

# Complexity Issues in Justification Logic

by

Roman Kuznets

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

2008

UMI Number: 3310747

Copyright 2008 by  
Kuznets, Roman

All rights reserved

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI<sup>®</sup>

---

UMI Microform 3310747  
Copyright 2008 by ProQuest LLC  
All rights reserved. This microform edition is protected against  
unauthorized copying under Title 17, United States Code.

---

ProQuest LLC  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106-1346

©2008

Roman Kuznets

All Rights Reserved

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirements for the degree of Doctor of Philosophy.

---

Date

---

Sergei Artemov  
Chair of Examining Committee

---

Date

---

Theodore Brown  
Executive Officer

Melvin Fitting

---

Robert Milnikel

---

Rohit Parikh

---

Supervisory Committee

Abstract

## Complexity Issues in Justification Logic

by

Roman Kuznets

Adviser: Professor Sergei Artemov

Justification Logic is an emerging field that studies provability, knowledge, and belief via explicit proofs or justifications that are part of the language. There exist many justification logics closely related to modal epistemic logics of knowledge and belief. Instead of modality  $\Box$  in pure justification logics, or in addition to modality  $\Box$  in hybrid logics, which has an existential epistemic reading ‘there exists a proof of  $F$ ,’ all justification logics use constructs  $t : F$ , where a justification term  $t$  represents a blueprint of a Hilbert-style proof of  $F$ . The first justification logic, LP, introduced by Sergei Artemov, was shown to be a justification counterpart of modal logic S4 and serves as a missing link between S4 and Peano arithmetic, thereby solving a long-standing problem of provability semantics for S4 and Int.

The machinery of explicit justifications can be used to analyze well-known

epistemic paradoxes, e.g. Gettier's examples of justified true belief that can hardly be considered knowledge, and to find new approaches to the concept of common knowledge. Yet another possible application is the Logical Omniscience Problem, which reflects an undesirable property of knowledge as described by modality when an agent knows all the logical consequences of his/her knowledge. The language of justification logic opens new ways to tackle this problem.

This thesis focuses on quantitative analysis of justification logics. We explore their decidability and complexity of Validity Problem for them. A closer analysis of the realization phenomenon in general and of one procedure in particular enables us to deduce interesting corollaries about self-referentiality for several modal logics. A framework for proving decidability of various justification logics is developed by generalizing the Finite Model Property. Limitations of the method are demonstrated through an example of an undecidable justification logic. We study reflected fragments of justification logics and provide them with an axiomatization and a decision procedure whose complexity (the upper bound) turns out to be uniform for all justification logics, both pure and hybrid. For many justification logics, we also present lower and upper complexity bounds.

# Acknowledgments

First and foremost, I would like to express my deepest gratitude to my teacher, Sergei Artemov. I am using here an old-fashioned word “teacher” rather than “research advisor” because there has been much more to learn from him than pure math (or applied math for that matter). That is exactly what a real Teacher does: provides opportunities to learn rather than just lectures, and I have been blessed with many such an opportunity in all spheres of life.

I am thankful to my dissertation committee for their comments and suggestions as well as for many insightful discussions, in which they have showed me how to see further by looking deeper: Melvin Fitting, Robert Milnikel, and Rohit Parikh.

I thank Vladimir Krupski, who supported me throughout the early stages of my scientific carrier at Moscow State University and still generously shares his expertise and advice.

I would also like to mention Vladimir Uspensky, who inspired me to go into Mathematical Logic, and Tatiana Yavorskaya (Sidon), who helped me make the first logical steps.

My mathematical consciousness was shaped by my high-school teacher Vera Petrovna Filinova. It all really started with her relentless devotion to the purity and rigor of mathematics.

I have been fortunate to work alongside colleagues and friends whose suggestions and personalities have had a profound effect on my work and on how I feel about it: Eva Antonakos, Amotz Bar-Noy, Walter Dean, Evan Goris, Eric Pacuit, Bryan Renne, Stathis Zachos.

I also thank Arnon Avron, Balder ten Cate, Paul Égré, Valentin Goranko, Wojtek Jamroga, Natalia Rubtsova for suggestions provided at various conferences.

I thank CUNY Graduate Center and the Research Foundation of CUNY for their financial support.

Needless to say, nothing would have happened in the first place without my family and their ever-lasting support: my musician mother, who has had to put up with her math-inclined son; my father; my grandfather, who have taught me to be strong; my late grandmother, who has taught me to be kind; my uncle, who has helped me to combine these two and still tries to teach

me to restrict the use of binary logic to math only; Stas and Noemi, who have taught me how to be a New Yorker; my little niece, who is an eternal source of joy and happiness for everybody around her. And of course I could not have made it this far without Galina, who has made my life sound and complete and dramatically increased my personal complexity by means of duality, at the same time simplifying all decision procedures.

# Contents

<b>Abstract</b>	<b>iv</b>
<b>Acknowledgments</b>	<b>vi</b>
<b>Table of Contents</b>	<b>viii</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Short Reference Guide</b>	<b>4</b>
2.1 Modal Logic: Language and Hilbert Systems . . . . .	4
2.2 Tableau Systems for Several Modal Logics . . . . .	8
2.3 Gentzen Systems for Several Modal Logics . . . . .	10
2.4 Modal Logic: Kripke Frames and Models . . . . .	10

<i>CONTENTS</i>	x
2.5 Complexity of Various Logics . . . . .	15
2.6 Maximal Consistent Set Construction . . . . .	17
<b>3 Justification Logics Defined</b>	<b>21</b>
3.1 Justification Logic and Forgetful Projection . . . . .	21
3.1.1 Language of Pure Justification Logic . . . . .	21
3.1.2 Justification and Modal Counterparts . . . . .	25
3.2 Axiom Systems for Pure Justification Logics . . . . .	27
3.2.1 Axioms and Rules for Pure Justification Logics . . . . .	27
3.2.2 Constant Specifications . . . . .	29
3.2.3 Common Pure Justification Logics . . . . .	34
3.2.4 Realization Theorems . . . . .	36
3.2.5 Internalization and Other Properties . . . . .	37
3.2.6 Historical Survey . . . . .	44
3.3 Semantics for Pure Justification Logics . . . . .	46
3.3.1 Symbolic M-Models . . . . .	46
3.3.2 Epistemic F-models . . . . .	58
3.3.3 M-Models vs. F-Models . . . . .	85
3.3.4 Minimal Evidence Functions . . . . .	92
3.3.5 Historical Survey . . . . .	114

<i>CONTENTS</i>	xi
3.4 Reflected Fragments of Pure Justification Logics . . . . .	117
3.5 Hybrid Justification Logics . . . . .	126
3.5.1 Axiom Systems for Hybrid Justification Logics . . . . .	127
3.5.2 Semantics for Hybrid Logics . . . . .	133
3.5.3 Minimal Evidence Functions for AF-Models . . . . .	136
3.5.4 Reflected Fragments of Hybrid Logics . . . . .	139
3.5.5 Historical Survey . . . . .	141
<b>4 Decidability</b>	<b>143</b>
4.1 Finite Model Property vs. Finite Frame Property . . . . .	144
4.2 Hidden Assumptions in FMP . . . . .	149
4.3 Finitary Model Property . . . . .	152
4.4 Decidability Results . . . . .	154
4.5 Undecidability Results . . . . .	190
4.6 Historical Survey . . . . .	192
<b>5 Complexity</b>	<b>194</b>
5.1 Upper Bounds for Reflected Fragments . . . . .	194
5.2 Upper Bounds for Pure Justification Logics . . . . .	211
5.3 Lower Bounds for Pure Justification Logics . . . . .	232
5.4 Complexity of Hybrid Logics . . . . .	237

<i>CONTENTS</i>	xii
5.5 Historical Survey . . . . .	266
<b>6 Self-Referentiality</b>	<b>268</b>
6.1 When Is Knowledge Self-Referential? . . . . .	269
6.2 Self-Referential Knowledge . . . . .	270
6.3 Knowledge without Self-Referentiality . . . . .	285
6.4 Conclusions and Future Work . . . . .	291
<b>Bibliography</b>	<b>297</b>
<b>Index</b>	<b>306</b>

# List of Tables

2.1.1 Axiom systems for several monomodal logics . . . . .	7
2.1.2 Axiom systems for several multimodal logics . . . . .	7
2.2.1 Tableau systems for K, D4, T, S4 . . . . .	9
3.2.1 Axiom systems for common justification logics . . . . .	35
3.2.2 Forgetful projections of justification axioms are modal theo- rems. Forgetful projections of justification rules are admissible	36
3.3.1 M-models . . . . .	49
3.3.2 F-models: Conditions on the admissible evidence function . . .	62
3.3.3 F-models: Conditions on $R$ and the Strong Evidence Property	63
3.3.4 *-calculi for pure justification logics . . . . .	104
5.3.1 Propositional translation of axioms of justification logics are propositional tautologies . . . . .	233

# List of Figures

3.3.1 Theorem 3.3.41: Proof that $\mathcal{A} \subseteq \mathcal{E}$ for any $\mathcal{E} \in \mathcal{AEF}_{\mathcal{B}}$ . . . . .	108
3.3.2 Theorem 3.3.41: Proof that $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}$ (main part) . . . . .	109
5.4.1 Recursive procedure $\mathbf{S4LP}_{CS}$ -WORLD . . . . .	244
6.2.1 Tableau derivation of $\diamond(p \rightarrow \Box p)$ in $\mathbf{T}$ and $\mathbf{S4}$ . . . . .	272
6.2.2 Tableau derivation of $\diamond(p \rightarrow \Box p)$ in $\mathbf{D4}$ . . . . .	273

# Chapter 1

## Introduction

Justification Logic is a relatively new field that studies provability, knowledge, and belief via explicit proofs or justifications that are part of the language. There exist many justification logics that closely resemble modal epistemic logics of knowledge and belief, with one important difference: instead of  $\Box\varphi$  with existential epistemic reading ‘there exists a proof of  $\varphi$ ’, justification logics operate with constructs  $t:F$ , where a justification term  $t$  represents a blueprint of a Hilbert-style proof of  $F$ .

The first justification logic, **LP**, was introduced in [Art95] (see also [Art98, Art01, Art04b]). It was shown to be a justification counterpart of modal logic **S4** and serves as a missing link between **S4** and Peano arithmetic, thereby solving a long-standing problem of provability semantics for **S4**, and hence for **Int**. Other justification logics were developed in [AKS99, Bre99, Bre00, Pac05, Rub06b, Art07]. The 2007 paper by Artemov demonstrates

how the machinery of explicit justifications can be used to analyze a well-known epistemic paradox such as Gettier's examples of justified true belief that can hardly be considered knowledge (see [Get63]).

Explicit justification terms can be combined with the traditional epistemic modality providing for a more nuanced structure of knowledge. Such systems were studied in [AN04, Art04a, AN05a, AN05c, AN05b, Rub06c, Art06, Rub06d]. The use of explicit justifications also suggests a new approach to the concept of common knowledge, which was explored in [Ant06a, Art06, Ant06b, Ant07a, Ant07b].

The language of explicit justification allows to study self-referential properties of modal logics through their justification counterparts. These results will be discussed in more detail in Chapter 6 (see also [BK05, Kuz06c, BK06, Kuz08]).

Yet another possible application of the justification logic language is the Logical Omniscience Problem. Logical omniscience is an undesirable property of knowledge as described by modality (see [Hin62, Hin75, Par87, Par95, Par05]). The language of justification logic opens new ways to tackle this problem. Some approaches are described in [Kuz06b, AK06a, AK06b].

Chapter 2 will serve as a collection of definitions and facts about modal logics and complexity that will be used in the following chapters. It also

introduces a notation for modal languages used throughout the thesis.

We will focus on quantitative analysis of justification logics, both pure and combined with various modal logics of knowledge and belief. We will explore their decidability and complexity of their Validity Problems. A closer analysis of the realization phenomenon in general and the specific procedure in particular will enable us to deduce interesting corollaries about self-referentiality in various modal logics.

In Chapter 3, we will describe the pure and hybrid justification logics that will be studied in this thesis.

In Chapter 4, we will develop a framework for proving decidability of various justification logics by generalizing the Finite Model Property. We will also show limitations of the method by presenting an example of a simple justification logic that is undecidable.

In Chapter 5, we will present several results on complexity of justification logics.

Finally, in Chapter 6, we will present results on self-referentiality of several modal logics proven via their justification counterparts.

# Chapter 2

## Short Reference Guide

This chapter is intended mostly as a reference for facts and definitions outside of justification logic that will be used in our research.

### 2.1 Modal Logic: Language and Hilbert Systems

We will consider several modal logics, both mono- and multimodal ones. Hence, we need to introduce notation for the multiple languages we will use.

**Definition 2.1.1.** *Modal formulas* are defined by the following grammar:

$$\varphi ::= p_i \mid \perp \mid (\varphi \rightarrow \varphi) \mid (\Delta\varphi) \tag{2.1.1}$$

where  $p_i$ ,  $i = 0, 1, 2, \dots$  are sentence letters,  $\Delta \in X$  is one of the modalities used in a particular modal language.

Most common examples of modal languages include

- monomodal language  $\mathcal{ML}$  with  $X = \{\Box\}$ ;
- multimodal language  $\mathcal{ML}_n$  with  $X = \{K_1, \dots, K_n\}$ . ◀

*Note 2.1.2.* We will denote the set of all sentence letters  $p_i$  by  $SLet$ .

*Note 2.1.3.* We will consider  $\Diamond$  and  $\Diamond_i$  to be abbreviations of  $\neg\Box\neg$  and  $\neg\Box_i\neg$  respectively.

*Note 2.1.4.* In the epistemic context, modalities  $K$  and  $K_i$  are typically used instead of  $\Box$  and  $\Box_i$  respectively.

*Note 2.1.5.* Language  $\mathcal{ML}_1$  can be identified with  $\mathcal{ML}$  if all occurrences of  $K_1$  are replaced by  $\Box$ .

Some common modal axioms and rules used in monomodal logics follow:

Prop. Finitely many schemes of classical propositional logic in the monomodal

language  $\mathcal{ML}$  along with

$$\text{Modus Ponens Rule} \quad \frac{\varphi \rightarrow \psi \quad \varphi}{\psi}$$

$$\text{K. Normality Axiom} \quad \Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$$

$$\text{T. Reflexivity Axiom} \quad \Box\varphi \rightarrow \varphi$$

$$4. \text{ Modal Positive Introspection} \quad \Box\varphi \rightarrow \Box\Box\varphi$$

$$5. \text{ Modal Negative Introspection} \quad \neg\Box\varphi \rightarrow \Box\neg\Box\varphi$$

D. *Seriality Axiom*

$$\Box \perp \rightarrow \perp$$

Nec. *Modal Necessitation Rule*

$$\frac{\vdash \varphi}{\vdash \Box \varphi}$$

where  $\varphi$  and  $\psi$  are arbitrary monomodal formulas in language  $\mathcal{ML}$ .

Some of these axioms and rules are generalized for the  $n$ -modal logics in the following ways:

Prop. Finitely many schemes of classical propositional logic in the multimodal

language  $\mathcal{ML}_n$  along with

*Modus Ponens Rule*

$$\frac{\varphi \rightarrow \psi \quad \varphi}{\psi}$$

 $K_i$ . *Normality Axiom*

$$K_i(\varphi \rightarrow \psi) \rightarrow (K_i\varphi \rightarrow K_i\psi)$$

 $T_i$ . *Reflexivity Axiom*

$$K_i\varphi \rightarrow \varphi$$

 $4_i$ . *Modal Positive Introspection*

$$K_i\varphi \rightarrow K_iK_i\varphi$$

 $5_i$ . *Modal Negative Introspection*

$$\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$$

Nec $_i$ . *Modal Necessitation Rule*

$$\frac{\vdash \varphi}{\vdash K_i\varphi}$$

where  $i = 1, \dots, n$ ,  $\varphi$  and  $\psi$  are arbitrary multimodal formulas in language  $\mathcal{ML}_n$ .

Table 2.1.1: Axiom systems for several monomodal logics

Logic	Prop	K	T	4	5	D	Nec
K	✓	✓					✓
D	✓	✓				✓	✓
T	✓	✓	✓				✓
K4	✓	✓		✓			✓
D4	✓	✓		✓		✓	✓
S4	✓	✓	✓	✓			✓
K5	✓	✓			✓		✓
K45	✓	✓		✓	✓		✓
KD45	✓	✓		✓	✓	✓	✓
S5	✓	✓	✓	✓	✓		✓

Table 2.1.2: Axiom systems for several multimodal logics

Logic	Prop	$K_i$	$T_i$	$4_i$	$5_i$	$Nec_i$
$K_n$	✓	✓				✓
$T_n$	✓	✓	✓			✓
$S4_n$	✓	✓	✓	✓		✓
$S5_n$	✓	✓	✓	✓	✓	✓

Tables 2.1.1 and 2.1.2 define several mono- and multimodal logics respectively. Further information about these modal logics can be found in [Fey65, FHMV95, CZ97, FM98, BdRV01].

## 2.2 Tableau Systems for Several Modal Logics

This section includes tableau rules for several modal logics. This particular version of tableaux, sometimes called Single Step Tableaux, uses prefixes to denote worlds. Each prefix  $\sigma = i_1i_2 \dots i_k$  is a finite non-empty sequence of integers  $i_j$ . By  $\sigma.n$  we understand sequence  $i_1i_2 \dots i_kn$ .

It is assumed that all propositional tableau rules are present in each modal tableau system. Propositional rules do not change prefixes. Below are the modal rules of various monomodal logics:

□

$$\frac{\sigma \quad \Box\varphi}{\sigma.n \quad \varphi} \qquad \frac{\sigma \quad \neg\Diamond\varphi}{\sigma.n \quad \neg\varphi} \qquad (2.2.1)$$

where  $\sigma.n$  has already occurred on the branch;

◇

$$\frac{\sigma \quad \neg\Box\varphi}{\sigma.n \quad \neg\varphi} \qquad \frac{\sigma \quad \Diamond\varphi}{\sigma.n \quad \varphi} \qquad (2.2.2)$$

where  $\sigma.n$  is a new prefix on the branch.

Table 2.2.1: Tableau systems for K, D4, T, S4

Logic	$\Box$	$\Diamond$	T	4	D
K	✓	✓			
T	✓	✓	✓		
D4	✓	✓		✓	✓
S4	✓	✓	✓	✓	

T

$$\frac{\sigma \Box \varphi}{\sigma \varphi} \qquad \frac{\sigma \neg \Diamond \varphi}{\sigma \neg \varphi} \qquad (2.2.3)$$

D

$$\frac{\sigma \Box \varphi}{\sigma \Diamond \varphi} \qquad \frac{\sigma \neg \Diamond \varphi}{\sigma \neg \Box \varphi} \qquad (2.2.4)$$

4

$$\frac{\sigma \Box \varphi}{\sigma.n \Box \varphi} \qquad \frac{\sigma \neg \Diamond \varphi}{\sigma.n \neg \Diamond \varphi} \qquad (2.2.5)$$

where  $\sigma.n$  has already occurred on the branch.

All these rules are  $\alpha$ -rules;  $\varphi$  is an arbitrary formula in language  $\mathcal{ML}$ ;  $\sigma$  is an arbitrary prefix.

We will only use tableaux for D4, T, and S4. The modal rules that should be used for each of them are listed in Table 2.2.1.

More details and tableaux for other modal logics can be found, for instance, in [Fit72, Mas94, FM98, Mas00, Fit07a].

## 2.3 Gentzen Systems for Several Modal Logics

Here are two modal Gentzen rules:

$$\frac{\varphi_1, \dots, \varphi_n \Rightarrow \psi}{\Box\varphi_1, \dots, \Box\varphi_n \Rightarrow \Box\psi} \quad (2.3.1)$$

$$\frac{\varphi_1, \dots, \varphi_n, \xi \Rightarrow}{\Box\varphi_1, \dots, \Box\varphi_n, \Box\xi \Rightarrow} \quad (2.3.2)$$

where  $\varphi_i$ 's,  $\psi$ , and  $\xi$  are arbitrary monomodal formulas in language  $\mathcal{ML}$ .

The Gentzen system for **K** is obtained by adding (2.3.1) to the propositional Gentzen system. The Gentzen system for **D** is obtained by adding both (2.3.1) and (2.3.2). Both resulting systems are cut-free. Moreover, it is possible to restrict the use of axioms to  $\perp \Rightarrow$  and  $p \Rightarrow p$  for sentence letters  $p$ .

For more information about cut-free Gentzen systems for modal logics see [Wan94, Fit07a].

## 2.4 Modal Logic: Kripke Frames and Models

**Definition 2.4.1.** A binary relation  $R \subseteq W \times W$  is called

- *reflexive* if  $uRu$  for each  $u \in W$ ;
- *transitive* if  $uRv$  and  $vRw$  yield  $uRw$  for any  $u, v, w \in W$ ;

- **serial** if for each  $u \in W$  there is  $v \in W$  such that  $uRv$ ;
- **symmetric** if  $uRv$  yields  $vRu$  for any  $u, v \in W$ ;
- **Euclidean** if  $uRv$  and  $uRw$  yield  $vRw$  for any  $u, v, w \in W$ . ◀

**Lemma 2.4.2.** *A binary relation  $R$  on set  $W$  that is both Euclidean and reflexive must be also symmetric and transitive. Hence, such an  $R$  is an equivalence relation.*

**Definition 2.4.3.** An  *$n$ -modal Kripke frame* for  $\mathcal{ML}_n$  is a  $(n + 1)$ -tuple

$$\mathfrak{F} = (W, R_1, \dots, R_n) ,$$

where  $W \neq \emptyset$  is a set of possible worlds and accessibility relations  $R_i$  are binary relations on  $W$ . ◀

**Definition 2.4.4.** An  *$n$ -modal Kripke model* for  $\mathcal{ML}_n$  is a  $(n + 2)$ -tuple

$$\mathfrak{M} = (W, R_1, \dots, R_n, V) ,$$

where  $(W, R_1, \dots, R_n)$  is a Kripke frame and *propositional valuation*

$$V : W \times SLet \rightarrow \{True, False\}$$

is a map that assigns a truth value to every sentence letter at every world of the model.

Truth relation  $\mathfrak{M}, u \Vdash \xi$  for  $u \in W$  and  $\xi \in \mathcal{ML}_n$  is defined by induction on the size of  $\xi$ :

$$\mathfrak{M}, u \Vdash p \quad \Leftrightarrow \quad u \in V(p) \quad (2.4.1)$$

$$\mathfrak{M}, u \not\Vdash \perp \quad (2.4.2)$$

$$\mathfrak{M}, u \Vdash \varphi \rightarrow \psi \quad \Leftrightarrow \quad \mathfrak{M}, u \not\Vdash \varphi \quad \text{or} \quad \mathfrak{M}, u \Vdash \psi \quad (2.4.3)$$

$$\mathfrak{M}, u \Vdash K_i \varphi \quad \Leftrightarrow \quad \mathfrak{M}, w \Vdash \varphi \quad (\forall w)uR_i w \quad (2.4.4)$$

where  $p$  is a sentence letter,  $u, w \in W$ ,  $\varphi, \psi \in \mathcal{ML}_n$ ,  $i = 1, \dots, n$ .  $\blacktriangleleft$

**Definition 2.4.5.** A *monomodal Kripke frame (model)* is a 1-modal Kripke frame (model). We will usually omit the subscript of  $R_1$  in monomodal frames and models, denoting it simply by  $R$ .  $\blacktriangleleft$

**Definition 2.4.6.** We say that a Kripke model  $(W, R_1, \dots, R_n, V)$  is *based on* the Kripke frame  $(W, R_1, \dots, R_n)$ .  $\blacktriangleleft$

**Definition 2.4.7.** A Kripke frame  $\mathfrak{F} = (W, R_1, \dots, R_n)$  and all Kripke models based on it are called *finite* if  $W$  is a finite set.  $\blacktriangleleft$

**Definition 2.4.8.** Formula  $\varphi \in \mathcal{ML}_n$  is called *valid in a Kripke model*  $\mathfrak{M} = (W, R_1, \dots, R_n, V)$  if  $\mathfrak{M}, w \Vdash \varphi$  for all  $w \in W$ .  $\blacktriangleleft$

**Definition 2.4.9.** Formula  $\varphi \in \mathcal{ML}_n$  is called *satisfiable in a Kripke model*  $\mathfrak{M} = (W, R_1, \dots, R_n, V)$  if  $\mathfrak{M}, w \Vdash \varphi$  for at least one  $w \in W$ .  $\blacktriangleleft$

**Definition 2.4.10.** Formula  $\varphi \in \mathcal{ML}_n$  is called *valid in an  $n$ -modal Kripke frame*  $\mathfrak{F}$  if  $\varphi$  is valid in all Kripke models based on frame  $\mathfrak{F}$ . ◀

**Definition 2.4.11.** Formula  $\varphi \in \mathcal{ML}_n$  is called *satisfiable in an  $n$ -modal Kripke frame*  $\mathfrak{F}$  if  $\varphi$  is satisfiable in at least one Kripke model based on frame  $\mathfrak{F}$ . ◀

**Definition 2.4.12.** Formula  $\varphi \in \mathcal{ML}_n$  is called *valid with respect to a class  $\mathcal{C}_F$  of  $n$ -modal Kripke frames (with respect to a class  $\mathcal{C}_M$  of  $n$ -modal Kripke models)* if  $\varphi$  is valid in all frames  $\mathfrak{F} \in \mathcal{C}_F$  (in all models  $\mathfrak{M} \in \mathcal{C}_M$ ). ◀

**Definition 2.4.13.** Formula  $\varphi \in \mathcal{ML}_n$  is called *satisfiable with respect to a class  $\mathcal{C}_F$  of  $n$ -modal Kripke frames (with respect to a class  $\mathcal{C}_M$  of  $n$ -modal Kripke models)* if  $\varphi$  is satisfiable in at least one frame  $\mathfrak{F} \in \mathcal{C}_F$  (in at least one model  $\mathfrak{M} \in \mathcal{C}_M$ ). ◀

**Definition 2.4.14.** Formula  $\varphi \in \mathcal{ML}_n$  is called *refutable with respect to a class  $\mathcal{C}_F$  of  $n$ -modal Kripke frames (with respect to a class  $\mathcal{C}_M$  of  $n$ -modal Kripke models)* if  $\neg\varphi$  is satisfiable in at least one frame  $\mathfrak{F} \in \mathcal{C}_F$  (in at least one model  $\mathfrak{M} \in \mathcal{C}_M$ ). ◀

Completeness results for several monomodal logics are listed below:

**Theorem 2.4.15 (Completeness Theorem for monomodal logics).**

- $K$  is sound and complete w.r.t. the class of all monomodal Kripke frames (models).
- $D$  is sound and complete w.r.t. the class of all monomodal Kripke frames (models) with serial  $R$ .
- $T$  is sound and complete w.r.t. the class of all monomodal Kripke frames (models) with reflexive  $R$ .
- $K4$  is sound and complete w.r.t. the class of all monomodal Kripke frames (models) with transitive  $R$ .
- $D4$  is sound and complete w.r.t. the class of all monomodal Kripke frames (models) with transitive and serial  $R$ .
- $S4$  is sound and complete w.r.t. the class of all monomodal Kripke frames (models) with transitive and reflexive  $R$ .
- $K5$  is sound and complete w.r.t. the class of all monomodal Kripke frames (models) with Euclidean  $R$ .
- $K45$  is sound and complete w.r.t. the class of all monomodal Kripke frames (models) with transitive and Euclidean  $R$ .

- $\text{KD45}$  is sound and complete w.r.t. the class of all monomodal Kripke frames (models) with serial, transitive, and Euclidean  $R$ .
- $\text{S5}$  is sound and complete w.r.t. the class of all monomodal Kripke frames (models) with  $R$  that is an equivalence relation.
- $\text{K}_n$  is sound and complete w.r.t. the class of all  $n$ -modal Kripke frames (models).
- $\text{T}_n$  is sound and complete w.r.t. the class of all  $n$ -modal Kripke frames (models) with reflexive  $R_i$ ,  $i = 1, \dots, n$ .
- $\text{S4}_n$  is sound and complete w.r.t. the class of all  $n$ -modal Kripke frames (models) with transitive and reflexive  $R_i$ ,  $i = 1, \dots, n$ .
- $\text{S5}_n$  is sound and complete w.r.t. the class of all  $n$ -modal Kripke frames (models) with  $R_i$  that are equivalence relations for  $i = 1, \dots, n$ .

Further information on Kripke models and frames for various modal logics can be found in [Fey65, FHMV95, CZ97, FM98, BdRV01].

## 2.5 Complexity of Various Logics

**Definition 2.5.1.** By *complexity of a logic*  $L$  we will mean complexity of the *validity problem* for  $L$ , i.e., complexity of determining, given a

formula  $F$  in the language of  $L$ , whether  $L \vdash F$ . ◀

**Definition 2.5.2.** The *satisfiability problem* for a logic  $L$  is the problem of determining, given a formula  $F$  in the language of  $L$ , whether  $L \not\vdash \neg F$ .

◀

**Theorem 2.5.3** ([Coo71]). *The satisfiability problem for classical propositional logic  $Cl$ , also known as SAT, is NP-complete. Accordingly,  $Cl$  is co-NP-complete.*

**Theorem 2.5.4** ([Sta79]). *The intuitionistic propositional logic  $Int$  is PSPACE-complete.*

**Theorem 2.5.5** ([Lad77]).

- $\top$  is PSPACE-complete.
- $S4$  is PSPACE-complete.
- $S5$  is co-NP-complete.

**Theorem 2.5.6** ([HM85, HM92]).

- $\top_n$  is PSPACE-complete for  $n \geq 1$ .
- $S4_n$  is PSPACE-complete for  $n \geq 1$ .
- $S5_n$  is PSPACE-complete for  $n \geq 2$ .

## 2.6 Maximal Consistent Set Construction

This section includes definitions and statements used for constructing maximal consistent sets. Throughout the section,  $\mathbf{L}$  is assumed to be a consistent logic, understood as a set of formulas in a (countable) language  $\mathcal{L}$ , with classical Boolean logic in the background. In particular, we assume that all Boolean connectives and constants are expressible in  $\mathcal{L}$ . All formulas are assumed to be in language  $\mathcal{L}$ .

**Definition 2.6.1.** A set  $\Gamma$  of  $\mathcal{L}$ -formulas is called ***L-consistent*** if

$$\neg(F_1 \wedge \dots \wedge F_n) \notin \mathbf{L}$$

for any finite subset  $\{F_1, \dots, F_n\} \subseteq \Gamma$ .

A set  $\Gamma$  is called ***maximal L-consistent*** if it is  $\mathbf{L}$ -consistent whereas no superset  $\Delta \supsetneq \Gamma$  is. ◀

The following lemma lists several useful properties of maximal consistent sets:

**Lemma 2.6.2.** *Let  $\Gamma$  be an arbitrary maximal  $\mathbf{L}$ -consistent set.*

1. *No  $\mathbf{L}$ -consistent set contains  $\perp$ .*
2. *If a set  $\Delta$  is  $\mathbf{L}$ -consistent, so are all subsets of  $\Delta$ .*

3. For each formula  $F$ , set  $\Gamma$  contains exactly one of formulas  $F$  and  $\neg F$ .
4. Set  $\Gamma$  is closed under modus ponens, i.e., for any formulas  $F$  and  $G$ , if  $F \rightarrow G \in \Gamma$  and  $F \in \Gamma$ , then  $G \in \Gamma$ .
5. Set  $\Gamma$  is closed under conjunctions, i.e., for any formulas  $F$  and  $G$ , if  $F \in \Gamma$  and  $G \in \Gamma$ , then  $F \wedge G \in \Gamma$ .
6.  $\mathsf{L} \subseteq \Gamma$ .
7. If  $F \notin \mathsf{L}$ , then the set  $\{\neg F\}$  is  $\mathsf{L}$ -consistent.
8. For each  $\mathsf{L}$ -consistent set  $\Delta$ , there exists a maximal  $\mathsf{L}$ -consistent set  $\Delta'$  such that  $\Delta' \supseteq \Delta$ .
9. If  $\mathsf{L}$  is supplied with a proof system that allows derivations from hypotheses and  $\mathsf{L}$  enjoys the Deduction Theorem, a set  $\Delta$  is  $\mathsf{L}$ -consistent iff

$$F_1, \dots, F_n \not\vdash_{\mathsf{L}} \perp$$

for any finite subset  $\{F_1, \dots, F_n\} \subseteq \Delta$ .

Let us restrict the notion of maximal consistency to formulas from a given set  $X$ . We will need this relativized version for decidability proofs.

**Definition 2.6.3.** A set  $\Gamma$  of  $\mathcal{L}$ -formulas is called *maximal L-consistent relative to a set X* if

- $\Gamma \subseteq X$ ,
- $\Gamma$  is L-consistent,
- $\Gamma \cup \{G\}$  is not L-consistent for any  $G \in X \setminus \Gamma$ . ◀

The following is a relativized version of Lemma 2.6.2:

**Lemma 2.6.4.** *Let X be a set of formulas. Let set  $\Gamma$  be maximal L-consistent relative to X.*

1. *For each formula  $F$ , set  $\Gamma$  contains at most one of formulas  $F$  and  $\neg F$ .  
Moreover, if  $\{F, \neg F\} \subseteq X$ , set  $\Gamma$  contains exactly one of them.*
2. *If  $\mathcal{L}$  is supplied with a proof system that allows derivations from hypotheses and  $\Gamma \vdash_{\mathcal{L}} F$  for some  $F \in X$ , then  $F \in \Gamma$ .*
3. *Set  $\Gamma$  is closed under modus ponens, i.e., for any formulas  $F$  and  $G$ , if  $F \rightarrow G \in \Gamma$ ,  $F \in \Gamma$ , then  $\neg G \notin \Gamma$ . Moreover, if  $G \in X$ , then  $G \in \Gamma$ .*
4. *Set  $\Gamma$  is closed under conjunctions, i.e., for any formulas  $F$  and  $G$ , if  $F \in \Gamma$  and  $G \in \Gamma$ , then  $\neg(F \wedge G) \notin \Gamma$ . Moreover, if  $F \wedge G \in X$ , then  $F \wedge G \in \Gamma$ .*

5.  $\mathbb{L} \cap X \subseteq \Gamma$ .

6. For each  $\mathbb{L}$ -consistent set  $\Delta \subseteq X$ , there exists a set  $\Delta'$  that is maximal  $\mathbb{L}$ -consistent relative to  $X$  such that  $X \supseteq \Delta' \supseteq \Delta$ .

# Chapter 3

## Justification Logics Defined

In this chapter, we will define major justification logics, both pure and hybrid, describe their semantics, and outline the relationships between pure justification and modal logics. At the end of the chapter, we will also provide a short historical survey of the development of justification logics.

### 3.1 Justification Logic and Forgetful Projection

First, we will describe the language of pure justification logic and give a precise meaning of the term “justification counterpart of a modal logic.”

#### 3.1.1 Language of Pure Justification Logic

We will start by defining the language of Justification Logic. It has two types of objects: formulas that we will mostly denote by  $F, G, \dots$  and *justification*

terms, denoted  $t, s, \dots$ , which are sometimes called *evidence terms*, *proof terms*, or *proof polynomials*.

**Definition 3.1.1.** *Justification terms* are built from *justification constants*  $c_i, i = 0, 1, 2, \dots$  and *justification variables*  $x_i, i = 0, 1, 2, \dots$  by means of several operations according to the following grammar:

$$t ::= c_i \mid x_i \mid (t \cdot t) \mid (t + t) \mid (!t) \mid (?t) \quad (3.1.1)$$

The binary operations *application*  $\cdot$  and *sum*  $+$ , the latter also called *union* or *choice*, and the unary operation *proof checker*  $!$  are present in all justification logics, whereas the unary operation *negative introspection*  $?$  may or may not be allowed depending on the desired modal counterpart.

We will, therefore, distinguish between

- basic language  $\mathcal{JL}$  of justification logic with  $+$ ,  $\cdot$ , and  $!$  only and
- language  $\mathcal{JL}(?)$ , obtained by adding the unary operation  $?$  to  $\mathcal{JL}$ .

We will denote the set of all justification terms in either language by  $Tm$ .



*Note 3.1.2.* As usual, whenever possible, we will omit parentheses according to the following order of operations: unary operations bind more strongly

than binary ones,  $\cdot$  binds more strongly than  $+$ . Thus,

$$!t \cdot s + ?r$$

should be read as

$$((!t) \cdot s) + (?r) .$$

**Definition 3.1.3.** *Justification formulas* in language  $\mathcal{JL}$  or  $\mathcal{JL}(?)$  are defined by the following grammar

$$F ::= p_i \mid \perp \mid (F \rightarrow F) \mid (t:F) \tag{3.1.2}$$

where  $p_i$ ,  $i = 0, 1, 2, \dots$ , are sentence letters and  $t$  is a justification term in language  $\mathcal{JL}$  or  $\mathcal{JL}(?)$  respectively.

The new construct  $t:F$  is read ‘term  $t$  serves as a justification (evidence, proof) of formula  $F$ .’

We will denote the set of all justification formulas in either language by  $Fm$ . ◀

**Definition 3.1.4.** The size of justification formulas and terms is defined by

$$|c_i| = 1$$

$$|x_i| = 1$$

$$|!t| = |t| + 1$$

$$|?t| = |t| + 1$$

$$|t \cdot s| = |t| + |s| + 1$$

$$|t + s| = |t| + |s| + 1$$

$$|p_i| = 1$$

$$|\perp| = 1$$

$$|F \rightarrow G| = |F| + |G| + 1$$

$$|t:F| = |t| + |F| + 1$$

where  $c_i$  is a justification constant,  $x_i$  is a justification variable,  $t$  and  $s$  are justification terms,  $p_i$  is a sentence letter,  $F$  and  $G$  are justification formulas.



*Note 3.1.5.* The remaining Boolean connectives  $\vee$ ,  $\wedge$ ,  $\neg$ ,  $\leftrightarrow$  and the Boolean constant  $\top$  are defined through  $\rightarrow$  and  $\perp$  in the standard way.

*Note 3.1.6.* Again we will omit parentheses using the standard operation order on Boolean connectives. The new construct ‘:’ binds more strongly

than any Boolean connective. Thus,

$$!t:t:F \rightarrow G$$

should be read as

$$(!t:(t:F)) \rightarrow G .$$

*Note 3.1.7.* We will denote justification formulas by Latin letters  $F, G, \dots$  whereas modal formulas will be denoted by Greek letters  $\varphi, \psi, \dots$ . This will allow to distinguish between the two easily. Of course, such a distinction will not be possible while considering hybrid languages with both justification terms and traditional modalities. We will continue to denote such hybrid formulas by Latin letters.

### 3.1.2 Justification and Modal Counterparts

**Definition 3.1.8.** The *forgetful projection* is a function  $^\circ: \mathcal{JL}(?) \rightarrow \mathcal{ML}$  that converts justification formulas into monomodal formulas. It is defined by induction on the size of the justification formula:

- $p^\circ = p$  ,
- $\perp^\circ = \perp$  ,
- $(F \rightarrow G)^\circ = F^\circ \rightarrow G^\circ$  ,

- $(t:F)^\circ = \Box(F^\circ)$  ,

where  $p$  is a sentence letter,  $F$  and  $G$  are justification formulas,  $t$  is a justification term.

The *forgetful projection of a set*  $X$  of justification formulas is the set of modal formulas  $X^\circ = \{F^\circ \mid F \in X\}$ . ◀

A logic can be identified with the set of its theorems. In this sense,

**Definition 3.1.9.** A monomodal logic  $ML$  is said to be the *forgetful projection of a justification logic*  $JL$  if  $JL^\circ = ML$ . In this case, we also say that  $JL$  is a *justification counterpart* of  $ML$ . ◀

*Note 3.1.10.* One monomodal logic may have several justification counterparts. A few examples will follow later.

To prove that a modal logic  $ML$  is the forgetful projection of a justification logic  $JL$ , two inclusions must be demonstrated:

1.  $JL^\circ \subseteq ML$ , i.e., the forgetful projection of every  $JL$ -theorem is derivable in  $ML$ .
2.  $ML \subseteq JL^\circ$ , i.e., it is possible to *realize* each occurrence of  $\Box$  in every  $ML$ -theorem in such a way that the resulting justification formula is a theorem of  $JL$ .

The first statement is typically more or less trivial and is proven by induction on the JL-derivation. The main difficulty is presented by the second statement. That is why the two statements combined are usually called a *Realization Theorem* (just like soundness and completeness combined are usually branded Completeness Theorem). Realization Theorems have been proven for many pairs of modal-justification counterparts using a variety of methods.

## 3.2 Axiom Systems for Pure Justification Logics

### 3.2.1 Axioms and Rules for Pure Justification Logics

Various justification logics are obtained by combining the following axioms and rules:

A1. Finitely many schemes of classical propositional logic in language  $\mathcal{JL}$

(or  $\mathcal{JL}(?)$ )

*Modus Ponens Rule* 
$$\frac{F \rightarrow G \quad F}{G}$$

A2. *Application Axiom* 
$$s:(F \rightarrow G) \rightarrow (t:F \rightarrow s \cdot t:G)$$

A3. *Monotonicity Axiom* 
$$s:F \rightarrow s + t:F$$

$$t:F \rightarrow s + t:F$$

A4. *Factivity Axiom*  $t:F \rightarrow F$

A5. *Positive Introspection*  $t:F \rightarrow !t:t:F$

A6. *Negative Introspection*  $\neg t:F \rightarrow ?t:\neg t:F$

A7. *Consistency Axiom*  $t:\perp \rightarrow \perp$

R4. *Axiom Internalization Rule*  $\frac{}{c:A}$

R4<sup>!</sup>. *Axiom Internalization Rule*

with positive introspection

$$\frac{\underbrace{!!\dots!}_n c:\dots:!!c:!!c:c:A}{c:A}$$

where  $F$  and  $G$  are justification formulas in language  $\mathcal{JL}$  (or  $\mathcal{JL}(?)$  respectively),  $t$  and  $s$  are justification terms in language  $\mathcal{JL}$  (or  $\mathcal{JL}(?)$  respectively),  $A$  is an axiom of the logic,  $c$  is a justification constant, and  $n \geq 0$  is an integer.

*Note 3.2.1.* Depending on the justification logic, all formulas and terms in these axioms and rules are taken either from language  $\mathcal{JL}$  or from  $\mathcal{JL}(?)$ . Naturally, axiom A6 can only be used for logics in language  $\mathcal{JL}(?)$ .

*Note 3.2.2.* For each justification logic, axiom  $A$  in rules R4 and R4<sup>!</sup> stands for an arbitrary axiom of *this* logic, not for an arbitrary axiom from A1–A7.

*Note 3.2.3.* Rule R4<sup>!</sup> is admissible in presence of axiom A5 and rule R4. Thus, rules R4 and R4<sup>!</sup> should be used interchangeably: R4 in conjunction

with A5, and  $R4^!$  in the absence of A5.

*Note 3.2.4.* Similarly, axiom A7 is an instance of axiom A4; hence only one of them should be used for each particular logic.

### 3.2.2 Constant Specifications

Both rules  $R4$  and  $R4^!$  postulate that each constant justifies all axioms of the logic. But there are situations when it is desirable to supervise or restrict the use of constants, e.g., to reserve a particular constant for a particular scheme of axioms or for a particular axiom instance. Such restrictions were used in [Mil07] for establishing lower complexity bounds, in [Kuz05] for demonstrating potential undecidability, and for such applications as Logical Omniscience Problem (see [Kuz06b, AK06a, AK06b]) and self-referentiality in modal logic (see [BK05, Kuz06c, BK06, Kuz08]). For this purpose, rules  $R4$  and  $R4^!$  can be restricted to a particular *constant specification*.

**Definition 3.2.5.** A *constant specification for a justification logic*  $JL$  is any set of formulas

$$CS \subseteq \{c:A \mid c \text{ is a justification constant, } A \text{ is an axiom of } JL\} .$$



*Note 3.2.6.* A constant specification for a justification logic  $JL_1$  is not always

a suitable constant specification for another justification logic  $\mathbf{JL}_2$  because some axioms of  $\mathbf{JL}_1$  may not be axioms of  $\mathbf{JL}_2$ .

**Proposition 3.2.7.** *Let  $\mathcal{CS}$  be a constant specification for a justification logic  $\mathbf{JL}_1$ . If all axioms of  $\mathbf{JL}_1$  are also axioms of another justification logic  $\mathbf{JL}_2$ , then  $\mathcal{CS}$  can also be used as a constant specification for  $\mathbf{JL}_2$ .*

*Note 3.2.8.* The ability to transfer a constant specification  $\mathcal{CS}$  to a stronger justification logic implicitly depends on the system of propositional axioms chosen in A1. Although any complete propositional axiom system can be used in A1, which is why it is almost never specified, this axiomatization should better remain intact if we are to transfer  $\mathcal{CS}$  from one justification logic to another. It would not suffice that an axiom of the weaker logic be derivable in the stronger one as Def. 3.2.5 requires that formula  $A$  in  $c:A$  be an axiom of the logic, not just a theorem.

**Definition 3.2.9.** Let  $\mathcal{CS}$  be a constant specification for a justification logic  $\mathbf{JL}$ . The *justification logic*  $\mathbf{JL}_{\mathcal{CS}}$  is obtained by replacing rule R4 (or rule R4<sup>!</sup>) in  $\mathbf{JL}$  by rule R4 <sub>$\mathcal{CS}$</sub>  (or rule R4<sup>!</sup> <sub>$\mathcal{CS}$</sub>  respectively):

R4 <sub>$\mathcal{CS}$</sub> . *Axiom Internalization Rule* restricted to  $\mathcal{CS}$  
$$\frac{c:A \in \mathcal{CS}}{c:A}$$

R4<sup>!</sup> <sub>$\mathcal{CS}$</sub> . *Axiom Internalization Rule* with positive introspection

restricted to  $\mathcal{CS}$

$$\frac{c:A \in \mathcal{CS}}{\underbrace{!! \dots !!}_{n} c: \dots : !! c: ! c: c: A}$$

where  $n \geq 0$  is an integer. ◀

*Note 3.2.10.* By Note 3.2.3, only one of  $R4$  and  $R4^!$  is present in any justification logic. Therefore, only one of  $R4_{\mathcal{CS}}$  and  $R4^!_{\mathcal{CS}}$  is present in its  $\mathcal{CS}$ -restriction.

**Definition 3.2.11.** Let  $JL$  be any justification logic. The *justification logic*  $JL_0$  is the logic  $JL_\emptyset$  with the empty constant specification  $\mathcal{CS} = \emptyset$ , or equivalently, the logic  $JL$  with neither  $R4$  nor  $R4^!$ . ◀

**Definition 3.2.12.** Let  $JL$  be a justification logic. The *total constant specification for*  $JL$  is the largest constant specification

$$\mathcal{TCS}_{JL} = \{c:A \mid c \text{ is a justification constant, } A \text{ is an axiom of } JL\} .$$

Thus,  $JL_{\mathcal{TCS}_{JL}} = JL$ . We will omit the subscript  $JL$  in  $\mathcal{TCS}_{JL}$  whenever it is clear from the context. ◀

*Note 3.2.13.* Many theorems are formulated for  $JL_{\mathcal{CS}}$  with an arbitrary  $\mathcal{CS}$ . According to Def. 3.2.12, such theorems also apply to  $JL$  itself.

It is sometimes convenient to view  $\mathcal{CS}$  as a function from constants to sets of axioms justified by them:

**Definition 3.2.14.** Let  $\mathcal{CS}$  be a constant specification for a justification logic. For each justification constant  $c$ ,

$$\mathcal{CS}(c) = \{A \mid c:A \in \mathcal{CS}\} . \quad (3.2.1)$$



For each logic  $\mathbf{JL}$ , between its smallest constant specification  $\emptyset$  and its largest constant specification  $\mathcal{TCS}_{\mathbf{JL}}$  there is a multitude of possibilities. Some types of constant specifications present special interest and were studied in more detail. Among them are the following types:

**Definition 3.2.15.** A constant specification  $\mathcal{CS}$  for a justification logic  $\mathbf{JL}$  is called

- *axiomatically appropriate* if

$$\bigcup_{i=0}^{\infty} \mathcal{CS}(c_i)$$

contains all axioms of  $\mathbf{JL}$ , i.e., if each axiom is justified by at least one constant;

- *injective* if for each constant  $c$  the set  $\mathcal{CS}(c)$  contains at most one axiom, i.e., if every constant proves at most one axiom;

- **schematic** if for each constant  $c$  the set  $\mathcal{CS}(c)$  consists of one or several (possibly zero) axiom schemes, i.e., every constant proves certain axiom schemes;
- **schematically injective** if it is schematic and for each constant  $c$  the set  $\mathcal{CS}(c)$  consists of at most one axiom scheme, i.e., every constant proves at most one scheme;
- **finite** if  $\mathcal{CS}$  is a finite set;
- **almost schematic** if  $\mathcal{CS}$  is a disjoint union of a schematic  $\mathcal{CS}_1$  and a finite  $\mathcal{CS}_2$ . ◀

*Note 3.2.16.* The name might deceptively suggest that a schematically injective constant specification is simply the one that is both schematic and injective. However, a schematically injective  $\mathcal{CS}$  must indeed be schematic, but it is only injective when it is empty.

*Note 3.2.17.* The total constant specification is schematic and axiomatically appropriate, but is not schematically injective.

*Note 3.2.18.* Some notes on terminology:

1. The definitions of “constant specification” in [Art98, Art01, Art04b] and of “axiom specification” in [Art95] corresponded to what we call

here a “finite constant specification.” The definition used here was perhaps first presented in [Mkr97].

2. “Total constant specifications” were called “maximal” in the earlier papers. The term “total” was probably first used in [Art07] although its idea goes back to [Mkr97, Kuz00].
3. The term “schematic” was first introduced in [Mil07] although its idea again goes back to [Mkr97, Kuz00].
4. The term “schematically injective” is due to Robert Milnikel ([Mil07]).
5. The term “axiomatically appropriate” is due to Melvin Fitting ([Fit05]).
6. The term “almost schematic” is new.

### 3.2.3 Common Pure Justification Logics

We will now define several pure justification logics by listing which axioms and rules from Section 3.2.1 should be used for each of them.

**Definition 3.2.19.** Justification logics  $J$ ,  $JD$ ,  $JT$ ,  $J4$ ,  $JD4$ , and  $LP$  in language  $\mathcal{JL}$  and justification logics  $J5$ ,  $J45$ ,  $JD45$ , and  $JT45$  in language  $\mathcal{JL}(?)$  are defined by the axioms and rules specified in Table 3.2.1. ◀

Table 3.2.1: Axiom systems for common justification logics

Logic	A1	A2	A3	A4	A5	A6	A7	R4	R4'
J	✓	✓	✓						✓
JD	✓	✓	✓				✓		✓
JT	✓	✓	✓	✓					✓
J4	✓	✓	✓		✓			✓	
JD4	✓	✓	✓		✓		✓	✓	
LP	✓	✓	✓	✓	✓			✓	
J5	✓	✓	✓			✓			✓
J45	✓	✓	✓		✓	✓		✓	
JD45	✓	✓	✓		✓	✓	✓	✓	
JT45	✓	✓	✓	✓	✓	✓		✓	

It is apparent from Table 3.2.1 that J is the minimal justification logic. Hence, all the names, except for LP, start with prefix J. The name LP, chronologically the first justification logic, was kept to avoid confusion as it has been used in virtually all the papers on the subject. The logic LP, the original Logic of Proofs, could also be named JT4 in the new uniform notation.

To understand the naming conventions for these justification logics, it would help to compare Table 3.2.1 with Table 2.1.1. The similarity of the names of modal logics in Table 2.1.1 and the names of justification logics in Table 3.2.1 should be immediate. In the next section, we will explain that this is not a mere coincidence. For now, it suffices to note that the name of a justification logic with axiom A4, A5, A6, and/or A7 typically contains

Table 3.2.2: Forgetful projections of justification axioms are modal theorems. Forgetful projections of justification rules are admissible

Justification Axioms		Their forgetful projections	
A1	propositional axioms	propositional axioms	
MP	$F, F \rightarrow G \vdash G$	$F^\circ, F^\circ \rightarrow G^\circ \vdash G^\circ$	MP
A2	$s:(F \rightarrow G) \rightarrow (t:F \rightarrow s \cdot t:F)$	$\Box(F^\circ \rightarrow G^\circ) \rightarrow (\Box F^\circ \rightarrow \Box G^\circ)$	K
A3	$s:F \rightarrow s + t:F$	$\Box F^\circ \rightarrow \Box F^\circ$	taut
A3	$t:F \rightarrow s + t:F$	$\Box F^\circ \rightarrow \Box F^\circ$	taut
A4	$t:F \rightarrow F$	$\Box F^\circ \rightarrow F^\circ$	T
A5	$t:F \rightarrow !t:t:F$	$\Box F^\circ \rightarrow \Box \Box F^\circ$	4
A6	$\neg t:F \rightarrow ?t:\neg t:F$	$\neg \Box F^\circ \rightarrow \Box \neg \Box F^\circ$	5
A7	$t:\perp \rightarrow \perp$	$\Box \perp \rightarrow \perp$	D
R4	$c:A$	$\Box A^\circ$	
R4 <sup>!</sup>	$\underbrace{!\dots!c:\dots:c:A}_n$	$\underbrace{\Box \dots \Box A^\circ}_{n+1}$	

the symbol denoting the modal axiom that is the forgetful projection of this justification axiom (see Table 3.2.2), e.g., all logics with axiom A7 have letter ‘D’ in their name.

### 3.2.4 Realization Theorems

**Theorem 3.2.20 (Realization Theorem, [Art95, Bre99, Rub06b, Art07]).**

*The following correspondences hold:*

1.  $J^\circ = K$
2.  $JD^\circ = D$
3.  $JT^\circ = T$
4.  $J4^\circ = K4$
5.  $JD4^\circ = D4$
6.  $LP^\circ = S4$
7.  $J45^\circ = K45$
8.  $JT45^\circ = S5$

In addition, most probably,  $J5^\circ = K5$  and  $JD45^\circ = KD45$ .

**Theorem 3.2.21.** *Justification logics  $J_{CS}$ ,  $JD_{CS}$ ,  $JT_{CS}$ ,  $J4_{CS}$ ,  $JD4_{CS}$ ,  $LP_{CS}$ ,  $J5_{CS}$ ,  $J45_{CS}$ ,  $JD45_{CS}$ , and  $JT45_{CS}$  are consistent for any  $CS$ .*

*Proof.* Let  $JL_{CS}$  be one of these justification logics. First of all, it is sufficient to prove that  $JL$  is consistent since  $JL_{CS} \subseteq JL$  for any constant specification  $CS$  suitable for  $JL$ . If  $JL \vdash \perp$ , then by the Realization Theorem 3.2.20 for  $JL$ , we would have  $JL^\circ \vdash \perp^\circ = \perp$ , which would contradict the well-established consistency of the modal logics in the right side of the equations in Theorem 3.2.20.

This argument can potentially leave the question open for logics  $J5_{CS}$  and  $JD45_{CS}$ . However, it is sufficient to note that the forgetful projections of all justification axioms are derivable and the forgetful projections of all instances of justification rules are admissible in **S5** (see Table 3.2.2), which is well known to be consistent.  $\square$

### 3.2.5 Internalization and Other Properties

A crucial role in the proof of the Realization Theorems is played by the following fundamental property of justification logics:

**Lemma 3.2.22 (Internalization Property, [Art01, Art07]).** *For any justification logic*

$$JL_{CS} \in \{J_{CS}, JD_{CS}, JT_{CS}, J4_{CS}, JD4_{CS}, LP_{CS}, J5_{CS}, J45_{CS}, JD45_{CS}, JT45_{CS}\} ,$$

where  $\mathcal{CS}$  is an axiomatically appropriate constant specification for  $\mathbf{JL}$ , if

$$F_1, \dots, F_n, \quad \vdash_{\mathbf{JL}_{\mathcal{CS}}} \quad B \quad ,$$

then there exists a term  $t(x_1, \dots, x_n)$  for some fresh justification variables  $x_i$ ,  $i = 1, \dots, n$ , such that

$$x_1:F_1, \dots, x_n:F_n \quad \vdash_{\mathbf{JL}_{\mathcal{CS}}} \quad t(x_1, \dots, x_n):B \quad .$$

*Note 3.2.23.* The requirement for  $\mathcal{CS}$  to be axiomatically appropriate cannot be dropped. Since axioms are derivable, Internalization demands that for each axiom, there be a justification term, which can only be a justification constant. The requirement that  $\mathcal{CS}$  be axiomatically appropriate guarantees the existence of at least one such constant for each axiom.

*Proof of Lemma 3.2.22.* For the logics listed in the Lifting Lemma 3.2.25 below, the Internalization Property is an instance of the Lifting Lemma; the proof can be found there.

Thus, we only need to supply a proof for the remaining justification logics:  $\mathbf{J}$ ,  $\mathbf{JD}$ ,  $\mathbf{JT}$ , and  $\mathbf{J5}$ . The procedure below shows, by induction on the given derivation, how to prefix every formula in this derivation with an extra justification term:

$$\begin{array}{lll}
A & \rightsquigarrow & c:A & \text{by } R4_{\mathcal{CS}}^! \text{ where } A \text{ is an axiom.} \\
& & & \text{Such } c \text{ exists because } \mathcal{CS} \text{ is} \\
& & & \text{axiomatically appropriate} \\
F_i & \rightsquigarrow & x_i:F_i & \text{hypotheses} \\
\frac{D \rightarrow G \quad D}{G} & \rightsquigarrow & \frac{s_1:(D \rightarrow G) \quad s_2:D}{(s_1 \cdot s_2):G} & \text{by A2 and } \textit{modus ponens} \text{ twice} \\
\underbrace{!\dots!}_{k} c:\dots:c:A & \rightsquigarrow & \underbrace{!!\dots!}_{k+1} c:\dots:c:A & \text{where } c:A \in \mathcal{CS} \text{ by } R4_{\mathcal{CS}}^! \\
& & & k \geq 0 \text{ is an integer}
\end{array}$$

□

If  $n = 0$ , the resulting statement is called *constructive necessitation*, which essentially is a justification counterpart of the modal Necessitation Rule:

**Corollary 3.2.24 (Constructive Necessitation).** *For any justification logic*

$$\mathbb{JL}_{\mathcal{CS}} \in \{\mathbb{JL}_{\mathcal{CS}}, \mathbb{JD}_{\mathcal{CS}}, \mathbb{JT}_{\mathcal{CS}}, \mathbb{J4}_{\mathcal{CS}}, \mathbb{JD4}_{\mathcal{CS}}, \mathbb{LP}_{\mathcal{CS}}, \mathbb{J5}_{\mathcal{CS}}, \mathbb{J45}_{\mathcal{CS}}, \mathbb{JD45}_{\mathcal{CS}}, \mathbb{JT45}_{\mathcal{CS}}\} ,$$

where  $\mathcal{CS}$  is an axiomatically appropriate constant specification for  $\mathbb{JL}$ , if

$$\mathbb{JL}_{\mathcal{CS}} \vdash B ,$$

then there exists a ground term  $t$  such that

$$\mathbb{JL}_{\mathcal{CS}} \vdash t:B .$$

For logics with positive introspection, an even stronger result can be formulated:

**Lemma 3.2.25 (Lifting Lemma, [Art01, Art07]).** *For any justification logic with positive introspection*

$$\mathbf{JL}_{\mathcal{CS}} \in \{\mathbf{J4}_{\mathcal{CS}}, \mathbf{JD4}_{\mathcal{CS}}, \mathbf{LP}_{\mathcal{CS}}, \mathbf{J45}_{\mathcal{CS}}, \mathbf{JD45}_{\mathcal{CS}}, \mathbf{JT45}_{\mathcal{CS}}\} ,$$

where  $\mathcal{CS}$  is an axiomatically appropriate constant specification for  $\mathbf{JL}$ , if

$$F_1, \dots, F_n, \quad q_1 : G_1, \dots, q_k : G_k \quad \vdash_{\mathbf{JL}_{\mathcal{CS}}} \quad B$$

for some justification terms  $q_1, \dots, q_k$ , then there exists a term

$$t(x_1, \dots, x_n, y_1, \dots, y_k)$$

for some fresh variables  $x_i, i = 1, \dots, n$ , and  $y_j, j = 1, \dots, k$  such that

$$x_1 : F_1, \dots, x_n : F_n, \quad q_1 : G_1, \dots, q_k : G_k \quad \vdash_{\mathbf{JL}_{\mathcal{CS}}} \quad t(x_1, \dots, x_n, q_1, \dots, q_k) : B .$$

*Note 3.2.26.* Lifting Lemma is often formulated with  $q_j$  restricted to justification variables. A more general version formulated here comes at no additional price. It will be used for proving seriality in finitary canonical models in Lemma 4.4.21.

*Proof of Lemma 3.2.25.* The procedure below shows, by induction on the given derivation, how to prefix every formula in this derivation by an extra justification term.

$A$	$\rightsquigarrow$	$c : A$	by $R4_{\mathcal{CS}}$ where $A$ is an axiom. Such $c$ exists because $\mathcal{CS}$ is axiomatically appropriate
$F_i$	$\rightsquigarrow$	$x_i : F_i$	hypotheses
$q_j : G_j$	$\rightsquigarrow$	$!q_j : q_j : G_j$	by A5 and <i>modus ponens</i> from hypotheses $q_j : G_j$
$\frac{D \rightarrow G \quad D}{G}$	$\rightsquigarrow$	$\frac{s_1 : (D \rightarrow G) \quad s_2 : D}{(s_1 \cdot s_2) : G}$	by A2 and <i>modus ponens</i> twice
$c : A$	$\rightsquigarrow$	$!c : c : A$	where $c : A \in \mathcal{CS}$ by A5, $R4_{\mathcal{CS}}$ , and <i>modus ponens</i>

□

Justification logics also enjoy the Deduction Theorem:

**Lemma 3.2.27 (Deduction Theorem, [Art01, Art07]).** *For any justification logic*

$$\mathbb{JL}_{\mathcal{CS}} \in \{ \mathbb{JL}_{\mathcal{CS}}, \mathbb{JD}_{\mathcal{CS}}, \mathbb{JT}_{\mathcal{CS}}, \mathbb{J4}_{\mathcal{CS}}, \mathbb{JD4}_{\mathcal{CS}}, \mathbb{LP}_{\mathcal{CS}}, \mathbb{J5}_{\mathcal{CS}}, \mathbb{J45}_{\mathcal{CS}}, \mathbb{JD45}_{\mathcal{CS}}, \mathbb{JT45}_{\mathcal{CS}} \} ,$$

where  $\mathcal{CS}$  is a constant specification for  $\mathbb{JL}$ , if

$$\Gamma, F \vdash_{\mathbb{JL}_{\mathcal{CS}}} G ,$$

then

$$\Gamma \vdash_{\mathbb{JL}_{\mathcal{CS}}} F \rightarrow G .$$

The following Substitution Property requires certain flexibility from the constant specification. In fact, there are two slightly different formulations.

**Lemma 3.2.28 (Substitution Property, [Art01, Art07]).** *For any justification logic*

$$\mathbf{JL}_{\mathcal{CS}} \in \{ \mathbf{J}_{\mathcal{CS}}, \mathbf{JD}_{\mathcal{CS}}, \mathbf{JT}_{\mathcal{CS}}, \mathbf{J4}_{\mathcal{CS}}, \mathbf{JD4}_{\mathcal{CS}}, \mathbf{LP}_{\mathcal{CS}}, \mathbf{J5}_{\mathcal{CS}}, \mathbf{J45}_{\mathcal{CS}}, \mathbf{JD45}_{\mathcal{CS}}, \mathbf{JT45}_{\mathcal{CS}} \} ,$$

where  $\mathcal{CS}$  is a schematic constant specification for  $\mathbf{JL}$ , if

$$\Gamma \vdash_{\mathbf{JL}_{\mathcal{CS}}} F ,$$

then

$$\Gamma[s \setminus x, G \setminus p] \vdash_{\mathbf{JL}_{\mathcal{CS}}} F[s \setminus x, G \setminus p] ,$$

where  $[s \setminus x, G \setminus p]$  means substituting justification term  $s$  for justification variable  $x$  and/or formula  $G$  for sentence letter  $p$ .

*Note 3.2.29.* The requirement for  $\mathcal{CS}$  to be schematic cannot be dropped completely. Consider  $c: A(p) \in \mathcal{CS}$ . It is derivable in  $\mathbf{JL}_{\mathcal{CS}}$ . The Substitution Property states, in particular, that no matter what formula  $G$  we substitute for  $p$  in  $c: A(p)$ , the result  $c: A(G)$  should still be derivable in  $\mathbf{JL}_{\mathcal{CS}}$ . Therefore, constant  $c$  must justify all substitution instances of  $A$ , i.e.,  $\mathcal{CS}$  has to be schematic.

