be defined in terms of $\to$ and $\bot$ as follows:

$$\overline{x} := x \to \bot.$$

Then $\overline{x} \wedge x \Rightarrow \bot$ is obtained from $x \to \bot \Rightarrow x \to \bot$ by (PR).

Peirce then introduced the rules for conjunction (multiplication) and disjunction (addition). The definition of his calculus is now complete. He proved all the axioms of lattices and stated the distributive laws. For convenience, we summarize his 1880 calculus for classical propositional logic as follows:

Defintion 2. *Peirce's calculus* **PC** *consists of the following axioms and rules:*

(1) Axioms:

$$\text{(Id)}\ \ x \Rightarrow x \quad (\top)\ \ x \Rightarrow \top \quad (\bot)\ \ \bot \Rightarrow x \quad (Em)\ \ \top \Rightarrow x \vee \overline{x}$$

(2) Rules:

$$\frac{x \wedge y \Rightarrow z}{x \Rightarrow y \rightarrow z}\,(\text{PR}) \qquad \frac{x \Rightarrow y \quad y \Rightarrow z}{x \Rightarrow z}\,(\text{Tr})$$

$$\frac{x_1 \Rightarrow z \quad x_2 \Rightarrow z}{x_1 \vee x_2 \Rightarrow z}\,(\vee\text{I}) \qquad \frac{z \Rightarrow x_1 \quad z \Rightarrow x_2}{z \Rightarrow x_1 \wedge x_2}\,(\wedge\text{I})$$

$$\frac{x_1 \vee x_2 \Rightarrow z}{x_i \Rightarrow z}\,(\vee\text{E}) \qquad \frac{z \Rightarrow x_1 \wedge x_2}{z \Rightarrow x_i}\,(\wedge\text{E})$$

In $(\vee\text{E})$ *and* $(\wedge\text{E})$, $i \in \{1, 2\}$. *A* derivation *of a sequent* $x \Rightarrow y$ *in* **PC** *is a proof tree with the root* $x \Rightarrow y$ *such that each node is either an axiom or derived by a rule of inference. A sequent* $x \Rightarrow y$ *is* derivable *in* **PC** *(notation* $\vdash_{\mathbf{PC}} x \Rightarrow y$*) if there is a derivation of* $x \Rightarrow y$ *in* **PC**.

Peirce also introduced the equality sign $(=)$: $x = y$ is a shorthand for "$x \Rightarrow y$ and $y \Rightarrow x$". One can now easily derive the following lattice-theoretic equalities:

(Idempotency)	$x = x \vee x$	$x \wedge x = x$
(Commutativity)	$x \vee y = y \vee x$	$x \wedge y = y \wedge x$
(Associativity)	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$
(Absorption)	$x \vee (y \wedge z) = x$	$x \wedge (y \vee z) = x$.

3 Distributivity Laws

3.1 Derivation in Peirce's Sequent Calculus

In the 1880 paper, Peirce stated the following distributive laws:

$$\text{(D1)}\ (x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z) \qquad \text{(D2)}\ (x \wedge y) \vee z = (x \vee z) \wedge (y \vee z).$$

He casually mentions that "they are easily proved . . ., but the proof is too tedious to give" [11, p. 33]. This passage provoked a challenge by Schröder, who took the laws of distributivity to be independent from the lattice axioms. A rejoinder and a lively discussion ensued and Peirce's lost and subsequently recovered version of the proof was finally added to Huntington's 1904 paper [8]. [6] provides a rich account of the context of Peirce's casual note and the debates that followed.

We can give a proof of (D1) and (D2) in Peirce's calculus. The following lemmas stated by Peirce can be derived in the calculus **PC**.

Lemma 1. *The sequents* $x_i \Rightarrow x_1 \vee x_2$ *and* $x_1 \wedge x_2 \Rightarrow x_i$ *for* $i = 1, 2$ *are derivable in* **PC**.

Proof. By replacing z in ($\vee$E) and ($\wedge$E) with $x_1 \vee x_2$ and $x_1 \wedge x_2$ respectively.

Lemma 2. *The following rules are derivable in* **PC***:*

$$\text{(R1)}\ \frac{x \Rightarrow y}{z \to x \Rightarrow z \to y} \qquad \text{(R2)}\ \frac{x \Rightarrow z \quad y \Rightarrow u}{x \wedge y \Rightarrow z \wedge u}$$

$$\text{(R3)}\ \frac{x \Rightarrow y}{y \to z \Rightarrow x \to z} \qquad \text{(R4)}\ \frac{x \Rightarrow z \quad y \Rightarrow u}{x \vee y \Rightarrow z \vee u}$$

Proof. (R1) is derived as follows:

$$\dfrac{\dfrac{\dfrac{z \to x \Rightarrow z \to x}{(z \to x) \wedge z \Rightarrow x}\,\text{(PR)} \qquad x \Rightarrow y}{(z \to x) \wedge z \Rightarrow y}\,\text{(Tr)}}{z \to x \Rightarrow z \to y}\,\text{(PR)}$$

(R2) is derived as follows:

$$\dfrac{\dfrac{x \wedge y \Rightarrow x \quad x \Rightarrow z}{x \wedge y \Rightarrow z}\,\text{(Tr)} \qquad \dfrac{x \wedge y \Rightarrow y \quad y \Rightarrow u}{x \wedge y \Rightarrow u}\,\text{(Tr)}}{x \wedge y \Rightarrow z \wedge u}\,\text{(}\wedge\text{I)}$$

For (R3), assume $x \Rightarrow y$. By (R2), we get $x \wedge (y \to z) \Rightarrow y \wedge (y \to z)$. One easily derives that $y \wedge (y \to z) \Rightarrow z$ by (PR) and commutativity. Then by (Tr), we have $x \wedge (y \to z) \Rightarrow z$. Finally, by commutativity and (PR), we get $y \to z \Rightarrow x \to z$. The rule (R4) is likewise easily shown.

Before deriving the distributive laws, we observe that since $\wedge$ is commutative, we also have the following version of Peirce's Rule:

$$\dfrac{\dfrac{x \wedge y \Rightarrow z}{}}{y \Rightarrow x \to z}\,\text{(PR)}$$

We can thus apply (PR) without considering the first or the second coordinate of the conjunction.

Theorem 1. *The distributive laws (D1) and (D2) are derivable in* **PC***.*

Proof. (D1) First, we derive $(x \wedge y) \vee (x \wedge z) \Rightarrow x \wedge (y \vee z)$ as follows:

$$\dfrac{x \wedge y \Rightarrow x \qquad \dfrac{x \wedge y \Rightarrow y \quad y \Rightarrow y \vee z}{x \wedge y \Rightarrow y \vee z}\,\text{(Tr)}}{x \wedge y \Rightarrow x \wedge (y \vee z)}\,\text{(}\wedge\text{I)}$$

Similarly we get $x \wedge z \Rightarrow x \wedge (y \vee z)$. By ($\vee$I), we get $(x \wedge y) \vee (x \wedge z) \Rightarrow x \wedge (y \vee z)$. Secondly, we derive $x \wedge (y \vee z) \Rightarrow (x \wedge y) \vee (x \wedge z)$ as follows:

(1) We have the following derivation:

$$\dfrac{\dfrac{x \wedge y \Rightarrow x \wedge y}{y \Rightarrow x \rightarrow (x \wedge y)}\,(\mathrm{PR}) \qquad \dfrac{x \wedge z \Rightarrow x \wedge z}{z \Rightarrow x \rightarrow (x \wedge z)}\,(\mathrm{PR})}{y \vee z \Rightarrow (x \rightarrow (x \wedge y)) \vee (x \rightarrow (x \wedge z))}\,(\mathrm{R4})$$

(2) We have the following derivation:

$$\dfrac{x \wedge y \Rightarrow (x \wedge y) \vee (x \wedge z)}{x \rightarrow (x \wedge y) \Rightarrow x \rightarrow (x \wedge y) \vee (x \wedge z)}\,(\mathrm{R1})$$

Similarly, we have $x \rightarrow (x \wedge z) \Rightarrow x \rightarrow (x \wedge y) \vee (x \wedge z)$. Then by ($\vee$I), we get $(x \rightarrow (x \wedge y)) \vee (x \rightarrow (x \wedge z)) \Rightarrow x \rightarrow (x \wedge y) \vee (x \wedge z)$.

By (1) and (2), and using (Tr), we get $y \vee z \Rightarrow x \rightarrow (x \wedge y) \vee (x \wedge z)$. By (PR), we get $x \wedge (y \vee z) \Rightarrow (x \wedge y) \vee (x \wedge z)$.

(D2) First, we easily derive $x \wedge y \Rightarrow x \vee z$ and $x \wedge y \Rightarrow y \vee z$. By ($\wedge$I), we get $x \wedge y \Rightarrow (x \vee z) \wedge (y \vee z)$. From $z \Rightarrow x \vee z$ and $z \Rightarrow y \vee z$, we get $z \Rightarrow (x \vee z) \wedge (\vee z)$. By ($\vee$I), we get $(x \wedge y) \vee z \Rightarrow (x \vee z) \wedge (y \vee z)$. Second, we have the following:

(3) By (D1), we get $(x \vee z) \wedge (y \vee z) \Rightarrow (x \wedge (y \vee z)) \vee (z \wedge (y \vee z))$. Note that $z \wedge (y \vee z) \Rightarrow z$ by absorption. Then by (R4) we get $(x \wedge (y \vee z)) \vee (z \wedge (y \vee z)) \Rightarrow (x \wedge (y \vee z)) \vee z$. By (Tr), we get $(x \vee z) \wedge (y \vee z) \Rightarrow (x \wedge (y \vee z)) \vee z$.

(4) By commutativity and (D1), we get $x \wedge (y \vee z) \Rightarrow (x \wedge y) \vee (x \wedge z)$. By (R4), we get $(x \wedge (y \vee z)) \vee z \Rightarrow ((x \wedge y) \vee (x \wedge z)) \vee z$. By associativity and absorption, we get $((x \wedge y) \vee (x \wedge z)) \vee z \Rightarrow (x \wedge y) \vee z$. Then by (Tr) we get $(x \wedge (y \vee z)) \vee z \Rightarrow (x \wedge y) \vee z$.

From (3) and (4), by (Tr), we get $(x \vee z) \wedge (y \vee z) \Rightarrow (x \wedge y) \vee z$.

Remark 1. One of Peirce's own proofs is found in manuscript R 417 written in 1893. At the end of that proof, he remarked: "This is what I had in mind in a statement Am. Math. J.III.33. which has been severely criticized". That 1880 statement was that the cases of distributive principle are "easily proved ... but the proof is too tedious to give". The 1893 proof is correct but it is not equivalent to the proof that Peirce apparently recovered from his earlier work on the 1880 paper and which he sent to Huntington on the Christmas Eve of 1903. The reason is that the 1893 proof uses a new principle of iteration.

Remark 2. Huntington published in 1904 the proof Peirce had sent to him, including Peirce's footnote about it [8]. In the published proof, the axiom—Huntington's "postulate"—number 9 is crucial. Without it, the axioms 1–8 only define uniquely complemented lattices which need not be distributive. The axiom 9 says that, if it is not the case that $a \leq \overline{b}$, then there is a non-zero element x such that $x \leq a$ and $x \leq b$. This is equivalent to the condition:

(i) If $\forall x(x \leq a \;\&\; x \leq b \Rightarrow x = 0)$, then $a \leq \overline{b}$.

The condition (i) is also equivalent to the condition:

(ii) If $\forall x(x \leq a \wedge b \Rightarrow x = 0)$, then $a \leq \overline{b}$.

Moreover, (ii) is equivalent to the condition:

(iii) $a \wedge b \leq 0$ implies $a \leq \overline{b}$.

This is a special case of Peirce's leading principle (PR). Conversely, from the axiom 9 one can derive (PR):

> Assume $a \wedge b \leq c$ but not $a \leq \overline{b \wedge \overline{c}}$. Then there is $x \neq 0$ such that $x \leq a$ and $x \leq b \wedge \overline{c}$. Then $x \leq a \wedge b \wedge \overline{c} \leq c \wedge \overline{c} = 0$, a contradiction.

Hence the axiom 9 and Peirce's Rule are equivalent. This equivalence justifies our proof of distributivity in the fashion presented here.

3.2 Negation, Contraposition and Completeness

Peirce's Rule (PR) is also closely related with the rules for negation and implication. Recall that Peirce defined negation as $\overline{x} := x \to \bot$.[1]

Proposition 1. *The following sequents and rules are derivable in* **PC***:*

(1) Double negation laws: (DB1) $x \Rightarrow \overline{\overline{x}}$; (DB2) $\overline{\overline{x}} \Rightarrow x$.
(2) Rules of Contraposition:

$$\frac{x \Rightarrow y}{\overline{y} \Rightarrow \overline{x}}(\text{CP}) \quad \frac{\overline{y} \Rightarrow \overline{x}}{x \Rightarrow y}(\text{ICP})$$

Proof. For (DB1), first we have $x \wedge (x \to \bot) \Rightarrow \bot$. By (PR), we get $x \Rightarrow \overline{\overline{x}}$. For (DB2), we start from (Em) $\top \Rightarrow x \vee \overline{x}$. Then we have $\overline{\overline{x}} \wedge \top \Rightarrow \overline{\overline{x}} \wedge (x \vee \overline{x})$. Clearly $\overline{\overline{x}} \Rightarrow \overline{\overline{x}} \wedge \top$. Then $\overline{\overline{x}} \Rightarrow \overline{\overline{x}} \wedge (x \vee \overline{x})$. By (D1), we have $\overline{\overline{x}} \wedge (x \vee \overline{x}) \Rightarrow (\overline{\overline{x}} \wedge x) \vee (\overline{\overline{x}} \wedge \overline{x})$. Clearly $\overline{\overline{x}} \wedge \overline{x} \Rightarrow \bot$. Then we have $\overline{\overline{x}} \wedge (x \vee \overline{x}) \Rightarrow (\overline{\overline{x}} \wedge x) \vee \bot$. Clearly $(\overline{\overline{x}} \wedge x) \vee \bot \Rightarrow \overline{\overline{x}} \wedge x$ and $\overline{\overline{x}} \wedge x \Rightarrow x$. By (Tr), we have $\overline{\overline{x}} \wedge (x \vee \overline{x}) \Rightarrow x$. Finally by (Tr), we have $\overline{\overline{x}} \Rightarrow x$.

The rule (CP) is an instance of (R3). The inversion of contraposition (ICP) follows immediately from (CP) and double negation laws.

Lemma 3. *The following rules are derivable in* **PC***:*

$$\frac{x \wedge y \Rightarrow z}{y \Rightarrow \overline{x} \vee z}(\text{R5}) \quad \frac{y \Rightarrow \overline{x} \vee z}{x \wedge y \Rightarrow z}(\text{R6})$$

Proof. For (R5), assume $x \wedge y \Rightarrow z$. Then $\overline{x} \vee (x \wedge y) \Rightarrow \overline{x} \vee z$. By distributivity and (Tr), we have $(\overline{x} \vee x) \wedge (\overline{x} \vee y) \Rightarrow \overline{x} \vee z$. Clearly $\overline{x} \vee x = \top$. Then we have $\overline{x} \vee y \Rightarrow \overline{x} \vee z$. Since $y \Rightarrow \overline{x} \vee y$, we have $y \Rightarrow \overline{x} \vee z$. (R6) is shown similarly.

[1] This definition of negation, that "from x anything you please necessarily follows" was, from the "*formal* point of view", perfectly acceptable to Peirce. But he also thought that it does not "really define denial in terms of consequence" (Peirce to Huntington, February 14, 1904; see also [2]).

Proposition 2. *The sequent* $x \to y = \overline{x} \vee y$ *is derivable in* **PC**.

Proof. First, we derive $\overline{x} \vee y \Rightarrow x \to y$ as follows. It is easy to show $x \wedge (\overline{x} \vee y) \Rightarrow x \wedge y$. Clearly $x \wedge y \Rightarrow y$. By (Tr), $x \wedge (\overline{x} \vee y) \Rightarrow y$. By (PR), we get $\overline{x} \vee y \Rightarrow x \to y$. Second, we have $x \wedge (x \to y) \Rightarrow y$. By (R5), we get $x \to y \Rightarrow \overline{x} \vee y$.

Peirce also admitted the importance of De Morgan laws and proved them.

Proposition 3. *The De Morgan laws* (DM1) $\overline{x \wedge y} = \overline{x} \vee \overline{y}$ *and* (DM2) $\overline{x \vee y} = \overline{x} \wedge \overline{y}$ *are derivable in* **PC**.

Proof. For (DM1), first we have $\overline{x} \Rightarrow \overline{x} \vee \overline{y}$. By (CP), we get $\overline{\overline{x} \vee \overline{y}} \Rightarrow \overline{\overline{x}}$. By (DB2) and (Tr), we have $\overline{\overline{x} \vee \overline{y}} \Rightarrow x$. Similarly we get $\overline{\overline{x} \vee \overline{y}} \Rightarrow y$. Then by ($\wedge$I), we get $\overline{\overline{x} \vee \overline{y}} \Rightarrow x \wedge y$. By (CP), (DB2) and (Tr), we ge $\overline{x \wedge y} = \overline{x} \vee \overline{y}$. The law (DM2) is shown similarly.

We can also establish the soundness and completeness of the calculus **PC** with respect to the class of all Boolean algebras. The soundness is obtained by induction on the derivation of sequents in **PC**. For the completeness, the Lindenbaum–Tarski construction is applied. We can use the equality sign that Peirce proposed to establish a congruence relation on the set of all terms constructed from a set of propositional variables Prop and $\bot$ using $\wedge, \vee, \to$.

Theorem 2 (Completeness). *A sequent* $x \Rightarrow y$ *is derivable in* **PC** *if and only if* $x \Rightarrow y$ *is valid in all Boolean algebras.*

Peirce's calculus is therefore fully adequate as a calculus for Boolean algebras. It follows that the calculus **PC**, as Peirce had presented it, is a successful calculus that agrees with classical propositional logic. Peirce achieved his sequent calculus for his logic immediately in the 1880 paper.

3.3 Peirce's Rule in Perspective

The rule (PR) is a formulation of Peirce's leading principle of inference. Leading principles that have a "maximum abstractness" (NEM 4, p. 175, 1898) are logical principles. The maximal abstractness means that such principles add nothing to the premises of the inference which they govern.

The distributivity laws are derivable from (PR), together with the lattice rules for conjunction and disjunction, the identity and transitivity (Tr). But they are not derivable without (PR). Now Schröder thought that distributive laws are independent of the theory of lattices [29, Ch. XII]. Peirce stated the point clearly in a footnote to his 1885 paper:

> It is interesting to observe that [the] reasoning [example in Peirce's 1885 paper] is dilemmatic. [...] The dilemma was only introduced into logic from rhetoric by the humanists of the *renaissance*; and at that time logic was studies with so little accuracy that the peculiar nature of this mode of reasoning escaped notice. I was thus led to suppose that the whole non-relative logic was derivable from the principles of the ancient syllogistic, and this error[2] is involved in Chapter II of my

[2] Peirce later added a note in the margin of his copy of the paper: "But it was not an error!!! See my original demonstration in marginal note." This marginal note has not been recovered.

> paper in the third volume of this Journal [the 1880 paper]. My friend, Professor Schröder, detected the mistake and showed that the distributive formulæ
>
> $$(x + y)z \prec xz + yz$$
> $$(x + z)(y + z) \prec xy + z$$
>
> could not be deduced from syllogistic principles. I had myself independently discovered and virtually stated the same thing. (*Studies in Logic*, p. 189.)[3] There is some disagreement as to the definition of the dilemma (see Keynes's excellent *Formal Logic*, p. 241); but the most useful definition would be a syllogism depending on the above distribution formulæ.[4] The distribution formulæ
>
> $$xz + yz \prec (x + y)z$$
> $$xy + z \prec (x + z)(y + z)$$
>
> are strictly syllogistic. De Morgan's added moods are virtually dilemmatic, depending on the principle of excluded middle.[5] [12, p. 190]

Clearly $(x \wedge z) \vee (y \wedge z) \Rightarrow (x \vee y) \wedge z$ (D1, right-to-left) and $(x \wedge y) \vee z \Rightarrow (x \vee z) \wedge (y \vee z)$ (D2, left-to-right) are derivable using only lattice rules and without Peirce's Rule. But our reformulation of Peirce's calculus also explains the derivability of the distributive laws $(x \vee y) \wedge z \Rightarrow (x \wedge z) \vee (y \wedge z)$ (D1, left-to-right) and $(x \vee z) \wedge (y \vee z) \Rightarrow (x \wedge y) \vee z$ (D2, right-to-left) perfectly. The calculus **PC** is not to be conceived in the style of natural deduction but as a sequent calculus. Consequently, the proof of distributive laws is not hard.

The second aspect of Peirce's Rule concerns implication which is intuitionistic. As noted, (PR) can be viewed algebraically as the law of residuation:

$$(RES) a \wedge c \leq b \text{ if and only if } c \leq a \rightarrow b.$$

This is exactly the way to introduce intuitionistic implication. A *Heyting algebra* is identified with an algebra $(H, \wedge, \vee, 0, 1, \rightarrow)$, where $(A, \wedge, \vee, 0, 1)$ is a bounded lattice, and $\rightarrow$ is a binary operation on H satisfying (RES). Now distributivity can be derived from the lattice rules and (PR). Equally, the algebraic distributive

[3] In that 1883 publication of the "Note B" in his *Studies in Logic* Peirce stated that two relatives are "undistributed" in a relative product and in a relative sum.

[4] Keynes distinguishes five different formulations of dilemmatic arguments: those given by (i) Mansel, Whately and Jevons, (ii) by Fowler, (iii) Keynes's own formulation, (iv) by Hamilton, and (v) by Thomson. Peirce appears to mean none of theirs as "the most useful definition". For example, he proposes the rule of dilemma to be "If $(a\bar{b} + c)(\bar{a} + c)$ then c" (see R 736, NEM IV, p. 115). This is proved using the distributivity principle thus: $c + (a\bar{b}\bar{a})$ implies c, and by the law of contradiction, $c + 0$ implies c. This direction of the derivation of the dilemmatic rule depends on the second distribution principle as given here in Peirce's footnote, namely one that is not derivable from the lattice rules alone. In the other direction, the distribution principle applied is strictly syllogistic.

[5] Later in 1893 Peirce takes a dilemmatic argument to be "any argument whose validity depends upon the principle of excluded middle" (CP 2.474). Dilemmatic arguments would thus not be intuitionistically valid.

laws can also be derived from (RES). Moreover, if we remove the law of excluded middle (Em) from the calculus **PC**, we would obtain not a propositional calculus but a calculus for Heyting algebras, and hence intuitionistic logic. Peirce was not far from this invention, and there are further and collateral reasons why his logic and its philosophy took steps towards intuitionism [24, Ch. 3].

4 Distributivity Law in the Alpha System

Peirce's 1896 system α has, due to the 'deep inference' nature of its proofs, very short proofs of distributivity. We shall briefly present the definition of alpha graphs and reformulate Peirce's system alpha. The short proof of distributivity is presented in alpha.

Peirce's alpha graphs are syntactic objects which are formed from simple propositions (propositional variables) on the sheet of assertion (SA) using the operation of cut $\bigcirc$ and juxtaposition. The sheet of assertion itself is a blank space, denoted by $\top$. It can be viewed as a primitive graph. The logical meaning of cut is negation, and that of juxtaposition conjunction.

Defintion 3 (Alpha Graphs). *The set* $\mathfrak{G}_\alpha$ *of all* alpha graphs *is defined inductively by the following rule:*

$$\mathfrak{G}_\alpha \ni G := p \mid \top \mid G_1\, G_2 \mid \textcircled{G}\ ,$$

where p *is a propositional letter.*

An *area* of an alpha graph is a continuous region over the sheet of assertion as defined by the interior of a cut (the oval around a graph). The graph $G_1\, G_2$ is the juxtaposition of G_1 and G_2 by placing them on the SA or within the same area. The graph $\textcircled{G}$ is the *enclosure* of G. A *partial graph* of a graph G is a graph H as a part of G.

A *position* in a graph is a point on the sheet of assertion (but not on the line of cut). A position in a graph is *positive* if it is enclosed by an even number of cuts. A position in a graph is *negative* if it is enclosed by an odd number of cuts. A graph can be inserted or deleted at a position in a graph. The notation $G\{\ \}$ stands for the graph with a distinguished position. The graph $G\{H\}$ is obtained from $G\{\ \}$ by filling the position in $G\{\ \}$ by H. Moreover, we use the notation $G\{H^+\}$ and $G\{H^-\}$ to mean that the position taken by H is positive and negative respectively.

A graphical rule is $\frac{G}{H}$ where G is the premiss and H is the conclusion. Peirce presented his system alpha in 1897–8. Here we reformulate it as follows.

