

**Sergei Artemov  
Anil Nerode (Eds.)**

**LNCS 10703**

# **Logical Foundations of Computer Science**

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

**LF  
CS  
2018**

 **Springer**

Sergei Artemov · Anil Nerode (Eds.)

# Logical Foundations of Computer Science

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

 Springer

*Editors*

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

Anil Nerode  
Cornell University  
Ithaca, NY  
USA

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

Library of Congress Control Number: 2017960856

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

© Springer International Publishing AG 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper


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

# Contents

The Completeness Problem for Modal Logic . . . . .	1
<i>Antonis Achilleos</i>	
Justification Awareness Models. . . . .	22
<i>Sergei Artemov</i>	
A Minimal Computational Theory of a Minimal Computational Universe. . . . .	37
<i>Arnon Avron and Liron Cohen</i>	
A Sequent-Calculus Based Formulation of the Extended First Epsilon Theorem . . . . .	55
<i>Matthias Baaz, Alexander Leitsch, and Anela Lolic</i>	
Angluin Learning via Logic . . . . .	72
<i>Simone Barlocco and Clemens Kupke</i>	
A Universal Algebra for the Variable-Free Fragment of $RC^\nabla$ . . . . .	91
<i>Lev D. Beklemishev</i>	
A Logic of Blockchain Updates . . . . .	107
<i>Kai Brännler, Dandolo Flumini, and Thomas Studer</i>	
From Display to Labelled Proofs for Tense Logics . . . . .	120
<i>Agata Ciabattoni, Tim Lyon, and Revantha Ramanayake</i>	
Notions of Cauchyness and Metastability . . . . .	140
<i>Hannes Diener and Robert Lubarsky</i>	
A Gödel-Artemov-Style Analysis of Constructible Falsity . . . . .	154
<i>Thomas Macaulay Ferguson</i>	
Probabilistic Reasoning About Simply Typed Lambda Terms . . . . .	170
<i>Silvia Ghilezan, Jelena Ivetić, Simona Kašterović, Zoran Ognjanović, and Nenad Savić</i>	
Polyteam Semantics. . . . .	190
<i>Miika Hannula, Juha Kontinen, and Jonni Virtema</i>	
On the Sharpness and the Single-Conclusion Property of Basic Justification Models. . . . .	211
<i>Vladimir N. Krupski</i>	

Founded Semantics and Constraint Semantics of Logic Rules . . . . .	221
<i>Yanhong A. Liu and Scott D. Stoller</i>	
Separating the Fan Theorem and Its Weakenings II . . . . .	242
<i>Robert S. Lubarsky</i>	
Dialectica Categories for the Lambek Calculus . . . . .	256
<i>Valeria de Paiva and Harley Eades III</i>	
From Epistemic Paradox to Doxastic Arithmetic . . . . .	273
<i>V. Alexis Peluce</i>	
A Natural Proof System for Herbrand's Theorem . . . . .	289
<i>Benjamin Ralph</i>	
Metastability and Higher-Order Computability . . . . .	309
<i>Sam Sanders</i>	
The Completeness of <i>BCD</i> for an Operational Semantics . . . . .	331
<i>Rick Statman</i>	
A Tableau System for Instantial Neighborhood Logic . . . . .	337
<i>Junhua Yu</i>	
Interpretations of Presburger Arithmetic in Itself . . . . .	354
<i>Alexander Zapryagaev and Fedor Pakhomov</i>	
<b>Author Index</b> . . . . .	<b>369</b>

# The Completeness Problem for Modal Logic

Antonios Achilleos<sup>(✉)</sup> 

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

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

**Keywords:** Modal logic · Completeness · Computational complexity  
Bisimulation

## 1 Introduction

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

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

---

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

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

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

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

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

Normal forms for Modal Logic were introduced by Fine [12] and they can be used to prove soundness, completeness, and the finite frame property for several modal logics with respect to their classes of frames. Normal forms are modal formulas that completely describe the behavior of a Kripke model up to a certain distance from a state, with respect to a certain number of propositional

variables. Therefore, every complete formula is equivalent to a normal form, but not all normal forms are complete, as they may be agnostic with respect to states located further away. We may define that a formula is complete up to depth  $d$  for logic  $l$  when it is equivalent to a normal form of modal depth (the nesting depth of a formula's modalities) at most  $d$ . We briefly discuss these topics in Sect. 6.