Still the substitution that we will often use does not need the exact formula  $F[s \setminus x, G \setminus p]$  derivable after the substitution. Instead, for  $F = t: H$  we

will sometimes simply need a  $t'$  to exist such that  $t' : H[s \setminus x, G \setminus p]$  is derivable; it will not matter whether this  $t'$  is an exact substitution instance of  $t$  or not. In this case, an axiomatically appropriate  $\mathcal{CS}$  can be used instead of a schematic one:

**Lemma 3.2.30 (Substitution Property with renaming of constants, [Fit05]).** *For any justification logic*

$$\mathbf{JL}_{\mathcal{CS}} \in \{ \mathbf{J}_{\mathcal{CS}}, \mathbf{JD}_{\mathcal{CS}}, \mathbf{JT}_{\mathcal{CS}}, \mathbf{J4}_{\mathcal{CS}}, \mathbf{JD4}_{\mathcal{CS}}, \mathbf{LP}_{\mathcal{CS}}, \mathbf{J5}_{\mathcal{CS}}, \mathbf{J45}_{\mathcal{CS}}, \mathbf{JD45}_{\mathcal{CS}}, \mathbf{JT45}_{\mathcal{CS}} \} ,$$

where  $\mathcal{CS}$  is an axiomatically appropriate constant specification for  $\mathbf{JL}$ , if

$$\Gamma \vdash_{\mathbf{JL}_{\mathcal{CS}}} F ,$$

then

$$\Gamma[s \setminus x, G \setminus p] \vdash_{\mathbf{JL}_{\mathcal{CS}}} \overline{F}[s \setminus x, G \setminus p] ,$$

where  $[s \setminus x, G \setminus p]$  means substituting justification term  $s$  for justification variable  $x$  and/or formula  $G$  for sentence letter  $p$ , and formula  $\overline{F}$  is obtained from formula  $F$  by (possibly) replacing some justification constants with other constants.

*Note 3.2.31.* Here, once again, the requirement for  $\mathcal{CS}$  to be axiomatically appropriate cannot be dropped. Consider  $c : A(p) \in \mathcal{CS}$ . It is derivable in  $\mathbf{JL}_{\mathcal{CS}}$ . When  $p$  is replaced by a formula  $G$ , the resulting  $c : A(G)$  may not

be in  $\mathcal{CS}$ , so in this case we need another constant  $c'$  such that  $c':A(G) \in \mathcal{CS}$ . Axiomatic appropriateness of  $\mathcal{CS}$  guarantees this because  $A(G)$  is still an axiom. So we simply replace  $c$  with  $c'$  as needed.

### 3.2.6 Historical Survey

In the earlier papers, pure justification logics have also been called “operational modal logics,” “explicit modal logics,” “explicit counterparts of modal logics,” “logics of knowledge with justifications.”

The first justification logic, Logic of Proofs LP, was introduced by Sergei Artemov in [Art95] (see also [Art98, Art01, Art04b]), where its forgetful projection was shown to be S4.

Artemov *et al.* in [AKS99] introduced justification logic LPS5 and showed it to be a justification counterpart of S5. This logic was slightly different from JT45 later adopted for this role in [Pac05, Rub06b, Art07]. Instead of axiom A6, logic LPS5 had axiom scheme

$$t:(F \rightarrow \neg s:G) \rightarrow (F \rightarrow ?t:\neg s:G) ,$$

which is, in some sense, a guarded variant of A6. This enabled us to develop an arithmetic semantics for LPS5 by avoiding a situation when one term proves infinitely many formulas.

Justification counterparts J, JD, JT, J4, and JD4<sup>1</sup> for modal logics K, D, T, K4, and D4 respectively were developed and the Realization Theorem for them was proven by Vladimir Brezhnev in [Bre00].

Eric Pacuit in [Pac05] suggested axiom systems J5, JD45, and JT45,<sup>2</sup> the latter independently formulated by Natalia Rubtsova in [Rub06a]. Rubtsova in [Rub06b] proved the Realization Theorem for JT45, i.e., that JT45 is a justification counterpart of S5.

The logic J45 was first formulated by Artemov in [Art07]. The proof of the Realization Theorem for it is very similar to the case of JT45 and was omitted there. Most probably, the same method can be easily applied to prove that  $J5^\circ = K5$  and  $JD45^\circ = KD45$ .

Strictly speaking, formulations of justification logics without axiom A5 in [Pac05, Art07], e.g., Pacuit's J5, are slightly different from those given in Table 3.2.1. Terms  $!\dots!c$  in rule R4<sup>1</sup> are replaced there by justification constants. This minor change seems to have a profound effect on decidability and complexity results, which is the reason we went back to Brezhnev's original formulation.

It should be mentioned that apart from the realization techniques devel-

---

<sup>1</sup>Under the names LP(K), LP(D), LP(T), LP(K4), and LP(D4) respectively.

<sup>2</sup>Under the names LP(K5), LP(KD45), and LP(S5) respectively.

oped by Artemov, there is a different technique for proving realization due to Melvin Fitting (see [Fit03a, Fit05, Fit06b, Fit07c, Fit07b]), but we will not use it in this thesis.

All the logics discussed in this thesis use the multi-conclusion framework when one justification term is allowed to (and justification constants often have to) justify many, sometimes infinitely many formulas. Single-conclusion justification terms have been studied in [Kru97, Kru01, Kru06d, Kru06c], but they remain outside the scope of our research.

Similarly, outside the scope of this research are various justification logics with quantifiers (see [Yav01b, Fit04a, Fit06a]).

## 3.3 Semantics for Pure Justification Logics

### 3.3.1 Symbolic M-Models

**Definition 3.3.1.** An *M-model for a justification logic*

$$\mathbb{JL}_{\mathcal{CS}} \in \{\mathbb{J}_{\mathcal{CS}}, \mathbb{JD}_{\mathcal{CS}}, \mathbb{JT}_{\mathcal{CS}}, \mathbb{J4}_{\mathcal{CS}}, \mathbb{JD4}_{\mathcal{CS}}, \mathbb{LP}_{\mathcal{CS}}\}$$

in language  $\mathcal{JL}$ , where  $\mathcal{CS}$  is a constant specification for  $\mathbb{JL}$ , is a pair

$$\mathfrak{M} = (V, \mathcal{A}) ,$$

where *propositional valuation*

$$V : SLet \rightarrow \{True, False\} \tag{3.3.1}$$

assigns a truth value to each sentence letter and

$$\mathcal{A}: Tm \times Fm \rightarrow \{True, False\} \quad (3.3.2)$$

is an *admissible evidence function*. Informally,  $\mathcal{A}(t, F)$  specifies whether term  $t$  is considered admissible evidence for formula  $F$ . We will use  $\mathcal{A}(t, F)$  as an abbreviation of  $\mathcal{A}(t, F) = True$  and also  $\neg\mathcal{A}(t, F)$  as an abbreviation of  $\mathcal{A}(t, F) = False$ .

The admissible evidence function  $\mathcal{A}$  must satisfy several closure conditions that depend on the axioms and rules of  $JL_{CS}$ :

- *Application Closure*: if  $\mathcal{A}(s, F \rightarrow G)$  and  $\mathcal{A}(t, F)$ , then  $\mathcal{A}(s \cdot t, G)$ ;
- *Sum Closure*: if  $\mathcal{A}(s, F)$ , then  $\mathcal{A}(s + t, F)$ ;  
if  $\mathcal{A}(t, F)$ , then  $\mathcal{A}(s + t, F)$ ;
- *CS Closure*: if  $c:A \in CS$ , then  $\mathcal{A}(c, A)$  and

$$\mathcal{A}(\underbrace{! \dots !}_n, \underbrace{! \dots !}_{n-1} c : \dots : ! c : c : A) \text{ for } n \geq 1;$$

- *Positive introspection Closure* (only if A5 is an axiom of JL):  
if  $\mathcal{A}(t, F)$ , then  $\mathcal{A}(!t, t:F)$ ;
- *Consistent Evidence condition* (only if A7 is an axiom of JL):

$$\mathcal{A}(t, \perp) = False$$

for any formulas  $F$  and  $G$ , any terms  $t$  and  $s$ , any  $c : A \in \mathcal{CS}$ , and any integer  $n \geq 1$ .

The *truth relation*  $\mathfrak{M} \models H$  is defined as follows:

$$\mathfrak{M} \models p \quad \Leftrightarrow \quad V(p) = \text{True} \quad (3.3.3)$$

$$\mathfrak{M} \not\models \perp \quad (3.3.4)$$

$$\mathfrak{M} \models F \rightarrow G \quad \Leftrightarrow \quad \mathfrak{M} \not\models F \text{ or } \mathfrak{M} \models G \quad (3.3.5)$$

$$\mathfrak{M} \models t:F \quad \Leftrightarrow \quad \mathfrak{M} \models F \text{ and } \mathcal{A}(t, F) \text{ (if A4 is an axiom of JL)} \quad (3.3.6)$$

$$\mathfrak{M} \models t:F \quad \Leftrightarrow \quad \mathcal{A}(t, F) \text{ (if A4 is not an axiom of JL)} \quad (3.3.7)$$

for any formulas  $F$  and  $G$ , any term  $t$ , and any sentence letter  $p$ .  $\blacktriangleleft$

*Note 3.3.2.* So far no M-models have been developed for logics with Negative Introspection axiom A6, hence the absence of a negative introspection Closure similar to the Positive Introspection Closure from the list of closure conditions above.

The following trivial proposition simplifies verification of the  $\mathcal{CS}$  Closure condition for the justification logics with positive introspection (axiom A5):

**Proposition 3.3.3.** *Let  $\mathcal{A} : \mathcal{T}m \times \mathcal{F}m \rightarrow \{\text{True}, \text{False}\}$  satisfy both the Positive Introspection Closure condition and the following*

- Simplified  $\mathcal{CS}$  Closure: *if  $c : A \in \mathcal{CS}$ , then  $\mathcal{A}(c, A)$ .*

Table 3.3.1: M-models

Logic	Appl. Clos.	Sum Clos.	$\mathcal{CS}$ Clos.	Simp. $\mathcal{CS}$ Closure	Pos. Intr. Closure	Cons. Ev. Cond.	Def. (3.3.6)	Def. (3.3.7)
$J_{\mathcal{CS}}$	✓	✓	✓					✓
$JD_{\mathcal{CS}}$	✓	✓	✓			✓		✓
$JT_{\mathcal{CS}}$	✓	✓	✓				✓	
$J4_{\mathcal{CS}}$	✓	✓		✓	✓			✓
$JD4_{\mathcal{CS}}$	✓	✓		✓	✓	✓		✓
$LP_{\mathcal{CS}}$	✓	✓		✓	✓		✓	

Then,  $\mathcal{A}$  also satisfies the full  $\mathcal{CS}$  Closure condition.

Table 3.3.1 (cf. Table 3.2.1) summarizes which closure conditions and which definition of truth for formulas  $t : F$  should be used for various justification logics. In this table, using Prop. 3.3.3, the  $\mathcal{CS}$  Closure condition is replaced by its simplified version whenever possible.

**Theorem 3.3.4 (Completeness Theorem for M-models, [Mkr97, Kuz00]).**

Each justification logic

$$JL_{\mathcal{CS}} \in \{J_{\mathcal{CS}}, JD_{\mathcal{CS}}, JT_{\mathcal{CS}}, J4_{\mathcal{CS}}, JD4_{\mathcal{CS}}, LP_{\mathcal{CS}}\} ,$$

where  $\mathcal{CS}$  is a constant specification for  $JL$ , is sound and complete w.r.t. its M-models.

*Proof.* We will first prove *soundness* by induction on the derivation in  $JL_{\mathcal{CS}}$ .

Consider an arbitrary M-model  $\mathfrak{M} = (V, \mathcal{A})$  for  $JL_{\mathcal{CS}}$ :

A1. All propositional axioms are valid and the *modus ponens* rule is admissible in  $\mathfrak{M}$  since the propositional cases (3.3.4)–(3.3.5) for  $\Vdash$  in Def. 3.3.1 are classical.

A2. *Application Axiom*  $s:(F \rightarrow G) \rightarrow (t:F \rightarrow s \cdot t:G)$

Let  $\mathfrak{M} \Vdash s:(F \rightarrow G)$  and  $\mathfrak{M} \Vdash t:F$ . To show validity of A2, we need to show that  $\mathfrak{M} \Vdash s \cdot t:G$ .

Independent of whether (3.3.6) or (3.3.7) is used, both  $\mathcal{A}(s, F \rightarrow G)$  and  $\mathcal{A}(t, F)$  hold. Hence, by the Application Closure condition, we have  $\mathcal{A}(s \cdot t, G)$ . In the case of (3.3.7), this alone is sufficient to conclude that  $\mathfrak{M} \Vdash s \cdot t:G$ .

In the case of (3.3.6), we also know that  $\mathfrak{M} \Vdash F \rightarrow G$  and  $\mathfrak{M} \Vdash F$ . Hence, by (3.3.5),  $\mathfrak{M} \Vdash G$ . Combined with  $\mathcal{A}(s \cdot t, G)$ , this yields  $\mathfrak{M} \Vdash s \cdot t:G$ .

A3. *Monotonicity Axiom*  $s:F \rightarrow s + t:F$

$$t:F \rightarrow s + t:F$$

W.l.o.g. we will show validity of the first formula. Let  $\mathfrak{M} \Vdash s:F$ . To show validity of A3, we need to show that  $\mathfrak{M} \Vdash s + t:F$ .

Firstly,  $\mathcal{A}(s, F)$  holds. Hence, by the Sum Closure condition,  $\mathcal{A}(s+t, F)$

holds. In the case of (3.3.7), this is sufficient to conclude that  $\mathfrak{M} \Vdash s + t : F$ .

In the case of (3.3.6), we additionally know that  $\mathfrak{M} \Vdash F$ . Combined with  $\mathcal{A}(s + t, F)$ , this yields  $\mathfrak{M} \Vdash s + t : F$ .

A4. *Factivity Axiom*  $t : F \rightarrow F$

Let  $\mathfrak{M} \Vdash t : F$ . To show validity of A4, we need to show that  $\mathfrak{M} \Vdash F$ .

In both factive logics  $\text{JT}_{CS}$  and  $\text{LP}_{CS}$ , (3.3.6) is used. Therefore,  $\mathfrak{M} \Vdash t : F$  implies  $\mathfrak{M} \Vdash F$ .

A5. *Positive Introspection*  $t : F \rightarrow !t : t : F$

Let  $\mathfrak{M} \Vdash t : F$ . To show validity of A5, we will show that  $\mathfrak{M} \Vdash !t : t : F$ .

Firstly,  $\mathcal{A}(t, F)$  holds. M-models for the logics  $\text{J4}_{CS}$ ,  $\text{JD4}_{CS}$ , and  $\text{LP}_{CS}$  with positive introspection must satisfy the Positive Introspection Closure condition. Hence,  $\mathcal{A}(!t, t : F)$  holds. In the case of (3.3.7), this is sufficient to conclude that  $\mathfrak{M} \Vdash !t : t : F$ .

In the case of (3.3.6), we combine  $\mathcal{A}(!t, t : F)$  with the assumption that  $\mathfrak{M} \Vdash t : F$ . Together, they yield  $\mathfrak{M} \Vdash !t : t : F$ .

A7. *Consistency Axiom*  $t : \perp \rightarrow \perp$

To show validity of A7, we need to show that  $\mathfrak{M} \not\Vdash t : \perp$  for any term  $t$ .

M-models for both logics  $\text{JD}_{\mathcal{CS}}$  and  $\text{JD4}_{\mathcal{CS}}$  with Consistency Axiom must satisfy the Consistent Evidence Condition  $(\forall t) \neg \mathcal{A}(t, \perp)$ . According to (3.3.7) used in either case,  $\mathfrak{M} \not\vdash t: \perp$ .

$\text{R4}_{\mathcal{CS}}$ . *Axiom Internalization Rule* restricted to  $\mathcal{CS}$  
$$\frac{c: A \in \mathcal{CS}}{c: A}$$

To show the admissibility of  $\text{R4}_{\mathcal{CS}}$ , we need to show that  $\mathfrak{M} \Vdash c: A$  for each  $c: A \in \mathcal{CS}$ .

By the  $\mathcal{CS}$  Closure condition,  $\mathcal{A}(c, A)$  must hold, which is sufficient to conclude  $\mathfrak{M} \Vdash c: A$  in the case of (3.3.7).

We have already shown that  $\mathfrak{M} \Vdash A$  for any axiom  $A$  of logic  $\text{JL}$ . Combined with  $\mathcal{A}(c, A)$ , this yields  $\mathfrak{M} \Vdash c: A$  in the case of (3.3.6).

$\text{R4}_{\mathcal{CS}}^!$ . *Axiom Internalization Rule* with positive introspection

restricted to  $\mathcal{CS}$  
$$\frac{c: A \in \mathcal{CS}}{\underbrace{!! \dots !}_n c: \dots: !! c: ! c: c: A}$$

To show the admissibility of  $\text{R4}_{\mathcal{CS}}^!$ , we need to show that

$$\mathfrak{M} \Vdash \underbrace{!! \dots !}_n c: \dots: !! c: ! c: c: A \quad (3.3.8)$$

for each  $c: A \in \mathcal{CS}$  and each integer  $n \geq 0$ .

By the  $\mathcal{CS}$  Closure condition,  $\mathcal{A}(c, A)$  for  $n = 0$  and

$$\mathcal{A}(\underbrace{!! \dots !}_n c, \underbrace{! \dots !}_{n-1} c: \dots: !! c: ! c: c: A) \quad (3.3.9)$$

for  $n \geq 1$  hold.

In the case of (3.3.7), this alone is sufficient to conclude (3.3.8) for any  $n \geq 0$ .

In the case of (3.3.6), we will use induction on  $n$ .

*Base.*  $n = 0$ . This case coincides with rule  $R4_{CS}$  and has already been proven.

*Step.* Assume for  $n = k$  that

$$\mathfrak{M} \Vdash \underbrace{!!! \dots !}_k c : \dots : !!! c : ! c : c : A . \quad (3.3.10)$$

Then

$$\mathfrak{M} \Vdash \underbrace{!!! \dots !}_{k+1} c : \underbrace{!!! \dots !}_k c : \dots : !!! c : ! c : c : A$$

follows from (3.3.9) for  $n = k + 1$  and the IH (3.3.10).

This completes the proof of soundness.

The *completeness* is shown by the standard maximal consistency argument.

**Lemma 3.3.5.** *Let a justification logic*

$$\mathbb{J}L_{CS} \in \{J_{CS}, JD_{CS}, JT_{CS}, J4_{CS}, JD4_{CS}, LP_{CS}\} ,$$

where  $\mathcal{CS}$  is a constant specification for  $\mathbf{JL}$ . For each maximal  $\mathbf{JL}_{\mathcal{CS}}$ -consistent set  $\Gamma$ , there exists an  $M$ -model  $\mathfrak{M}_\Gamma$  such that

$$\mathfrak{M}_\Gamma \Vdash F \quad \iff \quad F \in \Gamma .$$

*Proof.* This model  $\mathfrak{M}_\Gamma = (V_\Gamma, \mathcal{A}_\Gamma)$ , sometimes called the *canonical  $M$ -model* for  $\Gamma$ , is defined as follows:

$$V_\Gamma(p) = \text{True} \quad \iff \quad p \in \Gamma \quad (3.3.11)$$

$$\mathcal{A}_\Gamma(t, F) = \text{True} \quad \iff \quad t : F \in \Gamma \quad (3.3.12)$$

for any sentence letter  $p$ , any term  $t$ , and any formula  $F$ .

To show that  $\mathcal{A}_\Gamma$  is indeed an admissible for  $\mathbf{JL}_{\mathcal{CS}}$  evidence function we need to verify the closure conditions for each logic:

- *Application Closure:* if  $\mathcal{A}_\Gamma(s, F \rightarrow G)$  and  $\mathcal{A}_\Gamma(t, F)$ , then  $\mathcal{A}_\Gamma(s \cdot t, G)$ .

By (3.3.12),  $\mathcal{A}_\Gamma(s, F \rightarrow G)$  and  $\mathcal{A}_\Gamma(t, F)$  mean that

$$\{s : (F \rightarrow G), \quad t : F\} \subset \Gamma .$$

Axiom A2 states that  $\mathbf{JL}_{\mathcal{CS}} \vdash s : (F \rightarrow G) \rightarrow (t : F \rightarrow s \cdot t : G)$ , so by

Lemma 2.6.2.6,

$$s : (F \rightarrow G) \rightarrow (t : F \rightarrow s \cdot t : G) \in \Gamma .$$

Closing by *modus ponens* twice by Lemma 2.6.2.4, we get  $s \cdot t : G \in \Gamma$

and, by (3.3.12),  $\mathcal{A}_\Gamma(s \cdot t, G)$ .

- *Sum Closure*: if  $\mathcal{A}_\Gamma(s, F)$ , then  $\mathcal{A}_\Gamma(s + t, F)$ ;  
if  $\mathcal{A}_\Gamma(t, F)$ , then  $\mathcal{A}_\Gamma(s + t, F)$ .

Again w.l.o.g. we will only prove the first statement. By (3.3.12),  $\mathcal{A}_\Gamma(s, F)$  means that  $s : F \in \Gamma$ . According to axiom A3, we have  $\mathbb{J}\mathcal{L}_{\mathcal{CS}} \vdash s : F \rightarrow s + t : F$ ; thus, by Lemma 2.6.2.6,

$$s : F \rightarrow s + t : F \in \Gamma .$$

Closing by *modus ponens* by Lemma 2.6.2.4, we get  $s + t : F \in \Gamma$  and, by (3.3.12),  $\mathcal{A}_\Gamma(s + t, F)$ .

- *CS Closure*: if  $c : A \in \mathcal{CS}$ , then  $\mathcal{A}_\Gamma(c, A)$  and for each integer  $n \geq 1$

$$\mathcal{A}_\Gamma(\underbrace{!! \dots !}_n c, \underbrace{! \dots !}_{n-1} c : \dots : ! c : c : A) .$$

For each  $c : A \in \mathcal{CS}$  and each  $n \geq 0$ ,

$$\mathbb{J}\mathcal{L}_{\mathcal{CS}} \vdash \underbrace{!! \dots !}_n c : \dots : ! c : c : A$$

- by rule  $R4_{\mathcal{CS}}^!$ , for logics  $\mathbb{J}\mathcal{CS}$ ,  $\mathbb{J}\mathcal{D}_{\mathcal{CS}}$ , and  $\mathbb{J}\mathcal{T}_{\mathcal{CS}}$  or
- by rule  $R4_{\mathcal{CS}}$ , axiom A5, and *modus ponens*, for logics  $\mathbb{J}4_{\mathcal{CS}}$ ,  $\mathbb{J}\mathcal{D}4_{\mathcal{CS}}$ , and  $\mathbb{L}\mathcal{P}_{\mathcal{CS}}$ .

By Lemma 2.6.2.6,

$$\underbrace{!! \dots !}_n c : \dots : ! c : c : A \in \Gamma ,$$

so by (3.3.12),  $\mathcal{A}_\Gamma(c, A)$  and, for any  $n \geq 1$ ,

$$\mathcal{A}_\Gamma(\underbrace{!!\dots!}_n c, \underbrace{!\dots!}_{n-1} c : \dots : !c : c : A) .$$

- *Positive Introspection Closure* (for  $\mathbf{J4}_{CS}$ ,  $\mathbf{JD4}_{CS}$ , and  $\mathbf{LP}_{CS}$ ):

if  $\mathcal{A}_\Gamma(t, F)$ , then  $\mathcal{A}_\Gamma(!t, t : F)$ .

By (3.3.12),  $\mathcal{A}_\Gamma(t, F)$  means that  $t : F \in \Gamma$ . All the three logics listed above have axiom A5, so  $\mathbf{JL}_{CS} \vdash t : F \rightarrow !t : t : F$ . By Lemma 2.6.2.6,

$$t : F \rightarrow !t : t : F \in \Gamma .$$

Closing by *modus ponens* by Lemma 2.6.2.4, we get  $!t : t : F \in \Gamma$  and,

by (3.3.12),  $\mathcal{A}_\Gamma(!t, t : F)$ .

- *Consistent Evidence condition* (for  $\mathbf{JD}_{CS}$  and  $\mathbf{JD4}_{CS}$ ):

$$\mathcal{A}_\Gamma(t, \perp) = \text{False} \quad \text{for all terms } t.$$

Both logics listed above have axiom A7, so  $\mathbf{JL}_{CS} \vdash \neg t : \perp$  for each term  $t$ .

By Lemma 2.6.2.6,  $\neg t : \perp \in \Gamma$ . By Lemma 2.6.2.3,  $t : \perp \notin \Gamma$ . Therefore,

by (3.3.12),  $\mathcal{A}_\Gamma(t, \perp) = \text{False}$ .

Thus,  $\mathcal{A}_\Gamma$  is indeed an admissible evidence function.

We will now show that

$$\mathfrak{M}_\Gamma \Vdash F \quad \iff \quad F \in \Gamma$$

by induction on complexity of formula  $F$ :

$F = p$ . For a sentence letter  $p$ , the statement follows directly from (3.3.11)

and (3.3.3):

$$\mathfrak{M}_\Gamma \Vdash p \iff V_\Gamma(p) = \text{True} \iff p \in \Gamma .$$

Boolean cases are trivial.

$F = t:G$ . Let  $t:G \in \Gamma$ . Then, by (3.3.12),  $\mathcal{A}_\Gamma(t, G)$ , which alone is sufficient to conclude that  $\mathfrak{M}_\Gamma \Vdash t:G$  in the case of (3.3.7).

In the case of (3.3.6), we need to show additionally that  $\mathfrak{M} \Vdash G$ .

Both logics  $\text{JT}_{\mathcal{CS}}$  and  $\text{LP}_{\mathcal{CS}}$ , where (3.3.6) is used, have axiom A4, so for them  $\text{JL}_{\mathcal{CS}} \vdash t:G \rightarrow G$ . By Lemma 2.6.2.6,  $t:G \rightarrow G \in \Gamma$ .

By Lemma 2.6.2.4,  $G \in \Gamma$ . Thus, by IH,  $\mathfrak{M}_\Gamma \Vdash G$ .

Let  $t:G \notin \Gamma$ . Then, by (3.3.12),  $\mathcal{A}_\Gamma(t, G) = \text{False}$ , so  $\mathfrak{M}_\Gamma \not\Vdash t:G$ .

This completes the proof of Lemma 3.3.5. □

Showing completeness is now easy. We need to provide a countermodel for each  $F$  such that  $\text{JL}_{\mathcal{CS}} \not\vdash F$ . By Theorem 3.2.21,  $\text{JL}_{\mathcal{CS}}$  is consistent. By Lemma 2.6.2.7, the set  $\{\neg F\}$  is  $\text{JL}_{\mathcal{CS}}$ -consistent. By Lemma 2.6.2.8, it can be extended to a maximal  $\text{JL}_{\mathcal{CS}}$ -consistent set  $\Gamma$ . By Lemma 3.3.5, there exists

an M-model  $\mathfrak{M}_\Gamma$  canonical for  $\Gamma$ . Since  $\neg F \in \Gamma$ , by Lemma 3.3.5,  $\mathfrak{M}_\Gamma \Vdash \neg F$ . Therefore,  $\mathfrak{M}_\Gamma \not\models F$ .

This completes the proof of Completeness Theorem 3.3.4.  $\square$

### 3.3.2 Epistemic F-models

F-models are a hybrid of M-models with Kripke models. They are closer to modal epistemic semantics and thus can be adapted to hybrid logics with both modal and justification knowledge assertions.

**Definition 3.3.6.** An *F-model for a justification logic*

$$\mathbf{JL}_{\mathcal{CS}} \in \{\mathbf{J}_{\mathcal{CS}}, \mathbf{JD}_{\mathcal{CS}}, \mathbf{JT}_{\mathcal{CS}}, \mathbf{J4}_{\mathcal{CS}}, \mathbf{JD4}_{\mathcal{CS}}, \mathbf{LP}_{\mathcal{CS}}\}$$

in language  $\mathcal{JL}$ , or for

$$\mathbf{JL}_{\mathcal{CS}} \in \{\mathbf{J5}_{\mathcal{CS}}, \mathbf{J45}_{\mathcal{CS}}, \mathbf{JD45}_{\mathcal{CS}}, \mathbf{JT45}_{\mathcal{CS}}\}$$

in language  $\mathcal{JL}(?)$ , where  $\mathcal{CS}$  is a constant specification for  $\mathbf{JL}$ , is a quadruple

$$\mathfrak{M} = (W, R, V, \mathcal{A}) ,$$

where  $W \neq \emptyset$  is a set of worlds,  $R \subseteq W \times W$  is a binary accessibility relation on  $W$ , the *propositional valuation*

$$V: SLet \rightarrow 2^W \tag{3.3.13}$$

assigns to each sentence letter  $p$  a set of worlds  $V(p)$  where  $p$  is true, and

$$\mathcal{A}: Tm \times Fm \rightarrow 2^W \quad (3.3.14)$$

is an *admissible evidence function*. Informally,  $\mathcal{A}(t, F) \subseteq W$  is a set of worlds where term  $t$  is considered admissible evidence for formula  $F$ .

The accessibility relation  $R$  must be

- *reflexive* if A4 is an axiom of JL;
- *transitive* if A5 is an axiom of JL;
- *serial* if A7 is an axiom of JL.

The admissible evidence function  $\mathcal{A}$  must satisfy the following closure conditions:

- *Application Closure*:  $\mathcal{A}(s, F \rightarrow G) \cap \mathcal{A}(t, F) \subseteq \mathcal{A}(s \cdot t, G)$ ;
- *Sum Closure*:  $\mathcal{A}(s, F) \cup \mathcal{A}(t, F) \subseteq \mathcal{A}(s + t, F)$ ;
- *CS Closure*: if  $c:A \in \mathcal{CS}$ , then  $\mathcal{A}(c, A) = W$  and for each  $n \geq 1$

$$\mathcal{A}(\underbrace{! \dots !}_n c, \underbrace{! \dots !}_{n-1} c : \dots : ! c : c : A) = W ;$$

- *Positive Introspection Closure* (if A5 is an axiom of JL):

$$\mathcal{A}(t, F) \subseteq \mathcal{A}(!t, t:F);$$

- *Monotonicity* (if A5 is an axiom of JL):

$$u \in \mathcal{A}(t, F) \text{ and } uRv \quad \text{yield} \quad v \in \mathcal{A}(t, F);$$

- *Negative Introspection Closure* (if A6 is an axiom of JL):<sup>3</sup>

$$[\mathcal{A}(t, F)]^c \subseteq \mathcal{A}(?t, \neg t : F)$$

for any formulas  $F$  and  $G$ , any terms  $t$  and  $s$ , any worlds  $u, v \in W$ , any  $c : A \in \mathcal{CS}$ , and any integer  $n \geq 1$ .

The *truth relation*  $\mathfrak{M}, u \models H$  is defined as follows:

$$\mathfrak{M}, u \Vdash p \quad \Leftrightarrow \quad u \in V(p) \quad (3.3.15)$$

$$\mathfrak{M}, u \not\Vdash \perp \quad (3.3.16)$$

$$\mathfrak{M}, u \Vdash F \rightarrow G \quad \Leftrightarrow \quad \mathfrak{M}, u \not\Vdash F \quad \text{or} \quad \mathfrak{M}, u \Vdash G \quad (3.3.17)$$

$$\begin{aligned} \mathfrak{M}, u \Vdash t : F \quad \Leftrightarrow \quad u \in \mathcal{A}(t, F) \quad \text{and} \\ \mathfrak{M}, w \Vdash F \text{ for all } w \in W \text{ such that } uRw \quad (3.3.18) \end{aligned}$$

for any sentence letter  $p$ , any formulas  $F$  and  $G$ , any world  $u \in W$ , and any term  $t$ .

In addition, logics with axiom A6 must satisfy

- *Strong Evidence Property*: if  $u \in \mathcal{A}(t, F)$ , then  $\mathfrak{M}, u \Vdash t : F$

for any formula  $F$ , any term  $t$ , and any world  $u \in W$ . ◀

---

<sup>3</sup>Here  $[X]^c$  denotes the complement of set  $X$ .

**Definition 3.3.7.** A formula  $F$  is *valid in an F-model*  $\mathfrak{M} = (W, R, V, \mathcal{A})$ , written  $\mathfrak{M} \Vdash F$ , if  $F$  is true in all worlds  $w \in W$ . ◀

**Definition 3.3.8.** A formula  $F$  is *satisfiable in an F-model*  $\mathfrak{M} = (W, R, V, \mathcal{A})$  if  $F$  is true in at least one world  $w \in W$ . ◀

**Definition 3.3.9.** A formula  $F$  is called  $\text{JL}_{\mathcal{CS}}$ -*valid* if  $F$  is valid in all F-models for  $\text{JL}_{\mathcal{CS}}$ . ◀

**Definition 3.3.10.** A formula  $F$  is called  $\text{JL}_{\mathcal{CS}}$ -*satisfiable* if  $F$  is satisfiable in at least one F-model for  $\text{JL}_{\mathcal{CS}}$ . ◀

**Definition 3.3.11.** A formula  $F$  is called  $\text{JL}_{\mathcal{CS}}$ -*refutable* if  $\neg F$  is satisfiable in at least one F-model for  $\text{JL}_{\mathcal{CS}}$ . ◀

The following proposition, analogous to Prop. 3.3.3, can be used for F-models with Positive Introspection Closure condition:

**Proposition 3.3.12.** *Let  $\mathcal{A}: Tm \times Fm \rightarrow 2^W$  satisfy both the Positive Introspection Closure condition and the following*

- Simplified  $\mathcal{CS}$  Closure: *if  $c: A \in \mathcal{CS}$ , then  $\mathcal{A}(c, A) = W$ .*

*Then,  $\mathcal{A}$  also satisfies the full  $\mathcal{CS}$  Closure condition.*

Table 3.3.2 (cf. Tables 3.2.1 and 3.3.1) summarizes which closure conditions should be used for various justification logics. In this table, using

Table 3.3.2: F-models: Conditions on the admissible evidence function

Logic	Appl. Clos.	Sum Clos.	$\mathcal{CS}$ Clos.	Simp. $\mathcal{CS}$ Closure	Pos. Intr. Closure	Monot.	Neg. Intr. Closure
$J_{CS}$	✓	✓	✓				
$JD_{CS}$	✓	✓	✓				
$JT_{CS}$	✓	✓	✓				
$J4_{CS}$	✓	✓		✓	✓	✓	
$JD4_{CS}$	✓	✓		✓	✓	✓	
$LP_{CS}$	✓	✓		✓	✓	✓	
$J5_{CS}$	✓	✓	✓				✓
$J45_{CS}$	✓	✓		✓	✓	✓	✓
$JD45_{CS}$	✓	✓		✓	✓	✓	✓
$JT45_{CS}$	✓	✓		✓	✓	✓	✓

Prop. 3.3.12, the  $\mathcal{CS}$  Closure is replaced by its simplified version whenever possible. Table 3.3.3 details the requirements on the binary relation  $R$  and the necessity of the Strong Evidence Property.

**Definition 3.3.13.** Let  $\mathfrak{M} = (W, R, V, \mathcal{A})$  be an F-model for a justification logic  $JL_{CS}$ . We will sometimes consider the admissible evidence function  $\mathcal{A}$  separately from any F-models. In such cases, we still need to know the set  $W$  in order to verify the  $\mathcal{CS}$  Closure condition. Therefore, we will call  $\mathcal{A}$  an *admissible for  $JL_{CS}$  evidence function on set  $W \neq \emptyset$* .

For justification logics with positive introspection,  $\mathcal{A}$  must satisfy the Monotonicity condition, which also depends on the binary relation  $R$ . For this reason, we will also call  $\mathcal{A}$  an *admissible for  $JL_{CS}$  evidence function on a (monomodal) Kripke frame  $(W, R)$* . ◀

Table 3.3.3: F-models: Conditions on  $R$  and the Strong Evidence Property

Logic	Reflexive Bin. Rel.	Transitive Bin. Rel.	Serial Bin. Rel.	Strong Ev. Prop.
$J_{CS}$				
$JD_{CS}$			✓	
$JT_{CS}$	✓			
$J4_{CS}$		✓		
$JD4_{CS}$		✓	✓	
$LP_{CS}$	✓	✓		
$J5_{CS}$				✓
$J45_{CS}$		✓		✓
$JD45_{CS}$		✓	✓	✓
$JT45_{CS}$	✓	✓		✓

**Theorem 3.3.14 (Completeness Theorem for F-models, [Fit05, Pac05, Rub06b, Art07, Kuz08]).** *Let  $CS$  be*

1. *a constant specification for  $JL \in \{J, JT, J4, LP, J5, J45, JT45\}$  or*
2. *an axiomatically appropriate constant specification for  $JL \in \{JD, JD4, JD45\}$ .*

*Then, for any formula  $F$ ,*

$$JL_{CS} \vdash F \quad \iff \quad F \text{ is } JL_{CS}\text{-valid.}$$

*Note 3.3.15.* Logics  $JD_{CS}$ ,  $JD4_{CS}$ ,  $JD45_{CS}$  are sound w.r.t. F-models even when  $CS$  is not axiomatically appropriate.

*Proof.* Let  $JL_{CS}$  satisfy one of the cases described in the formulation. Throughout the proof we will write “valid” instead of “ $JL_{CS}$ -valid.”

We will first prove *soundness* by induction on the derivation in  $\text{JL}_{CS}$ .

Consider an arbitrary F-model  $\mathfrak{M} = (W, R, V, \mathcal{A})$  for  $\text{JL}_{CS}$ :

A1. All propositional axioms are valid and the *modus ponens* rule is admissible in  $\mathfrak{M}$  since the propositional cases (3.3.16)–(3.3.17) for  $\Vdash$  in Def. 3.3.6 are classical and local, i.e., work entirely within each world.

A2. *Application Axiom*  $s:(F \rightarrow G) \rightarrow (t:F \rightarrow s \cdot t:G)$

Let  $\mathfrak{M}, u \Vdash s:(F \rightarrow G)$  and  $\mathfrak{M}, u \Vdash t:F$ . To show validity of A2, we need to show that  $\mathfrak{M}, u \Vdash s \cdot t:G$ .

By (3.3.18),  $u \in \mathcal{A}(s, F \rightarrow G) \cap \mathcal{A}(t, F)$ . Hence, by the Application Closure condition,  $u \in \mathcal{A}(s \cdot t, G)$ .

Also, by (3.3.18),  $\mathfrak{M}, w \Vdash F \rightarrow G$  and  $\mathfrak{M}, w \Vdash F$  for any  $uRw$ . Hence, by (3.3.17),  $\mathfrak{M}, w \Vdash G$  for any  $uRw$ . Combined with  $u \in \mathcal{A}(s \cdot t, G)$ , this yields  $\mathfrak{M}, u \Vdash s \cdot t:G$ .

A3. *Monotonicity Axiom*  $s:F \rightarrow s + t:F$

$$t:F \rightarrow s + t:F$$

W.l.o.g. we will show validity of the first formula. Let  $\mathfrak{M}, u \Vdash s:F$ .

To show validity of A3, we need to show that  $\mathfrak{M}, u \Vdash s + t:F$ .

By (3.3.18),  $u \in \mathcal{A}(s, F)$ . By the Sum Closure,  $u \in \mathcal{A}(s + t, F)$ .

Also, by (3.3.18),  $\mathfrak{M}, w \Vdash F$  for any  $uRw$ . Taking into account that  $u \in \mathcal{A}(s + t, F)$ , this yields  $\mathfrak{M}, u \Vdash s + t : F$ .

A4. *Factivity Axiom*  $t : F \rightarrow F$

Let  $\mathfrak{M}, u \Vdash t : F$ . To show validity of A4, we need to demonstrate  $\mathfrak{M}, u \Vdash F$ .

By (3.3.18),  $\mathfrak{M}, w \Vdash F$  for any  $uRw$ . F-models for all logics with axiom A4, i.e.,  $\text{JT}_{CS}$ ,  $\text{LP}_{CS}$ , and  $\text{JT45}_{CS}$ , must have reflexive  $R$ ; hence,  $uRu$  and  $\mathfrak{M}, u \Vdash F$ .

A5. *Positive Introspection*  $t : F \rightarrow !t : t : F$

Let  $\mathfrak{M}, u \Vdash t : F$ . To show validity of A5, we need to show that  $\mathfrak{M}, u \Vdash !t : t : F$ .

By (3.3.18),  $u \in \mathcal{A}(t, F)$ . F-models for all logics with axiom A5, i.e.,  $\text{J4}_{CS}$ ,  $\text{JD4}_{CS}$ ,  $\text{LP}_{CS}$ ,  $\text{J45}_{CS}$ ,  $\text{JD45}_{CS}$ , and  $\text{JT45}_{CS}$ , must satisfy the Positive Introspection Closure condition. Hence,  $u \in \mathcal{A}(!t, t : F)$ .

It remains to show that  $\mathfrak{M}, w \Vdash t : F$  for any  $uRw$ . By (3.3.18),  $\mathfrak{M}, w \Vdash F$  for any such  $w$ . F-models for all logics with axiom A5 must also satisfy the Monotonicity condition and have a transitive  $R$ . By Monotonicity,  $w \in \mathcal{A}(t, F)$ . In addition, for any  $wRz$ , by transitivity,

also  $uRz$ , so  $\mathfrak{M}, z \Vdash F$  for any  $wRz$ .

By (3.3.18), indeed  $\mathfrak{M}, w \Vdash t : F$  for any  $uRw$ . Since  $u \in \mathcal{A}(!t, t : F)$ ,

again by (3.3.18), we have  $\mathfrak{M}, u \Vdash !t : t : F$ .

A6. *Negative Introspection*  $\neg t : F \rightarrow ?t : \neg t : F$

Let  $\mathfrak{M}, u \Vdash \neg t : F$ . To show validity of A6, we need to show that

$\mathfrak{M}, u \Vdash ?t : \neg t : F$ .

F-models for all logics with axiom A6, i.e.,  $J5_{CS}$ ,  $J45_{CS}$ ,  $JD45_{CS}$ , and  $JT45_{CS}$ , must satisfy both the Negative Introspection Closure condition and the Strong Evidence Property.

From  $\mathfrak{M}, u \not\Vdash t : F$ , by Strong Evidence, we conclude that  $u \notin \mathcal{A}(t, F)$ .

Then,  $u \in \mathcal{A}(?t, \neg t : F)$  by the Negative Introspection Closure. Thus,

$\mathfrak{M}, u \Vdash ?t : \neg t : F$  by Strong Evidence.

A7. *Consistency Axiom*  $t : \perp \rightarrow \perp$

To show validity of A7, we need to show that  $\mathfrak{M}, u \not\Vdash t : \perp$  for any term  $t$  and any world  $u \in W$ .

F-models for all logics with axiom A7, i.e.,  $JD_{CS}$ ,  $JD4_{CS}$ , and  $JD45_{CS}$ , must have serial  $R$ , i.e., there must exist a world  $w$  accessible from  $u$ .

By (3.3.16),  $\mathfrak{M}, w \not\Vdash \perp$ . Hence, by (3.3.18),  $\mathfrak{M}, u \not\Vdash t : \perp$ .

$R4_{\mathcal{CS}}$ . *Axiom Internalization Rule* restricted to  $\mathcal{CS}$

$$\frac{c:A \in \mathcal{CS}}{c:A}$$

To show admissibility of  $R4_{\mathcal{CS}}$ , we need to show that  $\mathfrak{M} \Vdash c:A$  for any  $c:A \in \mathcal{CS}$ .

We have already shown  $JL_{\mathcal{CS}}$ -validity of all axioms of  $JL$ , so  $\mathfrak{M} \Vdash A$ .

By the  $\mathcal{CS}$  Closure Condition,  $\mathcal{A}(c, A) = W$ . Hence,  $\mathfrak{M} \Vdash c:A$ .

$R4_{\mathcal{CS}}^!$ . *Axiom Internalization Rule* with positive introspection

restricted to  $\mathcal{CS}$

$$\frac{c:A \in \mathcal{CS}}{\underbrace{!!!\dots!}_n c:\dots:!!c:!!c:c:A}$$

To show admissibility of  $R4_{\mathcal{CS}}^!$ , we need to show that

$$\mathfrak{M} \Vdash \underbrace{!!!\dots!}_n c:\dots:!!c:!!c:c:A$$

for any  $c:A \in \mathcal{CS}$  and any integer  $n \geq 0$ . We will use induction on  $n$ .

*Base.*  $n = 0$ . This case coincides with rule  $R4_{\mathcal{CS}}$  and has already been proven.

*Step.* Assume for  $n = k$  that

$$\mathfrak{M} \Vdash \underbrace{!!!\dots!}_k c:\dots:!!c:!!c:c:A .$$

By the  $\mathcal{CS}$  Closure condition,

$$\mathcal{A}(\underbrace{!!!\dots!}_{k+1} c, \underbrace{!!!\dots!}_k c:\dots:!!c:!!c:c:A) = W .$$

Thus, by (3.3.18),

$$\mathfrak{M} \Vdash \underbrace{!!! \dots !}_{k+1} c : \underbrace{!!! \dots !}_k c : \dots : !!! c : ! c : c : A .$$

This completes the soundness proof. No particular properties of  $\mathcal{CS}$ , such as being axiomatically appropriate, have been used in it. Hence,  $\mathbb{J}\mathcal{L}_{\mathcal{CS}}$  is sound w.r.t. F-models for an arbitrary  $\mathcal{CS}$ .

The completeness is shown by the standard maximal consistency argument through construction of the canonical model for the logic.

Before we define the canonical model, we will need the following notation:

**Definition 3.3.16.** Let  $\Gamma$  be a set of justification formulas.

$$\Gamma^\# = \{F \mid t:F \in \Gamma \text{ for some term } t\} . \quad (3.3.19)$$

◀

**Definition 3.3.17.** The *canonical F-model* for logic  $\mathbb{J}\mathcal{L}_{\mathcal{CS}}$  is a quadruple

$$\mathfrak{M} = (W, R, V, \mathcal{A})$$

defined as follows:

$$W \Leftrightarrow \{\Gamma \mid \Gamma \text{ is a maximal } \mathbf{JL}_{\mathcal{CS}}\text{-consistent set}\} \quad (3.3.20)$$

$$\Gamma R \Delta \Leftrightarrow \Gamma^\# \subseteq \Delta \quad (3.3.21)$$

$$V(p) \Leftrightarrow \{\Gamma \in W \mid p \in \Gamma\} \quad (3.3.22)$$

$$\mathcal{A}(t, F) \Leftrightarrow \{\Gamma \in W \mid t:F \in \Gamma\} \quad (3.3.23)$$

for any sentence letter  $p$ , any term  $t$ , and any formula  $F$ . ◀

We will first prove that the canonical model constructed in such a way is actually an F-model.

**Lemma 3.3.18.** *Let  $\mathcal{CS}$  be*

1. *a constant specification for  $\mathbf{JL} \in \{\mathbf{J}, \mathbf{JT}, \mathbf{J4}, \mathbf{LP}, \mathbf{J5}, \mathbf{J45}, \mathbf{JT45}\}$  or*
2. *an axiomatically appropriate constant specification for  $\mathbf{JL} \in \{\mathbf{JD}, \mathbf{JD4}, \mathbf{JD45}\}$ .*

*Then, the canonical F-model  $\mathfrak{M}$  for  $\mathbf{JL}_{\mathcal{CS}}$  from Def. 3.3.17 is an F-model for  $\mathbf{JL}_{\mathcal{CS}}$ .*

*Proof.* There are many conditions to be verified. Let us start with the condition on  $W$ .  $\mathbf{JL}_{\mathcal{CS}}$  is consistent by Theorem 3.2.21, so there exist consistent sets that can be extended to maximal consistent sets by Lemma 2.6.2.8. Hence,  $W \neq \emptyset$ .

Let us verify that  $R$  defined in (3.3.21) satisfies all the necessary conditions:

- *Reflexivity* (required for  $\text{JT}_{CS}$ ,  $\text{LP}_{CS}$ , and  $\text{JT45}_{CS}$ ):

We need to show that  $\Gamma R \Gamma$  for any maximal  $\text{JL}_{CS}$ -consistent set  $\Gamma$ . In other words, we need to show that  $\Gamma^\# \subseteq \Gamma$ , i.e.,  $t:F \in \Gamma$  implies  $F \in \Gamma$  for any term  $t$  and any formula  $F$ .

Let  $t:F \in \Gamma$ . All the three logics listed above have axiom A4; therefore,  $\text{JL}_{CS} \vdash t:F \rightarrow F$ . By Lemma 2.6.2.6,  $t:F \rightarrow F \in \Gamma$ . By Lemma 2.6.2.4,  $F \in \Gamma$ .

- *Transitivity* (required for  $\text{J4}_{CS}$ ,  $\text{JD4}_{CS}$ ,  $\text{LP}_{CS}$ ,  $\text{J45}_{CS}$ ,  $\text{JD45}_{CS}$ ,  $\text{JT45}_{CS}$ ):

We need to show that  $\Gamma R \Delta$  and  $\Delta R \Sigma$  imply  $\Gamma R \Sigma$  for any maximal  $\text{JL}_{CS}$ -consistent sets  $\Gamma$ ,  $\Delta$ , and  $\Sigma$ .

Let  $\Gamma R \Delta$ ,  $\Delta R \Sigma$ , and  $t:F \in \Gamma$ . We need to show that  $F \in \Sigma$ .

All the six logics listed above have axiom A5, so  $\text{JL}_{CS} \vdash t:F \rightarrow !t:t:F$ .

By Lemma 2.6.2.6,  $t:F \rightarrow !t:t:F \in \Gamma$ . By Lemma 2.6.2.4,  $!t:t:F \in \Gamma$ .

Therefore,  $t:F \in \Delta$  since  $\Gamma^\# \subseteq \Delta$ . Finally,  $F \in \Sigma$  since  $\Delta^\# \subseteq \Sigma$ .

- *Seriality* (required for  $\text{JD}_{CS}$ ,  $\text{JD4}_{CS}$ , and  $\text{JD45}_{CS}$ ):

For these three logics, we will use an extra assumption that  $\mathcal{CS}$  is axiomatically appropriate.

We need to show that  $(\forall \Gamma \in W)(\exists \Delta \in W)\Gamma R\Delta$ . It is sufficient to show that  $\Gamma^\sharp$  is  $\mathbf{JL}_{\mathcal{CS}}$ -consistent for any  $\Gamma \in W$ . Indeed, if  $\Gamma^\sharp$  is consistent, by Lemma 2.6.2.8, it can be extended to some maximal  $\mathbf{JL}_{\mathcal{CS}}$ -consistent set  $\Delta \supseteq \Gamma^\sharp$  that would be accessible from  $\Gamma$  by definition (3.3.21) of  $R$ . Suppose towards a contradiction that  $\Gamma^\sharp$  is not  $\mathbf{JL}_{\mathcal{CS}}$ -consistent, which would imply, by Lemma 2.6.2.9, that

$$F_1, \dots, F_n \vdash_{\mathbf{JL}_{\mathcal{CS}}} \perp$$

for some  $s_i : F_i \in \Gamma$ ,  $i = 1, \dots, n$ . Internalizing this derivation by Lemma 3.2.22, using the axiomatic appropriateness of  $\mathcal{CS}$ , we would get

$$x_1 : F_1, \dots, x_n : F_n \vdash_{\mathbf{JL}_{\mathcal{CS}}} t(x_1, \dots, x_n) : \perp$$

for fresh variables  $x_i$ ,  $i = 1, \dots, n$ , and some term  $t(x_1, \dots, x_n)$ . The simultaneous substitution of  $s_i$  for  $x_i$ ,  $i = 1, \dots, n$ , in this derivation, given the axiomatic appropriateness of our  $\mathcal{CS}$ , would yield, by Lemma 3.2.30,

$$s_1 : F_1, \dots, s_n : F_n \vdash_{\mathbf{JL}_{\mathcal{CS}}} \bar{t}(s_1, \dots, s_n) : \perp$$

for some term  $\bar{t}(x_1, \dots, x_n)$  obtained from  $t(x_1, \dots, x_n)$  by (possibly) replacing some justification constants with other constants. All the three logics listed above have axiom A7, so

$$\mathbf{JL}_{CS} \vdash \bar{t}(s_1, \dots, s_n) : \perp \rightarrow \perp ;$$

hence,

$$s_1 : F_1, \dots, s_n : F_n \vdash_{\mathbf{JL}_{CS}} \perp .$$

The latter statement clearly contradicts the consistency of  $\Gamma$ . This contradiction shows that  $\Gamma^\sharp$  is  $\mathbf{JL}_{CS}$ -consistent.

We will now turn to showing that  $\mathcal{A}$ , defined in (3.3.23), is indeed an admissible for  $\mathbf{JL}_{CS}$  evidence function. We need to verify the following conditions:

- *Application Closure:*  $\mathcal{A}(s, F \rightarrow G) \cap \mathcal{A}(t, F) \subseteq \mathcal{A}(s \cdot t, G)$ .

Let  $\Gamma \in \mathcal{A}(s, F \rightarrow G) \cap \mathcal{A}(t, F)$ . By (3.3.23), it means that

$$\{s : (F \rightarrow G), \quad t : F\} \subset \Gamma .$$

Since  $\mathbf{JL}_{CS} \vdash s : (F \rightarrow G) \rightarrow (t : F \rightarrow s \cdot t : G)$ , by Lemma 2.6.2.6,

$$s : (F \rightarrow G) \rightarrow (t : F \rightarrow s \cdot t : G) \in \Gamma .$$

Closing by *modus ponens* twice by Lemma 2.6.2.4, we get  $s \cdot t : G \in \Gamma$ , and by (3.3.23),  $\Gamma \in \mathcal{A}(s \cdot t, G)$ .

- *Sum Closure*:  $\mathcal{A}(s, F) \cup \mathcal{A}(t, F) \subseteq \mathcal{A}(s + t, F)$ .

Let, w.l.o.g.  $\Gamma \in \mathcal{A}(s, F)$ . By (3.3.23), it means that  $s : F \in \Gamma$ . Since

$\text{JL}_{\mathcal{CS}} \vdash s : F \rightarrow s + t : F$ , by Lemma 2.6.2.6,

$$s : F \rightarrow s + t : F \in \Gamma .$$

Closing by *modus ponens* by Lemma 2.6.2.4, we get  $s + t : F \in \Gamma$ , and

by (3.3.23),  $\Gamma \in \mathcal{A}(s + t, F)$ .

- *CS Closure*: for any  $c : A \in \mathcal{CS}$ , we need to show  $\mathcal{A}(c, A) = W$  and, in addition,

$$\mathcal{A}(\underbrace{!! \dots !}_n c, \underbrace{! \dots !}_{n-1} c : \dots : !! c : ! c : c : A) = W$$

for any integer  $n \geq 1$ .

For each  $c : A \in \mathcal{CS}$  and each integer  $n \geq 0$ ,

$$\text{JL}_{\mathcal{CS}} \vdash \underbrace{!! \dots !}_n c : \dots : !! c : ! c : c : A$$

– by rule  $\text{R4}_{\mathcal{CS}}^!$ , for logics  $\text{J}_{\mathcal{CS}}$ ,  $\text{JD}_{\mathcal{CS}}$ ,  $\text{JT}_{\mathcal{CS}}$ , and  $\text{J5}_{\mathcal{CS}}$  or

– by rule  $\text{R4}_{\mathcal{CS}}$ , axiom  $\text{A5}$ , and *modus ponens*, for logics  $\text{J4}_{\mathcal{CS}}$ ,  $\text{JD4}_{\mathcal{CS}}$ ,

$\text{LP}_{\mathcal{CS}}$ ,  $\text{J45}_{\mathcal{CS}}$ ,  $\text{JD45}_{\mathcal{CS}}$ , and  $\text{JT45}_{\mathcal{CS}}$ .

By Lemma 2.6.2.6, for any maximal  $\text{JL}_{\mathcal{CS}}$ -consistent set  $\Gamma$ ,

$$\underbrace{!! \dots !}_n c : \dots : !! c : ! c : c : A \in \Gamma ,$$

in particular, for  $n = 0$ ,  $c : A \in \Gamma$ . Therefore, by (3.3.23),  $\Gamma \in \mathcal{A}(c, A)$  for any  $\Gamma$  and, in addition,

$$\Gamma \in \mathcal{A}(\underbrace{!! \dots !}_n c, \underbrace{! \dots !}_{n-1} c : \dots : !! c : ! c : c : A)$$

for any integer  $n \geq 1$ .

- *Positive Introspection Closure* (required for  $\mathbf{J4}_{CS}$ ,  $\mathbf{JD4}_{CS}$ ,  $\mathbf{LP}_{CS}$ ,  $\mathbf{J45}_{CS}$ ,  $\mathbf{JD45}_{CS}$ , and  $\mathbf{JT45}_{CS}$ ):  $\mathcal{A}(t, F) \subseteq \mathcal{A}(!t, t : F)$ .

Let  $\Gamma \in \mathcal{A}(t, F)$ . By (3.3.23), it means that  $t : F \in \Gamma$ . All the six logics listed above have axiom A5, so  $\mathbf{JL}_{CS} \vdash t : F \rightarrow !t : t : F$ . By Lemma 2.6.2.6,

$$t : F \rightarrow !t : t : F \in \Gamma .$$

Closing by *modus ponens* by Lemma 2.6.2.4, we get  $!t : t : F \in \Gamma$ , and by (3.3.23),  $\Gamma \in \mathcal{A}(!t, t : F)$ .

- *Monotonicity* (required for  $\mathbf{J4}_{CS}$ ,  $\mathbf{JD4}_{CS}$ ,  $\mathbf{LP}_{CS}$ ,  $\mathbf{J45}_{CS}$ ,  $\mathbf{JD45}_{CS}$ ,  $\mathbf{JT45}_{CS}$ ): if  $\Gamma \in \mathcal{A}(t, F)$  and  $\Gamma R \Delta$ , then  $\Delta \in \mathcal{A}(t, F)$ .

Let  $\Gamma \in \mathcal{A}(t, F)$  and  $\Gamma R \Delta$ . For all the six logics listed above, the Positive Introspection Closure has just been proven; therefore,  $\Gamma \in \mathcal{A}(t, F)$  implies  $\Gamma \in \mathcal{A}(!t, t : F)$ . By (3.3.23), the latter means that

$!t:t:F \in \Gamma$ . Thus, by the definition (3.3.21) of  $R$ , we have  $t:F \in \Delta$ .

Finally, by (3.3.23),  $\Delta \in \mathcal{A}(t, F)$ .

- *Negative Introspection Closure* (required for  $\mathbf{J5}_{CS}$ ,  $\mathbf{J45}_{CS}$ ,  $\mathbf{JD45}_{CS}$ ,  $\mathbf{JT45}_{CS}$ ):

$$[\mathcal{A}(t, F)]^c \subseteq \mathcal{A}(?t, \neg t:F).$$

Let  $\Gamma \notin \mathcal{A}(t, F)$ . By (3.3.23), it means that  $t:F \notin \Gamma$ . Since  $\Gamma$  is maximal  $\mathbf{JL}_{CS}$ -consistent, by Lemma 2.6.2.3,  $\neg t:F \in \Gamma$ . All the four logics listed above have axiom A6, so  $\mathbf{JL}_{CS} \vdash \neg t:F \rightarrow ?t:\neg t:F$ . By Lemma 2.6.2.6,

$$\neg t:F \rightarrow ?t:\neg t:F \in \Gamma .$$

Closing by *modus ponens* by Lemma 2.6.2.4, we get  $?t:\neg t:F \in \Gamma$  and, by (3.3.23),  $\Gamma \in \mathcal{A}(?t, \neg t:F)$ .

It only remains to show that the Strong Evidence Property is satisfied for logics  $\mathbf{J5}_{CS}$ ,  $\mathbf{J45}_{CS}$ ,  $\mathbf{JD45}_{CS}$ ,  $\mathbf{JT45}_{CS}$ .

We will actually prove a stronger statement that the canonical F-models for all the ten logics considered so far enjoy Strong Evidence. But first we will need to show the fundamental property of canonical models, which, after Melvin Fitting, we will call the *Truth Lemma*.

**Lemma 3.3.19 (Truth Lemma).** *Let  $CS$  be a constant specification for*

$$\mathbf{JL} \in \{\mathbf{J}, \mathbf{JD}, \mathbf{JT}, \mathbf{J4}, \mathbf{JD4}, \mathbf{LP}, \mathbf{J5}, \mathbf{J45}, \mathbf{JD45}, \mathbf{JT45}\} .$$

The canonical  $F$ -model  $\mathfrak{M}$  for  $\mathbb{JL}_{CS}$  from Def. 3.3.17 enjoys the following property:

$$\mathfrak{M}, \Gamma \Vdash F \quad \iff \quad F \in \Gamma .$$

*Note 3.3.20.* Strictly speaking, for logics with negative introspection, we do not yet know whether  $\mathfrak{M}$  is a proper  $F$ -model, but we can still operate with  $\Vdash$  as prescribed in (3.3.15)-(3.3.18).

*Proof of the Truth Lemma.* Induction on  $|F|$ :

$F = p$ . For any sentence letter  $p$ , the statement follows directly from (3.3.22) and (3.3.15):

$$\mathfrak{M}, \Gamma \Vdash p \quad \iff \quad \Gamma \in V(p) \quad \iff \quad p \in \Gamma .$$

Boolean cases are trivial.

$F = t : G$ . Let  $t : G \in \Gamma$ . First of all, by (3.3.23),  $\Gamma \in \mathcal{A}(t, G)$ . Further,  $G \in \Delta$  for any  $\Delta$  accessible from  $\Gamma$ , by (3.3.21). So by IH,  $\mathfrak{M}, \Delta \Vdash G$  for any  $\Gamma R \Delta$ . Combined with  $\Gamma \in \mathcal{A}(t, G)$ , this yields  $\mathfrak{M}, \Gamma \Vdash t : G$  by (3.3.18).

Let  $t : G \notin \Gamma$ . Then, by (3.3.23),  $\Gamma \notin \mathcal{A}(t, G)$ , so  $\mathfrak{M}, \Gamma \not\Vdash t : G$  by (3.3.18).

This completes the proof of the Truth Lemma 3.3.19.  $\square$

We are now ready to finish the proof of Lemma 3.3.18 by showing the Strong Evidence Property of all canonical F-models:

- *Strong Evidence Property:* if  $\Gamma \in \mathcal{A}(t, F)$ , then  $\mathfrak{M}, \Gamma \Vdash t:F$ .

By (3.3.23),  $\Gamma \in \mathcal{A}(t, F)$  means that  $t:F \in \Gamma$ . By the Truth Lemma 3.3.19,  $\mathfrak{M}, \Gamma \Vdash t:F$ .

Thus, the canonical F-model for each  $\mathbf{JL}_{\mathcal{CS}}$  is indeed an F-model for  $\mathbf{JL}_{\mathcal{CS}}$ . In addition, this F-model satisfies the Truth Lemma 3.3.19. This completes the proof of Lemma 3.3.18.  $\square$

We are finally ready to show completeness of  $\mathbf{JL}_{\mathcal{CS}}$  w.r.t. its F-models. The canonical model  $\mathfrak{M} = (W, R, V, \mathcal{A})$  for  $\mathbf{JL}_{\mathcal{CS}}$  constructed in Def. 3.3.17 is sufficient to refute all formulas  $F$  such that  $\mathbf{JL}_{\mathcal{CS}} \not\vdash F$ . By Lemma 3.3.18,  $\mathfrak{M}$  is an F-model for  $\mathbf{JL}_{\mathcal{CS}}$ .

Consider any such  $F$ . By Theorem 3.2.21,  $\mathbf{JL}_{\mathcal{CS}}$  is consistent. Then, by Lemma 2.6.2.7, the set  $\{\neg F\}$  is  $\mathbf{JL}_{\mathcal{CS}}$ -consistent. By Lemma 2.6.2.8, it can be extended to a maximal  $\mathbf{JL}_{\mathcal{CS}}$ -consistent set  $\Delta \ni \neg F$ . By the Truth Lemma 3.3.19,  $\mathfrak{M}, \Delta \Vdash \neg F$ , so  $\mathfrak{M}, \Delta \not\vdash F$ .

This completes the proof of Completeness Theorem 3.3.14.  $\square$

As mentioned in the proof above, the canonical F-model for each of the ten justification logics enjoys the Strong Evidence Property. Thus, although

it is not necessary for the soundness of the logics without axiom A6, the Strong Evidence Property can still be added to strengthen the completeness claim for them. The following theorem lists several other properties of the canonical F-models for several logics and formulates stronger completeness results for them:

**Theorem 3.3.21 (Strong Completeness Theorem for F-models, [Fit05, Pac05, Rub06b, Art07, Kuz08]).**

1.  $J_{CS}$ ,  $JT_{CS}$ ,  $J4_{CS}$ , and  $LP_{CS}$  are complete w.r.t. the class of their F-models that additionally satisfy
  - *Strong Evidence Property*
2.  $JD_{CS}$  and  $JD4_{CS}$  with axiomatically appropriate  $CS$  are complete w.r.t. the class of their F-models that additionally satisfy
  - *Strong Evidence Property*
3.  $J5_{CS}$  is complete w.r.t. the class of its F-models  $\mathfrak{M} = (W, R, V, \mathcal{A})$  with Euclidean  $R$  that additionally satisfy
  - *Strong Evidence Property*
  - *Anti-Monotonicity: if  $u \notin \mathcal{A}(t, F)$  and  $uRw$ , then  $w \notin \mathcal{A}(t, F)$  for any term  $t$ , any formula  $F$ , and any worlds  $u, w \in W$ .*

4.  $J45_{CS}$  is complete w.r.t. the class of its  $F$ -models  $\mathfrak{M} = (W, R, V, \mathcal{A})$  with Euclidean  $R$  that additionally satisfy

- *Strong Evidence Property*
- *Stability: if  $uRw$ , then  $u \in \mathcal{A}(t, F) \iff w \in \mathcal{A}(t, F)$  for any formula  $F$ , any term  $t$ , and any worlds  $u, w \in W$ .*

5.  $JD45_{CS}$  with an axiomatically appropriate  $CS$  is complete w.r.t. the class of its  $F$ -models with Euclidean  $R$  that additionally satisfy

- *Strong Evidence Property*
- *Stability*

6.  $JT45_{CS}$  is complete w.r.t. the class of its  $F$ -models, with  $R$  being an equivalence relation, that additionally satisfy

- *Strong Evidence Property*
- *Stability*

7. In addition, for each of logics  $J_{CS}$ ,  $JD_{CS}$ ,  $JT_{CS}$ ,  $J4_{CS}$ ,  $JD4_{CS}$ ,  $LP_{CS}$ ,  $J5_{CS}$ ,  $J45_{CS}$ ,  $JD45_{CS}$ , and  $JT45_{CS}$  with a schematic and axiomatically appropriate  $CS$ , the following property can be added to the list of requirements on the model  $\mathfrak{M} = (W, R, V, \mathcal{A})$ :

- Fully Explanatory Property:

*For any world  $u \in W$  and any formula  $F$ , if  $\mathfrak{M}, w \Vdash F$  for all  $w$  such that  $uRw$ , there must exist a justification term  $t$  such that  $\mathfrak{M}, u \Vdash t:F$ .*

*Note 3.3.22.* For logics  $\text{JD}_{\mathcal{CS}}$ ,  $\text{JD4}_{\mathcal{CS}}$ , and  $\text{JD45}_{\mathcal{CS}}$ , the axiomatic appropriateness of  $\mathcal{CS}$  is necessary already for the basic completeness theorem. For the remaining seven logics, as will be seen from the proof below, the schemativeness and axiomatic appropriateness of  $\mathcal{CS}$  is only used in the proof of the Fully Explanatory Property.