Defintion 4. *The system alpha consists of the following axiom and rules:*

(1) Axiom*:* $\top$ *(Sheet of Assertion)*
(2) Deletion rule*:*

$$\frac{G\{H^+\}}{G\{\top\}}(\mathrm{DR})$$

Every positive partial graph H *in a graph* G *can be deleted.*

(3) Insertion rule*:*

$$\frac{G\{H^-\}}{G\{JH\}}(\mathrm{IR})$$

Any graph can be inserted into a negative position in a graph G*.*

(4) Double negation rule*:*

$$\frac{G\{H\}}{G\{\textcircled{\textcircled{H}}\}}(\mathrm{DN})$$

The double line means that the upper graph can derive the lower graph, and vice versa. (DN) means that any partial graph H *of a graph* G *can be replaced by a doubly enclosed* H*, and any doubly enclosed* G *can be replaced by* H*, where there is nothing between the two cuts.*

(5) Iteration/deiteration rule*:*

$$\frac{K\{GH\{J\}\}}{K\{GH\{GJ\}\}}(\mathrm{IT}) \quad \frac{K\{GH\{GJ\}\}}{K\{GH\{J\}\}}(\mathrm{DIT})$$

(IT) means that, in any graph $K\{GH\}$*, the partial graph* G *can be iterated at any position in* H*. (DIT) is the converse of (IT).*

A proof *of a graph* G *in* α *is a finite sequence of graphs* $G_0, \ldots, G_n$ *such that* $G_n = G$*, and each* G_i *is either* $\top$*, or derived from previous graphs by a rule. A graph* G *is* provable *in* α *if it has a derivation in* α*.*

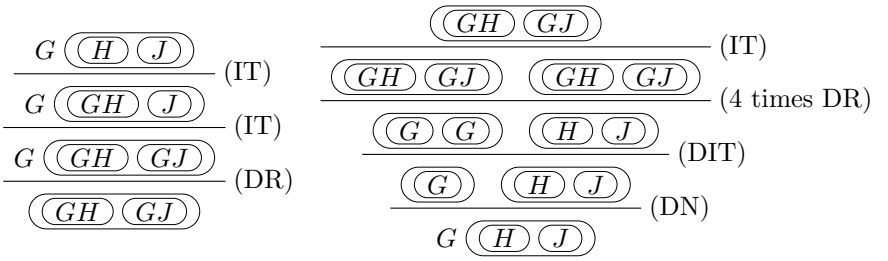
A graphical rule $\frac{G}{H}$ *is* derivable *in* α *if there is a finite sequence of graphs* $H_0, \ldots, H_n$ *such that* $H_n = H$*, and each* H_i *is either* $\top$*,* G *or derived from previous graphs by a rule.*

The alpha system is essentially a type of deep inference in the sense that inference rules apply inside graphs in the positions. In the standard sequent calculus, rules are applied only to the outermost connectives of formulas. The system alpha can be shown to be sound and complete with respect to Boolean algebras. The natural interpretation of graphs is clear from the intended meaning of graphical operations. For the purpose of the present paper, we only present the proof of the full law of distributivity in the alpha system.

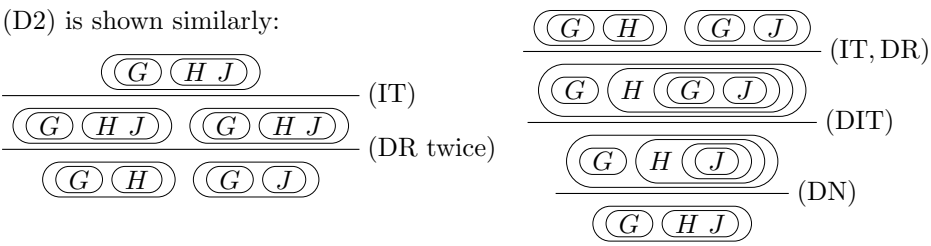
Proposition 4. *The following distributivity rules are derivable in* α*:*

$$\frac{G\,\textcircled{H}\,\textcircled{J}}{\textcircled{GH}\,\textcircled{GJ}}(\mathrm{D1}) \qquad \frac{\textcircled{G}\,\textcircled{HJ}}{\textcircled{G}\,\textcircled{H}\;\textcircled{G}\,\textcircled{J}}(\mathrm{D2})$$

Proof. For (D1), we have the following proofs:



(D2) is shown similarly:

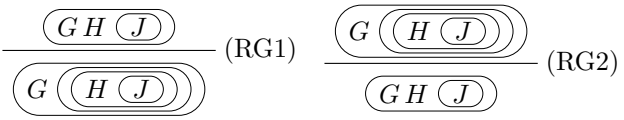


The proof of distributivity in the alpha calculus is short compared to its proof in **PC**. The reason is the deep inference nature of proofs in alpha.

Remark 3. Importantly, the bottom-up direction of (D1) and the top-down direction of (D2), the premises are first duplicated by iteration. Peirce had noticed in the *Grand Logic* (R 418, 1893) how this non-syllogistic operation of reusing the premise can be eminently useful: "The same premise may be written in more than once ... the student of exclusively non-relative logic is quite unaware that anything can be gained by bringing in again a premise already used. Ordinary syllogistic gives no hint of such a thing; nay seems rather opposed to it" (R 418). The alpha proofs of (D1) and (D2) are exactly the graphical versions of his 1893 proof of distribution given in R 418.

Remark 4. In the previous chapter of his *Grand Logic* (R 417), Peirce had produced nearly equivalent proofs to the above proof in the alpha system which do use the rule of iteration but without the initial duplication of the premises by iteration. He tells that this R 417 proof was what he always intended his demonstration of distributivity to be. He must thus also mean his lost 1880 proof that Schröder pressed Peirce to reproduce, namely one that Peirce had stated, correctly, to be "easy but tedious". However, in the 1880 system he does not yet have the rule of iteration, which casts some doubt on the credibility of the claim that he had a proof similar to the 1893 proof with iteration at hand already in 1880.

Proposition 5. *The following rules are provable in α:*



Proof. (RG1) is obtained immediately by the addition of double cuts. (RG2) is obtained by the removal of those double cuts.

The rules (RG1) and (RG2) are the graphical correlates of Peirce's leading principle. In the α system, Peirce's leading principle becomes provable in a trivial fashion. For (RG1), the proof only needs one application of (DN) that introduces a double cut. For (RG2), the proof only needs one application of (DN) in order to delete a double cut. One can therefore conclude that the validity of Peirce's Rule is in the alpha system a matter of immediate observation. This justifies its use in the earlier proofs.

5 Conclusions

Peirce created numerous logical systems that coincide with Boolean algebra. The full law of distributivity can be proved in the 1880 system by using what we term Peirce's Rule, which is a version of residuation. The full law also has an immediate proof in Peirce's 1896 graphical α system, which has a deep inference nature of proofs and which is an outgrowth of his developments of the algebra of logic and its notation. His calculi are indeed of the nature of sequents and not of natural deduction with hypotheses that need to be discharged. The central notion that characterizes his logical investigations is the illative relation of a consequence. A sequent calculus is a theory about that consequence relation. It was such sequent calculus that Peirce was developing for systems that agree with Boolean algebras.

References

1. Badesa, C.: The Birth of Model Theory: Löwenheim's Theorem in the Frame of the Theory of Relatives. Princeton University Press, Princeton (2004)
2. Bellucci, F., Pietarinen, A.-V.: Existential graphs as an instrument of logical analysis: part 1. Alpha. Rev. Symbolic Logic **9**(2), 209–237 (2016a)
3. Bellucci, F., Pietarinen, A.V.: From Mitchell to Carus: Fourteen Years of Logical Graphs in the Making. Transactions of the Charles S. Peirce Society (2016, in press)
4. Brady, G.: From Peirce to Skolem: A Neglected Chapter in the History of Logic. Elsevier Science, Amsterdam (2000)
5. Dipert, R.: Peirce's deductive logic: its development, influence, and philosophical significance. In: Misak, C. (ed.) The Cambridge Companion to Peirce, pp. 257–286. Cambridge University Press, Cambridge (2004)
6. Houser, N.: Peirce and the law of distribution. In: Drucker, T. (ed.) Perspectives on the History of Mathematical Logic, pp. 10–32. Birkhäuser, Boston (1991)
7. Houser, N., Roberts, D., Van Evra, J. (eds.): Studies in the Logic of Charles S. Peirce. Indiana University Press, Bloomington (1997)
8. Huntington, E.V.: Sets of independent postulates for the algebra of logic. Trans. Am. Math. Soc. **5**, 288–309 (1904)
9. Martin, R.M.: Peirce's Logic of Relations and Other Studies. Foris, Dordrecht (1980)

10. Peirce, C.S.: On an improvement in Boole's calculus of logic. Proc. Am. Acad. Arts Sci. **7**, 250–261 (1867)
11. Peirce, C.S.: On the algebra of logic. Am. J. Math. **3**(1), 15–57 (1880). (Reprinted in [22, vol. 4, pp. 163–209])
12. Peirce, C.S.: On the algebra of logic: a contribution to the philosophy of notation. Am. J. Math. **7**(2), 180–196 (1885)
13. Peirce, C.S.: Algebra of the Copula [Version 1]. In: Writings of Charles S. Peirce, vol. 8 (1890–1892), pp. 210–211. Indiana University Press (2010)
14. Peirce, C.S.: Grand Logic. Division I. Stecheology. Part I. Non Relative Logic. Chapter VIII. The Algebra of the Copula (R 411) (1893a)
15. Peirce, C.S.: Grand Logic. Chapter XI. The Boolian Calculus (R 417) (1893b)
16. Peirce, C.S.: Grand Logic. Book II. Division I. Part 2. Logic of Relatives. Chapter XII. The Algebra of Relatives (R 418) (1893c)
17. Peirce, C.S.: 1896–7. On Logical Graphs (R 482)
18. Peirce, C.S.: Letter to E. V. Huntington, February 14, 1904 (R L 210) (1904b)
19. Peirce, C.S.: The Collected Papers of Charles S. Peirce. vol. 8, ed. by Hartshorne, C., Weiss, P., Burks, A. W. Cambridge: Harvard University Press. Cited as CP followed by volume and paragraph number, pp. 1931–1966
20. Peirce, C.S.: Manuscripts in the Houghton Library of Harvard University, as identified by Richard Robin. Annotated Catalogue of the Papers of Charles S. Peirce, Amherst: University of Massachusetts Press (1967). Cited as R followed by manuscript number
21. Peirce, C.S.: The New Elements of Mathematics by Charles S. Peirce. vol. 4, ed. by Eisele, C. The Hague: Mouton. Cited as NEM followed by volume and page number (1976)
22. Peirce, C.S.: Writings of Charles S. Peirce: A Chronological Edition, vol. 7, ed. by Moore, E.C., Kloesel, C.J.W., et al. Bloomington: Indiana University Press. Cited as W followed by volume and page number (1982)
23. Pietarinen, A.V.: Peirce's diagrammatic logic in IF perspective. In: Blackwell, A.F., Marriott, K., Shimojima, A. (eds.) Diagrams 2004. LNCS (LNAI), vol. 2980, pp. 97–111. Springer, Heidelberg (2004). doi:10.1007/978-3-540-25931-2_11
24. Pietarinen, A.-V.: Signs of Logic: Peircean Themes on the Philosophy of Language, Games, and Communication. Springer, Dordrecht (2006)
25. Pietarinen, A.-V.: Moving pictures of thought II: graphs, games, and pragmaticism's proof. Semiotica **186**, 315–331 (2011)
26. Prior, A.N.: The algebra of the copula. In: Moore, E., Robin, R. (eds.) Studies in the Philosophy of Charles Sanders Peirce, pp. 79–94. The University of Massachusetts Press, Amherst (1964)
27. Roberts, D.D.: Existential graphs and natural deduction. In: Moore, E., Robin, R. (eds.) Studies in the Philosophy of Charles Sanders Peirce, pp. 109–121. The University of Massachusetts Press, Amherst (1964)
28. Russell, B.: Sur la logique des relations avec des applications á la théorie des séries. Revue de mathématiques/Rivista di Matematiche **7**, 115–148 (1901)
29. Schröder, E.: Vorlesungen über die Algebra der Logik, vol. 1. B. G. Teubner, Leipzig (1890)
30. Sowa, J.: Peirce's contributions to the 21st century. In: Schärfe, H., Hitzler, P., Øhrstrøm, Peter (eds.) ICCS-ConceptStruct 2006. LNCS (LNAI), vol. 4068, pp. 54–69. Springer, Heidelberg (2006). doi:10.1007/11787181_5
31. Turquette, A.: Peirce's icons for deductive logic. In: Moore, E., Robin, R. (eds.) Studies in the Philosophy of Charles Sanders Peirce, pp. 95–108. The University of Massachusetts Press, Amherst (1964)

On Semantic Gamification

Ignacio Ojea Quintana(✉)

Columbia University, New York City, USA
ignacio.ojea@columbia.edu

Abstract. The purpose of this essay is to study the extent in which the semantics for different logical systems can be represented game theoretically. I will begin by considering different definitions of what it means to *gamify* a semantics, and show completeness and limitative results. In particular, I will argue that under a proper definition of gamification, all finitely algebraizable logics can be gamified, as well as some infinitely algebraizable ones (like Łukasiewicz) and some non-algebraizable (like intuitionistic and van Fraassen supervaluation logic).

1 Introduction

1.1 Logic Gamification

The present work builds on the well established work on game semantics for classical logic developed by Jaakko Hintikka [5,6], and independently by Rohit Parikh [13]. It is embedded in a research line that seeks to build formal connections between logic and game theory, systematically developed in van Benthem [18]. Its contribution amounts extending some of those results to non-classical logics, and to provide an answer to the general question of which semantics can be represented game-theoretically.

In the past few years there have been several developments in game semantics for many valued logics, for example by Fermüller [2,3]. In particular there are several well studied applications on games in Łukasiewicz-style (fuzzy) logics; Mundici [11] provides an alternative semantics for finite-valued Łukasiewicz logics in terms of Ulam's games and Cintula and Majer [1] develop an approach similar to what is going to be done here. Here I will briefly discuss the differences between my approach and Fermüller's, and provide a justification for my account.

But the central issue in this essay is to clearly define and discuss what it means for a semantics to be *gamifiable*, and to show that under an appropriate definition (a) all finitely-algebraizable logics are gamifiable, and (b) some non-algebraizable logics are gamifiable.

1.2 The Basic Case

Perfect information games in extensive form are trees whose nodes are possible states and turns for the players, arrows from a node to its children represent

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 183–197, 2017.
DOI: 10.1007/978-3-662-54069-5_14

the available moves or actions that the player has at that node. A strategy for a player i is a function that assigns a move at each node corresponding to a turn for player i. In terminal nodes payoffs are assigned for the players.

In Evaluation Games for Classical Propositional logic, two players **V** and **F** (for *Verifier* and *Falsifier*) dispute over the truth value of a formula ϕ of a language $\mathcal{L}$ in some model **M**. To avoid unnecesary complications, I assume that $\mathcal{L}$ is Classical a Propositional language. It is possible to assign a game $G^{\mathbf{M}}_{\phi}$ to each pair $< \phi, \mathbf{M} >$ of formulas and Classical Propositional models in the following way:

- If $\phi = p$ for atomic sentence p, then $G^{\mathbf{M}}_{p}$ is a single (terminal) node tree in which **V** wins if $\mathbf{M} \vDash p$ and **F** wins otherwise.
- If $\phi = \neg\alpha$, then $G^{\mathbf{M}}_{\phi}$ is $G^{\mathbf{M}}_{\alpha}$, with turns and win-lose markings reversed.
- If $\phi = \alpha \vee \beta$, then $G^{\mathbf{M}}_{\phi}$ is a tree that starts with a node which has $G^{\mathbf{M}}_{\alpha}$ and $G^{\mathbf{M}}_{\beta}$ as its only children and that it is a turn for **V**. The basic idea is that she decides with which subformula to continue the game.
- Finally, if $\phi = \alpha \wedge \beta$, then $G^{\mathbf{M}}_{\phi}$ is a tree that starts with a node which has $G^{\mathbf{M}}_{\alpha}$ and $G^{\mathbf{M}}_{\beta}$ as its only children and that it is a turn for **F**. The basic idea is that he decides with which subformula to continue the game.

The point of such assignment is that the following bridging result holds:

Proposition 1 (Success for Classical Propositional Logic). *For all formulas ϕ and Propositional models* **M***: **V** has a winning strategy in $G^{\mathbf{M}}_{\phi}$ if and only if* $\mathbf{M} \vDash \phi$.

The well-known result is due to Hintikka and it generalizes to first order logic. A first intuitive definition of what it means for a semantic to be *gamifiable* can be generalized from this result.

Definition 1 (Semantic gamification - intuitive). *We start with a logic* **L** *and a semantics $\mathcal{S}$ for that logic that assigns truth values to the formulas in the language of* **L**. *We say $\mathcal{S}$ is* intuitively gamifiable *if there is a game theoretic representation $G^{\mathcal{S}}$ and a game-theoretic condition (expressed using a solution concept) $\mathcal{C}$ such that for all formulas ϕ in the language, $\mathcal{S}$ assigns certain truth v to ϕ if and only if the condition $\mathcal{C}$ applies to the game theoretic representation of the formula $G^{\mathcal{S}}_{\phi}$.*

1.3 Structure of the Essay

In the next section I will discuss gamification for finitely algebraizable logics. In particular, I will present a hierarchy of notions of *gamification* and show the extent to which those logics can be gamified. Also, I will discuss some of the philosophical and technical aspects of those definitions.

In section three I will discuss non-finitely algebraizable logics, in particular intuitionistic and supervaluationist. The purpose of this section is to show that game semantics can be viewed as more general than the standard approaches to logics for semantics.

The last section includes concluding remarks on the significance of the results.

2 Finitely Algebraizable Logics

2.1 Logical Matrices

Logical Matrices were first introduced by Łukasiewicz and Tarski [9] in the 1920's as a general concept that was implicitly used in the work of other logicians. The reader can refer to [4,16] for a more advanced treatment than the one given here. The basic idea is a generalization of the Boolean Algebra underlying truth values in Classical Logic. Formulas are assigned truth values in the domain of the algebra and connectives are interpreted as the algebraic operations over those truth values.

Given a propositional language $\mathcal{L}$, a $\mathcal{L}-$**matrix** is a pair $< \mathbf{A}, F >$ where $\mathbf{A}$ is an algebra of type-$\mathcal{L}$ with universe $\mathcal{A}$, and $F \subseteq \mathcal{A}$; where F is the set of designated values. An assignment h is an homomorphism from the algebra of formulas $\mathbf{Fm}$ to the algebra $\mathbf{A}$ of the same $\mathcal{L}$-type $[h \in Hom(\mathbf{Fm}, \mathbf{A})]$. Here the elements of the algebra serve as the truth-values of the semantics. One of the key features here is compositionality. h starts by assigning elements of $\mathcal{A}$ to the set Var of propositional variables and can be extended to all of $\mathcal{L}$ by interpreting operations in the language as operations in the algebra:

- $h(p_i) = a_i$, where $p_i \in Var$ and $a_i \in \mathcal{A}$.
- $h(\neg\phi) = \neg^{\mathbf{A}} h(\phi)$.
- $h(\phi * \psi) = h(\phi) *^{\mathbf{A}} h(\psi)$, for any diadic connective $*$.

The notion of **model** is the same as before. A logic $\mathbf{L}$ in the language $\mathcal{L}$ is said to be *complete relative to a class of* $\mathbf{L}$*-matrices* $\mathbf{M}$, if all the elements of $\mathbf{M}$ are models of $\mathbf{L}$ and for every $\Gamma \cup \{\phi\} \subseteq Fm$ such that $\Gamma \nvdash_{\mathbf{L}} \phi$ there is a matrix $< \mathbf{A}, F > \in \mathbf{M}$ and $h \in Hom(\mathbf{Fm}, \mathbf{A})$ such that $h[\Gamma] \subseteq F$ but $h(\phi) \notin F$. If this is the case, then it is said that $\mathbf{M}$ is a *matrix semantics for* $\mathbf{L}$, or that $\mathbf{M}$ is *strongly characteristic for* $\mathbf{L}$. In particular, if $\mathbf{M}$ is a singleton with matrix M, then M is the characteristic matrix of $\mathcal{S}$.

Definition 2 (Finitely algebraizable). *A logic* $\mathbf{L}$ *is finitely algebraizable if it is complete relative to a class of* finite $\mathbf{L}$*-matrices* $\mathbf{M}$.

2.2 Games

In this work we are interested in a very restricted class of games: two player perfect information extensive games of finite depth [and in almost all cases, strictly competitive or zero-sum games]. As before the players are $\mathbf{V}$ and $\mathbf{F}$. We will introduce the basic notions following [10,12]; where the reader should turn for a more elaborate presentation.

An *extensive game model* is a tree $G =< S, R, turn, \mathcal{V} >$ with a set of state-nodes S and a family R of binary transition relations for the available moves, pointing from parent to daughter nodes. R is assumed here to be well-founded[1] in

[1] Since R is well founded, branches of the trees have only finite depth.

that there is no infinite sequence $< a_1, a_2, ... >$ of nodes such that $< a_i, a_{i+1} > \in M$ for all $i \in N$. *turn* is a function that assigns players to non-terminal nodes, indicating the player whose turn it is. $\mathcal{V}$ is a function that assigns utility values for players at all terminal nodes, but possibly also to any other node.[2]

A *strategy* for player i is a function s_i that assigns at each of i's turns one of the available actions. A *mixed strategy* for a player i is a function $\sigma_i : S_i \rightarrow [0, 1]$ which assigns a probability $\sigma_i(s_i) \geqslant 0$ to each pure strategy $s_i \in S_i$, satisfying that $\sum_{s_i \in S_i} \sigma_i(s_i) = 1$.

Given a set of players $I = \{1, ..., n\}$, a *pure strategy profile* is an n-tuple $< s_1, ..., s_n >$ where each s_i is a pure strategy for player i. Each pure strategy profile is associated with a terminal node in the game model, the one that would be reached if players played the strategy in the profile. Furthermore, given a pure strategy profile $< s_1, ..., s_n >$, $\mathcal{V}_i(< s_1, ..., s_n >) =_{df} \mathcal{V}_i(a)$, where a is the terminal node of that strategy profile and $\mathcal{V}_i$ is the utility for any player i. The payoff of a (possibly mixed) strategy profile $< \sigma_1, ..., \sigma_n >$, $\mathcal{V}_i(< \sigma_1, ..., \sigma_n >) = \sum_{< s > \in S} [\sigma_1(s_1)...\sigma_1(s_n)] \mathcal{V}_i(< s_1, ..., s_n >)$.

The solution concept that we will be using in almost all cases is that of Nash Equilibrium: A strategy profile $< \sigma_1, ..., \sigma_n >$ is a *Nash equilibrium* if and only if for any player $i \in \{1, ..., n\}$ and any strategy $\sigma_i' \neq \sigma_i$ for that player, $\mathcal{V}_i(< \sigma_1, ..., \sigma_i, ..., \sigma_n >) \geqslant \mathcal{V}_i(< \sigma_1, ..., \sigma_i', ..., \sigma_n >)$. The insight behind Nash Equilibrium is that unilateral deviation is not profitable. Once the strategy profile is reached, no player has an incentive to change strategies given the other player's strategic choices are fixed. Yet a particular subset of the Nash Equilibra will be used here, namely those obtained by the Backward Induction procedure.

2.3 Strong Gamification

When evaluation games for Classical Propositional logic were introduced before, there was an implicit function **game** that assigned extensive game trees of the ones just presented to formulas in $\mathcal{L}$ in some model $\mathbf{M}$; so that ϕ in $\mathbf{M}$ got assigned to $G_\phi^{\mathbf{M}}$. This way of presenting the evaluation games followed van Benthem in [17] and Parikh in [13,14].

The simple generalization proposed here requires us to drop the model dependence of the function, so that each formula $\phi \in \mathcal{L}$ gets an *game form*, a tree G_ϕ in all which terminal nodes $< p_i >$ corresponding to atomic sentences p_i have no assigned payoffs for the players. We later define $\mathcal{V}$ in a way that assigns members of the relevant (non-Classical) matrix to *terminal* nodes - but can be extended to other nodes. In general, given a game G, $\mathcal{V}(G)$ is the payoff that *Verifier* gets in the (relevant) equilibria of G.[3]

In detail, state-nodes of the trees are members of S and are denoted here with tuples $< \phi >$, where $\phi \in \mathcal{L}$. It is useful to reformulate the definition of $\mathbf{game}(\phi) = G_\phi$:

[2] A further assumptions is that there is complete and perfect information.

[3] For the games considered, it is not hard to show existence of equilibria as well as uniqueness of payoff under all equilibria.

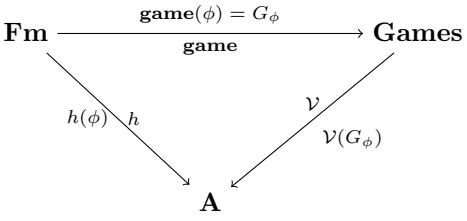
- G_{p_i} is a single node tree $< p_i >$, which can be seen as a test or payoff gaining game.
- $G_{\neg\phi}$ is G_ϕ with turns reversed, replacing each terminal atomic node $< p_i >$ by a node $< \overline{p_i} >$ *and vice versa*. Also, formulas $< \phi >$ in game nodes are syntactically dualized, interchanging conjunctions and disjunctions.
- $G_{\phi\vee\psi}$ is the disjoint union of two game trees G_ϕ and G_ψ put under a common root node $< \phi \vee \psi >$ that is a turn for $\mathbf{V}$.
- $G_{\phi\wedge\psi}$ is the disjoint union of two game trees G_ϕ and G_ψ put under a common root node $< \phi \wedge \psi >$ that is a turn for $\mathbf{F}$.

It is worth noticing that in this definition G_ϕ is generated solely from the syntactic structure of ϕ.

Let us go back briefly to the *alethic* or Model-theoretic approach to logic. Part of the gist of it is that we are able to model our natural or intuitive understanding of the connectives that appear in the formula algebra $\mathbf{Fm}$ with operations in our modeling algebra $\mathbf{A}$. This was captured by the fact that for any assignment (homomorphism) $h : \mathbf{Fm} \to \mathbf{A}$, $h(\neg\phi) = \neg^{\mathbf{A}} h(\phi)$ and $h(\phi * \psi) = h(\phi) *^{\mathbf{A}} h(\psi)$. Thus the algebra can successfully represent the alethic structure that we want it to embody. In the *pragmatic* or game-theoretic approach to logic, we want to have a relation of the same sort between the games and some underlying algebra. This will be captured by analogous principles:

- $\mathcal{V}(G_{p_i}) = a_i$, where g_i is an atomic game and $a_i \in \mathcal{A}$.
- $\mathcal{V}(G_{\neg\phi}) = \neg^{\mathbf{A}}\mathcal{V}(G_\phi)$.
- $\mathcal{V}(G_{\phi * \psi}) = \mathcal{V}(G_\phi) *^{\mathbf{A}} \mathcal{V}(G_\psi)$ for any diadic connective $*$.