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

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

## 2 Background

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

### 2.1 Modal Logic

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

$$\varphi ::= \perp \mid \top \mid p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Box \varphi \mid \Diamond \varphi.$$



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 2.2 Bisimulation

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

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

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

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

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

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

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

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

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

### 2.3 The Complexity of Satisfiability

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

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

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

## 3 The Completeness Problem and Axiom 5

*The completeness problem for  $l$*  asks, given a formula  $\varphi$ , if  $\varphi$  is complete for  $l$ . In this section, we explain how to adjust Halpern and Rêgo's techniques from [2] to prove similar complexity bounds for the completeness problem for logics with Negative Introspection. In the course of proving the coNP upper bound for logics with Negative Introspection, Halpern and Rêgo give in [2] a construction that provides a small model for a satisfiable formula. We can adjust parts of

their construction and conclude with Corollary 2 and from that, Lemma 1 and Corollary 1. The remaining results in this section are consequence of these.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

*Proof.* By Corollary 2 and Theorem 3.  $\square$

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

## 4 The Completeness Problem and Triviality

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

### 4.1 Completeness and $\mathbf{K}$

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

**Lemma 3.** *Formula  $\bigwedge P \wedge \Box \perp$  is complete and satisfiable for  $\mathbf{K}$  and for  $\mathbf{K4}$ .*

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

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

### 4.2 Completeness and Consistency

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

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

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

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

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

$$\begin{aligned} W' &= \Pi_d \cup \{x\}; \\ R' &= \{(\alpha, \alpha b) \in W'^2 \mid b \in W\} \cup \{(a_0 a_1 \cdots a_d, x) \in W'^2\} \cup \{(x, x)\} \\ V'_i(\alpha b) &= V(b), \text{ for } i = 1, 2, 0 \leq |\alpha| < d; \\ V'_1(x) &= \emptyset; \text{ and } V'_2(x) = P. \end{aligned}$$

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

every  $a_k R' b$ ,  $\mathcal{M}, b \models \psi'$  (by the Inductive Hypothesis) iff  $\mathcal{M}, a_k \models \psi$ ; the case of  $\psi = \Diamond\psi'$  is symmetric.

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

### 4.3 Completeness, Consistency, and Positive Introspection

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

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

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

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

**Corollary 3.**  *$\varphi^{\mathbf{D4}}$  is satisfiable and complete for every extension of **D4**.<sup>2</sup>*

### 4.4 Consistency and Negative Introspection

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

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

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

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

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

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

<sup>3</sup> This is also a corollary of Lemma 4, as these are extensions of **D** and **T**.

## 4.5 Completeness and Modal Logics

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

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

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

## 5 The Complexity of Completeness

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

### 5.1 A Lower Bound

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

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



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

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

- $\mathcal{M}_l, a_l \not\models \varphi$ , in which case  $\varphi$  is not provable and we set  $\varphi_c = \bigwedge P$ .
- $\mathcal{M}_l, a_l \models \varphi$ , so  $\neg\varphi \wedge \varphi_P^l$  is not satisfiable, in which case we set  $\varphi_c = \varphi \rightarrow \varphi_P^l$ .

We demonstrate that  $\varphi$  is provable if and only if  $\varphi \rightarrow \varphi_P^l$  is complete.

- If  $\varphi$  is provable, then  $\varphi \rightarrow \varphi_P^l$  is equivalent to  $\varphi_P^l$ , which is complete.
- On the other hand, if  $\varphi \rightarrow \varphi_P^l$  is complete and  $(\mathcal{M}, a)$  is any pointed model, we show that  $\mathcal{M}, a \models \varphi$ , implying that if  $\varphi \rightarrow \varphi_P^l$  is complete, then  $\varphi$  is provable. If  $(\mathcal{M}, a) \sim_P (\mathcal{M}_l, a_l)$ , then from our assumptions  $\mathcal{M}, a \not\models \neg\varphi$ , thus  $\mathcal{M}, a \models \varphi$ . On the other hand, if  $(\mathcal{M}, a) \not\sim_P (\mathcal{M}_l, a_l)$ , since  $(\mathcal{M}_l, a_l) \models \varphi \rightarrow \varphi_P^l$  and  $\varphi \rightarrow \varphi_P^l$  is complete,  $\mathcal{M}, a \not\models \varphi \rightarrow \varphi_P^l$ , therefore  $\mathcal{M}, a \models \varphi$ .  $\square$

