Logical Omniscience as a Computational Complexity Problem

Sergei Artemov
Ph.D. Program in Computer Science
CUNY Graduate Center
New York City, NY, United States
sartemov@gc.cuny.edu

Roman Kuznets
Institut für Informatik und angewandte Mathematik
Universität Bern
Bern, Switzerland
kuznets@iam.unibe.ch

Abstract

The logical omniscience feature assumes that an epistemic agent knows all logical consequences of her assumptions. This paper offers a general theoretical framework that views logical omniscience as a computational complexity problem. We suggest the following approach: we assume that the knowledge of an agent is represented by an epistemic logical system E; we call such an agent *not logically omniscient* if for any valid knowledge assertion $\mathcal{A}$ of type F *is known*, a proof of F in E can be found in polynomial time in the size of $\mathcal{A}$. We show that agents represented by major modal logics of knowledge and belief are logically omniscient, whereas agents represented by justification logic systems are not logically omniscient with respect to t *is a justification for* F.

1 INTRODUCTION

Modal logic of knowledge contains the epistemic closure principle in the following modal logical form:

$$\Box(F \to G) \to (\Box F \to \Box G),$$

which yields an unrealistic feature called *logical omniscience* whereby an agent knows all logical consequences of her assumptions. In particular, a logically omniscient agent who knows the rules of chess would also know whether White has a non-losing strategy, an agent who knows the product of two primes would also know both of those primes,[1] etc.

The logical omniscience defect, which was identified in (Fagin and Halpern, 1987; Hintikka, 1962, 1975; Moses, 1988; Parikh, 1987), was studied in (Alechina and Logan, 2002; Aumann, 1986; Elgot-Drapkin *et al.*, 1991; Fagin *et al.*, 1995; Halpern and Moses, 1992; Konolige, 1986; Levesque, 1984; Montague, 1970; Moore, 1986; Parikh, 1995, 2005, 2008; Rantala, 1982; Scott, 1970; Shin and Williamson, 1994; Vardi, 1986; Wansing, 1990), among others. Most of these papers adjust specific epistemic models to keep logical omniscience at bay and provide a range of practical tools to handle this problem.

We adopt a general, complexity-based, view of the logical omniscience problem. Acquiring knowledge consumes certain resources (time, space, attention, etc.), and an adequate model of knowledge should reflect this fact in some degree of generality; the complexity theory provides a reasonable platform for such an approach.

Our approach assumes that for an agent, there is an epistemic logical system E in a language capable of representing epistemic assertions: proofs in E provide constructive evidence of knowledge. In particular, for each valid assertion F *is known*, there is a proof of F in E. We attribute the logical omniscience effect to a situation where for some 'short' valid knowledge assertions F *is known*, it is impossible to feasibly find proofs of F in E.

In (Artemov and Kuznets, 2006), the following Logical Omniscience Test (LOT) was suggested: an epistemic system E is *not logically omniscient* if for any valid-in-E knowledge assertion $\mathcal{A}$ of type F *is known*, there is a proof of F in E, the complexity of which is bounded by some polynomial in the size of $\mathcal{A}$. LOT was inspired by the Cook–Reckhow theory of proof complexity (Cook and Reckhow, 1974; Pudlák, 1998).

In this paper, we suggest a more general Strong Logical Omniscience Test (SLOT) based on time complexity: an epistemic system E is *not logically omniscient* if for any valid-in-E knowledge assertion $\mathcal{A}$ of type F *is known*, a proof of F in E can be found in polynomial time in the size of $\mathcal{A}$.

Both LOT and SLOT connect the size of a knowledge assertion of F with the ability of the system to feasibly provide an adequate evidence for F. In LOT, the feasibility measure is the proof length, whereas in SLOT, it is the time

[1] The latter example is due to Joseph Halpern.

required to obtain a proof.

We argue that both in LOT and SLOT, the natural complexity measure for proofs and formulas is bit size.

