Lower Complexity Bounds in Justification Logic

Samuel R. Buss[a,1], Roman Kuznets[*,b,2]

[a] *Department of Mathematics, University of California, San Diego*
La Jolla, CA 92093-0112, USA
[b] *Institut für Informatik und angewandte Mathematik, Universität Bern*
Neubrückstrasse 10, CH-3012 Bern, Switzerland

Abstract

Justification Logic studies epistemic and provability phenomena by introducing justifications/proofs into the language in the form of justification terms. Pure justification logics serve as counterparts of traditional modal epistemic logics, and hybrid logics combine epistemic modalities with justification terms. The computational complexity of pure justification logics is typically lower than that of the corresponding modal logics. Moreover, the so-called reflected fragments, which still contain complete information about the respective justification logics, are known to be in NP for a wide range of justification logics, pure and hybrid alike. This paper shows that, under reasonable additional restrictions, these reflected fragments are NP-complete, thereby proving a matching lower bound. The proof method is then extended to provide a uniform proof that the corresponding full pure justification logics are Π_2^p-hard, reproving and generalizing an earlier result by Milnikel.

1. Introduction

Justification Logic is an emerging field that studies provability, knowledge, and belief via explicit proofs or justifications that are part of the language. A justification logic is essentially a refined analogue of a modal epistemic logic. Whereas the latter uses $\Box F$ to indicate that F is known to be true, a justification logic uses $t : F$ instead, where t is a term that describes a 'justification' or proof of F. This construction enables justification logics to reason about both formulas and proofs at the same time, avoiding the need to treat provability at the metalevel.

Because Justification Logic can reason directly about explicit proofs, it provides more concrete and constructive analogues of modal epistemic logics. For example, the modal distribution axiom $\Box(F \to G) \to (\Box F \to \Box G)$ is replaced in Justification Logic by the axiom $s : (F \to G) \to (t : F \to (s \cdot t) : G)$. The latter replaces the distribution axiom with a computationally explicit construction. Justification logics are very

[*] Corresponding author

Email addresses: `sbuss@ucsd.edu` (Samuel R. Buss), `kuznets@iam.unibe.ch` (Roman Kuznets)

[1] Partially supported by NSF grant DMS–0700533.

[2] Supported by Swiss National Science Foundation grant 200021–117699. The initial stages of the research were partially supported by a CUNY Graduate Center Research Grant for Doctoral Students.

promising for structural proof theory and have already proved to be fruitful in finding new approaches to common knowledge ([3]), the Logical Omniscience Problem ([6]), and self-referentiality of proofs ([19]). For further discussion on the various applications of Justification Logic, see [5].

The goal of the present paper[3] is to provide a uniform method of proving lower bounds for the Derivability Problems in various justification logics and their reflected fragments by reduction from problems similar to the Vertex Cover Problem. We begin by reviewing some definitions of justification logics.

The historically first justification logic, the Logic of Proofs LP, was introduced by Sergei Artemov [1] to provide a provability semantics for the modal logic $\mathsf{S4}$ (see also [2]). The language of LP

$$F ::= p \mid \bot \mid (F \to F) \mid t{:}F \ ,$$
$$t ::= x \mid c \mid (t \cdot t) \mid (t + t) \mid {!}\,t$$

contains an additional operator $t : F$, read 'term t serves as a justification/proof of formula F.' Here p stands for a sentence letter, x for a justification variable, and c for a justification constant. Formulas of the form $t{:}F$ are called *justification assertions*.

Statements $t : F$ can be seen as refinements of modal statements $\Box F$ because the latter say that F is known, whereas the former additionally provide a rationale for such knowledge. This relationship is demonstrated through the recursively defined operation of *forgetful projection* that maps justification formulas to modal formulas: $(t : F)^\circ = \Box(F^\circ)$, and commutes with Boolean connectives: $(F \to G)^\circ = F^\circ \to G^\circ$, where $p^\circ = p$ and $\bot^\circ = \bot$.

Axioms and rules of LP:

A1. A complete axiomatization of classical propositional logic by finitely many axiom schemes; rule modus ponens;

A2. *Application Axiom* $\quad s{:}(F \to G) \to (t{:}F \to (s \cdot t){:}G)$;

A3. *Monotonicity Axiom* $\quad s{:}F \to (s + t){:}F, \qquad t{:}F \to (s + t){:}F$;

A4. *Factivity Axiom* $\quad t{:}F \to F$;

A5. *Positive Introspection Axiom* $\quad t{:}F \to {!}\,t{:}t{:}F$;

R4. *Axiom Internalization Rule* $\quad \dfrac{}{c{:}A}$,

where A is an axiom of LP and c is a justification constant.

LP is the exact counterpart of $\mathsf{S4}$ (note the similarity of their axioms): namely, let $X^\circ = \{F^\circ \mid F \in X\}$ for a set X of justification formulas and let LP be identified with the set of its theorems, then

Theorem 1 (Realization Theorem, [1, 2])**.** $\mathsf{LP}^\circ = \mathsf{S4}$.

[3] An earlier version of this paper appeared in the proceedings of LFCS 2009 ([10]).

Table 1: Axioms for Justification Logics

Justification axiom scheme	Present in logics
A4. $t{:}F \to F$	JT, LP
A5. $t{:}F \to\, !t{:}t{:}F$	J4, JD4, LP
A7. $t{:}\bot \to \bot$	JD, JD4

Other epistemic modal logics have their own justification counterparts in the same sense. Counterparts of the modal logics K, D, T, K4, and D4 were developed by Vladimir Brezhnev in [9]. These justification logics, named J, JD, JT, J4, and JD4 respectively, are all subsystems of LP and share the A1–A3 portion of its axiom system. The remaining two axiom schemes are included dependent on whether or not their forgetful projections are axioms of the respective modal logic. In addition, JD and JD4 require a new axiom scheme:[4]

A7. *Consistency* $t{:}\bot \to \bot$,

whose forgetful projection is the modal Seriality Axiom. Complete details can be found in Table 1.

Finally, the rule R4 for J4 and JD4 is written the same way as for LP, but of course it now applies to the axioms of J4, respectively JD4. The logics without the Positive Introspection Axiom A5 still require some restricted form of positive introspection for constants which is embedded into the Axiom Internalization Rule:

R4$^!$. *Axiom Internalization Rule* $$\overline{\underbrace{!\,!\cdots!}_{n}c{:}\ldots{:}!\,!c{:}!c{:}c{:}A},$$

where A is an axiom of the logic, c is a justification constant, and $n \geq 0$ is an integer.

This form of the Axiom Internalization Rule is used for J, JD, and JT.

Theorem 2 (Realization Theorem, [9])**.**

$$\mathsf{J}^\circ = \mathsf{K}, \qquad \mathsf{JD}^\circ = \mathsf{D}, \qquad \mathsf{JT}^\circ = \mathsf{T},$$
$$\mathsf{J4}^\circ = \mathsf{K4}, \qquad \mathsf{JD4}^\circ = \mathsf{D4}.$$

All these justification logics are *pure* in the sense that only terms are present in the language, but not modalities. In [3], Artemov studied *hybrid* justification logics $\mathsf{T}_n\mathsf{LP}$, $\mathsf{S4}_n\mathsf{LP}$, and $\mathsf{S5}_n\mathsf{LP}$. These combine terms with modalities for several agents (a single-agent variant $\mathsf{S4}_1\mathsf{LP}$ was originally developed by Artemov jointly with Elena Nogina, see [7]).

Axioms and rules of $\mathsf{T}_n\mathsf{LP}$, $\mathsf{S4}_n\mathsf{LP}$, and $\mathsf{S5}_n\mathsf{LP}$:
Let $\mathsf{ML} \in \{\mathsf{T}, \mathsf{S4}, \mathsf{S5}\}$.

[4] The apparent break in the numeration of axioms is due to the Negative Introspection Axiom A6 that remains outside the scope of this paper. The numbering of rules follows [4].

1. Axioms and rules of the multimodal logic ML_n.

2. Axioms and rules of the justification logic LP.

3. *Connection axiom.* For each $i = 1, \dots, n$, $$t{:}F \to \Box_i F.$$

The Axiom Internalization Rule R4 in 2. is extended to apply to all axioms of $\mathsf{ML}_n\mathsf{LP}$.

For some applications (e.g., to avoid Logical Omniscience [6] or to study self-referentiality [19]) the use of constants needs to be restricted; this is achieved using *constant specifications*. A *constant specification* $\mathcal{CS}$ for a justification logic JL is a set of instances of the rule R4 for this logic:

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

Given a constant specification $\mathcal{CS}$ for JL, the logic $\mathsf{JL}_{\mathcal{CS}}$ is the result of replacing the Axiom Internalization Rule in JL (R4 or R4$^!$) by its relativized version, respectively by:

R4$_{\mathcal{CS}}$. $$\frac{c{:}A \in \mathcal{CS}}{c{:}A};$$

R4$^!_{\mathcal{CS}}$. $$\frac{c{:}A \in \mathcal{CS}}{\underbrace{!\,!\cdots!}_{n}c{:}\ldots{:}!\,!\,c{:}!\,c{:}c{:}A},$$

where $n \geq 0$ is an integer.

The Realization Theorem holds for a pure justification logic JL with a constant specification $\mathcal{CS}$, i.e., $(\mathsf{JL}_{\mathcal{CS}})^\circ = \mathsf{ML} = \mathsf{JL}^\circ$, iff $\mathcal{CS}$ is *axiomatically appropriate*:

Definition 3. A constant specification $\mathcal{CS}$ for a logic JL is called:

- *axiomatically appropriate*[5] if every axiom of JL is justified by at least one constant;
- *schematic*[6] if each constant justifies several (maybe 0) axiom schemes and only them;
- *schematically injective*[7] if it is schematic and each constant justifies no more than one axiom scheme.

The following is the fundamental property of justification logics, closely related to the Realization Theorem:

Lemma 4 (Constructive Necessitation. [1, 3, 4])**.** *Let $\mathcal{CS}$ be an axiomatically appropriate constant specification for a justification logic* JL*. For any theorem F of* $\mathsf{JL}_{\mathcal{CS}}$*, there exists a $+$-free ground*[8] *justification term s such that* $\mathsf{JL}_{\mathcal{CS}} \vdash s{:}F$.

[5] The term is due to Melvin Fitting.

[6] The term is due to Robert Milnikel although the idea goes back to Alexey Mkrtychev.

[7] The term is due to Milnikel.

[8] A justification term is called *ground* if it contains no occurrences of justification variables.

Whereas it is well known that the Derivability Problems for the modal logics K, D, T, $\mathsf{K4}$, $\mathsf{D4}$, and $\mathsf{S4}$ are PSPACE-complete ([20]), it was shown that

Theorem 5 ([15, 18])**.** *Let* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{LP}\}$ *and* $\mathcal{CS}$ *be an axiomatically appropriate constant specification*[9] *for* JL*. Then the Derivability Problem for* $\mathsf{JL}_{\mathcal{CS}}$ *is in* Π_2^p*.*

In particular, LP itself is in Π_2^p. (The complexity of $\mathsf{JD4}_{\mathcal{CS}}$ remains open, see [18].) Robert Milnikel proved some matching lower bounds, namely:

Theorem 6 ([21])**.**

1. $\mathsf{LP}_{\mathcal{CS}}$ *is* Π_2^p*-hard provided* $\mathcal{CS}$ *is axiomatically appropriate and schematically injective;*
2. $\mathsf{J4}_{\mathcal{CS}}$ *is* Π_2^p*-hard provided* $\mathcal{CS}$ *is axiomatically appropriate and schematic.*

The so-called *reflected fragment* rLP of the Logic of Proofs was studied by Nikolai Krupski in [14].

Definition 7. For any justification logic $\mathsf{JL}_{\mathcal{CS}}$ with a constant specification $\mathcal{CS}$, the *reflected fragment* of the logic consists of all provable justification assertions:

$$\mathsf{rJL}_{\mathcal{CS}} = \{t{:}F \mid \mathsf{JL}_{\mathcal{CS}} \vdash t{:}F\} \ .$$

We will write $\mathsf{rJL}_{\mathcal{CS}} \vdash t : F$ to mean $t : F \in \mathsf{rJL}_{\mathcal{CS}}$. At the end of this section, we will present an axiomatization for several reflected fragments via $*$-calculi, which would make the use of $\vdash$ more natural.

A reflected fragment bears complete information about the underlying logic as the following theorem shows:

Theorem 8 ([14, 17])**.** *Let* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}, \mathsf{T}_n\mathsf{LP}, \mathsf{S4}_n\mathsf{LP}, \mathsf{S5}_n\mathsf{LP}\}$ *and* $\mathcal{CS}$ *be an axiomatically appropriate constant specification for* JL*. Then*

$$\mathsf{JL}_{\mathcal{CS}} \vdash F \qquad \Longleftrightarrow \qquad (\exists t)\mathsf{rJL}_{\mathcal{CS}} \vdash t{:}F \ .$$

(The requirement of axiomatic appropriateness is necessary only for the $\Longrightarrow$*-direction.)*

The $\Longrightarrow$-direction constitutes the Constructive Necessitation Property (Lemma 4). The $\Longleftarrow$-direction easily follows from the Factivity Axiom A4 for all logics but J, JD, $\mathsf{J4}$, and $\mathsf{JD4}$ that do not have Factivity. For these four logics, the statement can be proved semantically using F-models (see [12] for their description) or syntactically by transforming a derivation of $t : F$ in the respective $*$-calculus into a derivation of F in the underlying justification logic (the details of this transformation can be found in [17, proof of Lemma 3.4.10]).