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

## 5.2 Upper Bounds

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

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

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

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

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

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

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

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

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

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

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

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

**Theorem 6.** *The CC Procedure accepts  $\varphi$  if and only if  $\varphi$  is incomplete.*

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

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

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

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

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

We demonstrate that if  $\varphi$  is incomplete, then the CC Procedure will accept  $\varphi$  after a finite number of steps. As we have seen above, the procedure, given non-bisimilar pointed models  $(A, a)$  and  $(B, b)$  of  $\varphi$ , always has a child to play

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

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

*Proof.* See [13]. □

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

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

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

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

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

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

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

*Proof.* See [13]. □

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

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

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

*Proof.* See [13]. □

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

*Proof.* See [13]. □

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

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

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

## 6 Variations and Other Considerations

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

### 6.1 Satisfiable and Complete Formulas

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

### 6.2 Completeness with Respect to a Model

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

---

<sup>4</sup> We note that US is different from UP; for UP, if  $T$  has an accepting path for  $x$ , then it is *guaranteed* that it has a unique accepting path for  $x$ .

### 6.3 Completeness and Normal Forms for Modal Logic

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

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

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

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

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

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

### 6.4 Completeness up to Depth

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

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

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

*Proof.* From Theorem 7, if  $\varphi$  is satisfiable, then it is equivalent to some  $\bigvee S$ , where  $S \subseteq F_P^{md(\varphi)}$ , but if it is also complete up to its depth, then it can derive a

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

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

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

*Proof.* The proof is similar to that of Theorem 5 and can be found in [13].  $\square$

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

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

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

## 6.5 More Logics

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

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



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

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


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

## References

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

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

# Justification Awareness Models

Sergei Artemov<sup>(✉)</sup> 

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

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

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

## 1 Context and Motivations

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

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

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

rather than to propositions “agent is aware/unaware of a proposition  $F$ ”; this approach allows for the maintaining of natural closure properties.

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

## 2 Preliminaries

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

$$\Vdash t:F,$$

and  $G$  has the same truth value as  $F$

$$\Vdash F \leftrightarrow G,$$

we still cannot conclude that  $t$  is a justification for  $G$

$$\not\Vdash t:G.$$

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

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

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

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

To keep it simple, we consider proposition  $Q$

*the late Prime Minister’s last name began with a ‘B,’*

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

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

<sup>1</sup> From [6]: “Hyperintensional contexts are simply contexts which do not respect logical equivalence”.

<sup>2</sup> Which was true in 1912.

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

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

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

*Consider the picture<sup>4</sup>:*

$$\frac{16}{64} = \frac{1}{4}. \quad (1)$$

*The true proposition is “ $16/64 = 1/4$ ,” the right justification is dividing both the numerator and the denominator by 16, and the wrong (but shorter and more attractive) justification is simplifying as in (1).*

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

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

- $Q$  holds;
- **J** $Q$  holds, since the agent has a justification  $w$  for  $Q$ ;
- **K** $Q$  does not hold;

thus yielding the set of assumptions

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

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

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

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

<sup>4</sup> Which the author saw on the door of the Mathematics Support Center at Cornell in 2017.

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

$$\not\vdash P \rightarrow t:P.$$

Again, this holds since  $P \rightarrow t:P$  is not a propositional tautology. For example, put  $t^* = \emptyset$  and  $P^* = 1$ . In this model,  $t$  is not a justification for  $P$  (i.e.,  $\not\vdash_* t:P$ ) and  $P \rightarrow t:P$  is false.