Furthermore, in principle nothing ensures that the algebraic operation will coincide with our strategic intuitions and theories about how games are resolved (i.e. its equilibria). Conversely, it should not be clear *prima facie* that concepts in game theory and game structures function the same way as algebraic transformations. Yet, at least for some algebraic structures we know that the relation holds. The overall project is then:



Given a formula algebra $\mathbf{Fm}$, an underlying algebra $\mathbf{A}$ and $h \in Hom(\mathbf{Fm}, \mathbf{A})$, the central purpose of evaluation games is to provide a translation function **game** and a payoff function $\mathcal{V}_h$ so that for any formula $\phi \in \mathbf{Fm}$:

$$h(\phi) = \mathcal{V}^h \circ \mathbf{game}(\phi)$$

Definition 3 (Strong Semantic Gamification). *Begin with a logic* **L** *and a semantics* $\mathcal{S}$ *for that logic. We say* $\mathcal{S}$ *is* strongly gamifiable *if for each formula* ϕ *in the language and each assignment* $h_{\mathcal{S}}$ *there is (a) a game theoretic translation* G^{ϕ}*, (b) a payoff assignment* $\mathcal{V}_h$ *to* G^{ϕ} *that is defined in terms of* h *and (c) a game-theoretic condition (solution concept)* $\mathcal{C}$ *such that: For all formulas* ϕ*, an assignment* $h_{\mathcal{S}}$ *assigns certain truth* v *to* ϕ *if and only if the condition* $\mathcal{C}$ *applies to the game* G^{ϕ} *with payoffs determined by* $\mathcal{V}_h$*.*

I will now overview a few results that show that some many valued logics are *strongly* gamifiable. Since the precise cases considered here are not the main focus of the present essay, I will only provide a superficial presentation of each case. Nevertheless, the reader is invited to read some of the proofs in the appendix to get a gist of the basic techniques used here.

2.4 Strong Kleene

We start with Kleene's 3-valued system developed in [7,8] because generalizing evaluation games for it is straightforward. The set of truth values is $\mathcal{K} = \{1, \frac{1}{2}, 0\}$, where "1" codes truth, "0" codes falsity and "$\frac{1}{2}$" codes undefined. The operations $\neg^{\mathbf{K_3}}, \vee^{\mathbf{K_3}}, \wedge^{\mathbf{K_3}}$ are defined in analogy to Classical logic: (a) $\neg^{\mathbf{K_3}} x = 1 - x$, (b) $x \vee^{\mathbf{K_3}} y = max\{x, y\}$ and (c) $x \wedge^{\mathbf{K_3}} y = min\{x, y\}$.

In order to develop evaluation games for Kleene's 3-valued system I need, given a homomorphism h, a translation function G (or **game**) and an evaluation function $\mathcal{V}^h$ that assigns payoffs to terminal nodes of those trees and values to complex game trees using a solution concept. The translation function is the same as for the classical case. The valuation function $\mathcal{V}^h$ needs first to assign members of $\mathcal{K} = \{1, \frac{1}{2}, 0\}$ to the terminal nodes $< p_i >$ and $\overline{< p_i >}$ such that the payoff of *both* players are specified. $\mathcal{V}^h(G_{\phi})$ is the payoff that *Verifier* gets in the equilibria of the game G_{ϕ}; and since I am considering strictly competitive games, the payoff that *Falsifier* gets will be $1 - \mathcal{V}^h(G_{p_i})$.

The valuation for terminal nodes is:

- $\mathcal{V}^h(G_{p_i}) = h(p_i)$. Hence *Verifier* gets $h(p_i)$ and *Falsifier* gets 1-$h(p_i)$.
- $\mathcal{V}^h(\overline{G_{p_i}}) = 1 - h(p_i)$. Hence *Verifier* gets $1 - h(p_i)$ and *Falsifier* gets $h(p_i)$.

Proposition 2 (Success for Strong Kleene). *Given the matrix* $\mathbf{K_3}$ *and arbitrary assignment* h*, for all formulas* $\phi \in \mathcal{L}_{\mathcal{K}}$*:* $h(\phi) = x$ *if and only if* $\mathcal{V}^h(G_{\phi}) = x$ *[i.e. in all the Nash Equilibria in* G_{ϕ} Verifier *gets a payoff of* x*].*

The proof of the proposition is included in the appendix. Furthermore, the observant reader might have noticed that there is nothing essential in the fact that only *three* truth-values were considered. What is crucial is that the truth values are linearly ordered and the (Kleene) operations correspond to *max*, *min* and dualization. Then any logic of this form, with finite or infinite truth values, can be modeled analogously with a strictly-competitive two player game.

2.5 Gamification and Some Results

A natural question is whether all finitely algebraizable logics can be strongly gamified. I do not have an answer to this. Nevertheless, it is possible to show that all finitely algebraizable logics can be gamified, under a weaker notion of gamification.

Definition 4 (Semantic Gamification). *Begin with a logic* **L** *and a semantics* $\mathcal{S}$ *for that logic. We say* $\mathcal{S}$ *is* gamifiable *if for each formula* ϕ *in the language and each assignment* $h_{\mathcal{S}}$ *there is a game* G_h^{ϕ} *whose* structure *and payoffs* depend on h *and (b) a game-theoretic condition (solution concept)* $\mathcal{C}$ *such that: For all formulas* ϕ*, an assignment* $h_{\mathcal{S}}$ *assigns certain truth* v *to* ϕ *if and only if the condition* $\mathcal{C}$ *applies to the game* G_h^{ϕ}.

The crucial difference here is that formulas are *not* mapped to game *forms*, but rather the mapping goes from formulas and assignments to completely specified games. In particular, the same formula can be mapped to different games under different assignments (and of course different matrices).

The strategy adopted here to show that all finitely algebraizable logics can be gamified is indirect. The first step consists in showing that Post logics are gamifiable. The second, to argue that this is sufficient given the truth-functional completeness of those logics.

2.6 Post and Truth-Functional Completeness

In 1921 [15] Emil Post presented a finitely many valued logic and *showed that it is truth-functionally complete* i.e. that all truth functions $f : \mathcal{A}^n \rightarrow \mathcal{A}$ are expressible in terms of the truth functions corresponding to the connectives provided by that logic. Post's intepretation of the disjunction and conjunction is similar to that of Strong Kleene and Łukasiewicz, *max* and *min* respectively. The most salient feature of Post logic is Post's negation $\sim$; so let $\mathcal{L}_{\sim}$ by $\mathcal{L}$ augmented with that connective. Its interpretation -when $\mathcal{A} = \{0, ..., n\}$ - is the following:

– $h(\sim \phi) = h(\phi) - 1 (mod\ n + 1)$.

Proposition 3 (Success for Post). *For every formula* ϕ *in the language of Post, truth value* v *and assignment* h*:* $h(\phi) = v$ *if and only if in all the Nash Equilibria in* G_{ϕ}^{h} Verifier *gets a payoff of* v.

The proof of this result, although innelegant and tedious, is included in the Appendix.

Any matrix $\mathbf{M} =< \mathbf{A}, \mathbf{F} >$ with finite universe $\mathcal{A}$ can be represented in a Post matrix of size $|\mathcal{A}|$, making use of the fact that it is truth-functionally complete. This is done in two steps. First, by corresponding each truth value in the matrix with a truth value in the Post logic. Second, the interpretation that matrix **M** gives to each connective is nothing more than a truth function; which by truth functional completeness can be captured in the Post logic by some composition

of Post connectives. In a nutshell, providing a game semantic for Post logic is virtually the same as providing a game semantic for any finite matrix.

A similar result was given by Fermüller [3] in 2013, but with a different approach. There Fermüller associates games with *signed* formulas, which capture the idea that Verifier asserts a certain truth value for the formula at hand. For example, the expression '$v : \phi$' stands for Verifier's claim that the formula ϕ has truth value v in the relevant assignment. His basic idea is to have win-lose games in which Verifier makes the assertion that ϕ has certain truth value and Falsifier contests that assertion. In this way, his result are also expressed in terms of winning strategies, rather than Nash Equilibria or Backwards Induction solutions. This is, h assigns v to ϕ if and only if Verifier has a winning strategy in the game corresponding to $v : \phi$.

3 General Gamification

So far the focus of the paper has been on finitely-algebraizable logics, but what about other kinds of logics? Allow me to slightly generalize the definition of semantic gamification so that formulas in a semantics are represented by a *set* of games, rather than a single game.

Definition 5 (General Semantic Gamification). *Begin with a logic* **L** *and a semantics* $\mathcal{S}$ *for that logic. We say* $\mathcal{S}$ *is* gamifiable *if for each formula* ϕ *in the language and each assignment* $h_{\mathcal{S}}$ *there is a* set *of games* G_h^{ϕ}*, each of whose* structure *and payoffs* depend on h *and (b) a game-theoretic condition (solution concept)* $\mathcal{C}$ *such that: For all formulas* ϕ*, an assignment* $h_{\mathcal{S}}$ *assigns certain truth* v *to* ϕ *if and only if the condition* $\mathcal{C}$ *applies to all the games in* G_h^{ϕ}*.*

Under this simple generalization, it is not hard to show that some non-algebraizable logics, such as Intuitionistic and Supervaluationistic, are gamifiable in general.

3.1 Supervaluationist

Supervaluationist logic was developed by Van Fraassen [19,20] to treat issues of vagueness while satisfying some classical logic principles. The basic idea to evaluate a formula is to start with a partial assignment with three truth values and consider all the classical extensions of that assignment. If the formula is true in all its classical extensions, then it is true in the supervaluation; similarly for falsehood. If the formula is true in some extensions and false in others, then it gets an intermediate value.

More formally, an *initial* truth-value assignment is any function h such that for $h(p_i) \in \{0, \frac{1}{2}, 1\}$ for all propositional variables p_i and that is extended to all formulas using the Strong Kleene compositional rules. A *classical extension* h' to a initial truth-value assignment h $[h \leqslant h']$ is a function such that (a) $h'(p_i) \in \{0, 1\}$ for all propositional variables p_i and extends to all formulas as

expected, and (b) for all $p_i \in Var$, if $h(p_i) \in \{0,1\}$, then $h(p_i) = h'(p_i)$. A supervaluation induced by an assignment h is a function f_h such that for all $\phi \in \mathcal{L}$: (a) $f_h(\phi) = 1$ if and only if for all classical extensions h' of h, $h'(\phi) = 1$; (b) $f_h(\phi) = 0$ if and only if for all classical extensions h' of h, $h'(\phi) = 0$; and (c) $f_h(\phi) = \frac{1}{2}$ otherwise. One interesting aspect of supervaluatinist logic is that it is not compositional. For example, if $h(\phi) = h(\psi) = \frac{1}{2}$, then $f_h(\phi \vee \psi) = 1$ if $\psi = \neg\phi$ but $f_h(\phi \vee \psi) = \frac{1}{2}$ if $\phi = p_1$ and $\psi = p_2$.

The basic idea of gamifying supervaluationist logic involves mapping each formula and assignment pair (ϕ, h) to a *set* of classical games, namely those classical games that correspond to the classical extensions of h.

Proposition 4 (Success for Supervaluation). *Given an arbitrary assignment h and a supervaluational semantics f_h, for all formulas ϕ: (a) $f_h(\phi) = 1$ if and only if* $\mathbf{V}$ *has a winning strategy in every game in G^h_ϕ, (b) $f_h(\phi) = 0$ if and only if* $\mathbf{F}$ *has a winning strategy in every game in G^h_ϕ, and (c) $f_h(\phi) = \frac{1}{2}$ otherwise.*

3.2 Intuitionistic Logic

Intuitionistic logic requires no introduction, and I will presume the reader is familiar with the Kripke semantics for intuitionistic logic. The only subtelty that is involved in gamifying the Kripke semantics for intuitionistic logic is that given a structure $\mathbf{K}$ of partially ordered nodes, the translation function associates to each formula-node pair (ϕ, k) a game $G_{(\phi,k)}$. Once again, the shape of the game depends on the structure provided by the Krike *frame*. The basic idea is that games represent what is for the formula to be true in that node. As an example, consider the usual clauses for the conditional and negation:

- A node k *forces* $\phi \rightarrow \psi$ if, for every $k' \geqslant k$, if k' *forces* ϕ then k' *forces* ψ.
- A node k *forces* $\neg\phi$ if, for **no** $k' \geqslant k$ does k' *forces* ϕ.

Then the translation functions are the following:

- The game corresponding to $(\phi \rightarrow \psi, k)$, $G_{(\phi\rightarrow\psi,k)}$ has a root node that is a move for Falsifier whose children are the games $G_{(\sim\phi\vee\psi,k')}$[4], for all $k' \geqslant k$.
- The game corresponding to $(\neg\phi, k)$, $G_{(\neg\phi,k)}$ has a root node that is a move for Falsifier whose children are the games $\overline{G_{(\phi,k')}}$, for all $k' \geqslant k$. Here $\overline{G_{(\phi,k')}}$ is just like $G_{(\phi,k')}$ but with roles and payoffs switched [just like in the classical negation clause].

The next obvious step is to match, for each formula ϕ the set of all games $G_{(\phi,k)}$ for all k in the Kripke structure $\mathbf{K}$. In this way we obtain $G^{\mathbf{K}}_\phi$, the set of games corresponding to ϕ in $\mathbf{K}$.

Proposition 5 (Success for Intuitionism). *Given an arbitrary Kripke structure* $\mathbf{K}$*, for all formulas ϕ: (a)* $\mathbf{K} = \mathbf{1}$ *if and only if* $\mathbf{V}$ *has a winning strategy in every game in $G^{\mathbf{K}}_\phi$.*

[4] Here $\sim$ is just classical negation.

As far as I know, Proposition 7 is new - although it is a natural application of dynamic reasoning.

4 Conclusion and Discussion

To gamify a semantics means, intuitively, to provide a game-theoretic representation of it. The purpose of this essay was to clarify different notions of gamification and to study the extension to which different propositional logical systems can be gamifiable. I presented three notions of gamification - weak, basic and general. I argued that several finitely-algebraizable logics *strongly* gamifiable, but it is still open whether all of them are. In the next section I presented a result that shows that all finitely-algebraizable logics are gamifiable. The last section shows that even non-algebraizable and non-compositional logics are easily gamifiable if we relax the condition of uniqueness and allow formulas to be represented as sets of games.

So far, I have not provided any philosophical account of what we learn about a semantics by knowing if it is, or not, gamifiable. That was not the main purpose of the essay, but a few words are worth saying. Hintikka's original motivation to provide a game semantics had to do more with the pragmatic nature of assertion, or the meaning of conditionals, than with purely logical concerns. The purpose here was to advance an approach to logic that is neither semantic nor syntactic, but rather pragmatic. Valuation functions for formulas usually express the truth values that formulas have under some assignment or model *a la* Tarski, so that -in general- the truth value of a compound expression depends in some way on the truth value of its components. When providing game-theoretic semantics, I intended to avoid *alethic* considerations and ideas and substitute them by instrumental, pragmatic or operational concepts. The hope is that furthering this approach will provide us with more insights about the relation between Theoretical Reason - captured in our logical systems - and Practical Rationality - captured in our game and decision theoretic ideas. For a logical system to be gamifiable, then, means that its Theoretical import can be captured strategically.

To conclude, two questions and potential lines of research emerge from here. To begin, it would be interesting to answer whether all algebraizable logics can be strongly gamified. Furthermore, the converse problem for weak gamification is also interesting: Given a class of two-player games closed under some operations and with payoffs in a set V, whether there is a language $\mathcal{L}$ closed under some operations and a matrix-semantics $\mathbf{A}$ with assignment h such that (a) there is a function that translates games into formulas and (b) the (BI, Nash, etc.) solutions of the game correspond in some way to the value that its corresponding formula gets in h.[5]

[5] Notice here that nothing secures uniqueness of solutions for these games, so solving this problem might require generalizing the presented definition of matrix algebra in some way.

A Appendix: Proofs

Success for Strong Kleene. The proof of this proposition is by induction, and it is analogous to the traditional proof of Proposition 1. The atomic case is trivial and guaranteed by the definition of $\mathcal{V}$ in labeled and unlabeled terminal nodes [i.e. literals]. For complex expressions, assuming by Induction Hypothesis that Proposition 2 holds for all the subformulas, we need to ensure that: (a) $\mathcal{V}(G_{\neg\phi}) = \neg^{\mathbf{K_3}}\mathcal{V}(G_\phi)$; (b) $\mathcal{V}(G_{\phi\vee\psi}) = \mathcal{V}(G_\phi)\vee^{\mathbf{K_3}}\mathcal{V}(G_\psi)$; and (c) $\mathcal{V}(G_{\phi\wedge\psi}) = \mathcal{V}(G_\phi)\wedge^{\mathbf{K_3}}\mathcal{V}(G_\psi)$.

In the case of binary connectives we get the identity easily, since the Nash Equilibria are obtained by the Backward Induction procedure and the player's payoffs are such that *Verifier* prefers maximizing between $\mathcal{V}(G_\phi)$ and $\mathcal{V}(G_\psi)$, and *Falsifier* prefers minimizing between those two alternatives. The case for negation requires an observation:

Observation 1 (Mirroring of Pure Strategies and NE). *If s_V is a pure strategy for* Verifier *in G_ϕ, then s_V is a pure strategy for* Falsifier *in $G_{\neg\phi}$ [and vice versa]. Furthermore, given some payoff assignment $\mathcal{V}$ to the terminal nodes, if $< s_V, s_F >$ is a Nash Equilibrium in G_ϕ, then $< s_F, s_V >$ is a Nash Equilibrium in $G_{\neg\phi}$.*

A short proof of the observation is the following. Any pure strategy profile $< s_V, s_F >$ in G_ϕ is associated with a terminal node, the one that is reached by the path indicated by the strategies, with some payoffs $(x, 1-x)$ for *Verifier* and *Falsifier* respectively. Also, $< s_F, s_V >$ in $G_{\neg\phi}$ leads to the same node, but now with payoffs $(1-x, x)$. Notice that in $G_{\neg\phi}$ there was a *turn switch*; so $< s_F, s_V >$ is the profile where *Verifier* plays s_F and *Falsifier* plays s_V. If the *second* profile $< s_F, s_V >$ in game $G_{\neg\phi}$ is **not** a Nash Equilibrium, then at least one player, say *Falsifier*, can change the strategy to s'_V so that $< s_F, s'_V >$ terminates in a node with payoff $y > x$ for him [leaving s_F fixed]. But then *Verifier* can change her strategy in G_ϕ to also obtain a better payoff. Obviously, the argument is symmetric, and hence one profile is a Nash Equilibrium if and only if the other is.

With this observation, we get that $\mathcal{V}(G_{\neg\phi}) = 1 - \mathcal{V}(G_\phi)$, which is what we needed. □

Success for Post. It is easy to see that Post's negation does not satisfy De Morgan's properties in general. Yet, a weaker (partial) form of De Morgan's is satisfied:

Observation 2 (Partial De Morgan's for Post's Logic).

- $h(\sim(\phi\vee\psi)) = \begin{cases} h(\sim\phi\wedge\sim\psi) & \text{if } h(\phi) = 0 \neq h(\psi) \text{ or } h(\phi) = 0 \neq h(\psi) \\ h(\sim\phi\vee\sim\psi) & \text{if otherwise} \end{cases}$
- $h(\sim(\phi\wedge\psi)) = \begin{cases} h(\sim\phi\vee\sim\psi) & \text{if } h(\phi) = 0 \neq h(\psi) \text{ or } h(\phi) = 0 \neq h(\psi) \\ h(\sim\phi\wedge\sim\psi) & \text{if otherwise} \end{cases}$

For simplicity, I avoid this proof here.

Providing a game semantics for Post requires a slight change in methodology. In all the cases presented here, the function $\mathbf{game} : \mathcal{L}_{\sim} \rightarrow \mathcal{A}$ was defined *independently* of the *evaluation* function $\mathcal{V}$ for the games. In this way we had *game forms*. For disjunctions and conjunctions, nothing different is needed. Yet, in order to provide a game that corresponds to a formula that involves Post's negation, it is necessary to look at the truth values of the components. Notice then that for formulas ϕ *not* involving Post negation, we can rely in the success lemmas already shown, so that $h(\phi) = \mathcal{V} \circ \mathbf{game}(\phi)$. Hence in when defining **game** we can use this fact. Se we only need to show that the success holds for formulas that involve the negation. $\mathbf{game}(\sim \phi)$ is $\mathbf{game}(\phi)$ transformed inductively in the following way:

For terminal nodes, we need to generalize the $\overline{x}$ function that we had before because now 'bars' are cummulative. So we have a function in the exponent that tracks the amount of times terminal nodes were negated. Non negated terminal nodes $< p_i >$ are now replaced by $< p_i^0 >$. Give $\mathbf{game}(\phi)$, we start by replacing each terminal node $< p_i^k >$ with $< p_i^{k+1} >$. As expected, we stipulate that $\mathcal{V}(G_{p_i^k}) = [h(p_i) - k](mod\ n+1) = h(\sim ... \sim (p_i))$ where the negation is iterated k times. If $< \psi >$ is a non-terminal node and has children $< \alpha >$ and $< \beta >$, and if $h(\alpha) = n \neq h(\beta)$ or $h(\beta) = n \neq h(\alpha)$:

$$turn(< \psi >) = \begin{cases} \mathbf{V} & \text{if } turn(< \psi >) = \mathbf{F} \\ \mathbf{F} & \text{if } turn(< \psi >) = \mathbf{V} \end{cases}$$

Otherwise, turns are not changed.

The only thing that is really needed now is to show the case for Post-negated formulas, i.e. that $\mathcal{V}(\mathbf{game}(\sim \phi)) = [\mathcal{V}(\mathbf{game}(\phi)) - 1](mod\ n+1)$. In order to do this we need an observation. Notice that the tree corresponding to $\mathbf{game}(\sim \phi)$ and the tree coresponding to $\mathbf{game}(\phi)$ are the *same*, but the games are different in that turn assignments to non-terminal nodes and exponential markings in terminal nodes might have changed.

Observation 3. *Given trees* $\mathbf{game}(\sim \phi)$ *and* $\mathbf{game}(\phi)$*, for any node* $< \psi >_{\neg\phi}$ *in the former corresponding to a node* $< \psi >_{\phi}$ *in the latter we have that* $\mathcal{V} \circ \mathbf{game}(\psi_{\neg\phi}) = [\mathcal{V} \circ \mathbf{game}(\psi_{\phi}) - 1](mod\ n + 1)$.

The proof is by (Backwards) Induction. If $< p_i^k >$ is terminal, then the observation follows by definition. Say $< \psi >$ is a non-terminal node and has children $< \alpha >$ and $< \beta >$ such that $\mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi}) = [\mathcal{V} \circ \mathbf{game}(\alpha_{\phi}) - 1](mod\ n + 1)$ and $\mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi}) = [\mathcal{V} \circ \mathbf{game}(\beta_{\phi}) - 1](mod\ n + 1)$.

Case 1: $\mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi}) = n \neq \mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi})$ or $\mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi}) = n \neq \mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi})$. Suppose the former, the latter case is symmetrical. By inductive hypothesis, $\mathcal{V} \circ \mathbf{game}(\alpha_{\phi}) = 0 \neq \mathcal{V} \circ \mathbf{game}(\beta_{\phi})$. If $turn(< \alpha >_{\phi}) = \mathbf{V}$, $\mathcal{V} \circ \mathbf{game}(\psi_{\phi}) = max\{\mathcal{V} \circ \mathbf{game}(\alpha_{\phi}), \mathcal{V} \circ \mathbf{game}(\beta_{\phi})\} = \mathcal{V} \circ \mathbf{game}(\beta_{\phi})$. Also, by definition, $turn(< \alpha >_{\neg\phi}) = \mathbf{F}$, and then $\mathcal{V} \circ \mathbf{game}(\psi_{\neg\phi}) = min\{\mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi}), \mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi})\} = min\{n, \mathcal{V} \circ \mathbf{game}(\beta_{\phi}) - 1(mod\ n + 1)\} = \mathcal{V} \circ \mathbf{game}(\beta_{\phi}) - 1(mod\ n+1)$. So $\mathcal{V} \circ \mathbf{game}(\psi_{\neg\phi}) = [\mathcal{V} \circ \mathbf{game}(\psi_{\phi}) - 1](mod\ n+1)$. If $turn(< \alpha >_{\phi}) = \mathbf{F}$, $\mathcal{V} \circ \mathbf{game}(\psi_{\phi}) = min\{\mathcal{V} \circ \mathbf{game}(\alpha_{\phi}), \mathcal{V} \circ \mathbf{game}(\beta_{\phi})\} = 0 = \mathcal{V} \circ \mathbf{game}(\alpha_{\phi})$. Also, by definition, $turn(< \alpha >_{\neg\phi}) = \mathbf{V}$, and then

$\mathcal{V} \circ \mathbf{game}(\psi_{\neg\phi}) = max\{\mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi}), \mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi})\} = max\{n, \mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi})\} = n = \mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi}) = [\mathcal{V} \circ \mathbf{game}(\alpha_{\phi}) - 1](mod\ n + 1)$.

Case 2: Otherwise. Either (i) $\mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi}) = \mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi}) = n$ or (ii) $\mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi}) \neq n \neq \mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi})$. This proof is left to the reader for its simplicity. □

Success for Supervaluationism. The main consideration here is to translate each formula and assignment pair (ϕ, h) to a *set* of classical games. This is done in two steps. First, map each formula ϕ to its game *form* without any specified payoffs, using the original translation method. Second, consider in order the propositional letters p_i that appears in the game form and the value that h assigns to p_i. If $h(p_i) \in \{0, 1\}$, then assign the corresponding payoff to p_i and move to the next. If $h(p_i) = \frac{1}{2}$, then split the game into two games, one in which the payoff of p_i is (1,0) and another in which it is (0,1). Continue the procedure with all the games that were generated in the steps before.

It should be clear to the reader that the set of games obtained are the games corresponding to all of the classical extensions of h. Then the proposition follows by mere definition. □

Success for Intuitionism. Begin with a Kripke structure $\mathbf{K}$ of partially ordered nodes $\{k_i\}_{i \in I}$. Recall that this proof is just for the propositional case; nothing conceptually different is added for the first order case.