Theorem 9 ([14, 17])**.** *Let* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}, \mathsf{T}_n\mathsf{LP}, \mathsf{S4}_n\mathsf{LP}, \mathsf{S5}_n\mathsf{LP}\}$ *and* $\mathcal{CS}$ *be a schematic constant specification for* JL*. The Derivability Problem for* $\mathsf{rJL}_{\mathcal{CS}}$*, the reflected fragment of* $\mathsf{JL}_{\mathcal{CS}}$*, is in NP.*

[9] In all complexity results, we always assume $\mathcal{CS}$ to be polynomial-time decidable.

Table 2: $*$-Calculi

Calculus	Axioms and rules	Used for
$*_{\mathcal{CS}}$	$*\mathcal{CS}^!$, $*$A2, $*$A3	$\mathsf{rJ}_{\mathcal{CS}}$, $\mathsf{rJD}_{\mathcal{CS}}$, $\mathsf{rJT}_{\mathcal{CS}}$
$*!_{\mathcal{CS}}$	$*\mathcal{CS}$, $*$A2, $*$A3, $*$A5	$\mathsf{rJ4}_{\mathcal{CS}}$, $\mathsf{rJD4}_{\mathcal{CS}}$, $\mathsf{rLP}_{\mathcal{CS}}$, $\mathsf{rT}_n\mathsf{LP}_{\mathcal{CS}}$, $\mathsf{rS4}_n\mathsf{LP}_{\mathcal{CS}}$, $\mathsf{rS5}_n\mathsf{LP}_{\mathcal{CS}}$

To prove Theorem 9 for $\mathsf{rLP}_{\mathcal{CS}}$, N. Krupski developed an axiomatization for $\mathsf{rLP}_{\mathcal{CS}}$ that we will call the *$*!_{\mathcal{CS}}$-calculus*.

Axioms and rules of the $*!_{\mathcal{CS}}$-calculus:

$*\mathcal{CS}$. *Axioms*: for any $c : A \in \mathcal{CS}$, $\qquad c : A;$

$*$A2. *Application Rule* $\qquad \dfrac{s : (F \to G) \qquad t : F}{s \cdot t : G};$

$*$A3. *Sum Rule* $\qquad \dfrac{s : F}{s + t : F}, \qquad \dfrac{t : F}{s + t : F};$

$*$A5. *Positive Introspection Rule* $\qquad \dfrac{t : F}{!\,t : t : F}.$

In [17], this calculus was shown to also axiomatize the logics $\mathsf{rJ4}_{\mathcal{CS}}$, $\mathsf{rJD4}_{\mathcal{CS}}$, $\mathsf{rT}_n\mathsf{LP}_{\mathcal{CS}}$, $\mathsf{rS4}_n\mathsf{LP}_{\mathcal{CS}}$, and $\mathsf{rS5}_n\mathsf{LP}_{\mathcal{CS}}$. In particular, the three logics $\mathsf{LP}_{\mathcal{CS}}$, $\mathsf{J4}_{\mathcal{CS}}$, and $\mathsf{JD4}_{\mathcal{CS}}$ all use the $*!_{\mathcal{CS}}$-calculus to axiomatize their reflected fragments. The reflected fragments $\mathsf{rJ}_{\mathcal{CS}}$, $\mathsf{rJD}_{\mathcal{CS}}$, and $\mathsf{rJT}_{\mathcal{CS}}$ of the three theories which do not have positive introspection are all axiomatized by the *$*_{\mathcal{CS}}$-calculus* which is obtained by omitting the rule $*$A5 from the $*!_{\mathcal{CS}}$-calculus while simultaneously extending the set of axioms to include:

$*\mathcal{CS}^!$. *Axioms*: for any $c : A \in \mathcal{CS}$ and any integer $n \geq 0$, $\qquad \underbrace{!\,!\cdots!}_{n}\, c : \ldots : !\,!\, c : !\, c : c : A.$

Note that axioms $*\mathcal{CS}$ are instances of $*\mathcal{CS}^!$ with $n = 0$. Therefore, $*\mathcal{CS}$ can be used both in the $*_{\mathcal{CS}}$- and the $*!_{\mathcal{CS}}$-calculi.

We collectively call the $*_{\mathcal{CS}}$- and the $*!_{\mathcal{CS}}$-calculi the *$*$-calculi*, which are summarized in Table 2. As can be seen from the preceding discussion and the summarizing table, there are only two calculi that axiomatize the reflected fragments of various pure and hybrid justification logics. More precisely, the rules of the $*$-calculus for a given justification logic $\mathsf{JL}_{\mathcal{CS}}$ depend solely on whether JL enjoys full positive introspection while the axioms of this $*$-calculus are read from $\mathcal{CS}$ and thus indirectly depend on the axioms of JL.

In Theorem 36 below, we will show that the same rules can be used in the setting where there are non-logical axioms in addition to the $*\mathcal{CS}$ or $*\mathcal{CS}^!$ axioms.

The first main result of the present paper, Theorem 32, is a lower bound on the complexity of reflected fragments that matches the upper bound of Theorem 9; namely, we show that the Derivability Problems for many reflected fragments are NP-complete. The proof is by a many-one polynomial-time reduction from a known NP-complete problem, the Vertex Cover Problem. As in Milnikel's lower bound for $\mathsf{LP}_{\mathcal{CS}}$, we have to impose an additional restriction that $\mathcal{CS}$ be axiomatically appropriate and schematically injective. The reduction method is then extended to establish a lower bound on the complexity of full pure justification logics that also matches the upper bound of Theorem 5; this gives a reproof of the Π_2^p-hardness results of [21] and extends the results to additional justification logics.

The paper is structured as follows. Section 2 defines a coding of graphs by propositional formulas and shows how the existence of a vertex cover can be described in terms of these formulas. Section 3 develops justification terms that encode several standard methods of propositional reasoning. Although the formulas that describe the existence of a vertex cover depend on the cover itself rather than only on its size, Sect. 4 shows how to eliminate this dependency by using the terms from Sect. 3 to encode particular derivations of the formulas from Sect. 2. Section 5 finishes the proof of the polynomial-time reduction. This reduction is used in Sect. 6 to establish a criterion for NP-hardness of reflected fragments and to apply it to a wide range of them. Section 7 lays the groundwork for proving lower bounds for full pure justification logics, which is done in Sect. 8 by generalizing the Vertex Cover Problem to a Π_2^p-complete version. Finally, Sect. 9 explores the restrictions on the constant specification necessary for the proved lower bounds.

2. Graph Coding and Preliminaries

A graph $G = \langle V, E\rangle$ has a finite set V of vertices and a finite set E of undirected edges. We assume w.l.o.g. that $V = \{1, \dots, N\}$ for some N and represent an edge e between vertices k and l as the set $e = \{k, l\}$ with its endpoints denoted by $v_1(e) < v_2(e)$. A vertex cover for G is a set C of vertices such that each edge $e \in E$ has at least one endpoint in C. The Vertex Cover (VC) Problem is the problem of determining whether a given graph G has a vertex cover of a size $\leq L$ for a given integer $L \geq 0$. The Vertex Cover Problem is one of the classic NP-complete problems.

We define below formulas F_V, F_C, and F_G that will help build a many-one reduction from VC to the reflected fragments of justification logics. These formulas will include large conjunctions. To avoid the dependence of derivations on a vertex cover, we will use balanced conjunctions (see [8]):

Definition 10. Each formula is a *balanced conjunction of depth* 0. If A and B are both balanced conjunctions of depth k, then $A \wedge B$ is a *balanced conjunction of depth* $k + 1$.

Clearly, a balanced conjunction of depth k is also a balanced conjunction of depth l for any $0 \leq l \leq k$. Thus, we are mainly interested in how deeply a given formula is conjunctively balanced. Unless stated otherwise, for any conjunction $C_1 \wedge \dots \wedge C_{2^k}$ of 2^k formulas, we assume that the omitted parentheses are such that the resulting balanced conjunction has the maximal possible depth, i.e., depth $\geq k$.

We also need to refer to C_i's that form a conjunction $C_1 \wedge \dots \wedge C_{2^k}$. The following inductive definition of *depth k conjuncts*, or simply *k-conjuncts*, generalizes the definition of *conjuncts* in an ordinary conjunction:

Definition 11. Each formula is a 0-*conjunct* of itself. If $C \wedge D$ is a k-conjunct of a formula F, then C and D are both $(k+1)$-*conjuncts* of F.

For instance, the conjuncts of an ordinary conjunction are its 1-conjuncts; all C_i's in $C_1 \wedge \cdots \wedge C_{2^k}$ are its k-conjuncts. More generally, any balanced conjunction of depth k has exactly 2^k occurrences of k-conjuncts (with possibly several occurrences of the same formula).

To make a full use of balanced conjunctions, it is convenient to restrict attention to instances of the Vertex Cover Problem for graphs in which both the number of vertices and the number of edges are powers of 2. These are called *binary exponential graphs*. It is also helpful to only consider vertex covers whose size is a power of 2; these we call *binary exponential vertex covers*. Fortunately, the version of the Vertex Cover Problem restricted to binary exponential graphs and their binary exponential vertex covers is also NP-complete:

Theorem 12. *The Binary Vertex Cover (BVC) Problem of determining whether a given binary exponential graph G has a vertex cover of size* $\leq 2^l$ *for a given integer* $l \geq 0$ *is NP-complete.*

Proof. Since each instance of BVC is also an instance of the standard VC problem, and since VC is NP-complete, it suffices to construct a polynomial-time many-one reduction from VC to BVC. Suppose we are given an instance of VC; namely, we are given a graph G_0 and an integer L and wish to determine if G_0 has a vertex cover of size $\leq L$. We give a polynomial-time procedure that constructs a binary exponential graph G and a value l so that G_0 has a vertex cover of size $\leq L$ iff G has a vertex cover of size $\leq 2^l$. The graph G is constructed in three stages; each stage causes only a constant factor increase in the size of the graph.

Stage 1. Increasing the size of vertex covers. Choose an integer $0 \leq L' < L$ such that $L + L' = 2^l - 1$ for some integer $l \geq 0$. A graph $G' = \langle V', E' \rangle$ is obtained from G_0 by adding $2L'$ new vertices broken into L' disjoint pairs with the vertices in each pair joined by a new edge (L' new edges overall). The graph G_0 has a vertex cover of size $\leq L$ iff the graph G' has a vertex cover of size $\leq 2^l - 1$.

Stage 2. Increasing the number of edges. Choose an integer $0 < M'' \leq |E'|$ such that $|E'| + M'' = 2^m$ for some integer $m \geq 0$. A graph $G'' = \langle V'', E'' \rangle$ is obtained by adding $M'' + 1$ new vertices to G' with one of these vertices joined to all M'' others (M'' new edges overall). The graph G' has a vertex cover of size $\leq 2^l - 1$ iff the graph G'' has a vertex cover of size $\leq 2^l$.

Stage 3. Increasing the number of vertices. Choose an integer $0 \leq N''' < |V''|$ such that $|V''| + N''' = 2^k$ for some integer $k \geq 0$. A graph $G = G'''$ is obtained by adding N''' isolated vertices to G''. The graph G'' has a vertex cover of size $\leq 2^l$ iff the graph G''' has a vertex cover of size $\leq 2^l$.

It is clear from the construction that G is a binary exponential graph such that G_0 has a vertex cover of size $\leq L$ iff G has a vertex cover of size $\leq 2^l$. □

Definition 13. Let $G = \langle V, E \rangle$ be a binary exponential graph with $E = \{e_1, \ldots, e_{2^m}\}$. We define the following formulas:

a. For each edge $e_i = \{i_1, i_2\} \in E$, where $i_1 < i_2$, $F_e = p_{i_1} \vee p_{i_2} = p_{v_1(e)} \vee p_{v_2(e)}$.

b. Let $C = \{i_1, i_2, \dots, i_{2^l}\} \subseteq V$ be a possible binary exponential vertex cover for G, where $i_1 < i_2 < \dots < i_{2^l}$. Define $F_C = p_{i_1} \wedge \dots \wedge p_{i_{2^l}}$.

c. $F_G = F_{e_1} \wedge \dots \wedge F_{e_{2^m}}$.