4. A somewhat less trivial example illustrating hyperintensionality: for a justification variable  $x$  and formula  $F$

$$\not\vdash x:F \rightarrow x:(F \wedge F).$$

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

## 4 Basic Justification Logic $J^-$

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

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

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

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

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

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

$$S \triangleright T = \{F \mid G \rightarrow F \in S \text{ and } G \in T \text{ for some } G\}.$$

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

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

$$s^* \triangleright t^* \subseteq [s \cdot t]^*. \quad (2)$$

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

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

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

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

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

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

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

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

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

## 4.1 Other Justification Logics

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

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

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

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

$$\begin{aligned} D &= \neg t:\perp, \\ T &= t:F \rightarrow F, \\ 4 &= t:F \rightarrow !t:t:F, \\ 5 &= \neg t:F \rightarrow ?t:\neg t:F. \end{aligned}$$

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

## 4.2 Sharp Models

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

Indeed, fix a justification constant 0 and consider logic

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

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

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

$$[s\cdot t]^* = s^* \triangleright t^*. \quad (3)$$

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

## 5 Justification Awareness

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

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



## 5.1 Justification Awareness Models

Fix  $J^-(CS)$  for some axiomatically appropriate constant specification  $CS$ .

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

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

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

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

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

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

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

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

## 5.2 Single-Conclusion Justifications

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

*there is at most one formula  $F$  such that  $t:F$  holds.*

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

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

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

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

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

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

## 6 Russell Scenario as a JAM

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

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

Define a model  $*$  such that

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

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

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

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

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

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

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

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

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

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

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

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

## 8 Discussion

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

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

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

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

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

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

## References

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

15. Krupski, V.: On the sharpness and the single-conclusion property of basic justification models. In: Artemov, S., Nerode, A. (eds.) LFCS 2018. LNCS, vol. 10703, pp. 211–220. Springer, Cham (2018)
16. Kuznets, R., Struder, T.: Justifications, ontology, and conservativity. In: Bolander, T., Braüner, T., Ghilardi, S., Moss, L. (eds.) Advances in Modal Logic, vol. 9, pp. 437–458. College Publications, London (2012)
17. Meyer, J.-J.C., van der Hoek, W.: Epistemic Logic for AI and Computer Science. CUP, Cambridge (1995)
18. Mkrtychev, A.: Models for the logic of proofs. In: Adian, S., Nerode, A. (eds.) LFCS 1997. LNCS, vol. 1234, pp. 266–275. Springer, Heidelberg (1997). [https://doi.org/10.1007/3-540-63045-7\\_27](https://doi.org/10.1007/3-540-63045-7_27)
19. Russell, B.: The Problems of Philosophy. Williams and Norgate, London (1912)
20. Sedlár, I.: Justifications, awareness and epistemic dynamics. In: Artemov, S., Nerode, A. (eds.) LFCS 2013. LNCS, vol. 7734, pp. 307–318. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-35722-0\\_22](https://doi.org/10.1007/978-3-642-35722-0_22)
21. Williamson, T.: A note on Gettier cases in epistemic logic. *Philos. Stud.* **172**(1), 129–140 (2015)

# A Minimal Computational Theory of a Minimal Computational Universe

Arnon Avron<sup>1</sup> and Liron Cohen<sup>2</sup>(✉)

<sup>1</sup> Tel Aviv University, Tel-Aviv, Israel  
aa@post.tau.ac.il

<sup>2</sup> Cornell University, Ithaca, NY, USA  
lironcohen@cornell.edu

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

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

## 1 Introduction

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

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

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

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

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

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

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

<sup>3</sup> The thesis that  $J_2$  is sufficient for core mathematics was already put forward in [33].

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

<sup>5</sup> Due to page constraints, all proofs in the paper were omitted, and will appear in an extended version of the current paper.

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

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

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

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

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

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

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

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

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

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

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

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

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



## 2.2 Computational Theories and Universes

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

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

2. A set is called *computational* if it is  $\mathcal{S}_{\mathcal{T}}$  for some computational theory  $\mathcal{T}$ .

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

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

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

### Theorem 11

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

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

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

## 3 Static Extensions by Definitions

When working in a minimal computational universe such as  $J_2$  (as done in the next section), many of the standard mathematical objects (such as the real line

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

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

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

<sup>10</sup> The use of  $n$ -ary predicates can standardly be reduced, of course, to unary predicates.

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

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

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

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

**Proposition 14.** *The following holds:*

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

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

**Proposition 16.** *The following holds:*

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

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

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

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

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