There is an atomic forcing relation defined for all nodes k such that for all propositional letters p_i, either k *forces* p_i [i.e. k makes p_i true, or the forcing relation is not defined for that node and propositional letter. This atomic forcing relation is subject to the constrain that if $k \leqslant k'$ and k forces p_i, then k' forces p_i. The extensio of the *forcing* relation to all formulas is the following: (a) A node k *forces* $\phi \wedge \psi$ if it forces ϕ and ψ; (b) A node k hforces $\phi \vee \psi$ if it forces ϕ or ψ; (c) A node k *forces* $\phi \rightarrow \psi$ if, for every $k' \geqslant k$, if k' *forces* ϕ then k' forces ψ; and (d) A node k *forces* $\neg\phi$ if, for **no** $k' \geqslant k$ does k' *forces* ϕ.

Given a formula and a node k in a Kripke structure $\mathbf{K}$, the translation functions are the following: (a) The game corresponding to (p_i, k) is a one node game with payoffs (1, 0) if k forces p_i and (0, 1) otherwise; (b) The game corresponding to $(\phi \wedge \psi, k)$, $G_{(\phi \wedge \psi, k)}$, consists of a root node for Falsifier with two subgames,$G_{(\phi,k)}$ and $G_{(\psi,k)}$; (c) The one corresponding to disjunction is as expected; (d) The game corresponding to $(\phi \rightarrow \psi, k)$, $G_{(\phi \rightarrow \psi, k)}$, has a root node that is a move for Falsifier whose children are the games $G_{(\sim\phi \vee \psi, k')}$, for all $k' \geqslant k$; and (e) The game corresponding to $(\neg\phi, k)$, $G_{(\neg\phi,k)}$ has a root node that is a move for Falsifier whose children are the games $\overline{G_{(\phi,k')}}$, for all $k' \geqslant k$. Here $\overline{G_{(\phi,k')}}$ is just like $G_{(\phi,k')}$ but with roles and payoffs switched [just like in the classical negation clause].

To show the result is sufficies to show that for any pair (ϕ, k), k forces ϕ if and only if Verifier has a winning strategy in $G_{(\phi,k)}$. The atomic case is trivial. So are the cases for conjunction, disjunction and *classical negation.* This is just the same proof as for classical logic. The case for the conditional is slightly more

complicated. It is worth noticing is that here $\sim$ refers to classical negation, and hence $\sim \phi \vee \psi$ is just code for the material conditional. We begin with $(\phi \to \psi, k)$, and suppose k forces $\phi \to \psi$. Then, in a nutshell, for every $k' \geqslant k$, k' forces $\sim \phi \vee \psi$. But then, by inductive hypothesis, Verifier has a winning strategy in every such $G_{(\sim\phi\vee\psi,k')}$. So it has a winning strategy in $G_{(\phi\to\psi,k)}$. Suppose Verifier has a winning strategy in $G_{(\phi\to\psi,k)}$. Then whichever choice Falsifier makes at the root node, Verifier still has a winning strategy. That means that for all $k' \geqslant k$, Verifier has a winning strategy in $G_{(\sim\phi\vee\psi,k')}$. By inductive hypothesis this just means that for every $k' \geqslant k$, if k' *forces* ϕ then k' *forces* ψ. Consider $(\neg\phi, k)$. Suppose k forces $\neg\phi$. Then there is **no** $k' \geqslant k$ such that k' *forces* ϕ. The game $G_{(\neg\phi,k)}$ has a root node that is a move for Falsifier whose children are the games $\overline{G_{(\phi,k')}}$, for all $k' \geqslant k$. Now, by (a minor extension of) Observation 2 - Mirroring of pure strategies and Nash Equilibria -, we know that for all such k' Falsifier has a winning strategy in $\overline{G_{(\phi,k')}}$ if and only if Verifier has a winning strategy in $G_{(\phi,k')}$, and viceversa. We also know, by inductive hypothesis, that Verifier has a winning strategy in $G_{(\phi,k')}$ if and only if k' forces ϕ. Since k forces $\neg\phi$, there is no $k' \geqslant k$ such that Verifier has a winning strategy in $G_{(\phi,k')}$. But all of the $G_{(\phi,k')}$ are two-player perfect information games, and therefore are determined. *Ergo*, Falsifier has a winning strategy in all those $G_{(\phi,k')}$. By Observation 2, Verifier has a winning strategy in all the $\overline{G_{(\phi,k')}}$. Hence, whichever move Falsifier makes in the root node of $G_{(\neg\phi,k)}$, it leads to a game won by Verifier. To conclude, Verifier has a winning strategy for $G_{(\neg\phi,k)}$. Now for the converse. Suppose Verifier has a winning strategy for $G_{(\neg\phi,k)}$. This just means that whatever $\overline{G_{(\phi,k')}}$ Falsifier chooses at the root node, Verifier has a winning strategy there. Therefore, again by Observation 2, Falsifier has a winning strategy in all the $G_{(\phi,k')}$ with $k' \geqslant k$. By inductive hypothesis, this just means that there is no $k' \geqslant k$ such that k' forces ϕ. □

References

1. Cintula, P., Majer, O.: Towards evaluation games for fuzzy logics. In: Majer, O., Pietarinen, A.V., Tulenheimo, T. (eds.) Games: Unifying Logic, Language, and Philosophy. Logic, Epistemology, and the Unity of Science, vol. 15, pp. 117–138. Springer, Dordrecht (2009)
2. Fermüller, C.G.: Dialogue games for many-valued logics - an overview. Stud. Logica. **90**(1), 43–68 (2008)
3. Fermüller, C.G.: On matrices, Nmatrices and Games. J. Logic Comput. (2013)
4. Hähnle, R.: Advanced many-valued logics. In: Gabbay, D.M., Guenthner, F. (eds.) Handbook of Philosophical Logic. Handbook of Philosophical Logic, vol. 2, pp. 297–395. Springer, Dordrecht (2001)
5. Hintikka, J.: Logic, Language Games, and Information. Clarendon Press, Oxford (1973)
6. Hintikka, J., Sandu, G.: Game-Theoretical Semantics (1997)
7. Kleene, S.C.: On notation for ordinal numbers. J. Symbolic Logic **3**(4), 150–155 (1938)
8. Kleene, S.C.: Introduction to Metamathematics: Bibliotheca Mathematica. Wolters-Noordhoff, Groningen (1952)

9. Łukasiewicz, J., Borkowski, L.: Selected Works: Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Co., Amsterdam (1970)
10. Mas-Colell, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford University Press, Oxford (1995)
11. Mundici, D.: Ulam's games, Łucasiewicz logic, and AFC*-algebras. Fundamenta Informaticae **18**, 151 (1993)
12. Osborne, M.J., Rubinstein, A.: A Course in Game Theory. MIT press, Cambridge (1994)
13. Parikh, R.: D-structures and their semantics. Not. AMS **19**, A329 (1972)
14. Parikh, R.: The logic of games and its applications. Ann. Discrete Math. **102**, 111–140 (1985)
15. Post, E.L.: Introduction to a general theory of elementary propositions. Am. J. Math. **43**(3), 163–185 (1921)
16. Urquhart, A.: Basic many-valued logic. In: Gabbay, D.M., Guenthner, F. (eds.) Handbook of Philosophical Logic. Handbook of Philosophical Logic, vol. 2, pp. 249–295. Springer, Dordrecth (2001)
17. van Benthem, J.: Logic games are complete for game logics. Studia Logica. Int. J. Symbolic Logic **75**, 183–203 (2003)
18. van Benthem, J.: Logic in Games. MIT Press, Cambridge (2014)
19. van Fraassen, B.C.: Presuppositions: supervaluations and free logic. In: Lambert, K. (ed.) The Logical Way of doing Things, pp. 67–92. Yale University Press (1969)
20. van Fraassen, B.C.: Singular terms, truth-value gaps, and free logic. J. Philos. **63**(17), 481–495 (1966)

Ancient Indian Logic and Analogy

Jeff B. Paris(✉) and Alena Vencovská

University of Manchester, Manchester M13 9PL, UK
{jeff.paris,alena.vencovska}@manchester.ac.uk

Abstract. B.K. Matilal, and earlier J.F. Staal, have suggested a reading of the 'Nyāya five limb schema' (also sometimes referred to as the Indian Schema or Hindu Syllogism) from Gotama's Nyāya-Sūtra in terms of a binary *occurrence* relation. In this paper we provide a rational justification of a version of this reading as Analogical Reasoning within the framework of Polyadic Pure Inductive Logic.

1 Introduction

In the Nyāya-Sūtra (∼150CE), Gotama discussed the structure of logical reasoning, offering a fundamental schema consisting of:

- statement of the thesis,
- statement of a reason,
- an example supporting the reason on the grounds of similarity to the present case,
- application of the above to the present case,
- conclusion.

B.M. Matilal [5] gives this 'time-honoured' illustration of the schema:

- There is fire on the hill.
- For there is smoke.
- (Wherever there is smoke, there is fire), as in the kitchen.
- This is such a case (smoke on the hill).
- Therefore it is so, i.e. there is fire on the hill.

It is often emphasised that this reasoning should be understood as occurring in the context of a debate, employed to persuade an opponent. Hence the apparently unnecessary number of steps; they each have a role. Considering the argument taken out of this context, it is commonly rephrased as

- (Wherever there is smoke, there is fire), as in the kitchen.

J.B. Paris—Supported by a UK Engineering and Physical Sciences Research Council (EPSRC) Research Grant.
A. Vencovská—Supported by a UK Engineering and Physical Sciences Research Council Research Grant.

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 198–210, 2017.
DOI: 10.1007/978-3-662-54069-5_15

- There is smoke on the hill. A
- Therefore there is fire on the hill.

This is clearly close to one of the Aristotelian syllogisms, but the *Indian Schema*, as we shall call it, can be reduced to it only at the cost of imposing the perspective of our contemporary deductive logic and rendering the example (almost[1] redundant. See for instance [2] for a collection of papers relating to attempts at understanding and formalising the schema in various ways. We have suggested in [7,8] that returning to the position where the example itself carries the weight of the evidence, somehow itself representing the universal implication, allows formulations of the argument within Pure Inductive Logic (to be introduced shortly) which can be justified as rational on the grounds of following from principles usually accepted in that subject as rational. When the example is so taken to encapsulate the evidence, the argument may be rephrased as[2]

- When there was smoke in the kitchen, there was fire.
- There is smoke on the hill. B
- Therefore there is fire on the hill.

– with the rider that the kitchen is a good example, which is taken to mean that the example captures all the relevant information.

Regarding this rider the Nyāya-Sūtra is a cryptic text and does not elaborate on its methodology. Nevertheless it is clear that the relationship here between smoke and fire is not simply taken to be contingent, coincidental, but fundamental, a concomitance, or even causal relationship, that cannot be otherwise. Being a good example then can be equated with capturing this link, rather as in mathematics we may give a 'proof by example'. Of course the problem in practice of precisely demarcating what we mean by this notion in general appears immensely difficult but fortunately that is not our problem in this short paper. We shall simply be interested in providing a justification for this inference on the grounds of its logical form alone.

2 The Pakṣa Formalisation

In our previous attempts [7,8] at formalising B we worked within Unary Predicate Logic, so using S, F, h and k in the obvious sense we employed $S(k)$ to express *There is smoke in the kitchen*, $F(h)$ to express *There is fire on the hill* etc. Within Pure Inductive Logic, B then becomes the assertion that, in the absence of any other pertinent information $S(h)$ and $S(k) \rightarrow F(k)$ provide grounds for accepting $F(h)$. In [7,8] we elaborated on the background and evidence for this reading of the schema (and so will not repeat ourselves here) and

[1] It has been suggested that under such a perspective, the role of the example may be to ensure existential import, see e.g. [4, p. 16].

[2] Notice that we are taking the evidence as a single instance of a kitchen, hence the switch from 'whenever' on line 1 to 'when'.

showed that such inference is indeed justified by certain well accepted rational symmetry principles of probability assignment and in consequence is itself rational.

Some authors however, notably Staal [12] and Matilal [5], have suggested that it is much closer to the Indian way of thinking to formalise the Indian Schema by employing a binary relation standing for 'occurring at': According to Staal in Indian logic an entity is never regarded in isolation but always considered as occurring at a locus, and the fundamental relation which underlies all expressions is that between an entity and its locus (*pakṣa*). Using R for this relation and f, s, h, k for $fire, smoke, hill$ and $kitchen$ respectively, B becomes the claim that, in the absence of any other pertinent information, $R(s,h)$ and $R(s,k) \rightarrow R(f,k)$ provide grounds for accepting $R(f,h)$. In this note we show that Pure Inductive Logic supports this version as a rational inference. To facilitate this we first need to summarize some necessary background from Pure Inductive Logic and briefly explain what this logic is attempting to elucidate.

3 Pure Inductive Logic

The framework for Pure Inductive Logic is Predicate Logic employing a language L with a finite set of relation symbols $R_1, \ldots, R_q$, countably many constants $a_1, a_2, a_3, \ldots$ and no function symbols nor equality.[3] SL denotes the set of sentences of L and $QFSL$ denotes the set of quantifier free sentences in SL.

A probability function on L is a function $w : SL \rightarrow [0,1]$ such that for any $\theta, \phi, \exists x\, \psi(x)$ from SL,

(i) If $\models \theta$ then $w(\theta) = 1$.
(ii) If $\theta \models \neg\phi$ then $w(\theta \vee \phi) = w(\theta) + w(\phi)$.
(iii) $w(\exists x\, \psi(x)) = \lim_{n\rightarrow\infty} w\left(\bigvee_{i=1}^{n} \psi(a_i)\right)$.

Any function w satisfying the above conditions has the properties we usually expect of probability (see [10, Proposition 3.1]), in particular if ψ logically implies θ then $w(\psi) \leq w(\theta)$.

Given a probability function w and $\theta, \phi \in SL$ with $w(\phi) > 0$ we define the conditional probability of θ given ϕ as usual by

$$w(\theta \,|\, \phi) = \frac{w(\theta \wedge \phi)}{w(\phi)}. \tag{1}$$

With a fixed $\phi \in SL$, $w(\phi) > 0$, the function defined by (1) is also a probability function.

The aim of Pure Inductive Logic (see for example [10]) is to investigate the logical or rational assignment of belief, as subjective probability,[4] in the

[3] In place of a_i we sometimes use other letters to avoid subscripts or double subscripts.
[4] In our view this makes it an obvious logic to investigate 'analogical arguments' where it is subjective probability which is being propagated by considerations of rationality.

absence of any intended interpretation. To explain this, consider a valid natural language argument, such as A where lines 1 and 2 are the premises and line 3 the conclusion. What we understand here by 'valid' is that this conclusion is true whenever the premises are true independently of the meaning of 'fire' 'smoke', 'kitchen' etc. In other words the conclusion is a *logical consequence* of the premises depending only on their form and not on the meaning or interpretation we give to 'fire, kitchen' etc.

Most natural language 'arguments' however are not so valid. Instead the premises only seem to provide some support for the conclusion rather than deem it categorically true. B is just such an example (though as Matilal points out at [4, p. 16] and [5, p. 197] contemporary scholars have commonly understood, and in consequence criticised, the Indian schema as aiming to render a *valid* conclusion). Nevertheless we can still investigate the question of how much of this support is logical or rational, depending only on the form of the premises and conclusion and not on the actual interpretation of 'fire', 'smoke' etc. So, just as Predicate Logic seeks to understand the notion of logical consequence by considering sentences of a formal language devoid of any particular interpretation, Pure Inductive Logic aims to address the more general issue of the logical or rational assignment of probabilities to sentences of a formal language (such as L above) in the absence of any particular interpretation. Note that this is indeed a more general issue since the support given by some evidence to a hypothesis arguably can be measured by the conditional probability of the hypothesis given the evidence. A hypothesis is a logical consequence of the evidence just when this support is total (probability 1) for *all* probability functions giving non-zero probability to the evidence.

A key requirement here is the *rationality* of the probability assignment (without it we would get no further than simply standard Predicate Logic). Whilst we may not know exactly what we mean by 'rational' here nevertheless there are, in this completely uninterpreted context, some constraints or principles governing this assignment that we feel are rational and should be enforced. The most basic of these is that since there is no reason to treat any one constant any differently from any other interchanging constants should not alter the assigned probabilities. Precisely, a rational probability function should satisfy:

The Constant Exchangeability Principle, Ex. *If $\theta \in SL$ and the constant symbol a_j does not appear in θ then $w(\theta) = w(\theta')$ where θ' is the result of replacing each occurrence of a_i in θ by a_j.*[5]

Similarly, in the absence of any particular interpretation there is no reason to treat a relation any differently from its negation. This leads to the rationality requirement on a probability function that it satisfy,

The Strong Negation Principle, SN. *$w(\theta) = w(\theta')$ where θ' is the result of replacing each occurrence of the relation symbol P_i in θ by $\neg P_i$.*

A word of caution here however. In our main theorem below we will formalise B in a predicate language and then, in this rarified set-up, argue that adopting

[5] This formulation of Ex is equivalent to that given in, say, [10], and avoids introducing extra notation.

the above principles Ex+SN, the conditional probability of the conclusion given the conjunction of the premises must be at least 1/2 (in fact strictly greater than 1/2 in all except exceptional circumstances). However for one to accept this conclusion requires one to agree, or allow for the sake of argument, that all the *relevant* information is given in the premises,[6] so that the actual interpretation ceases to matter and nothing essential is lost in the resulting formalisation as simply uninterpreted sentences of a predicate logic.[7] This is what we intend by a 'good example'.

4 The Main Result

The following theorem shows that when formalising the Indian Schema as in the section before last (that is, via a binary relation representing 'occurring at') and assuming that the condition on the example being a good one is taken to be that it represents all the relevant information, the Schema is at least as rational as Ex+SN. By this we mean that any probability function on L (where from now on L is the fixed language with single binary relation symbol R) satisfying Ex+SN gives probability at least 1/2 to fire occurring on the hill given (just) that smoke occurs on the hill, and that smoke in the kitchen implied fire in the kitchen.

Theorem 1. *Let w be a probability function on SL satisfying Ex+SN. Let h, k, s, f be distinct constants from amongst the $a_1, a_2, a_3, \ldots$.*

Then[8]

$$w(R(f,h) \,|\, R(s,h) \wedge (R(s,k) \rightarrow R(f,k))) \geq w(R(f,h) \,|\, R(s,h)) \geq 1/2. \qquad (2)$$

A few remarks are in order here. Firstly one might object that for the claimed justification one really requires the left hand term to be strictly greater than 1/2. In fact it is not difficult to show that if for a particular probability function w satisfying Ex+SN the left hand term of (2) - and hence also the middle term - does take the value 1/2 then this w must give the same value 1/2 to

$$w(R(k_{n+1}, s) \,|\, R(k_1, s) \wedge R(k_2, s) \wedge \ldots \wedge R(k_n, s)) \qquad (3)$$

for any number of 'kitchens' $k_1, k_2, \ldots, k_{n+1}$. In other words w must completely dismiss any *inductive influence*, informally, no matter if all the many kitchens seen in the past have been smokey this evidence amounts to nothing when it comes to the probability assigned to the next kitchen seen being also smokey. Thus to say that a purportedly rational w could fail to give the left hand side

[6] Of course one has a vast background knowledge about fires and kitchens etc. none of which is alluded to in these premises.

[7] In other words such reasoning is appropriate only in so far as one is content to apply a principle of ceteris paribus.

[8] To avoid problems with zero denominators we identify $w(\theta \,|\, \phi) \geq w(\psi \,|\, \eta)$ with $w(\theta \wedge \phi) \cdot w(\eta) \geq w(\psi \wedge \eta) \cdot w(\phi)$.

of (2) a value not strictly greater than 1/2 entails saying that it is rational to give (3) a value of 1/2 for all n, a not-inconsistent position to take but one which is hardly popular.

Of course one might wish that the support is not simply greater than 1/2 but actually substantially greater. However that can only be achieved by making additional assumptions beyond simply Ex+SN and currently we cannot envisage any such assumption which would avoid introducing a subjective element (just how much is 'substantially greater'?). This would seem to directly conflict with the idea of probabilities being assigned on purely rational or logical grounds.

A second remark here concerns our formalization of $\underline{\mathrm{B}}$. We have chosen to capture 'when there was smoke in the kitchen there was fire', by $R(s,k) \rightarrow R(f,k)$. Various other formulations are possible here, for example

$$R(s,k) \longleftrightarrow R(f,k), \qquad R(s,k) \wedge R(f,k).$$

In each case one can prove by the same methods that for a probability function satisfying Ex+SN conditioning $R(f,h)$ on this evidence together with $R(s,h)$ gives a value of at least one half, see Theorem 5 in the appendix. However in these cases it is currently open whether or not we can still interleaf $w(R(f,h) \mid R(s,h))$ as in Theorem 1.[9]

Thirdly, in case the reader might object here that the second inequality in (2) already gives the claimed 'support' for $R(f,h)$ from evidence $R(s,h)$ alone we are at pains to point out that by the assumption that all pertinent evidence has been included one cannot simply throw away the $R(s,k) \rightarrow R(f,k)$.

Finally we remark that Matilal's suggestion from [5, p. 197] that the reasoning in the Indian Schema may be more correctly understood as *inductive*, and for practical purposes providing knowledge of the real world, seems to us along the lines of the approach we have adopted here: we take the assignment of a probability of at least one half to the conclusion (equivalently, the conclusion being at least as probable as its negation) to be a justification for giving the conclusion the status of a *working assumption*.

5 Conclusion

We have shown that a version of the Indian Schema expressed in terms of the binary occurrence relation, as suggested by Staal and Matilal, is actually a consequence of the two of the central principles in Pure Inductive Logic, Constant Exchangeability and Strong Negation. By this we certainly do not wish to imply that the ancients were somehow aware of these principles (so this paper is not at all intended as a contribution to the *History* of Indian Logic). Rather we simply intend to point out that the everyday common senseness of the Indian Schema

[9] There are several other currently open problems with these, and other formulations (see for example [7–9]), in particular when the evidence involves multiple smokey kitchen, and the heterogenous non-smokey lakes, a case not treated at all in this paper.

does in fact have a *formal* justification as rational within the context of Pure Inductive Logic.

This paper has left much open for further research and investigation. For example in the way we formalise the schema in terms of the pakṣa, the concomitance, should it be implication, bi-implication or conjunction? Should 'hill' be thought of as a constant or a predicate etc., etc.? There is also the issue of the effect of heterogenous examples and of mixtures of multiple reasons of both kinds. We have already considered some of these questions in [7–9] within the context of Pure Inductive Logic but much remains unanswered. One advantage of using this framework is that following recent advances (see [10]) it is now equipped with some powerful representation theorems and a choice of attractive rational principles in addition to Ex+SN. Nevertheless there is the question whether this is the best framework in which to investigate such classical analogical reasoning, and certainly other have previously been proposed, for example [3,6,11]. Hopefully this short note will stimulate answers to these questions, not least from the Indian Logic community who clearly (unlike the present authors) have first hand access to the original texts and language.

Appendix

To prove the theorem we need to appeal to a representation theorem for probability functions on L satisfying Ex. First we introduce some notation.

For the language L as above a *state description* for $a_1, \ldots, a_n$ is a sentence of L of the form

$$\bigwedge_{i,j \leq n} R(a_i, a_j)^{\epsilon_{i,j}}$$

where the $\epsilon_{i,j} \in \{0, 1\}$ and $R(a_i, a_j)^1 = R(a_i, a_j), R(a_i, a_j)^0 = \neg R(a_i, a_j)$. By a theorem of Gaifman, see [1], or [10, Chap. 7], a probability function on SL is determined by its values on the state descriptions.

Let $D = (d_{i,j})$ be an $N \times N$ $\{0, 1\}$-matrix. Define a probability function w^D on SL by setting

$$w^D \left(\bigwedge_{i,j \leq n} R(a_i, a_j)^{\epsilon_{i,j}} \right)$$

to be the probability of (uniformly) randomly picking, with replacement, $h(1), h(2), \ldots, h(n)$ from $\{1, 2, \ldots, N\}$ such that for each $i, j \leq n$, $d_{h(i),h(j)} = \epsilon_{i,j}$. This uniquely determines a probability function on SL satisfying Ex. (For details see e.g. [10, Chap. 7]).

Clearly convex mixtures of these w^D also satisfy Ex. Indeed by the proof of [10, Theorem 25.1] it follows that any probability function w satisfying Ex can be approximated arbitrarily closely on $QFSL$ by such convex mixtures. More precisely:

Lemma 2. *For a probability function w on SL satisfying Ex and $\theta_1, \ldots, \theta_m \in QFSL$ and $\epsilon > 0$ there is an $N \in \mathbb{N}$ and $\lambda_D \geq 0$ for each $N \times N$ $\{0,1\}$-matrix D such that $\sum_D \lambda_D = 1$ and for $j = 1, \ldots, m$,*

$$|w(\theta_j) - \sum_D \lambda_D w^D(\theta_j)| < \epsilon.$$

We can extend this representation result to probability functions satisfying additionally SN as follows.

For $\theta \in SL$ let $\theta^\neg$ be the result of replacing each occurrence of R in θ by $\neg R$ and similarly for matrix D as above let $D^\neg$ be the result of replacing each occurrence of $0/1$ in D by $1/0$ respectively. For w a probability function on SL set $w^\neg$ to be the function on SL defined by

$$w^\neg(\theta) = w(\theta^\neg).$$

Then $w^\neg$ satisfies Ex and the probability function $2^{-1}(w + w^\neg)$ satisfies Ex+SN. Conversely if w satisfies Ex+SN then $w = w^\neg$ so

$$w = 2^{-1}(w + w^\neg).$$

Thus every probability function satisfying Ex+SN is of the form $2^{-1}(v + v^\neg)$ for some probability function v satisfying Ex and conversely every such probability function satisfies Ex+SN.