*Proof of Theorem 3.3.21.* In the proof of Theorem 3.3.14, we have already established the Strong Evidence Property of the canonical F-models for all the ten logics. It suffices to show that the canonical F-model  $\mathfrak{M} = (W, R, V, \mathcal{A})$  for each logic  $\text{JL}_{\mathcal{CS}}$  additionally satisfies the remaining properties:

- *Fully Explanatory Property:* If  $\mathfrak{M}, \Delta \Vdash F$  for all  $\Delta$  such that  $\Gamma R\Delta$ , there must exist a justification term  $t$  such that  $\mathfrak{M}, \Gamma \Vdash t:F$ .

For some  $\Gamma \in W$  and some formula  $F$ , let

$$\mathfrak{M}, \Delta \Vdash F \quad \text{for all } \Gamma R\Delta \quad . \quad (3.3.24)$$

Suppose towards a contradiction that

$$\text{there is no justification term } t \text{ such that } \mathfrak{M}, \Gamma \Vdash t : F. \quad (3.3.25)$$

Then, the set

$$\Gamma^\sharp \cup \{\neg F\} \quad (3.3.26)$$

would have to be  $\mathbf{JL}_{\mathcal{CS}}$ -consistent. Indeed, according to Lemma 2.6.2.9, the inconsistency of set (3.3.26) would mean that

$$G_1, \dots, G_n, \neg F \vdash_{\mathbf{JL}_{\mathcal{CS}}} \perp$$

for some  $G_i \in \Gamma^\sharp$ ,  $i = 1, \dots, n$ , or equivalently that

$$G_1, \dots, G_n \vdash_{\mathbf{JL}_{\mathcal{CS}}} F \quad (3.3.27)$$

for some terms  $s_i$  and formulas  $G_i$ ,  $i = 1, \dots, n$ , such that  $s_i : G_i \in \Gamma$ .

Internalizing derivation (3.3.27) by Lemma 3.2.22, using axiomatic appropriateness of  $\mathcal{CS}$ , we would obtain a term  $t(x_1, \dots, x_n)$  with fresh variables  $x_1, \dots, x_n$  such that

$$x_1 : G_1, \dots, x_n : G_n \vdash_{\mathbf{JL}_{\mathcal{CS}}} t(x_1, \dots, x_n) : F. \quad (3.3.28)$$

Finally, the simultaneous substitution of  $s_i$  for  $x_i$  in (3.3.28), using

schematicness of  $\mathcal{CS}$ ,<sup>4</sup> would yield by Lemma 3.2.28

$$s_1 : G_1, \dots, s_n : G_n \vdash_{\mathbf{JL}_{\mathcal{CS}}} t(s_1, \dots, s_n) : F ,$$

and by the Deduction Theorem 3.2.27,

$$\mathbf{JL}_{\mathcal{CS}} \vdash s_1 : G_1 \wedge \dots \wedge s_n : G_n \rightarrow t(s_1, \dots, s_n) : F .$$

For the maximal  $\mathbf{JL}_{\mathcal{CS}}$ -consistent set  $\Gamma$ , by Lemma 2.6.2.5,

$$s_1 : G_1 \wedge \dots \wedge s_n : G_n \in \Gamma .$$

Thus, by Lemma 2.6.2.4,

$$t(s_1, \dots, s_n) : F \in \Gamma ,$$

and, by the Truth Lemma 3.3.19,

$$\mathfrak{M}, \Gamma \Vdash t(s_1, \dots, s_n) : F ,$$

in clear violation of (3.3.25). This contradiction shows that set (3.3.26)

would have to be  $\mathbf{JL}_{\mathcal{CS}}$ -consistent if (3.3.25) were true.

Further, if set (3.3.26) were  $\mathbf{JL}_{\mathcal{CS}}$ -consistent, it could then be extended

by Lemma 2.6.2.8 to a maximal  $\mathbf{JL}_{\mathcal{CS}}$ -consistent set  $\Delta_0 \supseteq \Gamma^\# \cup \{\neg F\}$ .

---

<sup>4</sup>Here we cannot allow renaming of constants in  $F$ ; therefore, axiomatic appropriateness of  $\mathcal{CS}$  alone is not sufficient.

By the definition (3.3.21) of  $R$  for canonical F-models,  $\Gamma R \Delta_0$ . But since  $\neg F \in \Delta_0$ , by the Truth Lemma 3.3.19,

$$\mathfrak{M}, \Delta_0 \not\models F ,$$

which would contradict (3.3.24). This contradiction completes the proof of the Fully Explanatory Property.

- *Anti-Monotonicity* (for  $J5_{CS}$ ,  $J45_{CS}$ ,  $JD45_{CS}$ , and  $JT45_{CS}$ ):

if  $\Gamma \notin \mathcal{A}(t, F)$  and  $\Gamma R \Delta$ , then  $\Delta \notin \mathcal{A}(t, F)$

Let  $\Gamma \notin \mathcal{A}(t, F)$  for some term  $t$ , some formula  $F$ , and some  $\Gamma \in W$ ; let  $\Gamma R \Delta$  for some  $\Delta \in W$ . By the Completeness Theorem 3.3.14, the canonical F-model for each of the four logics listed above satisfies both the Negative Introspection Closure and the Strong Evidence Property. By the former,  $\Gamma \in \mathcal{A}(?t, \neg t : F)$ . By the latter,  $\mathfrak{M}, \Gamma \Vdash ?t : \neg t : F$ . By (3.3.18),  $\mathfrak{M}, \Delta \Vdash \neg t : F$ . So  $\mathfrak{M}, \Delta \not\models t : F$  and, by Strong Evidence,  $\Delta \notin \mathcal{A}(t, F)$ .

- *Stability* (for  $J45_{CS}$ ,  $JD45_{CS}$ , and  $JT45_{CS}$ ):

if  $\Gamma R \Delta$ , then  $\Gamma \in \mathcal{A}(t, F) \iff \Delta \in \mathcal{A}(t, F)$ .

The  $\implies$  direction is equivalent to the Monotonicity condition that was proven for these three logics in Theorem 3.3.14. The  $\impliedby$  direction is

equivalent to Anti-Monotonicity demonstrated above.

- $R$  is Euclidean (for  $J5_{CS}$ ,  $J45_{CS}$ ,  $JD45_{CS}$ , and  $JT45_{CS}$ ):

Let  $\Gamma R \Delta$  and  $\Gamma R \Sigma$  for some  $\Gamma, \Delta, \Sigma \in W$ . We need to prove that  $\Delta R \Sigma$ , i.e., that  $\Delta^\# \subseteq \Sigma$ .

For any  $t: F \in \Delta$ , by the Truth Lemma 3.3.19,  $\mathfrak{M}, \Delta \Vdash t: F$ ; hence,  $\Delta \in \mathcal{A}(t, F)$ . By Anti-Monotonicity proven for these logics earlier,  $\Gamma \in \mathcal{A}(t, F)$  since  $\Gamma R \Delta$ . By Strong Evidence, proven in Completeness Theorem 3.3.14,  $\mathfrak{M}, \Gamma \Vdash t: F$ . Since  $\Gamma R \Sigma$ , by (3.3.18),  $\mathfrak{M}, \Sigma \Vdash F$  and finally, by the Truth Lemma 3.3.19,  $F \in \Sigma$ .

- $R$  is an equivalence relation (for  $JT45_{CS}$ ):

Reflexivity of  $R$  for this logic was established in Completeness Theorem 3.3.14. In addition, we have just shown that  $R$  must be Euclidean. By Lemma 2.4.2, a reflexive Euclidean binary relation is an equivalence relation.

This completes the proof of the Strong Completeness Theorem 3.3.21.  $\square$

Given that one of the foci of this research is decidability, one would expect to find some version of *Finite Model Property* (FMP), which is an even stronger version of the completeness theorem. But since the traditional for-

mulation of FMP is not sufficient for justification logics, we postpone the discussion of these stronger completeness results till Chapter 4.

### 3.3.3 M-Models vs. F-Models

It may be noted that in most cases, an M-model is nothing more than a single-world F-model. It is not coincidental that the conditions on the admissible evidence function are very similar and even bear the same name for M- and F-models. The definition (3.3.18) of  $\Vdash$  for F-models with a single reflexive world is equivalent to the definition (3.3.6) of  $\Vdash$  for M-models; similarly, (3.3.18) for an F-model with a single irreflexive world is nothing but (3.3.7). The reader is encouraged to explore the similarities further.

The completeness of justification logics (without negative introspection) w.r.t. M-models shows that the machinery of admissible evidence functions is really very strong and can often replace the whole Kripke structure of an F-model. At the same time, in many cases F-models constructed to illustrate specific epistemic situations such as Wise Men Puzzle (see [Art06]) or Gettier Examples (see [Art07]) are simpler and more elegant than their equivalent M-models.

On the other hand, being more laconic, M-models are often convenient for proofs, especially constructive proofs and proofs involving complexity. It can

be observed from the literature that there is something of a truce between the two semantics. Instead of competing, they rather complement each other.

As will be discussed in Chapter 4, Theorem 3.3.4 establishes a very strong form of the Finite Model Property for F-models: every satisfiable formula is satisfiable in a single-world model.

There are two important exceptions to this rule:

- No M-models are known for justification logics with the Negative Introspection axiom A6.

- The Consistency Axiom A7 is treated differently in the two semantics.

One of the possible explanations is that a single-world model with a serial accessibility relation is automatically reflexive, which would cause an undesirable conflation of Consistency Axiom with the Factivity Axiom A4. This prompts the transfer of the responsibilities carried out by seriality of  $R$  in F-models to the Consistent Evidence condition on  $\mathcal{A}$  in M-models.

This transfer may be the best place to showcase the relationship between the Kripke structure and the admissible evidence function apparatus. It may seem strange that completeness w.r.t. F-models for logics with axiom A7 requires an extra condition on  $\mathcal{CS}$  to be axiomatically appropriate. This

condition is, nevertheless, necessary as the following example demonstrates:

*Example 3.3.23.* Consider  $\text{JD}_0$  with the empty constant specification. We can freely use M-models for this logic by the Completeness Theorem 3.3.4. But the empty constant specification is, of course, not axiomatically appropriate. We will show that for distinct justification variables  $x$  and  $y$ , the formula  $y:x:\perp$ , although satisfiable in M-models, cannot be satisfied in any F-model for  $\text{JD}_0$ .

Showing unsatisfiability in F-models is easier. Consider any F-model  $\mathfrak{M} = (W, R, V, \mathcal{A})$  for  $\text{JD}_0$ . Consider any world  $u \in W$ . By (3.3.18), for

$$\mathfrak{M}, u \Vdash y:x:\perp$$

to hold, formula  $x:\perp$  would have to be true in all the worlds accessible from  $u$ .

At least one such world exists by seriality of  $R$ . Let  $uRw$ , for instance. In turn, by (3.3.18), for

$$\mathfrak{M}, w \Vdash x:\perp$$

to hold,  $\perp$  should be true in all the worlds accessible from  $w$ . Again, at least one such world exists by seriality, but  $\perp$  cannot be true in it by (3.3.16).

This shows that  $\neg y:x:\perp$  is valid w.r.t. F-models for  $\text{JD}_0$ . But

$$\text{JD}_0 \not\vdash \neg y:x:\perp$$

because there is an M-model where  $y : x : \perp$  is satisfied. Let for any term  $t$  and any formula  $F$

$$\mathcal{E}(t, F) \Leftrightarrow \begin{cases} True & \text{if } F = x : \perp \text{ and } t = t_1 + \dots + y + \dots + t_n \text{ ,} \\ False & \text{otherwise ,} \end{cases}$$

where  $t_1 + \dots + y + \dots + t_n$  is any sum of terms with one of the summands being  $y$  (the order of summation is unimportant). Note that  $n$  may be equal to zero, in which case the whole sum collapses to  $y$ .

Take an arbitrary M-type propositional valuation  $U$ . Then,  $\mathfrak{M} = (U, \mathcal{E})$  is an M-model for  $\text{JD}_0$ . The only thing we need to prove is the conditions on the admissible evidence function.

*CS Closure* is vacuously satisfied since  $\mathcal{CS} = \emptyset$ .

*Consistent Evidence Condition* is clearly satisfied since  $\mathcal{E}(t, F)$  only holds for  $F = x : \perp$  and never for  $F = \perp$ .

*Sum Closure* is satisfied too. Indeed, if  $\mathcal{E}(t, F)$  holds, then  $t$  is a sum containing  $y$ . Both  $t + s$  and  $s + t$  are also sums containing  $y$ ; hence,  $\mathcal{E}(t + s, F)$  and  $\mathcal{E}(s + t, F)$ .

*Application Closure* is satisfied vacuously. Indeed, there is not a single implication  $F \rightarrow G$  for which  $\mathcal{E}(t, F \rightarrow G)$  would hold. This admissible evidence function is so tiny that we never have a chance to apply

Application Closure.

We have shown that  $\mathfrak{N}$  is an M-model for  $\text{JD}_0$ . It remains to note that  $\mathcal{E}(y, x : \perp)$  holds; therefore, by (3.3.7),

$$\mathfrak{N} \Vdash y : x : \perp .$$

This contradiction shows that F-models are not adequate for  $\text{JD}_0$ . In particular, the canonical “F-model” for  $\text{JD}_0$  is not serial. Indeed, since the set  $\{y : x : \perp\}$  is  $\text{JD}_0$ -consistent, by Lemma 2.6.2.8, there exists a maximal  $\text{JD}_0$ -consistent set  $\Gamma \ni y : x : \perp$ . Unfortunately, this  $\Gamma$  is isolated in the canonical model for  $\text{JD}_0$  because  $\Gamma^\# \ni x : \perp$ . The set  $\{x : \perp\}$  is perfectly  $\text{JD}_0$ -inconsistent, so by Lemma 2.6.2.2, no maximal  $\text{JD}_0$ -consistent  $\Delta \supseteq \Gamma^\#$ .

◀

Given that traditional F-models used for JD and JD45 in [Pac05], for JD45 in [Art07], and for JD and JD4 in [Kuz08] only work for axiomatically appropriate  $\mathcal{CS}$ , it makes sense to define alternative F-models for  $\text{JD}_{\mathcal{CS}}$ ,  $\text{JD4}_{\mathcal{CS}}$ , and  $\text{JD45}_{\mathcal{CS}}$  that would work with an arbitrary  $\mathcal{CS}$ . Below we develop a variant of F-models specifically for this purpose.

**Definition 3.3.24.** Let  $\mathcal{CS}$  be a constant specification for

$$\text{JL} \in \{\text{JD}, \text{JD4}, \text{JD45}\} .$$

An **Fk-model** for  $\mathbf{JL}_{CS}$  is an F-model  $\mathfrak{M} = (W, R, V, \mathcal{A})$  for  $\mathbf{JL}_{CS}$ , except that  $R$  is not required to be serial; instead, the following Consistent Evidence condition is imposed on  $\mathcal{A}$ :

- *Consistent Evidence condition:*

$$\mathcal{A}(t, \perp) = \emptyset \quad \text{for all terms } t.$$

**Theorem 3.3.25 (Completeness Theorem for Fk-models).**  $\mathbf{JD}_{CS}$ ,  $\mathbf{JD4}_{CS}$ , and  $\mathbf{JD45}_{CS}$  are sound and complete w.r.t. their Fk-models.

*Proof.* The proof mostly repeats the proof of Theorem 3.3.14. We will only outline the differences.

In the soundness proof, the seriality of  $R$  was only used to show validity of axiom A7,  $t: \perp \rightarrow \perp$ . So for the new models, we need to reestablish validity of A7, based on the Consistent Evidence condition. Let  $\mathfrak{M} = (W, R, V, \mathcal{A})$  be an Fk-model for  $\mathbf{JL}_{CS}$ . Since for any term  $t$ ,  $\mathcal{A}(t, \perp) = \emptyset$ , we have  $w \notin \mathcal{A}(t, \perp)$  for any  $w \in W$ . Thus,  $\mathfrak{M}, w \not\vdash t: \perp$  by (3.3.18). This is the only necessary change in the soundness proof.

For completeness, we have to show that the canonical F-model  $\mathfrak{M} = (W, R, V, \mathcal{A})$  for  $\mathbf{JL}_{CS}$  from Def. 3.3.17 is an Fk-model for  $\mathbf{JL}_{CS}$ , i.e., that it additionally satisfies the Consistent Evidence Condition. For no term  $t$  is the set  $\{t: \perp\}$   $\mathbf{JL}_{CS}$ -consistent due to axiom A7. By Lemma 2.6.2.2,  $t: \perp \notin \Gamma$  for

any maximal  $\text{JL}_{CS}$ -consistent  $\Gamma \in W$ . So by definition (3.3.23),  $\Gamma \notin \mathcal{A}(t, \perp)$  for any  $\Gamma \in W$ .  $\square$

As in the Strong Completeness Theorem 3.3.21, additional conditions can be imposed on these Fk-models without losing completeness:

**Theorem 3.3.26 (Strong Completeness Theorem for Fk-models).**

1.  $\text{JD}_{CS}$  and  $\text{JD4}_{CS}$  are complete w.r.t. the class of their Fk-models that additionally satisfy
  - *Strong Evidence Property*
2.  $\text{JD45}_{CS}$  is complete w.r.t. the class of their Fk-models with Euclidean  $R$  that additionally satisfy
  - *Strong Evidence Property*
  - *Stability*
3. In addition, for any of the logics  $\text{JD}_{CS}$ ,  $\text{JD4}_{CS}$ , or  $\text{JD45}_{CS}$  with a schematic and axiomatically appropriate  $CS$ 
  - *Fully Explanatory Property*

can be added to the list of requirements.

*Proof.* The proof repeats the proof of Theorem 3.3.21.  $\square$

### 3.3.4 Minimal Evidence Functions

It is important, especially for applications, to be able to effectively construct models that satisfy particular conditions. Constructing a Kripke model in modal logic is easy: the only difficulty might be showing that the accessibility relation is reflexive, transitive, symmetric, and/or Euclidean, but we can always resort to specifying some relation with the intention of taking its reflexive, transitive, symmetric, and/or Euclidean closure.

Turning to models for justification logics, be it M- or F-models, we now have to construct an admissible evidence function, which always requires compliance with certain closure conditions. In this section, we intend to describe a general way of constructing models for justification logics along with the closure procedures necessary for creating admissible evidence functions.

**Definition 3.3.27.** Let  $Tm$  and  $Fm$  stand for the sets of all terms and all formulas respectively in language  $\mathcal{JL}$ .

An *M-type possible evidence function* is any function

$$\mathcal{B}: Tm \times Fm \rightarrow \{True, False\} .$$

Let  $W$  be a set of possible worlds. An *F-type possible evidence function on  $W \neq \emptyset$*  is any function

$$\mathcal{B}: Tm \times Fm \rightarrow 2^W .$$



*Note 3.3.28.* Unlike in the case of admissible evidence functions, we never need to know a binary relation  $R \subseteq W \times W$  to work with a possible evidence function on  $W$ . A possible evidence function does not depend on a justification logic either.

An M-type (F-type) possible evidence function has the same input and output as an admissible evidence function for M-models (F-models), but has no closure or other conditions imposed on it. Naturally, every admissible evidence function is also a possible evidence function of the respective type.

We will provide two proofs that for logics without negative introspection, any possible evidence function can be *extended* to an admissible evidence function; moreover, there is a *minimal* extension of this type. This operation is routinely needed for constructing models for specific epistemic examples as well as for proving decidability and evaluating complexity of justification logics. In this section, we will present a non-constructive proof that a minimal admissible evidence function always exists.

In Chapters 4 and 5, we will extensively use “finite” possible evidence functions to show decidability or evaluate complexity of justification logics. We will, therefore, describe an effective way to construct the minimal ad-

missible evidence function if it exists. Then, also in Chapters 4 and 5, we will use this constructive procedure in decision algorithms. There, it will be made recursive under the additional requirement for  $\mathcal{CS}$  to be recursive.

**Definition 3.3.29.** We say that an M-type possible evidence function  $\mathcal{B}_2$  is *based* on an M-type possible evidence function  $\mathcal{B}_1$  and write  $\mathcal{B}_1 \subseteq \mathcal{B}_2$  if, for any term  $t$  and any formula  $F$ , statement  $\mathcal{B}_2(t, F)$  holds whenever  $\mathcal{B}_1(t, F)$  does.

Similarly, for a given set  $W \neq \emptyset$ , we say that an F-type possible evidence function  $\mathcal{B}_2$  on  $W$  is *based* on an F-type possible evidence function  $\mathcal{B}_1$ , also on  $W$ , and write  $\mathcal{B}_1 \subseteq \mathcal{B}_2$  if  $\mathcal{B}_1(t, F) \subseteq \mathcal{B}_2(t, F)$  for any term  $t$  and any formula  $F$ . ◀

**Definition 3.3.30.** Let  $\mathcal{EF}$  be

- a class of M-type possible evidence functions      or
- a class of F-type possible evidence functions on the same set  $W \neq \emptyset$ .

A possible evidence function  $\mathcal{B} \in \mathcal{EF}$  is called the *minimal<sup>5</sup> evidence function in  $\mathcal{EF}$*  if

$$\mathcal{B} \subseteq \mathcal{B}' \quad \forall \mathcal{B}' \in \mathcal{EF} . \quad (3.3.29)$$

---

<sup>5</sup>It would, perhaps, be better to call it the *minimum evidence function*, but historically the term “minimal” has already taken root.



**Proposition 3.3.31.** *If the minimal function in a class exists, it is unique.*

**Definition 3.3.32 (Classes of admissible evidence functions).**

1. Let  $\mathcal{CS}$  be a constant specification for a justification logic

$$\text{JL} \in \{\text{J}, \text{JD}, \text{JT}, \text{J4}, \text{JD4}, \text{LP}\} .$$

Let  $\mathcal{B}$  be an M-type possible evidence function. We will denote the class of all M-type admissible for  $\text{JL}_{\mathcal{CS}}$  evidence functions by  $\mathcal{AEF}_{\mathcal{B}}(\text{JL}_{\mathcal{CS}})$ .

2. Let  $\mathcal{CS}$  be a constant specification for a justification logic

$$\text{JL} \in \{\text{J}, \text{JD}, \text{JT}\} .$$

Let  $\mathcal{B}$  be an F-type possible evidence function on  $W \neq \emptyset$ . We will denote the class of all F-type admissible for  $\text{JL}_{\mathcal{CS}}$  evidence functions on  $W$  by  $\mathcal{AEF}_{\mathcal{B}}(\text{JL}_{\mathcal{CS}}, W)$ .

3. Let  $\mathcal{CS}$  be a constant specification for a justification logic

$$\text{JL} \in \{\text{J4}, \text{JD4}, \text{LP}\} .$$

Let  $\mathcal{B}$  be an F-type possible evidence function on set  $W \neq \emptyset$  and let  $R \subseteq W \times W$  be a binary relation on  $W$  that is

- transitive for  $\text{JL} = \text{J4}$ ,
- transitive and serial for  $\text{JL} = \text{JD4}$ ,
- transitive and reflexive for  $\text{JL} = \text{LP}$ .

We will denote the class of all F-type admissible for  $\text{JL}_{\text{CS}}$  evidence functions on  $(W, R)$  by  $\mathcal{AEF}_{\mathcal{B}}(\text{JL}_{\text{CS}}, W, R)$ .

*Note 3.3.33.* Although the type of evidence functions (M or F) is not explicitly present in the  $\mathcal{AEF}$ -notation, it can be easily read from the number of arguments of  $\mathcal{AEF}_{\mathcal{B}}$ : M-type functions require only one argument, the logic, whereas F-type functions take two or three arguments depending on whether positive introspection is absent or present respectively.

**Theorem 3.3.34.**

1. Let  $\text{CS}$  be a constant specification for  $\text{JL} \in \{\text{J}, \text{JT}, \text{J4}, \text{LP}\}$ . For any M-type possible evidence function  $\mathcal{B}$ , the class

$$\mathcal{AEF}_{\mathcal{B}}(\text{JL}_{\text{CS}}) \neq \emptyset$$

and has a (unique) minimal element.

2. Let  $\text{CS}$  be a constant specification for  $\text{JL} \in \{\text{J}, \text{JD}, \text{JT}\}$ . For any F-type possible evidence function  $\mathcal{B}$  on a set  $W \neq \emptyset$ , the class

$$\mathcal{AEF}_{\mathcal{B}}(\text{JL}_{\text{CS}}, W) \neq \emptyset$$

and has a (unique) minimal element.

3. Let  $\mathcal{CS}$  be a constant specification for  $\text{JL} \in \{\text{J4}, \text{JD4}, \text{LP}\}$ . For any  $F$ -type possible evidence function  $\mathcal{B}$  on set  $W \neq \emptyset$  and any binary relation  $R \subseteq W \times W$  that is

- transitive for  $\text{J4}_{\mathcal{CS}}$ ,
- transitive and serial for  $\text{JD4}_{\mathcal{CS}}$ ,
- transitive and reflexive for  $\text{LP}_{\mathcal{CS}}$ ,

the class

$$\mathcal{AEF}_{\mathcal{B}}(\text{JL}_{\mathcal{CS}}, W, R) \neq \emptyset$$

and has a (unique) minimal element.

*Proof.*

1. The constant M-type evidence function

$$\mathcal{A}_{\text{True}}(t, F) \equiv \text{True} \quad \text{for all terms } t \text{ and formulas } F$$

is clearly admissible for  $\text{J}_{\mathcal{CS}}$ ,  $\text{JT}_{\mathcal{CS}}$ ,  $\text{J4}_{\mathcal{CS}}$ , and  $\text{LP}_{\mathcal{CS}}$  because the Application, Sum,  $\mathcal{CS}$ , and Positive Introspection Closure conditions require the admissible evidence function to be *True* in certain circumstances, but never insist on it being *False*. It is equally clear that  $\mathcal{A}_{\text{True}}$  is

based on every M-type possible evidence function imaginable. Thus,

$\mathcal{A}_{True} \in \mathcal{AEF}_{\mathcal{B}}(\mathbf{JL}_{CS})$  for any of the four logics and any  $\mathcal{B}$ .

To find the unique minimal element in  $\mathcal{AEF}_{\mathcal{B}}(\mathbf{JL}_{CS})$ , we simply take the “conjunction” of all functions from  $\mathcal{AEF}_{\mathcal{B}}(\mathbf{JL}_{CS})$ : for all terms  $t$  and formulas  $F$ ,

$$\mathcal{A}_{\min}(t, F) \Leftrightarrow \begin{cases} False & \text{if } \mathcal{A}(t, F) = False \\ & \text{for some } \mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}(\mathbf{JL}_{CS}), \\ True & \text{otherwise.} \end{cases} \quad (3.3.30)$$

Let us show that  $\mathcal{A}_{\min}$  is indeed an M-type admissible for  $\mathbf{JL}_{CS}$  evidence function, i.e., that it satisfies all the necessary closure conditions.

- *CS Closure*: For any  $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}(\mathbf{JL}_{CS})$  and any  $c : A \in \mathcal{CS}$ ,  $\mathcal{A}(c, A) = True$  and, in addition,

$$\mathcal{A}(\underbrace{!! \dots !}_n c, \underbrace{! \dots !}_{n-1} c : \dots : !! c : ! c : c : A) = True$$

for any integer  $n \geq 1$ , by *CS Closure* for the admissible evidence function  $\mathcal{A}$ . Hence, for any  $c : A \in \mathcal{CS}$ ,

$$\mathcal{A}_{\min}(c, A) = True$$

and, in addition,

$$\mathcal{A}_{\min}(\underbrace{!! \dots !}_n c, \underbrace{! \dots !}_{n-1} c : \dots : !! c : ! c : c : A) = True$$

for any integer  $n \geq 1$ .

– *Application Closure*: Let

$$\mathcal{A}_{\min}(s, F \rightarrow G) = True ,$$

$$\mathcal{A}_{\min}(t, F) = True .$$

Then, by (3.3.30), for any  $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}(\mathbf{JL}_{CS})$ ,

$$\mathcal{A}(s, F \rightarrow G) = True ,$$

$$\mathcal{A}(t, F) = True .$$

By Application Closure, for any  $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}(\mathbf{JL}_{CS})$ ,

$$\mathcal{A}(s \cdot t, G) = True .$$

Therefore, by (3.3.30),  $\mathcal{A}_{\min}(s \cdot t, G) = True$ .

– The arguments for the *Positive Introspection Closure* (for  $\mathbf{J4}_{CS}$  and  $\mathbf{LP}_{CS}$ ) and for the *Sum Closure* are similar to the one for the Application Closure above.

Thus,  $\mathcal{A}_{\min}$  is an M-type admissible for  $\mathbf{JL}_{CS}$  evidence function.

For any term  $t$  and formula  $F$  such that  $\mathcal{B}(t, F) = True$ ,

$$(\forall \mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}(\mathbf{JL}_{CS})) \quad \mathcal{A}(t, F) = True$$

since  $\mathcal{B} \subseteq \mathcal{A}$  for any such  $\mathcal{A}$ . By (3.3.30),  $\mathcal{A}_{\min}(t, F) = True$ . Thus,

$\mathcal{B} \subseteq \mathcal{A}_{\min}$ .

It remains to show that  $\mathcal{A}_{\min} \subseteq \mathcal{A}$  for every  $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}(\text{JL}_{CS})$ : this easily follows from (3.3.30).

2–3. The F-type total evidence function on  $W$

$$\mathcal{A}_W^{\text{tot}}(t, F) \equiv W \quad \text{for each term } t \text{ and each formula } F \quad (3.3.31)$$

serves as an analog of  $\mathcal{A}_{\text{True}}$  for F-models. It is clearly admissible for  $\text{J}_{CS}$ ,  $\text{JD}_{CS}$ ,  $\text{JT}_{CS}$ ,  $\text{J4}_{CS}$ ,  $\text{JD4}_{CS}$ , and  $\text{LP}_{CS}$ . The Monotonicity condition for F-models still only asks for some worlds to be included into  $\mathcal{A}(t, F)$ , but never excluded, so  $\mathcal{A}_W^{\text{tot}}$  satisfies Monotonicity independent of  $R$ . Also  $\mathcal{A}_W^{\text{tot}}$  is based on every possible evidence function on  $W$  imaginable.

Thus,

- $\mathcal{A}_W^{\text{tot}} \in \mathcal{AEF}_{\mathcal{B}}(\text{JL}_{CS}, W)$  for  $\text{J}_{CS}$ ,  $\text{JD}_{CS}$ , and  $\text{JT}_{CS}$ ;
- $\mathcal{A}_W^{\text{tot}} \in \mathcal{AEF}_{\mathcal{B}}(\text{JL}_{CS}, W, R)$  for  $\text{J4}_{CS}$ ,  $\text{JD4}_{CS}$ , and  $\text{LP}_{CS}$  for any binary relation  $R$  on  $W$ .

Let  $\mathcal{AEF}_{\mathcal{B}}$  denote either  $\mathcal{AEF}_{\mathcal{B}}(\text{JL}_{CS}, W)$  or  $\mathcal{AEF}_{\mathcal{B}}(\text{JL}_{CS}, W, R)$ , depending on  $\text{JL}_{CS}$ . To find the minimal element in  $\mathcal{AEF}_{\mathcal{B}}$ , we “intersect” all functions from it: for any term  $t$  and any formula  $F$ ,

$$\mathcal{A}_{\min}(t, F) \Leftarrow \bigcap_{\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}} \mathcal{A}(t, F) . \quad (3.3.32)$$

Let us show that  $\mathcal{A}_{\min}$  is an F-type admissible for  $\mathbf{JL}_{CS}$  evidence function on  $W$  or on  $(W, R)$  depending on  $\mathbf{JL}_{CS}$ , i.e., that  $\mathcal{A}_{\min}$  satisfies all the necessary closure conditions.

- *CS Closure*: For any  $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}$  and any  $c : A \in \mathcal{CS}$ ,  $\mathcal{A}(c, A) = W$  and, in addition,

$$\mathcal{A}(\underbrace{!! \dots !}_n c, \underbrace{! \dots !}_{n-1} c : \dots : ! c : c : A) = W$$

for any integer  $n \geq 1$ , by *CS Closure* for the admissible evidence function  $\mathcal{A}$ . Hence, by (3.3.32),  $\mathcal{A}_{\min}(c, A) = W$  and

$$\mathcal{A}_{\min}(\underbrace{!! \dots !}_n c, \underbrace{! \dots !}_{n-1} c : \dots : ! c : c : A) = W .$$

- *Application Closure*: Let  $u \in \mathcal{A}_{\min}(s, F \rightarrow G)$  and  $u \in \mathcal{A}_{\min}(t, F)$ . Then, by (3.3.32),  $u \in \mathcal{A}(s, F \rightarrow G)$  and  $u \in \mathcal{A}(t, F)$  for any  $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}$ . By *Application Closure* for any such  $\mathcal{A}$ , we have  $u \in \mathcal{A}(s \cdot t, G)$ . Therefore, by (3.3.32),  $u \in \mathcal{A}_{\min}(s \cdot t, G)$ .
- The argument for the *Positive Introspection Closure* (for  $\mathbf{J4}_{CS}$ ,  $\mathbf{JD4}_{CS}$ , and  $\mathbf{LP}_{CS}$ ) and for the *Sum Closure* is similar to the one for the *Application Closure* above.
- *Monotonicity* (for  $\mathbf{J4}_{CS}$ ,  $\mathbf{JD4}_{CS}$ , and  $\mathbf{LP}_{CS}$ ): Let  $u \in \mathcal{A}_{\min}(t, F)$  and  $uRw$ . Then, by (3.3.32),  $u \in \mathcal{A}(t, F)$  for any  $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}$ . By

Monotonicity for any such  $\mathcal{A}$ , we have  $w \in \mathcal{A}(t, F)$ . By (3.3.32),  
 $w \in \mathcal{A}_{\min}(t, F)$ .

Thus,  $\mathcal{A}_{\min}$  is an F-type admissible for  $\text{JL}_{\mathcal{CS}}$  evidence function.

$\mathcal{B}(t, F) \subseteq \mathcal{A}(t, F)$  for all  $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}$  since  $\mathcal{B} \subseteq \mathcal{A}$ . So

$$\mathcal{B}(t, F) \subseteq \bigcap_{\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}} \mathcal{A}(t, F) = \mathcal{A}_{\min}(t, F) .$$

Thus,  $\mathcal{B} \subseteq \mathcal{A}_{\min}$ .

It remains to show that  $\mathcal{A}_{\min} \subseteq \mathcal{A}$  for every  $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}$ . But this is immediate from (3.3.32).

This completes the proof of Theorem 3.3.34. □

*Note 3.3.35.* Theorem 3.3.34.1 does not hold for  $\text{JD}_{\mathcal{CS}}$  and  $\text{JD4}_{\mathcal{CS}}$ ; Theorem 3.3.34.2 does not hold for Fk-models for  $\text{JD}_{\mathcal{CS}}$ ; and Theorem 3.3.34.3 does not hold for Fk-models for  $\text{JD4}_{\mathcal{CS}}$  because of the Consistent Evidence condition. This condition requires statements  $\mathcal{A}(t, F)$  or  $w \in \mathcal{A}(t, F)$  to be false in certain cases, which may conflict with other closure conditions that could require these statements to be true. The following example illustrates such a situation:

*Example 3.3.36.* Let  $\mathcal{B}(x, p \rightarrow \perp) = \mathcal{B}(y, p) = \text{True}$ . Then, no M-type admissible for  $\text{JD}_{\mathcal{CS}}$  or  $\text{JD4}_{\mathcal{CS}}$  evidence function can be based on  $\mathcal{B}$ . Indeed,

any such function  $\mathcal{A}$ , according to Application Closure, would have  $\mathcal{A}(x \cdot y, \perp) = \text{True}$ , violating the Consistent Evidence condition.

This example shows that constructing an M- or an Fk-model with given properties for  $\text{JD}_{\mathcal{CS}}$  or  $\text{JD4}_{\mathcal{CS}}$  may not be as easy as constructing an F-model. It would, therefore, make sense to resort to F-models for axiomatically appropriate  $\mathcal{CS}$ . ◀

We will now describe minimal evidence functions axiomatically:

**Definition 3.3.37.** Let  $\mathcal{CS}$  be a constant specification for one of justification logics. The axioms and rules of \*-calculi are as follows:

$*\mathcal{CS}^!$ . Axioms  $*(c, A)$  and  $*(\underbrace{!! \dots !}_n c, \underbrace{! \dots !}_{n-1} c : \dots : !! c : ! c : c : A)$ ,  
where  $c : A \in \mathcal{CS}$  and  $n \geq 1$  is an integer.

$*\mathcal{CS}$ . Axiom  $*(c, A)$ , where  $c : A \in \mathcal{CS}$ .

$*\text{A2. Application Rule}$  
$$\frac{*(s, F \rightarrow G) \quad *(t, F)}{*(s \cdot t, G)}$$

$*\text{A3. Sum Rule}$  
$$\frac{*(s, F)}{*(s + t, F)} \qquad \frac{*(t, F)}{*(s + t, F)}$$

$*\text{A5. Positive Introspection Rule}$  
$$\frac{*(t, F)}{*(!t, t : F)}$$

◀

Table 3.3.4: \*-calculi for pure justification logics

Logic	*CS <sup>!</sup>	*CS	*A2	*A3	*A5	Calculus
J <sub>CS</sub>	✓		✓	✓		* <sub>CS</sub> -calculus
JD <sub>CS</sub>	✓		✓	✓		* <sub>CS</sub> -calculus
JT <sub>CS</sub>	✓		✓	✓		* <sub>CS</sub> -calculus
J4 <sub>CS</sub>		✓	✓	✓	✓	*! <sub>CS</sub> -calculus
JD4 <sub>CS</sub>		✓	✓	✓	✓	*! <sub>CS</sub> -calculus
LP <sub>CS</sub>		✓	✓	✓	✓	*! <sub>CS</sub> -calculus

*Note 3.3.38.* As in Note 3.2.3, axiom \*CS<sup>!</sup> is derivable from axiom \*CS and rule \*A5.

**Definition 3.3.39.** There are two types of \*-calculi used for justification logics:

- the \*<sub>CS</sub>-calculus for J<sub>CS</sub>, JD<sub>CS</sub>, and JT<sub>CS</sub> and
- the \*!<sub>CS</sub>-calculus for J4<sub>CS</sub>, JD4<sub>CS</sub>, and LP<sub>CS</sub>,

both described in Table 3.3.4.

To conveniently use the \*-calculi, we will need notation for translating between formulas, statements about evidence functions, and \*-expressions.

**Definition 3.3.40.** For an M-type possible evidence function  $\mathcal{B}$ ,

$$\mathcal{B}^* = \{*(t, F) \mid \mathcal{B}(t, F) = \text{True}\} . \quad (3.3.33)$$

For an F-type possible evidence function  $\mathcal{B}$  on  $W$  and  $w \in W$ ,

$$\mathcal{B}_w^* = \{*(t, F) \mid w \in \mathcal{B}(t, F)\} . \quad (3.3.34)$$

**Theorem 3.3.41.**

1. Let  $\mathbb{JL}_{CS} \in \{\mathbb{J}_{CS}, \mathbb{JD}_{CS}, \mathbb{JT}_{CS}\}$ . For an  $M$ -type possible evidence function  $\mathcal{B}$ , define an  $M$ -type possible evidence function  $\mathcal{A}$  so that for any term  $t$  and any formula  $F$ ,

$$*(t, F) \in \mathcal{A}^* \iff \mathcal{B}^* \vdash_{*_{CS}} *(t, F) . \quad (3.3.35)$$

Then,  $\mathcal{B} \subseteq \mathcal{A}$ . Moreover, if the class  $\mathcal{AEF}_{\mathcal{B}}(\mathbb{JL}_{CS}) \neq \emptyset$ , then  $\mathcal{A}$  is the minimal admissible evidence function in it.

2. Let  $\mathbb{JL}_{CS} \in \{\mathbb{J4}_{CS}, \mathbb{JD4}_{CS}, \mathbb{LP}_{CS}\}$ . For an  $M$ -type possible evidence function  $\mathcal{B}$ , define an  $M$ -type possible evidence function  $\mathcal{A}$  so that for any term  $t$  and any formula  $F$ ,

$$*(t, F) \in \mathcal{A}^* \iff \mathcal{B}^* \vdash_{*!_{CS}} *(t, F) . \quad (3.3.36)$$

Then,  $\mathcal{B} \subseteq \mathcal{A}$ . Moreover, if the class  $\mathcal{AEF}_{\mathcal{B}}(\mathbb{JL}_{CS}) \neq \emptyset$ , then  $\mathcal{A}$  is the minimal admissible evidence function in it.

3. Let  $\mathbb{JL}_{CS} \in \{\mathbb{J}_{CS}, \mathbb{JD}_{CS}, \mathbb{JT}_{CS}\}$ . For an  $F$ -type possible evidence function  $\mathcal{B}$  on  $W \neq \emptyset$ , define an  $F$ -type possible evidence function  $\mathcal{A}$  on  $W$  so that for any term  $t$ , any formula  $F$ , and any  $w \in W$ ,

$$*(t, F) \in \mathcal{A}_w^* \iff \mathcal{B}_w^* \vdash_{*_{CS}} *(t, F) . \quad (3.3.37)$$

Then,  $\mathcal{B} \subseteq \mathcal{A}$ . Moreover,  $\mathcal{A}$  is the minimal admissible evidence function in the class  $\mathcal{AEF}_{\mathcal{B}}(\mathbf{JL}_{CS}, W)$ .

4. Let  $\mathbf{JL}_{CS} \in \{\mathbf{J4}_{CS}, \mathbf{JD4}_{CS}, \mathbf{LP}_{CS}\}$  and  $R$  be a binary relation on  $W$  that is

- transitive for  $\mathbf{J4}_{CS}$ ,
- transitive and serial for  $\mathbf{JD4}_{CS}$ ,
- transitive and reflexive for  $\mathbf{LP}_{CS}$ .

For an  $F$ -type possible evidence function  $\mathcal{B}$  on  $W \neq \emptyset$ , define an  $F$ -type possible evidence function  $\mathcal{A}$  on  $W$  so that for any term  $t$ , any formula  $F$ , and any  $w \in W$ ,

$$*(t, F) \in \mathcal{A}_w^* \iff \mathcal{B}_w^* \cup \bigcup_{uRw} \mathcal{B}_u^* \vdash_{*!_{CS}} *(t, F) . \quad (3.3.38)$$

Then,  $\mathcal{B} \subseteq \mathcal{A}$ . Moreover,  $\mathcal{A}$  is the minimal admissible evidence function in the class  $\mathcal{AEF}_{\mathcal{B}}(\mathbf{JL}_{CS}, W, R)$ .

*Proof.* We will use  $\mathcal{AEF}_{\mathcal{B}}$  for any of the classes of admissible evidence functions based on  $\mathcal{B}$  whenever safe. Essentially, we need to prove three things:

- $\mathcal{B} \subseteq \mathcal{A}$ ,
- $\mathcal{A} \subseteq \mathcal{E}$  for any  $\mathcal{E} \in \mathcal{AEF}_{\mathcal{B}}$ ,

- $\mathcal{A}$  is admissible.

We will prove them one by one. Throughout the proof, we will keep statements concerning the M-type functions in the left column and statements about F-type functions in the right column.  $\vdash$  will stand for either  $\vdash_{*_{CS}}$  or  $\vdash_{*!_{CS}}$ .

Let us start with  $\mathcal{B} \subseteq \mathcal{A}$ .

Suppose $\mathcal{B}(t, F) = True$ . Then, $*(t, F) \in \mathcal{B}^*$ . So $\mathcal{B}^* \vdash *(t, F)$ . Hence, $*(t, F) \in \mathcal{A}^*$ , i.e., $\mathcal{A}(t, F) = True$ .	Suppose $w \in \mathcal{B}(t, F)$ . Then, $*(t, F) \in \mathcal{B}_w^*$ . So $\mathcal{B}_w^* \vdash *(t, F)$ . Hence, $*(t, F) \in \mathcal{A}_w^*$ , i.e., $w \in \mathcal{A}(t, F)$ .
--	--

This completes the proof that  $\mathcal{B} \subseteq \mathcal{A}$ .

$\mathcal{A} \subseteq \mathcal{E}$  for any  $\mathcal{E} \in \mathcal{AEF}_B$  because the derivation rules in  $*$ -calculi are nothing but reworded closure conditions on evidence functions. The proof by induction on the  $\vdash$ -derivation can be found in Fig. 3.3.1 on p. 108.

Finally, the main part of the proof that  $\mathcal{A}$  itself is an admissible evidence function, namely, that it satisfies all the proper closure conditions, can be found in Fig. 3.3.2 on p. 109. In the figure, ‘...’ is used to denote hypotheses in a  $\vdash$ -derivation in the cases where the hypotheses are not changed by this step.

The only condition not verified in Fig. 3.3.2 is the Consistent Evidence condition for the M-type functions for  $JD_{CS}$  (Clause 1) and  $JD4_{CS}$  (Clause 2).

Figure 3.3.1: Theorem 3.3.41: Proof that  $\mathcal{A} \subseteq \mathcal{E}$  for any  $\mathcal{E} \in \mathcal{AEF}_{\mathcal{B}}$

$*\mathcal{CS}^!$	$\vdash *(\underbrace{!!\dots!}_n c, \dots! c : c : A),$ where $c : A \in \mathcal{CS}$ and $n \geq 0$ is an integer. But $\mathcal{E}(\underbrace{!!\dots!}_n c, \dots! c : c : A) = \text{True}$ by the $\mathcal{CS}^!$ Closure. So $*(\underbrace{!!\dots!}_n c, \dots! c : c : A) \in \mathcal{E}^*$ .	$\vdash *(\underbrace{!!\dots!}_n c, \dots! c : c : A),$ where $c : A \in \mathcal{CS}$ and $n \geq 0$ is an integer. But $\mathcal{E}(\underbrace{!!\dots!}_n c, \dots! c : c : A) = W$ by the $\mathcal{CS}^!$ Closure. So $*(\underbrace{!!\dots!}_n c, \dots! c : c : A) \in \mathcal{E}_w^*$ for any $w \in W$ .
$*\mathcal{CS}$	is an instance of $*\mathcal{CS}^!$ .	
Hyp	Let $*(t, F) \in \mathcal{B}^*$ . Then, $\mathcal{B}(t, F) = \text{True}$ . Since $\mathcal{B} \subseteq \mathcal{E}$ , $\mathcal{E}(t, F) = \text{True}$ . So $*(t, F) \in \mathcal{E}^*$ .	Let $*(t, F) \in \mathcal{B}_w^*$ . Then, $w \in \mathcal{B}(t, F)$ . Since $\mathcal{B} \subseteq \mathcal{E}$ , $w \in \mathcal{E}(t, F)$ . So $*(t, F) \in \mathcal{E}_w^*$ .
Hyp in Clause 4		Let $*(t, F) \in \mathcal{B}_u^*$ for $uRw$ . Then, $u \in \mathcal{B}(t, F)$ . Since $\mathcal{B} \subseteq \mathcal{E}$ , $u \in \mathcal{E}(t, F)$ . $w \in \mathcal{E}(t, F)$ by Monotonicity of $\mathcal{E}$ . So $*(t, F) \in \mathcal{E}_w^*$ .
*A2	By IH, $*(s_1, G \rightarrow F) \in \mathcal{E}^*$ and $*(s_2, G) \in \mathcal{E}^*$ . Then, $\mathcal{E}(s_1, G \rightarrow F) = \text{True}$ and $\mathcal{E}(s_2, G) = \text{True}$ . Hence, by Application Closure, $\mathcal{E}(s_1 \cdot s_2, F) = \text{True}$ . So $*(s_1 \cdot s_2, F) \in \mathcal{E}^*$ .	By IH, $*(s_1, G \rightarrow F) \in \mathcal{E}_w^*$ and $*(s_2, G) \in \mathcal{E}_w^*$ . Then, $w \in \mathcal{E}(s_1, G \rightarrow F)$ and $w \in \mathcal{E}(s_2, G)$ . Hence, by Application Closure, $w \in \mathcal{E}(s_1 \cdot s_2, F)$ . So $*(s_1 \cdot s_2, F) \in \mathcal{E}_w^*$ .
*A3	is similar to *A2.	
*A5	is similar to *A2 (used only for $\mathcal{J4}_{CS}$ , $\mathcal{JD4}_{CS}$ , and $\mathcal{LP}_{CS}$ ).	

Figure 3.3.2: Theorem 3.3.41: Proof that  $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}$  (main part)

Appl. clos.	<p>Let <math>\mathcal{A}(s, F \rightarrow G) = True</math> and <math>\mathcal{A}(t, F) = True</math>. Then, <math>\dots \vdash *(s, F \rightarrow G)</math> and <math>\dots \vdash *(t, F)</math>. Hence, <math>\dots \vdash *(s \cdot t, G)</math> by *A2. So <math>\mathcal{A}(s \cdot t, G) = True</math>.</p>	<p>Let <math>w \in \mathcal{A}(s, F \rightarrow G)</math> and <math>w \in \mathcal{A}(t, F)</math>. Then, <math>\dots \vdash *(s, F \rightarrow G)</math> and <math>\dots \vdash *(t, F)</math>. Hence, <math>\dots \vdash *(s \cdot t, G)</math> by *A2. So <math>w \in \mathcal{A}(s \cdot t, G)</math>.</p>
Sum Closure is similar to Application Closure.		
Positive Introspection Closure is similar to Application Closure (used only for $\mathbf{J4}_{CS}$ , $\mathbf{JD4}_{CS}$ , and $\mathbf{LP}_{CS}$ ).		
$\mathcal{CS}$ clos.	<p>Let <math>c: A \in \mathcal{CS}</math> and <math>n \geq 0</math>. By <math>*\mathcal{CS}^!</math> (or <math>*\mathcal{CS}</math> and *A5), <math>\vdash *(\underbrace{!! \dots !}_n c, \dots : !c: c: A)</math>. Thus, <math>\mathcal{A}(\underbrace{!! \dots !}_n c, \dots : !c: c: A) = True</math>.</p>	<p>Let <math>c: A \in \mathcal{CS}</math> and <math>n \geq 0</math>. By <math>*\mathcal{CS}^!</math> (or <math>*\mathcal{CS}</math> and *A5), <math>\vdash *(\underbrace{!! \dots !}_n c, \dots : !c: c: A)</math>. Thus, <math>\mathcal{A}(\underbrace{!! \dots !}_n c, \dots : !c: c: A) = W</math>.</p>
Monot. cond.		<p>Let <math>uRw</math> and <math>u \in \mathcal{A}(t, F)</math>. Then, <math>\mathcal{B}_u^* \cup \bigcup_{zRu} \mathcal{B}_z^* \vdash_{*\mathcal{CS}} *(t, F)</math>. <math>zRu</math> implies <math>zRw</math> since <math>R</math> is transitive. So <math>\mathcal{B}_u^* \cup \bigcup_{zRu} \mathcal{B}_z^* \subseteq \mathcal{B}_w^* \cup \bigcup_{zRw} \mathcal{B}_z^*</math>. Thus, <math>\mathcal{B}_w^* \cup \bigcup_{zRw} \mathcal{B}_z^* \vdash_{*\mathcal{CS}} *(t, F)</math>. Therefore, <math>w \in \mathcal{A}(t, F)</math> (used only for <math>\mathbf{J4}_{CS}</math>, <math>\mathbf{JD4}_{CS}</math>, and <math>\mathbf{LP}_{CS}</math>).</p>

This is how we deal with the potential problem described in Note 3.3.35.

Let  $\mathbb{JL}_{CS} \in \{\mathbb{JD}_{CS}, \mathbb{JD4}_{CS}\}$ . Let us assume that  $\mathcal{AEF}_{\mathcal{B}}(\mathbb{JL}_{CS}) \neq \emptyset$ .<sup>6</sup> Let  $\mathcal{E} \in \mathcal{AEF}_{\mathcal{B}}(\mathbb{JL}_{CS})$ . By the Consistent Evidence condition for  $\mathcal{E}$ , for every term  $t$ ,  $\mathcal{E}(t, \perp) = \text{False}$ . Since we proved  $\mathcal{A} \subseteq \mathcal{E}$ , it follows that for every term  $t$ ,  $\mathcal{A}(t, \perp) = \text{False}$ . Thus, the last of the conditions on  $\mathcal{A}$  is satisfied, and the proof of Theorem 3.3.41 is complete.  $\square$

Combining the results of Theorems 3.3.34 and 3.3.41 we conclude that

**Corollary 3.3.42.**

1. *For any logic*

$$\mathbb{JL}_{CS} \in \{\mathbb{J}_{CS}, \mathbb{JD}_{CS}, \mathbb{JT}_{CS}, \mathbb{J4}_{CS}, \mathbb{JD4}_{CS}, \mathbb{LP}_{CS}\} ,$$

*any F-type possible evidence function  $\mathcal{B}$  on  $W \neq \emptyset$ , and any suitable binary relation  $R \subseteq W \times W$ , there exists a unique F-type minimal admissible for  $\mathbb{JL}_{CS}$  evidence function on  $W$  or on  $(W, R)$  based on  $\mathcal{B}$ , defined according to*

- (3.3.37) for  $\mathbb{JL}_{CS} \in \{\mathbb{J}_{CS}, \mathbb{JD}_{CS}, \mathbb{JT}_{CS}\}$  or
- (3.3.38) for  $\mathbb{JL}_{CS} \in \{\mathbb{J4}_{CS}, \mathbb{JD4}_{CS}, \mathbb{LP}_{CS}\}$ .

---

<sup>6</sup>This is the only place in the proof of Theorem 3.3.41 where this assumption is used.

2. For any logic

$$\mathbb{JL}_{CS} \in \{\mathbb{J}_{CS}, \mathbb{JT}_{CS}, \mathbb{J4}_{CS}, \mathbb{LP}_{CS}\}$$

and any  $M$ -type possible evidence function  $\mathcal{B}$ , there exists a unique  $M$ -type minimal admissible for  $\mathbb{JL}_{CS}$  evidence function based on  $\mathcal{B}$ , defined by

- (3.3.35) for  $\mathbb{JL}_{CS} \in \{\mathbb{J}_{CS}, \mathbb{JT}_{CS}\}$  or
- (3.3.36) for  $\mathbb{JL}_{CS} \in \{\mathbb{J4}_{CS}, \mathbb{LP}_{CS}\}$ .

**Minimal functions do not work for negative introspection.**

Unfortunately, the apparatus of minimal functions breaks down in the presence of negative introspection, which is a major obstacle in proving decidability. Minimal functions are the main tool in building countermodels constructively. So far, no similar tool has been found for logics  $\mathbb{J5}_{CS}$ ,  $\mathbb{J45}_{CS}$ ,  $\mathbb{JD45}_{CS}$ , and  $\mathbb{JT45}_{CS}$ ; thus, the only robust model we have for them is the canonical model.

The Strong Evidence Property is one source of trouble. Consider an admissible evidence function  $\mathcal{A}$  on  $(W, R)$  for  $\mathbb{JT45}_{CS}$ . It is still not trivial to construct a full F-model. Whenever  $u \in \mathcal{A}(t, F)$ , by Strong Evidence,  $u \Vdash t:F$  must be guaranteed; the latter depends on the truth value of  $F$  in

all the worlds accessible from  $u$ . It is not immediately clear how to define a propositional valuation  $V$  to comply with this requirement or even how to determine whether such  $V$  exists.

But usually, instead of a complete admissible evidence function  $\mathcal{A}$  we are given some conditions it should satisfy, most commonly in the form of a possible evidence function that  $\mathcal{A}$  should be based on. It is equally unclear how to construct  $\mathcal{A}$  in this case. It is true that the total function  $\mathcal{A}_{\text{tot}}$  from the proof of Theorem 3.3.34 satisfies all the closure conditions, including the Negative Introspection Closure. But it assigns evidence terms to too many formulas in too many worlds, notably to  $\perp$ , which can never be true, a clear violation of the Strong Evidence Property.

Minimality seems to be the answer. Unfortunately, there is no such thing as a minimal function satisfying the Negative Introspection Closure, as the following example shows:

*Example 3.3.43.* Consider, for instance, the simplest case of  $\mathbf{J5}_0$  and the empty F-type possible evidence function  $\mathcal{B}_\emptyset$  on  $\{w\}$ :

$$\mathcal{B}_\emptyset(t, F) \equiv \emptyset \quad \text{for all terms } t \text{ and formulas } F.$$

As a reminder,  $\mathbf{J5}_0$  has  $\mathcal{CS} = \emptyset$ . Let us try to construct an F-type admissible for  $\mathbf{J5}_0$  evidence function on  $\{w\}$ .

Since for a justification variable  $x$  and a sentence letter  $p$ , both sets

$$\{x:p, \quad \neg?x:\neg x:p\} \quad \text{and} \quad \{\neg x:p\}$$

are  $\mathbf{J5}_0$ -consistent,<sup>7</sup> by Lemma 2.6.2.8, there must exist maximal  $\mathbf{J5}_0$ -consistent sets

$$\Gamma \supset \{x:p, \quad \neg?x:\neg x:p\} \quad \text{and} \quad \Delta \ni \neg x:p$$

in the canonical F-model  $\mathfrak{M}_{\text{can}} = (W_{\text{can}}, R_{\text{can}}, V_{\text{can}}, \mathcal{A}_{\text{can}})$  for  $\mathbf{J5}_0$ . Consider two admissible for  $\mathbf{J5}_0$  evidence functions on  $\{w\}$  obtained by restricting  $\mathcal{A}_{\text{can}}$  to  $\Gamma$  and  $\Delta$  respectively:

$$\begin{aligned} w \in \mathcal{A}_{\Gamma}(t, F) &\iff \Gamma \in \mathcal{A}_{\text{can}}(t, F) \\ w \in \mathcal{A}_{\Delta}(t, F) &\iff \Delta \in \mathcal{A}_{\text{can}}(t, F) \end{aligned}$$

Note that we are not building a model on  $\{w\}$ : we have not even defined  $V$  or  $R$ . Our goal is to show that there can be no minimal admissible for  $\mathbf{J5}_0$  evidence function on  $\{w\}$ .

It should be clear that both  $\mathcal{A}_{\Gamma}$  and  $\mathcal{A}_{\Delta}$  satisfy all the closure conditions. Indeed,  $\mathcal{A}_{\text{can}}$  does satisfy them and all conditions for  $\mathbf{J5}_0$  are local, i.e., operate wholly within each world.

---

<sup>7</sup>Their consistency can be shown the same way the consistency of  $\mathbf{J5}_0$  was proven in Theorem 3.2.21, i.e., by using the forgetful projection.

Now  $x : p \in \Gamma$ ; hence,  $\Gamma \in \mathcal{A}_{\text{can}}(x, p)$  by definition (3.3.23) of  $\mathcal{A}_{\text{can}}$ . Therefore,  $w \in \mathcal{A}_{\Gamma}(x, p)$ . At the same time,  $?x : \neg x : p \notin \Gamma$  by Lemma 2.6.2.3, so  $\Gamma \notin \mathcal{A}_{\text{can}}(?x, \neg x : p)$  by (3.3.23). Thus,  $w \notin \mathcal{A}_{\Gamma}(?x, \neg x : p)$ .

Similarly,  $w \notin \mathcal{A}_{\Delta}(x, p)$  because  $\neg x : p \in \Delta$ . Since

$$\mathbf{J5}_0 \vdash \neg x : p \rightarrow ?x : \neg x : p ,$$

$?x : \neg x : p \in \Delta$  by Lemma 2.6.2.6 and Lemma 2.6.2.4, which again implies that  $w \in \mathcal{A}_{\Delta}(?x, \neg x : p)$ .

To summarize,

$$\begin{aligned} w \in \mathcal{A}_{\Gamma}(x, p) , & \quad w \notin \mathcal{A}_{\Gamma}(?x, \neg x : p) , \\ w \notin \mathcal{A}_{\Delta}(x, p) , & \quad w \in \mathcal{A}_{\Delta}(?x, \neg x : p) . \end{aligned}$$

So these functions are clearly incomparable:  $\mathcal{A}_{\Gamma} \not\subseteq \mathcal{A}_{\Delta}$  and  $\mathcal{A}_{\Delta} \not\subseteq \mathcal{A}_{\Gamma}$ . But there can be no smaller admissible for  $\mathbf{J5}_0$  evidence function  $\mathcal{A}$  on  $\{w\}$  such that  $\mathcal{A} \subseteq \mathcal{A}_{\Gamma}$  and  $\mathcal{A} \subseteq \mathcal{A}_{\Delta}$ . Such  $\mathcal{A}$  would have to satisfy

$$w \notin \mathcal{A}(x, p) , \quad w \notin \mathcal{A}(?x, \neg x : p) ,$$

which contradicts the Negative Introspection Closure. ◀

### 3.3.5 Historical Survey

M-models were originally developed by Alexey Mkrtychev in [Mkr97] for  $\text{LP}_{CS}$ . More precisely, Mkrtychev defined there two types of models, proved their

equivalency, and showed soundness and completeness w.r.t. them. The models presented in Def. 3.3.1 correspond to what he called *pre-models*.

Later these models were generalized in [Kuz00] to  $J_{CS}$ ,  $JD_{CS}$ ,  $JT_{CS}$ ,  $J4_{CS}$ , and  $JD4_{CS}$ , and a soundness and completeness proof was provided.

The term *evidence function* was introduced by Melvin Fitting in [Fit03b] for what is now called F-models (see Sect. 3.3.2). Mkrtychev originally used the term *proof-theorem assignment*. The definition of an admissible evidence function we used here originates from Sergei Artemov (see [Art07]).

F-models were first developed for  $LP_{CS}$  by Fitting in [Fit03b] (see also [Fit05]), where he showed soundness and completeness of  $LP_{CS}$  w.r.t. them. More precisely, he introduced two types of models and showed soundness and completeness w.r.t. both semantics. F-models are what Fitting called *weak models*.

*Fully Explanatory Property* (see Theorem 3.3.21) was introduced by Fitting in [Fit05] as an additional condition for his *strong models* for  $LP_{CS}$ .

In addition, in [Fit05], Fitting also considered F-models for J, JT, and J4.

In two independent works [Pac05] and [Rub06a], presented almost simultaneously, Eric Pacuit and Natalia Rubtsova suggested very similar formulations of F-models for JT45. Pacuit, in addition, developed F-models for J5 and JD45. Soundness and completeness proofs for J, JD, and J5 can be

found in [Pac05]. It was also noted there that a combination of these results with Fitting's technique from [Fit05] would yield soundness and completeness results for JD45 and JT45. A direct proof for JT45 can also be found in [Rub06b].

Sergei Artemov in [Art07] systematized and unified the existing results, streamlined models for logics with negative introspection, and introduced F-models for J45. There, *Stability* (see Theorem 3.3.21) and *Strong Evidence Property* were first formulated; full soundness and completeness proofs for JT and J4 first appeared there.

The F-models for JD4 were, perhaps, first explicitly formulated in [Kuz08].

It should be noted that both Pacuit's F-models for logics J5<sub>CS</sub>, JD45<sub>CS</sub>, and JT45<sub>CS</sub> and Rubtsova's F-models for JT45<sub>CS</sub> differed from the ones presented in Def. 3.3.6, which follows [Art07]. Instead of the Strong Evidence Property, Pacuit used *Anti-Monotonicity* (see Theorem 3.3.21) in conjunction with the requirement for  $R$  to be Euclidean. Rubtsova, while using a property easily equivalent to the Strong Evidence, required that  $R$  be an equivalence relation on  $W$ , i.e., reflexive, transitive, and symmetric, whereas in Def. 3.3.6 it is only required to be reflexive and transitive.

As Artemov showed in [Art07], these formulations are equivalent to the one given here (see Theorem 3.3.21).