The proof of the following properties is an easy exercise ($\vdash$ denotes derivability in classical propositional logic):

Lemma 14. *For any binary exponential graph $G = \langle V, E\rangle$ and any binary exponential set $C \subseteq V$,*
1. $\vdash F_V \to F_G$ *;*
2. $\vdash F_V \to F_C$ *;*
3. $\vdash F_C \to F_G$ *iff* *C is a vertex cover for G.*

Our goal is to reduce BVC to derivability in a given reflected fragment. To this end, we consider a particular derivation of $F_V \to F_G$ that proceeds by first proving $F_V \to F_C$, then attempting to prove $F_C \to F_G$, succeeding in the attempt iff C is a vertex cover, and finally applying hypothetical syllogism (HS) to infer $F_V \to F_G$. We further encode this derivation as a justification term t so that $\mathsf{rJL}_{\mathcal{CS}} \vdash t\!:\!(F_V \to F_G)$ iff C is a vertex cover. In BVC we need to determine whether there exists a vertex cover of (at most) a given size rather than whether a given set of vertices is a vertex cover. Thus, $t\!:\!(F_V \to F_G)$ should not depend on C but may (and should) depend on the size of C. Since C has already been "syllogized away" from the formula $F_V \to F_G$, it remains to make sure that the term t only depends on the size of C. Although any derivations of $F_V \to F_C$ and of $F_C \to F_G$ necessarily explicitly depend on C, the terms encoding them, and therefore t, can be made independent of C. This is the main reason why we use balanced conjunctions: this way all k-conjuncts are interchangeable.

Instead of giving a proof for one particular type of reflected fragments and explaining how to adjust it to other cases as in [10], we will now formulate conditions under which a reflected fragment *fits* our construction. These conditions have the following form: for certain individual axiom schemes or their sets there must exist a term that justifies exactly the axioms from this scheme or this set of schemes respectively.

Definition 15. A reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$ is called *fitting* if it has ground terms with the following properties:

$$\begin{array}{lll}
\mathsf{rJL}_{\mathcal{CS}} \vdash c_1 : F & \Longleftrightarrow & F \equiv (X \to (Y \to X)),\\
\mathsf{rJL}_{\mathcal{CS}} \vdash c_2 : F & \Longleftrightarrow & F \equiv ((X \to (Y \to Z)) \to ((X \to Y) \to (X \to Z))),\\
\mathsf{rJL}_{\mathcal{CS}} \vdash c_{\wedge 1, \wedge 2} : F & \Longleftrightarrow & F \equiv (X_1 \wedge X_2 \to X_i), \quad \text{where } i = 1 \text{ or } i = 2,\\
\mathsf{rJL}_{\mathcal{CS}} \vdash c_{\wedge} : F & \Longleftrightarrow & F \equiv (X \to (Y \to X \wedge Y)),\\
\mathsf{rJL}_{\mathcal{CS}} \vdash c_{\vee 1, \vee 2} : F & \Longleftrightarrow & F \equiv (X_i \to X_1 \vee X_2), \quad \text{where } i = 1 \text{ or } i = 2,
\end{array} \tag{1}$$

where X, Y, Z, X_1, and X_2 are arbitrary formulas.

Most natural schematically injective constant specifications for justification logics yield fitting reflected fragments. Note that terms c_1, c_2, and $c_\wedge$ should justify exactly one commonly used propositional axiom scheme each. In fact, if these axiom schemes are part of A1 for a particular justification logic JL and if $\mathcal{CS}$ is schematically injective, these terms may have an especially simple form: they can be constants justifying their respective axiom schemes. The two terms $c_{\wedge 1,\wedge 2}$ and $c_{\vee 1,\vee 2}$ should justify two commonly used axiom schemes each. In general, they can be modeled by the sums of terms corresponding to those axiom schemes. That is to say, $c_{\wedge 1,\wedge 2}$ can be defined to be $c_{\wedge 1} + c_{\wedge 2}$, where $c_{\wedge i}$ justifies exactly the scheme $X_1 \wedge X_2 \to X_i$. Similarly, $c_{\vee 1,\vee 2}$ can generally be set equal to $c_{\vee 1} + c_{\vee 2}$ for appropriate terms $c_{\vee 1}$ and $c_{\vee 2}$.

We shall prove the NP-hardness of fitting reflected fragments by giving a reduction from BVC to derivability in the reflected fragment. Therefore, our complexity lower bounds hold for any fitting reflected logic, and they do not depend on the particular propositional axiomatization chosen, or the particular form of the five terms from (1). In fact, as will be shown, it is not even important that the operation $+$ be present.

3. Justification Terms Encoding Propositional Reasoning

Throughout the section, we assume that a reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$ is fitting. All $*$-derivations in this and the next two sections can be performed in either of the $*$-calculi. In each case, the choice of the $*$-calculus is made based on the underlying reflected fragment according to Table 2.

The size of terms is defined in a standard way: $|c| = |x| = 1$ for any constant c and any variable x, $|(t \cdot s)| = |(t + s)| = |t| + |s| + 1$, $|\,!\,t| = |t| + 1$.

Note that all the terms from (1) have size $O(1)$ because there are only five of them.

Lemma 16 (Encoding the Hypothetical Syllogism Rule)**.** *The operation*

$$\mathrm{syl}(t, s) = (c_2 \cdot (c_1 \cdot s)) \cdot t$$

with $|\mathrm{syl}(t, s)| = |t| + |s| + O(1)$ *encodes the Hypothetical Syllogism Rule, i.e.,*

$$\mathsf{rJL}_{\mathcal{CS}} \vdash \mathrm{syl}(t, s){:}H \quad\Longleftrightarrow\quad \begin{array}{l} H = A \to C \textit{ such that for some } B \\ \mathsf{rJL}_{\mathcal{CS}} \vdash t{:}(A \to B) \qquad \textit{and} \qquad \mathsf{rJL}_{\mathcal{CS}} \vdash s{:}(B \to C). \end{array}$$

Proof. ($\Longleftarrow$). Here are the key elements of a derivation of $t : (A \to B), s : (B \to C) \vdash \mathrm{syl}(t, s) : (A \to C)$ (parts of the derivation following from the “fit” of the reflected fragment are omitted):

$$\begin{array}{rcll}
c_1 & : & ((B \to C) \to (A \to (B \to C))) & (\text{fit}) \\
s & : & (B \to C) & (\text{Hyp}) \\
c_1 \cdot s & : & (A \to (B \to C)) & ({*}\text{A2}) \\
c_2 & : & ((A \to (B \to C)) \to ((A \to B) \to (A \to C))) & (\text{fit}) \\
c_2 \cdot (c_1 \cdot s) & : & ((A \to B) \to (A \to C)) & ({*}\text{A2}) \\
t & : & (A \to B) & (\text{Hyp}) \\
(c_2 \cdot (c_1 \cdot s)) \cdot t & : & (A \to C) & ({*}\text{A2})
\end{array}$$

($\Longrightarrow$). Consider an arbitrary derivation of $\mathrm{syl}(t, s) : H$ in the $*$-calculus. It can easily be seen that any such derivation must have the same key elements as the one used for the $\Longleftarrow$-direction above: the only difference can be in the choice of formulas for the terms c_1, c_2, s, and t. Since the reflected fragment is fitting, we know which formulas can be proved by c_1 and c_2. Thus, we can shape this as a unification problem: find formulas X_1, Y_1, X_2, Y_2, Z_2, X_s, and X_t such that $\mathsf{rJL}_{\mathcal{CS}} \vdash s : X_s$, $\mathsf{rJL}_{\mathcal{CS}} \vdash t : X_t$, and the following is a $*$-calculus derivation of $s : X_s, t : X_t \vdash \mathrm{syl}(t, s) : H$ modulo derivability of statements from (1):

1.	$c_1 : (X_1 \to (Y_1 \to X_1))$	(fit)
2.	$s : X_s$	(Hyp)
3.	$c_1 \cdot s : (Y_1 \to X_1)$	($*$A2)
4.	$c_2 : ((X_2 \to (Y_2 \to Z_2)) \to ((X_2 \to Y_2) \to (X_2 \to Z_2)))$	(fit)
5.	$c_2 \cdot (c_1 \cdot s) : ((X_2 \to Y_2) \to (X_2 \to Z_2))$	($*$A2)
6.	$t : X_t$	(Hyp)
7.	$(c_2 \cdot (c_1 \cdot s)) \cdot t : H$	($*$A2)

To make the applications of the rule $*$A2 work in lines 3, 5, and 7, the unification variables have to satisfy the following equations:

$$X_1 = X_s \qquad \text{from 3.} \qquad (2)$$

$$X_2 \to (Y_2 \to Z_2) = Y_1 \to X_1 \qquad \text{from 5.} \qquad (3)$$

$$X_2 \to Y_2 = X_t \qquad \text{from 7.} \qquad (4)$$

$$X_2 \to Z_2 = H \qquad \text{from 7.} \qquad (5)$$

By (2) and (3), $X_s = X_1 = Y_2 \to Z_2$. This equation combined with (4) and (5) shows that H is indeed an implication that follows by HS from X_t and X_s justified by t and s respectively. □

Lemma 17 (Stripping k conjunctions)**.** *For any integer $k \geq 0$ there exists a term t_k of size $O(k)$ that encodes the operation of stripping k conjunctions, i.e.,*

$$\mathsf{rJL}_{\mathcal{CS}} \vdash t_k : D \qquad \Longleftrightarrow \qquad D = H \to C, \text{ where } C \text{ is a } k\text{-conjunct of } H.$$

Proof. We prove by induction on k that the conditions are satisfied for

$$t_0 = (c_2 \cdot c_1) \cdot c_1 \ ,$$
$$t_{k+1} = \mathrm{syl}(c_{\wedge 1, \wedge 2}, t_k) \ .$$

Since $|t_{k+1}| = |t_k| + |c_{\wedge 1,\wedge 2}| + O(1) = |t_k| + O(1)$, it is clear that $|t_k| = |t_0| + kO(1) = O(k)$.

Base case, $k = 0$. ($\Longleftarrow$). If C is a 0-conjunct of H, then $H = C$, and it is easy to see that t_0 corresponds to the standard derivation of the tautology $C \to C$ from propositional axioms (cf. combinator *skk*).
($\Longrightarrow$). Any $*$-derivation of $t_0 : D$ must have the following key elements:

1.	$c_2 : ((X_2 \to (Y_2 \to Z_2)) \to ((X_2 \to Y_2) \to (X_2 \to Z_2)))$	(fit)
2.	$c_1 : (X_1 \to (Y_1 \to X_1))$	(fit)
3.	$c_2 \cdot c_1 : ((X_2 \to Y_2) \to (X_2 \to Z_2))$	($*$A2)
4.	$c_1 : (X_3 \to (Y_3 \to X_3))$	(fit)
5.	$(c_2 \cdot c_1) \cdot c_1 : D$	($*$A2)

For $*$A2 from line 5 to be valid, it is necessary that $D = X_2 \to Z_2$. It follows from $*$A2 in line 3 that $X_2 \to (Y_2 \to Z_2) = X_1 \to (Y_1 \to X_1)$, in which case $X_2 = X_1 = Z_2$. Therefore, $D = X_2 \to X_2$, which is an implication from a formula to its 0-conjunct.

Induction step. ($\Longleftarrow$). Let H be a formula with a $(k+1)$-conjunct C. Then H must be of the form $H_1 \wedge H_2$ with C being a k-conjunct of H_i for some $i = 1, 2$. By the induction hypothesis, $\mathsf{rJL}_{CS} \vdash t_k : (H_i \to C)$ for this i. For both $i = 1$ and $i = 2$ $\mathsf{rJL}_{CS} \vdash (c_{\wedge 1, \wedge 2}) \mathbin{:} (H \to H_i)$. Then, by Lemma 16, $\mathsf{rJL}_{CS} \vdash t_{k+1} \mathbin{:} (H \to C)$.
($\Longrightarrow$). By the induction hypothesis, t_k justifies only implications from a formula to one of its k-conjuncts. Since rJL_{CS} is fitting, $c_{\wedge 1, \wedge 2}$ justifies only implications from a formula to one of its 1-conjuncts. By Lemma 16, t_{k+1} justifies only hypothetical syllogisms obtained from the latter and the former, but a k-conjunct of a 1-conjunct of a formula is its $(k+1)$-conjunct. □

Lemma 18. *For any term s and any integer $l \geq 0$ there exists a term $\mathrm{conj}(s, l)$ of size $O\left(|s| 2^l\right)$ with the following property:*

$$\mathsf{rJL}_{CS} \vdash \mathrm{conj}(s,l) \mathbin{:} D \qquad \Longleftrightarrow \qquad \begin{array}{l} D = B \to C_1 \wedge \cdots \wedge C_{2^l} \textit{ such that} \\ \mathsf{rJL}_{CS} \vdash s \mathbin{:} (B \to C_i) \textit{ for all } i = 1, \ldots, 2^l. \end{array}$$

Proof. We prove by induction on l that the conditions are satisfied for

$$\mathrm{conj}(s, 0) = \mathrm{syl}(s, t_0) \ ,$$
$$\mathrm{conj}(s, l+1) = \left(c_2 \cdot \mathrm{syl}(\mathrm{conj}(s,l), c_\wedge)\right) \cdot \mathrm{conj}(s, l) \ .$$

It is not hard to see that $|\mathrm{conj}(s,l)| = 2^l(|s| + K + L) - L$, where K and L are constants such that $|\mathrm{conj}(s, 0)| = |s| + K$ and $|\mathrm{conj}(s, l+1)| = 2|\mathrm{conj}(s,l)| + L$.