We show that major modal logics of knowledge and belief are logically omniscient with respect to both LOT[2] and SLOT, whereas justification logic systems, which contain evidence assertions *t is a justification for F*, are not logically omniscient. These results agree with our intuition and could be interpreted as saying that LOT and SLOT capture the logical omniscience phenomenon and that justification-carrying representation of knowledge could be used to explain and control logical omniscience.

Unlike many semantic approaches that avoid logical omniscience by denying the agents some or all deductive abilities, which results in a trivialized logic of knowledge (Halpern and Pucella, 2007), the justification logic approach is axiomatic. Built-in justification terms in the language of justifications symbolically model reasons why a given fact is known to an agent, which provides a flexible control over the agent's reasoning without imposing rigid bounds. At the same time, annotated knowledge assertions help to lower the complexity of knowledge acquisition by directing the proof search, which allows for a syntactic rather than a semantic representation of the bounds of the agent's reasoning. A natural consequence is an increased length of knowledge assertions. Finding a proof is fast because a justified knowledge assertion $t{:}F$ contains the justification term t, which is a symbolic footprint of such a proof.

We will start by making these notions precise. Let L be a logic in some language $\mathcal{L}$. We will use the notion of an abstract proof system for L developed in (Cook and Reckhow, 1974); cf. also (Pudlák, 1998):

Definition 1. A *proof system* for L is a polynomial-time computable function $E\colon \Sigma^* \to \mathsf{L}$ from the set of words in some alphabet, called proofs, *onto* the set of L-valid formulas. In addition, we consider a measure of size for proofs, which is a function $\ell\colon \Sigma^* \to \mathbb{N}$, and a measure of size for individual formulas $|\cdot|\colon \mathcal{L} \to \mathbb{N}$.

Definition 2. We will call L an *epistemic system* if some subset $r\mathcal{L} \subseteq \mathcal{L}$ is designated as a set of *knowledge assertions*. Each knowledge assertion $\mathcal{A} \in r\mathcal{L}$ has an intended meaning *formula F is known* for a unique formula F. Moreover, we require that function $OK : r\mathcal{L} \to \mathcal{L}$ that extracts the *object of knowledge F* from a given knowledge assertion $\mathcal{A}$ be

- computable in time polynomial in $|\mathcal{A}|$ and
- preserving L-validity: for any $\mathcal{A} \in r\mathcal{L}$

$$\mathsf{L} \vdash \mathcal{A} \quad \Longrightarrow \quad \mathsf{L} \vdash OK(\mathcal{A}) \ .$$

We will now define the reflected fragments[3] of epistemic systems.

Definition 3. Let L be an epistemic system with a set of knowledge assertions $r\mathcal{L}$. The *reflected fragment* rL is the set of all valid knowledge assertions: $\mathsf{rL} = \mathsf{L} \cap r\mathcal{L}$.

By definition of an epistemic system, $OK(\mathsf{rL}) \subseteq \mathsf{L}$. If $OK(\mathsf{rL}) = \mathsf{L}$, the reflected fragment rL is called *complete*.[4] In other words, the reflected fragment is complete if its knowledge covers all theorems of L.

We may want to consider different types of knowledge assertions for the same logic. Then, there will be several epistemic systems associated with one logic. In such cases, we will specify the reflected fragment for each of them, e.g., *the epistemic system L with respect to the reflected fragment* rL.

It should be noted that in a multiagent setting, it is natural to consider several reflected fragments simultaneously, one fragment per agent. Then rL will be the union of individual reflected fragments. Since an introduction of many agents does not have any significant impact on the phenomena discussed, for brevity's sake, we will only consider single-agent situations.

We now have all the necessary ingredients to formulate the logical omniscience tests:

Definition 4. Let E be a proof system for an epistemic system L, or simply an *epistemic proof system* where L can be determined from the context.