Notice that if

$$w = \sum_D \lambda_D w^D$$

then

$$w^\neg = \sum_D \lambda_D w^{D^\neg}$$

and

$$2^{-1}(w + w^\neg) = \sum_D \lambda_D 2^{-1}(w^D + w^{D^\neg}).$$

In particular then by Lemma 2,

Lemma 3. *For a probability function w on SL satisfying Ex+SN and $\theta_1, \ldots, \theta_m \in QFSL$ and $\epsilon > 0$ there is an $N \in \mathbb{N}$ and $\lambda_D \geq 0$ for each $N \times N$ $\{0,1\}$-matrix D such that $\sum_D \lambda_D = 1$ and for $j = 1, \ldots, m$,*

$$|w(\theta_j) - 2^{-1} \sum_D \lambda_D (w^D(\theta_j) + w^{D^\neg}(\theta_j))| < \epsilon.$$

Let w be a probability function on SL satisfying Ex and for a 2×2 $\{0,1\}$-matrix

$$E = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix}$$

let

$$|E|_w = w(R(a_1,a_3)^{e_{11}} \wedge R(a_1,a_4)^{e_{12}} \wedge R(a_2,a_3)^{e_{21}} \wedge R(a_2,a_4)^{e_{22}}).$$

We will omit the subscript w if it is clear from the context. Notice that when $D = (d_{i,j})$ is an $N \times N$ $\{0,1\}$-matrix, then for E as above we have

$$|E|_{w^D} = N^{-4} \sum_{i,j,r,s} d_{i,r}^{e_{11}} d_{i,s}^{e_{12}} d_{j,r}^{e_{21}} d_{j,s}^{e_{22}}, \tag{4}$$

where $x^1 = x, x^0 = 1 - x$. We will write $|E|_D$ in place of $|E|_{w^D}$.

A useful observation is that for any probability function w satisfying Ex, $|E|$ is invariant under permuting rows and permuting columns so for example

$$\begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 \\ 0 & 1 \end{vmatrix}, \qquad \begin{vmatrix} 1 & 1 \\ 0 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 0 \\ 1 & 1 \end{vmatrix}, \qquad \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix},$$

$$\begin{vmatrix} 1 & 0 \\ 0 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 \\ 0 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 0 \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 \\ 1 & 0 \end{vmatrix}, \tag{5}$$

etc. We will use this observation frequently in what follows.

Let

$$X = \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix} + \begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix}, \quad Y = \begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix} + \begin{vmatrix} 0 & 0 \\ 0 & 1 \end{vmatrix}, \quad T = \begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix}, \quad U = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}, \quad Z = \begin{vmatrix} 0 & 0 \\ 1 & 1 \end{vmatrix}.$$

Lemma 4. *For any probability function w satisfying Ex we have $T, Z \geq U$ and $X \geq 2Z, 2T$.*

Proof. We shall prove that $T \geq U$, the other inequalities follow similarly. Let $D = (d_{i,j})$ be an $N \times N$ $\{0,1\}$-matrix and assume first that $w = w^D$. By the above observation,

$$T = \frac{1}{2}\left(\begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix}_D + \begin{vmatrix} 0 & 1 \\ 0 & 1 \end{vmatrix}_D\right) \qquad U = \frac{1}{2}\left(\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}_D + \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}_D\right)$$

so $T \geq U$ is the inequality

$$\sum_{i,j,r,s} d_{i,r}(1-d_{i,s})d_{j,r}(1-d_{j,s}) + \sum_{i,j,r,s} (1-d_{i,r})d_{i,s}(1-d_{j,r})d_{j,s}$$
$$\geq \sum_{i,j,r,s} d_{i,r}(1-d_{i,s})(1-d_{j,r})d_{j,s} + \sum_{i,j,r,s} (1-d_{i,r})d_{i,s}d_{j,r}(1-d_{j,s})$$

which is equivalent to the sum over r, s of

$$\left(\sum_i d_{i,r}(1-d_{i,s})\right)^2 + \left(\sum_j (1-d_{j,r})d_{j,s}\right)^2 - 2\left(\sum_i d_{i,r}(1-d_{i,s})\right)\left(\sum_j (1-d_{j,r})d_{j,s}\right)$$

being nonnegative, and hence clearly true. From this it follows that the result holds for convex combinations of the w^D and hence by Lemma 2 for general w satisfying Ex.

Proof of Theorem 1. We start with the left hand side inequality. Let w be a probability function satisfying Ex+SN. If $w(R(s,h)\wedge(R(s,k)\rightarrow R(f,k))$ and/or $w(R(s,h))$ equals 0 then (2) holds by our convention, so assume that these values are nonzero. Consider an approximation $2^{-1}\sum_D \lambda_D(w^D + w^{D^-})$ of w for the θ of the form

$$R(f,h)^{e_{11}} \wedge R(f,k)^{e_{12}} \wedge R(s,h)^{e_{21}} \wedge R(s,k)^{e_{22}}$$

with small ϵ and $N \in \mathbb{N}$ as guaranteed by Lemma 3.

For an $N \times N$ $\{0,1\}$-matrix $D = (d_{i,j})$, write u for $2^{-1}(w^D + w^{D^-})$. We have

$$\begin{aligned}
u(R(f,h) \wedge R(s,h) \wedge (R(s,k) \rightarrow R(f,k)) &= 2^{-1}(X_D + 2T_D + Y_D),\\
u(R(s,h) \wedge (R(s,k) \rightarrow R(f,k)) &= 2^{-1}(X_D + 2T_D + 3Y_D + 2U_D),\\
u(R(f,h) \wedge R(s,h)) &= 2^{-1}(X_D + 2T_D + 2Y_D),\\
u(R(s,h)) &= 2^{-1}(X_D + 2T_D + 4Y_D + 2U_D + 2Z_D).
\end{aligned}$$

Let $\hat{D}$ be another (not necessarily distinct) $N \times N$ $\{0,1\}$ matrix. Working with approximations of w for arbitrarily small ϵ it can be seen that to show (2) for w it suffices to demonstrate that for any pair $D, \hat{D}$ we have

$$\begin{aligned}
&(X_D + 2T_D + Y_D)(X_{\hat{D}} + 2T_{\hat{D}} + 4Y_{\hat{D}} + 2U_{\hat{D}} + 2Z_{\hat{D}})\\
&\quad + (X_{\hat{D}} + 2T_{\hat{D}} + Y_{\hat{D}})(X_D + 2T_D + 4Y_D + 2U_D + 2Z_D)\\
&\geq (X_D + 2T_D + 3Y_D + 2U_D)(X_{\hat{D}} + 2T_{\hat{D}} + 2Y_{\hat{D}})\\
&\quad + (X_{\hat{D}} + 2T_{\hat{D}} + 3Y_{\hat{D}} + 2U_{\hat{D}})(X_D + 2T_D + 2Y_D).
\end{aligned}$$

This simplifies to

$$2X_D Z_{\hat{D}} + 4T_D Z_{\hat{D}} + 2Y_D Z_{\hat{D}} + 2X_{\hat{D}} Z_D + 4T_{\hat{D}} Z_D + 2Y_{\hat{D}} Z_D \geq 4Y_{\hat{D}} Y_D + 2U_D Y_{\hat{D}} + 2U_{\hat{D}} Y_D$$

and since by Lemma 4 we have $Z_D \geq U_D$, $Z_{\hat{D}} \geq U_{\hat{D}}$, it suffices to show that

$$(X_D + 2T_D)Z_{\hat{D}} + (X_{\hat{D}} + 2T_{\hat{D}})Z_D \geq 2Y_{\hat{D}} Y_D. \tag{6}$$

We have

$$\begin{aligned}
X_D + 2T_D &= \sum_{i,j} \big[\big(\sum_r d_{i,r}d_{j,r}\big)^2 + \big(\sum_s (1-d_{i,s})(1-d_{j,s})\big)^2\\
&\quad + 2\big(\sum_r d_{i,r}d_{j,r}\big)\big(\sum_s (1-d_{i,s})(1-d_{ijs})\big)\big]\\
&= \sum_{i,j} \big(\sum_r d_{i,r}d_{j,r} + \sum_s (1-d_{i,s})(1-d_{j,s})\big)^2\\
&= \sum_{i,j} (x_{i,j} + y_{i,j})^2,
\end{aligned} \tag{7}$$

where

$$x_{i,j} = \sum_r d_{i,r}d_{j,r}, \quad y_{i,j} = \sum_s (1-d_{i,s})(1-d_{j,s}).$$

Similarly

$$Z_D = \sum_{i,j} \big(\sum_{r,s} d_{i,r}d_{i,s}(1-d_{j,r})(1-d_{j,s}) \big) = \sum_{i,j} z_{i,j}^2 \tag{8}$$

where

$$z_{i,j} = \sum_r d_{i,r}(1-d_{j,r}),$$

and, using (5),

$$\begin{aligned} Y_D &= \sum_{i,j} \big(\sum_r (1-d_{i,r})d_{j,r} \big) \big(\sum_s d_{i,s}d_{j,s} + \sum_s (1-d_{i,s})(1-d_{j,s}) \big) \\ &= \sum_{i,j} z_{i,j}(x_{i,j}+y_{i,j}). \end{aligned} \tag{9}$$

Similarly for $\hat{D} = (\hat{d}_{i,j})$. Writing $u_{i,j}$ for $x_{i,j}+y_{i,j}$ etc., the inequality (6) becomes

$$\big(\sum_{i,j} u_{i,j}^2 \big) \big(\sum_{i,j} \hat{z}_{i,j}^2 \big) + \big(\sum_{i,j} \hat{u}_{i,j}^2 \big) \big(\sum_{i,j} z_{i,j}^2 \big) \geq 2 \big(\sum_{i,j} z_{i,j}u_{i,j} \big) \big(\sum_{i,j} \hat{z}_{i,j}\hat{u}_{i,j} \big)$$

which holds since for any particular pairs i,j and g,h,

$$u_{i,j}^2 \hat{z}_{g,h}^2 + \hat{u}_{g,h}^2 z_{i,j}^2 \geq 2 z_{i,j}u_{i,j}\hat{z}_{g,h}\hat{u}_{g,h}.$$

Turning to the right hand side inequality it is enough to show that

$$w(R(f,h) \wedge R(s,h)) \geq 2^{-1} w(R(s,h)),$$

equivalently

$$w(R(f,h) \wedge R(s,h)) \geq w(\neg R(f,h) \wedge R(s,h)).$$

Proceeding as above (but much simpler since it does not need to involve the $\hat{D}$) it is sufficient to show that

$$X_D + 2T_D \geq 2U_D + 2Z_D,$$

and indeed this holds by Lemma 4. □

Theorem 5. *Let w be a probability function on SL satisfying Ex+SN. Let h,k,s,f be distinct constants from amongst the $a_1, a_2, a_3, \ldots$.*

Then

$$w(R(f,h) \,|\, R(s,h) \wedge (R(s,k) \longleftrightarrow R(f,k))) \geq 1/2.$$

$$w(R(f,h) \,|\, R(s,h) \wedge (R(s,k) \wedge R(f,k))) \geq 1/2.$$

Proof. Starting with the bi-implication case and proceeding as in the proof of the second inequality in Theorem 1 it is enough to show that

$$X_D + 2T_D \geq 2Y_D. \tag{10}$$

To this end notice that

$$X_D = \sum_{r,s} \big(\big(\sum_i d_{i,r} d_{i,s} \big)^2 + \big(\sum_i (1 - d_{i,r})(1 - d_{i,s}) \big)^2 \big),$$

$$2T_D = 2 \sum_{r,s} \big(\sum_i d_{i,r} (1 - d_{i,s}) \big)^2,$$

$$2Y_D = \sum_{r,s} 2 \big(\big(\sum_i d_{i,r} (1 - d_{i,s}) \big) \big(\sum_i (1 - d_{i,r})(1 - d_{i,s}) + \sum_i d_{i,r} (1 - d_{i,s}) \big) \big(\sum_i d_{i,r} d_{i,s} \big) \big).$$

Writing

$$A_{r,s} = \sum_i d_{i,r} d_{i,s}, \quad B_{r,s} = \sum_i (1 - d_{i,r})(1 - d_{i,s}), \quad C_{r,s} = \sum_i d_{i,r} (1 - d_{i,s})$$

the required inequality becomes

$$\sum_{r,s} \big(A_{r,s}^2 + B_{r,s}^2 + 2C_{r,s}^2 - 2A_{r,s} C_{r,s} + 2B_{r,s} C_{r,s} \big) \geq 0,$$

which clearly holds.

The second inequality in the theorem can likewise be reduced to showing that $X_D \geq Y_D$ and this follows from (10) and Lemma 4. □

References

1. Gaifman, H.: Concerning measures on first order calculi. Israel J. Math. **2**, 1–18 (1964)
2. Ganeri, J.: Indian Logic: A Reader. Routledge, New York (2001)
3. Ganeri, J.: Ancient Indian logic as a theory of case based reasoning. J. Indian Philos. **31**, 33–45 (2003)
4. Matilal, B.K.: The Character of Logic in India. SUNY Series in Indian Thought. State University of New York Press, Albany (1998) (Ed. Halbfass, W.)
5. Matilal, B.M.: Introducing Indian logic. In: Ganeri, J. (ed.) Indian Logic, A Reader. Routledge (2001)
6. Oetke, C.: Ancient Indian logic as a theory of non-monotonic reasoning. J. Indian Philos. **24**, 447–539 (1996)
7. Paris, J.B., Vencovská, A.: The Indian schema as analogical reasoning. http://eprints.ma.man.ac.uk/2436/01/covered/MIMS_ep2016_10.pdf
8. Paris, J.B., Vencovská, A.: The Indian schema analogy principles. IfCoLog J. Logics Appl. http://eprints.ma.man.ac.uk/2436/01/covered/MIMS_ep2016_8.pdf
9. Paris, J.B., Vencovská, A.: Ancient Indian Logic, Pakṣa and Analogy. In: Proceedings of the joint Conference of the 3rd Asian Workshop on Philosophical Logic (AWPL 2016) and the 3rd Taiwan Philosophical Logic Colloquium (TPLC 2016), Taipei, October 2016 (to appear)

10. Paris, J.B., Vencovská, A.: Pure Inductive Logic. Association of Symbolic Logic Perspectives in Mathematical Logic Series. Cambridge University Press, New York (2015)
11. Schayer, S.: On the method of research into Nyāya (translated by J. Tuske). In: Ganeri, J. (ed.) Indian Logic: A Reader, pp. 102–109. Routledge, London, New York (2001)
12. Staal, J.F.: The concept of Pakṣa in Indian Logic. In: Ganeri, J. (ed.) Indian Logic: A Reader, pp. 151–161. Routledge, London, New York (2001)

Definability of Recursive Predicates in the Induced Subgraph Order

Ramanathan S. Thinniyam(✉)

The Institute of Mathematical Sciences,
CIT Campus, Taramani, Chennai 600113, India
thinniyam@imsc.res.in

Abstract. Consider the set of all finite simple graphs $\mathcal{G}$ ordered by the induced subgraph order $\leq_i$. Building on previous work by Wires [14] and Jezek and Mckenzie [5–8], we show that every recursive predicate over graphs is definable in the first order theory of $(\mathcal{G}, \leq_i, P_3)$ where P_3 is the path on 3 vertices.

1 Introduction

Finite graphs and graph theory have become of central importance with the advent of computer science since many computational problems can be modelled using them. Alongside this, the logical study of graphs has gained importance.

The "graph as a model" way of looking at graphs is the flourishing field of descriptive complexity, which has had success in creating logical objects equivalent to computational complexity classes. However, we will use a different way of looking at graphs. We will study the set of all isomorphism types of simple finite graphs (referred to as "graphs" from here on and denoted $\mathcal{G}$) with a single relation on $\mathcal{G}$, namely the induced subgraph relation (please see Fig. 1). This and other such relations such as the subgraph relation and the minor relation form interesting partial orders and their first order theory has been studied [13,14].

Note in particular that we do not have explicit access to the edge relation inside a particular graph, since we only have the single order relation as the vocabulary. Inspite of this, many graph families such as paths, cycles, cliques etc. and graph theoretical concepts such as connectivity, maximum degree etc. can be expressed in the first order theory of such objects, though in an indirect way. Thus we continue the exploration of the definablity and decidability in these first order theories (and their fragments).

Our work can be considered as continuing that of Jezek and Mckenzie [5–8], who studied the substructure orderings on various kinds of finite objects such as posets, lattices etc. This was later extended to the induced subgraph order by Wires [14]. The primary objective of these model theoretic studies is the determination of the automorphism group of these objects. On the other hand, our motivation is to explore the computational content of these objects.

To further this aim, we prove that the set of all recursive predicates is definable in the object $(\mathcal{G}, \leq_i, P_3)$ i.e. the induced subgraph order with a constant P_3

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 211–223, 2017.
DOI: 10.1007/978-3-662-54069-5_16

for the path on three vertices. The notion of recursive predicate we use is that of recognizability by a Turing machine of the encodings of graphs as numbers, for a fixed encoding that we define. We obtain the result by combining classical results on arithmetical definability and previous work by Jezek and McKenzie, and Wires.

Other work on orders on combinatorial objects includes that by Kunos [11] on the subdigraph order; and on word orders by Kuske [12].

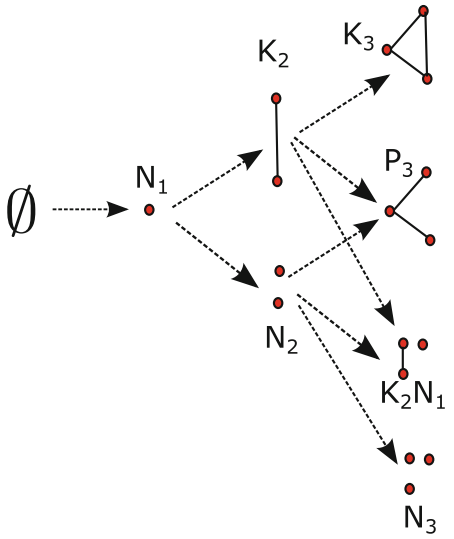


Fig. 1. The first few levels of the induced subgraph order. The arrows indicate the covering relation. ∅ is the empty graph.

While we are able to answer the question about definability of recursive predicates, our methods are too coarse to handle questions of definability of complexity classes (which are of course a strict subset of recursive predicates), say the set of all PTIME predicates over graphs. In addition, we do not take up the problem of precisely determining the logical resources required for the result. This paper is part of a preliminary investigation of the strength of such theories of combinatorial objects.

2 Preliminaries

First we give some definitions regarding graphs.

Definition 1 (Labelled Graph). *A (finite) labelled graph g is a structure (V_g, E_g, L_g) with*

1. finite domain (aka vertex) set V_g,

2. *a symmetric binary relation* $E_g \subseteq V_g \times V_g$ *which is the edge set of the graph, and*
3. *a bijective function* $L_g : V_g \to [n]$ *where* $[n]$ *stands for the initial segment* $\{1, 2, 3..., n\}$ *of the natural numbers with* $n = |V_g|$ *i.e.* n *is the number of vertices in the graph.*

We will write v_i to denote the vertex whose image under L_g is i. We will write v_iv_j to denote the edge (if it exists) between v_i and v_j. In addition, we restrict ourselves to simple graphs i.e. graphs which dont have edges of the form (v_i, v_i) for some $v_i \in V_g$.

Definition 2. *An isomorphism between two labelled graphs* g_1 *and* g_2 *is a bijection* $\eta : V_{g_1} \to V_{g_2}$ *such that for any two vertices* v_i, v_j *of* g_1*, the edge* v_iv_j *exists if and only if there is an edge between vertices* $\eta(v_i), \eta(v_j)$ *in* g_2*.*

We say g_1 is isomorphic to g_2 if there is an isomorphism between them, and write $g_1 \simeq g_2$. The relation $\simeq$ is an equivalence relation on the set of all finite labelled graphs.

Definition 3 (Graph). *By a graph* g*, we mean an equivalence class under the relation* $\simeq$ *over the set of all finite labelled graphs. The set of all graphs will be denoted* $\mathcal{G}$*.*

We will write $g = [g']$ to denote that the graph g is the isomorphism type of the labelled graph g'.

All variables x, y, z occuring in formulae denote graphs and not labelled graphs. However, we will however need to talk about specific vertices or edges inside a graph and thus will require a labelling. So we will abuse notation and use u, v to talk of vertices of a graph (not a labelled one), uv for the edge joining u and v, and e to denote the edge of a graph. We denote graphs by g, h, and graph families by caligraphic letters such as $\mathcal{P}, \mathcal{C}$.

We will denote by N_i, K_i, C_i, S_i, P_i the graph consisting of i isolated vertices, the i-clique, the cycle on i vertices, the star on i vertices and the path on i vertices respectively (see Fig. 2); and by $\mathcal{N}, \mathcal{K}, \mathcal{C}, \mathcal{S}, \mathcal{P}$ the corresponding families of isolated vertices, cliques, cycles, stars and paths. We denote the cardinality (number of vertices) of a graph g by $|g|$, and the disjoint union of graphs g and h by $g \cup h$.

Next we need some definitions regarding the first order structures we study and definability in them.

For the standard syntax and semantics of first order logic, we refer the reader to Enderton [2].

Definition 4 (Induced Subgraph Order). *We consider the first order theory of the structure* $(\mathcal{G}, \leq_i, P_3)$ *where* P_3 *is a constant symbol for the path on three vertices and the* $\leq_i$ *is the induced subgraph order which is defined as:* $g \leq_i g'$ *iff* g *can be obtained from* g' *by deleting some (arbitrarily many) vertices of* g'*.*

The constant symbol P_3 is used to break the symmetry of the induced subgraph order which by itself cannot distinguish between a graph and its complement since the map sending a graph to its complement is an automorphism of the order.

Definition 5 (Arithmetic). *By arithmetic, we mean the first order theory of the structure $(\mathbb{N}, \phi_+, \phi_\times)$ where $\mathbb{N}$ is the set of all natural numbers and $\phi_+, \phi_\times$ are ternary predicates for addition and multiplication respectively.*

We will also use variables x, y, z to denote numbers in arithmetical formulae; and lower case letters k, l, m, n to denote numbers.

Definition 6 (Constant Definability). *Fix a first order language $\mathcal{L}$. Let e be an element of the domain of an $\mathcal{L}$-structure $\mathcal{A}$. We say that e is definable in $\mathcal{A}$, if there exists an $\mathcal{L}$ formula $\alpha_e(x)$ in one free variable, such that $\mathcal{A}, e \vDash \alpha_e(x)$ and for any $e' \neq e$ in the domain of $\mathcal{A}$, $\mathcal{A}, e' \nvDash \alpha_e(x)$.*

For any definable domain element e, we use e as a constant symbol representing the domain element because an equivalent formula can be written in the language $\mathcal{L}$ via use of the defining formula α_e.

Definition 7 (Covering Relation of a Poset). *Given elements x, y of a poset $(P, \leq)$ we define the covering relation $x \lessdot y$ as $x \lessdot y$ iff $x < y$ and there exists no element z of P such that $x < z < y$. This can easily written using a first order formula in the vocabulary of $\{\leq\}$.*

Definition 8 (Definability of Predicates). *We say a predicate is* definable in arithmetic *iff it is definable in $(\mathbb{N}, \phi_+, \phi_\times)$ and a predicate is* definable in graph theory *iff it is definable in $(\mathcal{G}, \leq_i, P_3)$.*

We use the symbol ϕ for arithmetical formulae and ψ for graph theory formulae to aid the reader.

Observation 1. *For any definable family $\mathcal{F}$ of $(\mathcal{G}, \leq_i, P_3)$ which forms a total order under $\leq_i$, every member of $\mathcal{F}$ is definable as a constant.*

To see this, first observe that there exists a minimum element f_1 in $\mathcal{F}$ by well foundedness of the order $\leq_i$.

$$f_1(x) := \mathcal{F}(x) \wedge (\forall y\, \mathcal{F}(y) \supset (y \leq_i x))$$

Assuming f_n (the n^{th} smallest element of $\mathcal{F}$) has been defined, f_{n+1} can be defined as the unique cover of f_n in $\mathcal{F}$.

Next we have the definitions we need to formalize the meaning of "recursive predicate over graphs." There exist notions of computability and recursive predicates over abstract structures (see [3]), but these are fairly technical. For our purposes, we use a fixed encoding of graphs as strings so that the standard notion of a computable predicate as one accepted by a Turing machine can be used. We encode graphs as numbers (equivalently binary strings). These encodings were originally introduced by us in previous work [13].

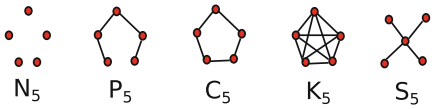


Fig. 2. Isolated points, path, cycle, clique and star of order 5 from left to right.

Definition 9 (Number Representation of a Graph). *A number representation of a graph g is defined using the following procedure.*

1. *Choose an labelled graph g' such that $g = [g']$. The order on vertices given by $L_{g'}$ induces an order $\leq_e$ on set S of all tuples of vertices (v_i, v_j) of g with $j < i$. Let (v_i, v_j) and (v_k, v_l) belong to S (i.e. $j < i, l < k$). Then $(v_i, v_j) \leq_e (v_k, v_l)$ iff $i < k$ or $i = k, j < l$.*
2. *Arrange all the tuples belonging to S in descending order by $\leq_e$ to form the sequence seq.*
3. *Create the number m whose binary expansion is $\binom{n}{2} + 1$ bits long and has the following property: the most significant bit is 1 (always true for a number). The i^{th} most significant bit is 0 or 1 according to whether the $i-1^{th}$ tuple in seq corresponds to a non-edge or edge (respectively) of the labelled graph g'.*

The number m is called a number representation of the graph g.

Definition 10 (Unique Number Representation). *The unique number representation of a graph g is the least number m such that it is a number representation of g and is denoted $UN(g)$. Note that the map $UN : \mathcal{G} \rightarrow \mathbb{N}$ is a one-one map. (See Fig. 3 for an example.)*

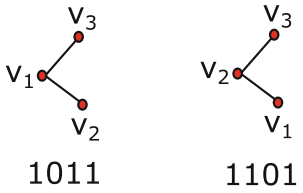


Fig. 3. Two different number representations of P_3 corresponding to two different labellings. The one on the left (i.e. 1011 in binary which is the number 11) is $UN(P_3)$.

Observation 2. *The representation UN induces an ordering on the vertices of the graph which comes from the underlying labelled graph.*

We can finally state what we mean by recursive predicates over graphs.