The new notation we used for the  $*$ -calculus is an homage to Mkrtychev, who was the first to use the machinery of minimal functions for his symbolic models of  $\text{LP}_{\mathcal{CS}}$  in [Mkr97]. He used  $*$  in place of  $\mathcal{A}$ .

### 3.4 Reflected Fragments of Pure Justification Logics

**Definition 3.4.1.** For a justification logic  $\text{JL}_{\mathcal{CS}}$ , its *reflected fragment*  $\text{rJL}_{\mathcal{CS}}$  is defined as

$$\text{rJL}_{\mathcal{CS}} = \{t:F \mid \text{JL}_{\mathcal{CS}} \vdash t:F\} . \quad (3.4.1)$$



The study of reflected fragments was initiated by Nikolai Krupski in [Kru03] (see also [Kru06a, Kru06b]), who found an axiomatization for  $\text{rLP}_{\mathcal{CS}}$  with an arbitrary constant specification  $\mathcal{CS}$ .

The reflected fragments of justification logics happen to have a rather elegant axiomatization of their own that resembles the closure conditions on admissible evidence functions, which will be actively exploited in future decidability and complexity proofs.

**Theorem 3.4.2.**

1. The reflected fragment  $\text{rJL}_{\mathcal{CS}}$  of  $\text{JL}_{\mathcal{CS}} \in \{\text{J}_{\mathcal{CS}}, \text{JD}_{\mathcal{CS}}, \text{JT}_{\mathcal{CS}}\}$  is axiomatized

by the  $*_{CS}$ -calculus:

$$\mathbf{rJL}_{CS} \vdash t:F \iff \mathbf{JL}_{CS} \vdash t:F \iff *_{CS}\text{-calculus} \vdash *(t, F) .$$

2. The reflected fragment  $\mathbf{rJL}_{CS}$  of  $\mathbf{JL}_{CS} \in \{\mathbf{J4}_{CS}, \mathbf{JD4}_{CS}, \mathbf{LP}_{CS}\}$  is axiomatized by the  $*!_{CS}$ -calculus:

$$\mathbf{rJL}_{CS} \vdash t:F \iff \mathbf{JL}_{CS} \vdash t:F \iff *!_{CS}\text{-calculus} \vdash *(t, F) .$$

**Corollary 3.4.3.** *If  $CS$  can serve as a constant specification for justification logics  $\mathbf{JL}_{CS}$  and  $\mathbf{JL}'_{CS}$ , either both from Clause 1 or both from Clause 2 of Theorem 3.4.2, then*

$$\mathbf{rJL}_{CS} = \mathbf{rJL}'_{CS} .$$

*Note 3.4.4.* Cor. 3.4.3 by no means implies that, for instance,  $\mathbf{rLP} = \mathbf{rJ4}$  or that  $\mathbf{rJ} = \mathbf{rJD}$ . Each of these logics uses its respective total constant specification:  $\mathcal{TCS}_{LP}$ ,  $\mathcal{TCS}_{J4}$ ,  $\mathcal{TCS}_J$ , or  $\mathcal{TCS}_{JD}$ . Thus,  $\mathcal{TCS}_{LP} \supsetneq \mathcal{TCS}_{J4}$ , whereas  $\mathcal{TCS}_J \subsetneq \mathcal{TCS}_{JD}$ .

*Proof of Theorem 3.4.2.* The left equivalence in both clauses is by Def. 3.4.1 of the reflected fragment.

The  $\Leftarrow$  direction of the right equivalence is easily proven by induction on the derivation in the respective  $*$ -calculus (we will use  $\vdash_*$  to denote derivations in either  $*$ -calculus whenever safe):

$*\mathcal{CS}^!$ . (Clause 1.) For each  $c:A \in \mathcal{CS}$  and each integer  $n \geq 0$ , by  $R4_{\mathcal{CS}}^!$ ,

$$\mathbb{J}\mathcal{L}_{\mathcal{CS}} \vdash \underbrace{!! \dots !}_n c : \dots : !c : c : A .$$

$*\mathcal{CS}$ . (Clause 2.) For each  $c:A \in \mathcal{CS}$ , by  $R4_{\mathcal{CS}}$ ,

$$\mathbb{J}\mathcal{L}_{\mathcal{CS}} \vdash c : A .$$

$*A2$ . Let

$$\vdash_* *(s, H \rightarrow G) \quad \text{and} \quad \vdash_* *(s', H) .$$

By IH,

$$\mathbb{J}\mathcal{L}_{\mathcal{CS}} \vdash s : (H \rightarrow G) \quad \text{and} \quad \mathbb{J}\mathcal{L}_{\mathcal{CS}} \vdash s' : H .$$

By A2,

$$\mathbb{J}\mathcal{L}_{\mathcal{CS}} \vdash s : (H \rightarrow G) \rightarrow (s' : H \rightarrow s \cdot s' : G) .$$

So, using *modus ponens* twice, we get

$$\mathbb{J}\mathcal{L}_{\mathcal{CS}} \vdash s \cdot s' : G .$$

Rules \*A3 and \*A5, the latter being necessary only in Clause 2, are similar to rule \*A2.

It now remains to demonstrate the  $\implies$  direction of the right equivalence.

Let  $\text{JL}_{\mathcal{CS}} \vdash t:F$ . Suppose towards a contradiction that  $\not\vdash_* *(t, F)$  for the respective \*-calculus. Consider  $W = \{w\}$  and  $R = \{(w, w)\}$ . Let  $B_\emptyset \equiv \emptyset$  be the empty possible evidence function on  $W$ . By Cor. 3.3.42.1, the minimal function exists in the class

- $\mathcal{AEF}_{\mathcal{B}_\emptyset}(\text{JL}_{\mathcal{CS}}, W)$ , defined by (3.3.37) for Clause 1;
- $\mathcal{AEF}_{\mathcal{B}_\emptyset}(\text{JL}_{\mathcal{CS}}, W, R)$ , defined by (3.3.38) for Clause 2.

Let us denote this minimal admissible evidence function by  $\mathcal{A}$ .

Note that the chosen  $R$  is reflexive, serial, and transitive. Moreover, for this  $R$  and for  $\mathcal{B}_\emptyset$ , (3.3.37) turns into

$$w \in \mathcal{A}(t, F) \iff \vdash_{*\mathcal{CS}} *(t, F) ,$$

whereas (3.3.38) becomes

$$w \in \mathcal{A}(t, F) \iff \vdash_{*!\mathcal{CS}} *(t, F) .$$

In either case, we assume the right side to be false, which would require the common left side to be false too, i.e.,  $w \notin \mathcal{A}(t, F)$ . Let us choose an

arbitrary propositional valuation  $V$  to get an F-model  $\mathfrak{M} = (W, R, V, \mathcal{A})$ . Since all conditions on  $R$  and  $\mathcal{A}$  are satisfied,  $\mathfrak{M}$  is an F-model for  $\text{JL}_{\mathcal{CS}}$ .

In this model,  $\mathfrak{M}, w \not\vdash t : F$  since  $w \notin \mathcal{A}(t, F)$ . By soundness of  $\text{JL}_{\mathcal{CS}}$ ,<sup>8</sup> this would contradict the initial assumption that  $\text{JL}_{\mathcal{CS}} \vdash t : F$ . This contradiction completes the proof of the  $\implies$  direction and of Theorem 3.4.2.  $\square$

It would seem that Theorem 3.4.2 can be easily extended to derivations from hypotheses, but the situation is not so simple. Imagine a set of formulas  $\Gamma = \{s_i : G_i \mid i = 1, \dots, n\}$  that is  $\text{JL}_{\mathcal{CS}}$ -inconsistent. Then, using classical propositional logic,  $\Gamma \vdash_{\text{JL}_{\mathcal{CS}}} t : F$  for any  $t : F$ . But it may not be that  $\Gamma \vdash_{r\text{JL}_{\mathcal{CS}}} t : F$  in the absence of the basic level of propositional reasoning as the following example shows:

*Example 3.4.5.* The set  $\{x : \perp\}$  is  $\text{JT}$ -inconsistent for any justification variable  $x$ . Indeed,  $\text{JT} \vdash x : \perp \rightarrow \perp$  is an instance of axiom A4. Hence,

$$x : \perp \vdash_{\text{JT}} \perp$$

and, more generally, for any formula  $F$ ,

$$x : \perp \vdash_{\text{JT}} F$$

---

<sup>8</sup> $\text{JL}_{\mathcal{CS}}$  is sound w.r.t. F-models even if  $\mathcal{CS}$  is not axiomatically appropriate (see Note 3.3.15).

In particular, for a sentence letter  $p$ ,

$$x : \perp \vdash_{\text{JT}} x : p .$$

At the same time,

$$*(x, \perp) \not\vdash_{*_{\text{TCSTJ}}} *(x, p) .$$



Inconsistency of  $\Gamma$  may be sufficient but is certainly not necessary to break the equivalence:

*Example 3.4.6.* Since by Factivity Axiom A4,  $\text{JT}_0 \vdash y : x : p \rightarrow x : p$  for distinct justification variables  $x$  and  $y$  and for a sentence letter  $p$ ,

$$y : x : p \vdash_{\text{JT}_0} x : p .$$

But it is clear that

$$*(y, x : p) \not\vdash_{*_{\emptyset}} *(x, p) .$$



Nevertheless, one direction does hold for derivations from hypotheses:

**Definition 3.4.7.** For a set  $X$  of  $*$ -expressions of type  $*(t, F)$ ,

$$X^\cdot = \{t : F \mid *(t, F) \in X\} .$$



**Lemma 3.4.8.** *Let  $X$  be a set of  $*$ -expressions.*

1. For  $\mathbb{JL}_{CS} \in \{\mathbb{J}_{CS}, \mathbb{JD}_{CS}, \mathbb{JT}_{CS}\}$

$$X \vdash_{*\mathcal{CS}} *(t, F) \quad \Longrightarrow \quad X^\cdot \vdash_{\mathbb{JL}_{CS}} t:F .$$

2. For  $\mathbb{JL}_{CS} \in \{\mathbb{J4}_{CS}, \mathbb{JD4}_{CS}, \mathbb{LP}_{CS}\}$

$$X \vdash_{*!\mathcal{CS}} *(t, F) \quad \Longrightarrow \quad X^\cdot \vdash_{\mathbb{JL}_{CS}} t:F .$$

*Proof.* Proof by induction on the  $*$ -derivation.

$*\mathcal{CS}^!$ . (Clause 1.) For any  $c:A \in \mathcal{CS}$  and any integer  $n \geq 0$ , by  $R4_{CS}^!$ ,

$$\mathbb{JL}_{CS} \vdash \underbrace{!! \dots !}_n c : \dots !! c : ! c : c : A .$$

$*\mathcal{CS}$ . (Clause 2.) For any  $c:A \in \mathcal{CS}$ , by  $R4_{CS}$ ,  $\mathbb{JL}_{CS} \vdash c:A$ .

Hyp. If  $*(t, F) \in X$ , then  $t:F \in X^\cdot$ , so  $X^\cdot \vdash_{\mathbb{JL}_{CS}} t:F$ .

$*A2$ . *Application Rule* 
$$\frac{*(s_1, G \rightarrow F) \quad *(s_2, G)}{*(s_1 \cdot s_2, F)}$$

By IH,  $X^\cdot \vdash_{\mathbb{JL}_{CS}} s_1 : (G \rightarrow F)$  and  $X^\cdot \vdash_{\mathbb{JL}_{CS}} s_2 : G$ . By Application

Axiom A2 and *modus ponens*,  $X^\cdot \vdash_{\mathbb{JL}_{CS}} s_1 \cdot s_2 : F$ .

$*A3$ . *Sum Rule* and *Positive Introspection Rule* (the latter only for Clause 2)

are similar.

□

Another interesting connection between \*-calculi and justification logics is the ability to “strip” the outer terms from a \*-derivation.

**Definition 3.4.9.** For a set  $X$  of \*-expressions of type  $*(t, F)$ ,

$$X^\sharp = \{F \mid *(t, F) \in X\} .$$

◀

**Lemma 3.4.10.** *Let  $X$  be a set of \*-expressions.*

1. For  $JL_{CS} \in \{JCS, JD_{CS}, JT_{CS}\}$ ,

$$X \vdash_{*CS} *(t, F) \quad \Longrightarrow \quad X^\sharp \vdash_{JL_{CS}} F .$$

2. For  $JL_{CS} \in \{J4_{CS}, JD4_{CS}, LP_{CS}\}$ ,

$$X \vdash_{*CS} *(t, F) \quad \Longrightarrow \quad X^\sharp, X^\dagger \vdash_{JL_{CS}} F .$$

*Proof.* Proof by induction on the \*-derivation.

$*CS^\dagger$ . (Clause 1.) For any  $c:A \in CS$  and any integer  $n \geq 1$ ,

$$\vdash_{*CS} *(\underbrace{!\dots!}_n c, \underbrace{!\dots!}_{n-1} c : \dots : !c : !c : c : A) .$$

We can use  $R4_{\mathcal{CS}}^!$  to show

$$\mathbf{JL}_{\mathcal{CS}} \vdash \underbrace{! \dots !}_{n-1} c : \dots : !! c : ! c : c : A .$$

In addition, for any  $c : A \in \mathcal{CS}$ ,  $\vdash_{*\mathcal{CS}} *(c, A)$ . Here  $\mathbf{JL}_{\mathcal{CS}} \vdash A$  because  $A$  is an axiom of  $\mathbf{JL}_{\mathcal{CS}}$ .

*\*CS.* (Clause 2.) For any  $c : A \in \mathcal{CS}$ ,  $\vdash_{*\mathcal{CS}} *(c, A)$ . Again  $\mathbf{JL}_{\mathcal{CS}} \vdash A$  because  $A$  is an axiom of  $\mathbf{JL}_{\mathcal{CS}}$ .

*Hyp.* If  $*(t, F) \in X$ , then  $F \in X^\sharp$ , so  $X^\sharp \vdash_{\mathbf{JL}_{\mathcal{CS}}} F$ .

*\*A2. Application Rule* 
$$\frac{*(s_1, G \rightarrow F) \quad *(s_2, G)}{*(s_1 \cdot s_2, F)}$$

Let  $Y$  denote  $X^\sharp$  for the  $*\mathcal{CS}$ -calculus or  $X^\sharp \cup X^\circ$  for the  $!\mathcal{CS}$ -calculus.

By IH,  $Y \vdash_{\mathbf{JL}_{\mathcal{CS}}} G \rightarrow F$  and  $Y \vdash_{\mathbf{JL}_{\mathcal{CS}}} G$ . By *modus ponens*,  $Y \vdash_{\mathbf{JL}_{\mathcal{CS}}} F$ .

*\*A3. Sum Rule* is similar to *\*A2*.

*\*A5. (Clause 2.) Positive Introspection Rule* 
$$\frac{*(s, G)}{*(!s, s : G)}$$

By Lemma 3.4.8.2,  $X^\circ \vdash_{\mathbf{JL}_{\mathcal{CS}}} s : G$ .

□

*Note 3.4.11.* The addition of  $X^\circ$  to the set of hypotheses in the case of the  $!\mathcal{CS}$ -calculus is necessary, as was first pointed out by Vladimir Krupski in a

private conversation. The following example is due to him:

$$*(x, p) \vdash_{*!_{CS}} *(!x, x:p) ,$$

but

$$p \not\vdash_{\text{JL}_{CS}} x:p$$

for any justification logic  $\text{JL}_{CS}$  with Positive Introspection Axiom, any justification variable  $x$ , and any sentence letter  $p$ .

### 3.5 Hybrid Justification Logics

Modality and justifications present two sides of the epistemic coin. The use of modality to represent knowledge, although convenient, does not reflect the “justified” part of the centuries-old definition of knowledge as justified true belief, which goes back to Plato. (Needless to say, as any centuries-old idea, this definition has been hotly contested from the very beginning, even by Plato himself. Its detailed analysis by means of justification logic with a survey of literature can be found in [Art07].) Using justification terms seems to take care of this gap. At the same time, an assumption that we will always be given a concrete justification seems to be overoptimistic. We may know but not know *why* we know, in which case modality seems a better choice than a justification term. Hybrid logics combine the two worlds allowing to

use both explicitly stated reasons  $t:F$  and knowledge without specifying any reason  $\Box F$ .

We will consider a multiple agent situation. Therefore, in this section, modalities will be denoted by  $K_i$ ,  $i = 1, \dots, n$ , rather than by  $\Box_i$ . We will also assume that any evidence is undeniable and is accepted by all the agents. These assumptions underly the axiom systems described below.

### 3.5.1 Axiom Systems for Hybrid Justification Logics

**Definition 3.5.1.** *Formulas of hybrid justification language  $\mathcal{HL}_n$*  are obtained by combining modal constructs from language  $\mathcal{ML}_n$  with justification constructs from  $\mathcal{JL}$ :

$$F ::= p_i \mid \perp \mid (F \rightarrow F) \mid (K_j F) \mid (t:F) \quad (3.5.1)$$

where  $p_i$ ,  $i = 0, 1, 2, \dots$ , are sentence letters,  $t$  is a justification term of  $\mathcal{JL}$ , and  $j = 1, \dots, n$ .

We will call these formulas hybrid justification formulas, or simply hybrid formulas. The set of all hybrid formulas in language  $\mathcal{HL}_n$  will be denoted by  $Fm_n$ . ◀

*Note 3.5.2.* We will continue to denote hybrid formulas by Latin letters.

**Definition 3.5.3.** The size of hybrid formulas and terms is measured in the same way as in Def. 3.1.4, with an addition of a new case

$$|K_i G| = |G| + 1 ,$$

where  $G$  is a hybrid formula,  $i \geq 1$  is an integer. ◀

**Definition 3.5.4.** Axioms and rules of  $T_nLP_{CS}$  include

***Propositional part***

A1. Finitely many schemes of classical propositional logic in language

$\mathcal{HL}_n$  along with

*Modus Ponens Rule*

$$\frac{F \rightarrow G \quad F}{G}$$

***Justification part***

A2. *Application Axiom*

$$s : (F \rightarrow G) \rightarrow (t : F \rightarrow s \cdot t : G)$$

A3. *Monotonicity Axiom*

$$s : F \rightarrow s + t : F$$

$$t : F \rightarrow s + t : F$$

A4. *Factivity Axiom*

$$t : F \rightarrow F$$

A5. *Positive Introspection*

$$t : F \rightarrow !t : t : F$$

$R_{CS}$ . *Axiom Internalization Rule*

$$\frac{c : A \in CS}{c : A}$$

**Modal part**

$$K_i. \text{ Normality Axiom for } K_i \quad K_i(F \rightarrow G) \rightarrow (K_iF \rightarrow K_iG)$$

$$T_i. \text{ Reflexivity Axiom for } K_i \quad K_iF \rightarrow F$$

$$\text{Nec}_i. \text{ Modal Necessitation Rule for } K_i \quad \frac{\vdash F}{\vdash K_iF}$$

and the **Connection Principle** that details the relationship between justifications and knowledge

$$C1. \text{ Connection principle} \quad t:F \rightarrow K_iF$$

where  $F$  and  $G$  are hybrid formulas in language  $\mathcal{HL}_n$ ,  $t$  and  $s$  are justification terms in language  $\mathcal{JL}$ ,  $A$  is an axiom of the logic,  $c$  is a justification constant, and  $1 \leq i \leq n$  is an integer.  $\blacktriangleleft$

**Definition 3.5.5.** The system  $S4_nLP_{CS}$  is obtained by adding to the modal section of  $T_nLP_{CS}$ 's axioms the following

$$4_i. \text{ Modal Positive Introspection for } K_i \quad K_iF \rightarrow K_iK_iF \quad \blacktriangleleft$$

**Definition 3.5.6.** The system  $S5_nLP_{CS}$  is obtained by adding to the modal section of  $S4_nLP_{CS}$ 's axioms the following

$$5_i. \text{ Modal Negative Introspection for } K_i \quad \neg K_iF \rightarrow K_i\neg K_iF \quad \blacktriangleleft$$

**Definition 3.5.7.** As with justification logics, the *total constant specification* for  $\text{HL} \in \{\text{T}_n\text{LP}, \text{S4}_n\text{LP}, \text{S5}_n\text{LP}\}$  is

$$\mathcal{TCS}_{\text{HL}} = \{c:A \mid c \text{ is a justification constant, } A \text{ is an axiom of HL}\} .$$

$\text{T}_n\text{LP}$ ,  $\text{S4}_n\text{LP}$ , and  $\text{S5}_n\text{LP}$  will denote respective logics with their respective total constant specifications.

For  $n = 1$ , we will often omit the index and write  $\text{TLP}$ ,  $\text{S4LP}$ , and  $\text{S5LP}$  instead of  $\text{T}_1\text{LP}$ ,  $\text{S4}_1\text{LP}$ , and  $\text{S5}_1\text{LP}$  respectively.

We will use the term *hybrid logic* for any of  $\text{T}_n\text{LP}_{\mathcal{CS}}$ ,  $\text{S4}_n\text{LP}_{\mathcal{CS}}$ ,  $\text{S5}_n\text{LP}_{\mathcal{CS}}$ .



Hybrid logics enjoy the standard set of properties, such as Lifting Lemma, Deduction Theorem, and Substitution Property.

**Lemma 3.5.8 (Lifting Lemma, [AN04, Art04a]).** *For any hybrid logic  $\text{HL}_{\mathcal{CS}}$  with axiomatically appropriate  $\mathcal{CS}$ , if*

$$F_1, \dots, F_m, \quad y_1:G_1, \dots, y_k:G_k \quad \vdash_{\text{HL}_{\mathcal{CS}}} \quad B ,$$

*then there exists a term  $t(x_1, \dots, x_m)$  for some fresh justification variables  $x_i$ ,  $i = 1, \dots, m$ , such that*

$$x_1:F_1, \dots, x_m:F_m, \quad y_1:G_1, \dots, y_k:G_k \quad \vdash_{\text{HL}_{\mathcal{CS}}} \quad t(x_1, \dots, x_m):B .$$

In particular, for  $k = 0$ ,

**Corollary 3.5.9 (Internalization Property).** *For any hybrid logic  $\text{HL}_{\mathcal{CS}}$  with axiomatically appropriate  $\mathcal{CS}$ , if*

$$F_1, \dots, F_m \vdash_{\text{HL}_{\mathcal{CS}}} B ,$$

*then there exists a term  $t(x_1, \dots, x_m)$  for some fresh justification variables  $x_i$ ,  $i = 1, \dots, m$ , such that*

$$x_1:F_1, \dots, x_m:F_m \vdash_{\text{HL}_{\mathcal{CS}}} t(x_1, \dots, x_m):B .$$

If both  $k$  and  $m$  are put to 0,

**Corollary 3.5.10 (Constructive Necessitation).** *For any hybrid logic  $\text{HL}_{\mathcal{CS}}$  with axiomatically appropriate  $\mathcal{CS}$ , if*

$$\text{HL}_{\mathcal{CS}} \vdash B ,$$

*then there exists a ground term  $t$  such that*

$$\text{HL}_{\mathcal{CS}} \vdash t:B .$$

**Lemma 3.5.11 (Deduction Theorem, [AN04, Art04a]).** *For any hybrid logic  $\text{HL}_{\mathcal{CS}}$ , if*

$$\Gamma, F \vdash_{\text{HL}_{\mathcal{CS}}} G ,$$

then

$$\Gamma \vdash_{\text{HL}_{\mathcal{CS}}} F \rightarrow G .$$

**Lemma 3.5.12 (Substitution Property, [AN04, Art04a]).** *For any hybrid logic  $\text{HL}_{\mathcal{CS}}$  with schematic  $\mathcal{CS}$ , if*

$$\Gamma \vdash_{\text{HL}_{\mathcal{CS}}} F ,$$

then

$$\Gamma[s \setminus x, G \setminus p] \vdash_{\text{HL}_{\mathcal{CS}}} F[s \setminus x, G \setminus p] ,$$

where  $[s \setminus x, G \setminus p]$  means substituting justification term  $s$  for justification variable  $x$  and/or formula  $G$  for sentence letter  $p$ .

**Lemma 3.5.13 (Substitution Property with renaming of constants).**

*For any hybrid logic  $\text{HL}$  and any axiomatically appropriate  $\mathcal{CS}$  for  $\text{HL}$ , if*

$$\Gamma \vdash_{\text{HL}_{\mathcal{CS}}} F ,$$

then

$$\Gamma[s \setminus x, G \setminus p] \vdash_{\text{HL}_{\mathcal{CS}}} \overline{F}[s \setminus x, G \setminus p] ,$$

where  $[s \setminus x, G \setminus p]$  means substituting justification term  $s$  for justification variable  $x$  and/or formula  $G$  for sentence letter  $p$ , and formula  $\overline{F}$  is obtained from formula  $F$  by replacing some justification constants with other constants.

### 3.5.2 Semantics for Hybrid Logics

**Definition 3.5.14.** An *AF-model for a hybrid logic*  $\text{HL}_{CS}$  in language  $\mathcal{HL}_n$  is an  $(n + 4)$ -tuple

$$\mathfrak{M} = (W, R_e, R_1, \dots, R_n, V, \mathcal{A}) ,$$

where  $W \neq \emptyset$  is a set of worlds,  $R_i \subseteq W \times W$ ,  $i = 1, \dots, n$ , and  $R_e \subseteq W \times W$  are binary accessibility relations on  $W$ ,

$$V: SLet \rightarrow 2^W \tag{3.5.2}$$

is a *propositional valuation* that assigns to each sentence letter  $p$  a set of worlds where  $p$  is true,

$$\mathcal{A}: Tm \times Fm \rightarrow 2^W \tag{3.5.3}$$

is an *admissible evidence function*. Informally,  $\mathcal{A}(t, F) \subseteq W$  is a set of worlds where term  $t$  is considered admissible evidence for formula  $F$ .

Accessibility relations  $R_i$ ,  $i = 1, \dots, n$ , must be reflexive;  $R_e$  must be reflexive and transitive.  $R_i \subseteq R_e$ ,  $i = 1, \dots, n$ .

In addition, for  $\text{S4}_n\text{LP}_{CS}$ , binary relations  $R_i$ ,  $i = 1, \dots, n$ , must be transitive.

For  $\text{S5}_n\text{LP}_{CS}$ , binary relations  $R_i$ ,  $i = 1, \dots, n$ , must also be symmetric and transitive.

The admissible evidence function  $\mathcal{A}$  must satisfy the following closure conditions:

- *Application Closure*:  $\mathcal{A}(s, F \rightarrow G) \cap \mathcal{A}(t, F) \subseteq \mathcal{A}(s \cdot t, G)$ ;
- *Sum Closure*:  $\mathcal{A}(s, F) \cup \mathcal{A}(t, F) \subseteq \mathcal{A}(s + t, F)$ ;
- *Simplified CS Closure*: if  $c:A \in \mathcal{CS}$ , then

$$\mathcal{A}(c:A) = W ;$$

- *Positive Introspection Closure*:

$$\mathcal{A}(t, F) \subseteq \mathcal{A}(!t, t:F);$$

- *Monotonicity*:

$$u \in \mathcal{A}(t, F) \text{ and } uR_e v \text{ yield } v \in \mathcal{A}(t, F)$$

for any formulas  $F$  and  $G$  in language  $\mathcal{HL}_n$ , any terms  $t$  and  $s$  in language  $\mathcal{JL}$ , any justification constant  $c$ , any axiom  $A$  of the hybrid logic, and any worlds  $u, v \in W$ .

The *truth relation*  $\mathfrak{M}, u \models H$  is defined as follows:

$$\mathfrak{M}, u \models p \quad \Leftrightarrow \quad u \in V(p) \quad (3.5.4)$$

$$\mathfrak{M}, u \not\models \perp \quad (3.5.5)$$

$$\mathfrak{M}, u \models F \rightarrow G \quad \Leftrightarrow \quad \mathfrak{M}, u \not\models F \quad \text{or} \quad \mathfrak{M}, u \models G \quad (3.5.6)$$

$$\mathfrak{M}, u \models K_i F \quad \Leftrightarrow \quad \mathfrak{M}, w \models F \text{ for all } uR_i w \quad (3.5.7)$$

$$\mathfrak{M}, u \models t:F \quad \Leftrightarrow \quad \mathfrak{M}, w \models F \text{ for all } uR_e w \quad \text{and} \quad u \in \mathcal{A}(t, F) \quad (3.5.8)$$

As usual, a formula  $F$  is called *valid in an AF-model*  $\mathfrak{M} = (W, R, V, \mathcal{A})$ ,  $\mathfrak{M} \Vdash F$ , if  $F$  is true at every world  $w \in W$ :

$$\mathfrak{M}, w \Vdash F \quad \text{for each } w \in W.$$

A formula  $F$  is called *HL<sub>CS</sub>-valid* if  $F$  is valid in all AF-models of HL<sub>CS</sub>. ◀

**Theorem 3.5.15 (Completeness Theorem, [Art04a]).** *Let*

$$\text{HL} \in \{\text{T}_n\text{LP}, \text{S4}_n\text{LP}, \text{S5}_n\text{LP}\} .$$

*Then, the following holds:*

$$\text{HL}_{CS} \vdash F \quad \Leftrightarrow \quad F \text{ is HL}_{CS}\text{-valid.}$$

F-models are instances of AF-models with  $n = 1$ ,  $R_1 = R_e$ , and with  $\mathfrak{M}, w \Vdash \Box F$  defined as in (2.4.4).

**Theorem 3.5.16 (Completeness Theorem, [AN04, Fit04b]).** *S4LP is sound and complete w.r.t. F-models for LP.*

**Corollary 3.5.17.** *Hybrid logics  $T_nLP_{CS}$ ,  $S4_nLP_{CS}$ ,  $S5_nLP_{CS}$  are consistent.*

*Proof.* It is sufficient to present one model for each. We will present one model that fits all of them. Let  $W = \{w\}$ ,  $R_i = R_e = \{(w, w)\}$ ,  $V(p) = W$ , and  $\mathcal{A}(t, F) = W$ . It is easy to verify that all conditions for any of the logics are satisfied.  $\square$

### 3.5.3 Minimal Evidence Functions for AF-Models

We will now extend the main results about the minimal evidence functions to hybrid logics. Most proofs and some definitions can be applied literally, so we will only outline the necessary changes, if any.

F-type possible evidence functions (see Def. 3.3.27) can still be used for AF-models with a natural proviso that formulas now include all hybrid formulas. The definitions of one possible function being based on another (see Def. 3.3.29) and of the minimal evidence function in the given class of possible evidence functions (see Def. 3.3.30) also remain unchanged. Proposition 3.3.31 still holds.

**Theorem 3.5.18.** *Let  $HL_{CS} \in \{T_nLP_{CS}, S4_nLP_{CS}, S5_nLP_{CS}\}$ . For any (A)F-type possible evidence function  $\mathcal{B}$  on set  $W$  and any reflexive and transitive*

binary relation  $R_e \subseteq W \times W$ , the class  $\mathcal{AEF}_{\mathcal{B}}(\text{HL}_{\mathcal{CS}}, W, R_e)$  is not empty and has a (unique) minimal element.

*Note 3.5.19.* For AF-models, the Monotonicity Condition involves only  $R_e$ , hence its appearance in the formulation in place of  $R$  in Theorem 3.3.34.3. Note also that the justification part for all these hybrid logics is of LP type, hence the requirements of transitivity and reflexivity on  $R_e$ .

*Proof.* The proof is a word-for-word repetition of the proof of Theorem 3.3.34 for  $\text{LP}_{\mathcal{CS}}$  with the only change:  $R$  in the Monotonicity Condition has to be replaced by  $R_e$ .  $\square$

We will use the  $*!_{\mathcal{CS}}$ -calculus from Def. 3.3.39 (see also Table 3.3.4 and Def. 3.3.37) with rules  $*\mathcal{CS}$ ,  $*A2$ ,  $*A3$ , and  $*A5$  that was used for  $\text{LP}_{\mathcal{CS}}$ .

**Theorem 3.5.20.** *Let  $\text{HL}_{\mathcal{CS}} \in \{\text{T}_n\text{LP}_{\mathcal{CS}}, \text{S4}_n\text{LP}_{\mathcal{CS}}, \text{S5}_n\text{LP}_{\mathcal{CS}}\}$ , and let  $R_e$  be a reflexive and transitive binary relation on  $W$ . For any (A)F-type possible evidence function  $\mathcal{B}$  on  $W$ , define an (A)F-type possible evidence function  $\mathcal{A}$  on  $W$  according to*

$$*(t, F) \in \mathcal{A}_w^* \iff \bigcup_{uR_e w} \mathcal{B}_u^* \vdash_{*!_{\mathcal{CS}}} *(t, F) . \quad (3.5.9)$$

*Then,  $\mathcal{A} \in \mathcal{AEF}_{\mathcal{B}}(\text{HL}_{\mathcal{CS}}, W, R_e)$  is the minimal function in this class.*

*Note 3.5.21.* Note that  $\mathcal{B}_w^*$  is not a separate term in the union, unlike (3.3.38).

*Proof.* Again, we need to prove that

- $\mathcal{A}$  is based on  $\mathcal{B}$ ,
- $\mathcal{A} \subseteq \mathcal{E}$  for any  $\mathcal{E} \in \mathcal{AEF}_{\mathcal{B}}(\mathbf{HL}_{\mathcal{CS}}, W, R_e)$ ,
- $\mathcal{A}$  is admissible.

Suppose  $w \in \mathcal{B}(t, F)$ . Then,  $*(t, F) \in \mathcal{B}_w^*$ . Relation  $R_e$  is reflexive, so  $wR_e w$ . Thus,  $\bigcup_{uR_e w} \mathcal{B}_u^* \vdash_{*\!|\mathcal{CS}} *(t, F)$ . Hence,  $*(t, F) \in \mathcal{A}_w^*$ , i.e.,  $w \in \mathcal{A}(t, F)$ . This completes the proof that  $\mathcal{A}$  is based on  $\mathcal{B}$ .

The proof that any function  $\mathcal{E} \in \mathcal{AEF}_{\mathcal{B}}(\mathbf{HL}_{\mathcal{CS}}, W, R_e)$  must be based on  $\mathcal{A}$  is a repetition of the cases for  $\mathbf{LP}_{\mathcal{CS}}$  in Theorem 3.3.41, with  $R$  replaced by  $R_e$  again.

Finally, we need to show that  $\mathcal{A}$  itself is an admissible for  $\mathbf{HL}_{\mathcal{CS}}$  evidence function on  $(W, R_e)$ , namely, that it satisfies all the closure conditions. The only change necessary here is to the Monotonicity Condition:

Let  $uRw$  and  $u \in \mathcal{A}(t, F)$ . Then,  $\bigcup_{zR_e u} \mathcal{B}_z^* \vdash_{*\!|\mathcal{CS}} *(t, F)$ . By transitivity of  $R_e$ , if  $zR_e u$ , then  $zR_e w$ . So  $\bigcup_{zR_e u} \mathcal{B}_z^* \subseteq \bigcup_{zR_e w} \mathcal{B}_z^*$ . Thus,  $\bigcup_{zR_e w} \mathcal{B}_z^* \vdash_{*\!|\mathcal{CS}} *(t, F)$ . Therefore,  $w \in \mathcal{A}(t, F)$ .

Thus, the proof of Theorem 3.5.20 is complete.  $\square$

### 3.5.4 Reflected Fragments of Hybrid Logics

**Definition 3.5.22.** Again, for each hybrid logic  $\text{HL}_{CS}$  its *reflected fragment*  $\text{rHL}_{CS}$  is defined as

$$\text{rHL}_{CS} = \{t:F \mid \text{HL}_{CS} \vdash t:F\} . \quad (3.5.10)$$



**Theorem 3.5.23.** Let  $\text{HL}_{CS} \in \{\text{T}_n\text{LP}_{CS}, \text{S4}_n\text{LP}_{CS}, \text{S5}_n\text{LP}\}$ . Its reflected fragment  $\text{rHL}_{CS}$  is axiomatized by the  $*!_{CS}$ -calculus:

$$\text{rHL}_{CS} \vdash t:F \iff \text{HL}_{CS} \vdash t:F \iff *!_{CS}\text{-calculus} \vdash *(t, F)$$

**Corollary 3.5.24.** If  $CS$  can serve as a constant specification for both hybrid logics  $\text{HL}$  and  $\text{HL}'$ , then

$$\text{rHL}_{CS} = \text{rHL}'_{CS} .$$

*Proof of Theorem 3.5.23.* The left equivalency is by Def. 3.5.22 of the reflected fragment.

The  $\Leftarrow$  direction of the right equivalence is easily proven by induction on the derivation in the  $*!_{CS}$ -calculus (see an identical proof in Theorem 3.4.2).

It now remains to demonstrate the  $\Rightarrow$  direction of the right equivalence.

Let  $\text{HL}_{CS} \vdash t : F$ . Suppose towards a contradiction that  $\not\vdash_{*!_{CS}} *(t, F)$ . Let  $W = \{w\}$  and  $R_e = R_1 = \dots = R_n = \{(w, w)\}$ . Let  $B_\emptyset \equiv \emptyset$  be the empty possible evidence function on  $W$ . By Theorem 3.5.20, the class  $\mathcal{AEF}_{B_\emptyset}(\text{HL}_{CS}, W, R_e)$  has a minimal function defined by (3.5.9). Let us denote this minimal function by  $\mathcal{A}$ .

Note that the chosen  $R_e$  and  $R_i$ ,  $i = 1, \dots, n$ , are reflexive, transitive, and symmetric;  $R_e \subseteq R_i$  for  $i = 1, \dots, n$ . Moreover, for this  $R_e$  and for  $B_\emptyset$ , (3.5.9) becomes

$$w \in \mathcal{A}(t, F) \iff \vdash_{*!_{CS}} *(t, F) .$$

We assumed the right side to be false, which requires the left side to be false too, i.e.,  $w \notin \mathcal{A}(t, F)$ . Choose a propositional valuation  $V$  arbitrarily to get an AF-model  $\mathfrak{M} = (W, R_e, R_1, \dots, R_n, V, \mathcal{A})$  for  $\text{HL}_{CS}$ . All conditions on  $R_e$  and  $R_i$  are satisfied.

In this model,  $\mathfrak{M}, w \not\Vdash t : F$  since  $w \notin \mathcal{A}(t, F)$ . By soundness of  $\text{HL}_{CS}$ , this contradicts the initial assumption that  $\text{HL}_{CS} \vdash t : F$ . The contradiction completes the proof of the  $\implies$  direction and of Theorem 3.5.23.  $\square$

**Lemma 3.5.25.** *Let  $\text{HL} \in \{\text{T}_n\text{LP}, \text{S4}_n\text{LP}, \text{S5}_n\text{LP}\}$  and  $CS$  be a constant specification for  $\text{HL}$ . Then,*

$$X \vdash_{*!_{CS}} *(t, F) \implies X : \vdash_{\text{HL}_{CS}} t : F$$

*Proof.* The proof repeats word-for-word the proof of Lemma 3.4.8.  $\square$

### 3.5.5 Historical Survey

The first studies of hybrid logics combining modal operators with justification terms were started by Sergei Artemov in [Art94] and Elena Nogina in [Nog94] where the authors were trying to model arithmetical provability without any operations on justification terms. This line of research was continued by Tatiana Yavorskaya (Sidon) in [Sid97, Yav01a] and in joint work by Artemov and Nogina [AN04]. In these works, Kripke-style models were developed and arithmetical completeness and decidability were demonstrated. Our research is concentrated on the epistemic modal logics rather than on modeling the properties of arithmetical proofs; therefore, the logics involving **GL** or **Grz** are outside of the scope of this thesis.

The first paper to combine the epistemic modal logic **S4** with justification terms was [AN04]. Two systems were introduced there, **LPS4** and **LPS4<sup>-</sup>**. The former can be identified with **S4<sub>1</sub>LP** in the modern notation, whereas the latter is obtained by adding to it the *weak principle of negative introspection*, also called *explicit negative introspection principle*

$$\neg t : F \rightarrow \Box \neg t : F \text{ .}$$

**LPS4<sup>-</sup>** was supplied with a somewhat antiquated Kripke-style semantics

whereas  $S4_1LP$  turned out to be sound w.r.t. F-models.

The completeness of  $S4_1LP$  w.r.t. F-models was shown by Melvin Fitting in [Fit04b].

Artemov in [Art04a] (see also [Art06]) generalized  $S4_1LP$  to multiple modalities of one of the three types:  $T$ ,  $S4$ , or  $S5$ , thus creating logics  $T_nLP$ ,  $S4_nLP$ , and  $S5_nLP$ . In that paper he used the term logics of *evidence-based knowledge* for the logics we call hybrid here. Artemov suggested AF-models as the new semantical framework that generalizes F-models and proved soundness and completeness of all three series of hybrid logics w.r.t. their respective AF-models.

AF-models were applied in [AN05a] (see also [AN05c, AN05b]) to create a more elegant semantics for  $LPS4^-$  rebranded  $S4LPN$ . Namely, it was shown that  $S4LPN$  is sound and complete w.r.t. AF-models with symmetric  $R_e$ . Note that  $R_e$  must also be reflexive and transitive, which makes it an equivalence relation.

In all these logics the justification part is based on LP. Natalia Rubtsova considered logics with justifications based on JT45:  $S4_nLP(S5)$  in [Rub06a] and  $S5_nLP(S5)$  in [Rub06c, Rub06d]. But these logics remain outside the scope of this thesis.

# Chapter 4

## Decidability

Finite Model Property (FMP) is often the tool used for proving decidability in modal logic. As we discussed in Sect. 3.3.3, in many cases M-models are nothing but one-world F-models. Thus, completeness w.r.t. M-models is a very strong form of FMP. The question of decidability should then be closed? Unfortunately, the situation is not as simple as it may seem (actually, it is not simple in modal logic either). No matter how small  $W$  is in an F-model (Fk-model, AF-model), the admissible evidence function is necessarily not a finite object. We need, therefore, to generalize the FMP traditionally used. We will start by its detailed analysis. Because of the extreme sensitivity of the issue, we will resort to quotes from popular textbooks and monographs in the next section.

## 4.1 Finite Model Property vs. Finite Frame Property

Here is how the Finite Model Property (FMP) and the finite frame property are traditionally defined:<sup>1</sup>

**Definition 4.1.1.** A logic  $L$  has the *Finite Model Property* if it is complete with respect to some class of finite Kripke models. ◀

**Definition 4.1.2.** A logic  $L$  is said to be *finitely approximable* (or to have the *finite frame property*) if there is a class  $\mathcal{C}_F$  of finite frames such that

$$L = \{ \varphi \mid \forall \mathfrak{F} \in \mathcal{C}_F \quad \varphi \text{ is valid in } \mathfrak{F} \} .$$

◀

Proving one of those properties is a road to establishing decidability by means of Post's argument:

**Theorem 4.1.3 (Post's Theorem).** *If both a set and its complement are recursively enumerable, then the set is decidable.*

Usually the set of theorems of a logic is recursively enumerable. So the finite frame property can be used to ensure that the complement of the

---

<sup>1</sup>These formulations are taken from [CZ97, p.119] and [CZ97, p.49] respectively.

logic, the set of all refutable formulas is recursively enumerable too. The idea is to enumerate refutable formulas of the logic through an enumeration of refuting frames from class  $\mathcal{C}_F$ . Indeed, Lemma 16.12 in [CZ97, p.497] explicitly states:<sup>2</sup>

If  $L$  is characterized by **recursively enumerable** class of finite  
 [...] frames [...] then the set of formulas which do not belong to  $L$   
 is recursively enumerable[...]

even though the requirement of being recursively enumerable is omitted from Harrop's Theorem 16.13 in [CZ97, p.497]:

**Theorem 4.1.4 (Harrop's Theorem).** *Every finitely axiomatizable and finitely approximable logic  $L$  is decidable.*

This omitted assumption rarely comes into play since most commonly studied modal logics are complete w.r.t decidable classes of Kripke frames, let alone recursively enumerable. In particular, all the classes of frames described in Theorem 2.4.15 are clearly decidable.

Switching from frames to models involves another hidden assumption, this time an assumption that paves the way to generalizing FMP to F- and AF-models. The problem is that the set of all (distinct representations of) finite

---

<sup>2</sup>In all the quotes in this section boldface is by RK.

models is uncountable, so it clearly cannot be recursively enumerable. Even the class of all single-world models is already uncountable because there are uncountably many propositional valuations for the countably many sentence letters. That is why the model variant of Harrop's Theorem in [BdRV01] is formulated with care (see Theorem 6.7 in [BdRV01, p.340]):

If  $L$  is a normal modal logic that has the strong Finite Model Property with respect to a **recursive** set of models  $\mathcal{C}_M$ , then  $L$  is decidable.<sup>3</sup>

A formulation more akin to Harrop's Theorem would be

If  $L$  is a **finitely axiomatizable** normal modal logic that has the Finite Model Property with respect to a **recursively enumerable** set of models  $\mathcal{C}_M$ , then  $L$  is decidable.

Note the requirement for the class of refuting models to be recursively enumerable. We need the generalized FMP to be formulated in a way that would guarantee such recursive enumerability.

Later in the same textbook, there is an application to **K4** (see proof of Corollary 6.8 in [BdRV01, pp.340–341]):

---

<sup>3</sup>*Strong* Finite Model Property is the requirement for the size (number of worlds) of the countermodel to be a computable function of the length of formula to be refuted. It is necessary here as is decidability of the class of models because the logic is not required to be finitely axiomatizable.

K4 has the finite model property with respect to the set of finite transitive models [...] It remains to check that the relevant sets of finite models are recursive. Checking for membership in these sets boils down to checking that the models possess [...] such properties as [...] transitivity [...] It is **clearly possible** to devise algorithms to test for the relevant properties [...]

The argument would have been correct were the set of all finite transitive models countable. As it is not, the desired algorithm does not exist for a trivial reason: the set of all finite transitive models cannot be encoded in any finite alphabet. This is exactly the problem pointed out in [FHMV95, p.63]:

There is no general procedure for doing model checking in an infinite Kripke structure. Indeed, **it is not even possible to represent arbitrary infinite structures effectively.**

The reason this small, but important point is often being silently bypassed lies, perhaps, in the following theorem (see, for example, [BdRV01, Theorem 3.28]):

**Theorem 4.1.5.** *A normal modal logic has the finite frame property iff it has the Finite Model Property.*

Even though the set of all finite models is uncountable, the set of all finite frames is certainly countable, and that is exactly what Blackburn *et al.* mean by “checking for membership in these sets.”

When efficiency becomes important, for instance, when complexity of the decision procedure is being studied, even more care is necessary. Another hidden assumption is uncovered in [FHMV95]. Not only is it stated that the class of models should be effectively described, but it is also made explicit that, to obtain a recursive enumeration of all refutable formulas from a recursive enumeration of all refuting models, it is necessary to be able to effectively check whether a given formula is true at a given world of a given model. Here is a sample proposition to this effect (see Proposition 3.2.1 in [FHMV95, p.63]):<sup>4</sup>

There is an algorithm that, given a [Kripke model]  $\mathfrak{M}$ , a [world]  $w$  of  $\mathfrak{M}$ , and a formula  $\varphi \in \mathcal{ML}_n$ , determines, in time  $O(|\mathfrak{M}| \times |\varphi|)$ , whether  $\mathfrak{M}, w \models \varphi$ .

Here  $|\mathfrak{M}|$  for a model  $\mathfrak{M} = (W, R, V)$  is the number of worlds in  $W$  plus the number of pairs in  $R$ .

This proposition, though false, probably best exemplifies the problems we are facing when an admissible evidence function is superimposed over a

---

<sup>4</sup>The notation in the quote is converted to the one used in this thesis.

Kripke model. The proposition is false because the problem in question is undecidable. In fact, it is not hard to construct a one-world model where this problem would be undecidable already for sentence letters. The recipe is simple: take an undecidable valuation  $V$ .

## 4.2 Hidden Assumptions in FMP

Not surprisingly, the culprit is again the propositional valuation function, which is an infinitary object, a function with an infinite (though countable) domain. This makes the set of all finite models uncountable because  $2^{\aleph_0} = \mathfrak{c}$ .

But most authors, as we saw, ignore the infinitary nature of  $V$ , and they have good reasons too. To refute one formula, say  $\varphi$ , we only need to take care of the sentence letters occurring in  $\varphi$ . All other sentence letters have no effect on the truth value of  $\varphi$ . But each formula contains only finitely many sentence letters, which effectively turns the propositional valuation into a finitary object.

There are two ways to make this official: ignoring the other variables altogether as in [BdRV01, p. 10]:<sup>5</sup>

Although we generally assume that the set  $SLet$  of sentence letters is a countably infinite set  $\{p_0, p_1, \dots\}$ , occasionally we need

---

<sup>5</sup>Again the notation is changed from the original.

to make other assumptions. For instance, when we are **after decidability results**, it may be useful to stipulate that  $SLet$  is **finite** [...]

as in “restricted to the sentence letters occurring in the formula.” Later it is formulated quite clearly (see [BdRV01, p.335]):

[W]hen evaluating a formula  $\varphi$  in some model, the only relevant information in the valuation is the assignments made to proposition letters actually occurring in  $\varphi$  [...]. Thus, instead of working with  $V$ , we can work with the finite valuation  $V'$  which is defined on the (finite) language consisting of exactly the proposition letters in  $\varphi$ , and which agrees with  $V$  on these letters.

In effect, this requires to consider partial Kripke models that are formula-dependent, or rather models in which some formulas are true, some are false, and some are undefined (if the formula has variables not assigned a truth value by the finite valuation). Soundness does not hold with respect to these models, but certain variant of completeness does, namely, a formula is refutable iff it is refutable in such a partial model. This is exactly what is needed to prove decidability via Post’s argument. The set of all *partial* Kripke models is decidable and truth in such models can be effectively determined,

so all the hurdles are cleared.

The alternative is to acknowledge the unimportance of most variables by forcibly making them all false. Consider a subclass of Kripke models with *finitely true* valuations:

**Definition 4.2.1.** A propositional valuation  $V: SLet \rightarrow 2^W$  is called *finitely true* if the set

$$\{p \mid V(p) \neq \emptyset\}$$

is finite. ◀

In this way we change neither language nor models. Theorem 2.4.15 lists the restrictions on the accessibility relation  $R$  for common modal logics. The soundness and completeness statements survive the restriction of  $W$  to finite sets and the restriction of  $V$  to finitely true valuations.

Which of the two ways is more elegant is, of course, a matter of taste. The former solution, partial models, is, in a way, reader-friendly because many of the inelegant details (such as partial soundness) are relegated to the depths of the completeness proof, the reader does not have to deal with them while applying the theorem. The latter solution, on the contrary, keeps the completeness proof relatively tidy, but shifts part of the responsibility to the reader by requiring him/her to conform to the additional (rather trivial)

restriction on valuations.

To emphasize that care is indeed needed around FMP, it is useful to keep in mind the Alasdair Urquhart’s example of a recursively enumerable modal logic with a finite model property that is nevertheless undecidable ([Urq81]). This example shows that even within modal logic the hidden assumptions in FMP are a treacherous ground the moment you leave the beaten path.

### 4.3 Finitary Model Property

To summarize the discussion, restricting the class of models to finite ones does not by itself yield decidability as even a finite model harbors an infinitary object (propositional valuation). Let us factor all the hidden assumptions back into the formulation of FMP:

**Lemma 4.3.1.** *Let finitely axiomatizable logic  $\mathsf{L}$  be sound and complete with respect to a class of models  $\mathcal{C}_M$ , such that*

- *Class  $\mathcal{C}_M$  is recursively enumerable;*
- *the binary relation “formula  $\varphi$  is satisfiable in model  $\mathfrak{M}$ ” between formulas and models from  $\mathcal{C}_M$  is decidable.*

*Then,  $\mathsf{L}$  is decidable.*

*Proof.* It is well known that a finitely axiomatizable logic is recursively enumerable.

Here is an algorithm recursively enumerating the complement of the logic. Since both the set of models and the set of well-formed formulas are recursively enumerable, there exists an enumeration of all pairs  $(\mathfrak{M}, \varphi)$ . For each pair in this enumeration the algorithm checks whether  $\neg\varphi$  is satisfiable in  $\mathfrak{M}$ . If it is, the algorithm outputs  $\varphi$ , otherwise it skips to the next pair. In this way, the algorithm will list all the non-theorems of  $L$ , so the complement of  $L$  is recursively enumerable too.

By Post's Theorem,  $L$  is decidable. □

This leads to a formulation of a more specific finitary model property for Kripke models (not necessarily for modal logic):

**Definition 4.3.2.** A logic  $L$  has the *finitary model property* if it is sound and complete with respect to a class  $\mathcal{C}_M$  of finite models, such that

- All models  $\mathfrak{M} \in \mathcal{C}_M$  can be encoded in one finite alphabet;
- Ternary relation  $\lceil \mathfrak{M} \rceil, w \Vdash \varphi$  between codes of models from  $\mathcal{C}_M$ , worlds in such a model, and formulas is decidable. ◀

**Theorem 4.3.3.** *A recursively enumerable logic that has the finitary model property is decidable.*

*Proof.* The models are encoded in a finite alphabet. Encoding here does not mean that all the words in this alphabet must be codes of some models. Rather it means that it is decidable whether a given word is a code of some model, and if yes, then this model can be effectively restored. Therefore, the set of all such models is recursively enumerable.

Each model is finite; satisfiability in the model is defined as satisfiability in some world of that model. Since there is an algorithm checking satisfiability at a particular world and the number of worlds is finite, it is easy to check satisfiability in the whole model.

Now decidability follows from Lemma 4.3.1. □

We will now apply this framework to showing decidability of various logics described in Sect. 3.

## 4.4 Decidability Results

We will start with justification logics. We need to present a suitable encoding for finite models, notably an encoding for admissible evidence functions, and then present an algorithm for determining truth of a given formula at a given world.

**Definition 4.4.1.** A possible evidence function  $\mathcal{B}: Tm \times Fm \rightarrow 2^W$  on a

finite set of worlds  $W$  is called *finitary* if the set of pairs

$$\{(t, F) \mid \mathcal{B}(t, F) \neq \emptyset\}$$

is finite. ◀

A finitary possible evidence function  $\mathcal{B}$  can be easily encoded by the set

$$\ulcorner \mathcal{B} \urcorner = \{(w, t, F) \mid w \in \mathcal{B}(t, F)\} . \quad (4.4.1)$$

**Proposition 4.4.2.** *The set  $\ulcorner \mathcal{B} \urcorner$  is finite for any finitary possible evidence function  $\mathcal{B}$  on a finite set  $W$ .*

*Proof.* By Def. 4.4.1, there are only finitely many pairs  $(t, F)$  to be taken into account. For each of them there may be only finitely many  $w \in W$ .  $\square$

The finite set  $W$  and any accessibility relation  $R$  on  $W$  can be encoded by listing all worlds  $w \in W$  and all pairs  $(u, w) \in R$  respectively. Note that a binary relation on a finite set is always finite.

Finally, any finitely true valuation  $V$  can be encoded by

$$\ulcorner V \urcorner = \{(w, p) \mid w \in V(p)\} . \quad (4.4.2)$$

**Proposition 4.4.3.** *The set  $\ulcorner V \urcorner$  is finite for any finitely true valuation  $V$  on a finite set  $W$ .*

*Proof.* By Def. 4.2.1, there are only finitely many sentence letters  $p$  to look at.

For each of them there may be only finitely many  $w \in W$ .  $\square$

**Definition 4.4.4.** For each pure justification logic  $\mathbb{JL}_{CS}$  we will consider the class  $\mathcal{C}_{\mathbb{JL}_{CS}}$  of all ***finitary F-models for  $\mathbb{JL}_{CS}$*** , i.e., of all models  $\mathfrak{M} = (W, R, V, \mathcal{A})$  for  $\mathbb{JL}_{CS}$  with

- finite  $W$ ,
- finitely true  $V$ , and
- $\mathcal{A}$  that is the minimal evidence function based on a finitary possible evidence function  $\mathcal{B}$

encoded by quadruples

$$\ulcorner \mathfrak{M} \urcorner = (W, R, \ulcorner V \urcorner, \ulcorner \mathcal{B} \urcorner) . \quad (4.4.3)$$



**Definition 4.4.5.** Similarly, for each hybrid logic  $\mathbb{HL}_{CS}$  we will consider the class  $\mathcal{C}_{\mathbb{HL}_{CS}}$  of all ***finitary AF-models for  $\mathbb{HL}_{CS}$*** , i.e., of all models  $\mathfrak{M} = (W, R_e, R_1, \dots, R_n, V, \mathcal{A})$  for  $\mathbb{HL}_{CS}$  with

- finite  $W$ ,
- finitely true  $V$ , and

- $\mathcal{A}$  that is the minimal evidence function based on a finitary possible evidence function  $\mathcal{B}$

encoded by tuples

$$\ulcorner \mathfrak{M} \urcorner = (W, R_e, R_1, \dots, R_n, \ulcorner V \urcorner, \ulcorner \mathcal{B} \urcorner) . \quad (4.4.4)$$



**Lemma 4.4.6.**

1. The encoding of finitary  $F$ -models from  $\mathcal{C}_{\text{JL}_{CS}}$  described in (4.4.3) is effective for any  $\text{JL}_{CS} \in \{\text{J}_{CS}, \text{JD}_{CS}, \text{JT}_{CS}, \text{J4}_{CS}, \text{JD4}_{CS}, \text{LP}_{CS}\}$ .
2. The encoding of finitary  $AF$ -models from  $\mathcal{C}_{\text{HL}_{CS}}$  described in (4.4.4) is effective for any  $\text{HL}_{CS} \in \{\text{T}_n\text{LP}_{CS}, \text{S4}_n\text{LP}_{CS}, \text{S5}_n\text{LP}_{CS}\}$ .

*Proof.* The proof is almost trivial. The only thing we need from the encoding is to be able to effectively tell codes of models from non-codes.

This involves verifying conditions on  $R$  for pure justification logics or on  $R_i$ ,  $i = 1, \dots, n$ , and  $R_e$  for hybrid logics. All these conditions, i.e., transitivity, reflexivity, seriality, and/or  $R_i \subseteq R_e$ , depending on the logic, are clearly decidable for a finite domain  $W$ .

Needless to say, finiteness of  $W$  is implied by the fact that it can be fully written in the finite code.

Clearly, any finite set  $\ulcorner V \urcorner$  of type (4.4.2) describes a finitely true valuation

$$V(p) = \{w \mid (w, p) \in \ulcorner V \urcorner\} . \quad (4.4.5)$$

No additional conditions are imposed on the propositional valuation for any of the logics.

Similarly, the possible evidence function

$$\mathcal{B}(t, F) = \{w \mid (w, t, F) \in \ulcorner \mathcal{B} \urcorner\} \quad (4.4.6)$$

is finitary. By Theorem 3.3.34.2 and 3.3.34.3, for any such  $\mathcal{B}$  there exists a unique minimal admissible for  $\text{JL}_{CS}$  evidence function based on  $\mathcal{B}$ . By Theorem 3.5.18, for any such  $\mathcal{B}$  there exists a unique minimal admissible for  $\text{HL}_{CS}$  evidence function based on  $\mathcal{B}$ .  $\square$

**Lemma 4.4.7.** *Let  $\mathfrak{M}$  be*

1. *a finitary F-model  $(W, R, V, \mathcal{A})$  for a pure justification logic*

$$\text{JL}_{CS} \in \{\text{J}_{CS}, \text{JD}_{CS}, \text{JT}_{CS}, \text{J4}_{CS}, \text{JD4}_{CS}, \text{LP}_{CS}\}$$

*with a decidable schematic  $CS$ , or*

2. *a finitary AF-model  $(W, R_e, R_1, \dots, R_n, V, \mathcal{A})$  for a hybrid logic*

$$\text{HL}_{CS} \in \{\text{T}_n \text{LP}_{CS}, \text{S4}_n \text{LP}_{CS}, \text{S5}_n \text{LP}_{CS}\}$$

*with a decidable schematic  $CS$*

with  $\mathcal{A}$  encoded through a finitary possible evidence function  $\mathcal{B}$ . Then, the ternary relation

$$w \in \mathcal{A}(t, F)$$

between worlds  $w \in W$ , terms  $t$ , and formulas  $F$  is decidable.

*Proof.* The admissible evidence function  $\mathcal{A}$  based on  $\mathcal{B}$  is fully described at any given world  $w$

- by (3.3.37) for  $J_{CS}$ ,  $JD_{CS}$ , and  $JT_{CS}$ ;
- by (3.3.38) for  $J4_{CS}$ ,  $JD4_{CS}$ , and  $LP_{CS}$ ;
- by (3.5.9) for hybrid logics.

Let  $*_{\mathcal{B},w}$  stand for the set of hypotheses allowed in the  $*$ -derivation in the right side of the respective equivalence:

$$*_{\mathcal{B},w} = \begin{cases} \mathcal{B}_w^* & \text{for } J_{CS}, JD_{CS}, \text{ and } JT_{CS} \\ \mathcal{B}_w^* \cup \bigcup_{uRw} \mathcal{B}_u^* & \text{for } J4_{CS}, JD4_{CS}, \text{ and } LP_{CS} \\ \bigcup_{uR_{\epsilon}w} \mathcal{B}_u^* & \text{for hybrid logics} \end{cases}$$

Note that in all cases  $*_{\mathcal{B},w}$  is a finite set. Thus, to show decidability we need a decision algorithm for  $*_{CS}$ - and  $*!_{CS}$ -derivations from a finite set  $*_{\mathcal{B},w}$ .

First of all, given the particular term  $t$  we only need to check derivability of  $*(s, G)$  for subterms  $s$  of  $t$  since both  $*$ -calculi increase the complexity of the first term-argument after each rule application.

We would like to organize the derivation in such a way that after  $f(k)$  steps we would know all formulas in the sets

$$*(s) = \{G \mid *_{\mathcal{B},w} \vdash_* *(s, G)\}$$

for all subterms  $|s| \leq k$ , where  $f$  is some computable function. This way we would be able to complete the first  $f(|t|)$  steps, and then check whether  $*_{\mathcal{B},w} \vdash_* *(t, F)$  for the given formula  $F$ .

Unfortunately, the sets  $*(s)$  can be infinite already for atomic terms  $s$ , in particular, for justification constants. Therefore, to perform the procedure constructively, we will need to represent these infinite sets in a finite way. We will employ variables  $P, Q, \dots$  over formulas and also variables over justification terms (they will not be present explicitly, but they are nevertheless necessary to write justification axiom schemes such as A4). We will use letters  $X, Y, \dots$  to denote schemes of formulas as opposed to  $F, G, \dots$  reserved for formulas themselves. In this extended language, infinitely many axioms can be compressed into finitely many axiom schemes, which will require the use of unification in the course of a  $*$ -derivation.

In this extended language we construct a sequence of sets

$$*_0 \subseteq *_1 \subseteq \dots \subseteq *_n \subseteq \dots ,$$

each set containing finitely many schemes of formulas, in the following way.

Let  $*_0 = *_{\mathcal{B},w}$ . Let  $*_{n+1}$  be obtained from  $*_n$  by applying the following procedures corresponding to the rules and axioms of the respective  $*$ -calculus adjusted for schemes of formulas:

$*\mathcal{CS}$ . (Only for  $\mathbf{J4}_{\mathcal{CS}}$ ,  $\mathbf{JD4}_{\mathcal{CS}}$ ,  $\mathbf{LP}_{\mathcal{CS}}$  and hybrid logics.) Since  $\mathcal{CS}$  is schematic it is possible to write each set  $*(c)$  as a finite number of schemes of axioms for each justification constant  $c$ . For each subterm  $c$  of  $t$  and each axiom scheme  $X \in *(c)$  add  $*(c, X)$  to  $*_1$  if it was not in  $*_0$ . Do not do anything on further steps.

$*\mathcal{CS}^!$ . (Only for  $\mathbf{J}_{\mathcal{CS}}$ ,  $\mathbf{JD}_{\mathcal{CS}}$ , and  $\mathbf{JT}_{\mathcal{CS}}$ .) For  $*_1$  do the same as in  $*\mathcal{CS}$ . For each  $*(c, X)$  added to  $*_1$  in this step, add

$$*(\underbrace{!\dots!}_n c, \underbrace{!\dots!}_{n-1} c : \dots : !!c : !c : c : X)$$

to  $*_{n+1}$  if it was not in  $*_n$ .

$*\mathbf{A2}$ . For any  $*(s_1, X_1 \rightarrow Y_1) \in *_n$  and any  $*(s_2, X_2) \in *_n$ , where  $s_1 \cdot s_2$  is a subterm of  $t$ , find the most general unifier (mgu)  $\sigma$  of  $X_1$  and  $X_2$ . If it exists, add  $*(s_1 \cdot s_2, Y_1\sigma)$  to  $*_{n+1}$  if it was not in  $*_n$ . If  $X_1$  and  $X_2$  do not unify, do not add anything.

For any  $*(s_1, P) \in *_n$ , where  $P$  is a variable over formulas, and any  $*(s_2, X_2) \in *_n$ , where  $s_1 \cdot s_2$  is a subterm of  $t$ , add  $*(s_1 \cdot s_2, Q)$  to  $*_{n+1}$

if it was not in  $*_n$ , where  $Q$  is a fresh variable over formulas.

\*A3. For any  $*(s_1, X) \in *_n$  and any  $s_2$  such that  $s_1 + s_2$  is a subterm of  $t$ ,  
add  $*(s_1 + s_2, X)$  to  $*_{n+1}$  if it was not in  $*_n$ .