Base case, $l = 0$. ($\Longleftarrow$). For any formula C, $\mathsf{rJL}_{CS} \vdash t_0 \mathbin{:} (C \to C)$ by Lemma 17. Then, by Lemma 16, $\mathsf{rJL}_{CS} \vdash s \mathbin{:} (B \to C)$ implies $\mathsf{rJL}_{CS} \vdash \mathrm{syl}(s, t_0) \mathbin{:} (B \to C)$.
($\Longrightarrow$). By Lemma 16, $\mathrm{syl}(s, t_0)$ justifies only implications $B \to C$ for which there exists a formula A such that $\mathsf{rJL}_{CS} \vdash s \mathbin{:} (B \to A)$ and $\mathsf{rJL}_{CS} \vdash t_0 \mathbin{:} (A \to C)$. By Lemma 17, the latter implies $A = C$. Therefore, $\mathsf{rJL}_{CS} \vdash s \mathbin{:} (B \to C)$.[10]

Induction step. ($\Longleftarrow$). Let $H = C_1 \wedge \cdots \wedge C_{2^{l+1}}$ with $\mathsf{rJL}_{CS} \vdash s \mathbin{:} (B \to C_i)$ for all its $(l+1)$-conjuncts C_i. Then $H = H_1 \wedge H_2$, where $C_1, C_2, \ldots, C_{2^l}$ are l-conjuncts of H_1 and $C_{2^l+1}, C_{2^l+2}, \ldots, C_{2^{l+1}}$ are l-conjuncts of H_2. By the induction hypothesis,

$$\mathsf{rJL}_{CS} \vdash \mathrm{conj}(s,l) \mathbin{:} (B \to H_1) \ , \tag{6}$$
$$\mathsf{rJL}_{CS} \vdash \mathrm{conj}(s,l) \mathbin{:} (B \to H_2) \ . \tag{7}$$

In addition, $\mathsf{rJL}_{CS} \vdash c_\wedge \mathbin{:} (H_1 \to (H_2 \to H_1 \wedge H_2))$; in other words,

$$\mathsf{rJL}_{CS} \vdash c_\wedge \mathbin{:} (H_1 \to (H_2 \to H)) \ . \tag{8}$$

From (8) and (6) by Lemma 16, for $s' = \mathrm{syl}(\mathrm{conj}(s,l), c_\wedge)$ we have

$$\mathsf{rJL}_{CS} \vdash s' \mathbin{:} (B \to (H_2 \to H)) \ .$$

[10] Note that, in general, $\mathrm{conj}(s, 0) = s$ does not satisfy the $\Longrightarrow$-direction.

Then, from (7) and $\mathsf{rJL}_{\mathcal{CS}} \vdash c_2 : ((B \to (H_2 \to H)) \to ((B \to H_2) \to (B \to H)))$:

$$\mathsf{rJL}_{\mathcal{CS}} \vdash c_2 \cdot s' : ((B \to H_2) \to (B \to H)) \qquad \text{and, finally,}$$
$$\mathsf{rJL}_{\mathcal{CS}} \vdash (c_2 \cdot s') \cdot \mathrm{conj}(s,l) : (B \to H) \enspace .$$

It remains to note that $\mathrm{conj}(s, l+1) = (c_2 \cdot s') \cdot \mathrm{conj}(s,l)$.
$(\Longrightarrow)$. By Lemma 16, the rule

$$\frac{t : (A \to B) \quad s : (B \to C)}{\mathrm{syl}(t,s) : (A \to C)}(\mathrm{Syl})$$

is admissible in both $*$-calculi. So any $*$-derivation of $\mathrm{conj}(s, l+1) : D$ must contain the following key elements (we have already incorporated the induction hypothesis about $\mathrm{conj}(s,l)$ as well as Lemma 16):

1.	$\mathrm{conj}(s,l) : (B \to C_1 \wedge C_2 \wedge \cdots \wedge C_{2^l})$	(IH)
2.	$c_\wedge : (X_\wedge \to (Y_\wedge \to X_\wedge \wedge Y_\wedge))$	(fit)
3.	$s' : (B \to (Y_\wedge \to X_\wedge \wedge Y_\wedge))$	(Syl)
4.	$c_2 : ((X_2 \to (Y_2 \to Z_2)) \to ((X_2 \to Y_2) \to (X_2 \to Z_2)))$	(fit)
5.	$c_2 \cdot s' : ((X_2 \to Y_2) \to (X_2 \to Z_2))$	($*$A2)
6.	$\mathrm{conj}(s,l) : (B' \to C_{2^l+1} \wedge C_{2^l+2} \wedge \cdots \wedge C_{2^{l+1}})$	(IH)
7.	$(c_2 \cdot s') \cdot \mathrm{conj}(s,l) : D$	($*$A2)

where $\mathsf{rJL}_{\mathcal{CS}} \vdash s : (B \to C_i)$ and $\mathsf{rJL}_{\mathcal{CS}} \vdash s : (B' \to C_{2^l+i})$ for $i = 1, \ldots, 2^l$. Let us collect all unification equations necessary for this to be a valid fragment of a $*$-derivation:

$$C_1 \wedge C_2 \wedge \cdots \wedge C_{2^l} = X_\wedge \qquad \text{from 3.} \qquad (9)$$
$$B \to (Y_\wedge \to X_\wedge \wedge Y_\wedge) = X_2 \to (Y_2 \to Z_2) \qquad \text{from 5.} \qquad (10)$$
$$B' \to C_{2^l+1} \wedge C_{2^l+2} \wedge \cdots \wedge C_{2^{l+1}} = X_2 \to Y_2 \qquad \text{from 7.} \qquad (11)$$
$$X_2 \to Z_2 = D \qquad \text{from 7.} \qquad (12)$$

By (10) and (11), $B = X_2 = B'$. Thus, $\mathsf{rJL}_{\mathcal{CS}} \vdash s : (B \to C_i)$ for $i = 1, \ldots, 2^{l+1}$. Also

$$Y_\wedge = Y_2 = C_{2^l+1} \wedge C_{2^l+2} \wedge \cdots \wedge C_{2^{l+1}} \enspace ,$$

again by (10) and (11). So, by (9) and (10),

$$Z_2 = X_\wedge \wedge Y_\wedge = (C_1 \wedge C_2 \wedge \cdots \wedge C_{2^l}) \wedge (C_{2^l+1} \wedge C_{2^l+2} \wedge \cdots \wedge C_{2^{l+1}}) \enspace .$$

By (12), D is indeed an implication from B to this balanced conjunction for all of whose $(l+1)$-conjuncts the term s justifies their entailment from B. □

Lemma 19. *For the term* $\mathrm{disj} = c_{\vee1,\vee2}$ *of size* $O(1)$,

$$\mathsf{rJL}_{\mathcal{CS}} \vdash \mathrm{disj} : D \qquad \Longleftrightarrow \qquad D = B \to H, \ \textit{where } B \textit{ is a disjunct of } H.$$

Proof. Easily follows from the fact that the reflected fragment is fitting. □

4. Reduction from Vertex Cover, Part I

We now use the justification terms from the previous section to build a polynomial-time many-one reduction from BVC to a fitting reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$.

Lemma 20. *Let a term of size* $O\left(k2^l\right)$ *be defined by*

$$t_{k\to l} = \mathrm{conj}(t_k, l) \ .$$

For any binary exponential graph $G = \langle V, E\rangle$ *with* $|V| = 2^k$ *and any set* $C \subseteq V$ *of size* 2^l,

$$\mathsf{rJL}_{\mathcal{CS}} \vdash t_{k\to l} : (F_V \to F_C) \ .$$

Proof. $|\mathrm{conj}(t_k, l)| = O\left(|t_k| 2^l\right) = O\left(k2^l\right)$.

All l-conjuncts p_i of F_C, where $i \in C$, must be k-conjuncts of F_V. Thus, for any of them by Lemma 17, $\mathsf{rJL}_{\mathcal{CS}} \vdash t_k : (F_V \to p_i)$. Now, by Lemma 18, we have $\mathsf{rJL}_{\mathcal{CS}} \vdash \mathrm{conj}(t_k, l) : (F_V \to F_C)$. □

Lemma 21. *Let a term of size* $O(l)$ *be defined by*

$$t_{l\to\mathrm{edge}} = \mathrm{syl}(t_l, \mathrm{disj}) \ .$$

For any binary exponential graph $G = \langle V, E\rangle$, *any set* $C \subseteq V$ *of size* 2^l, *and any edge* $e \in E$,

$$\mathsf{rJL}_{\mathcal{CS}} \vdash t_{l\to\mathrm{edge}} : (F_C \to F_e) \qquad \Longleftrightarrow \qquad e \text{ is covered by } C.$$

Proof. $|\mathrm{syl}(t_l, \mathrm{disj})| = |t_l| + |\mathrm{disj}| + O(1) = O(l) + O(1) = O(l)$.

($\Longleftarrow$). If $i \in e \cap C$ is the vertex in C that covers e, then p_i is a disjunct of F_e, so $\mathsf{rJL}_{\mathcal{CS}} \vdash \mathrm{disj} : (p_i \to F_e)$ by Lemma 19. But p_i is also an l-conjunct of F_C, so, by Lemma 17, $\mathsf{rJL}_{\mathcal{CS}} \vdash t_l : (F_C \to p_i)$. Finally, $\mathsf{rJL}_{\mathcal{CS}} \vdash \mathrm{syl}(t_l, \mathrm{disj}) : (F_C \to F_e)$ by Lemma 16.

($\Longrightarrow$). If C does not cover e, it is easy to see that $F_C \to F_e$ is not valid propositionally. All justification logics are conservative over classical propositional logic, therefore $\mathsf{JL}_{\mathcal{CS}} \nvdash F_C \to F_e$. By Theorem 8, $\mathsf{rJL}_{\mathcal{CS}} \nvdash s : (F_C \to F_e)$ for any term s. □

Lemma 22. *Let a term of size* $O\left(l2^m\right)$ *be defined by*

$$s_{l\to m} = \mathrm{conj}(t_{l\to\mathrm{edge}}, m) \ .$$

For any binary exponential graph $G = \langle V, E\rangle$ *with* $|E| = 2^m$ *and any set* $C \subseteq V$ *of size* 2^l,

$$\mathsf{rJL}_{\mathcal{CS}} \vdash s_{l\to m} : (F_C \to F_G) \qquad \Longleftrightarrow \qquad C \text{ is a vertex cover for } G.$$

Proof. $|\mathrm{conj}(t_{l\to\mathrm{edge}}, m)| = O\left(|t_{l\to\mathrm{edge}}| 2^m\right) = O\left(l2^m\right)$.

($\Longleftarrow$). If C is a vertex cover, then $\mathsf{rJL}_{\mathcal{CS}} \vdash t_{l\to\mathrm{edge}} : (F_C \to F_e)$ for all $e \in E$, by Lemma 21. All m-conjuncts of F_G are F_e's with $e \in E$. Hence, by Lemma 18, we have $\mathsf{rJL}_{\mathcal{CS}} \vdash \mathrm{conj}(t_{l\to\mathrm{edge}}, m) : (F_C \to F_G)$.

($\Longrightarrow$). If C is not a vertex cover, by Lemma 14.3, formula $F_C \to F_G$ is not valid propositionally. The same argument as in the previous lemma shows that for any term s $\mathsf{rJL}_{\mathcal{CS}} \nvdash s : (F_C \to F_G)$. □

Theorem 23. *Let a term of size $O\left(k2^l\right) + O\left(l2^m\right)$ be defined by*

$$t_{k\to l\to m} = \mathrm{syl}(t_{k\to l}, s_{l\to m}) \ .$$

For any binary exponential graph $G = \langle V, E\rangle$ with $|V| = 2^k$ and $|E| = 2^m$ and any integer $0 \le l \le k$,

$$G \textit{ has a vertex cover of size} \le 2^l \qquad \Longrightarrow \qquad \mathsf{rJL}_{\mathcal{CS}} \vdash t_{k\to l\to m} : (F_V \to F_G) \ .$$

Proof. $|\mathrm{syl}(t_{k\to l}, s_{l\to m})| = |t_{k\to l}| + |s_{l\to m}| + O(1) = O\left(k2^l\right) + O\left(l2^m\right)$.

By Lemma 20, $\mathsf{rJL}_{\mathcal{CS}} \vdash t_{k\to l} : (F_V \to F_C)$ for any set $C \subseteq V$ of size 2^l. If G has a vertex cover of size $\le 2^l$, it can be enlarged to a vertex cover of size 2^l. Let C be such a vertex cover of size 2^l. Then, by Lemma 22, $\mathsf{rJL}_{\mathcal{CS}} \vdash s_{l\to m} : (F_C \to F_G)$. Thus, by Lemma 16, $\mathsf{rJL}_{\mathcal{CS}} \vdash \mathrm{syl}(t_{k\to l}, s_{l\to m}) : (F_V \to F_G)$. □

Note that the term $t_{k\to l\to m}$ depends only on size 2^l of a vertex cover and on how many vertices and edges G has.