Definition 11. *We say a predicate $R \subseteq \mathcal{G}^n$ is recursive if there exists a Turing machine M such that*

$$R(\bar{g}) \iff UN(\bar{g}) \in L(M)$$

i.e. the Turing machine accepts exactly the tuples of strings which correspond to UN encodings of tuples belonging to R.

3 Main Result

We note that the richness of a structure (for instance, its ability to interpret arithmetic) does not automatically imply the obtained result. Something more is required: the ability of the structure to perform operations on its elements, and in some sense, access its own internal structure in a way that is first order definable.

We will state the main result and show how the various modules come together to form the proof. Some of the details are postponed to make the presentation more understandable.

Theorem 1. *Every recursive predicate $R \subseteq \mathcal{G}^n$ on graphs is definable in $(\mathcal{G}, \leq_i, P_3)$.*

We need to show that for every recursive predicate $R \subseteq \mathcal{G}^n$ over graphs, there exists a formula $\psi_R(\bar{x})$ (where $|\bar{x}| = n$) in graph theory such that for any n-tuple of graphs $\bar{g}$,

$$R(\bar{g}) \iff (\mathcal{G}, \leq_i, P_3) \models \psi_R(\bar{g})$$

Since R is a recursive predicate, by Definition 11 there exists a machine M which accepts the UN number encodings of the set of graphs which belong to R.

$$R(\bar{g}) \iff UN(\bar{g}) \in L(M)$$

The following is a classical theorem (see Appendix for a proof sketch):

Theorem 2. *Every recursive predicate R on numbers is definable in arithmetic.*

Thus there is an arithmetic formula $\phi_{UN(R)}(\bar{x})$ such that for any tuple $\bar{n}$ of numbers,

$$(\mathbb{N}, \phi_+, \phi_\times) \models \phi_{UN(R)}(\bar{n}) \iff \bar{n} \in UN(R)$$

Next we recall that

Theorem 3 (Wires [14]). *Arithmetic i.e. $(\mathbb{N}, \phi_+, \phi_\times)$, is definable in graph theory i.e., $(\mathcal{G}, \leq_i, P_3)$.*

In particular, the image set of the following map from numbers to graphs is definable:

$$UG : \mathcal{G} \to \mathbb{N}$$

UG takes a number n to the graph N_n which consists of n isolated points. There also exist defining formulae in graph theory for the predicates:

$$\psi_{(+)}(x,y,z) \text{ iff } ; x,y,z \in \mathcal{N} \text{ and } |x|+|y| = |z|.$$
$$\psi_{\times}(x,y,z) \text{ iff } ; x,y,z \in \mathcal{N} \text{ and } |x| \times |y| = |z|$$

We will write $|x|$ to denote the graph $N_{|x|}$ since there is a formula which defines the binary predicate $Norder(x,y)$ iff $|x| = |y|$ and $y \in \mathcal{N}$.

We will abuse notation by writing i instead of N_i and expressions such as $i+j, ij$ will be taken to mean the member of $\mathcal{N}$ such that its order equals $i+j, i \times j$ respectively. Similarly, since the order relation $<$ over the naturals is definable using addition, we will use quantifiers such as $\forall 1 < j < n$ in graph theory whose meaning is really $\forall j\, \mathcal{N}(j) \wedge N_1 \leq_i j \wedge j \leq_i N_n$.

Observation 3. *For every formula $\phi(\bar{x})$ in arithmetic there is a formula $\psi^t(\bar{x})$ in graph theory such that*

$$(\mathbb{N}, \phi_+, \phi_\times) \models \phi(\bar{n}) \iff (\mathcal{G}, \leq_i, P_3) \models \psi^t(UG(\bar{n}))$$

Applying this translation to the formula $\phi_{UN(R)}(\bar{n})$ gives us the graph formula $\psi^t_{UN(R)}(UG(\bar{n}))$.

Given a graph g, the above formula essentially states what we require but in terms of the graph $UG(UN(g))$. If there were a definable way to go between these two graphs inside the induced order, we could potentially "do the computation inside arithmetic and come back". This is essentially what we do to get the formula we require. In order to do this we require two things:

1. Acess to the edge relation inside arithmetic so that we can carry out the required computation inside arithmetic.
2. The ability to access the internal structure of a graph using the induced subgraph order.

The first of these has already been accomplished in previous work:

Theorem 4 ([13]). *The following predicates are definable in arithmetic:*

1. *$\phi_{UN}(x)$ iff x is a number which represents an isomorphism type of a graph as given in Definition 10.*
2. *$\phi_{edge}(x,i,j)$ iff x is a number representation of graph g_x and $v_i v_j \in E_{g_x}$.*
3. *$\phi_{length}(n,x)$ iff the length of the binary representation of x is n. We will just write $length(x)$ to denote n.*

Now we tackle the second problem i.e. that of accessing the internal structure of a graph. This is accomplished by using definable "vertex labelled representations" of graphs (which are themselves graphs), called *o-presentations*. This was first introduced by Jezek and Mckenzie, and defined for graphs by Wires.

Definition 12 (o-presentation). *An o-presentation of $g \in \mathcal{G}$ is another graph $\tilde{g}$ defined as follows: Fix an enumeration $v_1, v_1, .., v_n$ of vertices of g. Let g' be the graph formed by the disjoint union of g and the cycles C_{n+i+2} for each $1 \leq i \leq n$. Add n additional edges to g' connecting each cycle C_{n+i+2} to the corresponding vertex v_i. The resulting graph is denoted $\tilde{g}$.*

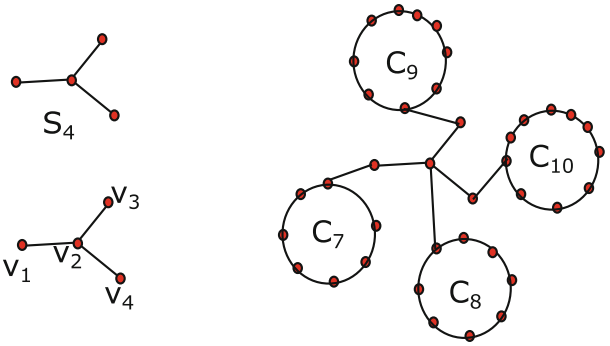


Fig. 4. Top left: the star graph S_4. Bottom left: a vertex labelling of S_4. Right: the o-presentation corresponding to the vertex labelling.

Given a graph g, an o-presentation can be regarded as the representation of a labelled graph g' with $g = [g']$, as another graph $\tilde{g}$. From the example in the Fig. 4, it is clear that there is a bijective correspondence between o-presentations and labellings of a graph.

The proof of the following lemma is deferred to the end of the section so as to not obstruct the flow of the main proof:

Lemma 1. *The following predicates are definable in* $(\mathcal{G} \leq_i, P_3)$*:*

1. *The set of all o-presentations, denoted* $\tilde{\mathcal{G}}$ *i.e.* $\tilde{\mathcal{G}}(x)$ *holds iff there is a graph* y *such that* x *is an o-presentation of* y*.*
2. *The predicate* $\psi_{opres}(x, y)$ *iff* y *is an o-presentation* $\tilde{x}$ *of* x*, also written* $y = \tilde{x}$ *for short.*
3. $\psi_{edgeOP}(x, i, j)$ *iff there exists a graph* y *such that* $y = \tilde{x}$ *and in the vertex labelling corresponding to the o-presentation, there is an edge between vertices* v_i *and* v_j *in the graph* y*.*

Using Lemma 1 and Theorem 4 we can now define the binary relation $n = UG(UN(x))$ by the formula $\psi_{enc}(x, n)$:

$$\begin{aligned}\psi_{enc}(x, n) := n \in \mathcal{N} \wedge \exists y\, y = \tilde{x} \wedge \psi^t_{graphOrder}(n, |x|) \\ \wedge\ \psi^t_{UN}(n) \wedge \forall 1 \leq i < j \leq |x| \\ \psi^t_{edge}(n, i, j) \iff \psi_{edgeOP}(y, i, j)\end{aligned}$$

The formula asserts that n is a trivial graph (has no edges) and there exists an o-presentation y of x such that there exists an edge between vertices v_i and v_j in the enumeration of the graph g corresponding to the o-presentation if and only if there is an edge between vertices v_i and v_j in an enumeration of the graph which is consistent with the UN representation.

By use of Lemma 1, we are able to write $y = \tilde{x}$. The formulae ψ^t_{edge} and $\psi^t_{UN}(n)$ are the translations of the formulae from Theorem 4 by using

Observation 3. $\psi^t_{graphOrder}$ is the translation of the following arithmetic formula

$$\phi_{graphOrder}(n, m) := length(n) = 1 + m(m-1)/2$$

Note that the arguments in the translated formulae have to be members of the family $\mathcal{N}$ and are applied to the image graph under the map UG while the arithmetic formulae are on the numbers obtained from the inverse of this map. Also note the use of $\forall 1 \leq i < j \leq |x|$ which is syntactic sugar for a more involved formula we can write in graph theory due to definability of arithmetic (recall remarks under Theorem 3). We can now define R in the induced subgraph order:

$$\psi_R(\bar{x}) := \exists \bar{y} \bigwedge_{i=1}^{n} \psi_{enc}(x_i, y_i) \wedge \psi^t_{UN(R)}(\bar{y})$$

ψ_R essentially inverts the encoding function UN to go back from the number encodings to the graphs.

All that remains to be done is the proof of Lemma 1. In order to do so, we need some machinery:

Lemma 2 (Wires [14]). *The following predicates are definable in $(\mathcal{G}, \leq_i, P_3)$.*

1. *The families $\mathcal{N}, \mathcal{K}, \mathcal{C}, \mathcal{P}$ standing for trivial graphs, complete graphs (cliques), cycles and paths respectively.*
2. *$|x| = |y|$ iff x and y have the same cardinality (i.e. same number of vertices, also known as order of the graph).*
3. *$maxComp(x, y)$ iff x is a maximal connected component of y.*
4. *$cover(x, y, n)$ iff there are exactly $n-1$ graphs between x and y in the order and $x \leq_i y$. Also denoted $x \lessdot_i^n y$.*

$$cover(x, y, n) := |x| + n = |y|$$

The order of the graph defines a layering of the induced subgraph order.

5. *$z = x \cup y$ iff z is the disjoint union of x and y.*
6. *$C_{\to 1}(x)$ iff x is the connected graph formed by adding one extra vertex and one extra edge to a cycle.*
7. *$conn(x)$ iff x is a connected graph.*
8. *$C_{\to 2}(x)$ iff x is graph formed by taking a graph g with $C_{\to 1}(g)$ and adding an additional vertex and joining it to the unique degree 1 vertex in g.*
9. *$pointedCycleSum(x, y, z)$ iff x and y are incomparable cycles and z is formed by starting with the graph $x \cup y$ and adding one extra vertex v and two extra edges, one from v to any vertex of x and another from v to any vertex of y. We will write $z = x +_p y$ for short.*

Notice that from the definability of $C_{\to 1}(x)$ we also have definability of the graph $C_{i \to 1}$ which stands for the member of $C_{\to 1}$ of order $i+1$ because the family is totally ordered by number of vertices and for similar reasons as Observation 1. Additionally, given a parameter n, we can obtain $C_{n \to 1}$.

We are now ready to give a proof of Lemma 1:

Proof of Lemma 1. We recollect the statement. The following are definable in graphs:

1. The set of all o-presentations, denoted $\tilde{\mathcal{G}}$ i.e. $\tilde{\mathcal{G}}(x)$ holds iff there is a graph y such that x is an o-presentation of y.
2. The predicate $\psi_{opres}(x, y)$ iff y is an o-presentation $\tilde{x}$ of x, also written $y = \tilde{x}$ for short.
3. $\psi_{edgeOP}(x, i, j)$ iff there exists a graph y such that $y = \tilde{x}$ and in the vertex labelling corresponding to the o-presentation, there is an edge between vertices v_i and v_j in the graph y.

Proof. We take up the definition of the family $\tilde{\mathcal{G}}$. First we note that given a number n, we can construct the object $\bigcup_{i=1}^{n} C_{n+i+2}$ as follows:

$$
\begin{aligned}
csum(n,x) \text{ iff } n \in \mathcal{N} \text{ and } x = \bigcup_{i=1}^{n} C_{n+i+2}.& \\
csum(n,x) :=& \forall z\, maxComp(z,x) \supset \mathcal{C}(z) \\
& \wedge\ cardCond(n,x) \wedge allCycles(n,x) \\
& \text{where} \\
cardCond(n,x) :=& \mathcal{N}(n) \wedge |x| = n^2 + n(n+1)/2 + 3n \\
allCycles(n,x) :=& \forall m\, (n+3 \leq m \leq 2n+2) \supset C_m \leq_i x
\end{aligned}
$$

The formula constrains every maximal component to be a cycle using the *maxComp* predicate. This forces all the cycles to be disjoint. Enforcing the cardinality condition and the fact that each cycle has to be present (*allCycles*) makes sure that the graph is made up of exactly one copy of each cycle and nothing else.

Now we can define the set of o-presentations as follows:

$$
\begin{aligned}
\tilde{\mathcal{G}}(x) :=& \exists n\, cardCond(n,x) \wedge hasC1s(n,x) \\
& \wedge\ hasUnionOfCycles(n,x) \\
& \text{where} \\
cardCond(n,x) :=& \mathcal{N}(n) \wedge |x| = n^2 + n(n+1)/2 + 3n \\
hasC1s(n,x) :=& \forall i\, (1 \leq i \leq n) \supset C_{i+n+2 \to 1} \leq_i x \\
hasUnionOfCycles(n,x) :=& \bigcup_{i=1}^{n} C_{n+i+2} \leq_i x
\end{aligned}
$$

The formula *cardCond* states that the graph has as many vertices as required to contain as induced subgraph a graph on n vertices and cycles of order $n+i+2$ for each i between 1 and n. *hasCycles* states that the $C_{\to 1}$ are induced subgraphs. *hasUnionOfCycles* states that the disjoint union of all the required cycles is an induced subgraph. Because of the cardinality constraint already imposed, this implies that there is a unique copy of each cycle in x. In addition, there are no chord or edges between the cycles. No restriction is place on the edges between the non-cycle vertices. Thus the resulting graph x is of the required form.

We take up the second predicate, $\psi_{opres}(x, y)$ iff y is an o-presentation of x.

$$\psi_{opres}(x,y) := |y| = |x|^2 + |x|(|x|+1)/2 + 3|x| \wedge \tilde{\mathcal{G}}(y) \\ \wedge \exists z\, z = x \cup \bigcup_{i=1}^{|x|} P_{|x|+1+i} \wedge z \lessdot_i^{|x|} y$$

The object $\bigcup_{i=1}^{n} P_{n+i+1}$ can be constructed given n by taking the appropriate $\bigcup_{i=1}^{n} C_{n+i+2}$, deleting n vertices from it, and enforcing the condition that no cycles are present.

The formula ψ_{opres} states that y is an o-presentation of appropriate order and deletion of $|x|$ vertices from y gives the disjoint union of x with paths of size $|x|+2$ to $2|x|+1$. The only way to get an o-presentation by adding $|x|$ vertices to z is to add two edges between every new vertex and and ends of one of the paths and one edge from the new vertex to a vertex in x. Thus any such y must be an o-presentation of x.

Moving on to the last predicate $\psi_{edgeOP}(x, i, j)$, we first need the following intermediate predicate:

$CP_4C(x, i, j)$ iff $i, j \in \mathcal{N}, 3 < i < j$ and x is formed by adding to the graph $C_i \cup C_j$ two additional vertices v_1, v_2 and the edge v_1v_2, one edge between C_i and v_1 and one edge between C_j and v_2. We denote x by $CP_4C(i, j)$.

$$CP_4C(x,i,j) := conn(x) \wedge \mathcal{N}(i) \wedge \mathcal{N}(j) \wedge 3 < i < j \\ \wedge C_i +_p C_j \not\leq_i x \\ \wedge C_{i\to 1} \cup C_j \lessdot_i x \wedge C_{j\to 1} \cup C_i \lessdot_i x$$

From the definition, x has to be obtained by adding one new vertex v to $g_0 = C_{i\to 1} \cup C_j$ and some number of edges which are incident on v. Notice that there is only one copy of C_j present as subgraph in x because of cardinality constraints. Thus there is exactly one edge between v and C_j (connectivity constraint). If there were multiple edges, we cannot get $C_{j\to 1}$ as induced subgraph. Now suppose there is also exactly one edge from v to copy of C_i in g_0, then we can get $C_i +_p C_j$ as induced subgraph, which is not allowed. Suppose there are multiple edges between v and copy of C_i in g_0, then we cannot get $C_{j\to 1} \cup C_i$ as induced subgraph from x by deleting a single vertex (since v remains connected to the rest of the graph not considering C_j on deleting only one vertex). But given the connectivity constraint, there must be an edge from v to the dangling vertex of $C_{i\to 1}$ inside g_0. Thus the graph we get is the required graph.

We can now write

$$\psi_{edgeOP}(x,i,j) := \exists y\, x = \tilde{y} \wedge \exists m\, (|x| = m^2 + m(m+1)/2 + 3m) \\ \wedge CP_4C(m+i+2, m+j+2) \leq_i x$$

The existence of an edge between vertices v_i and v_j in the graph x is captured by the presense of a CP_4C induced subgraph in y (which is an o-presentation of x) with appropriate parameters and this is stated by the formula ψ_{edgeOP}. □

This concludes the proof of Lemma 1 and thus completes the proof of Theorem 1.

4 Discussion

Our result leads to a number of interesting questions and potential areas for research.

There has been considerable work in the area of bounded arithmetic systems and their connection to complexity theory [1,10]. An intimate connection has been shown between propositional proof systems, systems of bounded arithmetic and complexity theory. Characterizing complexity classes of graph problems using fragments of the induced subgraph order may prove useful.

The way we have arrived at our result is very roundabout in the sense that we dont use any "natural computational predicates" over graphs. There may be such predicates over graphs which are the equivalent of the *bit* predicate and exponentiation in arithmetic. It is by carefully controlling these two (and further expanding the language) that the bounded arithmetic theories were discovered. In addition, we note that the method of computation we use essentially puts a total order on the vertices of the graph (via the o-presentation). This is closely related to the question of "order-invariant querying" which is of much interest in finite model theory and descriptive complexity [4].

There are related objects such as the subgraph order and the graph minor order whose expressive power is enough to interpret arithmetic [13] but it is not clear if o-presentations can be defined in them. On the other hand we do not have the tools to tackle the problem of proving inexpressibility in such rich structures. It would be interesting (though doubtful) to see if there are any general methods to generate o-presentations in different types of structures.

Acknowledgment. I would like to thank my guide Prof. R. Ramanujam for his advice and discussions on both technical matter and the presentation of this paper.

Appendix: Proof Sketch of Theorem 2

Theorem Statement: Every recursive predicate R on numbers is definable in first order arithmetic.

Proof (sketch). For simplicity we look at the case of only unary predicates, assume $R \subseteq \mathbb{N}$. Let $M = (Q, \delta, s, F)$ be a turing machine over the alphabet $\{0, 1\}$. First, consider strings over the alphabet $\Sigma = (0, 1, \#, s, q_1, ..., q_n)$ where $Q = \{s, q_1, ..., q_n\}$. They can be encoded as binary strings by using some encoding e.g. 0 is mapped to 01, 1 to 001, # to 0001, s to 00001, q_i to $0^{i+4}1$. Given any input x, we can encode the run of the Turing machine as a number y, which we will think of a string over the extended alphabet Σ (ignoring the 1 in the most significant digit). $y = c_1\#c_2\#...\#c_m$ where each c_i is a string containing exactly one state symbol and remaining 0's and 1's. The placement of the head of the machine is given by the position just after the state symbol. c_1 is sx i.e. the starting state s concatenated with the input x, c_m is a configuration containing a final state and the relationship between any two consecutive configurations is restricted based on the transition function δ. All of this can be written as a

formula $\phi_R(x)$ which essentially states "there exists a number y such that the binary encoding of the number represents an accepting run of the machine on x", making crucial use of the *bit* predicate and exponentiation. For details on definability in arithmetic, please see Kaye [9]. □

References

1. Cook, S., Nguyen, P.: Logical Foundations of Proof Complexity. Cambridge University Press, Cambridge (2010)
2. Enderton, H.: A Mathematical Introduction to logic. Academic Press, Burlington (2001)
3. Fitting, M.: Fundamentals of Generalized Recursion Theory. Elsevier, Amsterdam (2011)
4. Grohe, M.: The quest for a logic capturing PTIME. In: 23rd Annual IEEE Symposium on Logic in Computer Science, LICS 2008, pp. 267–271. IEEE (2008)
5. Ježek, J., McKenzie, R.: Definability in substructure orderings, IV: finite lattices. Algebra Univers. **61**(3–4), 301–312 (2009)
6. Ježek, J., McKenzie, R.: Definability in substructure orderings, I: finite semilattices. Algebra Univers. **61**(1), 59–75 (2009)
7. Ježek, J., McKenzie, R.: Definability in substructure orderings, III: finite distributive lattices. Algebra Univers. **61**(3–4), 283–300 (2009)
8. Ježek, J., McKenzie, R.: Definability in substructure orderings, II: finite ordered sets. Order **27**(2), 115–145 (2010)
9. Kaye, R.: Models of Peano arithmetic. Oxford University Press, Oxford (1991)
10. Krajicek, J.: Bounded Arithmetic, Propositional Logic and Complexity Theory. Cambridge University Press, Cambridge (1995)
11. Kunos, Á.: Definability in the embeddability ordering of finite directed graphs. Order **32**(1), 117–133 (2015)
12. Kuske, D.: Theories of orders on the set of words. RAIRO Theor. Inform. Appl. **40**(01), 53–74 (2006)
13. Ramanujam, R., Thinniyam, R.S.: Definability in first order theories of graph orderings. In: Artemov, S., Nerode, A. (eds.) LFCS 2016. LNCS, vol. 9537, pp. 331–348. Springer, Heidelberg (2016). doi:10.1007/978-3-319-27683-0_23
14. Wires, A.: Definability in the substructure ordering of simple graphs. Ann. Comb. **20**(1), 139–176 (2016)

Computational Complexity of a Hybridized Horn Fragment of Halpern-Shoham Logic

Przemysław Andrzej Wałęga(✉)

Institute of Philosophy, University of Warsaw, Warsaw, Poland
p.a.walega@gmail.com

Abstract. We propose hybridization of sub-propositional fragments of Halpern-Shoham logic as a way of obtaining expressive and decidable referential interval temporal logics. In the paper, we hybridize a Horn fragment of Halpern-Shoham logic whose language is restricted in its modal part to necessity modalities, and prove that satisfiability problem in this fragment is NP-complete over reflexive or an irreflexive and dense underlying structure of time.

Keywords: Interval logic · Hybrid logic · Computational complexity

1 Introduction

Temporal reasoning constitutes one of the main topics investigated within the field of AI and has been successfully applied in a number of areas, e.g., philosophy, program verification, automatic planning, etc. Logics that serve to formalise reasoning about time may be divided into two categories, namely *point-based* and *interval-based* depending on the type of primitive ontological objects involved in the representation. The latter approach seems to be more natural for human-like reasoning and more suitable for continuous process modelling or representing constructs from natural language [8].

An elegant and well studied interval-based temporal logic was introduced by Halpern and Shoham in [10] as a propositional multimodal logic. The logic (denoted by HS) introduces one modal operator for each of the well-known Allen relations [1], except "equals" relation. The Allen relations form a jointly exhaustive and pairwise disjoint set of binary relations between intervals, namely: begins (rel_B), during (rel_D), ends (rel_E), overlaps (rel_O), adjacent to (rel_A), later than (rel_L), their inverses denoted by $\mathsf{rel}_{\overline{\mathsf{B}}}$, $\mathsf{rel}_{\overline{\mathsf{D}}}$, $\mathsf{rel}_{\overline{\mathsf{E}}}$, $\mathsf{rel}_{\overline{\mathsf{O}}}$, $\mathsf{rel}_{\overline{\mathsf{A}}}$, and $\mathsf{rel}_{\overline{\mathsf{L}}}$ respectively, and an equality relation.

HS is highly expressive, in particular it is strictly more expressive than any point-based temporal logic over linear orders [10]. On the other hand, HS is undecidable for a range of linear orders including $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ [10]. Therefore, a number of methods to reduce its computational complexity have been studied. One approach for reducing the complexity is to restrict the set of modal operators [7,8]. Another, more recent approach is the investigation of sub-propositional

S. Ghosh and S. Prasad (Eds.): ICLA 2017, LNCS 10119, pp. 224–238, 2017.
DOI: 10.1007/978-3-662-54069-5_17

languages such as *Horn* and *core* fragments [5,6]. Importantly, full HS is *referential*, i.e., it enables us to label intervals and then to refer to a chosen interval with a concrete label. This kind of reference is a crucial construct in temporal knowledge representation [2,4] and the most straightforward way to provide it is to hybridize a logic. That is to add the second sort of expressions to the language (the so-called nominals), i.e., primitive formulas each of which is true in exactly one interval, and satisfaction operators indexed by nominals that enable to access a particular interval denoted by this nominal [4]. Although HS is not a hybrid logic, it is expressive enough to define the *difference operator* (which states that a formula is satisfied in some interval different from the current one), which in turn can be used to express nominals and satisfaction operators [2]. However, HS fragments are usually no longer able to express the difference operator and they lose the ability to refer to particular intervals. The most straightforward way to restore the referentiality in HS fragments is to hybridize them. Surprisingly, although hybridization of interval temporal logics was already recognised as a promising line of research [4], it has received only limited attention from the research community. One exception is an attempt of adding a very restricted reference property (enabling to state that some propositional variable is satisfied in a particular interval) [3].