For any  $*(s_2, X) \in *_n$  and any  $s_1$  such that  $s_1 + s_2$  is a subterm of  $t$ ,  
add  $*(s_1 + s_2, X)$  to  $*_{n+1}$  if it was not in  $*_n$ .

\*A5. (Only for  $J4_{CS}$ ,  $JD4_{CS}$ ,  $LP_{CS}$  and hybrid logics.) For any  $*(s, X) \in *_n$ ,  
where  $!s$  is subterm of  $t$ , add  $*(!s, s : X)$  to  $*_{n+1}$  if it was not in  $*_n$ .

It should be clear the each of the sets  $*_n$  is finite. Moreover, each  $*_n$  can  
be effectively constructed.

Construct  $*_{|t|}$ . We claim that

$F$  unifies with one of the  $X$  such that  $*(t, X) \in *_{|t|}$  iff  $*_{\mathcal{B}, w} \vdash_* *(t, F)$

Indeed, the procedure above faithfully represents the rules of the respective  
\*-calculus as far as subterms of  $t$  are concerned. Finally, on step  $n$ , we only  
add  $*(s, X)$  with  $|s| \geq n$ . Therefore, no new  $*(t, X)$  can be added after  
step  $|t|$ . □

**Corollary 4.4.8.** *Let  $\mathfrak{M}$  be*

1. *a finitary  $F$ -model  $(W, R, V, \mathcal{A})$  for a pure justification logic*

$$JL_{CS} \in \{J_{CS}, JD_{CS}, JT_{CS}, J4_{CS}, JD4_{CS}, LP_{CS}\}$$

with a decidable schematic  $\mathcal{CS}$ , or

2. a finitary AF-model  $(W, R_e, R_1, \dots, R_n, V, \mathcal{A})$  for a hybrid logic

$$\text{HL}_{\mathcal{CS}} \in \{\text{T}_n\text{LP}_{\mathcal{CS}}, \text{S4}_n\text{LP}_{\mathcal{CS}}, \text{S5}_n\text{LP}_{\mathcal{CS}}\}$$

with a decidable schematic  $\mathcal{CS}$

with a finitely true  $V$  and with  $\mathcal{A}$  encoded based on a finitary possible evidence function  $\mathcal{B}$  on  $W$ . Then, binary relation

$$\mathfrak{M}, w \Vdash F$$

between worlds  $w \in W$  and formulas  $F$  is decidable.

*Proof.* We prove decidability by induction on the size of  $F$ .

Deciding whether  $\mathfrak{M}, w \Vdash p$  for a given world  $w$  and a given sentence letter  $p$  amounts to deciding whether  $w \in V(p)$ , which is equivalent to  $(w, p) \in \ulcorner V \urcorner$  by (4.4.2).

Boolean cases are trivial.

Deciding whether  $\mathfrak{M}, w \Vdash t : G$  requires checking whether (1)  $\mathfrak{M}, u \Vdash G$  for all  $wR_e u$  and (2)  $w \in \mathcal{A}(t, G)$ . There are only finitely many such  $u$ 's and  $G$  has size smaller than  $t : G$ , which allows us to verify (1).

Decidability of (2) was demonstrated in Lemma 4.4.7.

Deciding whether  $\mathfrak{M}, w \Vdash K_i G$  (for hybrid logics) requires checking whether  $\mathfrak{M}, u \Vdash G$  for all  $wR_i u$ . There are only finitely many such  $u$ 's and  $G$  has size smaller than  $\Box G$ .

□

Only one thing remains to be shown to prove the finitary model property of pure and hybrid justification logics with decidable schematic  $\mathcal{CS}$ , namely completeness w.r.t to models encoded as described above.

**Theorem 4.4.9.** 1. *A pure justification logic*

$$\mathbf{JL}_{\mathcal{CS}} \in \{\mathbf{J}_{\mathcal{CS}}, \mathbf{JT}_{\mathcal{CS}}, \mathbf{J4}_{\mathcal{CS}}, \mathbf{LP}_{\mathcal{CS}}\}$$

*with a decidable schematic  $\mathcal{CS}$  is sound and complete w.r.t. the class of its finitary models  $\mathcal{C}_{\mathbf{JL}_{\mathcal{CS}}}$ .*

2. *A pure justification logic*

$$\mathbf{JL}_{\mathcal{CS}} \in \{\mathbf{JD}_{\mathcal{CS}}, \mathbf{JD4}_{\mathcal{CS}}\}$$

*with a decidable, schematic, and axiomatically appropriate  $\mathcal{CS}$  is sound and complete w.r.t. the class of its finitary models  $\mathcal{C}_{\mathbf{JL}_{\mathcal{CS}}}$ .*

3. *A hybrid logic*

$$\mathbf{HL}_{\mathcal{CS}} \in \{\mathbf{T}_n \mathbf{LP}_{\mathcal{CS}}, \mathbf{S4}_n \mathbf{LP}_{\mathcal{CS}}, \mathbf{S5}_n \mathbf{LP}_{\mathcal{CS}}\}$$

with a decidable schematic  $\mathcal{CS}$  is sound and complete w.r.t. the class of its finitary models  $\mathcal{C}_{\text{HL}_{\mathcal{CS}}}$ .

*Note 4.4.10.* An additional requirement in Case 2 for  $\mathcal{CS}$  to be axiomatically appropriate is inherited from Theorem 3.3.14.2: without it there is no completeness whatsoever, let alone completeness w.r.t. finitary models.

*Proof.* Since finitary models are actual F-models (AF-models), soundness follows from Theorem 3.3.14 (from Theorem 3.5.15 respectively).

We will, therefore, prove completeness in the following formulation:

$$\mathsf{L} \not\models F \quad \Longrightarrow \quad (\exists \ulcorner \mathfrak{M} \urcorner) \quad \mathfrak{M} \not\models F ,$$

where  $\mathsf{L}$  stands for any justification or hybrid logic considered in the theorem,  $\mathfrak{M}$  is a finitary model for that logic that can be encoded by  $\ulcorner \mathfrak{M} \urcorner$ .

We will once again resort to maximal consistent sets construction of a canonical model. But this time, to keep the number of such sets finite, we will focus our attention on sets of subformulas of the given  $F$ .

**Definition 4.4.11.** Let  $\text{Sub}(F)$  be the *set of all subformulas of  $F$* ,

namely the smallest set of formulas such that

$$F \in \text{Sub}(F) \quad (4.4.7)$$

$$G \rightarrow H \in \text{Sub}(F) \quad \Longrightarrow \quad G \in \text{Sub}(F) \quad \text{and} \quad H \in \text{Sub}(F) \quad (4.4.8)$$

$$t:G \in \text{Sub}(F) \quad \Longrightarrow \quad G \in \text{Sub}(F) \quad (4.4.9)$$

$$K_i G \in \text{Sub}(F) \quad \Longrightarrow \quad G \in \text{Sub}(F) \quad (4.4.10)$$

◀

**Definition 4.4.12.** Let us define two types of *extended subformula sets*

$$\text{Sub}^\neg(F) = \text{Sub}(F) \cup \{\neg G \mid G \in \text{Sub}(F)\} \quad (4.4.11)$$

$$\text{Sub}_n^\square(F) = \text{Sub}^\neg(F) \cup \{K_i t:G, \neg K_i t:G \mid i = 1, \dots, n, \quad t:G \in \text{Sub}(F)\} \quad (4.4.12)$$

◀

**Lemma 4.4.13.** *All three subformula sets are linear in the size of  $F$ :*

$$|\text{Sub}(F)| = O(|F|)$$

$$|\text{Sub}^\neg(F)| = O(|F|)$$

$$|\text{Sub}_n^\square(F)| = O(|F|)$$

*Proof.* The set of subformulas  $\text{Sub}(F)$  has no more elements than the number of main connectives in  $F$ , which is no larger than  $|F|$ .

The size of  $\text{Sub}^\neg(F)$  is twice that of  $\text{Sub}(F) \leq |F|$ .

The size of  $Sub_n^\square(F)$  is at most  $(2n + 2)$  times larger than the size of  $|F|$ .

□

All maximal  $L$ -consistent sets from Def. 2.6.1 are, of course, infinite. We will, therefore, use the relativized version of maximal consistency from Def. 2.6.3 with  $X$  being one of the subformula sets. In that case we can further refine Lemma 2.6.4.

**Lemma 4.4.14.** *Let*

- $L$  be one of pure justification logics and  $X$  be  $Sub^\neg(F)$ , or
- $L$  be  $T_nLP_{CS}$ ,  $S4_nLP_{CS}$ , or  $S5_nLP_{CS}$  and  $X$  be  $Sub_n^\square(F)$

for some formula  $F$ . Maximal  $L$ -consistent sets relative to  $X$  exist and for any such set  $\Gamma \subseteq X$ :

1.  $\Gamma$  is finite.
2. For each formula  $G \in Sub(F)$  set  $\Gamma$  contains exactly one of  $G$  and  $\neg G$ .
3. If  $\Gamma \vdash_L G$  for some  $G \in X$ , then  $G \in \Gamma$ .
4. Set  $\Gamma$  is closed under modus ponens, i.e., for any formulas  $G$  and  $H$ , if  $G \rightarrow H \in \Gamma$ ,  $G \in \Gamma$ , then  $H \in \Gamma$ .

5. Set  $\Gamma$  is closed under conjunctions, i.e., for any formulas  $G$  and  $H$ , if  $G \in \Gamma$ ,  $H \in \Gamma$ , and  $G \wedge H \in \text{Sub}(F)$ , then  $G \wedge H \in \Gamma$ .
6.  $L \cap X \subseteq \Gamma$ .
7. For each  $\Delta \subseteq X$  that is  $L$ -consistent relative to  $X$ , there exists a set  $\Delta'$  that is maximal  $L$ -consistent relative to  $X$  such that  $X \supseteq \Delta' \supseteq \Delta$ .
8. For  $L \in \{\text{JT}_{CS}, \text{LP}_{CS}, \text{T}_n \text{LP}_{CS}, \text{S4}_n \text{LP}_{CS}, \text{S5}_n \text{LP}_{CS}\}$ , if  $t : G \in \Gamma$ , then  $G \in \Gamma$ .
9. For any hybrid logic  $L$ , if  $K_i G \in \Gamma$ , then  $G \in \Gamma$ .
10. In case  $X = \text{Sub}_n^\square(F)$ , if  $t : G \in \Gamma$ , then  $K_i t : G \in \Gamma$ ,  $1 \leq i \leq n$ .

*Proof.* We first prove that such maximal consistent sets exist. By Def. 4.4.12 of extended subformula sets,  $\{F, \neg F\} \subseteq X$ . Either  $\{F\}$  or  $\{\neg F\}$  must be  $L$ -consistent. Otherwise both  $L \vdash \neg F$  and  $L \vdash \neg\neg F$ , which would imply inconsistency of  $L$  itself. By Lemma 2.6.4.6, this consistent singleton set can be extended to a maximal consistent relative to  $X$  set, which will have at least one element.

1. The size of  $\Gamma \subseteq X$  cannot be larger than the size of  $X$ , which is linear in  $|F|$  by Lemma 4.4.13.

2. By Lemma 2.6.4.1 since  $Sub^\neg(F)$  contains all subformulas of  $F$  together with their negations.
3. By Lemma 2.6.4.2
4. By Lemma 2.6.4.3: If  $G \rightarrow H \in \Gamma$ , then  $H \in \Gamma$  by (4.4.8).
5. By Lemma 2.6.4.4 since  $Sub(F) \subset Sub^\neg(F)$  and  $Sub(F) \subset Sub_n^\square(F)$ .
6. Identical to Lemma 2.6.4.5.
7. Identical to Lemma 2.6.4.6.
8.  $t:G \in \Gamma \subseteq X$ , hence  $t:G \in Sub(F)$  and  $G \in Sub(F)$ . For these logics  $\mathsf{L} \vdash t:G \rightarrow G$ , hence  $\Gamma \vdash_{\mathsf{L}} G$ . By Clause 3,  $G \in \Gamma$ .
9. If  $K_iG \in \Gamma \subseteq X$ , then either  $K_iG \in Sub(F)$  or  $G = t:H$ , in which case  $t:H \in Sub(F)$  for some  $t$ . In either case  $G \in Sub(F)$ . For hybrid logics  $\mathsf{L} \vdash K_iG \rightarrow G$ , so  $\Gamma \vdash_{\mathsf{L}} G$ . By Clause 3,  $G \in \Gamma$ .
10. If  $t:G \in \Gamma \subseteq Sub_n^\square(F)$ , then  $t:G \in Sub(F)$  and  $K_it:G \in Sub_n^\square(F)$ . For hybrid logics  $\mathsf{L} \vdash t:G \rightarrow K_it:G$ , so  $\Gamma \vdash_{\mathsf{L}} K_it:G$ . By Clause 3,  $K_it:G \in \Gamma$ .

In Clause 10, the derivation of  $t:G \rightarrow K_it:G$  in hybrid logics is easy to obtain from Positive Introspection  $t:G \rightarrow !t:t:G$  and Connection Principle

$!t:t:G \rightarrow K_i t:G$  by Syllogism.  $\square$

We are now ready to construct the finitary canonical model with the domain being the set of all maximal consistent sets relative to the given formula  $F$ .

*Note 4.4.15.* Unlike the case of infinite canonical models, we will have to take extra precautions to ensure transitivity of the frame in these finitary models. In the proofs of Theorems 3.3.14 and 3.5.15, transitivity of, say,  $R$  was guaranteed by the fact that  $t : G \in \Gamma$  entails  $!t : t : G \in \Gamma$  for any maximal consistent set  $\Gamma$ . For a finitary maximal consistent  $\Gamma$  this may not hold simply because  $!t:t:G$  may not be a subformula of  $F$ .

We will, therefore, need to adjust the definition of  $\Gamma^\sharp$  appropriately:

**Definition 4.4.16.** Let  $\Gamma$  be a set of pure justification formulas.

$$\Gamma^b = \{G, t:G \mid t:G \in \Gamma\}$$



**Definition 4.4.17.** Let  $\Gamma$  be a set of hybrid formulas.

$$\Gamma^{\sharp_i} = \{G \mid K_i G \in \Gamma\}$$



**Definition 4.4.18.** Let  $\Gamma$  be a set of hybrid formulas.

$$\Gamma^{b_i} = \{K_i G, G \mid K_i G \in \Gamma\}$$



**Definition 4.4.19.** The *finitary canonical model* for a pure justification logic  $\mathbf{JL}_{CS}$  relative to a formula  $F$  is a quadruple

$$\mathfrak{M} = (W, R, V, \mathcal{A})$$

defined as follows:

$$W = \{\Gamma \mid \Gamma \text{ is maximal } \mathbf{JL}_{CS}\text{-consistent relative to } Sub^\neg(F)\} \quad (4.4.13)$$

$$\Gamma R \Delta \Leftrightarrow \Gamma^\# \subseteq \Delta \quad \text{for } \mathbf{J}_{CS}, \mathbf{JD}_{CS}, \mathbf{JT}_{CS} \quad (4.4.14)$$

$$\Gamma R \Delta \Leftrightarrow \Gamma^b \subseteq \Delta \quad \text{for } \mathbf{J4}_{CS}, \mathbf{JD4}_{CS}, \mathbf{LP}_{CS} \quad (4.4.15)$$

$$V(p) = \{\Gamma \in W \mid p \in \Gamma\} \quad (4.4.16)$$

$\mathcal{A} =$  the minimal admissible for  $\mathbf{JL}_{CS}$  evidence function

$$\text{based on } \mathcal{B}(t, G) = \{\Gamma \in W \mid t : G \in \Gamma\} . \quad (4.4.17)$$



**Definition 4.4.20.** The *finitary canonical model* for a hybrid logic  $\mathbf{HL}_{CS} \in \{\mathbf{T}_n \mathbf{LP}_{CS}, \mathbf{S4}_n \mathbf{LP}_{CS}, \mathbf{S5}_n \mathbf{LP}_{CS}\}$  relative to a formula  $F$  is a tuple

$$\mathfrak{M} = (W, R_e, R_1, \dots, R_n, V, \mathcal{A})$$

defined as follows:

$$W = \{ \Gamma \mid \Gamma \text{ is maximal } \mathbf{HL}_{\mathcal{CS}}\text{-consistent rel. to } \mathit{Sub}_n^\square(F) \} \quad (4.4.18)$$

$$\Gamma R_e \Delta \Leftrightarrow \Gamma^b \subseteq \Delta \quad (4.4.19)$$

$$\Gamma R_i \Delta \Leftrightarrow \Gamma^{\sharp_i} \subseteq \Delta \quad \text{for } \mathbf{T}_n \mathbf{LP}_{\mathcal{CS}} \quad (4.4.20)$$

$$\Gamma R_i \Delta \Leftrightarrow \Gamma^{b_i} \subseteq \Delta \quad \text{for } \mathbf{S4}_n \mathbf{LP}_{\mathcal{CS}} \quad (4.4.21)$$

$$\Gamma R_i \Delta \Leftrightarrow \Gamma^{b_i} \subseteq \Delta \text{ and } \Delta^{b_i} \subseteq \Gamma \quad \text{for } \mathbf{S5}_n \mathbf{LP}_{\mathcal{CS}} \quad (4.4.22)$$

$$V(p) = \{ \Gamma \in W \mid p \in \Gamma \} \quad (4.4.23)$$

$\mathcal{A} =$  the minimal for  $\mathbf{HL}_{\mathcal{CS}}$  admissible evidence function

$$\text{based on } \mathcal{B}(t, G) = \{ \Gamma \in W \mid t : G \in \Gamma \} . \quad (4.4.24)$$

◀

We will now prove that the finitary canonical models so defined are indeed finitary models for their respective logics.

**Lemma 4.4.21.** *Let  $L$  be*

- a justification logic  $J_{\mathcal{CS}}, JT_{\mathcal{CS}}, J4_{\mathcal{CS}}, LP_{\mathcal{CS}},$
- a justification logic  $JD_{\mathcal{CS}}, JD4_{\mathcal{CS}}$  with an axiomatically appropriate  $\mathcal{CS},$
- or
- a hybrid logic  $T_n LP_{\mathcal{CS}}, S4_n LP_{\mathcal{CS}}, S5_n LP_{\mathcal{CS}}.$

Let  $F$  be a pure or hybrid justification formula respectively. Then, the finitary canonical model for formula  $F$  of logic  $\mathbf{L}$  is indeed a finitary model for  $\mathbf{L}$ .

*Proof.* We need to show that

1.  $W$  is finite.

All  $\Gamma \in W$  are subsets of one of the extended subformula sets of  $F$ , which are linear in  $|F|$ . The number of such subsets is at most  $2^{O|F|}$ .

2.  $W \neq \emptyset$ .

By Lemma 4.4.14.

3.  $R$  is reflexive (for  $\mathbf{JT}_{CS}$  and  $\mathbf{LP}_{CS}$ ).

For any  $t:G \in \Gamma$ , by Lemma 4.4.14.8,  $G \in \Gamma$ . Thus, both  $\Gamma^\# \subseteq \Gamma$  and  $\Gamma^\flat \subseteq \Gamma$ , and for all the logics  $\Gamma R \Gamma$  either by (4.4.14) or by (4.4.15).

4.  $R_e$  is reflexive (for hybrid logics).

Similar to the previous clause, using (4.4.19) for  $R_e$  instead of (4.4.14) and (4.4.15) for  $R$ .

5.  $R_i$  is reflexive (for hybrid logics).

For any  $K_i G \in \Gamma$ , by Lemma 4.4.14.9,  $G \in \Gamma$ . Thus, both  $\Gamma^{\#i} \subseteq \Gamma$  and  $\Gamma^{\flat i} \subseteq \Gamma$ , and for all the logics  $\Gamma R_i \Gamma$  by one of (4.4.20), (4.4.21), or (4.4.22).

6.  $R$  is transitive (for  $\mathbf{J4}_{CS}$ ,  $\mathbf{JD4}_{CS}$ , and  $\mathbf{LP}_{CS}$ ).

Let  $\Gamma R \Delta$  and  $\Delta R \Sigma$ . By (4.4.15),  $\Gamma^b \subseteq \Delta$  and  $\Delta^b \subseteq \Sigma$ . For any  $t:G \in \Gamma$ , we have  $t:G \in \Delta$  and  $\{t:G, G\} \subseteq \Sigma$  by Def. 4.4.16. Hence,  $\Gamma R \Sigma$  by (4.4.15).

7.  $R_e$  is transitive (for hybrid logics).

Similar to the previous clause, using (4.4.19) for  $R_e$  instead of (4.4.15) for  $R$ .

8.  $R_i$  is transitive (for  $\mathbf{S4}_n\mathbf{LP}_{CS}$  and  $\mathbf{S5}_n\mathbf{LP}_{CS}$ ).

Let  $\Gamma R_i \Delta$  and  $\Delta R_i \Sigma$ .

For  $\mathbf{S4}_n\mathbf{LP}_{CS}$ , by (4.4.21),  $\Gamma^{b_i} \subseteq \Delta$  and  $\Delta^{b_i} \subseteq \Sigma$ . For any  $K_i G \in \Gamma$ , we have  $K_i G \in \Delta$  and  $\{K_i G, G\} \subseteq \Sigma$  by Def. 4.4.18. Hence,  $\Gamma R_i \Sigma$  by (4.4.21).

For  $\mathbf{S5}_n\mathbf{LP}_{CS}$ , by (4.4.22), in addition  $\Delta^{b_i} \subseteq \Gamma$  and  $\Sigma^{b_i} \subseteq \Delta$ . For any  $K_i G \in \Sigma$ , we have  $K_i G \in \Delta$  and  $\{K_i G, G\} \subseteq \Gamma$  by Def. 4.4.18. Here both  $\Gamma^{b_i} \subseteq \Sigma$  and  $\Sigma^{b_i} \subseteq \Gamma$ . Hence,  $\Gamma R_i \Sigma$  by (4.4.22).

9.  $R_i$  is symmetric (for  $\mathbf{S5}_n\mathbf{LP}_{CS}$ ).

Let  $\Gamma R_i \Delta$ . By (4.4.22),  $\Gamma^{b_i} \subseteq \Delta$  and  $\Delta^{b_i} \subseteq \Gamma$ . Hence,  $\Delta R_i \Gamma$  by (4.4.22).

10.  $R_i \subseteq R_e$  (for hybrid logics).

Let  $\Gamma R_i \Delta$ . For any  $t:G \in \Gamma$ , by Lemma 4.4.14.10,  $K_it:G \in \Gamma$ , so that  $t:G \in \Delta$  by Lemma 4.4.14.9 and  $G \in \Delta$  by Lemma 4.4.14.8. Thus,  $\Gamma^b \subseteq \Delta$ , i.e.,  $\Gamma R_e \Delta$ .

11. *R is serial* (for  $\text{JD}_{\mathcal{CS}}$  and  $\text{JD4}_{\mathcal{CS}}$  with axiomatically appropriate  $\mathcal{CS}$ ).

Let  $\Gamma \in W$ . We need to show that there is  $\Delta \in W$  such that  $\Gamma R \Delta$ .

The set  $\Gamma$  itself is L-consistent. We claim that  $\Gamma^\sharp$  for  $L = \text{JD}_{\mathcal{CS}}$  and  $\Gamma^b$  for  $L = \text{JD4}_{\mathcal{CS}}$  are L-consistent too. In both cases we will use proofs by contradiction.

Suppose towards a contradiction that  $\Gamma^\sharp$  is not  $\text{JD}_{\mathcal{CS}}$ -consistent, i.e.,

$$G_1, \dots, G_k \vdash_{\text{JD}_{\mathcal{CS}}} \perp$$

for some  $s_j : G_j \in \Gamma$ ,  $j = 1, \dots, k$ . Internalizing this derivation by Lemma 3.2.22, which requires  $\mathcal{CS}$  to be axiomatically appropriate, we get

$$x_1 : G_1, \dots, x_k : G_k \vdash_{\text{JD}_{\mathcal{CS}}} t(x_1, \dots, x_k) : \perp$$

for some fresh justification variables  $x_1, \dots, x_k$  and some term  $t$ . The simultaneous substitution of  $s_j$  for  $x_j$  by Lemma 3.2.30, which again requires  $\mathcal{CS}$  to be axiomatically appropriate, will yield

$$s_1 : G_1, \dots, s_k : G_k \vdash_{\text{JD}_{\mathcal{CS}}} \bar{t}(s_1, \dots, s_k) : \perp$$

for some other term  $\bar{t}$ , obtained from  $t$  by possibly renaming constants.

Therefore,

$$\Gamma \vdash_{\text{JD}_{\mathcal{CS}}} \bar{t}(s_1, \dots, s_k) : \perp$$

and since  $\text{JD}_{\mathcal{CS}} \vdash \bar{t}(s_1, \dots, s_k) : \perp \rightarrow \perp$

$$\Gamma \vdash_{\text{JD}_{\mathcal{CS}}} \perp,$$

which contradict  $\text{JD}_{\mathcal{CS}}$ -consistency of  $\Gamma$ . This contradiction shows that

$\Gamma^\sharp$  is  $\text{JD}_{\mathcal{CS}}$ -consistent.

Suppose towards a contradiction that  $\Gamma^\flat$  is not  $\text{JD}4_{\mathcal{CS}}$ -consistent, i.e.,

$$G_1, \dots, G_k, \quad q_1 : H_1, \dots, q_l : H_l \vdash_{\text{JD}4_{\mathcal{CS}}} \perp$$

for some  $s_j : G_j \in \Gamma$ ,  $j = 1, \dots, k$  and some  $q_m : H_m \in \Gamma$ ,  $m = 1, \dots, l$ . Lifting this derivation by Lemma 3.2.25, which requires  $\mathcal{CS}$  to be axiomatically appropriate, we get

$$x_1 : G_1, \dots, x_k : G_k, \quad q_1 : H_1, \dots, q_l : H_l \vdash_{\text{JD}4_{\mathcal{CS}}} t(x_1, \dots, x_k, q_1, \dots, q_l) : \perp$$

for some fresh justification variables  $x_1, \dots, x_k$  and some term  $t$ . The simultaneous substitution of  $s_j$  for  $x_j$  by Lemma 3.2.30, which again requires  $\mathcal{CS}$  to be axiomatically appropriate, will yield

$$s_1 : G_1, \dots, s_k : G_k, \quad q_1 : H_1, \dots, q_l : H_l \vdash_{\text{JD}4_{\mathcal{CS}}} \bar{t}(s_1, \dots, s_k, q_1, \dots, q_l) : \perp$$

for some other term  $\bar{t}$ , obtained from  $t$  by possibly renaming constants.

Therefore,

$$\Gamma \vdash_{\text{JD4}_{\mathcal{CS}}} \bar{t}(s_1, \dots, s_k, q_1, \dots, q_l) : \perp$$

and since  $\text{JD4}_{\mathcal{CS}} \vdash \bar{t}(s_1, \dots, s_k, q_1, \dots, q_l) : \perp \rightarrow \perp$

$$\Gamma \vdash_{\text{JD4}_{\mathcal{CS}}} \perp,$$

which contradict  $\text{JD4}_{\mathcal{CS}}$ -consistency of  $\Gamma$ . This contradiction shows that

$\Gamma^{\flat}$  is  $\text{JD4}_{\mathcal{CS}}$ -consistent.

Whenever  $\Gamma \subseteq \text{Sub}^{\neg}(F)$ , both  $\Gamma^{\sharp} \subseteq \text{Sub}(F)$  and  $\Gamma^{\flat} \subseteq \text{Sub}(F)$ .

Thus, by Lemma 4.4.14.7, either of L-consistent sets  $\Gamma^{\sharp}$  or  $\Gamma^{\flat}$  can be extended to a maximal L-consistent relative to  $\text{Sub}^{\neg}(F)$  set  $\Delta \in W$ .

For  $\text{JD}_{\mathcal{CS}}$ ,  $\Delta \supseteq \Gamma^{\sharp}$ , hence  $\Gamma R \Delta$ . For  $\text{JD4}_{\mathcal{CS}}$ ,  $\Delta \supseteq \Gamma^{\flat}$ , hence  $\Gamma R \Delta$ .

12. *V is finitely true.*

If  $V(p) \neq \emptyset$ , i.e.,  $(\exists \Gamma)\Gamma \in V(p)$ , then  $(\exists \Gamma)p \in \Gamma$ , which can only happen for  $p \in \text{Sub}(F)$ . Formula  $F$  has finitely many sentence letters occurring in it; hence,  $V$  is finitely true by Def. 4.2.1.

13. *B is a finitary possible evidence function.*

If  $\mathcal{B}(t, G) \neq \emptyset$ , i.e.,  $(\exists \Gamma)\Gamma \in \mathcal{B}(t, G)$ , then  $(\exists \Gamma)t : G \in \Gamma$ , which can only

happen for  $t:G \in \text{Sub}(F)$ . Since  $\text{Sub}(F)$  is a finite set,  $\mathcal{B}$  is finitary by Def. 4.4.1.

This completes the proof of Lemma 4.4.21 that the finitary canonical model is an actual model.  $\square$

We now prove the relativized

**Lemma 4.4.22 (Truth Lemma).** *Let  $\mathfrak{M}$  be a finitary canonical model for formula  $F$  constructed in Def. 4.4.19 or Def. 4.4.20. For any  $G \in \text{Sub}(F)$ ,*

$$\mathfrak{M}, \Gamma \Vdash G \quad \iff \quad G \in \Gamma$$

*Proof.* Induction on complexity of formula  $G$ :

$G = p$ . *A sentence letter.* Follows directly from (4.4.16) and (3.3.15) for F-models or from (4.4.23) and (3.5.4) for AF-models, in other words

$$\mathfrak{M}, \Gamma \Vdash p \quad \iff \quad \Gamma \in V(p) \quad \iff \quad p \in \Gamma$$

Boolean cases are trivial.

$G = t:H$ . Let  $t:H \in \Gamma$ . First of all, by (4.4.17) or (4.4.24),  $\Gamma \in \mathcal{B}(t, H)$ . Since  $\mathcal{A}$  is based on  $\mathcal{B}$ , also  $\Gamma \in \mathcal{A}(t, H)$ .

For F-models  $H \in \Delta$  for any  $\Delta$  that is  $R$ -accessible from  $\Gamma$  by (4.4.14) or (4.4.15). For AF-models  $H \in \Delta$  for any  $\Delta$  that is  $R_e$ -accessible from  $\Gamma$  by (4.4.19).

In either case, by IH,  $\mathfrak{M}, \Delta \Vdash H$  for any  $\Gamma R \Delta$  ( $\Gamma R_e \Delta$  respectively).

Combined with  $\Gamma \in \mathcal{A}(t, H)$ , this yields  $\mathfrak{M}, \Gamma \Vdash t : H$ .

Let  $t : H \notin \Gamma$ . Then, by (4.4.17) or (4.4.24),  $\Gamma \notin \mathcal{B}(t, H)$ . But can it happen that despite it  $\Gamma \in \mathcal{A}(t, H)$ ? The answer is negative. We will prove it by contradiction.

Suppose towards a contradiction that  $\Gamma \in \mathcal{A}(t, H)$ . This implies that by Theorem 3.3.41 and Theorem 3.5.20

- $\mathcal{B}_\Gamma^* \vdash_{*cS} *(t, H)$  for  $J_{cS}, JD_{cS}, JT_{cS}$  per (3.3.37),
- $\mathcal{B}_\Gamma^* \cup \bigcup_{\Delta R \Gamma} \mathcal{B}_\Delta^* \vdash_{*l_{cS}} *(t, H)$  for  $J4_{cS}, JD4_{cS}, LP_{cS}$  per (3.3.38),
- or
- $\bigcup_{\Delta R_e \Gamma} \mathcal{B}_\Delta^* \vdash_{*l_{cS}} *(t, H)$  for  $\top_n LP_{cS}, S4_n LP_{cS}, S5_n LP_{cS}$  per (3.5.9).

In the latter two cases  $\mathcal{B}_\Delta^* \subseteq \mathcal{B}_\Gamma^*$  for any  $\Delta R \Gamma$  or  $\Delta R_e \Gamma$ . Indeed, if  $*(s, E) \in \mathcal{B}_\Delta^*$ , i.e.,  $\Delta \in \mathcal{B}(s, E)$ , then  $s : E \in \Delta$  by (4.4.17) or (4.4.24). Thus,  $s : E \in \Gamma$  by (4.4.15) or (4.4.19). Therefore,  $\Gamma \in \mathcal{B}(s, E)$  by (4.4.17) or (4.4.24). In other words,  $*(s, E) \in \mathcal{B}_\Gamma^*$ . Thus, in all three cases

$$\mathcal{B}_\Gamma^* \vdash_* *(t, H)$$

for the respective  $\vdash_*$ . By Lemma 3.4.8

$$(\mathcal{B}_\Gamma^*)^\dagger \vdash_{\mathbf{L}} t:H ,$$

where  $\mathbf{L}$  is the respective logic. Note that  $(\mathcal{B}_\Gamma^*)^\dagger \subseteq \Gamma$ . Indeed,

$$s:E \in (\mathcal{B}_\Gamma^*)^\dagger \Leftrightarrow *(s,E) \in \mathcal{B}_\Gamma^* \Leftrightarrow \Gamma \in \mathcal{B}(s,E) \Leftrightarrow s:E \in \Gamma$$

Hence,

$$\Gamma \vdash_{\mathbf{L}} t:H$$

and by Lemma 4.4.14.3,  $t:H \in \Gamma$ . This contradiction shows that  $\Gamma \notin \mathcal{A}(t,H)$ .

Therefore,  $\mathfrak{M}, \Gamma \not\models t:H$ .

$G = K_i H$ . (Only for hybrid logics.) Let  $K_i H \in \Gamma$ . Then,  $H \in \Delta$  for any  $\Delta$  that is  $R_i$ -accessible from  $\Gamma$  by one of (4.4.20), (4.4.21), or (4.4.22).

In all cases, by IH,  $\mathfrak{M}, \Delta \models H$  for any  $\Gamma R_i \Delta$ , which yields  $\mathfrak{M}, \Gamma \models K_i H$ .

Let  $K_i H \notin \Gamma$ . We need to prove the existence of a  $\Delta$  that is  $R_i$ -accessible from  $\Gamma$  but does not contain  $H$ , which by IH entails that  $H$  is false at  $\Delta$ . The construction of  $\Delta$  depends on which hybrid logic we are dealing with.

$\mathsf{T}_n\mathsf{LP}_{\mathcal{CS}}$ . We claim that  $\Gamma^{\sharp_i} \cup \{\neg H\}$  is  $\mathsf{T}_n\mathsf{LP}_{\mathcal{CS}}$ -consistent. Proof by contradiction. If not, then

$$E_1, \dots, E_k \vdash_{\mathsf{T}_n\mathsf{LP}_{\mathcal{CS}}} H$$

for some  $K_i E_m \in \Gamma$ ,  $m = 1, \dots, k$ . Then, by Deduction Theorem for  $\mathsf{T}_n\mathsf{LP}_{\mathcal{CS}}$ ,

$$\mathsf{T}_n\mathsf{LP}_{\mathcal{CS}} \vdash E_1 \rightarrow (\dots \rightarrow (E_k \rightarrow H) \dots) .$$

Using  $K_i$ -necessitation and distributing  $K_i$  through implication in the usual modal manner, we get

$$\mathsf{T}_n\mathsf{LP}_{\mathcal{CS}} \vdash K_i E_1 \rightarrow (\dots \rightarrow (K_i E_k \rightarrow K_i H) \dots)$$

and

$$K_i E_1, \dots, K_i E_k \vdash_{\mathsf{T}_n\mathsf{LP}_{\mathcal{CS}}} K_i H .$$

Therefore,

$$\Gamma \vdash_{\mathsf{T}_n\mathsf{LP}_{\mathcal{CS}}} K_i H .$$

Given that  $K_i H \in \mathit{Sub}(F)$ , by Lemma 4.4.14.3,  $K_i H \in \Gamma$ . This contradiction shows that  $\Gamma^{\sharp_i} \cup \{\neg H\}$  is  $\mathsf{T}_n\mathsf{LP}_{\mathcal{CS}}$ -consistent.

Since  $K_i H \in \mathit{Sub}(F)$ , clearly  $\neg H \in \mathit{Sub}_n^\square(F)$ . Therefore,  $\Gamma^{\sharp_i} \cup \{\neg H\} \subseteq \mathit{Sub}_n^\square(F)$  and it can be extended by Lemma 4.4.14.7 to a maximal  $\mathsf{T}_n\mathsf{LP}_{\mathcal{CS}}$ -consistent relative to  $\mathit{Sub}_n^\square(F)$  set  $\Delta$ .

By (4.4.20),  $\Gamma R_i \Delta$ .

Clearly,  $H \in \text{Sub}(F)$ . Since  $\neg H \in \Delta$ , by Lemma 4.4.14.2,  $H \notin \Delta$ .

So by IH,  $\mathfrak{M}, \Delta \not\models H$ .

Thus,  $\mathfrak{M}, \Gamma \not\models K_i H$ .

$\mathbf{S4}_n\text{LP}_{CS}$ . We claim that  $\Gamma^{b_i} \cup \{\neg H\}$  is  $\mathbf{S4}_n\text{LP}_{CS}$ -consistent. Proof by contradiction. If not, then

$$E_1, \dots, E_k, \quad K_i D_1, \dots, K_i D_l \vdash_{\mathbf{S4}_n\text{LP}_{CS}} H$$

for some  $K_i E_m \in \Gamma$ ,  $m = 1, \dots, k$ , and some  $K_i D_j \in \Gamma$ ,  $j = 1, \dots, l$ . Again using Deduction Theorem,  $K_i$ -necessitation, distributing  $K_i$  through implications, and using *modus ponens*, we get

$$K_i E_1, \dots, K_i E_k, \quad K_i K_i D_1, \dots, K_i K_i D_l \vdash_{\mathbf{S4}_n\text{LP}_{CS}} K_i H .$$

Since  $\mathbf{S4}_n\text{LP}_{CS} \vdash K_i D_j \rightarrow K_i K_i D_j$ ,  $j = 1, \dots, l$ , we can strip the second modalities:

$$K_i E_1, \dots, K_i E_k, \quad K_i D_1, \dots, K_i D_l \vdash_{\mathbf{S4}_n\text{LP}_{CS}} K_i H .$$

Therefore,

$$\Gamma \vdash_{\mathbf{S4}_n\text{LP}_{CS}} K_i H .$$

Given that  $K_i H \in \text{Sub}(F)$ , by Lemma 4.4.14.3,  $K_i H \in \Gamma$ . This contradiction shows that  $\Gamma^{b_i} \cup \{\neg H\}$  is  $\mathbf{S4}_n\text{LP}_{\mathcal{CS}}$ -consistent.

Since  $K_i H \in \text{Sub}(F)$ , clearly  $\neg H \in \text{Sub}_n^\square(F)$ . Therefore,  $\Gamma^{b_i} \cup \{\neg H\} \subseteq \text{Sub}_n^\square(F)$  and it can be extended by Lemma 4.4.14.7 to a maximal  $\mathbf{S4}_n\text{LP}_{\mathcal{CS}}$ -consistent relative to  $\text{Sub}_n^\square(F)$  set  $\Delta$ .

By (4.4.21),  $\Gamma R_i \Delta$ .

Clearly,  $H \in \text{Sub}(F)$ . Since  $\neg H \in \Delta$ , by Lemma 4.4.14.2,  $H \notin \Delta$ .

So by IH,  $\mathfrak{M}, \Delta \not\models H$ .

Thus,  $\mathfrak{M}, \Gamma \not\models K_i H$ .

$\mathbf{S5}_n\text{LP}_{\mathcal{CS}}$ . We claim that

$$Z = \Gamma^{b_i} \cup \{\neg K_i C \mid \neg K_i C \in \Gamma\} \cup \{\neg H\}$$

is  $\mathbf{S5}_n\text{LP}_{\mathcal{CS}}$ -consistent. Proof by contradiction. If not, then

$$E_1, \dots, E_k, \quad K_i D_1, \dots, K_i D_l, \quad \neg K_i C_1, \dots, \neg K_i C_r \vdash_{\mathbf{S5}_n\text{LP}_{\mathcal{CS}}} H$$

for some  $K_i E_m \in \Gamma$ ,  $m = 1, \dots, k$ , some  $K_i D_j \in \Gamma$ ,  $j = 1, \dots, l$ , and some  $\neg K_i C_h \in \Gamma$ ,  $h = 1, \dots, r$ . Again using the Deduction Theorem,  $K_i$ -necessitation, distributing  $K_i$  through implications,

and using *modus ponens*, we get

$$K_i E_1, \dots, K_i E_k, \quad K_i K_i D_1, \dots, K_i K_i D_l,$$

$$K_i \neg K_i C_1, \dots, K_i \neg K_i C_r \vdash_{\mathbf{S5}_n \mathbf{LP}_{CS}} K_i H \ .$$

Since  $\mathbf{S5}_n \mathbf{LP}_{CS} \vdash K_i D_j \rightarrow K_i K_i D_j$ ,  $j = 1, \dots, l$ , and in addition  $\mathbf{S5}_n \mathbf{LP}_{CS} \vdash \neg K_i C_h \rightarrow K_i \neg K_i C_h$ ,  $h = 1, \dots, r$ , we can strip the second modalities:

$$K_i E_1, \dots, K_i E_k, \quad K_i D_1, \dots, K_i D_l,$$

$$\neg K_i C_1, \dots, \neg K_i C_r \vdash_{\mathbf{S5}_n \mathbf{LP}_{CS}} K_i H \ .$$

Therefore,

$$\Gamma \vdash_{\mathbf{S5}_n \mathbf{LP}_{CS}} K_i H \ .$$

Given that  $K_i H \in \text{Sub}(F)$ , by Lemma 4.4.14.3,  $K_i H \in \Gamma$ . This contradiction shows that  $Z$  is  $\mathbf{S5}_n \mathbf{LP}_{CS}$ -consistent.

Since  $K_i H \in \text{Sub}(F)$ , clearly  $\neg H \in \text{Sub}_n^\square(F)$ . So  $Z \subseteq \text{Sub}_n^\square(F)$  and it can be extended by Lemma 4.4.14.7 to a maximal  $\mathbf{S5}_n \mathbf{LP}_{CS}$ -consistent relative to  $\text{Sub}_n^\square(F)$  set  $\Delta$ .

Clearly,  $\Gamma^{b_i} \subseteq \Delta$ . To show that  $\Gamma R_i \Delta$ , according to (4.4.22), we also need to show that  $\Delta^{b_i} \subseteq \Gamma$ . Let  $K_i C \in \Delta \subseteq \text{Sub}_n^\square$ . If  $K_i C \notin \Gamma$ , it would be that  $\neg K_i C \in \Gamma$  by Lemma 2.6.4.1. But then

$\neg K_i C \in Z \subseteq \Delta$ , which contradicts  $K_i C \in \Delta$ . This contradiction shows that  $K_i C \in \Gamma$ , in which case  $C \in \Gamma$  by Lemma 4.4.14.9.

This completes the proof that  $\Delta^{b_i} \subseteq \Gamma$ , and hence that  $\Gamma R_i \Delta$ .

Clearly,  $H \in \text{Sub}(F)$ . Since  $\neg H \in \Delta$ , by Lemma 4.4.14.2,  $H \notin \Delta$ .

So by IH,  $\mathfrak{M}, \Delta \not\models H$ .

Thus,  $\mathfrak{M}, \Gamma \not\models K_i H$ .

For all hybrid logics we have shown that  $K_i H \notin \Gamma$  entails  $\mathfrak{M}, \Gamma \not\models K_i H$ .

This completes the proof of the Truth Lemma 4.4.22.  $\square$

We are finally ready to finish the completeness part of the proof of Theorem 4.4.9. Take any formula  $F$  that is not derivable in logic  $L$ . Let  $\mathfrak{M}$  be the finitary canonical model relative to  $\neg F$ . By Lemma 4.4.21,  $\mathfrak{M}$  is a finitary model. Since  $L \not\vdash F$ , the set  $\{\neg F\}$  is  $L$ -consistent by Lemma 2.6.2.7. Clearly,  $\neg F \in \text{Sub}(\neg F)$ , so there must be a maximal  $L$ -consistent relative to  $\text{Sub}_n^\square(F)$  or to  $\text{Sub}_n^\square(\neg F)$  set  $\Gamma \ni \neg F$ . This  $\Gamma$  is one of the worlds of the canonical model  $\mathfrak{M}$ . By the Truth Lemma 4.4.22,  $\mathfrak{M}, \Gamma \Vdash \neg F$ , hence

$$\mathfrak{M}, \Gamma \not\models F ,$$

i.e.,  $F$  is refuted in one of the finitary models. Theorem 4.4.9 is proven.  $\square$

As a corollary, we immediately obtain

**Corollary 4.4.23.**

1. *A pure justification logic*

$$\text{JL}_{CS} \in \{\text{J}_{CS}, \text{JT}_{CS}, \text{J4}_{CS}, \text{LP}_{CS}\}$$

*with a decidable schematic CS,*

2. *A pure justification logic*

$$\text{JL}_{CS} \in \{\text{JD}_{CS}, \text{JD4}_{CS}\}$$

*with a decidable, schematic, and axiomatically appropriate CS, and*

3. *A hybrid logic*

$$\text{HL}_{CS} \in \{\text{T}_n \text{LP}_{CS}, \text{S4}_n \text{LP}_{CS}, \text{S5}_n \text{LP}_{CS}\}$$

*with a decidable schematic CS*

*all have the finitary model property.*

Usually, finite axiomatizability is sufficient to conclude that the logic is recursively enumerable. Unfortunately, the hidden assumption underlying this transition is that there are only finitely many effective inference rules, which is true for common modal logics. The  $\text{R4}_{CS}$  and  $\text{R4}_{CS}^!$  do not fit into that paradigm because they do not require assumptions. They are, in fact,

a lot like axioms. So we need to be careful about claiming  $\text{JL}_{CS}$  or  $\text{HL}_{CS}$  to be recursively enumerable.

**Lemma 4.4.24.** *Let  $L$  be a pure or hybrid justification logic and  $CS$  be a constant specification for it. If  $CS$  is recursively enumerable, the set of theorems of  $L_{CS}$  is also recursively enumerable.*

*Proof.* We will briefly outline the procedure. The set of axioms is clearly recursively enumerable (RE). If  $\text{R4}_{CS}^!$  is used, then the set of all formulas obtained by it is still RE. Create an enumeration of all theorems by taking the next axiom, then next  $\text{R4}_{CS}^!$ -formula, applying *modus ponens* to all theorems obtained so far in all possible ways, for hybrid logics apply all modal rules to all theorems obtained so far, add the next axiom, etc.  $\square$

**Theorem 4.4.25.**

1. *A pure justification logic*

$$\text{JL}_{CS} \in \{\text{J}_{CS}, \text{JT}_{CS}, \text{J4}_{CS}, \text{LP}_{CS}\}$$

*with a decidable schematic  $CS$ ,*

2. *A pure justification logic*

$$\text{JL}_{CS} \in \{\text{JD}_{CS}, \text{JD4}_{CS}\}$$

*with a decidable, schematic, and axiomatically appropriate  $CS$ , and*

## 3. A hybrid logic

$$\text{HL}_{\mathcal{CS}} \in \{\text{T}_n\text{LP}_{\mathcal{CS}}, \text{S4}_n\text{LP}_{\mathcal{CS}}, \text{S5}_n\text{LP}_{\mathcal{CS}}\}$$

with a decidable schematic  $\mathcal{CS}$

all are decidable.

*Proof.* By Lemma 4.4.24, each logic is recursively enumerable. By Cor. 4.4.23 they have finitary model property. Hence, by Theorem 4.3.3 these logics are decidable.  $\square$

**Theorem 4.4.26.** *Justification logics J, JD, JT, J4, JD4, LP and hybrid logics  $\text{T}_n\text{LP}$ ,  $\text{S4}_n\text{LP}$ ,  $\text{S5}_n\text{LP}$  are decidable.*

*Proof.* The total constant specification  $\mathcal{TCS}$  for each of these logics is clearly decidable, schematic, and axiomatically appropriate.  $\square$

**Theorem 4.4.27.** *Justification logics  $\text{J}_0$ ,  $\text{JT}_0$ ,  $\text{J4}_0$ ,  $\text{LP}_0$  and hybrid logics  $\text{T}_n\text{LP}_0$ ,  $\text{S4}_n\text{LP}_0$ ,  $\text{S5}_n\text{LP}_0$  are decidable.*

*Proof.* The empty constant specification  $\mathcal{CS} = \emptyset$  for each of these logics is clearly decidable and schematic.  $\square$

**Theorem 4.4.28.**

1. A pure justification logic

$$\mathsf{L}_{\mathcal{CS}} \in \{\mathsf{J}_{\mathcal{CS}}, \mathsf{JT}_{\mathcal{CS}}, \mathsf{J4}_{\mathcal{CS}}, \mathsf{LP}_{\mathcal{CS}}\}$$

with a decidable almost schematic  $\mathcal{CS}$ ,

2. A pure justification logic

$$\mathsf{JL}_{\mathcal{CS}} \in \{\mathsf{JD}_{\mathcal{CS}}, \mathsf{JD4}_{\mathcal{CS}}\}$$

with a decidable, almost schematic, and axiomatically appropriate  $\mathcal{CS}$ ,

and

3. A hybrid logic

$$\mathsf{L}_{\mathcal{CS}} \in \{\mathsf{T}_n\mathsf{LP}_{\mathcal{CS}}, \mathsf{S4}_n\mathsf{LP}_{\mathcal{CS}}, \mathsf{S5}_n\mathsf{LP}_{\mathcal{CS}}\}$$

with an almost schematic decidable  $\mathcal{CS}$

all are decidable.

*Proof.* Since  $\mathcal{CS}$  is almost schematic,  $\mathcal{CS} = \mathcal{CS}_1 \cup \mathcal{CS}_2$ , where  $\mathcal{CS}_1$  is schematic,  $\mathcal{CS}_2$  is finite, and  $\mathcal{CS}_1 \cap \mathcal{CS}_2 = \emptyset$ . Derivability in  $\mathsf{L}_{\mathcal{CS}}$  can be reduced to derivability in  $\mathsf{L}_{\mathcal{CS}_1}$  by the Deduction Theorem:

$$\mathsf{L}_{\mathcal{CS}} \vdash F \quad \iff \quad \mathcal{CS}_2 \vdash_{\mathsf{L}_{\mathcal{CS}_1}} F \quad \iff \quad \mathsf{L}_{\mathcal{CS}_1} \vdash \bigwedge \mathcal{CS}_2 \rightarrow F$$

$\mathcal{CS}_2$  is decidable as any finite set; hence,  $\mathcal{CS}_1 = \mathcal{CS} \setminus \mathcal{CS}_2$  is decidable. In Clauses 1 and 3, derivability in  $L_{\mathcal{CS}_1}$  is decidable by Theorem 4.4.25. Therefore, so is derivability in  $L_{\mathcal{CS}}$ .

For Clause 2 we additionally need to prove that  $\mathcal{CS}_1$  is axiomatically appropriate. Suppose it is not, i.e., there is an axiom  $A$  such that  $c : A \notin \mathcal{CS}_1$  for any justification constant  $c$ . Since  $\mathcal{CS}_1$  is schematic, for any axiom  $A'$  from the same axiom scheme as  $A$  we would also have  $c : A' \notin \mathcal{CS}_1$ . Each axiom scheme has infinitely many instances. Hence, the finite constant specification  $\mathcal{CS}_2$  cannot provide justification constants for all axioms  $A'$  not justified in  $\mathcal{CS}_1$ . This contradiction shows that  $\mathcal{CS}_1$  is axiomatically appropriate. Thus, Theorem 4.4.25 is once again applicable.  $\square$

**Corollary 4.4.29.** *Justification logics  $J_{\mathcal{CS}}$ ,  $J\mathbb{T}_{\mathcal{CS}}$ ,  $J4_{\mathcal{CS}}$ ,  $LP_{\mathcal{CS}}$  and hybrid logics  $T_nLP_{\mathcal{CS}}$ ,  $S4_nLP_{\mathcal{CS}}$ ,  $S5_nLP_{\mathcal{CS}}$  with finite  $\mathcal{CS}$ 's are decidable.*

*Proof.* A finite  $\mathcal{CS}$  is almost schematic since  $\mathcal{CS} = \emptyset \cup \mathcal{CS}$  and the empty constant specification is schematic. Any finite set is decidable. The statement follows by Cor. 4.4.28.  $\square$

## 4.5 Undecidability Results

The requirement for  $\mathcal{CS}$  to be schematic cannot be dropped from Theorem 4.4.25.

**Theorem 4.5.1.** *Let  $\mathsf{L}$  be any pure or hybrid justification logic. There exists a decidable  $\mathcal{CS}$  for  $\mathsf{L}$  such that  $\mathsf{L}_{\mathcal{CS}}$  is undecidable.*

*Proof.* The proof is by reducing the Halting Problem to provability in  $\mathsf{L}_{\mathcal{CS}}$  for a particular  $\mathcal{CS}$ . Let  $T_i$  stand for the  $i$ th Turing machine with one input; let  $T_i(m) \downarrow$  mean that  $T_i$  halts on input  $m$ . Let  $A_1, A_2, \dots$  be an effective enumeration of all axioms of  $\mathsf{L}$ . Consider the following  $\mathcal{CS}$ :

$$\begin{aligned} \mathcal{CS} = & \{a : (A_i \rightarrow (A_j \rightarrow A_i)) \mid T_i(i) \downarrow \text{ after at most } j \text{ steps}\} \cup \\ & \{b : A_i \mid i = 1, 2, \dots\} . \end{aligned}$$

Clearly this  $\mathcal{CS}$  is decidable. At the same time, it can be easily shown that

$$\mathsf{L}_{\mathcal{CS}} \vdash (a \cdot b) \cdot b : A_i \quad \iff \quad T_i(i) \downarrow .$$

The right side of this equivalence is the Halting Problem, which is known to be undecidable.  $\square$

*Note 4.5.2.* The constant specification  $\mathcal{CS}$  in the proof involves only two proof constants,  $a$  and  $b$ . A slightly more complex construction can be used to produce an undecidable theory  $\mathsf{LP}_{\mathcal{CS}}$  with a decidable  $\mathcal{CS}$  involving only one constant.

*Note 4.5.3.* The  $\mathcal{CS}$  used in the proof of Theorem 4.5.1 is, of course, neither schematic nor almost schematic.

## 4.6 Historical Survey

Decidability of  $\text{LP}_{\mathcal{CS}}$  with any finite  $\mathcal{CS}$  was established by Sergei Artemov in [Art95].

Later Alexey Mkrtychev in [Mkr97] showed that  $\text{LP}_{\mathcal{CS}}$  with any schematic  $\mathcal{CS}$  is decidable. Since  $\mathcal{TCS}_{\text{LP}}$  is schematic, decidability of  $\text{LP}$  is an easy corollary.

Decidability of  $\text{J}_{\mathcal{CS}}$ ,  $\text{JT}_{\mathcal{CS}}$ , and  $\text{J4}_{\mathcal{CS}}$  with schematic  $\mathcal{CS}$  follows from the results of [Kuz00].

An example of an undecidable  $\text{LP}_{\mathcal{CS}}$  with decidable  $\mathcal{CS}$  was first presented in [Kuz05].

Decidability of  $\text{T}_n\text{LP}_{\mathcal{CS}}$ ,  $\text{S4}_n\text{LP}_{\mathcal{CS}}$  and  $\text{S5}_n\text{LP}_{\mathcal{CS}}$  with schematic  $\mathcal{CS}$  is a new result, although decidability of  $\text{S4LP}$  was proven in [Kuz06a].

Several decidability results for single-conclusion justification logic were obtained by Vladimir Krupski in [Kru97, Kru01, Kru06d, Kru06c].

Decidability of several hybrid logics describing arithmetical provability was established by Tatiana Yavorskaya (Sidon) in [Sid97, Yav01a] and by Sergei Artemov and Elena Nogina in [AN04]. The attachment to arithmetical interpretations leads to the requirement for  $\mathcal{CS}$  to be finite in all these logics, so these decidability results apply to finite  $\mathcal{CS}$  only.

Decidability of  $S4LPN_{\mathcal{CS}}$  with finite  $\mathcal{CS}$  was established in [AN04] using Kripke-style semantics.

# Chapter 5

## Complexity

### 5.1 Upper Bounds for Reflected Fragments

One of the staples of all decision procedures for pure and hybrid justification logics as well as for their reflected fragments is the use of minimal functions pioneered by Alexey Mkrtychev in [Mkr97]. Theorems 3.3.41 and 3.4.2 for pure justification logics and Theorems 3.5.20 and 3.5.23 for hybrid logics outline the relationship between minimal evidence functions, reflected fragments, and  $*$ -calculi. This relationship allowed Nikolai Krupski to show in [Kru03] that  $\text{rLP}$  is in NP. We will generalize this result to all pure and hybrid justification logics with a decidable almost schematic  $\mathcal{CS}$  and formulate it in terms of  $*$ -calculi. We will also extend the complexity estimate to derivations with hypotheses.

**Theorem 5.1.1.** *Let  $\mathcal{CS}$  be a decidable schematic constant specification for*

one of pure or hybrid justification logics.

1. There exists an NP algorithm for determining for any given finite set  $S$  of  $*$ -expressions and a given  $*(t, F)$  whether

$$S \vdash_{*c_S} *(t, F) .$$

2. There exists an NP algorithm for determining for any given finite set  $S$  of  $*$ -expressions and a given  $*(t, F)$  whether

$$S \vdash_{*!c_S} *(t, F) .$$

*Proof.* We will present two algorithms:  $*c_S$ -DERIVE and  $*!c_S$ -DERIVE, for the respective calculi that are essentially effective implementations of the decision procedure for checking whether  $w \in \mathcal{A}(t, F)$  for the minimal admissible evidence function  $\mathcal{A}$  based on a given finitary possible evidence function  $\mathcal{B}$  from the proof of Lemma 4.4.7.

As in that proof, we will use variables  $P, Q, \dots$  over formulas and variables over justification terms (not present explicitly). We will use letters  $X, Y, \dots$  to denote schemes. Each axiom scheme can be written as one formula in this extended language. Let us write  $F \in X$  if formula  $F$  is an instance of scheme  $X$ . We will also consider the empty scheme  $\emptyset$  for which  $F \notin \emptyset$  for all  $F$ .

We will view schemes both as formulas in extended language and sets of formulas in the basic language hoping that the reader will be able to disambiguate between these two uses.

**procedure**  $*_{\mathcal{CS}}\text{-DERIVE}\langle S, *(t, F)\rangle$ ;

1. For each occurrence of a subterm  $s$  in  $t$ , where  $*(s, G) \in S$  for some  $G$ , non-deterministically choose one of two symbols: ‘ $S$ ’ or ‘ $\vdash$ ’.

If ‘ $S$ ’ was chosen for an occurrence of  $s'$  of which  $s$  is a proper suboccurrence, change the chosen symbol for  $s$  to ‘ $\#$ ’ no matter what was chosen for  $s$  originally.

2. For each occurrence of operation  $+$  in  $t$ , non-deterministically choose one of two symbols: ‘ $l$ ’ or ‘ $r$ ’, unless this occurrence of  $+$  is contained within an occurrence of  $s$  for which ‘ $S$ ’ was chosen in Step 1.

3. For each occurrence of  $r = \underbrace{!!\dots!}_n c$  in  $t$  for a constant  $c$  and an integer  $n \geq 0$ , non-deterministically choose an axiom scheme  $X$  such that  $c:X \subseteq \mathcal{CS}$  and make the assignment

$$\begin{cases} \underbrace{!!\dots!}_n c \rightsquigarrow \underbrace{!!\dots!}_{n-1} c : \dots : !c : c : X & \text{for } n \geq 1, \quad \text{or} \\ c \rightsquigarrow X & \text{for } n = 0 \end{cases}$$

to this occurrence of  $\underbrace{!!\dots!}_n c$ , unless it is contained within an occurrence

of  $s$  for which ‘ $S$ ’ was chosen in Step 1 or within an occurrence of  $!r$  in  $t$ .

4. For each occurrence of a justification variable  $x$ , make the assignment

$$x \rightsquigarrow \emptyset$$

to this occurrence of  $x$ , unless it is contained within an occurrence of  $s$  for which ‘ $S$ ’ was chosen in Step 1.

5. For each occurrence of  $s$  for which ‘ $S$ ’ was chosen in Step 1, non-deterministically choose a formula  $G$  such that  $*(s, G) \in S$  and make the following assignment to this occurrence of  $s$ :

$$s \rightsquigarrow G .$$

**repeat** Steps 6–8 until an assignment is made to  $t$ .

6. Non-deterministically choose an occurrence of a subterm  $s_1 + s_2$  in  $t$  such that assignments to these occurrences of  $s_1$  and  $s_2$  have already been made:

$$s_1 \rightsquigarrow X_1 , \quad s_2 \rightsquigarrow X_2 ,$$

but no assignment has been made to this occurrence of  $s_1 + s_2$ . Make

the following assignment to this occurrence of  $s_1 + s_2$ :

$$\begin{cases} s_1 + s_2 \rightsquigarrow X_1 & \text{if 'l' was chosen for this + ;} \\ s_1 + s_2 \rightsquigarrow X_2 & \text{if 'r' was chosen for this + .} \end{cases}$$

7. Non-deterministically choose an occurrence of a subterm  $!s$  in  $t$  such that an assignment to this occurrence of  $s$  has already been made, but no assignment has been made to this occurrence of  $!s$ . Make the following assignment to this occurrence of  $!s$ :

$$!s \rightsquigarrow \emptyset .$$

8. Non-deterministically choose an occurrence of a subterm  $s_1 \cdot s_2$  in  $t$  such that assignments to these occurrences of  $s_1$  and  $s_2$  have already been made:

$$s_1 \rightsquigarrow Z_1 , \quad s_2 \rightsquigarrow X_2 ,$$

but no assignment has been made to this occurrence of  $s_1 \cdot s_2$ . Make the following assignment to this occurrence of  $s_1 \cdot s_2$ :

$$\begin{cases} s_1 \cdot s_2 \rightsquigarrow \emptyset & \text{if } Z_1 = \emptyset \text{ or } X_2 = \emptyset ; \\ s_1 \cdot s_2 \rightsquigarrow Y_1\sigma & \text{if } Z_1 = X_1 \rightarrow Y_1 \text{ and } \sigma = \text{mgu}(X_1, X_2) ; \\ s_1 \cdot s_2 \rightsquigarrow \emptyset & \text{if } Z_1 = X_1 \rightarrow Y_1 \text{ and } \neg\exists\text{mgu}(X_1, X_2) ; \\ s_1 \cdot s_2 \rightsquigarrow Q & \text{if } Z_1 = P \text{ and } X_2 \neq \emptyset ; \\ s_1 \cdot s_2 \rightsquigarrow \emptyset & \text{otherwise ,} \end{cases}$$

where  $P$  is any variable over formulas,  $Q$  is a fresh variable over formulas,  $X_1$  and  $Y_1$  are any schemes.

**end repeat**

Let  $X$  be the scheme assigned to  $t$ .

9. **return true** if  $F$  is unifiable with  $X$ .
10. **backtrack** and use other choices in Steps 1–5 if  $F$  is not unifiable with  $X$  or if  $X = \emptyset$ .
11. **return false** if all choices in Steps 1–5 are exhausted.

The procedure  $*!_{\mathcal{CS}}\text{-DERIVE}$  is obtained by replacing Steps 3 and 7 in  $*_{\mathcal{CS}}\text{-DERIVE}$  by the following steps

- 3<sup>!</sup>. For each occurrence of a constant  $c$  in  $t$ , non-deterministically choose an axiom scheme  $X$  such that  $c : X \subseteq \mathcal{CS}$  and make the assignment

$$c \rightsquigarrow X$$

to this occurrence of  $c$ , unless it is contained within an occurrence of  $s$  for which ‘ $S$ ’ was chosen in Step 1.

- 7<sup>!</sup>. Non-deterministically choose an occurrence of a subterm  $!s$  in  $t$  such that an assignment  $s \rightsquigarrow X$  has already been made to this occurrence of  $s$ , but no assignment has been made to this occurrence of  $!s$ . Make

the following assignment to this occurrence of  $!s$ :

$$\begin{cases} !s \rightsquigarrow s : X & \text{if } X \neq \emptyset ; \\ !s \rightsquigarrow \emptyset & \text{if } X = \emptyset . \end{cases}$$

**Lemma 5.1.2 (Correctness of  $*_{\mathcal{CS}}$ -DERIVE and  $*!_{\mathcal{CS}}$ -DERIVE).**

1.  $*_{\mathcal{CS}}\text{-DERIVE}\langle S, *(t, F) \rangle$  returns **true** iff  $S \vdash *_{\mathcal{CS}} *(t, F)$ .
2.  $*!_{\mathcal{CS}}\text{-DERIVE}\langle S, *(t, F) \rangle$  returns **true** iff  $S \vdash *!_{\mathcal{CS}} *(t, F)$ .

*Proof.* Note that no assignments are ever made to a proper suboccurrence of any occurrence of  $s$  for which ‘ $S$ ’ was chosen in Step 1. Hence, in Step 6, the occurrence of  $s_1 + s_2$  cannot be inside any such  $s$  so that some choice of ‘l’ or ‘r’ must have been made for this occurrence of  $+$  in Step 2. Having this choice made is necessary to decide what to assign to  $s_1 + s_2$ .