5. Reduction from Vertex Cover, Part II

To finish the polynomial-time reduction from BVC to any fitting reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$ it now remains to prove the other direction:

$$\mathsf{rJL}_{\mathcal{CS}} \vdash t_{k\to l\to m} : (F_V \to F_G) \qquad \Longrightarrow \qquad G \text{ has a vertex cover of size} \le 2^l.$$

Lemma 24 (Converse to Lemma 20)**.**

$$\mathsf{rJL}_{\mathcal{CS}} \vdash t_{k\to l} : H \qquad \Longrightarrow \qquad \begin{array}{l} H = B \to D, \\ \textit{where } D \textit{ is a balanced conjunction of depth} \ge l \\ \textit{whose all } l\textit{-conjuncts are } k\textit{-conjuncts of } B. \end{array}$$

Proof. By definition, $t_{k\to l} = \mathrm{conj}(t_k, l)$, so by Lemma 18, it justifies only implications $B \to C_1 \land \dots \land C_{2^l}$ with $\mathsf{rJL}_{\mathcal{CS}} \vdash t_k : (B \to C_i)$ for $i = 1, \dots, 2^l$. By Lemma 17, the term t_k justifies only implications from a formula to its k-conjuncts. □

Lemma 25 (Converse to Lemma 21)**.**

$$\mathsf{rJL}_{\mathcal{CS}} \vdash t_{l\to \mathrm{edge}} : H \qquad \Longrightarrow \qquad \begin{array}{l} H = B \to D_1 \lor D_2, \\ \textit{where either } D_1 \textit{ or } D_2 \textit{ is an } l\textit{-conjunct of } B. \end{array}$$

Proof. By definition, $t_{l\to \mathrm{edge}} = \mathrm{syl}(t_l, \mathrm{disj})$. By Lemma 16, H can only be an implication $B \to D$ such that $\mathsf{rJL}_{\mathcal{CS}} \vdash t_l : (B \to C)$ and $\mathsf{rJL}_{\mathcal{CS}} \vdash \mathrm{disj} : (C \to D)$ for some formula C. By Lemma 19, the latter statement implies that $D = D_1 \lor D_2$ with $C = D_i$ for some $i = 1, 2$. By Lemma 17, D_i is an l-conjunct of B. □

Lemma 26 (Converse to Lemma 22)**.**

$$\mathsf{rJL}_{\mathcal{CS}} \vdash s_{l\to m} : H \qquad \Longrightarrow \qquad \begin{array}{l} H = B \to (C_1 \lor D_1) \land \dots \land (C_{2^m} \lor D_{2^m}), \\ \textit{where either } C_i \textit{ or } D_i \textit{ is an } l\textit{-conjunct of } B \\ \textit{for each } i = 1, \dots, 2^m. \end{array}$$

Proof. By definition, $s_{l\to m} = \mathrm{conj}(t_{l\to\mathrm{edge}}, m)$. By Lemma 18, H must be an implication from some formula B to a balanced conjunction of depth $\geq m$ such that, for all its m-conjuncts F, $\mathsf{rJL}_{\mathcal{CS}} \vdash t_{l\to\mathrm{edge}}\!:\!(B \to F)$. By Lemma 25, each of these m-conjuncts must be a disjunction with one of its disjuncts being an l-conjunct of B. □

Theorem 27 (Converse to Theorem 23)**.**

$$\mathsf{rJL}_{\mathcal{CS}} \vdash t_{k\to l\to m}\!:\!H \quad\Longrightarrow\quad \begin{array}{l} H = B \to (C_1 \vee D_1) \wedge \cdots \wedge (C_{2^m} \vee D_{2^m}) \\ \textit{and there is a size} \leq 2^l \textit{ set } X \textit{ of } k\textit{-conjuncts of } B \\ \textit{with either } C_i \in X \textit{ or } D_i \in X \textit{ for each } i = 1, \ldots, 2^m. \end{array}$$

Proof. By definition, $t_{k\to l\to m} = \mathrm{syl}(t_{k\to l}, s_{l\to m})$. By Lemma 16, $H = B \to F$ with (a) $\mathsf{rJL}_{\mathcal{CS}} \vdash t_{k\to l}\!:\!(B \to Q)$, (b) $\mathsf{rJL}_{\mathcal{CS}} \vdash s_{l\to m}\!:\!(Q \to F)$ for some formula Q. From (a), by Lemma 24, Q must be a conjunction $Q_1 \wedge \cdots \wedge Q_{2^l}$ such that all its l-conjuncts Q_i are also k-conjuncts of B. So the set $X = \{Q_i \mid i = 1, \ldots, 2^l\}$ has size $\leq 2^l$ (because of possible repetitions) and consists of k-conjuncts of B. It now follows from (b), by Lemma 26, that $F = (C_1 \vee D_1) \wedge \cdots \wedge (C_{2^m} \vee D_{2^m})$ with either C_i or D_i being an l-conjunct of Q for each $i = 1, \ldots, 2^m$, i.e., with either $C_i \in X$ or $D_i \in X$ for each $i = 1, \ldots, 2^m$. □

Theorem 28. *For any binary exponential graph $G = \langle V, E\rangle$ with $|V| = 2^k$ and $|E| = 2^m$ and any integer $0 \leq l \leq k$,*

$$\mathsf{rJL}_{\mathcal{CS}} \vdash t_{k\to l\to m}\!:\!(F_V \to F_G) \qquad\Longleftrightarrow\qquad \textit{G has a vertex cover of size} \leq 2^l.$$

Proof. The $\Longleftarrow$-direction was proved in Theorem 23. We now prove the $\Longrightarrow$-direction. $F_V \to F_G$ already has the form prescribed by Theorem 27. The only k-conjuncts of F_V are the sentence letters $p_1, \ldots, p_{2^k}$. Therefore, there must exist a set X of $\leq 2^l$ of these sentence letters such that for each m-conjunct F_e of F_G at least one of the disjuncts of F_e, i.e., either $p_{v_1(e)}$ or $p_{v_2(e)}$, is in X. This literally means that G has a set of $\leq 2^l$ vertices that covers all the edges of G. □

6. Lower Bounds for Reflected Fragments

Theorem 29. *For any fitting reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$, derivability in $\mathsf{rJL}_{\mathcal{CS}}$ is NP-hard.*

Proof. It is easy to see that both F_V and F_G have size polynomial in the size of G. As for the term $t_{k\to l\to m}$, it was shown in Theorem 23 that $|t_{k\to l\to m}| = O\left(k2^l\right) + O\left(l2^m\right)$, which is polynomial in the size of G provided $l \leq k$ (BVC for $l > k$ is trivial). Thus, Theorem 28 shows that $\mathsf{rJL}_{\mathcal{CS}}$ is NP-hard. □

It is time now to reap the fruits of the preceding theorem by showing that a wide range of constant specifications produce fitting reflected fragments.

In the following proof, we need to perform operations on schemes of formulas rather than on individual formulas. Thus, it is convenient to represent axiom schemes using the Substitution Rule:

$$\frac{X}{X\sigma}\ ,$$

where σ is any substitution of formulas for sentence letters (see, for instance, formulation of classical propositional logic in [11, Sect. 1.3]). In justification logics, we additionally have to allow substitutions σ to replace justification variables with justification terms.

The Substitution Rule allows to make axiomatizations finite because each axiom scheme can be replaced by a single axiom A such that each of infinitely many instances of the axiom scheme is a substitution instance of A. Note that in general we cannot use this representation to define $\mathsf{JL}_{\mathcal{CS}}$ because $\mathcal{CS}$ need not be schematic. It is easy to see that

Lemma 30 (Substitution Property, [2, 3, 4])**.** *The Substitution Rule is admissible for a justification logic $\mathsf{JL}_{\mathcal{CS}}$, and hence for $\mathsf{rJL}_{\mathcal{CS}}$, iff $\mathcal{CS}$ is schematic.*

Strictly speaking, we presented axioms schemes (e.g., for LP) using variables over formulas and variables over terms, e.g., $t{:}F \to F$ is understood in the sense that it is an axiom for any term t and any formula F. Each axiom scheme written in this way can be easily converted to an axiom in the corresponding system with the Substitution Rule by replacing distinct variables over formulas by distinct sentence letters and distinct variables over terms by distinct justification variables, e.g., the scheme $t{:}F \to F$ becomes a formula $x : p \to p$. The latter will be called a *most general instance* (*mgi*) of the former (note that an mgi is not unique: the choice of sentence letters and justification variables plays no role).

By analogy with axiom schemes, a *scheme of formulas* with an mgi F is the set

$$\{F\sigma \mid \sigma \text{ is a substitution}\} \enspace .$$

Lemma 31. *Let* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}, \mathsf{T}_n\mathsf{LP}, \mathsf{S4}_n\mathsf{LP}, \mathsf{S5}_n\mathsf{LP}\}$. *Any schematically injective and axiomatically appropriate constant specification $\mathcal{CS}$ for JL yields a fitting reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$.*

Proof. All formulas that fit the five patterns from (1) can be broken into seven schemes of propositional tautologies with mgi's

$$p \to (q \to p) \enspace , \tag{13}$$

$$(p \to (q \to r)) \to ((p \to q) \to (p \to r)) \enspace , \tag{14}$$

$$p_1 \wedge p_2 \to p_1 \enspace , \tag{15}$$

$$p_1 \wedge p_2 \to p_2 \enspace , \tag{16}$$

$$p \to (q \to p \wedge q) \enspace , \tag{17}$$

$$p_1 \to p_1 \vee p_2 \enspace , \tag{18}$$

$$p_2 \to p_1 \vee p_2 \enspace , \tag{19}$$

where p, q, r, p_1, and p_2 are distinct sentence letters (strictly speaking, we should have used a distinct set of sentence letters for each mgi). Let A stand for any of these seven mgi's. For each JL, its axiomatization contains A1, i.e., a full axiomatization of classical propositional logic. Therefore, the propositional tautology A is a theorem of $\mathsf{JL}_{\mathcal{CS}}$. Since $\mathcal{CS}$ is axiomatically appropriate, by Lemma 4, there exists a term s, which contains neither + nor any justification variables, such that $\mathsf{JL}_{\mathcal{CS}} \vdash s : A$, and hence $\mathsf{rJL}_{\mathcal{CS}} \vdash s : A$. By the Substitution Property (Lemma 30), $\mathsf{rJL}_{\mathcal{CS}} \vdash A\sigma$ for any substitution σ. In other words, the term s satisfies (1) in the $\Longleftarrow$-direction for the respective axiom scheme.

For the $\Longrightarrow$-direction, it is sufficient to note that for any +-free ground term t the set

$$\mathcal{CS}(t) = \{F \mid \mathsf{rJL}_{\mathcal{CS}} \vdash t{:}F\}$$

is either empty or a scheme whose mgi we will denote by A_t. This statement can be proved by induction on the size of t. For constants, it is guaranteed by the schematic injectivity of $\mathcal{CS}$. Justification variables do not occur in ground terms. If $\mathcal{CS}(t')$ is empty, so is $\mathcal{CS}(!\,t')$. Otherwise, for logics with $*!_{\mathcal{CS}}$ as their $*$-calculus,

$$\mathcal{CS}(!\,t') = \{t'{:}F \mid F \in \mathcal{CS}(t')\} =_{IH} \\ \{t'{:}(A_{t'}\sigma) \mid \sigma \text{ is a substitution}\} = \{(t'{:}A_{t'})\sigma \mid \sigma \text{ is a substitution}\} \ . \quad (20)$$

The last equality follows from the fact that t' does not contain variables. Thus, in this case $A_{!\,t'} = t'{:}A_{t'}$. In the logics with $*_{\mathcal{CS}}$ as their $*$-calculus, the same argument works for $t' = \underbrace{!\cdots!}_{n}\,c$, $n \geq 0$, whereas for all other terms, $\mathcal{CS}(!\,t') = \emptyset$ independent of $\mathcal{CS}(t')$. Finally,

$$\mathcal{CS}(t_1 \cdot t_2) = \{G \mid (\exists F)(F \to G \in \mathcal{CS}(t_1) \text{ and } F \in \mathcal{CS}(t_2))\} =_{IH} \\ \{G \mid (\exists F)(\exists\sigma_1)(\exists\sigma_2)(F \to G = A_{t_1}\sigma_1 \text{ and } F = A_{t_2}\sigma_2)\} \ . \quad (21)$$

It follows from Theorem 8 that A_{t_1} cannot be a single sentence letter. Clearly, if the main connective of A_{t_1} is not an implication, then $\mathcal{CS}(t_1 \cdot t_2) = \emptyset$. It remains to consider the case $A_{t_1} = B \to C$. If B cannot be unified with A_{t_2}, then $\mathcal{CS}(t_1 \cdot t_2) = \emptyset$. Otherwise, there must exist a most common unifier (mgu) $\tau = \mathrm{mgu}(B, A_{t_2})$. Any formula F in (21) must be a substitution instance of both B and A_{t_2}. Hence there must exist a substitution σ such that $F = B\tau\sigma = A_{t_2}\tau\sigma$. Accordingly, any formula $G \in \mathcal{CS}(t_1 \cdot t_2)$ must have the form $C\tau\sigma$. It follows that $\mathcal{CS}(t_1 \cdot t_2) = \{C\tau\sigma \mid \sigma \text{ is a substitution}\}$. Thus, in this case, $A_{t_1 \cdot t_2} = C\tau = C\,\mathrm{mgu}(B, A_{t_2})$. Clearly, the operation +, which enables us to combine several different schemes, would have broken this pattern, but it does not occur in t.