An interesting fragment of HS is a Horn fragment allowing only boxes, i.e., necessity modalities (diamonds, i.e., possibility modalities are forbidden) called $\mathsf{HS}^{\Box}_{horn}$ [3,5,11]. $\mathsf{HS}^{\Box}_{horn}$ is known to be tractable (P-complete) if the underlying structure of time is reflexive, or irreflexive and dense [5]. On the other hand, this logic is still expressive enough to be used as a template to define temporal ontology languages [3]. Since $\mathsf{HS}^{\Box}_{horn}$ maintains a good balance between computational complexity and expressive power, it has recently gained attention among researchers working on HS [3,5,6,11]. In this paper, we hybridize $\mathsf{HS}^{\Box}_{horn}$ and study the computational complexity of the obtained logic (called $\mathsf{HS}^{\Box,i,@}_{horn}$). Our main result is that over reflexive, or irreflexive and dense underlying time structures hybridization of $\mathsf{HS}^{\Box}_{horn}$ results in an NP-complete logic – recall that $\mathsf{HS}^{\Box}_{horn}$ is P-complete over such structures (in contrast to classical modal logic which is PSPACE-complete before and after hybridization). Hence, adding referentiality to $\mathsf{HS}^{\Box}_{horn}$ enables us to maintain decidability but it has a price of reaching NP-completeness, i.e., losing tractability of the logic (provided that P $\neq$ NP).

The paper is organized as follows. In Sect. 2 we describe HS, $\mathsf{HS}^{\Box}_{horn}$, and its hybrid version $\mathsf{HS}^{\Box,i,@}_{horn}$. In Sect. 3 we prove that satisfiability in $\mathsf{HS}^{\Box,i,@}_{horn}$ is NP-hard, and in Sect. 4 that this problem is in NP over reflexive and irreflexive and dense time structures. Finally, in Sect. 5 we briefly conclude the paper.

2 Halpern-Shoham Logic

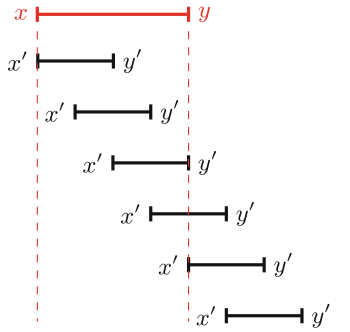
HS language is a modal language consisting of a set of propositional variables PROP, propositional constants $\top$ (true) and $\bot$ (false), classical propositional connectives $\neg, \wedge, \vee, \rightarrow$, and twelve modal operators of the form $\langle \mathsf{R} \rangle$ such that

$\mathsf{R} \in \{\mathsf{B}, \overline{\mathsf{B}}, \mathsf{D}, \overline{\mathsf{D}}, \mathsf{E}, \overline{\mathsf{E}}, \mathsf{O}, \overline{\mathsf{O}}, \mathsf{A}, \overline{\mathsf{A}}, \mathsf{L}, \overline{\mathsf{L}}\}$ (in what follows we denote this set by $\mathsf{HS}_{\mathsf{rel}}$), as well as the necessity modalities of the form $[\mathsf{R}]$ with $\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}$. Well-formed HS-formulas are defined by the following grammar

$$\varphi := \top \mid \bot \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \langle \mathsf{R} \rangle \varphi \mid [\mathsf{R}]\varphi,$$

where $p \in \mathsf{PROP}$, φ, ψ are HS-formulas, and $\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}$. An HS-model $\mathcal{M}$ is a pair $(\mathbb{D}, V)$ such that $\mathbb{D} = (D, \lhd)$ is a linear order (antisymmetric, transitive, and total relation) of time-points, $I(\mathbb{D}) = \{[x, y] \mid x, y \in D \text{ and } x \lhd y\}$ is a set of all intervals over $\mathbb{D}$, and $V : \mathsf{PROP} \rightarrow \mathcal{P}(I(\mathbb{D}))$ assigns to each propositional variable a set of intervals. Allen's relations between intervals are defined as follows:

$$
\begin{array}{lll}
[x,y]\mathsf{rel}_{\mathsf{B}}[x',y'] & \text{iff} & x = x',\ y' \lhd y \\
[x,y]\mathsf{rel}_{\mathsf{D}}[x',y'] & \text{iff} & x \lhd x',\ y' \lhd y \\
[x,y]\mathsf{rel}_{\mathsf{E}}[x',y'] & \text{iff} & x \lhd x',\ y = y' \\
[x,y]\mathsf{rel}_{\mathsf{O}}[x',y'] & \text{iff} & x \lhd x' \lhd y \lhd y' \\
[x,y]\mathsf{rel}_{\mathsf{A}}[x',y'] & \text{iff} & y = x' \\
[x,y]\mathsf{rel}_{\mathsf{L}}[x',y'] & \text{iff} & y \lhd x'
\end{array}
$$



whereas $\mathsf{rel}_{\overline{\mathsf{B}}}$, $\mathsf{rel}_{\overline{\mathsf{D}}}$, $\mathsf{rel}_{\overline{\mathsf{E}}}$, $\mathsf{rel}_{\overline{\mathsf{O}}}$, $\mathsf{rel}_{\overline{\mathsf{A}}}$, and $\mathsf{rel}_{\overline{\mathsf{L}}}$ are inverses of the respective relations (i.e., $\mathsf{rel}_{\overline{\mathsf{R}}} = \mathsf{rel}_{\mathsf{R}}^{-1}$ for any $\mathsf{R} \in \{\mathsf{B}, \mathsf{D}, \mathsf{E}, \mathsf{O}, \mathsf{A}, \mathsf{L}\}$). The satisfaction relation for a model $\mathcal{M}$ and an interval $[x, y]$ is defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, [x,y] \models \top & & \text{for all } [x,y] \in I(\mathbb{D}); \\
\mathcal{M}, [x,y] \not\models \bot & & \text{for all } [x,y] \in I(\mathbb{D}); \\
\mathcal{M}, [x,y] \models p & \text{iff} & [x,y] \in V(p), \text{ for } p \in \mathsf{PROP}; \\
\mathcal{M}, [x,y] \models \neg\varphi & \text{iff} & \mathcal{M}, [x,y] \not\models \varphi; \\
\mathcal{M}, [x,y] \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, [x,y] \models \varphi \text{ and } \mathcal{M}, [x,y] \models \psi; \\
\mathcal{M}, [x,y] \models \varphi \vee \psi & \text{iff} & \mathcal{M}, [x,y] \models \varphi \text{ or } \mathcal{M}, [x,y] \models \psi; \\
\mathcal{M}, [x,y] \models \varphi \rightarrow \psi & \text{iff} & \text{if } \mathcal{M}, [x,y] \models \varphi \text{ then } \mathcal{M}, [x,y] \models \psi; \\
\mathcal{M}, [x,y] \models \langle \mathsf{R} \rangle \varphi & \text{iff} & \text{there exists } [x',y'] \in I(\mathbb{D}) \text{ such that} \\
 & & [x,y]\mathsf{rel}_{\mathsf{R}}[x',y'] \text{ and } \mathcal{M}, [x',y'] \models \varphi; \\
\mathcal{M}, [x,y] \models [\mathsf{R}] \varphi & \text{iff} & \text{for every } [x',y'] \in I(\mathbb{D}) \text{ such that} \\
 & & [x,y]\mathsf{rel}_{\mathsf{R}}[x',y'] \text{ we have } \mathcal{M}, [x',y'] \models \varphi;
\end{array}
$$

where $\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}$. An HS-formula φ is true in an HS-model $\mathcal{M}$ (in symbols: $\mathcal{M} \models \varphi$) iff for all $[x, y] \in I(\mathbb{D})$ it holds that $\mathcal{M}, [x, y] \models \varphi$.

Decidability of the HS-satisfiability problem depends on the type of underlying temporal frame but for most interesting frames it is undecidable, e.g., over any class of temporal frames that contains an infinite ascending chain it is co-recursively enumerable-hard. (in particular over $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$) [10]. A recently introduced way to obtain a decidable logic is by limiting the 'propositional side' of the language [3,5,6]. In particular, attention was paid to a fragment containing

only Horn clauses and no diamond modalities (the so-called $\mathsf{HS}^{\Box}_{horn}$) which has a relatively low computational complexity and expressive power high enough for practical applications (see, e.g., [3]). A well-formed $\mathsf{HS}^{\Box}_{horn}$-formula φ is defined as follows:

$$\lambda := \top \mid \bot \mid p \mid [\mathsf{R}]\lambda; \qquad \varphi := \lambda \mid [\mathsf{U}](\lambda_1 \wedge \ldots \wedge \lambda_k \rightarrow \lambda) \mid \varphi \wedge \psi;$$

where $p \in \mathsf{PROP}$, $\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}$, and $[\mathsf{U}]$ is a universal modality, i.e., $[\mathsf{U}]\varphi$ is satisfied iff φ is satisfied in every $[x, y] \in I(\mathbb{D})$. Although operators of the form $\langle \mathsf{R} \rangle$ are forbidden in $\mathsf{HS}^{\Box}_{horn}$, clauses with $\langle \mathsf{R} \rangle$ in an antecedent are expressible in the logic as follows:

$$[\mathsf{U}](\lambda_1 \wedge \langle \mathsf{R} \rangle \lambda_2 \rightarrow \lambda_3) \stackrel{\text{df}}{=} [\mathsf{U}](\lambda_2 \rightarrow [\overline{\mathsf{R}}]p) \wedge [\mathsf{U}](p \wedge \lambda_1 \rightarrow \lambda_3),$$

where p is a new propositional variable, i.e., a variable that did not occur in the formula earlier.

The computational complexity of $\mathsf{HS}^{\Box}_{horn}$-satisfiability depends on the type of an underlying temporal frame $\mathbb{D}$. First, there are irreflexive and reflexive frames. Importantly, in the former case, when $\lhd$ is reflexive, point intervals are allowed, i.e., $[x, x] \in I(\mathbb{D})$ for any $x \in D$, and relations $\mathsf{rel}_{\mathsf{R}}$ for $\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}$ are no longer pairwise disjoint. Second distinction is between discrete and dense temporal frames. Interestingly, over irreflexive and discrete frames $\mathsf{HS}^{\Box}_{horn}$-satisfiability is undecidable, whereas in the other three cases it is P-complete (see Table 1).

Table 1. Cumulative results: contributions of this paper are on a gray background.

	Irreflexive	Reflexive
Discrete	$\mathsf{HS}^{\Box}_{horn}$: undecidable [5]	$\mathsf{HS}^{\Box}_{horn}$: P-complete [3]
	$\mathsf{HS}^{\Box,i,@}_{horn}$: undecidable	$\mathsf{HS}^{\Box,i,@}_{horn}$: NP-complete
Dense	$\mathsf{HS}^{\Box}_{horn}$: P-complete [5]	$\mathsf{HS}^{\Box}_{horn}$: P-complete [5]
	$\mathsf{HS}^{\Box,i,@}_{horn}$: NP-complete	$\mathsf{HS}^{\Box,i,@}_{horn}$: NP-complete

In what follows we hybridize $\mathsf{HS}^{\Box}_{horn}$, i.e., we add to the language the second sort of atoms – the set of the so-called *nominals* NOM, and *satisfaction operators* $@_i$ indexed by nominals. We define a well-formed $\mathsf{HS}^{\Box,i,@}_{horn}$-formula φ as follows:

$$\lambda := \top \mid \bot \mid p \mid i \mid [\mathsf{R}]\lambda \mid @_i\lambda; \qquad \varphi := \lambda | [\mathsf{U}](\lambda_1 \wedge \ldots \wedge \lambda_k \rightarrow \lambda) | \varphi \wedge \psi;$$

where $p \in \mathsf{PROP}$, $i \in \mathsf{NOM}$, $\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}$, $[\mathsf{U}]$ is the universal modality. We distinguish *literals* – expressions of the form λ and *clauses* – expressions of the form $[\mathsf{U}](\lambda_1 \wedge \ldots \wedge \lambda_k \rightarrow \lambda)$. For any $\varphi \in \mathsf{HS}^{\Box,i,@}_{horn}$, we call all conjuncts of φ that are not clauses *initial conditions* of φ. A hybrid HS-model $\mathcal{M}$ is a pair $(\mathbb{D}, V)$,

such that $V : \mathsf{ATOM} \to \mathcal{P}(I(\mathbb{D}))$ assigns to each atom ($\mathsf{ATOM} = \mathsf{PROP} \cup \mathsf{NOM}$) a set of intervals with an additional restriction that $V(i)$ is a singleton for any $i \in \mathsf{NOM}$. The additional satisfaction relation conditions for nominals and satisfaction operators are:

$$\begin{array}{ll} \mathcal{M}, [x,y] \models i & \text{iff } V(i) = \{[x,y]\}, \text{ for } i \in \mathsf{NOM}; \\ \mathcal{M}, [x,y] \models @_i\varphi & \text{iff } \mathcal{M}, [x',y'] \models \varphi, \text{ where } V(i) = \{[x',y']\} \text{ and } i \in \mathsf{NOM}. \end{array}$$

Hybridization increases expressive power of the logic, e.g., it enables to express identity of two intervals by $@_i j$. In the following sections we show the main contribution of this paper (see Table 1), i.e., that $\mathsf{HS}^{\Box,i,@}_{horn}$-satisfiability is NP-complete over reflexive and over irreflexive and dense time frames. The undecidability of $\mathsf{HS}^{\Box,i,@}_{horn}$ over irreflexive and discrete frames is a direct consequence of the already known undecidability of $\mathsf{HS}^{\Box}_{horn}$ over such frames [5].

3 NP-Hardness

In this section, we prove the lower bound of $\mathsf{HS}^{\Box,i,@}_{horn}$-satisfiability.

Theorem 1. *$\mathsf{HS}^{\Box,i,@}_{horn}$-satisfiability over linear orders is* NP*-hard.*

Proof. To prove NP-hardness of $\mathsf{HS}^{\Box,i,@}_{horn}$-satisfiability we construct a polynomial reduction from 3SAT problem (known to be NP-complete – see, e.g., [12]).

> 3SAT is the following decision problem:
> **Input**: $\varphi = (l^1_1 \vee l^2_1 \vee l^3_1) \wedge \ldots \wedge (l^1_n \vee l^2_n \vee l^3_n)$, where each l^j_i is a propositional literal, i.e., a propositional variable or its negation.
> **Output**: "yes" if φ is PC-satisfiable (PC is classical propositional calculus), "no" otherwise.

Fix a propositional calculus formula $\varphi = (l^1_1 \vee l^2_1 \vee l^3_1) \wedge \ldots \wedge (l^1_n \vee l^2_n \vee l^3_n)$ and let $x_1, \ldots, x_m$ be all propositional variables occurring in φ. We map φ into an $\mathsf{HS}^{\Box,i,@}_{horn}$-formula by means of the following translation:

$$\tau(\varphi) = \bigwedge_{1 \leq k \leq m} \psi_k \wedge \bigwedge_{1 \leq s \leq n} \chi_s,$$

where ψ_k and χ_s are defined in subsequent paragraphs. In $\tau(\varphi)$ we will use pairwise distinct nominals $i_0, i_1, \ldots, i_m$ and pairwise distinct propositional variables $x_1, \ldots, x_m, \overline{x_1}, \ldots, \overline{x_m}$.

First, for any $k \in \{1, \ldots, m\}$ let:

$$\psi_k = [\mathsf{U}](i_0 \wedge \langle \mathsf{L} \rangle i_k \to \overline{x_k}) \tag{1}$$

$$\wedge \bigwedge_{\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}/\{\mathsf{L}\}} [\mathsf{U}](i_0 \wedge \langle \mathsf{R} \rangle i_k \to x_k) \quad \wedge \quad [\mathsf{U}](i_0 \wedge i_k \to x_k) \tag{2}$$

$$\wedge\, [\mathsf{U}](x_k \wedge \overline{x_k} \to \bot), \tag{3}$$

where (according to the statement in Sect. 2 that we can express a diamond modality in the antecedent) $[\mathsf{U}](i \wedge \langle \mathsf{R} \rangle j \rightarrow p)$ is treated as an abbreviation in the following way:

$$[\mathsf{U}](i \wedge \langle \mathsf{R} \rangle j \rightarrow p) \stackrel{\text{df}}{=} [\mathsf{U}](j \rightarrow [\overline{\mathsf{R}}]q) \wedge [\mathsf{U}](q \wedge i \rightarrow p),$$

where p is a fresh variable (i.e., a variable not occurring in the formula anywhere else). Formula ψ_k enables us to simulate negation of x_k by means of $\overline{x_k}$. The 'trick' we use to encode such a negation consists in noticing that the interval denoted by i_k must be in some Allen's relation with the interval denoted by i_0. We enforce that (1) $\overline{x_k}$ is satisfied in i_0 if i_k is accessible from i_0 by means of $\mathsf{rel}_{\mathsf{L}}$, and (2) otherwise x_k is satisfied in i_0. Finally, (3) x_k and $\overline{x_k}$ cannot be satisfied in the same interval. Hence we have enforced that in i_0 a variable x_k is satisfied iff $\overline{x_k}$ is not satisfied there.

Second, for any $s \in \{1, \ldots, n\}$ we define:

$$\chi_s = [\mathsf{U}]\Big(i_0 \wedge \mathsf{neg}\left(l_s^1\right) \wedge \mathsf{neg}\left(l_s^2\right) \wedge \mathsf{neg}\left(l_s^3\right) \rightarrow \bot\Big),$$

where for any propositional literal l in φ we define

$$\mathsf{neg}(l) = \begin{cases} \overline{x_t}, & \text{if } l = x_t, \\ x, & \text{if } l = \neg x_t, \end{cases} \quad \text{for any } t \in (1, \ldots, m).$$

A formula χ_s assures that a clause $(l_s^1 \vee l_s^2 \vee l_s^3)$ is satisfied in i_0. It does it by excluding models in which negations of all three propositional literals occurring in the clause are simultaneously satisfied in i_0.

Notice that $\tau(\varphi)$ is a conjunction of formulas each preceded by the universal modality $[\mathsf{U}]$. Hence, $\tau(\varphi)$ is HS-satisfiable iff it is true (i.e., satisfied in all intervals) in some HS-model (we will use this fact afterwards in the proof). The number of formulas of the form ψ_k and χ_s is linear in the size of φ, and each ψ_k and χ_s is of a constant size. Hence the translation τ is feasible in polynomial time with respect to the size of φ. To finish the proof it remains to show that the following conditions are equivalent:

1. φ is PC-satisfiable;
2. $\tau(\varphi)$ is HS-satisfiable.

$(1 \Rightarrow 2)$ Assume that φ is PC-satisfiable. Then, there exists a PC-model (valuation) $\mathsf{v} : \mathsf{PROP}(\varphi) \rightarrow \{0, 1\}$ (by $\mathsf{PROP}(\varphi)$ we denote a set of all propositional variables occurring in φ) such that $\mathsf{v} \models_{\mathsf{PC}} \varphi$ (where $\models_{\mathsf{PC}}$ is a PC-satisfaction relation). We construct an HS-model $\mathcal{M} = (\mathbb{D}, V)$ as follows (see also Fig. 1):

– $\mathbb{D} = (D, \lhd)$ is a linear order;
– $V : \mathsf{ATOM}(\tau(\varphi)) \rightarrow \mathcal{P}\left(I(\mathbb{D})\right)$ is such that:
  - a, b, c, d are any pairwise distinct elements of D with $a \lhd b \lhd c \lhd d$;
  - $V(i_0) = \{[a, b]\}$;
  - for each $x_k \in \mathsf{PROP}(\varphi)$:

* if $\mathsf{v}(x_k) = 1$, then $V(i_k) = \{[a, b]\}$ and $V(x_k) = \{[a, b]\}$;
* if $\mathsf{v}(x_k) = 0$, then $V(i_k) = \{[c, d]\}$ and $V(\overline{x_k}) = \{[a, b]\}$.

We show that $\mathcal{M} \models \tau(\varphi)$. First, for any $x_k \in \mathsf{PROP}(\varphi)$ we have $\mathcal{M} \models \psi_k$ since x_k is satisfied in i_0 if $V(i_k) = \{[a, b]\}$ and $\overline{x_k}$ is satisfied in i_0 if $V(i_k) = \{[c, d]\}$, and $x_k, \overline{x_k}$ are not satisfied in any interval simultaneously. Furthermore, for any $s \in \{1, \dots, n\}$ in the clause $(l_s^1 \vee l_s^2 \vee l_s^3)$ in φ at least one of its propositional literals – without loss of generality say l_s^1 – is satisfied in v. From the construction of V it follows that $\mathsf{neg}(l_s^1)$ is not satisfied in i_0, so $\mathcal{M} \models \chi_s$.

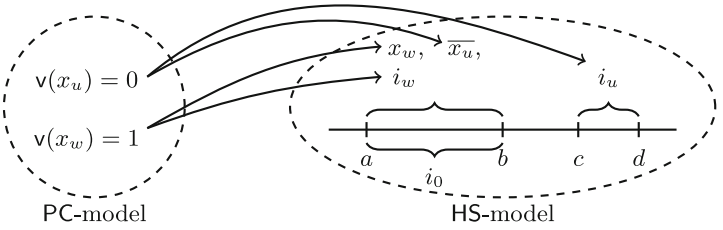


Fig. 1. Construction of a HS-model from a PC-model.

$(1 \Leftarrow 2)$ Assume that $\tau(\varphi)$ is HS-satisfiable. Fix an HS-model $\mathcal{M}$ such that $\mathcal{M} \models \tau(\varphi)$. We construct a PC-model v (as presented in Fig. 2) such that for any propositional variable $x_k \in \mathsf{PROP}(\varphi)$:

$$\mathsf{v}(x_k) = \begin{cases} 1 & \text{if} \quad \mathcal{M}, [i_0^-, i_0^+] \models x_k; \\ 0 & \text{if} \quad \mathcal{M}, [i_0^-, i_0^+] \models \overline{x_k}; \end{cases}, \quad \text{where } V(i_0) = \{[i_0^-, i_0^+]\}.$$

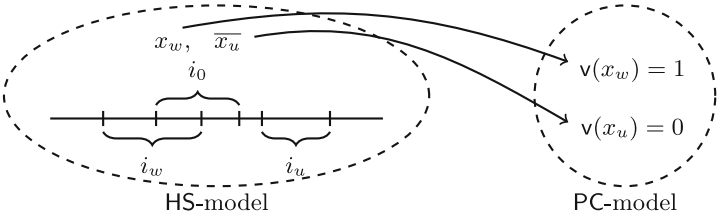


Fig. 2. Construction of a PC-model from a HS-model.

We show that $\mathsf{v} \models_{\mathsf{PC}} \varphi$. Fix a clause $(l_s^1 \vee l_s^2 \vee l_s^3)$ in φ. Since $\mathcal{M} \models \chi_s$, one of $\mathsf{neg}(l_s^1), \mathsf{neg}(l_s^2), \mathsf{neg}(l_s^3)$ – without loss of generality say $\mathsf{neg}(l_s^1)$ – is not satisfied in i_0. If l_s^1 is a propositional variable, say x_t for some $t \in (1, \dots, m)$, then $\mathcal{M}, [i_0^-, i_0^+] \models x_t$. By the construction $\mathsf{v} \models_{\mathsf{PC}} x_t$, hence $\mathsf{v} \models_{\mathsf{PC}} (l_s^1 \vee l_s^2 \vee l_s^3)$. On the other hand, if l_s^1 is a negated propositional variable, say $\neg x_t$, then $\mathcal{M}, [i_0^-, i_0^+] \models \overline{x_t}$. Hence $\mathsf{v} \models_{\mathsf{PC}} \neg x_t$, and $\mathsf{v} \models_{\mathsf{PC}} (l_s^1 \vee l_s^2 \vee l_s^3)$. □

Notice that the above theorem holds regardless of whether $\mathbb{D}$ is reflexive or irreflexive, and whether it is discrete or dense. Moreover, the proof does not use $@_i$ operators (there are no $@_i$ operators in a formula $\tau(\varphi)$). Hence, the nominals already make the logic NP-hard and consequently, NP-hardness holds also for the logic without $@_i$ operators.

4 Membership in NP

To prove that $\mathsf{HS}^{\Box,i,@}_{horn}$-satisfiability is in NP over reflexive, as well as over irreflexive and dense frames we exploit a technique that was presented in [5, Theorem 3.5], and [3, Theorem 6]. The main idea of our proof is that for a fixed interval $[a,b]$ and a fixed interpretation of nominals we are able to check in polynomial time if a given $\mathsf{HS}^{\Box,i,@}_{horn}$-formula is satisfiable in $[a,b]$ (Lemma 4). Then, we will show that there is only a bounded (by an exponential function in the size of the formula) number of significantly different choices of $[a,b]$ and interpretations of nominals. Hence, we can nondeterministically 'guess' them in NP. We start by defining the following problem.

$(a,b,\mathcal{I})$-*satisfaction* over $\mathbb{D}$ for a fixed $[a,b] \in I(\mathbb{D})$, $\mathcal{I} : \mathsf{NOM}(\varphi) \to I(\mathbb{D})$, and a linear order $\mathbb{D} = (D, \lhd)$ is the following decision problem:
Input: an $\mathsf{HS}^{\Box,i,@}_{horn}$-formula φ.
Output: "yes" if there is an HS-model $\mathcal{M} = (\mathbb{D}, V)$ with $V(i) = \{\mathcal{I}(i)\}$ for $i \in \mathsf{NOM}(\varphi)$ such that $\mathcal{M}, [a,b] \models \varphi$, "no" otherwise.