For the ‘only if’ direction we will show that

$$s \rightsquigarrow X \quad \Longrightarrow \quad (\forall G \in X) S \vdash_* *(s, G) ,$$

where  $\vdash_*$  corresponds to the procedure used. We will use an induction over the assignments made by the procedure.

Step 3. For  $c : X \subseteq \mathcal{CS}$  and any axiom  $A$  from scheme  $X$ , by  $*\mathcal{CS}^!$ ,  $\vdash_{*\mathcal{CS}} *(c, A)$

and for any integer  $n \geq 1$

$$\vdash_{*\mathcal{CS}} *(\underbrace{! \dots !}_n c, \underbrace{! \dots !}_{n-1} c : \dots : ! c : c : A)$$

Step 3<sup>!</sup>. For  $c: X \subseteq \mathcal{CS}$  and any axiom  $A$  from scheme  $X$ , by  $*\mathcal{CS}$

$$\vdash_{*\mathcal{CS}} *(c, A)$$

Step 5. If  $*(s, G) \in S$  then for either calculus

$$S \vdash_* *(s, G)$$

Step 6. By IH,  $S \vdash_* *(s_1, G_1)$  for any  $G_1 \in X_1$  and  $S \vdash_* *(s_2, G_2)$  for any  $G_2 \in X_2$ . Therefore, by \*A3

$$S \vdash_* *(s_1 + s_2, G_1) \quad \text{and} \quad S \vdash_* *(s_1 + s_2, G_2)$$

for any  $G_1 \in X_1$  and any  $G_2 \in X_2$ . This takes care of both possible assignments in this step.

Step 7<sup>!</sup>. By IH,  $S \vdash_{*\mathcal{CS}} *(s, G)$  for any  $G \in X$ . Therefore, for any  $G \in X$  by \*A5

$$S \vdash_{*\mathcal{CS}} *(!s, s:G)$$

Step 8. – Let  $Z_1 = X_1 \rightarrow Y_1$  and  $\sigma = \text{mgu}(X_1, X_2)$ . For any  $G \in Y_1\sigma$  there must exist a substitution  $\tau$  such that  $G = Y_1\sigma\tau$ . Since  $\sigma$  is the mgu of  $X_1$  and  $X_2$ , we have  $X_1\sigma = X_2\sigma$ . Therefore,  $X_1\sigma\tau = X_2\sigma\tau$ . If this expression is still a scheme, i.e., it still has variables over formulas and/or over terms, instantiate these variables arbitrarily

by a substitution  $\tau'$ . Since  $G = Y_1\sigma\tau$  is a formula, not a scheme, substitution  $\tau'$  does not affect  $G$ :  $Y_1\sigma\tau\tau' = Y_1\sigma\tau = G$ .  $X_2\sigma\tau\tau'$  is an instance of scheme  $X_2$ ; therefore, by IH

$$S \vdash_* *(s_2, X_2\sigma\tau\tau') .$$

Similarly,

$$Z_1\sigma\tau\tau' = X_1\sigma\tau\tau' \rightarrow Y_1\sigma\tau\tau' = X_2\sigma\tau\tau' \rightarrow G .$$

By IH,

$$S \vdash_* *(s_1, X_2\sigma\tau\tau' \rightarrow G) .$$

Therefore, by \*A2

$$S \vdash_* *(s_1 \cdot s_2, G) .$$

- Let  $Z_1 = P$  and  $X_2 \neq \emptyset$ . Any formula  $G \in Q$  for a variable over formulas  $Q$ . Let  $E \in X_2$ . Then, by IH,  $S \vdash_* *(s_2, E)$ . Since  $P$  is a variable over formulas,  $E \rightarrow G \in P$ . By IH,  $S \vdash_* *(s_1, E \rightarrow G)$ .

By \*A2

$$S \vdash_* *(s_1 \cdot s_2, G) .$$

This completes the proof of the ‘only if’ direction.

Let us now prove the ‘if’ direction. Suppose  $S \vdash_* *(s, G)$ . Throughout the remainder of the proof, we will talk about suboccurrences rather than

subterms because a term may occur several times in  $t$ . For instance, one constant can be used for different axioms on different derivation branches.

There is a natural association of nodes in the  $\vdash_*$ -derivation with occurrences of subterms of  $t$  whereby

- each use of  $*\mathcal{CS}^!$  rule in a  $*\mathcal{CS}$ -derivation is associated with a particular occurrence of  $\underbrace{!\dots!}_n c$  for some constant  $c$  and some integer  $n \geq 0$  (no nodes are associated with proper suboccurrences of  $\underbrace{!\dots!}_n c$  for  $n > 0$ );
- each use of  $*\mathcal{CS}$  rule in a  $*\mathcal{CS}$ -derivation is associated with a particular occurrence of a constant  $c$ ;
- each use of a hypothesis  $*(s, G) \in S$  is associated with an occurrence of  $s$  in  $t$  (no nodes are associated with proper suboccurrences of  $s$ );
- the root of the derivation tree is associated with term  $t$  itself;
- assumption(s) of each rule  $*A2$ ,  $*A3$ , or  $*A5$  is(are) associated with the immediate subterm(s) of the conclusion of the same rule.

We will now show how to make non-deterministic choices based on this derivation so as to end up with **true** as the returned value.

- In Step 1, choose ‘ $S$ ’ for all occurrences of subterms  $s$  that are associated with the use of hypotheses in the derivation. Choose ‘ $\vdash$ ’ for all

other occurrences of such  $s$ . If  $s$  is a proper suboccurrence of  $s'$  and ‘ $S$ ’ was chosen for this occurrence of  $s$ , it cannot happen that ‘ $S$ ’ is also chosen for this occurrence of  $s'$ . Indeed, the subterms with chosen ‘ $S$ ’ are associated with the leaves of  $\vdash_*$ -derivation. A proper suboccurrence is associated with a node higher on the same branch of the derivation, and two leaves cannot be on the same branch. Hence, no ‘ $S$ ’ will be changed to ‘ $\#$ ’ in Step 1.

- In Step 2, let an occurrence of  $s_1 + s_2$  be associated with a non-leaf node in the derivation. It must be a conclusion of a \*A3 rule of one of two forms:

$$\frac{*(s_1, G)}{*(s_1 + s_2, G)} \quad \text{or} \quad \frac{*(s_2, G)}{*(s_1 + s_2, G)} .$$

Choose ‘l’ for this occurrence of  $+$  in the former case or ‘r’ in the latter. This rule dictates the choice only for those occurrences of  $+$  that are not within any term with chosen ‘ $S$ ’, which complies with Step 2.

- In Step 3, let an occurrence of  $\underbrace{!\dots!}_n c$ ,  $n \geq 0$ , be associated with a node in the  $\vdash_{*cS}$ -derivation. It can only be a leaf node. Let this node be either  $*(c, A)$  for  $n = 0$  or

$$*(\underbrace{!\dots!}_n c, \underbrace{!\dots!}_{n-1} c : \dots : !c : c : A)$$

for  $n \geq 1$ , an instance of  $*\mathcal{CS}^!$  rule rather than a hypothesis. Let axiom  $A$  belong to an axiom scheme  $X$ . Then assign  $c \rightsquigarrow X$  to this occurrence of  $c$  (for  $n = 0$ ) or assign

$$\underbrace{!!\dots!}_n c \rightsquigarrow \underbrace{!\dots!}_{n-1} c : \dots : !c : c : X$$

to this occurrence of  $\underbrace{!!\dots!}_n c$  (for  $n \geq 1$ ). This rule dictates the choice only for those occurrences of  $\underbrace{!\dots!}_n c$ ,  $n \geq 0$  that are not within either any term with chosen ‘ $S$ ’ or a term of the form  $\underbrace{!!\dots!}_{n+1} c$ , which complies with Step 3.

- In Step 3<sup>!</sup>, let an occurrence of a constant  $c$  be associated with a node in the  $\vdash_{*\mathcal{CS}}$ -derivation. It can only be a leaf node. Let this node be an instance  $*(c, A)$  of  $*\mathcal{CS}$  rule rather than a hypothesis. Let axiom  $A$  belong to an axiom scheme  $X$ . Then assign  $c \rightsquigarrow X$  to this occurrence of  $c$ . This rule dictates the choice only for those occurrences of  $c$  that are not within any term with chosen ‘ $S$ ’, which complies with Step 3<sup>!</sup>.
- In Step 5, let an occurrence of  $s$  with chosen ‘ $S$ ’ be associated with a leaf node of the derivation where a hypothesis  $*(s, G) \in S$  is used. Assign  $s \rightsquigarrow G$  to this occurrence of  $s$ .

We will now prove that after these choices are made, for each assignment  $s \rightsquigarrow X$  with  $X \neq \emptyset$  made by the procedure to an occurrence  $s$ , the

corresponding node in the derivation is  $*(s, G)$ , where  $G \in X$ . All assignments made so far in Steps 3, 3<sup>!</sup>, and 5 satisfy this property.

- For Step 6, assume w.l.o.g. that ‘l’ was chosen for the occurrence of  $+$  in this occurrence of  $s_1 + s_2$  (the case of chosen ‘r’ is completely analogous) and that a scheme  $X_1 \neq \emptyset$  was assigned to  $s_1$ . By IH, the node associated with  $s_1$  is  $*(s_1, G)$  for some  $G \in X_1$ . Since ‘l’ was chosen for this  $+$ , rule \*A3 was used after this node in the derivation in the form

$$\frac{*(s_1, G)}{*(s_1 + s_2, G)}$$

Thus, the successor node is  $*(s_1 + s_2, G)$  with  $G \in X_1$ . This complies with the assignment of  $X_1$  to  $s_1 + s_2$  made in this Step 6.

- For Step 7<sup>!</sup>, assume that a scheme  $X \neq \emptyset$  was assigned to  $s$ . By IH, the node associated with  $s$  is  $*(s, G)$  for some  $G \in X$ . Since  $s$  is the immediate suboccurrence of  $!s$  in  $t$ , rule \*A5 was used after this node in the derivation:

$$\frac{*(s, G)}{*(!s, s:G)}$$

Thus, the successor node is  $*(!s, s:G)$  with  $s:G \in s:X$ . This complies with the assignment of  $s:X$  to  $!s$  made in this Step 7<sup>!</sup>.

- For Step 8, two situations lead to a non-empty assignment

- Assume that a scheme  $Z_1 = X_1 \rightarrow Y_1$  was assigned to  $s_1$  and a scheme  $X_2 \neq \emptyset$  was assigned to  $s_2$ . Let  $\sigma = \text{mgu}(X_1, X_2)$ . By IH, the node associated with  $s_1$  is  $*(s_1, H \rightarrow E)$  for some  $H \in X_1$  and  $E \in Y_1$ , and the node associated with  $s_2$  is  $*(s_2, H')$  for some  $H' \in X_2$ . Since  $s_1$  and  $s_2$  are the immediate suboccurrences of  $s_1 \cdot s_2$  in  $t$ , rule \*A2 was used after this node in the derivation. This requires that  $H = H'$ :

$$\frac{*(s_1, H \rightarrow E) \quad *(s_2, H')}{*(s_1 \cdot s_2, E)}$$

Thus, the successor node is  $*(s_1 \cdot s_2, E)$ . This Step 8 assigns  $Y_1\sigma$  to  $s_1 \cdot s_2$ , so we need to show that  $E \in Y_1\sigma$ .

Since  $(H \rightarrow E) \in (X_1 \rightarrow Y_1)$  there must exist a substitution  $\tau'$  such that  $X_1\tau' = H$  and  $Y_1\tau' = E$ . Since  $X_1\tau' \ni H = H' \in X_2$  and  $\sigma = \text{mgu}(X_1, X_2)$ , there must exist a substitution  $\tau$  such that  $\tau' = \sigma\tau$ . Therefore,  $E = Y_1\tau' = Y_1\sigma\tau$  is an instance of  $Y_1\sigma$ .

- Assume that a variable over formulas  $P$  was assigned to  $s_1$  and a scheme  $X_2 \neq \emptyset$  was assigned to  $s_2$ . By IH, the node associated with  $s_1$  is  $*(s_1, G)$  for some  $G$ , and the node associated with  $s_2$  is  $*(s_2, H)$  for some  $H \in X_2$ . Since  $s_1$  and  $s_2$  are the immediate suboccurrences of  $s_1 \cdot s_2$  in  $t$ , rule \*A2 was used after this node in

the derivation. This requires that  $G = H \rightarrow E$  for some  $E$ :

$$\frac{*(s_1, H \rightarrow E) \quad *(s_2, H)}{*(s_1 \cdot s_2, E)}$$

Thus, the successor node is  $*(s_1 \cdot s_2, E)$  with  $E \in Q$ . This complies with the assignment of  $Q$  to  $s_1 \cdot s_2$  made in this Step 8.

This completes the proof of Correctness Lemma 5.1.2. □

It remains to show that both procedures run in non-deterministic polynomial time (polynomial in the total of sizes of all  $*$ -expressions from  $S$  plus the size of  $*(t, F)$ ).

**Lemma 5.1.3.**

1.  $*_{\mathcal{CS}}\text{-DERIVE}$  is an NP algorithm.
2.  $*!_{\mathcal{CS}}\text{-DERIVE}$  is an NP algorithm.

*Proof.* Steps 1–5 provide no more than  $|t|$  various choices: for each occurrence of  $\underbrace{!\dots!}_n c$  for a constant  $c$  and an integer  $n \geq 0$ , each occurrence of  $+$ , and each occurrence of a subterm  $s$  such that  $*(s, G) \in S$ . The choice for each  $+$  is binary. The choice for  $\underbrace{!\dots!}_n c$ ,  $n \geq 0$ , is finite since there are only finitely many axiom schemes to choose from. Not all schemes might be applicable to a particular constant depending on the  $\mathcal{CS}$ . The set of applicable schemes

for each constant is decidable since  $\mathcal{CS}$  is. The choice for subterms  $s$  with  $*(s, G) \in S$  is linear in the number of  $*$ -expressions in  $S$ .

Note that there is at most one scheme (possibly  $\emptyset$ ) assigned to every subterm of  $t$ . Therefore, the number of productive steps in the 5–8 loop is bounded by  $|t|$ .

It is clear that each step requires only polynomial time. The only step where it is not completely evident is Step 8 in the case when  $X_1 \rightarrow Y_1$  is assigned to  $s_1$ . In this case, the algorithm tries to unify  $X_1$  with  $X_2$  and produces their most general unifier if possible. This can be done in polynomial (quadratic) time in the total of sizes of dags representing schemes  $X_1$  and  $X_2$  using the modified Robinson's unification algorithm from [CB83]. Moreover, the constructed mgu can be simultaneously applied to  $Y_1$ . The use of this algorithm requires that all schemes be stored in dags rather than trees.  $\square$

This completes the proof of Theorem 5.1.1.  $\square$

**Corollary 5.1.4.** *Let  $\mathcal{CS}$  be a decidable almost schematic constant specification for one of pure or hybrid justification logics.*

1. *There exists an NP algorithm for determining for any given finite set  $X$  of  $*$ -expressions and a given  $*(t, F)$  whether*

$$S \vdash_{*\mathcal{CS}} *(t, F) .$$

2. There exists an NP algorithm for determining for any given finite set  $X$  of  $*$ -expressions and a given  $*(t, F)$  whether

$$S \vdash_{*!_{\mathcal{CS}}} *(t, F) .$$

*Proof.* Since  $\mathcal{CS}$  is almost schematic, it can be broken into two disjoint parts:  $\mathcal{CS} = \mathcal{CS}_1 \cup \mathcal{CS}_2$ , where  $\mathcal{CS}_1$  is schematic and decidable,  $\mathcal{CS}_2$  is finite (and hence decidable), and  $\mathcal{CS}_1 \cap \mathcal{CS}_2 = \emptyset$ .

$$\begin{aligned} S \vdash_{*\mathcal{CS}} *(t, F) &\iff S \cup \mathcal{CS}_2^* \vdash_{*\mathcal{CS}_1} *(t, F) \\ S \vdash_{*!_{\mathcal{CS}}} *(t, F) &\iff S \cup \mathcal{CS}_2^* \vdash_{*!_{\mathcal{CS}_1}} *(t, F) \end{aligned}$$

Derivability of the right sides can be determined non-deterministically in polynomial time by Theorem 5.1.1 since  $\mathcal{CS}_1$  is schematic and decidable while  $S \cup \mathcal{CS}_2^*$  is still finite.  $\square$

**Theorem 5.1.5.** *Let  $\mathcal{CS}$  be a decidable almost schematic constant specification for*

$$L \in \{J, JD, JT, J4, JD4, LP, T_n LP, S4_n LP, S5_n LP\} .$$

*Then,  $rL_{\mathcal{CS}}$  is in NP.*

*Proof.* According to Theorem 3.4.2 for pure justification logics and Theorem 3.5.23 for hybrid logics,

$$\bullet \text{ r}_{\mathcal{L}_{\mathcal{CS}}} \vdash t:F \iff *_{\mathcal{CS}\text{-calculus}} \vdash *(t, F)$$

for  $L \in \{J, JD, JT\}$ , and

$$\bullet \text{ r}_{\mathcal{L}_{\mathcal{CS}}} \vdash t:F \iff *!_{\mathcal{CS}\text{-calculus}} \vdash *(t, F)$$

for  $L \in \{J4, JD4, LP, T_nLP, S4_nLP, S5_nLP\}$ .

The right side of each equivalence can be decided using  $*_{\mathcal{CS}\text{-DERIVE}}$  and  $*!_{\mathcal{CS}\text{-DERIVE}}$  procedures respectively, which, by Cor. 5.1.4, are both NP-algorithms provided  $\mathcal{CS}$  is decidable and almost schematic.  $\square$

**Theorem 5.1.6.**  $\text{rJ}, \text{rJD}, \text{rJT}, \text{rJ4}, \text{rJD4}, \text{rLP}, \text{rT}_n\text{LP}, \text{rS4}_n\text{LP}, \text{rS5}_n\text{LP}$  are all in NP.

## 5.2 Upper Bounds for Pure Justification Logics

**Theorem 5.2.1** ([Art98, Mil07]).  $\text{LP}_{\mathcal{CS}}$  with a finite  $\mathcal{CS}$  or decidable injective  $\mathcal{CS}$  is in co-NP.

**Theorem 5.2.2** ([Kuz00]).  $\text{J}_{\mathcal{CS}}, \text{JT}_{\mathcal{CS}}, \text{J4}_{\mathcal{CS}}, \text{LP}_{\mathcal{CS}}$  with a decidable almost schematic  $\mathcal{CS}$  are in  $\Pi_2^P$ .

*Proof.* The decision procedure for any of these logics consists of two parts. First a propositional tableau procedure is performed with two additional

rules. The rules to be added for  $J_{CS}$  and  $J4_{CS}$  are

$$\frac{T \ s:G}{T \ *(s, G)} \qquad \frac{F \ s:G}{F \ *(s, G)} \qquad (5.2.1)$$

The rules to be added for  $JT_{CS}$  and  $LP_{CS}$  are

$$\frac{T \ s:G}{T \ G} \qquad \frac{F \ s:G}{F \ *(s, G) \mid F \ G} \qquad (5.2.2)$$

$$T \ *(s, G)$$

The  $*$ -expressions are not analyzed further in the first tableau part of the algorithm.

As in the propositional case, whenever  $TG$  and  $FG$  appear on the same branch, such a branch is *propositionally closed*.

As in the propositional case, all rules decrease the complexity of formulas, therefore, each branch can be either completed or propositionally closed.

The second stage of the algorithm starts when all branches are either completed or propositionally closed. For each completed branch that is not propositionally closed, we attempt to close it using  $*$ -expressions. Namely, let  $X$  be the set of all  $*$ -expressions with prefix  $T$  on this branch. For every  $*$ -expression  $F \ *(s, G)$  on this branch, we run

- $*_{CS}$ -DERIVE $\langle X, *(s, G) \rangle$  for  $J_{CS}$  and  $JT_{CS}$  or
- $*!_{CS}$ -DERIVE $\langle X, *(s, G) \rangle$  for  $J4_{CS}$  and  $LP_{CS}$ .

If any such run returns **true** we close this branch. We will call such branches *\*-closed*. Otherwise, this branch is announced open.

**Lemma 5.2.3 (Correctness of the algorithm).** *For each of justification logics  $J_{CS}$ ,  $JT_{CS}$ ,  $J4_{CS}$ ,  $LP_{CS}$ , a formula  $G$  is not derivable in it iff there is a completed tableau constructed by the rules for that logic for  $F G$  with at least one branch open, i.e., neither propositionally closed nor *\*-closed*.*

*Proof.* Firstly, suppose  $G$  is not derivable. Then, by the Completeness Theorem 3.3.4, there exists an M-model  $\mathfrak{M} = (V, \mathcal{A})$  such that  $\mathfrak{M} \Vdash \neg G$ . We will show that there will always be an open branch, i.e., a branch that is neither propositionally nor *\*-closed*. Namely, we will show that throughout the tableau procedure there is at least one branch with all prefixed statements satisfied, i.e., with all  $T$ -prefixed formulas true, all  $F$ -prefixed formulas false, all  $T$ -prefixed *\*-expressions* true for  $\mathcal{A}^*$ , and with and  $F$ -prefixed *\*-expressions* false for  $\mathcal{A}^*$ .

$\mathfrak{M} \Vdash \neg G$  so  $F G$  is satisfied in the model.

The propositional cases are treated in the standard way. It remains to note that the new rules are synchronized with the definition of  $\Vdash$  for justification formulas in M-models. More precisely,

- For logics  $J_{CS}$  and  $J4_{CS}$ . Let  $T t : H$  be on a branch with all prefixed

statements satisfied. By IH,  $\mathfrak{M} \Vdash t : H$ . By (3.3.7),  $\mathcal{A}(t, H)$  holds; hence,  $T * (t, H)$  is satisfied.

Let  $F t : H$  be on a branch with all prefixed statements satisfied. By IH,  $\mathfrak{M} \not\Vdash t : H$ . By (3.3.7),  $\mathcal{A}(t, H)$  does not hold; hence,  $F * (t, H)$  is satisfied.

- For logics  $\text{JT}_{CS}$  and  $\text{LP}_{CS}$ . Let  $T t : H$  be on a branch with all prefixed statements satisfied. By IH,  $\mathfrak{M} \Vdash t : H$ . By (3.3.6),  $\mathcal{A}(t, H)$  holds and  $\mathfrak{M} \Vdash H$ ; hence, both  $T * (t, H)$  and  $T H$  are satisfied.

Let  $F t : H$  be on a branch with all prefixed statements satisfied. By IH,  $\mathfrak{M} \not\Vdash t : H$ . By (3.3.6), either  $\mathcal{A}(t, H)$  does not hold or  $\mathfrak{M} \not\Vdash H$ . In the former case  $F * (t, H)$  is satisfied; in the latter case  $F H$  is satisfied. Thus, at least one of the two resulting branches will still have all prefixed statements satisfied.

Thus, by the time the tableau is completed, there must remain a branch with all prefixed statements satisfied. Since it is not possible to satisfy  $T H$  and  $F H$  at the same time, this branch is not propositionally closed.

It remains to show that this branch is not  $*$ -closed either. A proof by contradiction. Suppose towards a contradiction that this branch is  $*$ -closed. It means that one of the runs of  $*(!)_{CS}\text{-DERIVE}\langle X, *(s, G) \rangle$  returned **true**,

where  $X$  is the set of all  $T$ -prefixed  $*$ -expressions on this branch and statement  $F * (s, G)$  is also on this branch. By Lemma 5.1.2,  $X \vdash_* *(s, G)$  for the  $*$ -calculus of this logic. Let  $\mathcal{B}_X$  be an M-type possible evidence function defined by

$$\mathcal{B}_X(t, H) = \text{True} \quad \iff \quad *(t, H) \in X \quad (5.2.3)$$

By Corollary 3.3.42.2,  $\mathcal{E}(s, G)$  holds for the minimal admissible evidence function  $\mathcal{E}$  based on  $\mathcal{B}_X$ . Since all the  $T$ -prefixed  $*$ -expressions on the branch are satisfied,  $\mathcal{A}$  is also an admissible evidence function based on  $\mathcal{B}_X$ . Therefore,  $\mathcal{E} \subseteq \mathcal{A}$  and  $\mathcal{A}(s, G)$  holds. On the other side,  $\mathcal{A}(s, G)$  cannot hold because  $F * (s, G)$  has to be satisfied. This contradiction shows that no  $*(!)_{CS}$ -DERIVE run can return **true**, and this branch is not  $*$ -closed.

This completes the proof of the ‘only if’ direction.

Let us now prove the ‘if’ direction. Suppose there is a completed tableau with an open branch. We will construct an M-model based on this open branch. Let

$$V(p) = \text{True} \quad \text{iff} \quad Tp \text{ is on the open branch} \quad (5.2.4)$$

Let  $\mathcal{A}$  be the minimal evidence function based on  $\mathcal{B}_X$  from (5.2.3) for this branch. We claim that for  $\mathfrak{M} = (V, \mathcal{A})$  all prefixed expressions from the open branch are satisfied.

First of all,  $\mathcal{A}$  is based on  $\mathcal{B}_X$ , i.e.,  $\mathcal{A}(t, H)$  holds for each  $T * (t, H)$  on the branch.

Since  $\mathcal{A}$  is the minimal function it is defined either by (3.3.35) for  $J_{CS}$  and  $JT_{CS}$  or by (3.3.36) for  $J4_{CS}$  and  $LP_{CS}$ . For any  $F * (s, G)$  on the branch the  $*(!)_{CS}$ -DERIVE $\langle X, *(s, G) \rangle$  returned **false** because the branch is open. Therefore,  $X \not\vdash_* *(s, G)$  and  $\mathcal{A}(s, G)$  does not hold.

Now let us prove that all prefixed formulas on the branch are satisfied in  $\mathfrak{M}$  by induction on the size of a formula. If  $T p$  is on the branch, then  $V(p) = True$ , so  $\mathfrak{M} \Vdash p$ . If  $F p$  is on the branch,  $T p$  is not on the branch because the branch is open.  $V(p) = False$ , so  $\mathfrak{M} \not\vdash p$ .

The Boolean cases are standard.

If  $T t:H$  is on the branch, then

- For  $J_{CS}$  and  $J4_{CS}$ ,  $T * (t, H)$  must be on the branch because the branch is completed. Therefore,  $\mathcal{A}(t, H)$  holds.
- For  $JT_{CS}$  and  $LP_{CS}$ , in addition,  $T H$  must be on the branch, and by IH,  $\mathfrak{M} \Vdash H$ .

In either case  $\mathfrak{M} \Vdash t:H$ .

If  $F t:H$  is on the branch, then

- For  $J_{\mathcal{CS}}$  and  $J4_{\mathcal{CS}}$ ,  $F * (t, H)$  must be on the branch because the branch is completed. Therefore,  $\mathcal{A}(t, H)$  does not hold.
- For  $JT_{\mathcal{CS}}$  and  $LP_{\mathcal{CS}}$ , in addition, another possibility is that  $F H$  could be on the branch if the right branch of the  $\beta$ -rule is open, in which case, by IH,  $\mathfrak{M} \not\vdash H$ .

In either case  $\mathfrak{M} \not\vdash t: H$ .

In particular,  $F G$ , the root of the branch must be satisfied, therefore,  $\mathfrak{M} \not\vdash G$ ; hence,  $G$  is not derivable.

This completes the proof of Lemma 5.2.3. □

It remains to show that the complexity of this algorithm is  $\Pi_2^p$ . The complexity of the propositional tableau procedure is NP. The new rules for formulas of type  $t: H$  clearly do not change that. This means that to show that a formula is not derivable we need to guess which branch of the tableau is open. The branch itself is of polynomial length, in fact linear in  $|G|$ .

Complexity of  $*(!)_{\mathcal{CS}}$ -DERIVE for a decidable almost schematic  $\mathcal{CS}$  is NP. In other words, to get the answer **true** it is sufficient to guess a  $*$ -calculus derivation of polynomial length. To show that a branch of the tableau is not  $*$ -closed we, on the contrary need to check all  $F$ -prefixed  $*$ -expressions and obtain the answer **false** for all of them. This requires checking all possible

\*-calculus derivation and is hence a dual problem, co-NP.

The size of  $X, *(s, H)$  for each call of  $*(!)_{CS}$ -DERIVE is clearly polynomial in  $|G|$ . Thus, the overall complexity of determining that  $G$  is not derivable is  $\Sigma_2^p$  and the dual Validity Problem is in  $\Pi_2^p$ .

This completes the proof of Theorem 5.2.2.  $\square$

The same result was announced in [Kuz00] for  $JD_{CS}$  and  $JD4_{CS}$ . Recently an omission was found in that proof. These logics feature an additional Consistent Evidence Condition on the admissible evidence function. This condition requires  $\mathcal{A}(t, \perp)$  to be *False* for all terms  $t$ , not only for the subterms of a given  $G$ . This may make the set of prefixed \*-statements on the branch inconsistent even though all  $*(!)_{CS}$ -DERIVE runs returned **false**. We will now correct the proof of the upper bound for  $JD_{CS}$ .

**Theorem 5.2.4.**  *$JD_{CS}$  with a decidable, almost schematic, and axiomatically appropriate  $CS$  is in  $\Pi_2^p$ .*

*Proof.* Following the ideas of Fitting and Massacci from [Fit72, Mas94, FM98, Mas00], we will use integer prefixes for our already prefixed formulas  $TG$  and  $FG$ . To disambiguate the two types of prefixes we will address them as a *truth prefix* and *integer prefix* respectively.

For most modal logics, sequences of integers are used as prefixes rather

than single integers. In this respect, our integer prefixes resemble prefixes for **S5** where single integers suffice. There is a crucial difference though. Prefixes for **S5** represent different worlds in the same Kripke model. The accessibility relation is assumed to be total, i.e. any world/prefix is accessible from any other world/prefix.

In our case, integer prefixes will represent different M-models with the underlying intuition that the existence of the  $(n + 1)$ st M-model justifies the Consistent Evidence condition for the  $n$ th M-model.

This is how we amend the tableau rules for  $\text{JD}_{\mathcal{CS}}$ . All the propositional rules act the same way with respect to formulas and truth prefixes; they do not change the integer prefix. The rule for  $n F s : G$  remains the same as for  $\text{J}_{\mathcal{CS}}$  and  $\text{J4}_{\mathcal{CS}}$  with the addition of integer prefix that once again is unchanged. The only significant change is in the rule for  $n T s : G$ :

$$\frac{n T s : G}{\begin{array}{l} n T *(s, G) \\ n+1 T G \end{array}} \quad (5.2.5)$$

Whenever  $n T G$  and  $n F G$  appear on the same branch, such a branch is *propositionally closed*.

As in the propositional case, all rules decrease the complexity of formulas regardless of whether the integer prefix is incremented. Therefore, each branch can be either completed or propositionally closed.

The second stage of the algorithm starts when all branches are either completed or propositionally closed. For each completed branch that is not propositionally closed, we attempt to close it using  $*$ -expressions. Namely, let  $X_n$  be the set of all  $*$ -expressions with prefix  $n T$  on this branch. For every integer prefix  $n$  occurring on this branch and every  $*$ -expression  $n F *(s, G)$  from the branch, we run  $*_{CS}$ -DERIVE $\langle X_n, *(s, G) \rangle$ . If any such run returns **true**, we close this branch. We will call such branches  $*$ -closed. Otherwise, the branch is announced open.

**Lemma 5.2.5 (Correctness of the algorithm).**  $JD_{CS} \not\vdash G$  iff there is a completed  $JD_{CS}$ -tableau for  $1 F G$  with at least one branch open, i.e., neither propositionally closed nor  $*$ -closed.

*Proof.* First suppose  $G$  is not derivable. Then, by the Completeness Theorem 3.3.4, there exists an M-model  $\mathfrak{M} = (V, \mathcal{A})$  such that  $\mathfrak{M} \not\vdash G$ . We will define an infinite sequence of M-models

$$\mathfrak{M}_1 = (V_1, \mathcal{A}_1), \dots, \mathfrak{M}_n = (V_n, \mathcal{A}_n), \dots$$

by induction on  $n$ . The first model in the sequence will be

$$\mathfrak{M}_1 = \mathfrak{M} .$$

Let M-model  $\mathfrak{M}_n = (V_n, \mathcal{A}_n)$  be already constructed. The admissible

evidence function  $\mathcal{A}_n$  is clearly based on itself. Not surprisingly, it is also the minimal such M-type admissible evidence function. Therefore, by Theorem 3.3.41.1,

$$*(s, H) \in \mathcal{A}_n^* \iff \mathcal{A}_n^* \vdash_{*cS} *(s, H)$$

By Consistent Evidence condition for  $\mathcal{A}_n$ ,

$$*(s, \perp) \notin \mathcal{A}_n^*$$

for any term  $s$ . Therefore,

$$\mathcal{A}_n^* \not\vdash_{*cS} *(s, \perp)$$

for any  $s$ . We will prove by contradiction that

$$(\mathcal{A}_n^*)^\sharp \not\vdash_{\text{JD}_{cS}} \perp \quad (5.2.6)$$

Suppose towards a contradiction that  $\perp$  is derivable from  $(\mathcal{A}_n^*)^\sharp$ . Only finitely many formulas can be used in this derivation, so

$$H_1, \dots, H_k \vdash_{\text{JD}_{cS}} \perp ,$$

where  $\mathcal{A}_n(s_i, H_i) = \text{True}$ ,  $i = 1, \dots, k$ , for some terms  $s_1, \dots, s_k$ . Internalizing this derivation by Lemma 3.2.22, we would get

$$x_1 : H_1, \dots, x_k : H_k \vdash_{\text{JD}_{cS}} t(x_1, \dots, x_k) : \perp$$

for fresh justification variables  $x_1, \dots, x_k$  and some term  $t$  (we use the fact that  $\mathcal{CS}$  is axiomatically appropriate). The simultaneous substitution of  $s_i$ 's for  $x_i$ 's would yield by Lemma 3.2.30 (again axiomatic appropriateness of  $\mathcal{CS}$  is used)

$$s_1 : H_1, \dots, s_k : H_k \vdash_{\text{JD}_{\mathcal{CS}}} \bar{t}(s_1, \dots, s_k) : \perp .$$

Since  $\text{JD}_{\mathcal{CS}} \vdash \bar{t}(s_1, \dots, s_k) : \perp \rightarrow \perp$ , in this case the set

$$\{s_1 : H_1, \dots, s_k : H_k\} \tag{5.2.7}$$

would be  $\text{JD}_{\mathcal{CS}}$ -inconsistent. But  $\mathcal{A}_n(s_i, H_i) = \text{True}$  and hence, by (3.3.7),  $\mathfrak{M}_n \Vdash s_i : H_i$  for all  $i = 1, \dots, k$  clearly making  $\mathfrak{M}_n$  an M-model for set (5.2.7). This contradiction completes the proof of (5.2.6).

Thus,  $(\mathcal{A}_n^*)^\sharp$  is  $\text{JD}_{\mathcal{CS}}$ -consistent. By Lemma 2.6.2.8, it can be extended to a maximal consistent set  $\Gamma \supseteq (\mathcal{A}_n^*)^\sharp$ . By Lemma 3.3.5, there is a canonical M-model  $\mathfrak{M}_{n+1}$  for  $\Gamma$  such that

$$\mathfrak{M}_{n+1} \Vdash F \iff F \in \Gamma .$$

This will be our next M-model. In particular,

$$\mathcal{A}_n(s, F) = \text{True} \implies \mathfrak{M}_{n+1} \Vdash F . \tag{5.2.8}$$

We will show that there will always be an open branch, i.e., a branch that is neither propositionally nor  $*$ -closed. Namely, we will show that through-

out the tableau procedure there is at least one branch where the following conditions are satisfied:

1. for all  $n \ T \ H$  on the branch  $\mathfrak{M}_n \Vdash H$
2. for all  $n \ F \ H$  on the branch  $\mathfrak{M}_n \not\Vdash H$
3. for all  $n \ T \ *(s, H)$  on the branch  $\mathcal{A}_n(s, H) = \text{True}$
4. for all  $n \ F \ *(s, H)$  on the branch  $\mathcal{A}_n(s, H) = \text{False}$

As usual, the proof is by induction on the tableau derivation. We start with  $1 \ F \ G$  at the root node of the future tableau, for which we know that  $\mathfrak{M}_1 \not\Vdash G$ .

The propositional cases do not change the integer prefix and, therefore, can be dealt with in the standard manner within each model.

Let us then look closely at the new rules for justification formulas. Let  $n \ F \ t:H$  be on a branch with all conditions 1–4 satisfied. By IH,  $\mathfrak{M}_n \not\Vdash t:H$ . By (3.3.7),  $\mathcal{A}_n(t, H) = \text{False}$ ; hence, condition 4 is satisfied for  $n \ F \ *(t, H)$ .

Let  $n \ T \ t:H$  be on a branch with all conditions 1–4 satisfied. By IH,  $\mathfrak{M}_n \Vdash t:H$ . By (3.3.7),  $\mathcal{A}_n(t, H) = \text{True}$ ; hence, condition 3 is satisfied for  $n \ T \ *(t, H)$ . In addition, by (5.2.8),  $\mathfrak{M}_{n+1} \Vdash H$ , which satisfies condition 1 for  $n+1 \ T \ H$ .

Thus, by the time the tableau is completed, there is still a branch with all conditions 1–4 satisfied. Since it is not possible to satisfy condition 1 for  $n T H$  and condition 2 for  $n F H$  at the same time, this branch is not propositionally closed.

It remains to show that this branch is not  $*$ -closed either. A proof by contradiction. Suppose towards a contradiction that this branch is  $*$ -closed. It means that one of the runs of  $*_{\mathcal{CS}}\text{-DERIVE}\langle X_n, *(s, G) \rangle$  returned **true**, where  $X_n$  is the set of all  $n T$ -prefixed  $*$ -expressions on this branch and, in addition,  $n F *(s, G)$  is also on this branch. By Lemma 5.1.2.1,

$$X_n \vdash_{*\mathcal{CS}} *(s, G) . \quad (5.2.9)$$

Let  $\mathcal{B}_{X_n}$  be an M-type possible evidence function defined by

$$\mathcal{B}_{X_n}(t, H) = \text{True} \iff *(t, H) \in X_n \quad (5.2.10)$$

By condition 3,  $\mathcal{A}_n$  is based on  $\mathcal{B}_{X_n}$ . So such admissible evidence functions do exist. By Theorem 3.3.41.1, there exists the minimal admissible evidence function  $\mathcal{E}_n$  based on  $\mathcal{B}_{X_n}$  that satisfies

$$*(s, H) \in \mathcal{E}_n^* \iff X_n \vdash_{*\mathcal{CS}} *(s, H) \quad (5.2.11)$$

by (3.3.35). Being the minimal function,  $\mathcal{E}_n \subseteq \mathcal{A}_n$ . Then, by (5.2.9) and (5.2.11),

$$\mathcal{A}_n(s, G) = \text{True} .$$

On the other hand,  $\mathcal{A}_n(s, G) = \text{False}$  according to condition 4 for  $n F *(s, G)$  on the branch. This contradiction shows that no  $*_{CS}$ -DERIVE run can return **true**, and this branch is not  $*$ -closed.

This completes the proof of the ‘only if’ direction.

Let us now prove the ‘if’ direction. Suppose there is a completed tableau with an open branch. We will construct a sequence of M-models based on this open branch. Let  $N$  be the largest integer prefix occurring on the open branch. Let

$$V_n(p) = \text{True} \quad \text{iff} \quad n T p \text{ is on the open branch} \quad (5.2.12)$$

Let  $\mathcal{A}_n$  be the defined by (3.3.35) based on  $\mathcal{B}_{X_n}$  from (5.2.10) for this branch. We claim that  $\mathcal{A}_n$ ’s are admissible evidence functions and that for the sequence  $\mathfrak{M}_n = (V_n, \mathcal{A}_n)$  conditions 1–4 are satisfied. The proof will be by induction on  $N - n$ .

*Base.*  $N - n = 0$ , i.e.,  $N = n$ . The absence of  $(N+1) T H$  on the branch implies the absence of  $N T t : H$  and hence of  $N T *(t, H)$  too. In other words,  $X_N = \emptyset$  and  $\mathcal{B}_{X_N}(t, H)$  is always *False*. By consistency of  $\text{JD}_{CS}$  (Theorem 3.2.21), there exist M-models with some admissible for  $\text{JD}_{CS}$  evidence functions, which are, of course, based on the empty  $\mathcal{B}_{X_N}$ . Thus, by Theorem 3.3.41.1,  $\mathcal{A}_N$  is the minimal M-type

admissible for  $\text{JD}_{CS}$  evidence function based on  $\mathcal{B}_{X_N}$ .

Now that we know  $\mathfrak{M}_N$  is an M-model, we are ready to prove conditions 1–4 for  $n = N$ .

Condition 3 is vacuously satisfied since there are no  $N T * (t, H)$  on the branch.

For any  $N F * (s, G)$  on the branch, the  $*_{CS}\text{-DERIVE}\langle X_N, *(s, G) \rangle$  returned **false** because the branch is open. Therefore,  $X_N \not\vdash_{*_{CS}} *(s, G)$  and  $\mathcal{A}_N(s, G) = \text{False}$ .

Now let us prove that conditions 1–2 are satisfied for  $N$  by induction on the size of formula. If  $N T p$  is on the branch,  $V_N(p) = \text{True}$ , so  $\mathfrak{M}_N \Vdash p$ . If  $N F p$  is on the branch,  $N T p$  is not on the branch because the branch is open.  $V_N(p) = \text{False}$ , so  $\mathfrak{M}_N \not\Vdash p$ .

The Boolean cases are standard.

$N T t:H$  does not occur on the branch.

If  $N F t:H$  is on the branch, then  $N F *(t, H)$  must be on the branch because the branch is completed. Therefore, by just proven condition 4,  $\mathcal{A}_N(t, H) = \text{False}$  and  $\mathfrak{M}_N \not\Vdash t:H$ .

*Step.* Let  $\mathcal{A}_{k+1}$  be an admissible for  $\text{JD}_{CS}$  evidence function and let condi-

tions 1–4 be satisfied for  $n = k+1$ .

Let us prove that  $\mathcal{A}_k$  is also an admissible for  $\text{JD}_{CS}$  evidence function. In the proof of Theorem 3.3.41 all the closure conditions for  $\text{JD}_{CS}$  were verified based solely on (3.3.35), except for the Consistent Evidence condition, for which an extra assumption of non-emptiness of  $\mathcal{AEF}_{\mathcal{B}}(\text{JD}_{CS})$  was used. As was noted in Footnote 6 on p. 110, this extra assumption is not used anywhere else in the proof. It follows that to show that  $\mathcal{A}_k$  is an admissible evidence function, it suffices to verify the Consistent Evidence condition for it.

Proof by contradiction. Suppose  $\mathcal{A}_k(s, \perp) = \text{True}$ . Then, by (3.3.35)

$$X_k \vdash_{*CS} *(s, \perp) .$$

By Lemma 3.4.10.1,

$$(X_k)^{\sharp} \vdash_{\text{JD}_{CS}} \perp ,$$

i.e.,  $(X_k)^{\sharp}$  is  $\text{JD}_{CS}$ -inconsistent. For any  $H \in (X_k)^{\sharp}$  there must be some  $k \ T \ t:H$  on the branch. Since the branch is completed, there also must be  $k+1 \ T \ H$  on the same branch. By IH,  $\mathfrak{M}_{k+1} \Vdash H$ . Therefore, the inconsistent set  $(X_k)^{\sharp}$  is satisfiable in the model

$$\mathfrak{M}_{k+1} \Vdash (X_k)^{\sharp} .$$

This contradiction shows that the Consistent Evidence condition for  $\mathcal{A}_k$  is satisfied.

Now that we know  $\mathfrak{M}_k$  is an M-model for  $\text{JD}_{\text{CS}}$ , we are ready to prove conditions 1–4 for  $n = k$ .

For any  $k \ T \ *(s, G)$  on the branch,  $*(s, G) \in X_k$ . Then, obviously,  $X_k \vdash_{*\text{CS}} *(s, G)$ ; therefore,  $\mathcal{A}_k(s, G) = \text{True}$ .

For any  $k \ F \ *(s, G)$  on the branch, the  $*\text{CS}$ -DERIVE $\langle X_k, *(s, G) \rangle$  returned **false** because the branch is open. Therefore,  $X_k \not\vdash_{*\text{CS}} *(s, G)$  and  $\mathcal{A}_k(s, G) = \text{False}$ .

Now let us prove that conditions 1–2 are satisfied for  $k$  by induction on the size of formula. If  $k \ T \ p$  is on the branch,  $V_k(p) = \text{True}$ , so  $\mathfrak{M}_k \Vdash p$ . If  $k \ F \ p$  is on the branch,  $k \ T \ p$  is not on the branch because the branch is open.  $V_k(p) = \text{False}$ , so  $\mathfrak{M}_k \not\Vdash p$ .

The Boolean cases are standard.

If  $k \ T \ t:H$ , then  $k \ T \ *(t, H)$  must be on the branch because the branch is completed. Therefore, by just proven condition 3,  $\mathcal{A}_k(t, H) = \text{True}$  and  $\mathfrak{M}_k \Vdash t:H$ .

If  $k \ F \ t:H$  is on the branch, then  $k \ F \ *(t, H)$  must be on the branch because the branch is completed. Therefore, by just proven condition 4,

$\mathcal{A}_k(t, H) = \text{False}$  and  $\mathfrak{M}_k \not\models t:H$ .

This completes the proof by induction on  $N - n$ .

In particular, by condition 2 for statement 1  $F G$  at the root of the tableau,  $\mathfrak{M}_1 \not\models G$ ; hence,  $G$  is not derivable.

This completes the proof of Lemma 5.2.5. □

It remains to show that the complexity of this algorithm is  $\Pi_2^p$ . The complexity of the propositional tableau procedure is NP. The new rules for  $t:H$  clearly do not change that. Note that  $*$ -expressions are not analyzed further. Note also that even when switching to the next model (incrementing the integer index), the complexity of the formulas strictly decreases on every step. Thus, the length of each tableau branch is still polynomial (in fact linear) in  $|G|$  for a given formula  $G$  that we try to refute. Since only some tableau steps warrant switches to a new model, the integer prefixes are also bounded by some  $N = O(|G|)$ .

Thus, the tableau portion including checking for propositional closures is NP as usual: to show that a formula is not derivable we need to guess which branch of polynomial length is open.

Complexity of each  $*_{\mathcal{CS}}$ -DERIVE call for a decidable almost schematic  $\mathcal{CS}$  is NP. In other words, to get the answer **true**, it is sufficient to guess a

$*_{\mathcal{CS}}$ -calculus derivation of polynomial length. To show that a branch of the tableau is not  $*$ -closed, on the contrary, we need to check for each  $k$  all  $k$   $F$ -prefixed  $*$ -expressions and obtain the answer **false** for them. This requires checking all possible  $*_{\mathcal{CS}}$ -calculus derivation and is hence a dual problem, a co-NP one.

The size of  $X_n, *(s, H)$  for each call of  $*_{\mathcal{CS}}$ -DERIVE is clearly polynomial in  $|G|$ . Thus, the overall complexity of determining that  $G$  is not derivable is  $\Sigma_2^p$  and the dual Validity Problem is in  $\Pi_2^p$ .

This completes the proof of Theorem 5.2.4.  $\square$

*Note 5.2.6.* The prefixed tableaux method used for  $\text{JD}_{\mathcal{CS}}$  does not work for  $\text{JD4}_{\mathcal{CS}}$ . The problem is that  $\text{JD4}_{\mathcal{CS}}$ -consistency of  $X^\sharp$  does not guarantee the existence of an admissible for  $\text{JD4}_{\mathcal{CS}}$  evidence function based on  $\mathcal{B}_X$ . For instance, the set

$$Y = \{p, \neg x:p\}$$

is perfectly  $\text{JD4}_{\mathcal{CS}}$ -consistent (it is sufficient to take  $\mathcal{A}$  to be the minimal admissible for  $\text{JD4}_{\mathcal{CS}}$  function and make sure that  $V(p) = \text{True}$ ).

On the other hand, for the set of  $*$ -expressions

$$X = \{*(x, p), *(y, \neg x:p)\}$$

with  $X^\sharp = Y$  there is no admissible evidence function  $\mathcal{A}$  such that

$$*(s, H) \in X \quad \Longrightarrow \quad \mathcal{A}(s, H) = \text{True}$$

The reason is very simple: there exists a term  $t$  such that

$$X \vdash_{*!_{\mathcal{CS}}} *(t, \perp)$$

for any axiomatically appropriate  $\mathcal{CS}$ . Indeed,

$$\text{JD4}_{\mathcal{CS}} \vdash \neg x:p \rightarrow (x:p \rightarrow \perp) .$$

Depending on the axiomatization, this may or may not be an axiom, but by the Constructive Necessitation (Cor. 3.2.24) there must exist a ground term  $s$  such that

$$\text{JD4}_{\mathcal{CS}} \vdash s:[\neg x:p \rightarrow (x:p \rightarrow \perp)] .$$

It follows that

$$\vdash_{*!_{\mathcal{CS}}} *(s, \neg x:p \rightarrow (x:p \rightarrow \perp)) .$$

Here is a derivation showing that any attempt to construct an admissible evidence function based on  $\mathcal{B}_X$  would violate the Consistent Evidence condition:

$$\frac{\begin{array}{c} \vdots \\ *(s, \neg x:p \rightarrow (x:p \rightarrow \perp)) \end{array} \quad \frac{*(y, \neg x:p) \quad *(x, p)}{*(s \cdot y, x:p \rightarrow \perp)} \quad \frac{*(x, p)}{*(!x, x:p)}}{*((s \cdot y) \cdot !x, \perp)}$$

Thus, the trick of reducing verification of the Consistent Evidence condition to checking satisfiability of simpler formulas does not work for  $\text{JD4}_{CS}$ .

### 5.3 Lower Bounds for Pure Justification Logics

There is a trivial lower bound for all justification logics:

**Theorem 5.3.1.** *Let  $\text{JL}$  be any of pure justification logics  $\text{J}$ ,  $\text{JD}$ ,  $\text{JT}$ ,  $\text{J4}$ ,  $\text{JD4}$ ,  $\text{LP}$ ,  $\text{J5}$ ,  $\text{J45}$ ,  $\text{JD45}$ ,  $\text{JT45}$ , and  $\text{CS}$  be any constant specification for  $\text{JL}$ . Then,  $\text{JL}_{CS}$  is co-NP-hard.*

*Proof.* The result follows from

**Lemma 5.3.2.** *Let  $\text{JL}$  be any of pure justification logics  $\text{J}$ ,  $\text{JD}$ ,  $\text{JT}$ ,  $\text{J4}$ ,  $\text{JD4}$ ,  $\text{LP}$ ,  $\text{J5}$ ,  $\text{J45}$ ,  $\text{JD45}$ ,  $\text{JT45}$ , and  $\text{CS}$  be any constant specification for  $\text{JL}$ . Then,  $\text{JL}_{CS}$  is conservative over classical propositional logic.*

*Proof.* Indeed, consider the following propositional translation  $[]^r$  from jus-

Table 5.3.1: Propositional translation of axioms of justification logics are propositional tautologies

	JL axiom	Its translation
A1	instance of A1	another instance of A1
A2	$s:(F \rightarrow G) \rightarrow (t:F \rightarrow s \cdot t:G)$	$(F^r \rightarrow G^r) \rightarrow (F^r \rightarrow G^r)$
A3	$s:F \rightarrow s + t:F$	$F^r \rightarrow F^r$
A3	$t:F \rightarrow s + t:F$	$F^r \rightarrow F^r$
A4	$t:F \rightarrow F$	$F^r \rightarrow F^r$
A5	$t:F \rightarrow !t:t:F$	$F^r \rightarrow F^r$
A6	$\neg t:F \rightarrow ?t:\neg t:F$	$\neg F^r \rightarrow \neg F^r$
A7	$t:\perp \rightarrow \perp$	$\perp \rightarrow \perp$

tification language to propositional language:

$$\perp^r \Leftrightarrow \perp$$

$$p^r \Leftrightarrow p$$

$$(F \rightarrow G)^r \Leftrightarrow F^r \rightarrow G^r$$

$$(t:F)^r \Leftrightarrow F^r$$

By induction on the  $\text{JL}_{\mathcal{CS}}$ -derivation we show that propositional translation of any  $\text{JL}_{\mathcal{CS}}$ -derivable formula is a propositional tautology.

As Table 5.3.1 shows, propositional translation of all justification axioms are either propositional axioms or simple propositional tautologies.

Translation of a *modus ponens* instance is another instance of *modus*

*ponens*:

$$\frac{F \rightarrow G \quad F}{G} \rightsquigarrow \frac{F^r \rightarrow G^r \quad F^r}{G^r}$$

Translation of the conclusion of a  $R4_{\mathcal{CS}}$  or a  $R4_{\mathcal{CS}}^!$  instance yields a translation of some axiom, which is shown to be a tautology in Table 5.3.1:

$$\begin{aligned} (c:A)^r &= A^r \\ (!\dots!c:\dots!c:c:A)^r &= A^r \end{aligned}$$

Suppose  $\mathcal{L}_{\mathcal{CS}} \vdash F$  for some propositional formula  $F$ . Then,  $F^r$  is a propositional tautology, but  $F = F^r$ . This completes the proof of Lemma 5.3.2.  $\square$

We are now ready to finish the proof of Theorem 5.3.1. As usual in the presence of conservativity, the identity function from the propositional language into the justification language provides for a polynomial-time reduction from classical propositional logic to  $\mathcal{JL}_{\mathcal{CS}}$ . Classical propositional logic was shown to be co-NP-hard by Stephen Cook in [Coo71].  $\square$

**Theorem 5.3.3** ([Mil07]).

1.  $\mathcal{J4}_{\mathcal{CS}}$  with a decidable schematic  $\mathcal{CS}$  is  $\Pi_2^p$ -hard.
2.  $\mathcal{LP}_{\mathcal{CS}}$  with a decidable schematically injective axiomatically appropriate  $\mathcal{CS}$  is  $\Pi_2^p$ -hard.

**Corollary 5.3.4** ([Mil07]).

1.  $J4_{\mathcal{CS}}$  with a decidable schematic  $\mathcal{CS}$  is  $\Pi_2^p$ -complete.
2.  $LP_{\mathcal{CS}}$  with a decidable, schematically injective, and axiomatically appropriate  $\mathcal{CS}$  is  $\Pi_2^p$ -complete.

**Corollary 5.3.5** ([Mil07]).  $J4$  is  $\Pi_2^p$ -complete.

*Note 5.3.6.*  $\mathcal{TCS}_{LP}$  is not schematically injective, so Theorem 5.3.3 does not give a lower bound on the complexity of LP itself.

There exists an elegant reduction of the Satisfiability Problem for  $\text{Int}$  to the Satisfiability Problem for  $J\mathcal{D}_{\mathcal{CS}}$  for a certain schematic though not axiomatically appropriate  $\mathcal{CS}$ :

**Lemma 5.3.7.** *Consider the axiomatization of classical propositional logic that consists of a complete axiomatization of the intuitionistic propositional logic  $\text{Int}$  with the law of double negation  $\neg\neg F \rightarrow F$  as an additional axiom scheme. Let*

$$\mathcal{CS} = \{c:A \mid A \text{ is an intuitionistic axiom instance}\} .$$

*Clearly such  $\mathcal{CS}$  is a decidable schematic constant specification for any justification logic.*

Let  $x$  be a fixed justification variable,  $Q$  be any propositional formula. The following statements are equivalent.

1.  $Q$  is  $\text{Int}$ -satisfiable
2.  $\text{Int} \not\vdash \neg Q$
3.  $Q \not\vdash_{\text{Int}} \perp$
4. there is no justification term  $t$  such that  $*(x, Q) \vdash_{*_{\text{CS}}} *(t, \perp)$
5.  $x:Q$  is  $\text{JD}_{\text{CS}}$ -satisfiable

But this reduction does not entail PSPACE-hardness of this  $\text{JD}_{\text{CS}}$ . Unlike classical logics, where the complexity of the validity problem is typically dual to the complexity of the satisfiability problem (cf. SAT is NP-complete, whereas classical propositional logic is co-NP-complete), intuitionistic logic is different. As noted in [Šve03, Remark 2 on p.715], “the set of all intuitionistically satisfiable formulas equals the set SAT of all classically satisfiable formulas.” This statement easily follows from Glivenko Theorem, for instance.

Therefore, rather counterintuitively, the Satisfiability Problem for  $\text{Int}$  is NP-complete even though the Validity Problem is PSPACE-complete.

So the reduction in Lemma 5.3.7 does not improve the trivial co-NP-hard lower bound for  $\text{JD}_{CS}$ . For this reason we omit the proof of Lemma 5.3.7 here.

## 5.4 Complexity of Hybrid Logics

**Theorem 5.4.1.**

- $\text{T}_n\text{LP}_{CS}$ ,  $n \geq 1$ , is *PSPACE-hard*.
- $\text{S4}_n\text{LP}_{CS}$ ,  $n \geq 1$ , is *PSPACE-hard*.
- $\text{S5}_n\text{LP}_{CS}$ ,  $n \geq 2$ , is *PSPACE-hard*.
- $\text{S5}_1\text{LP}_{CS}$  is *co-NP-hard*.

*Proof.* The results follow from

**Lemma 5.4.2.**

- $\text{T}_n\text{LP}_{CS}$  is *conservative over*  $\text{T}_n$ .
- $\text{S4}_n\text{LP}_{CS}$  is *conservative over*  $\text{S4}_n$ .
- $\text{S5}_n\text{LP}_{CS}$  is *conservative over*  $\text{S5}_n$ .

*Proof.* We will prove conservativity semantically. We need to prove that any modal formula  $\varphi \in \mathcal{ML}_n$  is derivable in a hybrid logic iff it is derivable in the

corresponding modal logic, i.e., in the modal logic whose name is obtained by omitting the  $\text{LP}_{CS}$  suffix from the name of the hybrid logic:

$$\mathbf{M}_n \vdash \varphi \quad \iff \quad \mathbf{M}_n \text{LP}_{CS} \vdash \varphi ,$$

where  $\mathbf{M} \in \{\mathbf{T}, \mathbf{S4}, \mathbf{S5}\}$ .

The  $\implies$  direction is trivial since any  $\mathbf{M}_n$ -derivation is also an  $\mathbf{M}_n \text{LP}_{CS}$ -derivation.

Let us now prove the  $\impliedby$  direction, or rather its contrapositive. Suppose  $\mathbf{M}_n \not\vdash \varphi$ . By completeness of  $\mathbf{M}_n$  w.r.t. its Kripke models there exists a Kripke model  $\mathfrak{K} = (W, R_1, \dots, R_n, V)$  and a world  $w \in W$  such that  $\mathfrak{K}, w \not\models \varphi$ .

Let  $\mathcal{A}_{\text{tot}}$  be the total evidence function from (3.3.31), i.e.

$$\mathcal{A}_{\text{tot}}(t, F) = W$$

for any term  $t$  and any hybrid formula  $F$ . This function was shown to be an admissible evidence function for any hybrid logic on any model with any  $R_e$  in the proof of Theorem 3.3.34.

Let  $R_e = W \times W$ . Clearly, such a binary relation is reflexive, symmetric, and transitive. It also contains all  $R_i$ , whatever they are.

The conditions on  $R_i$  are the same for Kripke models of  $\mathbf{M}_n$  and for AF-models of  $\mathbf{M}_n \text{LP}_{CS}$ .

Therefore,  $\mathfrak{M} = (W, R_e, R_1, \dots, R_n, V, \mathcal{A})$  with  $W$ ,  $R_i$ 's, and  $V$  taken from  $\mathfrak{K}$  is an AF-model for  $M_nLP_{CS}$ .

It remains to note that the definition of  $\Vdash$  for purely modal formulas in Kripke models coincides with that for AF-models. Thus,  $\mathfrak{M}, w \not\vdash \varphi$  for the same world  $w$ , and  $M_nLP_{CS} \not\vdash \varphi$ .

This completes the proof of Lemma 5.4.2. □

We are now ready to finish the proof of Theorem 5.4.1. As usual in the presence of conservativity, the identity function from  $\mathcal{ML}_n$  into  $\mathcal{HL}_n$  provides for a polynomial-time reduction from  $M_n$  to  $M_nLP_{CS}$ . Therefore, the lower bounds for hybrid logics follow from PSPACE-hardness of  $T_n$ ,  $S4_n$  for  $n \geq 1$  and  $S5_n$  for  $n \geq 2$  proved by Joseph Halpern and Yoram Moses in [HM85, HM92] as well as from co-NP-hardness of  $S5_1$  proved by Richard Ladner in [Lad77] (Ladner also proved PSPACE-hardness of  $T_1$  and  $S4_1$  there). □

*Note 5.4.3.* Conservativity is usually proved by a derivability-preserving translation from the richer language to the more basic one as in the proof of Lemma 5.3.2. But certain difficulties arise in constructing such a translation for Lemma 5.4.2. The goal is to translate the hybrid language  $\mathcal{HL}_n$  into the multimodal language  $\mathcal{ML}_n$ . In particular, we need the translation of axiom

$t:F \rightarrow K_i F$  to be valid for each  $1 \leq i \leq n$ . Thus,  $t:F$  should be translated as at least  $EF = K_1 F \wedge \dots \wedge K_n F$ . But even that is not enough. In addition, the translation of  $t:F \rightarrow !t:t:F$  has to be derivable in the respective modal logic. If we choose to translate  $t:F$  as  $EF$ , this Positive Introspection axiom will be translated as  $EF \rightarrow EEF$ , which is not derivable even in  $S5_n$  for  $n \geq 2$ . Indeed, already for  $n = 2$ ,  $EF \rightarrow EEF$  stands for

$$K_1 F \wedge K_2 F \rightarrow K_1(K_1 F \wedge K_2 F) \wedge K_2(K_1 F \wedge K_2 F) .$$

There is no reason why  $K_1 F \wedge K_2 F$  should entail  $K_1 K_2 F$  or  $K_2 K_1 F$ . This example shows that the translation of  $t:F$  should be something akin to common knowledge, which is not present in the hybrid language.

**Theorem 5.4.4** ([Kuz06a]). *S4LP<sub>CS</sub> with a decidable schematic CS is PSPACE-complete.*

*Note 5.4.5.* Since there is only one modality in S4LP, we will use  $\Box$  instead of  $K_1$ .

*Proof.* The lower bound, PSPACE-hardness of S4LP<sub>CS</sub> = S4<sub>1</sub>LP<sub>CS</sub> was proven in Theorem 5.4.1.

The upper bound is proven by generalizing and modifying Ladner's decision algorithm for S4 from [Lad77]. We describe a recursive procedure

S4LP<sub>CS</sub>-WORLD that tries to construct an F-model  $\mathfrak{M} = (W, R, V, \mathcal{A})$  refuting the given formula  $F$  if such a model exists.

The procedure has seven parameters

$$\langle \mathcal{T}, \mathcal{F}, \mathcal{T}^\square, \mathcal{F}^\square, \mathcal{T}^*, \mathcal{F}^*, \mathcal{L} \rangle ,$$

where

- $\mathcal{T}$  and  $\mathcal{F}$  are finite sets of hybrid formulas;
- $\mathcal{T}^\square$  and  $\mathcal{F}^\square$  are finite sets of *boxed* formulas, i.e. formulas of form  $\square C$ ;
- $\mathcal{T}^*$  and  $\mathcal{F}^*$  are finite sets of *\**-expressions of form  $*(s, C)$ ;
- $\mathcal{L}$  is a triple  $(\mathcal{T}^\square, \mathcal{T}^*, \langle B_1, \dots, B_k \rangle)$ , where
  - $\mathcal{T}^\square$  is a finite (possibly empty) set of boxed formulas,
  - $\mathcal{T}^*$  is a finite (possibly empty) set of *\**-expressions, and
  - $\langle B_1, \dots, B_k \rangle$  is a sequence (possibly empty) of hybrid formulas.

The intuitive understanding is that each call of the procedure describes the conditions imposed on one world of the future model. The world is not present explicitly. We will denote it by  $w$ .

- $\mathcal{T}$  is a set of formulas that have to be true at  $w$  in the future model.

- $\mathcal{T}^\square$  is a set of boxed formulas that have to be true at  $w$  in the future model.
- $\mathcal{F}$  is a set of formulas that have to be false at  $w$  in the future model.
- $\mathcal{F}^\square$  is a set of boxed formulas that have to be false at  $w$  in the future model.
- $\mathcal{T}^*$  is a set of \*-expressions that has to be a subset of  $\mathcal{A}_w^*$  for the future admissible evidence function  $\mathcal{A}$ , i.e.  $\mathcal{T}^* \subseteq \mathcal{A}_w^*$ .
- $\mathcal{F}^*$  is a set of \*-expressions that has to be disjoint from  $\mathcal{A}_w^*$  for the future admissible evidence function  $\mathcal{A}$ , i.e.,  $\mathcal{F}^* \cap \mathcal{A}_w^* = \emptyset$ .
- Finally,  $\mathcal{L}$  represents a log of the previous recursive calls and is kept to prevent the algorithm from looping.

In order to determine whether  $F$  is a theorem of  $\mathbf{S4LP}_{CS}$  or, equivalently, whether  $F$  is valid in all F-models for  $\mathbf{S4LP}_{CS}$ , we start the procedure  $\mathbf{S4LP}_{CS}$ -WORLD on input

$$\langle \emptyset, \{F\}, \emptyset, \emptyset, \emptyset, \emptyset, (\emptyset, \emptyset, \lambda) \rangle ,$$

where  $\lambda$  stands for the empty sequence. In other words, we only need  $F \in \mathcal{F}$  to be false at some world of the future model. We want the procedure

to return **true** iff such a world and such a model exist. The procedure is described in Fig. 5.4.1.