Thus, our term s must justify some scheme, of which the formula A is an instance, i.e., $A = A_s\sigma$ for some substitution σ. It can be checked that, if $B\sigma = A$ and B is a tautology, then $B = A$. Hence, $A_s = A$ by Theorem 8.

Therefore, the ground term s justifies exactly the scheme with mgi A. This discussion shows that there exist terms c_1, c_2, and $c_\wedge$ that justify exactly the schemes with mgi's (13), (14), and (17) respectively, as well as terms $c_{\wedge 1}$, $c_{\wedge 2}$, $c_{\vee 1}$, and $c_{\vee 2}$ for the schemes with mgi's (15), (16), (18), and (19) respectively. It remains to note that the term $c_{\vee 1} + c_{\vee 2}$ satisfies all the requirements of $c_{\vee 1, \vee 2}$ and the term $c_{\wedge 1} + c_{\wedge 2}$ fits the role of $c_{\wedge 1, \wedge 2}$. □

Theorem 32. *Let $\mathcal{CS}$ be a schematically injective and axiomatically appropriate constant specification for* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}, \mathsf{T}_n\mathsf{LP}, \mathsf{S4}_n\mathsf{LP}, \mathsf{S5}_n\mathsf{LP}\}$. *Then derivability in* $\mathsf{rJL}_{\mathcal{CS}}$ *is NP-complete.*

Proof. It was proved in [14, 17] that $\mathsf{rJL}_{\mathcal{CS}}$ is in NP. By Lemma 31, $\mathsf{rJL}_{\mathcal{CS}}$ is fitting. Thus, by Theorem 28, $\mathsf{rJL}_{\mathcal{CS}}$ is NP-hard. □

7. Reflected Justification Logics with Hypotheses

The goal of this section is to extend the $*$-calculi to situations where rJL is augmented with additional axioms. This will be important for the proof of Theorem 43 in the next section that shows the Π_2^p-hardness of the Derivability Problems for pure justification logics.

Some proofs in this section will be semantic. Accordingly, we introduce the simplest semantics for pure justification logics, that of symbolic models, also called Mkrtychev models or simply *M-models*. This semantics was first introduced for LP by Alexey Mkrtychev in [22] and extended to other pure justification logics in [15].

Definition 33. Let $\mathcal{CS}$ be a constant specification for a pure justification logic $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$. An *M-model for* $\mathsf{JL}_{\mathcal{CS}}$ is a pair $\mathfrak{M} = \langle V, \mathcal{A}\rangle$, where V is a *propositional valuation* and $\mathcal{A}$ is an *admissible evidence function* for $\mathsf{JL}_{\mathcal{CS}}$. Informally, an admissible evidence function specifies for each term t and formula F whether t is considered admissible evidence for F. If $\mathcal{A}(t, F) = \mathit{True}$, we say that $\mathcal{A}$ *satisfies* $t\!:\!F$. Being satisfied by $\mathcal{A}$ is one of the criteria necessary for $t\!:\!F$ to hold in an M-model.

Formally, a function $\mathcal{A}\colon \mathit{Tm} \times \mathit{Fm} \to \{\mathit{True}, \mathit{False}\}$ is called an *admissible evidence function for a justification logic* $\mathsf{JL}_{\mathcal{CS}}$ iff it is closed under deduction in the $*$-calculus for the reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$ (see Table 2). That is to say, if an admissible evidence function $\mathcal{A}$ satisfies a set X of justification assertions and $X \vdash_* s : G$ in the respective $*$-calculus, then $\mathcal{A}$ must also satisfy $s\!:\!G$. In addition, if $\mathsf{JL} \in \{\mathsf{JD}, \mathsf{JD4}\}$, an admissible evidence function for $\mathsf{JL}_{\mathcal{CS}}$ must satisfy the following condition: $\mathcal{A}(t, \bot) = \mathit{False}$ for all terms t. We will use $\mathcal{A}(t, F)$ as an abbreviation for $\mathcal{A}(t, F) = \mathit{True}$ and also $\neg\mathcal{A}(t, F)$ as an abbreviation for $\mathcal{A}(t, F) = \mathit{False}$.[11]

Finally, the *truth relation* $\mathfrak{M} \vDash H$ is defined as follows:

$\mathfrak{M} \vDash P \qquad \text{iff} \qquad V(P) = \mathit{True};$

Boolean connectives behave classically;

$$\mathfrak{M} \vDash t\!:\!F \qquad \text{iff} \qquad \begin{cases} \mathfrak{M} \vDash F \text{ and } \mathcal{A}(t, F) & \text{if } \mathsf{JL} \in \{\mathsf{JT}, \mathsf{LP}\}, \\ \mathcal{A}(t, F) & \text{if } \mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{J4}, \mathsf{JD4}\}. \end{cases}$$

Let $\mathcal{CS}$ be a constant specification for a pure justification logic JL and Γ be any set of formulas. We write $\mathsf{JL}_{\mathcal{CS}}\{\Gamma\}$ to denote the closure of $\mathsf{JL}_{\mathcal{CS}} \cup \Gamma$ under modus ponens. It is easy to verify that the deduction theorem holds for $\mathsf{JL}_{\mathcal{CS}}$, and hence we have that

$$\mathsf{JL}_{\mathcal{CS}}\{\Gamma\} \vdash A \qquad \text{iff} \qquad \text{there exists a finite set } \Gamma_0 \subseteq \Gamma \text{ such that } \mathsf{JL}_{\mathcal{CS}} \vdash \bigwedge \Gamma_0 \to A \ .$$

Definition 34. Suppose that every formula in Γ has the form $t\!:\!A$, i.e., Γ consists only of justification assertions. We define the following calculi:

$$*_{\mathcal{CS}+\Gamma} = *_{\mathcal{CS}} + \Gamma \qquad \text{and} \qquad *!_{\mathcal{CS}+\Gamma} = *!_{\mathcal{CS}} + \Gamma \ . \tag{22}$$

[11] N. Krupski and Mkrtychev used the notation $F \in *(t)$ instead of $\mathcal{A}(t, F)$ and $F \notin *(t)$ instead of $\neg\mathcal{A}(t, F)$.

That is to say, in each case the set of axioms is extended by all justification assertions from Γ while the rules remain the same and can be freely applied to the new axioms. When the type of a $*$-calculus and a particular $\mathcal{CS}$ are clear from the context or when they can be arbitrary, we will use the term $*_\Gamma$-calculus.

It is natural to ask whether a given $*_\Gamma$-calculus can prove every formula in the reflected fragment of $\mathsf{JL}_{\mathcal{CS}}\{\Gamma\}$ for the respective $\mathsf{JL}_{\mathcal{CS}}$. Unfortunately, there are cases where this does not hold. As an example from Kuznets [17], consider the situation for $\mathsf{LP}_{\mathcal{CS}}$ where $\Gamma = \{c : p, c : \neg p\}$. Then, via the Factivity Axiom, $\mathsf{LP}_{\mathcal{CS}}\{\Gamma\}$ is inconsistent, whereas there are certainly justification assertions that cannot be proved in the $*!_{\mathcal{CS}+\Gamma}$-calculus. Thus, we need additional restrictions on Γ. For this, we introduce the following

Definition 35. A set of justification assertions Γ is called *factive* provided that, whenever $t : A \in \Gamma$, either (a) A is of the form $s : B$ and $A \in \Gamma$ or (b) A is a purely propositional[12] formula. The set of purely propositional formulas A such that $t : A \in \Gamma$ for some term t is called the *propositional content* of Γ, and is denoted $\mathsf{Prop}(\Gamma)$. We call a factive set Γ *consistent* provided $\mathsf{Prop}(\Gamma)$ is (propositionally) consistent.

The next theorem generalizes N. Krupski's Theorem 5.1 from [14].

Theorem 36. *Let Γ be a consistent factive set of justification assertions and $\mathcal{CS}$ be a constant specification for a pure justification logic $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$. Then, for any formula of the form $t{:}F$, we have*

$$\mathsf{JL}_{\mathcal{CS}}\{\Gamma\} \vdash t{:}F \quad \Longleftrightarrow \quad \vdash_{*_\Gamma} t{:}F$$

for the respective $*_\Gamma$-calculus.

Proof. The proof is similar to the proof of Theorem 5.1 from [14], which in turn uses constructions of Mkrtychev [22]. The $\Longleftarrow$-direction follows from the fact that any $*_\Gamma$-derivation can be easily converted into a derivation in $\mathsf{JL}_{\mathcal{CS}}\{\Gamma\}$. Indeed, all axioms of the $*_\Gamma$-calculus are either instances of axiom internalization in $\mathsf{JL}_{\mathcal{CS}}$ or members of Γ and hence axioms of $\mathsf{JL}_{\mathcal{CS}}\{\Gamma\}$. Each rule in $*_\Gamma$-calculus translates into the corresponding axiom of JL followed by one or two applications of modus ponens.

We prove the $\Longrightarrow$-direction by showing its contrapositive. Suppose $*_\Gamma \nvdash t{:}F$. Let the function $\mathcal{A}_\Gamma\colon Tm \times Fm \to \{True, False\}$ be defined by

$$\mathcal{A}_\Gamma(s, G) \qquad \Longleftrightarrow \qquad *_\Gamma \vdash s{:}G \ .$$

For $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}\}$, whereby $*_\Gamma = *_{\mathcal{CS}+\Gamma}$, it is clear that $\mathcal{A}_\Gamma$ is an admissible evidence function for $\mathsf{JT}_{\mathcal{CS}}$; similarly, for $\mathsf{JL} \in \{\mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$, when $*_\Gamma = *!_{\mathcal{CS}+\Gamma}$, we have that $\mathcal{A}_\Gamma$ is an admissible evidence function for $\mathsf{LP}_{\mathcal{CS}}$. Note that any constant specification for J or JD can also serve as a constant specification for JT because all axioms of J and JD are also axioms of JT. Similarly, if a constant specification can be used

[12] A *purely propositional* formula is one that does not contain any justification terms.

for $\mathsf{J4}$ or $\mathsf{JD4}$, it can also be used for LP. In either case, by definition $\mathcal{A}_\Gamma$ satisfies all justification assertions from Γ but does not satisfy $t\!:\!F$ by our assumption.

Since Γ is consistent, there exists a propositional valuation V that satisfies $\mathsf{Prop}(\Gamma)$. Consider the M-model $\mathfrak{M} = \langle V, \mathcal{A}_\Gamma \rangle$ for $\mathsf{JT}_{\mathcal{CS}}$ or for $\mathsf{LP}_{\mathcal{CS}}$ respectively. Note that $\mathfrak{M} \nvDash t : F$ since $\neg\mathcal{A}_\Gamma(t, F)$. In addition, for either $\mathsf{JT}_{\mathcal{CS}}$ or $\mathsf{LP}_{\mathcal{CS}}$ we can prove that $\mathfrak{M} \vDash \Gamma$. Indeed, it is easy to show by induction on k that $\mathfrak{M} \vDash s_k : \ldots : s_1 : G$ for each $s_k : \ldots : s_1 : G \in \Gamma$, where G is a purely propositional formula. The base case, $k = 1$, follows from $G \in \mathsf{Prop}(\Gamma)$.

The existence of a $\mathsf{JT}_{\mathcal{CS}}$-model (an $\mathsf{LP}_{\mathcal{CS}}$-model) where all formulas from Γ hold while $t : F$ is false shows that $\mathsf{JT}_{\mathcal{CS}}\{\Gamma\} \nvdash t : F$ (respectively $\mathsf{LP}_{\mathcal{CS}}\{\Gamma\} \nvdash t : F$). But every $\mathsf{J}_{\mathcal{CS}}$- or $\mathsf{JD}_{\mathcal{CS}}$-derivation is also a $\mathsf{JT}_{\mathcal{CS}}$-derivation (for the same $\mathcal{CS}$); similarly, any $\mathsf{J4}_{\mathcal{CS}}$- or $\mathsf{JD4}_{\mathcal{CS}}$-derivation is also an $\mathsf{LP}_{\mathcal{CS}}$-derivation. Hence $\mathsf{JL}_{\mathcal{CS}}\{\Gamma\} \nvdash t\!:\!F$. This completes the proof of Theorem 36. □

By analogy with $\mathsf{rJL}_{\mathcal{CS}}$ we will denote the reflected fragment of $\mathsf{JL}_{\mathcal{CS}}\{\Gamma\}$

$$\mathsf{rJL}_{\mathcal{CS}}\{\Gamma\} = \{t\!:\!F \mid \mathsf{JL}_{\mathcal{CS}}\{\Gamma\} \vdash t\!:\!F\} \enspace .$$

As we proved, $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma\}$ is axiomatized by its respective $*_\Gamma$-calculus. Note that the only difference between a $*$-calculus and its corresponding $*_\Gamma$-calculus is the addition of axioms Γ. Therefore, a reflected fragment $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma\}$ is axiomatized by the same $*$-calculus as $\mathsf{rJL}_{\mathcal{CS}}$ as far as rules are concerned. Consequently, there are still only two sets of rules chosen based on whether full positive introspection holds in JL. The differences between these $*$-calculi, as in the case of $\mathsf{rJL}_{\mathcal{CS}}$, is in their axioms.