If the answer is positive, we say that φ is $(a,b,\mathcal{I})$-satisfiable over $\mathbb{D}$. At first, we will construct a model that will enable us to check if φ is $(a,b,\mathcal{I})$-satisfiable over $\mathbb{D}$. Let φ be an $\mathsf{HS}^{\Box,i,@}_{horn}$-formula, $\mathbb{D} = (D, \lhd)$ be a linear order, $[a,b] \in I(\mathbb{D})$, and $\mathcal{I} : \mathsf{NOM}(\varphi) \to I(\mathbb{D})$. We will define a set of triples of the form (ψ, x, y), where each such triple has an intuitive meaning that in order to satisfy φ in $[a,b]$, formula ψ must be satisfied in $[x,y]$. We start with the set:

$$\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}} = \big\{(\lambda, a, b) \mid \lambda \text{ is an initial condition of } \varphi\big\} \cup \big\{(\top, x, y) \mid [x,y] \in I(\mathbb{D})\big\} \cup \big\{(i, x, y) \mid i \in \mathsf{NOM}(\varphi) \text{ and } \mathcal{I}(i) = [x,y]\big\}.$$

$\mathsf{cl}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ is the result of applying non-recursively the below rules to $\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$:

(cl1) if $([\mathsf{R}]\lambda, x, y) \in \mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$, then add to $\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ all (λ, x', y') such that $[x', y'] \in I(\mathbb{D})$ and $[x,y]\mathsf{rel}_{\mathsf{R}}[x', y']$;
(cl2) if $(\lambda, x', y') \in \mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ for all $[x', y'] \in I(\mathbb{D})$ such that $[x,y]\mathsf{rel}_{\mathsf{R}}[x', y']$ and $[\mathsf{R}]\lambda$ occurs in φ, then add $([\mathsf{R}]\lambda, x, y)$ to $\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$;
(cl3) if $[\mathsf{U}](\lambda_1 \wedge \ldots \wedge \lambda_k \to \lambda)$ occurs in φ and $(\lambda_j, x, y) \in \mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ for all $j \in \{1, \ldots, k\}$, then add (λ, x, y) to $\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$;
(cl4) if $(@_i\lambda, x, y) \in \mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$, then add (λ, x', y') to $\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ where $[x', y'] = \mathcal{I}(i)$;

(cl5) if $(\lambda, x', y') \in \mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ for some $i \in \mathsf{NOM}(\varphi)$ with $\mathcal{I}(i) = [x', y']$, and $@_i\lambda$ occurs in φ, then add $(@_i\lambda, x, y)$ to $\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ for all $[x, y] \in I(\mathbb{D})$.

We define the following sets, obtained by subsequent applications of cl to $\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$:

$$\begin{aligned}
\mathsf{cl}^0\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right) &= \mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}};\\
\mathsf{cl}^{\alpha+1}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right) &= \mathsf{cl}\left(\mathsf{cl}^{\alpha}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)\right), \text{ for } \alpha+1 \text{ a succesor ordinal;}\\
\mathsf{cl}^{\beta}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right) &= \bigcup_{\gamma<\beta} \mathsf{cl}^{\gamma}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right), \text{ for } \gamma \text{ an ordinal, and } \beta \text{ a limit ordinal;}\\
\mathsf{cl}^{*}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right) &= \bigcup_{\gamma \text{ an ordinal}} \mathsf{cl}^{\gamma}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right).
\end{aligned}$$

Next we construct $\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}} = (\mathbb{D}, V)$ such that for any $at \in \mathsf{ATOM}(\varphi) \cup \{\top, \bot\}$:

$$V(at) = \left\{[x, y] \mid (at, x, y) \in \mathsf{cl}^{*}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)\right\}.$$

We say that a set A satisfies the condition (mod) if the following holds:

(mod) $(\bot, x, y) \notin A$ for any $[x, y] \in I(\mathbb{D})$, and
if $(i, x, y) \in A$ for some $i \in \mathsf{NOM}(\varphi)$,
then $(i, x', y') \notin A$ for any $[x', y'] \neq [x, y]$.

A straightforward consequence of the conditions (cl1)–(cl5) is: if $\mathsf{cl}^{*}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ satisfies (mod), then $\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ is an HS-model. Then, $\mathsf{cl}^{*}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ determines in which intervals particular literals of φ are satisfied in $\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ in the following sense.

Lemma 1. *If* $\mathsf{cl}^{*}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ *satisfies (*mod*), then for any literal* λ *in* φ*, and any* $[x, y] \in I(\mathbb{D})$*, the following conditions are equivalent:*

1. $(\lambda, x, y) \in \mathsf{cl}^{*}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$;
2. $\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, [x, y] \models \lambda$.

Proof (Sketch). ($1 \Rightarrow 2$) The implication holds for $\mathsf{cl}^{0}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$. Then, by transfinite induction on an ordinal γ the implication holds for any $\mathsf{cl}^{\gamma}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ and consequently for $\mathsf{cl}^{*}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$.

($1 \Leftarrow 2$) The implication holds for atoms by the definition of $\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$, for $\top$ by the definition of $\mathsf{cl}^{0}\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$, and for $\bot$ by (mod). Then, by induction on a literal structure the implication holds for any literal in φ. □

Lemma 2. *Let φ be an $\mathsf{HS}_{horn}^{\Box,i,@}$-formula. The following are equivalent:*

1. *φ is $(a,b,\mathcal{I})$-satisfiable over $\mathbb{D}$;*
2. $\mathsf{cl}^*\left(\mathfrak{V}_{\varphi,\mathbb{D}}^{(a,b,\mathcal{I})}\right)$ *satisfies* (mod).

Proof. $(1 \Rightarrow 2)$ Assume φ is $(a,b,\mathcal{I})$-satisfiable over $\mathbb{D}$, so there is an HS-model $\mathcal{M} = (\mathbb{D}, V)$ with $V(i) = \{\mathcal{I}(i)\}$ for any $i \in \mathsf{NOM}$, and $\mathcal{M}, [a,b] \models \varphi$. Define:

$$\mathfrak{V} = \{(\lambda, x, y) \mid [x,y] \in I(\mathbb{D}), \lambda \text{ is a literal occuring in } \varphi, \text{ and } \mathcal{M}, [x,y] \models \lambda\}.$$

It is easy to see that $\mathsf{cl}^0\left(\mathfrak{V}_{\varphi,\mathbb{D}}^{(a,b,\mathcal{I})}\right) \subseteq \mathfrak{V}$ and $\mathfrak{V}$ is closed under (cl1)–(cl5). As a result, $\mathsf{cl}^*\left(\mathfrak{V}_{\varphi,\mathbb{D}}^{(a,b,\mathcal{I})}\right) \subseteq \mathfrak{V}$. $\mathcal{M}$ is an HS-model, therefore it is easy to show that $\mathfrak{V}$ satisfies (mod). $\mathsf{cl}^*\left(\mathfrak{V}_{\varphi,\mathbb{D}}^{(a,b,\mathcal{I})}\right) \subseteq \mathfrak{V}$, therefore $\mathsf{cl}^*\left(\mathfrak{V}_{\varphi,\mathbb{D}}^{(a,b,\mathcal{I})}\right)$ also satisfies (mod). $(1 \Leftarrow 2)$ Assume $\mathsf{cl}^*\left(\mathfrak{V}_{\varphi,\mathbb{D}}^{(a,b,\mathcal{I})}\right)$ satisfies (mod). Then, by Lemma 1 it is easy to show that $\mathcal{K}_{\varphi,\mathbb{D}}^{(a,b,\mathcal{I})}, [a,b] \models \varphi$. Therefore, φ is $(a,b,\mathcal{I})$-satisfiable over $\mathbb{D}$. □

The proof of Lemma 2 ($2 \Rightarrow 1$ implication) leads to the following statement.

Corollary 1. *If φ is $(a,b,\mathcal{I})$-satisfiable over $\mathbb{D}$, then $\mathcal{K}_{\varphi,\mathbb{D}}^{(a,b,\mathcal{I})}, [a,b] \models \varphi$.*

Next, we define $\mathfrak{Z}_{\varphi,\mathbb{D}}^{(a,b,\mathcal{I})}$ – a *bounded morphic image* of $\mathcal{K}_{\varphi,\mathbb{D}}^{(a,b,\mathcal{I})}$ which will allow us to check $(a,b,\mathcal{I})$-satisfiability by inspection of a bounded (by a polynomial in the length of the formula φ) sized $\mathfrak{Z}_{\varphi,\mathbb{D}}^{(a,b,\mathcal{I})}$ (for a description of bounded morphisms see [9, Chap. 5]). Let φ be an $\mathsf{HS}_{horn}^{\Box,i,@}$-formula, $\mathbb{D} = (D, \lhd)$ a linear order, and $\mathcal{I} : \mathsf{NOM}(\varphi) \to I(\mathbb{D})$. Fix $[a,b] \in I(\mathbb{D})$ and the set X of all intervals denoted by nominals. Let $x_1, \ldots, x_n$ be the sequence (without multiple occurrences) of endpoints of intervals in $X \cup \{[a,b]\}$ in the ascending order, i.e., $x_1 \lhd \ldots \lhd x_n$. We define the set of *sections* of D:

$$\mathsf{all\_sec}(a,b,\mathcal{I}) = \big\{(-\infty, x_1), [x_1,x_1], (x_1,x_2), [x_2,x_2], (x_2,x_3), \\ \ldots, (x_{n-1},x_n), [x_n,x_n], (x_n,+\infty)\big\}.$$

Some sections may be empty, e.g., section $(3,4)$ in the case of a discrete $\mathbb{D}$. Hence, we define the set of all nonempty sections $\mathsf{sec}(a,b,\mathcal{I})$:

$$\mathsf{sec}(a,b,\mathcal{I}) = \big\{\sigma \in \mathsf{all\_sec}(a,b,\mathcal{I}) \mid x \in \sigma \text{ for some } x \in D\big\}.$$

$\mathsf{sec}(a,b,\mathcal{I})$ is a partition of D into at most $2\big(|\mathsf{NOM}(\varphi)| + 2\big) + 1$ sections. For any $\sigma, \sigma' \in \mathsf{sec}(a,b,\mathcal{I})$ we write $\sigma \preccurlyeq \sigma'$ if for some $x \in \sigma$ and $y \in \sigma'$ it holds that $[x,y] \in I(\mathbb{D})$. In Fig. 3 we present an example of $\mathsf{sec}(a,b,\mathcal{I})$, in the case of $\mathsf{NOM} = \{i\}$, $\mathcal{I}(i) = [x_1,x_2]$, $a = x_2$, and $b = x_3$.

Next, we define *zones*:

$$\zeta_{\sigma,\sigma'} = \big\{[x,y] \in I(\mathbb{D}) \mid x \in \sigma, y \in \sigma'\big\} \text{ for } \sigma, \sigma' \in \mathsf{sec}(a,b,\mathcal{I}), \sigma \preccurlyeq \sigma'.$$

While considering partition depicted in Fig. 3 we have, e.g., $\zeta_{\sigma_2,\sigma_6} = \{[x_1,x_3]\}$.

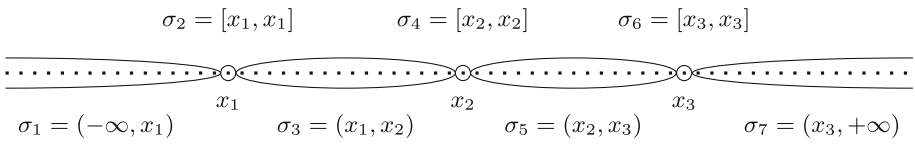


Fig. 3. A set of nonempty sections $\{\sigma_1, \dots \sigma_7\}$ determined by endpoints x_1, x_2, x_3.

We define a Kripke model $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}} = (Z, \{\mathsf{REL}_\mathsf{R}\}_{\mathsf{R}\in\mathsf{HS}_\mathsf{rel}}, V')$ such that:

- $Z = \{\zeta_{\sigma,\sigma'} \mid \sigma, \sigma' \in \mathsf{sec}(a,b,\mathcal{I}),\ \sigma \preccurlyeq \sigma',\ \sigma \neq \sigma'\} \cup \{\zeta_\sigma \mid \sigma \in \mathsf{sec}(a,b,\mathcal{I})\}$;
- $\zeta \mathsf{REL}_\mathsf{R} \zeta'$ iff $[x,y]\mathsf{rel}_\mathsf{R}[x',y']$ for some $[x,y] \in \zeta$ and $[x',y'] \in \zeta'$;
- $V'(at) = \{\zeta \mid f^{-1}(\zeta) \subseteq V(at)\}$ for any atom $at \in \mathsf{ATOM}(\varphi)$;

where $\mathsf{R} \in \mathsf{HS}_\mathsf{rel}$ and $f : I(\mathbb{D}) \to Z$ satisfies the following condition:

$$f([x,y]) = \zeta \quad \text{iff} \quad [x,y] \in \zeta.$$

Importantly, the size of $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ is bounded by a polynomial in the length of the formula φ. Indeed, as we have already stated, $\mathsf{sec}(a,b,\mathcal{I})$ is a partition of D into at most $2\big(|\mathsf{NOM}(\varphi)| + 2\big) + 1 = h$ sections, so there are at most $\frac{(1+h)}{h}$ zones, which constitute the universe Z.

Lemma 3. *$f : I(\mathbb{D}) \to Z$ is a surjective bounded morphism from $\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ to $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$, i.e., it is a surjection and for any $\mathsf{R} \in \mathsf{HS}_\mathsf{rel}$ the following conditions hold:*

(bm1) *if $[x,y]\mathsf{rel}_\mathsf{R}[x',y']$ then $f([x,y])\mathsf{REL}_\mathsf{R} f([x',y'])$;*
(bm2) *if $\zeta \mathsf{REL}_\mathsf{R} \zeta'$ then for every $[x,y] \in f^{-1}(\zeta)$ there exists $[x',y'] \in f^{-1}(\zeta')$ such that $[x,y]\mathsf{rel}_\mathsf{R}[x',y']$;*
(bm3) *for any $at \in \mathsf{ATOM}(\varphi)$ and any $[x,y] \in I(\mathbb{D})$ it holds that $\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, [x,y] \models at$ iff $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, f([x,y]) \models at$.*

The proof of Lemma 3 is analogous to the proof of a bounded morphism introduced to prove P-completeness of $\mathsf{HS}^{\Box}_{horn}$ [5, pp. 8–10]. Since f is a surjective bounded morphism from $\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ to $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$, it is known that $\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, [x,y] \models \varphi$ iff $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, f([x,y]) \models \varphi$ for any $[x,y] \in I(\mathbb{D})$, and any HS-formula φ [9, Chap. 5, Corollary 16]. As a result, by Corollary 1 we obtain the following statement.

Corollary 2. *Let φ be an $\mathsf{HS}^{\Box,i,@}_{horn}$-formula and $\mathbb{D}$ a reflexive, or a irreflexive and dense linear order. The following conditions are equivalent for any $[a,b] \in I(\mathbb{D})$, and any $\mathcal{I}$ such that $\mathcal{I} : \mathsf{NOM}(\varphi) \to I(\mathbb{D})$:*

1. *φ is $(a,b,\mathcal{I})$-satisfiable over $\mathbb{D}$;*
2. *$\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, [a,b] \models \varphi$;*
3. *$\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, f([a,b]) \models \varphi$.*

Hence, our aim is now to show that we can check in P if $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, f([a,b]) \models \varphi$.

Lemma 4. *The problem of $(a,b,\mathcal{I})$-satisfiability over $\mathbb{D}$ can be computed in polynomial time with respect to $|\varphi|$ if $\mathbb{D}$ is reflexive or irreflexive and dense.*

Proof (Sketch). To check if an $\mathsf{HS}^{\Box,i,@}_{horn}$-formula φ is $(a,b,\mathcal{I})$-satisfiable in (reflexive or irreflexive and dense) $\mathbb{D}$ it suffices (by Corollary 2) to construct $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ and then check if $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, f([a,b]) \models \varphi$. To construct $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ let:

$$\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}} = \big\{(\lambda, f([a,b])) \mid \lambda \text{ is an initial condition of } \varphi\big\} \cup \big\{(\top,\zeta) \mid \zeta \in Z\big\} \cup \big\{(i, f([x,y])) \mid i \in \mathsf{NOM}(\varphi) \text{ and } \mathcal{I}(i) = [x,y]\big\}.$$

Then define rules (cl1′)–(cl5′) analogously to (cl1)–(cl5), i.e., let:

(cl1′) if $([\mathsf{R}]\lambda,\zeta) \in \mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$, then add to $\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ all (λ,ζ') such that $\zeta \mathsf{REL}_{\mathsf{R}} \zeta'$;
(cl2′) if $(\lambda,\zeta') \in \mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ for all $\zeta' \in Z$ such that $\zeta \mathsf{rel}_{\mathsf{R}} \zeta'$ and $[\mathsf{R}]\lambda$ occurs in φ, then add $([\mathsf{R}]\lambda,\zeta)$ to $\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$;
(cl3′) if $[\mathsf{U}](\lambda_1 \wedge \ldots \wedge \lambda_k \rightarrow \lambda)$ occurs in φ and $(\lambda_j,\zeta) \in \mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ for all $j \in \{1,\ldots,k\}$, then add (λ,ζ) to $\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$;
(cl4′) if $(@_i\lambda,\zeta) \in \mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$, then add (λ,ζ') to $\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ where $\{\mathcal{I}(i)\} = \zeta'$;
(cl5′) if $(\lambda,\zeta') \in \mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ for some $i \in \mathsf{NOM}(\varphi)$ with $\{\mathcal{I}(i)\} = \zeta'$ and $@_i\lambda$ occurs in φ, then add $(@_i\lambda,\zeta)$ to $\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ for all $\zeta \in Z$.

$\mathsf{cl}'\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ is the result of applying non-recursively the rules (cl1′)–(cl5′) to $\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ and:

$$\mathsf{cl}'^{0}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right) = \mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}};$$
$$\mathsf{cl}'^{k+1}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right) = \mathsf{cl}'\left(\mathsf{cl}'^{k}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)\right).$$

We show that a subsequent application of (cl1′)–(cl5′) to $\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ reaches a fixed point after at most $|Z| \cdot |\varphi|$ iterations. The initial set $\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ contains elements of the form (λ,ζ) such that λ is a subformula of φ or it is a constant $\top$, where $\zeta \in Z$. Then, it is easy to see that each application of rules (cl1′)–(cl5′) results in adding elements of the form (λ,ζ), where λ is a subformula of φ and $\zeta \in Z$. Obviously, there are at most $|Z| \cdot |\varphi|$ such pairs, so after at most $|Z| \cdot |\varphi|$ iterations cl' reaches a fixed point.

We introduce a condition analogous to (mod), namely we say that a set A satisfies condition (mod′) if the following holds:

(mod′) $(\bot,\zeta) \notin A$ for any $\zeta \in Z$, and
if $(i,\zeta) \in A$ for some $i \in \mathsf{NOM}(\varphi)$,
then $(i,\zeta') \notin A$ for any $\zeta' \neq \zeta$.

We will prove now that $\mathsf{cl}'^{|Z|\cdot|\varphi|}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ enables us to construct the previously defined $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}} = \left(Z, \{\mathsf{R}'_{\mathsf{R}\in\mathsf{HS}_{\mathsf{rel}}}\}, V'\right)$ in the sense that the following lemma holds.

Lemma 5. *If* $\mathsf{cl}'^{|Z|\cdot|\varphi|}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ *satisfies* (mod′)*, then for any literal* λ *in* φ*, and any* $\zeta \in Z$ *the following conditions are equivalent:*

1. $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, \zeta \models \lambda$;
2. $(\lambda,\zeta) \in \mathsf{cl}'^{|Z|\cdot|\varphi|}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$.

Proof (Sketch). Notice that by (bm3) and Lemma 1 for any literal λ in φ, and any $\zeta \in Z$ the following conditions are equivalent:

- $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, \zeta \models \lambda$;
- $\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, [x,y] \models \lambda$ for some $[x,y] \in \zeta$;
- $\mathcal{K}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, [x,y] \models \lambda$ for all $[x,y] \in \zeta$;
- $(\lambda,x,y) \in \mathsf{cl}^*\left(\mathfrak{V}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ for all $[x,y] \in \zeta$.

Since rules (cl1′)–(cl5′) are analogous to (cl1)–(cl5), it is easy to show that another equivalent condition is

- $(\lambda,\zeta) \in \mathsf{cl}'^{|Z|\cdot|\varphi|}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$.

□

Finally, the following lemma shows how to check if $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, f([a,b]) \models \varphi$.

Lemma 6. *Let* φ *be a* $\mathsf{HS}^{\Box,i,@}_{horn}$*-formula. The following conditions are equivalent:*

1. $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, f([a,b]) \models \varphi$;
2. $\mathsf{cl}'^{|Z|\cdot|\varphi|}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ *satisfies condition* (mod′)*.*

Proof. $(1 \Rightarrow 2)$ Assume $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, f([a,b]) \models \varphi$. Let us define the following set:

$$\mathfrak{U} = \{(\lambda,\zeta) \mid \zeta \in Z, \lambda \text{ is a literal occuring in } \varphi, \text{ and } \mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, \zeta \models \lambda\}.$$

It is easy to see that $\mathsf{cl}'^{0}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right) \subseteq \mathfrak{U}$ and $\mathfrak{U}$ is closed under (cl1′)–(cl5′). As a result, $\mathsf{cl}'^{|Z|\cdot|\varphi|}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right) \subseteq \mathfrak{U}$. $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ is a model, therefore it is easy to see that $\mathsf{cl}'^{|Z|\cdot|\varphi|}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ must satisfy (mod′). $\mathsf{cl}'^{|Z|\cdot|\varphi|}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right) \subseteq \mathfrak{U}$, therefore $\mathsf{cl}'^{|Z|\cdot|\varphi|}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ also satisfies (mod′).

$(1 \Leftarrow 2)$ Assume $\mathsf{cl}'^{|Z|\cdot|\varphi|}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ satisfies (mod′). Then, with Lemma 5 it is easy to show that $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, [a,b] \models \varphi$. □

We have shown that after $|Z| \cdot |\varphi|$ applications of cl' to $\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ we reach a fixed point. To check if $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, f([a,b]) \models \varphi$ it remains to check if $\mathsf{cl}'^{|Z|\cdot|\varphi|}\left(\mathfrak{U}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}\right)$ satisfies (mod′) (by Lemma 6). The whole procedure is in P with respect to the size of φ, which by Corollary 2 ends the proof. □

Theorem 2. $\mathsf{HS}^{\Box,i,@}_{horn}$*-satisfiability over reflexive, or irreflexive and dense time frames is in* NP.

Proof. Fix an $\mathsf{HS}^{\Box,i,@}_{horn}$-formula φ and $\mathbb{D}$. The construction of $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ depends only on the sequence $x_1 \circ \ldots \circ x_n$ of endpoints of intervals in $X \cup \{[a,b]\}$ where $\circ \in \{<,=\}$, and X is a set of all intervals denoted by nominals. To encode such a sequence it suffices to use at most $n \cdot \log(n) \cdot n \cdot 2$ bits ($\log(n)$ bits for each x_k, 1 bit for each $\circ$, and 1 bit to separate x_k's from $\circ$'s). Since $n \leq (2 + 2 \cdot |\varphi|)$, the representation of the sequence is of polynomial size wrt $|\varphi|$, and can be nondeterministically 'guessed' and written on the tape by a machine working in NP. After such a 'guess' construct $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}$ and check if $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, f([a,b]) \models \varphi$, which by Lemma 4 can be done in polynomial time. By Corollary 2 φ is HS-satisfiable iff there is a sequence $x_1 \circ \ldots \circ x_n$ such that $\mathfrak{Z}^{(a,b,\mathcal{I})}_{\varphi,\mathbb{D}}, f([a,b]) \models \varphi$. Hence, $\mathsf{HS}^{\Box,i,@}_{horn}$-satisfiability is in NP. □

5 Conclusions

In this paper, we have introduced a hybridized version of the logic $\mathsf{HS}^{\Box}_{horn}$ and proved NP-completeness of its satisfiability problem in the case of reflexive, as well as irreflexive and dense time frames. Such hybridization provides referentiality, i.e., the capability of referring to particular intervals – which plays a key role in temporal knowledge representation. It seems that hybridization of sub-propositional fragments of HS is a promising line of research and may provide expressive and decidable referential interval logics. As a future work we plan to hybridize other fragments of HS, study their computational complexity, expressive power, and potential areas of application.

Acknowledgements. The author is supported by the Polish National Science Centre grant DEC-2011/02/A/HS1/00395. He thanks Michał Zawidzki for valuable comments and stimulating discussions on hybridization of temporal logics. Moreover, the author thanks Joanna Golińvska-Pilarek, Roman Kontchakov, Carl Schultz, Michael Zakharyaschev and anonymous reviewers for their comments and suggestions on how to improve this paper.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM **26**(11), 832–843 (1983)
2. Areces, C., Blackburn, P., Marx, M.: The computational complexity of hybrid temporal logics. Logic J. IGPL **8**(5), 653–679 (2000)

3. Artale, A., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M.: Tractable interval temporal propositional and description logics. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015), pp. 1417–1423 (2015)
4. Blackburn, P.: Representation, reasoning, and relational structures: a hybrid logic manifesto. Logic J. IGPL **8**(3), 339–625 (2000)
5. Bresolin, D., Kurucz, A., Muñoz-Velasco, E., Ryzhikov, V., Sciavicco, G., Zakharyaschev, M.: Horn fragments of the halpern-shoham interval temporal logic. Technical report. arXiv preprint arXiv:1604.03515 (2016)
6. Bresolin, D., Muñoz-Velasco, E., Sciavicco, G.: Sub-propositional fragments of the interval temporal logic of Allen's relations. In: Fermé, E., Leite, J. (eds.) JELIA 2014. LNCS (LNAI), vol. 8761, pp. 122–136. Springer, Heidelberg (2014). doi:10.1007/978-3-319-11558-0_9
7. Della Monica, D., Goranko, V., Montanari, A., Sciavicco, G., et al.: Interval temporal logics: a journey. Bull EATCS **3**(105), 73–99 (2013)
8. Goranko, V., Montanari, A., Sciavicco, G.: A road map of interval temporal logics and duration calculi. J. Appl. Non Class. Logics **14**(1–2), 9–54 (2004)
9. Goranko, V., Otto, M.: Model theory of modal logic. In: Blackburn, P., Wolter, F., van Benthem, J. (eds.) Handbook of Modal Logic, pp. 255–325. Elsevier, Amsterdam (2006)
10. Halpern, J.Y., Shoham, Y.: A propositional modal logic of time intervals. J. ACM (JACM) **38**(4), 935–962 (1991)
11. Kontchakov, R., Pandolfo, L., Pulina, L., Ryzhikov, V., Zakharyaschev, M.: Temporal and spatial OBDA with many-dimensional Halpern-Shoham logic. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016). AAAI Press (2016)
12. Papadimitriou, C.H.: Computational Complexity. Wiley, New York (2003)

Author Index