In Step 11 of the procedure,

$$(\mathcal{T}^\square, \mathcal{T}^*, B) \in \mathcal{L}$$

is a shorthand for the statement that

$$\mathcal{L} = (\mathcal{T}^\square, \mathcal{T}^*, \langle B_1, \dots, B_k \rangle)$$

with  $B = B_i$  for some  $1 \leq i \leq k$ . Accordingly, the condition

$$(\mathcal{T}^\square, \mathcal{T}^*, B) \notin \mathcal{L}$$

in the subscript of the second big conjunction is simply the negation of  $(\mathcal{T}^\square, \mathcal{T}^*, B) \in \mathcal{L}$ .

Operation  $\odot$  in Step 11 is defined as follows:

$$\mathcal{L} \odot \langle \mathcal{T}^\square, \mathcal{T}^*, B \rangle = \begin{cases} (\mathcal{T}^\square, \mathcal{T}^*, \langle B \rangle) & \text{if } \mathcal{L} = (\emptyset, \emptyset, \lambda) \\ (\mathcal{T}^\square, \mathcal{T}^*, \langle B_1, \dots, B_k, B \rangle) & \text{if } \mathcal{L} = (\mathcal{T}^\square, \mathcal{T}^*, \langle B_1, \dots, B_k \rangle) \\ (\mathcal{T}^\square, \mathcal{T}^*, \langle B \rangle) & \text{if } \mathcal{L} = (\mathcal{T}_0^\square, \mathcal{T}_0^*, \langle B_1, \dots, B_k \rangle) \text{ and} \\ & \mathcal{T}^\square \supsetneq \mathcal{T}_0^\square \text{ or } \mathcal{T}^* \supsetneq \mathcal{T}_0^* \end{cases}$$

In Step 11, procedure  $\text{S4LP}_{\mathcal{CS}}\text{-WORLD}$  uses an external subroutine  $*!_{\mathcal{CS}}\text{-DERIVE}$  from p. 196.

$$*!_{\mathcal{CS}}\text{-DERIVE}(\mathcal{T}^*, *(t, B))$$

Figure 5.4.1: Recursive procedure S4LP<sub>CS</sub>-WORLD

```

procedure S4LPCS-WORLD( $\mathcal{T}, \mathcal{F}, \mathcal{T}^\square, \mathcal{F}^\square, \mathcal{T}^*, \mathcal{F}^*, \mathcal{L}$ );
begin
  if  $\mathcal{T} \cup \mathcal{F} \not\subseteq SLet$  then
    begin
      1. choose  $G \in \mathcal{T} \cup \mathcal{F} \setminus SLet$ ;
      2. if  $G = \perp \in \mathcal{T}$  then return false;
      3. if  $G = \perp \in \mathcal{F}$  then return
         S4LPCS-WORLD( $\mathcal{T}, \mathcal{F} \setminus \{\perp\}, \mathcal{T}^\square, \mathcal{F}^\square, \mathcal{T}^*, \mathcal{F}^*, \mathcal{L}$ );
      4. if  $G = B \rightarrow C \in \mathcal{T}$  then return
         S4LPCS-WORLD( $\mathcal{T} \cup \{C\} \setminus \{B \rightarrow C\}, \mathcal{F}, \mathcal{T}^\square, \mathcal{F}^\square, \mathcal{T}^*, \mathcal{F}^*, \mathcal{L}$ )  $\vee$ 
         S4LPCS-WORLD( $\mathcal{T} \setminus \{B \rightarrow C\}, \mathcal{F} \cup \{B\}, \mathcal{T}^\square, \mathcal{F}^\square, \mathcal{T}^*, \mathcal{F}^*, \mathcal{L}$ );
      5. if  $G = B \rightarrow C \in \mathcal{F}$  then return
         S4LPCS-WORLD( $\mathcal{T} \cup \{B\}, \mathcal{F} \cup \{C\} \setminus \{B \rightarrow C\}, \mathcal{T}^\square, \mathcal{F}^\square, \mathcal{T}^*, \mathcal{F}^*, \mathcal{L}$ );
      6. if  $G = \Box B \in \mathcal{T}$  then return
         S4LPCS-WORLD( $\mathcal{T} \cup \{B\} \setminus \{\Box B\}, \mathcal{F}, \mathcal{T}^\square \cup \{\Box B\}, \mathcal{F}^\square, \mathcal{T}^*, \mathcal{F}^*, \mathcal{L}$ );
      7. if  $G = \Box B \in \mathcal{F}$  then return
         S4LPCS-WORLD( $\mathcal{T}, \mathcal{F} \setminus \{\Box B\}, \mathcal{T}^\square, \mathcal{F}^\square \cup \{\Box B\}, \mathcal{T}^*, \mathcal{F}^*, \mathcal{L}$ );
      8. if  $G = t:B \in \mathcal{T}$  then return
         S4LPCS-WORLD( $\mathcal{T} \cup \{\Box B\} \setminus \{t:B\}, \mathcal{F}, \mathcal{T}^\square, \mathcal{F}^\square, \mathcal{T}^* \cup \{*(t, B)\}, \mathcal{F}^*, \mathcal{L}$ );
      9. if  $G = t:B \in \mathcal{F}$  then return
         S4LPCS-WORLD( $\mathcal{T}, \mathcal{F} \setminus \{t:B\}, \mathcal{T}^\square, \mathcal{F}^\square, \mathcal{T}^*, \mathcal{F}^* \cup \{*(t, B)\}, \mathcal{L}$ )  $\vee$ 
         S4LPCS-WORLD( $\mathcal{T}, \mathcal{F} \cup \{\Box B\} \setminus \{t:B\}, \mathcal{T}^\square, \mathcal{F}^\square, \mathcal{T}^*, \mathcal{F}^*, \mathcal{L}$ );
    end;
  if  $\mathcal{T} \cup \mathcal{F} \subseteq SLet$  then
    begin
      10. if  $\mathcal{T} \cap \mathcal{F} \neq \emptyset$  then return false;
      11. if  $\mathcal{T} \cap \mathcal{F} = \emptyset$  then return
          $\bigwedge_{*(t,B) \in \mathcal{F}^*} \neg \left( *!_{CS}\text{-DERIVE}(\mathcal{T}^*, *(t, B)) \right) \wedge$ 
          $\bigwedge_{\substack{\Box B \in \mathcal{F}^\square \\ (\mathcal{T}^\square, \mathcal{T}^*, B) \notin \mathcal{L}}} \text{S4LP}_{CS}\text{-WORLD}(\mathcal{T}^\square \cup (\mathcal{T}^*)^\cdot, \{B\}, \mathcal{T}^\square, \emptyset, \mathcal{T}^*, \emptyset, \mathcal{L} \odot (\mathcal{T}^\square, \mathcal{T}^*, B));$ 
    end;
end.

```

returns **true** iff

$$\mathcal{T}^* \vdash_{*!_{CS}} *(t, B) .$$

Note that, whenever that happens, the current  $S4LP_{CS}$ -WORLD call immediately returns **false**.

To prove correctness, we will show how to extract a refuting F-model for  $F$  from a successful run of

$$S4LP_{CS}\text{-WORLD}\langle\emptyset, \{F\}, \emptyset, \emptyset, \emptyset, \emptyset, (\emptyset, \emptyset, \lambda)\rangle$$

This model will be based on a tree of polynomial in  $|F|$  depth. Of course, such a tree itself may be exponential in  $|F|$ . This does not prevent the procedure from using only polynomial space. Our procedure will be traversing this tree one node at a time. At any moment, the procedure will see only a single node and store certain information about the parent nodes from the (polynomial) branch of the current node.

**Lemma 5.4.6 (Correctness of  $S4LP_{CS}$ -WORLD).** *The call of procedure*

$$S4LP_{CS}\text{-WORLD}\langle\emptyset, \{F\}, \emptyset, \emptyset, \emptyset, \emptyset, (\emptyset, \emptyset, \lambda)\rangle \quad (5.4.1)$$

*returns **true** iff there exist an F-model  $\mathfrak{M} = (W, R, V, \mathcal{A})$  for  $S4LP_{CS}$  and a world  $\Gamma \in W$  such that  $\mathfrak{M}, \Gamma \not\models F$ .*

*Proof.* Let us first prove that the desired countermodel exists if **true** is returned as the result of call (5.4.1).

Consider the successful run of our procedure. This run consists of many successful recursive calls of  $\text{S4LP}_{\mathcal{CS}}\text{-WORLD}$  (a *successful call* is a call that returns **true**). There may have been some unsuccessful calls too that did not affect the final returned value. From now on we disregard all such unsuccessful calls.

The calls made in Step 11 of  $\text{S4LP}_{\mathcal{CS}}\text{-WORLD}$  (see Fig. 5.4.1) will be referred to as *essential*; all the other calls are *local*. The initial call (5.4.1) is also considered essential. We will associate a world  $\Gamma_{\mathcal{L}}$  with each essential call

$$\text{S4LP}_{\mathcal{CS}}\text{-WORLD}\langle \mathcal{T}, \mathcal{F}, \mathcal{T}^{\square}, \mathcal{F}^{\square}, \mathcal{T}^*, \mathcal{F}^*, \mathcal{L} \rangle .$$

We will refer to all essential calls with the last parameter  $\mathcal{L}$  as to  $\mathcal{L}$ -calls, because  $\mathcal{L}$  uniquely defines the essential call for each computation branch.<sup>1</sup>

For each  $\mathcal{L}$ -call, where

$$\mathcal{L} = (\mathcal{T}^{\square}, \mathcal{T}^*, \langle B_1, \dots, B_k \rangle)$$

---

<sup>1</sup>There may be different  $\mathcal{L}$ -calls on different branches, so it could be better to encode the essential calls and worlds associated with them by the full list of all parameters of the call rather than just the last parameter. But this would create enormously long subscripts for the worlds, which would greatly impact readability. At the same time  $\mathcal{L}$  is sufficient to identify a call within each branch, which prompted us to use this potentially ambiguous notation.

with  $k \geq 1$ , i.e., for each essential call with the exception of the initial call (5.4.1), the closest essential call preceding this  $\mathcal{L}$ -call in the run of (5.4.1) is uniquely defined. We will refer to this closest essential preceding call as the  $\mathcal{L}^{-1}$ -call

$$\text{S4LP}_{\mathcal{CS}\text{-WORLD}}\langle \mathcal{T}_0, \mathcal{F}_0, \mathcal{T}_0^\square, \mathcal{F}_0^\square, \mathcal{T}_0^*, \mathcal{F}_0^*, \mathcal{L}^{-1} \rangle ,$$

where

$$\mathcal{L}^{-1} = \begin{cases} (\mathcal{T}^\square, \mathcal{T}^*, \langle B_1, \dots, B_{k-1} \rangle) & \text{if } \mathcal{T}_0^\square = \mathcal{T}^\square, \mathcal{T}_0^* = \mathcal{T}^*, \text{ and } k \geq 2 ; \\ (\mathcal{T}_0^\square, \mathcal{T}_0^*, \langle C_1, \dots, C_l \rangle) & \text{if } \mathcal{T}_0^\square \subsetneq \mathcal{T}^\square, \mathcal{T}_0^* \subseteq \mathcal{T}^*, \text{ and } k = 1 \text{ or} \\ & \mathcal{T}_0^\square \subseteq \mathcal{T}^\square, \mathcal{T}_0^* \subsetneq \mathcal{T}^*, \text{ and } k = 1 ; \\ (\emptyset, \emptyset, \lambda) & \text{if } k = 1 \text{ and } \mathcal{L} \text{ is the second essential} \\ & \text{call on its branch.} \end{cases}$$

For each  $\mathcal{L}$ -call let  $\mathcal{T}$ ,  $\mathcal{F}$ ,  $\mathcal{T}^\square$ ,  $\mathcal{F}^\square$ ,  $\mathcal{T}^*$ , and  $\mathcal{F}^*$  be parameters of the closest consecutive call after this  $\mathcal{L}$ -call that will use Step 11 (this future call may be either essential or terminal). We will denote these sets by  $\mathcal{T}_{\mathcal{L}}$ ,  $\mathcal{F}_{\mathcal{L}}$ ,  $\mathcal{T}_{\mathcal{L}}^\square$ ,  $\mathcal{F}_{\mathcal{L}}^\square$ ,  $\mathcal{T}_{\mathcal{L}}^*$ , and  $\mathcal{F}_{\mathcal{L}}^*$  respectively.

For each computation branch, several local calls are generally made between any two consecutive essential calls. Let the earlier of these two calls be an  $\mathcal{L}$ -call. In the course of the intermediate local calls, formulas are being chosen from  $\mathcal{T}$  and  $\mathcal{F}$  to be discharged in Steps 2–9. Imagine an alternative procedure where exact same formulas are chosen in the same order, and exact same intermediate local calls are made with the only exception that

the chosen formulas are never discharged from parameters  $\mathcal{T}$  or  $\mathcal{F}$ . Let  $\overline{\mathcal{T}}_{\mathcal{L}}$  and  $\overline{\mathcal{F}}_{\mathcal{L}}$  denote the first two parameters that would have resulted in such an alternative run right before the next essential call.

We are now ready to define the countermodel.

- *The set of worlds  $W$*  consists of all  $\Gamma_{\mathcal{L}}$  for essential  $\mathcal{L}$ -calls in the original successful run of (5.4.1).
- *Accessibility relation.* For each essential  $\mathcal{L}$ -call other than (5.4.1) let

$$\Gamma_{\mathcal{L}^{-1}} R_0 \Gamma_{\mathcal{L}} . \quad (5.4.2)$$

Let also

$$\Gamma_{(\mathcal{T}^{\square}, \mathcal{T}^*, \langle B_1, \dots, B_k \rangle)} R_0 \Gamma_{(\mathcal{T}^{\square}, \mathcal{T}^*, \langle B_1 \rangle)} , \quad (5.4.3)$$

provided that the essential calls corresponding to the former and latter worlds occur on the same computation branch of the tree in the opposite order, i.e., first the  $(\mathcal{T}^{\square}, \mathcal{T}^*, \langle B_1 \rangle)$ -call and then the  $\mathcal{L} = (\mathcal{T}^{\square}, \mathcal{T}^*, \langle B_1, \dots, B_k \rangle)$ -call. In addition, we require that  $\mathcal{T}_{\mathcal{L}}^{\square} = \mathcal{T}^{\square}$  and  $\mathcal{T}_{\mathcal{L}}^* = \mathcal{T}^*$ .

Let  $R$  be the reflexive and transitive closure of  $R_0$ .

- *Admissible evidence function.* We define an F-type possible evidence

function  $\mathcal{B}$  such that for any essential call  $\mathcal{L}$  and corresponding world  $\Gamma_{\mathcal{L}}$

$$\Gamma_{\mathcal{L}} \in \mathcal{B}(t, G) \quad \iff \quad *(t, G) \in \mathcal{T}_{\mathcal{L}}^* \quad (5.4.4)$$

Let  $\mathcal{A}$  be the minimal F-type admissible for  $\text{S4LP}_{\mathcal{CS}}$  evidence function based on  $\mathcal{B}$ , defined according to (3.5.9).

- *Propositional valuation*  $V$  is defined for each essential  $\mathcal{L}$ -call and corresponding world  $\Gamma_{\mathcal{L}}$  by

$$\Gamma_{\mathcal{L}} \in V(p) \quad \iff \quad p \in \mathcal{T}_{\mathcal{L}} \quad (5.4.5)$$

It is easy to see that  $\mathfrak{M} = (W, R, V, \mathcal{A})$  is indeed an F-model for  $\text{S4LP}_{\mathcal{CS}}$ .

- $W \neq \emptyset$  because each run has at least one essential call, namely the initial call (5.4.1).
- Being a reflexive transitive closure, clearly  $R$  is reflexive and transitive.
- $\mathcal{A}$  is an admissible evidence function by Theorem 3.5.20.

Our goal is to show that for the initial call (5.4.1)

$$\mathfrak{M}, \Gamma_{(\emptyset, \emptyset, \lambda)} \not\models F \quad . \quad (5.4.6)$$

We will prove a more general fact:

**Lemma 5.4.7 (Truth Lemma).** *For each essential call  $\mathcal{L}$  and corresponding world  $\Gamma_{\mathcal{L}}$*

$$G \in \overline{\mathcal{T}}_{\mathcal{L}} \quad \Longrightarrow \quad \mathfrak{M}, \Gamma_{\mathcal{L}} \Vdash G \quad (5.4.7)$$

$$G \in \overline{\mathcal{F}}_{\mathcal{L}} \quad \Longrightarrow \quad \mathfrak{M}, \Gamma_{\mathcal{L}} \nVdash G \quad (5.4.8)$$

*Proof.* Induction on complexity of  $G$ .

$p \in \overline{\mathcal{T}}_{\mathcal{L}}$ . Sentence letters are never discharged by S4LP<sub>CS</sub>-WORLD, hence

$p \in \mathcal{T}_{\mathcal{L}}$ . Therefore,  $\Gamma_{\mathcal{L}} \in V(p)$  by (5.4.5), and  $\mathfrak{M}, \Gamma_{\mathcal{L}} \Vdash p$  by (3.3.15).

$p \in \overline{\mathcal{F}}_{\mathcal{L}}$ . Again  $p \in \mathcal{F}_{\mathcal{L}}$ . The  $\mathcal{L}$ -call was successful, so by Step 10  $\mathcal{F}_{\mathcal{L}} \cap \mathcal{T}_{\mathcal{L}} = \emptyset$ , and

$p \notin \mathcal{T}_{\mathcal{L}}$ . Therefore,  $\Gamma_{\mathcal{L}} \notin V(p)$  by (5.4.5), and  $\mathfrak{M}, \Gamma_{\mathcal{L}} \nVdash p$  by (3.3.15).

$\perp \in \overline{\mathcal{T}}_{\mathcal{L}}$ . The  $\mathcal{L}$ -call was successful, so by Step 2, this cannot happen.

$\perp \in \overline{\mathcal{F}}_{\mathcal{L}}$ .  $\mathfrak{M}, \Gamma_{\mathcal{L}} \nVdash \perp$  by (3.3.16).

$B \rightarrow C \in \overline{\mathcal{T}}_{\mathcal{L}}$ . By Step 4, either  $B \in \overline{\mathcal{F}}_{\mathcal{L}}$  or  $C \in \overline{\mathcal{T}}_{\mathcal{L}}$ . By IH, either  $\mathfrak{M}, \Gamma_{\mathcal{L}} \nVdash B$  or

$\mathfrak{M}, \Gamma_{\mathcal{L}} \Vdash C$ . In either case  $\mathfrak{M}, \Gamma_{\mathcal{L}} \Vdash B \rightarrow C$  by (3.3.17).

$B \rightarrow C \in \overline{\mathcal{F}}_{\mathcal{L}}$ . By Step 5,  $B \in \overline{\mathcal{T}}_{\mathcal{L}}$  and  $C \in \overline{\mathcal{F}}_{\mathcal{L}}$ . By IH,  $\mathfrak{M}, \Gamma_{\mathcal{L}} \Vdash B$  and  $\mathfrak{M}, \Gamma_{\mathcal{L}} \nVdash C$ .

Thus,  $\mathfrak{M}, \Gamma_{\mathcal{L}} \nVdash B \rightarrow C$  by (3.3.17).

$\Box B \in \overline{\mathcal{T}}_{\mathcal{L}}$ . For  $\Box B$  to be true at  $\Gamma_{\mathcal{L}}$  it is sufficient to show that

$$\Gamma_{\mathcal{L}} R \Gamma_{\mathcal{L}'} \quad \Longrightarrow \quad B \in \overline{\mathcal{T}}_{\mathcal{L}'} \quad (5.4.9)$$

Then, by IH, we will have  $\mathfrak{M}, \Gamma_{\mathcal{L}'} \Vdash B$  for all  $\Gamma_{\mathcal{L}} R \Gamma_{\mathcal{L}'}$  and hence  $\mathfrak{M}, \Gamma_{\mathcal{L}} \Vdash \Box B$  by (2.4.4).

According to Step 6,  $B \in \overline{\mathcal{T}}_{\mathcal{L}'}$  whenever  $\Box B \in \overline{\mathcal{T}}_{\mathcal{L}'}$ , so we will show

$$\Gamma_{\mathcal{L}} R \Gamma_{\mathcal{L}'} \quad \Longrightarrow \quad \Box B \in \overline{\mathcal{T}}_{\mathcal{L}'} \quad (5.4.10)$$

Since  $R$  is the reflexive and transitive closure of  $R_0$ , to show (5.4.10), we need to show that

- $\Box B \in \overline{\mathcal{T}}_{\mathcal{L}}$  and
- $\Gamma_{\mathcal{L}_1} R_0 \Gamma_{\mathcal{L}_2}$  and  $\Box B \in \overline{\mathcal{T}}_{\mathcal{L}_1} \quad \Longrightarrow \quad \Box B \in \overline{\mathcal{T}}_{\mathcal{L}_2}$

The former condition holds. Let us prove the latter. Assume that  $\Box B \in \overline{\mathcal{T}}_{\mathcal{L}_1}$ .  $\Gamma_{\mathcal{L}_1} R_0 \Gamma_{\mathcal{L}_2}$  may hold because of either (5.4.2) or (5.4.3).

(5.4.2)  $\mathcal{L}_1 = \mathcal{L}_2^{-1}$ . Then, by Step 11,  $\mathcal{L}_2$ -call has been initiated with

$$\mathcal{T}_{\mathcal{L}_1}^{\Box} \subseteq \mathcal{T}. \text{ By Step 6, } \Box B \in \mathcal{T}_{\mathcal{L}_1}^{\Box}. \text{ Hence, } \Box B \in \overline{\mathcal{T}}_{\mathcal{L}_2}.$$

(5.4.3)  $\mathcal{L}_1 = (\mathcal{T}^{\Box}, \mathcal{T}^*, \langle B_1, \dots, B_k \rangle)$  follows  $\mathcal{L}_2 = (\mathcal{T}^{\Box}, \mathcal{T}^*, \langle B_1 \rangle)$  on a

computation branch, where  $\mathcal{T}_{\mathcal{L}_1}^{\Box} = \mathcal{T}^{\Box}$  and  $\mathcal{T}_{\mathcal{L}_1}^* = \mathcal{T}^*$ . Note that

the parameter  $\mathcal{T}^{\Box}$  is non-decreasing along each branch and that

this parameter for any  $\mathcal{L}$ -call always coincides with the first ele-

ment in  $\mathcal{L}$ . Therefore,

$$\mathcal{T}^{\Box} \subseteq \mathcal{T}_{\mathcal{L}_2}^{\Box} \subseteq \mathcal{T}_{\mathcal{L}_1}^{\Box} = \mathcal{T}^{\Box} .$$

It follows that  $\mathcal{T}_{\mathcal{L}_2}^\square = \mathcal{T}_{\mathcal{L}_1}^\square$  and  $\square B \in \mathcal{T}_{\mathcal{L}_2}^\square$ . There are two ways how

$\square B$  could appear in  $\mathcal{T}_{\mathcal{L}_2}^\square$ :

in Step 6 or

in the  $\mathcal{L}_2$ -call itself (Step 11).

In either case  $\square B \in \overline{\mathcal{T}}_{\mathcal{L}_2}$ .

This completes the proof of (5.4.10).

$\square B \in \overline{\mathcal{F}}_{\mathcal{L}}$ . By Step 7,  $\square B \in \mathcal{F}_{\mathcal{L}}^\square$ . Therefore, at the closest consecutive Step 11

– either an  $\mathcal{L}'$ -call was made with parameters

$$\text{S4LP}_{CS}\text{-WORLD}\langle \mathcal{T}_{\mathcal{L}}^\square \cup (\mathcal{T}_{\mathcal{L}}^*)^\dagger, \{B\}, \mathcal{T}_{\mathcal{L}}^\square, \emptyset, \mathcal{T}_{\mathcal{L}}^*, \emptyset, \mathcal{L}' \rangle, \quad (5.4.11)$$

so that  $\Gamma_{\mathcal{L}} R_0 \Gamma_{\mathcal{L}'}$  and  $B \in \overline{\mathcal{F}}_{\mathcal{L}'}$ . By IH,  $\mathfrak{M}, \Gamma_{\mathcal{L}'} \not\# B$ . Clearly,

$\Gamma_{\mathcal{L}} R \Gamma_{\mathcal{L}'}$ , hence  $\mathfrak{M}, \Gamma_{\mathcal{L}} \not\# \square B$  by (2.4.4).

– Or call (5.4.11) was not made because

$$\mathcal{L} = (\mathcal{T}_{\mathcal{L}}^\square, \mathcal{T}_{\mathcal{L}}^*, \langle B_1, \dots, B_k \rangle),$$

with  $B = B_i$  for some  $1 \leq i \leq k$ . In this case, there must have

been a sequence of preceding  $\mathcal{L}_j$ -calls,  $j = 1, \dots, k$  on the same

branch with

$$\mathcal{L}_j = (\mathcal{T}_{\mathcal{L}}^\square, \mathcal{T}_{\mathcal{L}}^*, \langle B_1, \dots, B_j \rangle)$$

the last of them being  $\mathcal{L}_k = \mathcal{L}$  itself such that for  $j = 1, \dots, k-1$

$$\Gamma_{\mathcal{L}_j} R_0 \Gamma_{\mathcal{L}_{j+1}} \quad (5.4.12)$$

Moreover, a prerequisite for call (5.4.11) not to be initiated is that  $\mathcal{T}^\square$  and  $\mathcal{T}^*$  do not enlarge between the  $\mathcal{L}$ -call and the immediately following Step 11. This is sufficient to conclude by (5.4.3) that

$$\Gamma_{\mathcal{L}} R_0 \Gamma_{\mathcal{L}_1} . \quad (5.4.13)$$

Since  $R$  is the transitive closure of  $R_0$  it follows from (5.4.12) and (5.4.13) that

$$\Gamma_{\mathcal{L}} R \Gamma_{\mathcal{L}_i} .$$

On the other hand, it is clear that the parameters of  $\mathcal{L}_i$ -call must have been

$$\text{S4LP}_{\mathcal{CS}\text{-WORLD}} \langle \mathcal{T}_{\mathcal{L}}^\square \cup (\mathcal{T}_{\mathcal{L}}^*)^\dagger, \{B_i\}, \mathcal{T}_{\mathcal{L}}^\square, \emptyset, \mathcal{T}_{\mathcal{L}}^*, \emptyset, \mathcal{L}_i \rangle ,$$

which means that  $B = B_i \in \overline{\mathcal{F}}_{\mathcal{L}_i}$  and  $\mathfrak{M}, \Gamma_{\mathcal{L}_i} \not\ll B$  by IH. Since  $\Gamma_{\mathcal{L}} R \Gamma_{\mathcal{L}_i}$ , again  $\mathfrak{M}, \Gamma_{\mathcal{L}} \not\ll \square B$  by (2.4.4).

$t:B \in \overline{\mathcal{T}}_{\mathcal{L}}$ . By Step 8,  $\square B \in \overline{\mathcal{T}}_{\mathcal{L}}$  and  $*(t, B) \in \mathcal{T}_{\mathcal{L}}^*$ . Size of  $t:B$

$$|t:B| = |t| + 1 + |B| > |B| + 1 = |\square B| ,$$

so  $\mathfrak{M}, \Gamma_{\mathcal{L}} \Vdash \Box B$  by IH, which means that  $B$  is true in all the worlds accessible from  $\Gamma_{\mathcal{L}}$ . Also  $\Gamma_{\mathcal{L}} \in \mathcal{B}(t, B)$  by (5.4.4).  $\mathcal{A}$  is based on  $\mathcal{B}$  hence  $\Gamma_{\mathcal{L}} \in \mathcal{A}(t, B)$ . As a result,  $\mathfrak{M}, \Gamma_{\mathcal{L}} \Vdash t: B$  by (3.3.18).

$t: B \in \overline{\mathcal{F}}_{\mathcal{L}}$ . By Step 9, either

- $\Box B \in \overline{\mathcal{F}}_{\mathcal{L}}$ . In this case,  $\mathfrak{M}, \Gamma_{\mathcal{L}} \not\Vdash \Box B$  by IH, so  $B$  is false in one of the worlds accessible from  $\Gamma_{\mathcal{L}}$ . Hence,  $\mathfrak{M}, \Gamma_{\mathcal{L}} \not\Vdash t: B$  by (3.3.18).
- Or  $*(t, B) \in \mathcal{F}_{\mathcal{L}}^*$ . At the immediately following Step 11 external subroutine  $*!_{\mathcal{L}\mathcal{S}}\text{-DERIVE}\langle \mathcal{T}_{\mathcal{L}}^*, *(t, B) \rangle$  must have been called. Since the  $\mathcal{L}$ -call is successful, that routine must have returned failure, which means that

$$\mathcal{T}_{\mathcal{L}}^* \not\vdash_{*!_{\mathcal{L}\mathcal{S}}} *(t, B) . \quad (5.4.14)$$

Clearly by (5.4.4), for each  $\mathcal{L}'$ -call and corresponding world  $\Gamma_{\mathcal{L}'}$

$$\mathcal{B}_{\Gamma_{\mathcal{L}'}}^* = \mathcal{T}_{\mathcal{L}'}^* .$$

Therefore, the definition of  $\mathcal{A}$  via (3.5.9) can be reformulated:

$$*(s, C) \in \mathcal{A}_{\Gamma_{\mathcal{L}}}^* \iff \bigcup_{\Gamma_{\mathcal{L}'} R \Gamma_{\mathcal{L}}} \mathcal{T}_{\mathcal{L}'}^* \vdash_{*!_{\mathcal{L}\mathcal{S}}} *(s, C) \quad (5.4.15)$$

We will show that

$$\Gamma_{\mathcal{L}'} R \Gamma_{\mathcal{L}} \implies \mathcal{T}_{\mathcal{L}'}^* \subseteq \mathcal{T}_{\mathcal{L}}^* \quad (5.4.16)$$

Since  $\subseteq$  itself is reflexive and transitive, it is sufficient to prove

$$\Gamma_{\mathcal{L}'} R_0 \Gamma_{\mathcal{L}} \quad \Longrightarrow \quad \mathcal{T}_{\mathcal{L}'}^* \subseteq \mathcal{T}_{\mathcal{L}}^* \quad (5.4.17)$$

$\Gamma_{\mathcal{L}'} R_0 \Gamma_{\mathcal{L}}$  may hold because of either (5.4.2) or (5.4.3).

(5.4.2)  $\mathcal{L}' = \mathcal{L}^{-1}$ . Then, by Step 11,  $\mathcal{L}$ -call has been initiated with

$$(\mathcal{T}_{\mathcal{L}'}^*)^\cdot \subseteq \mathcal{T}. \text{ By Step 8, } ((\mathcal{T}_{\mathcal{L}'}^*)^\cdot)^* = \mathcal{T}_{\mathcal{L}'}^* \subseteq \mathcal{T}_{\mathcal{L}}^*.$$

(5.4.3)  $\mathcal{L}' = (\mathcal{T}^\square, \mathcal{T}^*, \langle B_1, \dots, B_k \rangle)$  follows  $\mathcal{L} = (\mathcal{T}^\square, \mathcal{T}^*, \langle B_1 \rangle)$  on a

computation branch, where  $\mathcal{T}_{\mathcal{L}'}^\square = \mathcal{T}^\square$  and  $\mathcal{T}_{\mathcal{L}'}^* = \mathcal{T}^*$ . Note

that the parameter  $\mathcal{T}^*$  is non-decreasing along each branch

and that this parameter for any  $\mathcal{L}'$ -call always coincides with

the second element in  $\mathcal{L}'$ . Therefore,

$$\mathcal{T}^* \subseteq \mathcal{T}_{\mathcal{L}}^* \subseteq \mathcal{T}_{\mathcal{L}'}^* = \mathcal{T}^* .$$

It follows that  $\mathcal{T}_{\mathcal{L}'}^* = \mathcal{T}_{\mathcal{L}}^*$

Using (5.4.17) for a reflexive  $R$ , we can reduce (5.4.15) to

$$*(s, C) \in \mathcal{A}_{\Gamma_{\mathcal{L}}}^* \quad \Longleftrightarrow \quad \mathcal{T}_{\mathcal{L}}^* \vdash_{*!_{\mathcal{L}S}} *(s, C) \quad (5.4.18)$$

Combined with (5.4.14) this yields

$$*(t, B) \notin \mathcal{A}_{\Gamma_{\mathcal{L}}}^*$$

or equivalently

$$\Gamma_{\mathcal{L}} \notin \mathcal{A}(t, B)$$

It immediately follows that  $\mathfrak{M}, \Gamma_{\mathcal{L}} \not\models t : B$  by (3.3.18).

In either case,  $\mathfrak{M}, \Gamma_{\mathcal{L}} \not\models t : B$ .

This completes the proof of the Truth Lemma 5.4.7.  $\square$

**Corollary 5.4.8.** *For the initial call (5.4.1) and its corresponding world  $\Gamma_{(\emptyset, \emptyset, \lambda)}$*

$$\mathfrak{M}, \Gamma_{(\emptyset, \emptyset, \lambda)} \not\models F .$$

*Proof.* According to (5.4.1),  $F \in \overline{\mathcal{F}}_{(\emptyset, \emptyset, \lambda)}$ .  $\square$

This corollary concludes the proof that a formula  $F$  is refutable whenever the algorithm claims it to be.

Let us now show the converse: if a  $F$  is refutable, the algorithm does return **true**.

**Lemma 5.4.9 (Successful Termination Lemma).** *Let  $\mathfrak{M} = \langle W, R, V, \mathcal{A} \rangle$  be a model and  $\Gamma_0 \in W$  be a world in it such that  $\mathfrak{M}, \Gamma_0 \not\models F$ . Let  $\Gamma = \Gamma_0$  be the current world at the initial call (5.4.1) of procedure **S4LP<sub>CS</sub>-WORLD**. We will show that there is a way to move the current world  $\Gamma$  within  $W$  after*

each essential call in such a way that throughout the run initiated by (5.4.1)

$$G \in \mathcal{T} \cup \mathcal{T}^\square \quad \Rightarrow \quad \mathfrak{M}, \Gamma \Vdash A \quad (5.4.19)$$

$$G \in \mathcal{F} \cup \mathcal{F}^\square \quad \Rightarrow \quad \mathfrak{M}, \Gamma \nVdash A \quad (5.4.20)$$

$$*(t, B) \in \mathcal{T}^* \quad \Rightarrow \quad \Gamma \in \mathcal{A}(t, B) \quad (5.4.21)$$

$$*(t, B) \in \mathcal{F}^* \quad \Rightarrow \quad \Gamma \notin \mathcal{A}(t, B) \quad (5.4.22)$$

In this case the algorithm will never return **false**.

*Proof.* Induction on the recursion depth.

*Base.* For the initial call (5.4.1),  $F \in \mathcal{F}$  and  $\mathfrak{M}, \Gamma \nVdash F$ .

*Step 2* will never be applied as  $\perp$  cannot be true; therefore, by IH, it never occurs in  $\mathcal{T}$ .

*Step 3* initiates a new call with no new formulas in either of the six sets mentioned.

*Step 4* may initiate one of two possible calls. The decision is made based on the current world  $\Gamma$ . Since  $B \rightarrow C \in \mathcal{T}$  by IH we have  $\mathfrak{M}, \Gamma \Vdash B \rightarrow C$ .

By (3.3.17) either

- $\mathfrak{M}, \Gamma \nVdash B$ , then invoke the recursive call with  $B$  added to  $\mathcal{F}$ , or
- $\mathfrak{M}, \Gamma \Vdash C$ , then invoke the recursive call with  $C$  added to  $\mathcal{T}$ .

In both cases (5.4.19)–(5.4.22) hold for the new recursive call.

*Step 5* initiates two calls one after another. Since  $B \rightarrow C \in \mathcal{F}$  by IH we have

$\mathcal{M}, \Gamma \not\models B \rightarrow C$ . Thus, both

- $\mathcal{M}, \Gamma \Vdash B$ , so that the recursive call with  $B$  added to  $\mathcal{T}$  satisfies (5.4.19)–(5.4.22); and
- $\mathcal{M}, \Gamma \not\models C$ , so that the recursive call with  $C$  added to  $\mathcal{F}$  satisfies (5.4.19)–(5.4.22).

*Step 6* initiates a new call with  $\Box B$  transferred from  $\mathcal{T}$  to  $\mathcal{T}^\Box$  and  $B$  added to  $\mathcal{T}$ . Condition (5.4.19) for  $\mathcal{T}$  is the same as for  $\mathcal{T}^\Box$ , so the transfer does not affect it. Further, since  $\Box B \in \mathcal{T}$ , by IH we have  $\mathfrak{M}, \Gamma \Vdash \Box B$ . Hence,  $B$  must be true in all the worlds accessible from  $\Gamma$ , including  $\Gamma$  itself, i.e.,  $\mathfrak{M}, \Gamma \Vdash B$ , which takes care of (5.4.19) for  $B$ .

*Step 7* initiates a new call with  $\Box B$  transferred from  $\mathcal{F}$  to  $\mathcal{F}^\Box$ . Condition (5.4.20) for  $\mathcal{F}$  is the same as for  $\mathcal{F}^\Box$ , so the transfer does not affect it.

*Step 8* initiates a new call with  $t : B$  replaced by  $\Box B$  in  $\mathcal{T}$  and  $*(t, B)$  added to  $\mathcal{T}^*$ . By IH, we have  $\mathfrak{M}, \Gamma \Vdash t : B$ , so

- $B$  is true in all the worlds accessible from  $\Gamma$  and
- $\Gamma \in \mathcal{A}(t, B)$ .

The former guaranties that  $\mathfrak{M}, \Gamma \Vdash \Box B$ , which takes care of (5.4.19) for  $\Box B$ . The latter ensures (5.4.21) for  $*(t, B)$ .

*Step 9* may initiate one of two recursive calls. The decision is made based on the current world. By IH,  $\mathfrak{M}, \Gamma \not\Vdash t : B$ , so

- either  $B$  is false in some world accessible from  $\Gamma$ , in which case  $\mathfrak{M}, \Gamma \not\Vdash \Box B$ , then invoke the call with  $t : B$  replaced by  $\Box B$  in  $\mathcal{F}$ , or
- $\Gamma \notin \mathcal{A}(t, B)$ , then invoke the call with  $t : B$  transferred from  $\mathcal{F}$  to  $\mathcal{F}^*$  in the form of  $*(t, B)$ .

In either case (5.4.19)–(5.4.22) are satisfied.

*Step 10* is never applied. By IH every formula from  $\mathcal{T}$  is true at  $\Gamma$  whereas every formula from  $\mathcal{F}$  is false at  $\Gamma$ . No intersection is possible.

*Step 11* calls several  $*!_{CS}$ -DERIVE subroutines with parameters  $\langle \mathcal{T}^*, *(t, B) \rangle$  for  $*(t, B) \in \mathcal{F}^*$ . All of them return **false**.

Indeed, if **true** were returned for some  $*(t, B) \in \mathcal{F}^*$ ,

$$\mathcal{T}^* \vdash_{*!_{CS}} *(t, B) . \quad (5.4.23)$$

Let us define an F-type possible evidence function  $\mathcal{B}$  such that

$$\Delta \in \mathcal{B}(s, C) \quad \iff \quad \Delta = \Gamma \text{ and } *(s, G) \in \mathcal{T}^* .$$

By Theorem 3.5.20, there exists a minimal admissible for  $\text{S4LP}_{\mathcal{CS}}$  evidence function  $\mathcal{E}$  based on  $\mathcal{B}$  such that

$$*(s, C) \in \mathcal{E}_\Gamma^* \quad \iff \quad \bigcup_{\Delta R \Gamma} \mathcal{B}_\Delta^* \vdash_{*!_{\mathcal{CS}}} *(s, C) .$$

Since  $\mathcal{B}_\Delta^* = \emptyset$  for  $\Delta \neq \Gamma$  and  $\mathcal{B}_\Gamma^* = \mathcal{T}^*$ , we can simplify this equivalence to

$$*(s, C) \in \mathcal{E}_\Gamma^* \quad \iff \quad \mathcal{T}^* \vdash_{*!_{\mathcal{CS}}} *(s, C) .$$

Using (5.4.23), we would get  $*(t, B) \in \mathcal{E}_\Gamma^*$ , i.e.,  $\Gamma \in \mathcal{E}(t, B)$ .

It remains to note that  $\mathcal{A}$  is also an admissible for  $\text{S4LP}_{\mathcal{CS}}$  evidence function that is based on  $\mathcal{B}$  by IH, namely by (5.4.21). Therefore,  $\mathcal{A}$  is based on  $\mathcal{E}$ . Then, we would have  $\Gamma \in \mathcal{A}(t, B)$ , which contradicts the IH, namely (5.4.22) for  $*(t, B)$ . This contradiction shows that all calls of  $*!_{\mathcal{CS}}\text{-DERIVE}$  return **false**.

Finally, in this step several essential recursive calls are made. These calls are independent of each other; each of them prompts us to move the current world  $\Gamma$  to a new position  $\Gamma'$  in  $W$ . Consider one of these

new calls

$$\text{S4LP}_{CS\text{-WORLD}}\langle \mathcal{T}^\square \cup (\mathcal{T}^*)^\cdot, \{B\}, \mathcal{T}^\square, \emptyset, \mathcal{T}^*, \emptyset, \mathcal{L} \odot (\mathcal{T}^\square, \mathcal{T}^\cdot, B) \rangle ,$$

where  $\Box B \in \mathcal{F}^\square$ . By IH, we have  $\mathfrak{M}, \Gamma \not\models \Box B$ , so there must exist a world  $\Gamma'$  accessible from  $\Gamma$  such that  $\mathfrak{M}, \Gamma' \not\models B$ . In that case we will move the current world from  $\Gamma$  to  $\Gamma'$ . Condition (5.4.20) for  $B$  is satisfied in  $\Gamma'$ .

$\mathfrak{M}, \Gamma \Vdash \Box C$  for each  $\Box C \in \mathcal{T}^\square$ . Axiom 4 is valid,  $\Vdash \Box C \rightarrow \Box \Box C$ , hence  $\mathfrak{M}, \Gamma \Vdash \Box \Box C$  for each  $\Box C \in \mathcal{T}^\square$ . Therefore,  $\mathfrak{M}, \Gamma' \Vdash \Box C$  for each  $\Box C \in \mathcal{T}^\square$  and (5.4.19) holds for  $\mathcal{T}^\square$  in  $\Gamma'$ .

For each  $*(s, C) \in \mathcal{T}^*$  by IH we have  $\Gamma \in \mathcal{A}(s, C)$ . Then,  $\Gamma' \in \mathcal{A}(s, C)$  by Monotonicity of  $\mathcal{A}$ . Therefore, (5.4.21) holds for  $\mathcal{T}^*$  in  $\Gamma'$ .

It remains to show that  $\mathfrak{M}, \Gamma' \Vdash s : C$  for each  $s : C \in (\mathcal{T}^*)^\cdot$ , i.e., for each  $*(s, C) \in \mathcal{T}^*$ . We already showed that  $\Gamma' \in \mathcal{A}(s, C)$  for each  $*(s, C) \in \mathcal{T}^*$ .

It is, therefore, sufficient to show that  $\mathfrak{M}, \Delta \Vdash C$  in all worlds  $\Delta$  accessible from  $\Gamma'$ , i.e. to show that  $\mathfrak{M}, \Gamma' \Vdash \Box C$ , for each  $*(s, C) \in \mathcal{T}^*$ . To that end we will show that  $\Box C \in \mathcal{T}^\square$ . Indeed, consider earliest moment (in all the preceding recursive calls) when  $*(s, C)$  appeared in  $\mathcal{T}^*$ . A careful observation shows that it could only happen in Step 8, whereby

$\square C$  must have been added to  $\mathcal{T}$  and shortly thereafter transferred to  $\mathcal{T}^\square$  in Step 6. We already observed that parameter  $\mathcal{T}^\square$  never loses formulas.

We have verified all the conditions for the new essential call.

This completes the proof of Lemma 5.4.9.  $\square$

This completes the proof of Correctness Lemma 5.4.6.  $\square$

*Note 5.4.10.* Normally, correctness of a recursive algorithm is proven by induction on the recursion depth.

Unfortunately, this method cannot be applied to the procedure  $\text{S4LP}_{\mathcal{CS}}$ -WORLD. Such an induction proof is based on the assumption that consecutive recursive calls are completely independent of the calls preceding them. For instance, we should be able to conclude that a world satisfying the intuitive conditions exists for any terminal call, i.e., for any call that does not spawn further calls of  $\text{S4LP}_{\mathcal{CS}}$ -WORLD, provided of course that this terminal call returns **true**.

This is not the case for  $\text{S4LP}_{\mathcal{CS}}$ -WORLD (or, for that matter, for Ladner's algorithm  $\text{S4}$ -WORLD from [Lad77]). As we saw, the important recursive calls are all made in Step 11. In terms of the F-model being constructed, they make us jump from the current world  $w$  to a world  $w'$  accessible from it

that is to become the new current world; all recursive calls in the other steps only refine the conditions within the confines of the then-current world.

These jumps from  $w$  to a new  $w'$  are prompted by the necessity to refute in  $w$  some negative boxed formula from  $\mathcal{F}^\square$ . At the same time all the positive boxed formulas from  $\mathcal{T}^\square$  are transferred to  $w'$ . As a result, if a negative boxed formula happens to hide inside a positive one, we are facing a possible *perpetuum mobile* with this negative formula always popping out of the positive one to prompt another jump.

This potential loop is the sole reason why Ladner introduced the log-argument  $\mathcal{L}$ , which have been adapted to our needs in  $\text{S4LP}_{\text{CS}}\text{-WORLD}$ . The condition  $(\mathcal{T}^\square, \mathcal{T}^*, B) \notin \mathcal{L}$  in the second big conjunction of Step 11 (see Fig. 5.4.1) guarantees that future recursive calls do not duplicate any recursive calls preceding them. This effectively prevents the procedure from looping, but at the same time creates rather complicated dependencies among recursive calls.

Indeed, a current call of the procedure may rely on the results of some preceding calls that *have not terminated yet*, of which the current call is but a part. Such a preceding call, apart from the computation branch that led to the current call, may need to explore other branches of the computation tree before this preceding call is terminated. It may well happen that the

preceding call will return **false** after all, based on information from other computation branches. Therefore, any conclusion based on the current call only is premature; there is not enough information. ◀

**Lemma 5.4.11.** *Procedure S4LP<sub>CS</sub>-WORLD is in PSPACE for any decidable schematic CS.*

*Proof.* First of all, the depth of recursion is at most polynomial in the size of the given formula  $F$ . Indeed, between any two essential calls, the procedure builds one branch of a propositional table with some extra steps for modality and justification formulas. Still each step of this tableau procedure decreases the sum of sizes of formulas in  $\mathcal{T} \cup \mathcal{F}$ . Only subformulas of  $F$  and boxed subformulas of  $F$  (because of Steps 8 and 9) can appear in these sets, thus the maximal possible size of  $\mathcal{T} \cup \mathcal{F}$  is polynomial (in fact, linear) in  $|F|$  throughout the run, making the number of consecutive non-essential calls polynomial along each branch.

The number of essential calls along each branch is also polynomial because sets  $\mathcal{T}^\square$  and  $\mathcal{T}^*$  are only gaining new formulas. The size of the third argument in  $\mathcal{L}$  is at most linear in  $|F|$  because formulas there do not repeat. Again we have at most linear number of increments to  $\mathcal{T}^\square$  and/or  $\mathcal{T}^*$  and at most linear number of essential calls with the same pair of  $\mathcal{T}^\square$  and  $\mathcal{T}^*$  in between

any two consecutive increments.

It is rather obvious that storing information necessary for each call only requires polynomial space. But we also have to keep certain information about prior calls to be able to backtrack. This means that we need to store the stack of all configurations preceding the current call. We already proved that the number of such configurations along each branch is polynomial. It remains to note that each configuration can also be stored in a polynomial space in  $|F|$ . As was noted earlier, the size of  $\mathcal{T}$  and  $\mathcal{F}$  is linear in  $|F|$ . The same obviously is true about  $\mathcal{T}^\square$ ,  $\mathcal{F}^\square$ ,  $\mathcal{T}^*$ , and  $\mathcal{F}^*$ . Each formula in these six sets can be stored by placing a marker on a subformula of  $F$ . The amount of markers needed is clearly finite: the markers will stipulate which set the subformula belongs to, and whether it is the subformula itself or its boxed version. Storing  $\mathcal{T}^\square$  and  $\mathcal{T}^*$  within  $\mathcal{L}$  can be done in a similar way, as well as storing the list of formulas in  $\mathcal{L}$ .

It remains to note that the external subroutine  $*!_{\mathcal{L}\mathcal{S}}\text{-DERIVE}$  in Step 11 is an NP-algorithm by Lemma 5.1.3.2, which can clearly be carried out on a polynomial space. This subroutine is not recursive, although it is called multiple times. Its input has size polynomial in  $|F|$ .

This completes the proof of Lemma 5.4.11. □

This completes the proof of Theorem 5.4.4.  $\square$

## 5.5 Historical Survey

NP-completeness of  $\text{LP}_{\mathcal{CS}}$  with a finite  $\mathcal{CS}$  easily follows from the results of Sergei Artemov in [Art98]. Robert Milnikel in [Mil07] noted that NP-completeness of  $\text{LP}_{\mathcal{CS}}$  with a decidable injective  $\mathcal{CS}$  also follows easily.

The upper complexity bound of  $\Pi_2^p$  (in the polynomial hierarchy) for  $\text{J}_{\mathcal{CS}}$ ,  $\text{JT}_{\mathcal{CS}}$ ,  $\text{J4}_{\mathcal{CS}}$ , and  $\text{LP}_{\mathcal{CS}}$  with a decidable schematic  $\mathcal{CS}$  was demonstrated in [Kuz00]. Since  $\mathcal{TCS}$  is schematic and decidable for these four logics,  $\text{J}$ ,  $\text{JT}$ ,  $\text{J4}$ , and  $\text{LP}$  themselves are in  $\Pi_2^p$ .

The same upper bound was also claimed in that paper for  $\text{JD}_{\mathcal{CS}}$  and  $\text{JD4}_{\mathcal{CS}}$ , but an omission was found in the proof during the work on this thesis. A finitary M-model refuting the given formula  $F$  was constructed in the proof. But the Consistent Evidence Condition cannot be so easily checked for the minimal evidence function constructed in the proof. The difficulty is that the Consistent Evidence condition has to be checked for all terms  $t$ , not only for subterms of  $F$ . Theorem 5.2.4 restores the result of [Kuz00] for  $\text{JD}_{\mathcal{CS}}$  with decidable, schematic, and axiomatically appropriate  $\mathcal{CS}$ . The complexity of  $\text{JD4}_{\mathcal{CS}}$  remains to be found.

Nikolai Krupski in [Kru03] showed that  $\text{rLP}$  is in NP.

PSPACE-completeness of  $\mathbf{S4}_1\mathbf{LP}$  was shown in [Kuz06a].

Milnikel in [Mil07] showed that  $\mathbf{J4}_{\mathcal{CS}}$  is  $\Pi_2^p$ -hard for any decidable, axiomatically appropriate, and schematic  $\mathcal{CS}$ . As a corollary, any such  $\mathbf{J4}_{\mathcal{CS}}$ , including  $\mathbf{J4}$  itself, is  $\Pi_2^p$ -complete.

Milnikel also showed that  $\mathbf{LP}_{\mathcal{CS}}$  is  $\Pi_2^p$ -hard for any decidable, axiomatically appropriate, and schematically injective  $\mathcal{CS}$ . As a corollary, any such  $\mathbf{LP}_{\mathcal{CS}}$  is  $\Pi_2^p$ -complete. This does not yield  $\Pi_2^p$ -completeness of  $\mathbf{LP}$  though because  $\mathcal{TCS}_{\mathbf{LP}}$  is not schematically injective.

# Chapter 6

## Self-Referentiality

In this chapter we will explore an application of justification logics to the question of self-referentiality.

The modality in GL corresponds to provability in the formal arithmetic. A whole textbook [Smo85] is devoted to the studies of self-reference of GL-modality through the arithmetical methods or methods inherited from Peano arithmetic.

Below we provide a similar analysis for epistemic logic by means of justifications. As in the case of GL even the definition of self-referentiality<sup>1</sup> is given through the justification language.

---

<sup>1</sup>We give it a slightly different name from the one used by Smoryński because our definition of ‘self-referentiality’ is indeed different from his ‘self-reference.’

## 6.1 When Is Knowledge Self-Referential?

Pure justification logics  $\text{JL}_{\mathcal{CS}}$  clearly exhibit self-referentiality when a term  $t$  proves something about itself:

$$\text{JL}_{\mathcal{CS}} \vdash t:F(t) .$$

Such constructions are, of course, perfectly legal in the pure justification language. In fact, there are many theorems of this type for any non-empty schematic  $\mathcal{CS}$  with  $t = c$  being a justification constant and  $F(c)$  being an axiom instance:

$$\text{JL}_{\mathcal{CS}} \vdash c:A(c) .$$

A natural question to ask is whether the use of such self-referential constants is necessary for the Realization Theorem 3.2.20 to hold. Apart from being *direct* as in  $\vdash c:A(c)$ , self-referentiality may also occur as a result of a cycle of references:

$$\vdash c_2:A_1(c_1), \quad \dots \quad , \quad \vdash c_n:A_{n-1}(c_{n-1}), \quad \vdash c_1:A_n(c_n) .$$

If direct self-referentiality is expendable, we should ask whether such self-referential cycles are still required for the Realization.

**Definition 6.1.1.** A constant specification  $\mathcal{CS}$  is called *directly self-referential* if  $c : A(c) \in \mathcal{CS}$  for some axiom  $A$  that contains at least one

occurrence of  $c$ .

A constant specification  $\mathcal{CS}$  is called *self-referential* if

$$\{c_2 : A_1(c_1), \dots, c_n : A_{n-1}(c_{n-1}), c_1 : A_n(c_n)\} \subseteq \mathcal{CS}$$

for some axioms  $A_i(c_i)$ ,  $i = 1, \dots, n$ , where each  $A_i$  contains at least one occurrence of  $c_i$ . ◀

**Definition 6.1.2.** Let  $\text{JL}$  be a justification counterpart of a modal logic  $\text{ML}$ , i.e.,  $\text{JL}^\circ = \text{ML}$ .

We will call knowledge/belief described by pair  $\text{ML}/\text{JL}$  *directly self-referential* if  $\text{JL}_{\mathcal{CS}}^\circ = \text{ML}$  implies that  $\mathcal{CS}$  is directly self-referential.

We will call knowledge/belief described by pair  $\text{ML}/\text{JL}$  *self-referential* if  $\text{JL}_{\mathcal{CS}}^\circ = \text{ML}$  implies that  $\mathcal{CS}$  is self-referential. ◀

## 6.2 Self-Referential Knowledge

It was shown by Roman Kuznets that the realization of **S4** in **LP** does require directly self-referential constants (see [BK05, Kuz06c, BK06]). In [Kuz08] this result was extended to **K4**, **D4**, and **T**.

For each modal logic  $\text{ML}$  from this list we will present a modal formula  $\Phi$  derivable in the logic,  $\text{M} \vdash \Phi$ . Let  $\text{JL}$  be a justification counterpart for  $\text{ML}$  from Theorem 3.2.20. We will consider the constant specification  $\mathcal{CS}$  for

JL that is the largest constant specification without directly self-referential constants. We will then show that any potential realization of  $\Phi$  in the pure justification language is not  $\text{JL}_{\mathcal{CS}}$ -valid by constructing an F-type counter-model for any such realization.

We will use  $\Phi = \diamond(p \rightarrow \Box p)$ , or equivalently

$$\Phi = \neg\Box\neg(p \rightarrow \Box p) , \quad (6.2.1)$$

for modal logics S4, D4, and T. For K4 we will use  $\Psi = \diamond T \rightarrow \diamond(p \rightarrow \Box p)$ , or equivalently,

$$\Psi = \Box\neg(p \rightarrow \Box p) \rightarrow \Box\perp \quad (6.2.2)$$

instead.

The suggestion to use (6.2.1) for S4 came from an anonymous referee when a preliminary version of this result was rejected from a conference. Melvin Fitting then suggested that the same formula (6.2.1) can also be used for D4 and T. He also suggested (6.2.2) as a transformation of (6.2.1) derivable in K4.

**Theorem 6.2.1** ([Kuz08]). *S4/LP, D4/JD4, and T/JT describe directly self-referential knowledge.*

*Proof.* First of all, we need to show that (6.2.1) is derivable in S4, D4, and T.

Figure 6.2.1: Tableau derivation of  $\diamond(p \rightarrow \Box p)$  in  $\mathsf{T}$  and  $\mathsf{S4}$ 

1.	1	$\neg$	$\diamond(p \rightarrow \Box p)$	
2.	1	$\neg$	$(p \rightarrow \Box p)$	by $\mathsf{T}$ -rule from 1.
3.	1		$p$	from 2.
4.	1	$\neg$	$\Box p$	from 2.
5.	1.1	$\neg$	$p$	by $\mathsf{K}$ -rule from 4.
6.	1.1	$\neg$	$(p \rightarrow \Box p)$	by $\mathsf{K}$ -rule from 1.
7.	1.1		$p$	from 6.
8.	1.1	$\neg$	$\Box p$	from 6.
			$\otimes$	

Prefix 1.1 in Line 5 is new. Prefix 1.1 in Line 6 has already occurred on Line 5. The branch is closed by Lines 5 and 7.

The tableau derivation of  $\Phi$  in  $\mathsf{T}$  or  $\mathsf{S4}$  can be found in Fig. 6.2.1. The tableau derivation for  $\mathsf{D4}$  is in Fig. 6.2.2.

Let  $\mathsf{ML} \in \{\mathsf{S4}, \mathsf{D4}, \mathsf{T}\}$ ; let  $\mathsf{JL} \in \{\mathsf{LP}, \mathsf{JD4}, \mathsf{JT}\}$  be its justification counterpart, and let  $\mathcal{CS}$  be the largest constant specification for  $\mathsf{JL}$  without directly self-referential constants. We will show that for any justification formula  $F$  such that  $F^\circ = \Phi$ , there exists an  $F$ -model  $\mathfrak{M} = (W, R, V, \mathcal{A})$  and a world  $w \in W$  such that  $\mathfrak{M}, w \not\models F$ . Therefore, by the Completeness Theorem 3.3.14,<sup>2</sup> no such  $F$  is derivable in  $\mathsf{JL}_{\mathcal{CS}}$ . Thus,  $(\mathsf{JL}_{\mathcal{CS}})^\circ \neq \mathsf{ML}$  and realization is impossible within this  $\mathcal{CS}$ .

For any pair of terms  $t$  and  $t'$  used in place of the two  $\Box$ 's in  $\Phi$ , we will

---

<sup>2</sup>Note that we only use soundness of justification logics w.r.t.  $F$ -models, which holds without any extra assumptions on  $\mathcal{CS}$  for  $\mathsf{JD4}$ . On the other hand, the  $\mathcal{CS}$  we consider is axiomatically appropriate since for any axiom instance  $A$  there exists a constant not occurring in  $A$  so that  $c:A \in \mathcal{CS}$ .

Figure 6.2.2: Tableau derivation of  $\diamond(p \rightarrow \Box p)$  in D4

1.	1	$\neg$	$\diamond(p \rightarrow \Box p)$	
2.	1	$\neg$	$\Box(p \rightarrow \Box p)$	by D-rule from 1.
3.	1.1	$\neg$	$(p \rightarrow \Box p)$	by K-rule from 2.
4.	1.1		$p$	from 3.
5.	1.1	$\neg$	$\Box p$	from 3.
6.	1.1.1	$\neg$	$p$	by K-rule from 5.
7.	1.1	$\neg$	$\diamond(p \rightarrow \Box p)$	by K4-rule from 1.
8.	1.1.1	$\neg$	$(p \rightarrow \Box p)$	by K-rule from 7.
9.	1.1.1		$p$	from 8.
10.	1.1.1	$\neg$	$\Box p$	from 8.
			$\otimes$	

Prefix 1.1 in Line 3 is new. Prefix 1.1.1 in Line 6 is new. Prefix 1.1 in Line 7 has already occurred on Line 3. Prefix 1.1.1 in Line 8 has already occurred on Line 6. The branch is closed by Lines 6 and 9.

construct an F-model for  $\mathbf{JL}_{CS}$  that falsifies  $\neg t : [\neg(p \rightarrow t' : p)]$ , thus showing that no realization of  $\Phi$  is  $\mathbf{JL}_{CS}$ -valid.

Given  $t$  and  $t'$ , consider the following F-model for  $\mathbf{JL}_{CS}$ :  $\mathfrak{M} = (W, R, V, \mathcal{A})$

with

- $W = \{w\}$
- $R = \{(w, w)\}$
- $v(q) = W = \{w\}$  for any sentence letter  $q$
- $\mathcal{B}(s, G) = \begin{cases} W & \text{if } s = t \text{ and } G = \neg(p \rightarrow t' : p) \\ \emptyset & \text{otherwise} \end{cases}$

- $\mathcal{A}$  is the minimal F-type admissible for  $\mathbf{JL}_{CS}$  evidence function based on  $\mathcal{B}$

Such  $R$  is obviously serial, reflexive, and transitive, thus making it suitable for LP, JD4, and JT.

Since  $w$  is the only world in the model, we will write

$$\begin{aligned} \Vdash F & \quad \text{instead of} \quad \mathcal{M}, w \Vdash F \\ \mathcal{A}(s, F) & \quad \text{instead of} \quad w \in \mathcal{A}(s, F) \\ \neg\mathcal{A}(s, F) & \quad \text{instead of} \quad w \notin \mathcal{A}(s, F) \end{aligned}$$

The admissible evidence function  $\mathcal{A}$  exists by Cor. 3.3.42.1. Note that  $\mathcal{A}$  depends on terms  $t$  and  $t'$ . In particular,  $\mathcal{A}(t, \neg(p \rightarrow t' : p))$  because  $\mathcal{A}$  is based on  $\mathcal{B}$ .

It suffices to show  $\neg\mathcal{A}(t', p)$  to falsify  $\neg t : [\neg(p \rightarrow t' : p)]$ . Indeed,  $\not\vdash t' : p$  if  $\neg\mathcal{A}(t', p)$ . Given  $\Vdash p$ , it yields  $\Vdash \neg(p \rightarrow t' : p)$ . Finally, with this formula true at the only world and with  $\mathcal{A}(t, \neg(p \rightarrow t' : p))$ , we will have

$$\Vdash t : [\neg(p \rightarrow t' : p)] .$$

$\neg\mathcal{A}(t', P)$  follows from the following technical lemma. Let  $\mathcal{A}_0$  be the minimal F-type admissible for  $\mathbf{JL}_{CS}$  evidence function based on  $\mathcal{B}_0(s, G) \equiv \emptyset$  for all terms  $s$  and all formulas  $G$ . Again,  $\mathcal{A}_0$  exists by Cor. 3.3.42.1. Since  $\mathcal{A}$

is (vacuously) based on  $\mathcal{B}_0$  too,  $\mathcal{A}_0 \subseteq \mathcal{A}$ . According to Cor. 3.3.42.1, evidence functions  $\mathcal{A}$  and  $\mathcal{A}_0$  are defined by

- (3.3.37) via  $*_{\mathcal{CS}}$ -calculus for  $\text{JT}_{\mathcal{CS}}$ ;
- (3.3.38) via  $*!_{\mathcal{CS}}$ -calculus for  $\text{JD4}_{\mathcal{CS}}$  or  $\text{LP}_{\mathcal{CS}}$ .

In other words, for the respective  $*$ -calculus,

$$\mathcal{A}_0(s', G) \iff \vdash_* *(s', G) \quad (6.2.3)$$

$$\mathcal{A}(s', G) \iff *(t, \neg(p \rightarrow t':p)) \vdash_* *(s', G) \quad (6.2.4)$$

**Lemma 6.2.2.** *For any subterm  $s$  of term  $t'$ :*

1. If  $\vdash_* *(s, F)$ ,

*then  $\text{JL}_{\mathcal{CS}} \vdash F$  and  $F$  does not contain occurrences of  $t'$ .*

2. If

$$*(t, \neg(p \rightarrow t':p)) \vdash_* *(s, F)$$

$$\not\vdash_* *(s, F)$$

*then  $F$  has at least one occurrence of  $t'$ . Moreover, if  $F$  is an implication,  $F = \neg(p \rightarrow t':p)$ .<sup>3</sup>*

---

<sup>3</sup>Remember that we consider  $\neg G$  to be an abbreviation of  $G \rightarrow \perp$ .

*Proof.* The proof is by induction on the size of  $s$ .

Essentially, we show that all applications of \*A2 in the \*-derivation happen in the derivation without hypotheses, so that any \*-derivation branch starting with the hypothesis  $*(t, \neg(p \rightarrow t':p))$  is, in a sense, “cut-free.”

$s = x$  is a justification variable.

1.  $\not\vdash_*(x, F)$  for any  $F$ .

Thus, Clause 1 is vacuously true.

2.  $*(t, \neg(p \rightarrow t':p)) \vdash_*(x, F)$

only if  $t = x$  and  $F = \neg(p \rightarrow t':p)$ . The latter does contain  $t'$  and is the only allowed implication.