As a consequence of the above construction, other results that N. Krupski [14] established for $\mathsf{LP}_{\mathcal{CS}}$ also hold for $\mathsf{JL}_{\mathcal{CS}}\{\Gamma\}$. First, by the minimality of the admissible evidence function $\mathcal{A}_\Gamma$, the disjunction property for formulas of the form $s\!:\!F \vee t\!:\!G$ holds for $\mathsf{JL}_{\mathcal{CS}}\{\Gamma\}$ (cf. Corollary 2 of [14]). Similarly, if $\mathcal{CS}$ is schematic (and polynomially decidable) and Γ is finite, then the Derivability Problem for a $*_\Gamma$-calculus is in NP for either of the calculi, i.e., $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma\}$ is in NP. This is proved by a construction similar to the one used in the proof of Theorem 5.2 in [14], the main difference being that derivations in $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma\}$ correspond to $\mathsf{rJL}_{\mathcal{CS}}$-derivations from hypotheses Γ, whereas N. Krupski considered only derivations without hypotheses.

For us, the importance of Theorem 36 lies in the fact that the results of Sects. 3–5 translate to $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma\}$ for a proper subclass of consistent factive sets Γ.

Lemma 37. *Let $\mathcal{CS}$ be a constant specification for a pure justification logic $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$ such that the reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$ is fitting. Let Γ be a consistent factive set of justification assertions such that the only terms that occur in Γ are justification variables. Then Lemmas 16–19, 21, 22, 25, and 26 all still hold if $\mathsf{rJL}_{\mathcal{CS}}$ is uniformly replaced by $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma\}$ in the statements of the lemmas.*

Proof. Since any derivation in $\mathsf{rJL}_{\mathcal{CS}}$ is also a derivation in $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma\}$, the proofs of the $\Longleftarrow$-directions in all these lemmas do not require any changes. The $\Longrightarrow$-directions also hold because the only terms that gain additional provably justified formulas in $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma\}$ are those that contain justification variables from Γ, but no variables have been used for construction of the terms in the proofs of all these lemmas. □

8. A Lower Bound for Full Justification Logics

Sections 4 and 5 established a reduction from the Vertex Cover Problem to the Derivability Problem in a given reflected fragment thereby proving NP-hardness of the latter. In the present section, we extend the proof method and obtain stronger lower bounds for full (non-reflected) pure justification logics $\mathsf{JL}_{\mathcal{CS}}$.[13] We first prove that a quantified version of the Vertex Cover Problem is Π_2^p-hard by reducing co-QSAT$_2$ to it. Then we reduce this Quantified Vertex Cover Problem to the Derivability Problem in a given justification logic.

Definition 38. By co-QSAT$_2$ we mean the following problem. Let φ be any 3CNF formula, i.e., a propositional formula in conjunctive normal form with exactly three literals in each clause. Given such a φ with its sentence letters partitioned into two sets $\vec{p} = \{p_1, \dots, p_{|\vec{p}|}\}$ and $\vec{q} = \{q_1, \dots, q_{|\vec{q}|}\}$, determine whether

$$\psi = (\forall p_1) \cdots (\forall p_{|\vec{p}|})(\exists q_1) \cdots (\exists q_{|\vec{q}|})\varphi \tag{23}$$

is true.

The following theorem is standard. Several of its slightly different variants can be found, for instance, in [23, 24, 25].

Theorem 39. *co-QSAT$_2$ is Π_2^p-complete.*

Let $\psi = (\forall\vec{p})(\exists\vec{q})(C_1 \wedge \cdots \wedge C_r)$ be a formula of type (23), where each C_i is a 3-clause: $C_i = L_{i,1} \vee L_{i,2} \vee L_{i,3}$, $i = 1, \dots r$. Each literal $L_{i,z}$ must be p_j, $\neg p_j$, q_j, or $\neg q_j$ for some j.

Given ψ we construct a graph $G_\psi = \langle V_\psi, E_\psi \rangle$ with vertices labeled by literals (the construction is identical with the reduction of 3SAT to VC as given in [13]). The graph is defined as follows:

- For each clause C_i, $i = 1, \dots, r$, in ψ we have a triangle of pairwise joined vertices $c_{i,1}$, $c_{i,2}$, and $c_{i,3}$ in G_ψ. Each vertex $c_{i,z}$ is labeled by the corresponding literal $L_{i,z}$. These are called *clause vertices* and *clause edges*.
- For each sentence letter q_j, there are two vertices $v_{j,0}$ and $v_{j,1}$ joined by an edge. The vertex $v_{j,0}$ is labeled with q_j and $v_{j,1}$ is labeled with $\neg q_j$. These are called *literal vertices* and *literal edges*.
- For each sentence letter p_j, there are two vertices $u_{j,0}$ and $u_{j,1}$ joined by an edge. The vertex $u_{j,0}$ is labeled with p_j and $u_{j,1}$ is labeled with $\neg p_j$. These are also called *literal vertices* and *literal edges*.
- A clause vertex and a literal vertex are joined by a *connecting edge* iff they are labeled by the same literal.

[13] Applying this method to hybrid logics does not make sense since they are mostly at least PSPACE-hard by virtue of being conservative over the corresponding modal logic. As for $\mathsf{S5_1LP}$, the only hybrid logic that may not be PSPACE-hard, its conservativity over LP would easily yield the lower bound that can be obtained by our method.

There are $3r + 2N$ vertices in G_ψ, where $N = |\vec{p}| + |\vec{q}|$; namely, $3r$ clause vertices and $2N$ literal vertices. Similarly, there are $6r + N$ edges in G_ψ; namely, $3r$ clause edges, N literal edges, and $3r$ connecting edges.

As is argued in [13], G_ψ has a vertex cover of size $\leq 2r + N$ iff φ is satisfiable. First, any vertex cover of G_ψ must have at least $2r + N$ vertices since a vertex cover must contain at least one vertex for each literal edge and at least two vertices from each clause triangle. Second, since any vertex cover C of size $2r + N$ must contain exactly one literal vertex per literal edge in the graph, it is possible to define a propositional valuation τ_C by letting $\tau_C(L) = \mathit{True}$ for exactly those literals that label literal vertices from the vertex cover. It is not hard to see that this valuation τ_C satisfies φ. Conversely, if τ is any valuation that satisfies φ, then there exists a vertex cover C of size $2r + N$ such that $\tau = \tau_C$.

Definition 40. Let π denote a (partial) valuation with domain $\vec{p}$. A vertex cover C of G_ψ is called its *π-cover* if C contains all literal vertices labeled by literals from the set

$$\{p_j \mid \pi(p_j) = \mathit{True}\} \cup \{\neg p_j \mid \pi(p_j) = \mathit{False}\} \ .$$

The above discussion yields the following proposition.

Proposition 41. *A sentence ψ as in (23) is true iff for every valuation π with domain $\vec{p}$, the graph G_ψ has a π-cover of size $2r + N$.*

In order to work with balanced conjunctions, we modify G_ψ to transform the questions about the existence of π-covers into Binary Vertex Cover Problems. For this, we construct a graph G'_ψ that has the following properties: (a) G'_ψ has $2^k + 2|\vec{p}|$ vertices, (b) G'_ψ has 2^m edges, (c) the sought-for π-covers have size 2^l, and (d) the size of G'_ψ is linear in the size of ψ. In effect, G'_ψ is a binary exponential graph, except that the vertices $u_{j,0}$ and $u_{j,1}$, labeled p_j and $\neg p_j$ respectively, are not counted. The construction of G'_ψ from G_ψ mimics that from the proof of Theorem 12: to ensure (c), add extra pairs of vertices joined by edges; to ensure (b), add a "star"-shape as in Stage 2 of the proof of Theorem 12; then, to ensure (a), add extra isolated vertices. By construction, Proposition 41 now implies the following property of G'_ψ:

Proposition 42. *A sentence ψ as in (23) is true iff for every valuation π with domain $\vec{p}$, the graph G'_ψ has a π-cover of size 2^l.*

We are now ready to prove the Π_2^p-hardness of the Derivability Problem for $\mathsf{JL}_{\mathcal{CS}}$.

Theorem 43. *Let $\mathcal{CS}$ be a constant specification for a pure justification logic $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$ such that the reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$ is fitting. Then the Derivability Problem for $\mathsf{JL}_{\mathcal{CS}}$ is Π_2^p-hard.*

Proof. We prove the theorem by reduction from co-QSAT$_2$. Given a formula ψ as in (23), we construct the graph G'_ψ as described above and apply to this graph the encoding from Definition 13. We define a set Γ_ψ of formulas by

$$\Gamma_\psi \quad = \quad \{x\colon p_j \vee x\colon \overline{p}_j \mid j = 1, \ldots, |\vec{p}|\} \ ,$$

where p_j is a sentence letter that corresponds to the literal vertex $u_{j,0}$ and $\overline{p}_j$ is a sentence letter that corresponds to the literal vertex $u_{j,1}$. Intuitively, the sentence letter p_j in the encoding corresponds to the literal p_j in the formula ψ while the sentence letter $\overline{p}_j$ corresponds to the literal $\neg p_j$, which explains the chosen notation. We hope that the resulting small collision of notation: p_j is the sentence letter that encodes the literal vertex that corresponds to the literal p_j, will facilitate understanding rather than hinder it.

Let γ_ψ be the conjunction of the formulas in Γ_ψ. (Unlike the other conjunctions we work with, γ_ψ need not be balanced.) Let V'_ψ be the set of all vertices of G'_ψ, and let $V''_\psi = V'_\psi \setminus \{u_{j,0}, u_{j,1} \mid j = 1, \ldots, |\vec{p}|\}$. Note that $|V''_\psi| = 2^k$ is a power of two. We define K_ψ to be

$$K_\psi \quad = \quad \gamma_\psi \to t'_{k\to l\to m} \colon (F_{V''_\psi} \to F_{G'_\psi}) \enspace ,$$

where a term $t'_{k\to l\to m}$ plays a role similar to $t_{k\to l\to m}$ and is defined below. To prove Theorem 43 it will suffice to show that $\mathsf{JL}_{\mathcal{CS}} \vdash K_\psi$ iff ψ is true.

By the deduction theorem, $\mathsf{JL}_{\mathcal{CS}} \vdash K_\psi$ iff

$$\mathsf{JL}_{\mathcal{CS}}\{\Gamma_\psi\} \vdash t'_{k\to l\to m} \colon (F_{V''_\psi} \to F_{G'_\psi}) \enspace . \tag{24}$$

For any valuation π with domain $\vec{p}$, define

$$V_\pi \quad = \quad \{p_j \mid \pi(p_j) = \mathit{True}\} \cup \{\overline{p}_j \mid \pi(p_j) = \mathit{False}\}$$

and let $\Gamma_{\psi,\pi}$ be the set of formulas $\{x\colon L \mid L \in V_\pi\}$. Note that for any valuation π with domain $\vec{p}$ the set $\Gamma_{\psi,\pi}$ is a finite consistent factive set of justification assertions and the only term that occurs in it is the justification variable x. Hence, by Lemma 37, for any valuation π with domain $\vec{p}$, Lemmas 16–19, 21, 22, 25, and 26 hold for $\mathsf{JL}_{\mathcal{CS}}\{\Gamma_{\psi,\pi}\}$. Consider the assertions

$$\mathsf{JL}_{\mathcal{CS}}\{\Gamma_{\psi,\pi}\} \vdash t'_{k\to l\to m} \colon (F_{V''_\psi} \to F_{G'_\psi}) \enspace . \tag{25}$$

Clearly, (24) holds iff (25) holds for all π. Thus, by Theorem 36 and Proposition 42, in order to prove Theorem 43, it will suffice to prove the following

Lemma 44. *For all valuations π with domain $\vec{p}$,*

$$\mathsf{rJL}_{\mathcal{CS}}\{\Gamma_{\psi,\pi}\} \vdash t'_{k\to l\to m} \colon (F_{V''_\psi} \to F_{G'_\psi}) \qquad \Longleftrightarrow \qquad G'_\psi \text{ has a } \pi\text{-cover of size } 2^l,$$

where by the derivability in $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma_{\psi,\pi}\}$ we understand the derivability in the corresponding $*_{\Gamma_{\psi,\pi}}$-calculus.

Proof. The proof of this lemma is very much like the proof of Theorem 28, but we still need to define the term $t'_{k\to l\to m}$. First, let $t'_{k\to l}$ be the term $\mathrm{conj}(t_k + c_1 \cdot x, l)$. By almost exactly the same reasoning as in Lemma 20, for any set C of size 2^l with $C \subseteq V''_\psi \cup V_\pi$

$$\mathsf{rJL}_{\mathcal{CS}}\{\Gamma_{\psi,\pi}\} \vdash t'_{k\to l} \colon (F_{V''_\psi} \to F_C) \enspace . \tag{26}$$

Indeed, by Lemma 17, $\mathsf{rJL}_{\mathcal{CS}} \vdash t_k \colon (F_{V''_\psi} \to L)$ for any sentence letter L that corresponds to a vertex from V''_ψ. It is easy to see that $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma_{\psi,\pi}\} \vdash c_1 \cdot x \colon (F_{V''_\psi} \to L)$ for any $L \in V_\pi$ because $\mathsf{rJL}_{\mathcal{CS}} \vdash c_1 \colon (L \to (F_{V''_\psi} \to L))$. Thus, (26) follows by Lemma 18.

The converse is proved in a way similar to Lemma 24. In particular, by Lemma 18, $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma_{\psi,\pi}\} \vdash t'_{k\to l} : H$ holds precisely for formulas H of the form $B \to D$, where D is a balanced conjunction of depth $\geq l$ such that for every l-conjunct C_i of D we have $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma_{\psi,\pi}\} \vdash (t_k + c_1 \cdot x) : (B \to C_i)$. By Lemma 17, the term t_k only justifies implications from a formula to its k-conjunct. Clearly, $c_1 \cdot x$ only justifies implications $Y \to L$ with $L \in V_\pi$. Therefore,

$$\text{all } l\text{-conjuncts of } D \text{ that are not in } V_\pi \text{ must be } k\text{-conjuncts of } B. \tag{27}$$

Note that $|t'_{k\to l}| = O\left(k2^l\right)$ as was $|t_{k\to l}|$.

Now define $t'_{k\to l\to m}$ to be the term $\mathrm{syl}(t'_{k\to l}, s_{l\to m})$. By exactly the same argument as in Theorem 23, using the same Lemmas 16 and 22, with (26) replacing the claim of Lemma 20, we have

$$G'_\psi \text{ has a } \pi\text{-cover of size} \leq 2^l \leq |V'_\psi| \quad\Longrightarrow\quad \mathsf{rJL}_{\mathcal{CS}}\{\Gamma_{\psi,\pi}\} \vdash t'_{k\to l\to m} : (F_{V''_\psi} \to F_{G'_\psi}) \ .$$

Conversely, Theorem 27 holds for $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma_{\psi,\pi}\}$ in place of $\mathsf{rJL}_{\mathcal{CS}}$, except that now the set X can contain sentence letters $L \in V_\pi$ in addition to k-conjuncts of B. Indeed, Lemmas 16 and 26 hold for $\mathsf{rJL}_{\mathcal{CS}}\{\Gamma_{\psi,\pi}\}$. The claim of Lemma 24 is here replaced by (27), which allows elements of V_π in X along with k-conjuncts of B.

Lemma 44 now follows by exactly the same argument as in the proof of Theorem 32. □

This completes the proof of Theorem 43. □

Theorem 45. *Let $\mathcal{CS}$ be a schematically injective and axiomatically appropriate constant specification for a pure justification logic $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$. Then the Derivability Problem for $\mathsf{JL}_{\mathcal{CS}}$ is Π^p_2-hard.*

Proof. By Lemma 31, $\mathsf{rJL}_{\mathcal{CS}}$ is fitting. Thus, by Theorem 43, $\mathsf{JL}_{\mathcal{CS}}$ is Π^p_2-hard. □

Theorem 46. *Let $\mathcal{CS}$ be a schematically injective and axiomatically appropriate constant specification for a pure justification logic $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{LP}\}$. Then the Derivability Problem for $\mathsf{JL}_{\mathcal{CS}}$ is Π^p_2-complete.*

Proof. It was proved in [15, 18] that $\mathsf{JL}_{\mathcal{CS}}$ is in Π^p_2. On the other hand, the Π^p_2-hardness follows from Theorem 45. □

The lower bound from Theorem 45 was first proved for $\mathsf{LP}_{\mathcal{CS}}$ by Milnikel in [21]. A slightly stronger result can be found there for $\mathsf{J4}_{\mathcal{CS}}$: it is Π^p_2-hard for any schematic and axiomatically appropriate $\mathcal{CS}$. The results for the other four logics are new. By analogy with Milnikel's result for $\mathsf{J4}_{\mathcal{CS}}$, we conjecture that the requirement of schematic injectivity in Theorem 45 for $\mathsf{J}_{\mathcal{CS}}$ can be relaxed to that of schematicness.

9. The Role of $\mathcal{CS}$

Our method for proving lower bounds for both justification logics and their reflected fragments as well as Milnikel's original proof of the lower bound for $\mathsf{LP}_{\mathcal{CS}}$ from [21] require $\mathcal{CS}$ to be axiomatically appropriate and schematically injective. A natural question arises: whether these two conditions on $\mathcal{CS}$ are essential for proving the lower bounds? In particular, is LP itself Π_2^p-hard? Although we cannot answer the latter question, in this section we will try to explore the dependency of the lower bound on a constant specification.

It is clear that neither schematic injectivity nor axiomatic appropriateness are necessary for the lower bounds to hold. In particular,

Lemma 47. *Let* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}, \mathsf{T}_n\mathsf{LP}, \mathsf{S4}_n\mathsf{LP}, \mathsf{S5}_n\mathsf{LP}\}$. *There exists a constant specification* $\mathcal{CS}$ *for* JL *that is neither schematically injective nor axiomatically appropriate such that* $\mathsf{rJL}_{\mathcal{CS}}$ *is NP-hard. If* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$*, then, in addition,* $\mathsf{JL}_{\mathcal{CS}}$ *is* Π_2^p*-hard.*

Proof. Let terms c_1, c_2, $c_{\wedge 1, \wedge 2}$, $c_\wedge$, and $c_{\vee 1, \vee 2}$ from (1) be constants. Let all tautologies from the right sides of the five equivalencies in (1) be axioms from A1. Finally, let a constant specification $\mathcal{CS}$ be such that all five equivalencies from (1) hold while no other constant justifies any axioms at all. Then the reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$ is clearly fitting. Thus, by Theorem 29, $\mathsf{rJL}_{\mathcal{CS}}$ is NP-hard. In addition, if JL is a pure justification logic, by Theorem 43, $\mathsf{JL}_{\mathcal{CS}}$ is Π_2^p-hard. At the same time, this $\mathcal{CS}$ is surely not axiomatically appropriate. It is not schematically injective either since constants $c_{\wedge 1, \wedge 2}$ and $c_{\vee 1, \vee 2}$ justify two axiom schemes each.

The constructed constant specification is schematic, but even the schematicness condition is easy to violate provided the constants c_1, c_2, $c_{\wedge 1, \wedge 2}$, $c_\wedge$, and $c_{\vee 1, \vee 2}$ remain schematic. It should noted that schematicness is often needed to prove the matching upper bound. □

The constant specification from the proof of the previous lemma also demonstrates another curious fact: the $+$-operation does not play a big role in the lower bound on the complexity of the logic.

Lemma 48. *Let* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}, \mathsf{T}_n\mathsf{LP}, \mathsf{S4}_n\mathsf{LP}, \mathsf{S5}_n\mathsf{LP}\}$. *There exists a constant specification* $\mathcal{CS}$ *such that the* $+$*-free reflected fragment of* $\mathsf{JL}_{\mathcal{CS}}$ *is NP-hard.*

Proof. Although $+$ was used to construct terms $c_{\wedge 1, \wedge 2}$ and $c_{\vee 1, \vee 2}$ for schematically injective constant specifications, it is not required for the constant specification $\mathcal{CS}$ from the proof of Lemma 47. Nowhere else in our reduction was $+$ used. Therefore, even if axiom A3 and rule $*$A3 are omitted from the axiomatizations of JL and of its reflected fragment respectively and the operation $+$ is dropped from the language completely (see [12] for precise definitions) the resulting $+$-free reflected fragment is still fitting and, hence, NP-hard. □

However, the ability of one term to justify several axiom schemes does seem to be necessary for the proven lower bounds. This ability can be ensured already on the level of constants, without the use of $+$. However, if the lack of $+$ is coupled with the schematic injectivity of a constant specification, then all terms effectively become schematically injective and the reflected fragment is polynomially decidable, which can be shown by analyzing N. Krupski's algorithm from [14].

The preceding discussion shows that the lower bounds are, in some sense, local. Namely, they can be ensured by finitely many constants using only a small portion of propositional reasoning. In fact, the proof of the existence of undecidable $\mathsf{LP}_{\mathcal{CS}}$ from [16] has a similar flavor: only a few constants are sufficient to make the logic undecidable. For instance, it is possible that $\mathsf{JL}_{\mathcal{CS}}$ can be shown to be Π_2^p-hard using one part of the constant specification $\mathcal{CS}$ and to be undecidable using another part. Therefore, the requirements of schematicness and/or schematic injectivity can be relaxed to apply only to a small subset of justification constants.

Since axiomatic appropriateness is also a local property, it becomes clear that it is independent of whether the reflected fragment is fitting. In particular, Lemma 47 can be easily reformulated for an axiomatically appropriate but not schematically injective constant specification. The only change in the proof would be an addition of a sixth constant that proves all the axioms.

However little of internalization is used in the proof of our lower bounds, it cannot be dispensed of completely:

Lemma 49. *Let* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}, \mathsf{T}_n\mathsf{LP}, \mathsf{S4}_n\mathsf{LP}, \mathsf{S5}_n\mathsf{LP}\}$. *There exists a schematically injective but not axiomatically appropriate constant specification* $\mathcal{CS}$ *for* JL *such that* $\mathsf{rJL}_{\mathcal{CS}}$ *is in P. If* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{LP}\}$, *then, in addition,* $\mathsf{JL}_{\mathcal{CS}}$ *is in co-NP.*

Proof. It has been known that LP_0 with the empty constant specification $\mathcal{CS} = \emptyset$ is in co-NP. Its reflected fragment is trivially in P since it is empty. Extending these results to other justification logics is straightforward. □

The preceding lemma can also be proved using a non-empty schematically injective constant specification, but the proof is much more involved.

Acknowledgments.

We are grateful to Sergei Artemov for playing the role of a catalyst for this research project.

References

[1] Sergei N. Artemov. Operational modal logic. Technical Report MSI 95–29, Cornell University, December 1995.

[2] Sergei N. Artemov. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic*, 7(1):1–36, March 2001.

[3] Sergei [N.] Artemov. Justified common knowledge. *Theoretical Computer Science*, 357(1–3):4–22, July 2006.

[4] Sergei [N.] Artemov. The logic of justification. *The Review of Symbolic Logic*, 1(4):477–513, December 2008.

[5] Sergei [N.] Artemov. Why do we need Justification Logic? Technical Report TR–2008014, CUNY Ph.D. Program in Computer Science, September 2008.

[6] 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.

[7] Sergei [N.] Artemov and Elena Nogina. Introducing justification into epistemic logic. *Journal of Logic and Computation*, 15(6):1059–1073, December 2005.

[8] Maria Luisa Bonet and Samuel R. Buss. The deduction rule and linear and near-linear proof simulations. *Journal of Symbolic Logic*, 58(2):688–709, June 1993.

[9] Vladimir N. Brezhnev. On explicit counterparts of modal logics. Technical Report CFIS 2000–05, Cornell University, 2000.

[10] Samuel R. Buss and Roman Kuznets. The NP-completeness of reflected fragments of justification logics. In Sergei [N.] Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science, International Symposium, LFCS 2009, Deerfield Beach, FL, USA, January 3–6, 2009, Proceedings*, volume 5407 of *Lecture Notes in Computer Science*, pages 122–136. Springer, 2009.

[11] Alexander Chagrov and Michael Zakharyaschev. *Modal Logic*, volume 35 of *Oxford Logic Guides*. Oxford University Press, 1997.

[12] Melvin Fitting. The logic of proofs, semantically. *Annals of Pure and Applied Logic*, 132(1):1–25, February 2005.

[13] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[14] Nikolai V. Krupski. On the complexity of the reflected logic of proofs. *Theoretical Computer Science*, 357(1–3):136–142, July 2006.

[15] 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.

[16] 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(1) of *Bulletin of Symbolic Logic*, page 111. Association for Symbolic Logic, March 2005. Abstract.

[17] Roman Kuznets. *Complexity Issues in Justification Logic*. PhD thesis, CUNY Graduate Center, May 2008.

[18] Roman Kuznets. Complexity through tableaux in justification logic. In *2008 European Summer Meeting of the Association for Symbolic Logic, Logic Colloquium '08, Bern, Switzerland, July 3–July 8, 2008*, volume 15(1) of *Bulletin of Symbolic Logic*, page 121. Association for Symbolic Logic, March 2009. Abstract.

[19] Roman Kuznets. Self-referential justifications in epistemic logic. *Theory of Computing Systems*, Online First, April 2009.

[20] Richard E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, September 1977.

[21] Robert [S.] Milnikel. Derivability in certain subsystems of the Logic of Proofs is Π_2^p-complete. *Annals of Pure and Applied Logic*, 145(3):223–239, March 2007.

[22] 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.

[23] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[24] Steven Rudich and Avi Wigderson, editors. *Computational Complexity Theory*, volume 10 of *IAS/Park City Mathematics Series*. AMS/PCMI, 2004.

[25] Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, October 1976.