$s = c$  is a justification constant.

1. If  $\vdash_*(c, F)$ ,

it was derived by  $*\mathcal{CS}$  or  $*\mathcal{CS}^!$ , so  $c : F \in \mathcal{CS}$  and  $F$  must be an axiom of  $\mathbf{JL}_{\mathcal{CS}}$ . Any axiom is derivable in its logic. At the same time,  $\mathcal{CS}$  is not directly self-referential, so  $F$  cannot contain occurrences of  $c$ , a subterm of  $t'$ . Thus,  $F$  cannot contain  $t'$  either.

2. It can only happen that

$$*(t, \neg(p \rightarrow t' : p)) \vdash_* *(c, F)$$

$$\not\vdash_* *(c, F)$$

if  $t = c$  and  $F = \neg(p \rightarrow t' : p)$ . The latter does contain  $t'$  and is the only allowed implication.

$s = s_1 + s_2$  1. If  $\vdash_* *(s_1 + s_2, F)$ ,

it was derived by rule \*A3, so  $\vdash_* *(s_i, F)$  for some  $i = 1, 2$ . By IH,  $F$  is a theorem that does not contain  $t'$ .

2. Only in two cases can it happen that

$$*(t, \neg(p \rightarrow t' : p)) \vdash_* *(s_1 + s_2, F)$$

$$\not\vdash_* *(s_1 + s_2, F)$$

(a)  $t = s_1 + s_2$  and  $F = \neg(p \rightarrow t' : p)$ , the latter satisfies Clause 2,

or

(b) \*A3 was used in the derivation from the hypothesis, so that

$$*(t, \neg(p \rightarrow t' : p)) \vdash_* *(s_i, F)$$

$$\not\vdash_* *(s_i, F)$$

for some  $i = 1, 2$ . By IH,  $F$  contains  $t'$ , and, if an implication, is  $\neg(p \rightarrow t' : p)$ .

$s = s_1 \cdot s_2$  1. If  $\vdash_* *(s_1 \cdot s_2, F)$ ,

it was derived by \*A2, so there must exist a formula  $G$  such that  $\vdash_* *(s_1, G \rightarrow F)$  and  $\vdash_* *(s_2, G)$ . By IH, both  $G \rightarrow F$  and  $G$  are derivable; hence,  $F$  is derivable by *modus ponens*. By IH,  $G \rightarrow F$  does not contain  $t'$ , thus neither does  $F$ .

2. Only in three cases can it happen that

$$*(t, \neg(p \rightarrow t':p)) \vdash_* *(s_1 \cdot s_2, F)$$

$$\not\vdash_* *(s_1 \cdot s_2, F)$$

(a)  $t = s_1 \cdot s_2$  and  $F = \neg(p \rightarrow t':p)$ , the latter satisfies Clause 2.

(b) Rule \*A2 was used and there exists a  $G$  such that

$$*(t, \neg(p \rightarrow t':p)) \vdash_* *(s_1, G \rightarrow F)$$

$$*(t, \neg(p \rightarrow t':p)) \vdash_* *(s_2, G)$$

$$\not\vdash_* *(s_1, G \rightarrow F)$$

We will show that these three statements are, in fact, inconsistent. By IH, Clause 2 for subterm  $s_1$ ,

$$G \rightarrow F = \neg(p \rightarrow t':p) = (p \rightarrow t':p) \rightarrow \perp .$$

So  $G = p \rightarrow t' : p$ , which is an implication different from the only allowed in Clause 2. Hence, by IH, Clause 2 for  $s_2$ ,

we would have  $\vdash_* *(s_2, G)$ , which would contradict the IH, Clause 1 for  $s_2$  since  $p \rightarrow t':p$  contains  $t'$ . This contradiction shows the impossibility of Case 2b.

(c) Rule \*A2 was used and there exists a  $G$  such that

$$\begin{aligned} *(t, \neg(p \rightarrow t':p)) &\vdash_* *(s_1, G \rightarrow F) \\ *(t, \neg(p \rightarrow t':p)) &\vdash_* *(s_2, G) \\ &\not\vdash_* *(s_2, G) \end{aligned}$$

We will show that these three statements are also inconsistent. By IH, Clause 2 for  $s_2$ , formula  $G$  should contain  $t'$ . Then,  $G \rightarrow F$  would also contain  $t'$ . Hence, by IH, Clause 1 for  $s_1$ , we should have  $\not\vdash_* *(s_1, G \rightarrow F)$ , the impossibility of which was shown in Case 2b. So Case 2c is also impossible.

$s = !s_1$  (for  $*_{\mathcal{CS}}$ -calculus used for  $\text{JT}_{\mathcal{CS}}$ ).

1. If  $\vdash_{*_{\mathcal{CS}}} *(!s_1, F)$ ,

it was derived by  $*_{\mathcal{CS}}^!$ , so  $s_1 = \underbrace{!\dots!}_n c$  for some constant  $c$  and integer  $n \geq 0$ , and  $F$  must be of form  $\underbrace{!\dots!}_n c:\dots!:c:c:A$  for some axiom  $A$  such that  $c:A \in \mathcal{CS}$ . By rule  $\text{RA}_{\mathcal{CS}}^!$ ,

$$\text{JT}_{\mathcal{CS}} \vdash \underbrace{!\dots!}_n c:\dots!:c:c:A .$$

Axiom  $A$  cannot contain  $c$  since  $\mathcal{CS}$  is not directly self-referential.

Constant  $c$  is a subterm of  $s = \underbrace{!\dots!}_{n+1}c$ , which in turn is a subterm of  $t'$ ; therefore,  $A$  cannot contain  $t'$ . Since  $c, !c, \dots, \underbrace{!\dots!}_n c$  are proper subterms of  $s$ , itself a subterm of  $t'$ , these ground terms cannot contain  $t'$  either. Summarizing,  $F$  does not contain  $t'$ .

2. The only case when

$$\begin{aligned} *(t, \neg(p \rightarrow t':p)) \vdash_{*\mathcal{CS}} *(!s_1, F) \\ \not\vdash_{*\mathcal{CS}} *(!s_1, F) \end{aligned}$$

is when  $t = !s_1$  and  $F = \neg(p \rightarrow t':p)$ , the latter satisfies Clause 2.

$s = !s_1$  (for  $*!\mathcal{CS}$ -calculus used for  $\text{JD4}_{\mathcal{CS}}$  and  $\text{LP}_{\mathcal{CS}}$ ).

1. If  $\vdash_{*!\mathcal{CS}} *(!s_1, F)$ ,

it was derived by  $*A5$ , so  $F = s_1 : G$  for some formula  $G$  such that  $\vdash_{*!\mathcal{CS}} *(s_1, G)$ . By IH, Clause 1 for  $s_1$ ,  $G$  is a theorem that does not contain  $t'$ .

For any F-model  $\mathfrak{M}' = (W', R', V', \mathcal{E}')$  for  $\text{JL}_{\mathcal{CS}} \in \{\text{JD4}_{\mathcal{CS}}, \text{LP}_{\mathcal{CS}}\}$  the admissible evidence function  $\mathcal{E}'$  is, of course, based on the empty F-type possible evidence function  $\mathcal{B}_0(s, G) \equiv \emptyset$ . Such models, in particular such admissible evidence functions  $\mathcal{E}'$ , exist by

consistency of  $\text{JD}_{\mathcal{CS}}$  and  $\text{LP}_{\mathcal{CS}}$  respectively. Therefore,  $\mathcal{E}' \supseteq \mathcal{A}'$ , where  $\mathcal{A}'$  is the minimal F-type admissible for  $\text{JL}_{\mathcal{CS}}$  evidence function on  $W'$  based on  $\mathcal{B}_0$ . According to Theorem 3.3.41.4,  $\mathcal{A}'$  is described by (3.3.38) for any  $w' \in W'$

$$\mathcal{A}'_{w'}(s', H) \iff \vdash_{*\!_{\mathcal{CS}}} *(s', H)$$

Therefore, for any such  $\mathcal{A}'$ , it must be that  $\mathcal{A}'(s_1, G) = W'$  and hence  $\mathcal{E}'(s_1, G) = W'$  for any F-type admissible for  $\text{JL}_{\mathcal{CS}}$  evidence function  $\mathcal{E}'$ .

Combining  $\mathcal{E}'(s_1, G) = W'$  with validity of  $G$ , we get validity of  $s_1 : G$  from Completeness Theorem 3.3.14 (for  $\text{JD4}_{\mathcal{CS}}$  we use the fact that our  $\mathcal{CS}$  is axiomatically appropriate).

Since  $G$  does not contain  $t'$  and  $s_1$  is a proper subterm of  $t'$ , formula  $s_1 : G$  cannot contain  $t'$  either.

2. Here are all the situations where it could happen that

$$*(t, \neg(p \rightarrow t' : p)) \vdash_{*\!_{\mathcal{CS}}} *(!s_1, F)$$

$$\not\vdash_{*\!_{\mathcal{CS}}} *(!s_1, F)$$

(a)  $t = !s_1$  and  $F = \neg(p \rightarrow t' : p)$ , the latter satisfies Clause 2, or

else

(b) Rule \*A5 was used, so that  $F = s_1 : G$  for some  $G$  such that

$$*(t, \neg(p \rightarrow t' : p)) \vdash_{*!_{CS}} *(s_1, G)$$

$$\not\vdash_{*!_{CS}} *(s_1, G)$$

By IH, Clause 2,  $G$  contains  $t'$ , thus so does  $s_1 : G$ , which is not an implication.

This completes the proof of Lemma 6.2.2 □

It remains to apply Lemma 6.2.2 to term  $t'$  itself.

$\mathbb{J}L_{CS} \not\vdash p$ , so by Lemma 6.2.2.1,  $\not\vdash_* *(t', p)$ .

But then, since  $t'$  does not occur in  $p$ , by Lemma 6.2.2.2,

$$*(t, \neg(p \rightarrow t' : p)) \not\vdash_* *(t', p) .$$

Thus, by (6.2.4)  $\neg\mathcal{A}(t', p)$ . As was noted earlier, this suffices for

$$\mathfrak{M} \not\vdash \neg t : [\neg(p \rightarrow t' : p)]$$

and hence

$$\mathbb{J}L_{CS} \not\vdash \neg t : [\neg(p \rightarrow t' : p)]$$

This completes the proof of Theorem 6.2.1. □

**Theorem 6.2.3** ([Kuz08]). *Knowledge described by K4/J4 is directly self-referential.*

*Proof.* The Hilbert formulation of **D4** is obtained from that of **K4** by adding the Seriality Axiom. Note that the Seriality Axiom is indeed a single axiom rather than an axiom scheme. Therefore,  $\mathbf{K4} \vdash \Diamond T \rightarrow \Diamond(p \rightarrow \Box p)$ , or equivalently, its contrapositive

$$\Psi = \Box \neg(p \rightarrow \Box p) \rightarrow \Box \perp$$

is derivable in **K4**.

**J4** is a justification counterpart for **K4**. Let  $\mathcal{CS}$  be the largest constant specification for **J4** that is not directly self-referential.

We will show that for any justification formula

$$F = t : [\neg(p \rightarrow t' : p)] \rightarrow k : \perp ,$$

such that  $F^\circ = \Psi$  there exists an F-model for  $\mathbf{J4}_{\mathcal{CS}}$  that falsifies  $F$ , thus showing that no realization of  $\Psi$  is  $\mathbf{J4}_{\mathcal{CS}}$ -valid.

Unlike in the proof of Theorem 6.2.1, the falsifying model here consists of a single irreflexive world. Given  $t$  and  $t'$ , we consider  $\mathfrak{M} = (W, R, V, \mathcal{A})$  with the

- $W = \{w\}$
- $R = \emptyset$
- $v(q) = W = \{w\}$  for any sentence letter  $q$

- $\mathcal{B}(s, G) = \begin{cases} W & \text{if } s = t \text{ and } G = \neg(p \rightarrow t' : p) \\ \emptyset & \text{otherwise} \end{cases}$
- $\mathcal{A}$  is the minimal F-type admissible for  $\mathbf{J4}_{\mathcal{CS}}$  evidence function based on  $\mathcal{B}$

Such  $R$  is vacuously transitive, thus making it suitable for  $\mathbf{J4}$ . We will again use abbreviated statements for  $\Vdash$  and  $\mathcal{A}$  since this is also a single-world model. Since in such a model any  $G$  is vacuously true at all accessible worlds,

$$\Vdash s : G \quad \iff \quad \mathcal{A}(s, G)$$

Since  $\mathcal{A}(t, \neg(p \rightarrow t' : p))$ , in order to falsify  $F$  it is sufficient to show that  $\neg\mathcal{A}(k, \perp)$ . Proof by contradiction. Suppose towards a contradiction that  $\mathcal{A}(k, \perp)$ . By Theorem 3.3.41.4, according to (3.3.38)

$$*(t, \neg(p \rightarrow t' : p)) \vdash_{*\mathcal{CS}} *(k, \perp) .$$

by Lemma 3.4.10.2,

$$\neg(p \rightarrow t' : p), \quad t : [\neg(p \rightarrow t' : p)] \quad \vdash_{\mathbf{J4}_{\mathcal{CS}}} \quad \perp .$$

But this cannot be the case since in the proof of Theorem 6.2.1 we have constructed an F-model with both hypotheses being true. It was a  $\mathbf{JD4}_{\mathcal{CS}'}$ -model  $\mathfrak{M}' = (W', R', V', \mathcal{A}')$ , where  $\mathcal{CS}'$  is the largest constant specification

for JD4 without self-referential constants. All axioms of **J4** are also axioms of JD4; self-referentiality of constants is logic independent. Thus,  $\mathcal{CS} \subseteq \mathcal{CS}'$ .

$R'$  in JD4-models is transitive,  $\mathcal{A}'$  satisfies Application and Sum Closure conditions.  $\mathcal{A}'$  also satisfies the Monotonicity condition. It remains to note that  $\mathcal{A}'$  satisfied  $\mathcal{CS}'$  Closure condition and hence  $\mathcal{CS}$  Closure condition too. Thus, the  $\text{JD4}_{\mathcal{CS}'}$ -model constructed in the proof of Theorem 6.2.1 is also a  $\text{J4}_{\mathcal{CS}}$ -model for the two hypotheses. A satisfiable set of formulas cannot be contradictory. This contradiction shows that  $\neg\mathcal{A}(k, \perp)$ .  $\square$

### 6.3 Knowledge without Self-Referentiality

Unlike the four modal logics discussed in the previous section, logics **K** and **D** can be realized without any self-referential cycles let alone self-referential constants, which are essentially cycles of length 1.

More precisely, we will show that  $(\text{JD}_{\mathcal{CS}})^\circ = \text{D}$  and  $(\text{J}_{\mathcal{CS}})^\circ = \text{K}$  for some non-self-referential constant specifications  $\mathcal{CS}$ .

To construct such constant specifications, we will divide the set of constants into levels indexed by non-negative integers, with each level consisting of countably many constants. Let  $\ell(c)$  denote the level of a constant  $c$ . For

either logic, let

$$\mathcal{CS} = \{c : A \in \mathcal{TCS} \mid \text{for all constants } a \text{ that occur in } A, \ell(a) < \ell(c)\} . \quad (6.3.1)$$

This constant specification is *axiomatically appropriate*, i.e., every axiom has at least one constant justifying it.

Since the constant specification (6.3.1) has infinitely many constants on each level, it is always possible to choose a fresh constant  $c$  whenever one is wanting.

**Theorem 6.3.1.** *Pairs D/JD and K/J describe knoweldge/belief that is not self-referential.*

*Proof.* We will prove that  $(JD_{\mathcal{CS}})^\circ = D$  and  $(J_{\mathcal{CS}})^\circ = K$  for the  $\mathcal{CS}$  from (6.3.1). Since  $JL_{\mathcal{CS}} \subseteq JL$ , we have  $(JD_{\mathcal{CS}})^\circ \subseteq JD^\circ = D$  and  $(J_{\mathcal{CS}})^\circ \subseteq J^\circ = K$ .

To show the other inclusion, we will reprove the Realization Theorem using the  $\mathcal{CS}$  from (6.3.1). One of the ways to prove Realization is by step-by-step transformation of a cut-free Gentzen derivation of a modal theorem  $\varphi$  into a Hilbert derivation of its realization  $\varphi^r$ . More precisely, a cut-free Gentzen derivation

$$\vdash \Gamma \Rightarrow \Delta$$

is being transformed into a Hilbert derivation

$$\Gamma^r \vdash \bigvee \Delta^r .$$

(As always, the empty disjunction is interpreted as  $\perp$ .) A detailed description of this procedure can be found in [Art01, BK06].

Axioms of the Gentzen modal system are restricted to  $\perp \Rightarrow$  and  $p \Rightarrow p$  for sentence letters  $p$  to have a better control over where and how  $\Box$ 's are introduced. All occurrences of  $\Box$  in the Gentzen modal derivation are divided into families of related occurrences. A cut-free derivation preserves polarity of formulas, so there are positive and negative families of  $\Box$ 's. We realize each negative family by a fresh justification variable. A positive family is realized by a sum of auxiliary variables  $v_1 + \dots + v_n$ , one variable per each use of a Gentzen modal rule to introduce a  $\Box$  from this family. If all  $\Box$ 's from a positive family are introduced by Weakening, the family is instantiated by a fresh justification variable. The transformation is done by induction on the depth of the Gentzen derivation.

The Gentzen axioms, propositional rules, and Contraction can be translated using the standard propositional translation from Gentzen into Hilbert. Since the reasoning involved is purely propositional, neither Axiom Internalization is used, nor are new constants introduced. Weakening does not require

Axiom Internalization either; it may bring constants from other branches, but never a fresh constant. Thus, new constants are introduced by Axiom Internalization only to translate modal rules. The only modal rule for logic  $\mathbf{K}$  is (2.3.1):

$$\frac{\varphi_1, \dots, \varphi_n \Rightarrow \psi}{\Box\varphi_1, \dots, \Box\varphi_n \Rightarrow \Box\psi} .$$

In addition, logic  $\mathbf{D}$  has rule (2.3.2):

$$\frac{\varphi_1, \dots, \varphi_n, \xi \Rightarrow}{\Box\varphi_1, \dots, \Box\varphi_n, \Box\xi \Rightarrow}$$

(see, for instance, [Wan94, Fit07a]). To translate both rules we use the Internalization Property (Lemma 3.2.22).

Consider the  $\mathbf{K}$ -rule (2.3.1) first. By IH, we already have a Hilbert derivation of

$$\varphi_1^r, \dots, \varphi_n^r \vdash \psi^r .$$

Internalizing this derivation, we get

$$x_1:\varphi_1^r, \dots, x_n:\varphi_n^r \vdash t:\psi^r$$

for some  $t$ , where each  $x_i$  is the chosen realization of the negative  $\Box$  in front of  $\varphi_i$ . We then substitute  $t$  for the auxiliary variable that corresponds to this modal rule in the sum realization of the  $\Box$  in front of  $\psi$  throughout the Hilbert proof.

The D-rule (2.3.2) is similar. Internalization here yields

$$x_1:\varphi_1^r, \dots, x_n:\varphi_n^r, x_{n+1}:\xi^r \vdash t:\perp .$$

Using axiom A7,  $t:\perp \rightarrow \perp$ , and *modus ponens*, we can derive  $\perp$ . Since no positive  $\Box$  is introduced, there is no global substitution of auxiliary variables.

The proof of Lemma 3.2.22 shows that the rule  $R4_{\mathcal{CS}}^!$  in the internalized derivation appears only where axioms or instances of  $R4_{\mathcal{CS}}^!$  were used in the original derivation. We are free to pick a fresh constant every time.

So how can a self-referential cycle appear if we always pick fresh constants? Where does it appear for stronger modal logics? Here is the answer. When a term  $t$  substitutes for an auxiliary variable  $v$ , which appears in an instance of  $R4_{\mathcal{CS}}^!$ ,

$$! \dots !c:\dots !c:c:A(v) ,$$

the constant  $c$  can *a priori* occur in  $t$ . As shown in Sect. 6.2, this cannot be avoided in many logics with other modal Gentzen rules.

We show how to avoid such occurrences of  $c$  in  $t$  for K and D while staying within constant specification (6.3.1).

**Definition 6.3.2.** The *depth of an occurrence of  $\Box$  in a modal formula*  $\varphi$  is defined by induction on the size of  $\varphi$ :

- the outer  $\Box$  in  $\Box\psi$  has depth 0 in  $\Box\psi$ ;

- for any occurrence of  $\Box$  inside  $\psi$ , its depth in  $\Box\psi$  is obtained by adding 1 to its depth in  $\psi$ .  $\blacktriangleleft$

**Definition 6.3.3.** The *level of an occurrence of  $\Box$  in a Gentzen derivation* is defined as its depth in the formula it occurs in plus the number of modal rules used on its branch after this occurrence.  $\blacktriangleleft$

**Lemma 6.3.4.** *In a cut-free Gentzen  $\mathsf{K}$  or  $\mathsf{D}$  derivation of  $\Rightarrow \varphi$ , levels of all occurrences of  $\Box$  from a given family are equal to the depth of the family's occurrence in  $\varphi$ .*

*Proof.* The proof is a rather easy induction on the depth of the derivation.  $\square$

Let  $N$  be the largest level of  $\Box$ 's in the given cut-free derivation. As we showed, a new constant can be introduced only as part of Internalization while translating a modal rule. For all rules of level  $i$ , let us always use constants of level  $N - i$ . When constants introduced later on a branch refer to constants introduced on this branch earlier, the former have larger levels because the levels of modal rules decrease toward the root of the derivation. It remains to show that the substitution of terms for auxiliary variables does not violate the level structure of (6.3.1).

Indeed, every time a modal rule is used on a branch, all  $\Box$ 's it introduces have the level of this rule, say  $m$ , which is strictly smaller than the levels of

all  $\Box$ 's already on the branch. Suppose the Internalization used to translate this modal rule introduced an Axiom Internalization  $c: A(v)$  with an auxiliary variable  $v$ . This  $v$  corresponds to a family of  $\Box$ 's already present on the branch, which must have a larger level  $l > m$ . Wherever the modal rule corresponding to  $v$  occurs, by Lemma 6.3.4, it has the same level  $l$ . Therefore, when a term  $t$  substitutes for  $v$ , all the constants in  $t$  will have level

$$N - l < N - m = \ell(c) .$$

Thus, substitutions do not violate the conditions of our constant specification. □

## 6.4 Conclusions and Future Work

This thesis was mostly devoted to decidability and complexity questions for pure and hybrid justification logics. The following is a list of main results obtained:

1. The Substitution Property in its traditional formulation only holds for schematic  $\mathcal{CS}$ . An alternative Substitution Property with Renaming of Constants is formulated and proven for axiomatically appropriate  $\mathcal{CS}$  (Lemmas 3.2.30 and 3.5.13).
2. Inadequacy of F-models is shown for  $\text{JD}_{\mathcal{CS}}$ ,  $\text{JD4}_{\mathcal{CS}}$ , and  $\text{JD45}_{\mathcal{CS}}$  when

$\mathcal{CS}$  is not axiomatically appropriate (Example 3.3.23).

3. Alternative Fk-models are developed for these logics (Def. 3.3.24). Soundness and completeness of Fk-models are demonstrated (Theorem 3.3.25).
4. A complete description of minimal admissible evidence functions for M-models, F-models, and AF-models is obtained for hybrid logics (Theorems 3.5.18 and 3.5.20) and pure justification logics without negative introspection (Theorems 3.3.34 and 3.3.41).
5. Nikolai Krupski's results about axiomatization of the reflected fragment of LP is generalized to reflected fragments of hybrid logics (Theorem 3.5.23) and other pure justification logics without negative introspection (Theorem 3.4.2).
6. Some interesting facts about the relationship of derivations from hypotheses in a justification logic and in its reflected fragment are studied (Examples 3.4.5 and 3.4.6, Lemmas 3.4.8, 3.4.10, and 3.5.25).
7. A general framework is developed for proving decidability of justification logics via the Finitary Model Property (Def. 4.3.2, Theorem 4.3.3, Theorem 4.4.9).
8. Decidability of hybrid logics and of pure justification logics without neg-

ative introspection provided that  $\mathcal{CS}$  is decidable and almost schematic (and additionally axiomatically appropriate for  $\text{JD}_{\mathcal{CS}}$  and  $\text{JD4}_{\mathcal{CS}}$ ) is obtained as a corollary of the Finitary Models method (Theorems 4.4.25 and 4.4.28). Although decidability of most of these pure justification logics was known, this result is new for most hybrid logics.

9. It is shown that the condition that  $\mathcal{CS}$  be almost schematic cannot be dropped by demonstrating examples of undecidable pure and hybrid justification logics with decidable  $\mathcal{CS}$  ([Kuz05], Theorem 4.5.1).
10. N. Krupski's NP upper bound on complexity of the reflected fragment of LP is extended to reflected fragments of all hybrid logics and all pure justification logics without negative introspection for decidable almost schematic  $\mathcal{CS}$ ; the result is also generalized to derivations from hypotheses (Theorem 5.1.5).
11. Upper bound on complexity of  $\text{J}_{\mathcal{CS}}$ ,  $\text{JT}_{\mathcal{CS}}$ ,  $\text{J4}_{\mathcal{CS}}$ , and  $\text{LP}_{\mathcal{CS}}$  with decidable almost schematic  $\mathcal{CS}$  is shown to be  $\Pi_2^p$  ([Kuz00]). The algorithm is shaped as a tableau derivation (Theorem 5.2.2).
12. An omission is found in the complexity estimate of  $\text{JD}_{\mathcal{CS}}$  and  $\text{JD4}_{\mathcal{CS}}$  in [Kuz00]. It is shown how prefixed tableaux *a la* Fitting-Massacci can be adapted to showing the same upper bound for  $\text{JD}_{\mathcal{CS}}$  with decidable,

almost schematic, and axiomatically appropriate  $\mathcal{CS}$  (Theorem 5.2.4).

It remains an open problem to show this upper bound for  $\text{JD4}_{\mathcal{CS}}$ ; some difficulties are outlined in Note 5.2.6.

13. Lower bound for hybrid logics, which are typically PSPACE-hard, is shown through a semantic proof of their conservativity over the respective multimodal logics (Theorem 5.4.1).
14. A matching upper bound is found for  $\text{S4}_1\text{LP}_{\mathcal{CS}}$  with a decidable and schematic  $\mathcal{CS}$ . Thus,  $\text{S4}_1\text{LP}_{\mathcal{CS}}$  with a decidable schematic  $\mathcal{CS}$  is PSPACE-complete ([Kuz06a], Theorem 5.4.4).
15. Strong self-referentiality of  $\text{T}$ ,  $\text{K4}$ ,  $\text{D4}$ , and  $\text{S4}$  is shown ([BK06], Theorems 6.2.1 and 6.2.3, [Kuz08]).
16. It is shown that  $\text{K}$  and  $\text{D}$  are not self-referential (Theorem 6.3.1, [Kuz08]).

Naturally, there are many open problems in the area.

- It is discussed why the apparatus of minimal functions fails in presence of negative introspection. It remains to find an adequate tool for constructing models for these justification logics. The absence of such tools presents a major obstacle to developing a decision procedure for these logics.

- Decidability of  $\text{JD}_{\mathcal{CS}}$  and  $\text{JD4}_{\mathcal{CS}}$  requires an extra condition compared to other justification logics, namely axiomatic appropriateness of  $\mathcal{CS}$ . It is unknown whether this condition is substantial, i.e., whether either of these justification logics can be undecidable if  $\mathcal{CS}$  is decidable and schematic but not axiomatically appropriate.
- Lemma 5.3.7 describes an interesting connection between subclassical propositional systems and  $\text{JD}_{\mathcal{CS}}$  with non-axiomatically appropriate  $\mathcal{CS}$ . It would be interesting to explore this relationship further. Can this relationship be exploited to learn more about decidability discussed in the previous item?
- It seems reasonably straightforward to construct a PSPACE decision procedure for  $\text{JD4}_{\mathcal{CS}}$  with decidable, almost schematic, and axiomatically appropriate  $\mathcal{CS}$  using F-models. This upper bound, nevertheless does not seem optimal given (presumably) much lower upper bounds of  $\Pi_2^p$  for other justification logics.
- Very few of the complexity bounds for justification logics are tight. Most prominently there is no nontrivial lower bound known for LP itself.
- Still very little is known about the complexity of hybrid logics. It

seems that Demri's methods from [Dem00] can be applied to  $T_1LP_{CS}$  and  $S5_1LP_{CS}$ , but generalizing them to a larger number of modalities  $n$  meets with substantial difficulties rooted in modal rather than in justification part. Already the case of  $S4_2LP$  is quite non-trivial.

# Bibliography

- [AK06a] Sergei [N.] Artemov and Roman Kuznets. Logical omniscience via proof complexity. Technical Report TR–2006005, CUNY Ph.D. Program in Computer Science, May 2006.
- [AK06b] Sergei [N.] Artemov and Roman Kuznets. Logical omniscience via proof complexity. In Zoltán Ésik, editor, *Computer Science Logic, 20th International Workshop, CSL 2006, 15th Annual Conference of the EACSL, Szeged, Hungary, September 25–29, 2006, Proceedings*, volume 4207 of *Lecture Notes in Computer Science*, pages 135–149. Springer, 2006.
- [AKS99] S[ergei N.] Artemov, E. Kazakov, and D. Shapiro. Logic of knowledge with justifications. Technical Report CFIS 99–12, Cornell University, 1999.
- [AN04] Sergei [N.] Artemov and Elena Nogina. Logic of knowledge with justifications from the provability perspective. Technical Report TR–2004011, CUNY Ph.D. Program in Computer Science, August 2004.
- [AN05a] Sergei [N.] Artemov and Elena Nogina. Basic systems of epistemic logic with justification. Technical Report TR–2005004, CUNY Ph.D. Program in Computer Science, February 2005.
- [AN05b] Sergei [N.] Artemov and Elena Nogina. Introducing justification into epistemic logic. *Journal of Logic and Computation*, 15(6):1059–1073, December 2005.
- [AN05c] Sergei [N.] Artemov and Elena Nogina. On epistemic logic with justification. In Ron van der Meyden, editor, *Theoretical Aspects of*

*Rationality and Knowledge, Proceedings of the Tenth Conference, June 10–12, 2005, National University of Singapore, Singapore*, pages 279–294. National University of Singapore, 2005.

- [Ant06a] Evangelia Antonakos. Comparing justified and common knowledge. In *2005 Summer Meeting of the Association for Symbolic Logic, Logic Colloquium '05, Athens, Greece, July 28–August 3, 2005*, volume 12 of *Bulletin of Symbolic Logic*, pages 323–324, June 2006. Abstract.
- [Ant06b] Evangelia Antonakos. Justified knowledge is sufficient. Technical Report TR–2006004, CUNY Ph.D. Program in Computer Science, April 2006.
- [Ant07a] Evangelia Antonakos. Epistemic logic with common and justified common knowledge. In *2007 Annual Meeting of the Association for Symbolic Logic, University of Florida, Gainesville, Florida, March 10–13, 2007*, volume 13 of *Bulletin of Symbolic Logic*, pages 402–403, September 2007. Abstract.
- [Ant07b] Evangelia Antonakos. Justified and common knowledge: Limited conservativity. In Sergei N. Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science, International Symposium, LFCS 2007, New York, NY, USA, June 4–7, 2007, Proceedings*, volume 4514 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2007.
- [Art94] Sergei [N.] Artëmov. Logic of proofs. *Annals of Pure and Applied Logic*, 67(1–3):29–59, May 1994.
- [Art95] Sergei N. Artemov. Operational modal logic. Technical Report MSI 95–29, Cornell University, 1995.
- [Art98] Sergei N. Artemov. Logic of Proofs: a unified semantics for modality and  $\lambda$ -terms. Technical Report CFIS 98–06, Cornell University, 1998.
- [Art01] Sergei N. Artemov. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic*, 7(1):1–36, March 2001.

- [Art04a] Sergei [N.] Artemov. Evidence-based common knowledge. Technical Report TR-2004018, CUNY Ph.D. Program in Computer Science, November 2004.
- [Art04b] S[ergei] N. Artemov. Kolmogorov and Gödel's approach to intuitionistic logic: current developments. *Russian Mathematical Surveys*, 59(2):203–229, 2004. Translated from Russian; first published in 2004.
- [Art06] Sergei [N.] Artemov. Justified common knowledge. *Theoretical Computer Science*, 357(1–3):4–22, July 2006.
- [Art07] Sergei [N.] Artemov. Justification logic. Technical Report TR-2007019, CUNY Ph.D. Program in Computer Science, October 2007.
- [BdRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.
- [BK05] Vladimir [N.] Brezhnev and Roman Kuznets. Making knowledge explicit: How hard it is. Technical Report TR-2005003, CUNY Ph.D. Program in Computer Science, February 2005.
- [BK06] Vladimir [N.] Brezhnev and Roman Kuznets. Making knowledge explicit: How hard it is. *Theoretical Computer Science*, 357(1–3):23–34, July 2006.
- [Bre99] Vladimir N. Brezhnev. Explicit counterparts of modal logic (in Russian). Master's thesis, Lomonosov Moscow State University, 1999.
- [Bre00] Vladimir N. Brezhnev. On explicit counterparts of modal logics. Technical Report CFIS 2000–05, Cornell University, 2000.
- [CB83] Jacques Corbin and Michel Bidoit. A rehabilitation of Robinson's unification algorithm. In R. E. A. Mason, editor, *Information Processing 83, Proceedings of the 9th World Computer Congress, Paris, France, September 19–23, 1983*, pages 909–914. North-Holland/IFIP, 1983.

- [Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Conference Record of Third Annual ACM Symposium on Theory of Computing, Papers Presented at the Symposium, Shaker Heights, Ohio, May 3, 4, 5, 1971*, pages 151–158. ACM, 1971.
- [CZ97] Alexander Chagrov and Michael Zakharyashev. *Modal Logic*, volume 35 of *Oxford Logic Guides*. Oxford University Press, 1997.
- [Dem00] Stéphane Demri. Complexity of simple dependent bimodal logics. In Roy Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX 2000, St Andrews, Scotland, UK, July 3–7, 2000, Proceedings*, volume 1847 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2000.
- [Fey65] Robert Feys. *Modal Logics*. E. Nauwelaerts/Gauthier-Villars, 1965.
- [FHMV95] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
- [Fit72] Melvin Fitting. Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic*, XIII(2):237–247, April 1972.
- [Fit03a] Melvin Fitting. A semantic proof of the realizability of modal logic in the Logic of Proofs. Technical Report TR–2003010, CUNY Ph.D. Program in Computer Science, September 2003.
- [Fit03b] Melvin Fitting. A semantics for the Logic of Proofs. Technical Report TR–2003012, CUNY Ph.D. Program in Computer Science, September 2003.
- [Fit04a] Melvin Fitting. Quantified LP. Technical Report TR–2004019, CUNY Ph.D. Program in Computer Science, December 2004.
- [Fit04b] Melvin Fitting. Semantics and tableaux for LPS4. Technical Report TR–2004016, CUNY Ph.D. Program in Computer Science, October 2004.
- [Fit05] Melvin Fitting. The logic of proofs, semantically. *Annals of Pure and Applied Logic*, 132(1):1–25, February 2005.

- [Fit06a] Melvin Fitting. A quantified logic of evidence. In R. de Queiroz, A. Macintyre, and G. Bittencourt, editors, *Proceedings of the 12th Workshop on Logic, Language, Information and Computation (WoLLIC 2005), Florianópolis, Santa Catarina, Brazil, 19–22 July 2005*, volume 143 of *Electronic Notes in Theoretical Computer Science*, pages 59–71. Elsevier, January 2006.
- [Fit06b] Melvin Fitting. A replacement theorem for LP. Technical Report TR–2006002, CUNY Ph.D. Program in Computer Science, March 2006.
- [Fit07a] Melvin Fitting. Modal proof theory. In Patrick Blackburn, Johan van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, chapter 2, pages 85–138. Elsevier, 2007.
- [Fit07b] Melvin Fitting. Realizations and LP. In Sergei N. Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science, International Symposium, LFCS 2007, New York, NY, USA, June 4–7, 2007, Proceedings*, volume 4514 of *Lecture Notes in Computer Science*, pages 212–223. Springer, 2007.
- [Fit07c] Melvin Fitting. Realizing substitution instances of modal theorems. Technical Report TR–2007006, CUNY Ph.D. Program in Computer Science, March 2007.
- [FM98] Melvin Fitting and Richard L. Mendelsohn. *First-Order Modal Logic*, volume 277 of *Synthese Library*. Kluwer Academic Publishers, 1998.
- [Get63] Edmund L. Gettier. Is justified true belief knowledge? *Analysis*, 23(6):121–123, June 1963.
- [Hin62] Jaakko Hintikka. *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press, 1962.
- [Hin75] Jaakko Hintikka. Impossible possible worlds vindicated. *Journal of Philosophical Logic*, 4(3):475–484, August 1975.

- [HM85] Joseph Y. Halpern and Yoram Moses. A guide to the modal logics of knowledge and belief: Preliminary draft. In Aravind K. Joshi, editor, *Proceedings of the 9th International Joint Conference on Artificial Intelligence, IJCAI 1985, Los Angeles, CA, August 1985*, volume 1, pages 480–490. Morgan Kaufmann, 1985.
- [HM92] Joseph Y. Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, April 1992.
- [Kru97] Vladimir N. Krupski. Operational logic of proofs with functionality condition on proof predicate. In Sergei Adian and Anil Nerode, editors, *Logical Foundations of Computer Science, 4th International Symposium, LFCS'97, Yaroslavl, Russia, July 6–12, 1997, Proceedings*, volume 1234 of *Lecture Notes in Computer Science*, pages 167–177. Springer, 1997.
- [Kru01] Vladimir N. Krupski. The single-conclusion proof logic and inference rules specification. In Yuri [V.] Matiyasevich, editor, *First St. Petersburg Conference on Days of Logic and Computability, May 26–29, 1999, Steklov Institute of Mathematics, St. Petersburg, Russia*, volume 113 of *Annals of Pure and Applied Logic*, pages 181–206. Elsevier, December 2001.
- [Kru03] Nikolai V. Krupski. On the complexity of the reflected logic of proofs. Technical Report TR–2003007, CUNY Ph.D. Program in Computer Science, May 2003.
- [Kru06a] Nikolai V. Krupski. *On Certain Algorithmic Problems for Formal Systems with Internalization Property*. PhD thesis, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, April 2006. In Russian.
- [Kru06b] Nikolai V. Krupski. On the complexity of the reflected logic of proofs. *Theoretical Computer Science*, 357(1–3):136–142, July 2006.
- [Kru06c] Vladimir N. Krupski. Reference constructions in the single-conclusion proof logic. In Valentin Shehtman, editor, *Computer*

*Science Applications of Modal Logic, International Conference, Moscow, Russia, September 5–9, 2005, Proceedings*, volume 16 of *Journal of Logic and Computation*, pages 645–661. Oxford University Press, October 2006.

- [Kru06d] Vladimir N. Krupski. Referential logic of proofs. *Theoretical Computer Science*, 357(1–3):143–166, July 2006.
- [Kuz00] Roman Kuznets. On the complexity of explicit modal logics. In Peter G. Clote and Helmut Schwichtenberg, editors, *Computer Science Logic, 14th International Workshop, CSL 2000, Annual Conference of the EACSL, Fischbachau, Germany, August 21–26, 2000, Proceedings*, volume 1862 of *Lecture Notes in Computer Science*, pages 371–383. Springer, 2000.
- [Kuz05] Roman Kuznets. On decidability of the logic of proofs with arbitrary constant specifications. In *2004 Annual Meeting of the Association for Symbolic Logic, Carnegie Mellon University, Pittsburgh, PA, May 19–23, 2004*, volume 11 of *Bulletin of Symbolic Logic*, page 111, March 2005. Abstract.
- [Kuz06a] Roman Kuznets. Complexity of evidence-based knowledge. In Sergei [N.] Artemov and Rohit Parikh, editors, *Proceedings of the Workshop on Rationality and Knowledge, 18th European Summer School in Logic, Language, and Information, 7–11 August 2006, Universidad de Málaga*, pages 66–75, 2006.
- [Kuz06b] Roman Kuznets. Logic of Proofs as a measure of Hilbert-style proof complexity. In *2005 Summer Meeting of the Association for Symbolic Logic, Logic Colloquium '05, Athens, Greece, July 28–August 3, 2005*, volume 12 of *Bulletin of Symbolic Logic*, page 355. June 2006. Abstract presented by title.
- [Kuz06c] Roman Kuznets. On self-referentiality in modal logic. In *2005–06 Winter Meeting of the Association for Symbolic Logic, The Hilton New York Hotel, New York, NY, December 27–29, 2005*, volume 12 of *Bulletin of Symbolic Logic*, page 510. September 2006. Abstract.

- [Kuz08] Roman Kuznets. Self-referentiality of justified knowledge. In E[dward A.] Hirsch, A. Razborov, A. Semenov, and A. Slissenko, editors, *Third International Computer Science Symposium in Russia, CSR 2008, Moscow, Russia, June 7–12, 2008, Proceedings*, volume 5010 of *Lecture Notes in Computer Science*, pages 228–239. Springer, 2008.
- [Lad77] Richard E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, September 1977.
- [Mas94] Fabio Massacci. Strongly analytic tableaux for normal modal logics. In Alan Bundy, editor, *Automated Deduction — CADE-12, 12th International Conference on Automated Deduction, Nancy, France, June 26 - July 1, 1994, Proceedings*, volume 814 of *Lecture Notes in Computer Science*, pages 723–737. Springer-Verlag, 1994.
- [Mas00] Fabio Massacci. Single step tableaux for modal logics: Computational properties, complexity and methodology. *Journal of Automated Reasoning*, 24(3):319–364, April 2000.
- [Mil07] Robert Milnikel. Derivability in certain subsystems of the Logic of Proofs is  $\Pi_2^p$ -complete. *Annals of Pure and Applied Logic*, 145(3):223–239, March 2007.
- [Mkr97] Alexey Mkrtychev. Models for the Logic of Proofs. In Sergei Adian and Anil Nerode, editors, *Logical Foundations of Computer Science, 4th International Symposium, LFCS'97, Yaroslavl, Russia, July 6–12, 1997, Proceedings*, volume 1234 of *Lecture Notes in Computer Science*, pages 266–275. Springer, 1997.
- [Nog94] Elena Nogina. Logic of Proofs with the strong provability operator. Technical Report ML-94-10, Institute for Logic, Language and Computation, University of Amsterdam, October 1994.
- [Pac05] Eric Pacuit. A note on some explicit modal logics. In *Proceedings of the 5th Panhellenic Logic Symposium, Athens, Greece, July 25–28, 2005*. University of Athens, 2005.

- [Par87] Rohit Parikh. Knowledge and the problem of logical omniscience. In Zbigniew W. Ras and Maria Zemankova, editors, *Methodologies for Intelligent Systems, Proceedings of the Second International Symposium, ISMIS 1987, Charlotte, North Carolina, USA, October 14–17, 1987*, pages 432–439. North-Holland/Elsevier, 1987.
- [Par95] Rohit Parikh. Logical omniscience. In Daniel Leivant, editor, *Logic and Computational Complexity, International Workshop LCC '94, Indianapolis, IN, USA, October 13–16, 1994, Selected Papers*, volume 960 of *Lecture Notes in Computer Science*, pages 22–29. Springer, 1995.
- [Par05] Rohit Parikh. Logical omniscience and common knowledge; WHAT do we know and what do WE know? In Ron van der Meyden, editor, *Theoretical Aspects of Rationality and Knowledge, Proceedings of the Tenth Conference, June 10–12, 2005, National University of Singapore, Singapore*, pages 62–77. National University of Singapore, 2005.
- [Rub06a] N[atalia] M. Rubtsova. Evidence-based knowledge for S5. In *2005 Summer Meeting of the Association for Symbolic Logic, Logic Colloquium '05, Athens, Greece, July 28–August 3, 2005*, volume 12 of *Bulletin of Symbolic Logic*, pages 344–345, June 2006. Abstract.
- [Rub06b] Natalia [M.] Rubtsova. Evidence reconstruction of epistemic modal logic S5. In Dima Grigoriev, John Harrison, and Edward A. Hirsch, editors, *Computer Science — Theory and Applications, First International Computer Science Symposium in Russia, CSR 2006, St. Petersburg, Russia, June 8–12, 2006, Proceedings*, volume 3967 of *Lecture Notes in Computer Science*, pages 313–321. Springer, 2006.
- [Rub06c] Natalia [M.] Rubtsova. On realization of S5-modality by evidence terms. In Valentin Shehtman, editor, *Computer Science Applications of Modal Logic, International Conference, Moscow, Russia, September 5–9, 2005, Proceedings*, volume 16 of *Journal of Logic and Computation*, pages 671–684. Oxford University Press, October 2006.

- [Rub06d] Natalia [M.] Rubtsova. Semantics for justification logic corresponding to **S5**. In Sergei [N.] Artemov and Rohit Parikh, editors, *Proceedings of the Workshop on Rationality and Knowledge, 18th European Summer School in Logic, Language, and Information, 7–11 August 2006, Universidad de Málaga*, pages 124–132, 2006.
- [Sid97] Tatiana Sidon. Provability logic with operations on proofs. In Sergei Adian and Anil Nerode, editors, *Logical Foundations of Computer Science, 4th International Symposium, LFCS'97, Yaroslavl, Russia, July 6–12, 1997, Proceedings*, volume 1234 of *Lecture Notes in Computer Science*, pages 342–353. Springer, 1997.
- [Smo85] C. Smoryński. *Self-Reference and Modal Logic*. Universitext. Springer-Verlag, 1985.
- [Sta79] Richard Statman. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science*, 9(1):67–72, July 1979.
- [Šve03] Vítězslav Švejdar. On the polynomial-space completeness of intuitionistic propositional logic. *Archive for Mathematical Logic*, 42(7):711–716, October 2003.
- [Urq81] Alasdair Urquhart. Decidability and the finite model property. *Journal of Philosophical Logic*, 10(3):367–370, August 1981.
- [Wan94] Heinrich Wansing. Sequent calculi for normal modal propositional logics. *Journal of Logic and Computation*, 4(2):125–142, April 1994.
- [Yav01a] Tatiana Yavorskaya (Sidon). Logic of proofs and provability. *Annals of Pure and Applied Logic*, 113(1–3):345–372, December 2001.
- [Yav01b] Rostislav E. Yavorsky. Provability logics with quantifiers on proofs. In Yuri [V.] Matiyasevich, editor, *First St. Petersburg Conference on Days of Logic and Computability, May 26–29, 1999, Steklov Institute of Mathematics, St. Petersburg, Russia*, volume 113 of *Annals of Pure and Applied Logic*, pages 373–387. Elsevier, December 2001.

# Index

- \*-calculus
  - \*! $\mathcal{CS}$ -calculus, 103, 104
  - \* $\mathcal{CS}$ -calculus, 103, 104
  - axioms and rules, 103
- \*-closed branch, 211, 219
- \*-expression, 122
- \*A2, \*A3, \*A5, \*-calculus rules, 103
- $\Phi$ , 266
- $\Psi$ , 267
- $\cdot^*$ 
  - for an evidence function, 104
- $\cdot^*$  for evidence function and world, 104
- $\cdot^\circ$ 
  - for a formula, 25
  - for a set of formulas, 26
- $\cdot^\#$ 
  - for a set of \*-expressions, 124
  - for a set of formulas, 68
- $\cdot^\cdot$ , for a set of \*-expressions, 122
- $\cdot^{b_i}$ , 170
- $\cdot^b$ , 170
- $\cdot^{\#_i}$ , 170
- $\cdot_0$ , for a justification logic, 31
- $\cdot_{\mathcal{CS}}$ , for a justification logic, 30
- $\odot$ , 241
- $\lambda$ , 239
- $\subseteq$  for evidence functions, 94, 136
- \*! $\mathcal{CS}$ -DERIVE, 198
- \* $\mathcal{CS}$ , \*-calculus axiom, 103
- \* $\mathcal{CS}^!$ , \*-calculus axiom, 103
- \* $\mathcal{CS}$ -DERIVE, 195–198

- $[\cdot]^c$ , 60
- $|\cdot|$ , 23, 127
- 4, modal axiom, 5
- $4_i$ 
  - EBK axioms, 129
  - modal axioms, 6
- 5, modal axiom, 5
- $5_i$ 
  - EBK axioms, 129
  - modal axioms, 6
- A1–A5, EBK axioms, 128
- A1–A7, justification axioms, 27–28
- $\mathcal{AEF}(\cdot)$ , 94–96
- AF-model, 132–134
- Antonakos, Evangelia, 2
- Application Axiom A2, 27
- Application Rule \*A2, 103
- Artëmov, Sergei, *see* Artemov, Sergei
- Artemov, Sergei, 1, 2, 29, 33, 34, 36, 37, 40–42, 44, 45, 63, 78, 85, 89, 115, 116, 126, 130, 131, 135, 140, 141, 191, 192, 210, 262, 282
- $\mathcal{A}^{\text{tot}}$ , 100
- $\mathcal{A}_{\text{True}}$ , 97
- Axiom Internalization Rule, 28
  - restricted to  $\mathcal{CS}$ , 30, 128
  - with positive introspection, 28
    - restricted to  $\mathcal{CS}$ , 31
- Axiom specification, *see* Constant specification
- $B_\emptyset$ , 112
- Based
  - Kripke model on Kripke frame, 12
  - evidence functions, 94
- Based on, 94, 136

- Bidoit, Michel, 208
- Binary relation
- Euclidean, 11
  - reflexive, 10
  - serial, 10
  - symmetric, 11
  - transitive, 10
- Blackburn, Patrick, 8, 15, 146–148, 150
- Brezhnev, Vladimir, 1, 36, 45, 282
- C1, EBK axiom, 129
- Canonical model
- AF-model
    - finitary, 171
  - F-model, 68
    - finitary, 170
  - M-model, 53
- $\mathcal{C}_{HL_{CS}}$ , 156
- Chagrov, Alexander, 8, 15, 144, 145
- $\mathcal{C}_{JL_{CS}}$ , 156
- Cl, 16
- Classical propositional logic, 16
- Closure conditions
- AF-models
    - Application closure, 133
    - Monotonicity, 134
    - Positive introspection closure, 134
    - $\mathcal{CS}$  closure, 134
    - Sum closure, 133
  - F-models
    - anti-monotonicity, 78, 116
    - Application closure, 59
    - $\mathcal{CS}$  closure, 59
    - Monotonicity, 60
    - Negative Introspection closure, 60
    - Positive Introspection closure, 59

- simplified  $\mathcal{CS}$  closure, 61
  - stability, 79, 116
  - Sum closure, 59
- Fk-models
  - Consistent Evidence condition, 89
- M-models
  - Application closure, 47
  - Consistent Evidence condition, 47
  - $\mathcal{CS}$  closure, 47
  - Positive introspection closure, 47
  - simplified  $\mathcal{CS}$  closure, 48
  - Sum closure, 47
- Completeness Theorem
  - w.r.t. finite models, 164
- EBK logics
  - AF-models, 135
- justification logics
  - F-models, 63, 115, 116
  - F-models (strong version), 78
  - Fk-models, 89
  - Fk-models (strong version), 90
  - M-models, 49, 114
- modal logics, 14–15
- S4LP
  - F-models, 135
- Complexity
  - Cl, 16
  - Int, 16
  - J4, 231
  - J4<sub>CS</sub>
    - schematic  $\mathcal{CS}$ , 231
- lower bounds
  - EBK logics, 233
  - justification logics, 229
  - S4<sub>n</sub>LP<sub>CS</sub>, 233
  - S5<sub>1</sub>LP<sub>CS</sub>, 234
  - S5<sub>n</sub>LP<sub>CS</sub>, 233

- $T_nLP_{CS}$ , 233  
 $LP_{CS}$   
    schematically injective  $CS$ , 231  
of a logic, 15  
 $S4$ , 16  
 $S4LP_{CS}$   
    schematic  $CS$ , 237  
 $S4_n$ , 16  
 $S5$ , 16  
 $S5_n$ ,  $n \geq 2$ , 16  
 $T$ , 16  
 $T_n$ , 16  
upper bounds  
    \*-calculi, 193, 208  
     $J_{CS}$ , 210  
     $J4_{CS}$ , 210  
     $JD_{CS}$ , 217  
     $JT_{CS}$ , 210  
     $LP_{CS}$ , 210  
     $LP_{CS}$ , 210  
     $rJ$ , 210  
     $rJ_{CS}$ , 209  
     $rJ4$ , 210  
     $rJ4_{CS}$ , 209  
     $rJD$ , 210  
     $rJD_{CS}$ , 209  
     $rJD4$ , 210  
     $rJD4_{CS}$ , 209  
     $rJT$ , 210  
     $rJT_{CS}$ , 209  
     $rLP$ , 210  
     $rLP_{CS}$ , 209  
     $rS4_nLP$ , 210  
     $rS4_nLP_{CS}$ , 209  
     $rS5_nLP$ , 210  
     $rS5_nLP_{CS}$ , 209  
     $rT_nLP$ , 210  
     $rT_nLP_{CS}$ , 209  
Connection principle C1, 129  
Consistency Axiom A7, 28

- Consistency Theorem
  - EBK logics, 135
  - justification logics, 37
- Consistent set, 17
  - maximal, 17
  - relative to a set of formulas, 18
- Constant specification, 29
  - almost schematic, 33
  - axiomatically appropriate, 32
  - directly self-referential, 265
  - empty, 31
  - finite, 33
  - injective, 32
  - maximal, *see* total
  - schematic, 33
  - schematically injective, 33
  - self-referential, 265
  - total, 31
- Constructive Necessitation
  - EBK logics, 131
  - justification logics, 39
- Cook, Stephen, 16, 231
- Corbin, Jacques, 208
- $\mathcal{CS}(\cdot)$ , 32
- $\mathcal{CS}$ , constant specification, 29
- D, modal axiom, 6
- Decidability
  - EBK logics, 187–190
  - justification logics, 187–190
- Deduction Theorem
  - EBK logics, 131
  - justification logics, 41
- Demri, StéphaneL, 291
- EBK logic
  - axioms and rules, 128–129
  - S4LP, 130
  - S4<sub>n</sub>LP, 129
  - S4<sub>n</sub>LP<sub>CS</sub>, 129

- S5LP, 130
- S5<sub>n</sub>LP, 129
- S5<sub>n</sub>LP<sub>CS</sub>, 129
- TLP, 130
- T<sub>n</sub>LP, 129
- T<sub>n</sub>LP<sub>CS</sub>, 128–129
- Empty disjunction, 282
- Empty sequence, 239
- Essential call, 242
- Evidence function, 114
  - admissible
    - AF-models, 133
    - F-models, 59, 115
    - M-models, 47
    - on a Kripke frame, 62
    - on a set, 62
  - minimal, 116, 136
    - axiomatic description, 104–106, 110, 137
    - axioms and rules, 136
    - existence, 96–97, 104–106, 110, 136
    - in a class of functions, 94
    - negative introspection, 111–114
    - possible
      - F-models, 92
      - M-models, 92
- Evidence term, *see* Justification term
- Explicit counterpart of a modal logic, *see* Justification logic
- Explicit modal logic, *see* Justification logic
- F-model, 58–63, 115, 116
  - strong model, 115
- Factivity Axiom A4, 28
- Fagin, Ronald, 8, 15, 147, 148
- Feys, Robert, 8, 15
- Finitary model

- for EBK logic, 156
  - for justification logic, 156
- Finitary model property, 153
- Finite frame property, 144
- Finite Model Property, 84, 144
  - refined, 152
  - strong, 146
- Finitely approximable, 144
- Fitting, Melvin, 8–10, 15, 34, 45,
  - 46, 63, 75, 78, 114, 115, 135,
  - 141, 217, 267, 289
- Fk-model, 89
- $Fm$ , 23
- $Fm_n$ , 127
- FMP, *see* Finite Model Property
- Forgetful projection
  - of a formula, 25–26
  - of a justification logic, 26
  - of a set of formulas, 26
- Fully Explanatory property, 79, 115
- Gettier, Edmund, 2, 85
- Glivenko, Valerii, 233
- Ground term, 39
- Halpern, Joseph, 8, 15, 16, 147,
  - 148, 236
- Harrop’s Theorem, 145
- Harrop, Ronald, 145, 146
- Hintikka, Jaakko, 2
- $\mathcal{HL}_n$ , 127
- Hybrid formulas, 127
- Int, 16
- Integer prefix, 217
- Internalization Property
  - EBK logics, 130
  - justification logic, 37
- Intuitionistic propositional logic, 16
- $\mathcal{JL}$ , 22
- $\mathcal{JL}(?)$ , 22
- Justification constant, 22

- Justification counterpart, 26
- Justification formula, 23
- Justification logic
- axioms and rules, 27–28, 30
  - J, 34, 35, 44, 49, 62, 63, 78, 79
  - J4, 34, 35, 44, 49, 62, 63, 78, 79
  - J45, 34, 35, 45, 62, 63, 78, 79
  - J5, 34, 35, 45, 62, 63, 78, 79
  - JD, 34, 35, 44, 49, 62, 63, 78, 79, 89–91
  - JD4, 34, 35, 44, 49, 62, 63, 78, 79, 89–91
  - JD45, 34, 35, 45, 62, 63, 79, 89, 91
  - JT, 34, 35, 44, 49, 62, 63, 78, 79
  - JT4, *see* LP
  - JT45, 34, 35, 44, 45, 62, 63, 79
  - LP, 34, 35, 44, 49, 62, 63, 78, 79
  - LP(D), *see* JD
  - LP(D4), *see* J4
  - LP(K), *see* J
  - LP(K4), *see* J4
  - LP(K5), *see* J5
  - LP(KD45), *see* JD45
  - LP(S5), *see* JT45
  - LP(T), *see* JT
  - LPS5, 44
  - with constant specification  $\mathcal{CS}$ , 30
  - with the empty constant specification, 31
- Justification term, 22
- Justification variable, 22
- K, modal axiom, 5
- Kazakov, Yevgeny, 1, 44

- $K_i$ , EBK axioms, 128
- $K_i$ , modal axioms, 6
- Kripke frame, 11
  - finite, 12
  - monomodal, 12
  - $n$ -modal, 11
- Kripke model, 11
  - finite, 12
  - monomodal, 12
  - $n$ -modal, 11
- Kripke, Saul, 10, 11
- Krupski, Nikolai, 117, 193, 209, 263, 287, 288
- Krupski, Vladimir, 46, 125, 192
- $\mathcal{L}^{-1}$ -call, 243
- $\mathcal{L}$ -call, 243
- Ladner, Richard, 16, 236, 237, 259
- Lifting Lemma
  - EBK logics, 130
  - justification logics, 40
- Local call, 242
- Logic of knowledge with justifications, *see* Justification logic
- Logic of Proofs LP, 35
- M-model, 46–49, 114
- Massacci, Fabio, 9, 217, 289
- Maximal consistent set, 17
  - relative to a set of formulas, 18
- Mendelsohn, Richard, 8, 9, 15
- mgu, 161
- Milnikel, Robert, 29, 34, 210, 231, 262, 263
- Mkrtychev, Alexey, 34, 49, 114, 116, 192, 193
- $\mathcal{ML}$ , 5
- $\mathcal{ML}_n$ , 5
- Modal logic
  - axioms and rules, 5, 6

- D, 7, 14
- D, 10
- D4, 7, 9, 14
- directly self-referential, 265
- K, 7, 9, 14
- K, 10
- K4, 7, 14
- K45, 7, 14
- K5, 7, 14
- KD45, 7, 15
- $K_n$ , 7, 15
- S4, 7, 9, 14, 16
- $S4_n$ , 7, 15, 16
- S5, 7, 15, 16
- $S5_n$ , 7, 15, 16
- self-referential, 266
- T, 7, 9, 14, 16
- $T_n$ , 7, 15, 16
- Modus ponens*, 5, 6, 27, 128
- Monotonicity Axiom A3, 27
- Moses, Yoram, 8, 15, 16, 147, 148, 236
- most general unifier, 161
- MP, *see Modus ponens*
- $Nec_i$ 
  - EBK rules, 128
  - modal rules, 6
- Nec, modal rule, 6
- Necessitation Rule
  - EBK rules, 128
  - modal rules, 6
- Negative Introspection
  - explicit for EBK logics, 141
  - justification axiom A6, 28
  - modal axiom 5, 5
  - modal axioms  $5_i$ , 6
  - weak for EBK logics, 141
- Nogina, Elena, 2, 130, 131, 135, 140, 141, 192

- Normality Axiom K, 5
- Normality Axiom  $K_i$ , 6, 128
- Operational modal logic, *see* Justification logic
- Operations on justifications
  - application  $\cdot$ , 22
  - choice, *see* sum  $+$
  - negative introspection  $?$ , 22
  - proof checker  $!$ , 22
  - sum  $+$ , 22
  - union, *see* sum  $+$
- Pacuit, Eric, 1, 44, 45, 63, 78, 89, 115, 116
- Parikh, Rohit, 2
- Peano, Giuseppe, 1
- Plato, 126
- Positive Introspection
  - justification axiom A5, 28
  - modal axiom 4, 5
  - modal axioms  $4_i$ , 6
- Positive Introspection Rule \*A5, 103
- Possible evidence function
  - AF-models, 136
  - finitary, 154
- Post's Theorem, 144
- Pre-model, *see* M-model
- Proof polynomial, *see* Justification term
- Proof term, *see* Justification term
- Proof-theorem assignment, *see* Evidence function
- Propositional valuation
  - AF-models, 133
  - F-models, 58
  - Kripke models, 11
  - M-models, 46
- Propositionally closed branch, 211, 218

- R4<sup>!</sup>, justification rule, 28
- R4<sub>CS</sub><sup>!</sup>, justification rule, 31
- R4, justification rule, 28
- R4<sub>CS</sub>, EBK rule, 128
- R4<sub>CS</sub>, justification rule, 30
- Realization Theorem, 26–27, 36, 45
- Reflected fragment
  - EBK logics, 138
    - axioms and rules, 138
  - justification logics, 117
    - axioms and rules, 117–118
- Reflexivity Axiom T, 5
- Reflexivity Axiom T<sub>i</sub>, 6, 128
- Refutability
  - F-models, 61
    - in a class of Kripke frames, 13
    - in a class of Kripke models, 13
- de Rijke, Maarten, 8, 15, 146, 147, 150
- Rubtsova, Natalia, 1, 2, 36, 44, 45, 63, 78, 115, 116, 142
- S4LP<sub>CS</sub>-WORLD, 240
- SAT, 16
- Satisfiability
  - F-models, 61
    - in a class of Kripke frames, 13
    - in a class of Kripke models, 13
    - in a Kripke frame, 13
    - in a Kripke model, 12
    - in an F-model, 61
- Satisfiability problem, 16
  - CI, 16
- Self-referential constant, directly, 265
- Seriality Axiom, 6
- Shapiro, D., 1, 44
- Sidon, Tatiana, *see* Yavorskaya, Tatiana
- Single Step Tableaux, 8

- Size
  - formulas, 23, 127
  - terms, 23, 127
- $SLet$ , 5
- Statman, Richard, 16
- Strong Evidence property, 60, 116
- STT, 8
- $Sub(F)$ , 165
- $Sub^\neg(F)$ , 165
- Subformula set, 165
- $Sub_n^\square(F)$ , 165
- Suboccurrence, 201
- Substitution Property
  - EBK logics, 131
  - justification logics, 42
  - with renaming of constants
    - EBK logics, 132
    - justification logics, 43
- Successful call, 242
- Sum Rule \*A3, 103
- Švejdar, Vítězslav, 233
- T, modal axiom, 5
- Tableau rules
  - $\square$ , 8
  - $\diamond$ , 8
  - 4, 9
  - D, 9
  - T, 9
- $\mathcal{TCS}$ , 31, 129
- $T_i$ , EBK axioms, 128
- $T_i$ , modal axioms, 6
- $Tm$ , 22
- Total constant specification
  - EBK logic, 129
  - justification logic, 31
- Truth Lemma
  - F-models, 75
  - finitary models, 178
  - M-models, 53

- S4LP<sub>CS</sub>-WORLD, 246
- Truth prefix, 217
- Undecidability
  - EBK logics, 190
  - justification logics, 190
- Validity
  - in a class of Kripke frames, 13
  - in a class of Kripke models, 13
  - in a Kripke frame, 13
  - in a Kripke model, 12
  - in an AF-model, 134
  - in an F-model, 61
  - w.r.t. AF-models, 134
  - w.r.t. F-models, 61
- Validity problem, 15
  - Cl, 16
  - Int, 16
  - J4, 231
  - J4<sub>CS</sub>
    - schematic *CS*, 231
- LP<sub>CS</sub>
  - schematically injective *CS*, 231
- S4, 16
- S4LP<sub>CS</sub>
  - schematic *CS*, 237
- S4<sub>n</sub>, 16
- S5, 16
- S5<sub>n</sub>,  $n \geq 2$ , 16
- T, 16
- T<sub>n</sub>, 16
- Valuation
  - finitely true, 151
- Vardi, Moshe, 8, 15, 147, 148
- Venema, Yde, 8, 15, 146, 147, 150
- Wansing, Heinrich, 10, 283
- Weak model, *see* F-model
- Wise Men Puzzle, 85
- Yavorskaya, Tatiana, 140, 192

Yavorsky, Rostislav, 46

Zakharyashev, Michael, 8, 15, 144,

145