- **Logical Omniscience Test (LOT)**: An epistemic proof system E is *not logically omniscient*, or *passes LOT*, if there exists a polynomial P such that for any valid knowledge assertion $\mathcal{A} \in \mathsf{rL}$, there is a proof of $OK(\mathcal{A})$ in E with the size bounded by $P(|\mathcal{A}|)$.
- **Strong Logical Omniscience Test (SLOT)**: An epistemic proof system E is *strongly not logically omniscient*, or *passes SLOT*, if there is a deterministic algorithm, polynomial in $|\mathcal{A}|$, that, for any valid knowledge assertion $\mathcal{A} \in \mathsf{rL}$, is capable of restoring a proof of $OK(\mathcal{A})$ in E.

Both tests are parameterized by the proof system used and by the way the size of formulas is measured. In addition, LOT depends on the size measure used for proofs. A discussion of various proof-size measures and their impact on LOT can be found in (Artemov and Kuznets, 2006). In this paper, we focus on the other test, SLOT, for which the proof-size measure becomes largely irrelevant.

As far as formulas are concerned, we will concentrate on the two most common measures:

[2] This was shown in (Artemov and Kuznets, 2006).

[3] The name originates from N. Krupski's studies of such a fragment for justification logic LP.

[4] Here, $OK(X) = \{OK(F) \mid F \in X\}$ for $X \subseteq r\mathcal{L}$.

- the number of logical symbols in the formula and
- the bit size of the formula

(the latter takes into account indices of sentence letters and the like). The use of other size measures is sometimes warranted in specialized applications, but their discussion remains outside the scope of this paper. Henceforth, we will always assume the size of formulas to be measured according to one of the two measures above.

The results of this paper hold for both measures. It is also natural to extend these measures to proofs whenever LOT is discussed.

2 EPISTEMIC AND JUSTIFICATION LOGICS

In this section, we will briefly recapture the language of modal logic and discuss the appropriate notion of reflected fragment for it. We will also provide an overview of justification logics, which requires a more extended discussion.

2.1 EPISTEMIC MODAL LOGICS

A (single-agent) epistemic modal logic is a logic in language $\mathcal{ML}$, i.e., in propositional language with an additional construct $\Box F$, read as *formula F is known*. It is, therefore, quite natural to define the set of knowledge assertions as

$$r\mathcal{ML} = \{\Box F \mid F \in \mathcal{ML}\}$$

with the associated object of knowledge extraction function

$$OK(\Box F) = F \quad .$$

Thus, according to Definition 3, the reflected fragment of an epistemic modal logic ML is

$$\mathsf{rML} = \{\Box F \mid \mathsf{ML} \vdash \Box F\} \quad .$$

For all epistemic modal logics of knowledge and many epistemic modal logics of belief (e.g., K, K4, D, and D4)

$$\mathsf{ML} \vdash \Box F \qquad \Longleftrightarrow \qquad \mathsf{ML} \vdash F \quad . \tag{1}$$

Therefore,

$$\mathsf{rML} = \{\Box F \mid \mathsf{ML} \vdash F\} \quad . \tag{2}$$

Note 5. Although (1) does not hold for other common logics of belief, e.g., K5, K45, and KD45,[5] their reflected fragments admit a simple description: for any of them, $\Box F$ is derivable iff F is derivable in the corresponding logic of knowledge, i.e., in the modal logic obtained by adding the reflection axiom scheme to the given logic of belief. Still the application of our logical omniscience tests to these logics is problematic since the tests rely on F being derivable. One possible solution would be to consider knowledge assertions in a logic of belief (say, KD45) while considering proofs of the asserted fact in the corresponding logic of knowledge (S5).

The reflected fragments described by (2) are very well behaved. For one thing, they are complete, which is certainly a desired property. Further, the object of knowledge extraction function $OK(\cdot)$ for these logics is a one-to-one correspondence between $r\mathcal{ML}$ and $\mathcal{ML}$. Moreover, both the function and its inverse are computable in linear time.

It turns out that the flip side of the coin is logical omniscience. Although the following theorem is an easy corollary of the more general Theorem 21 from Sect. 4, we will state it here as an assurance that the logical omniscience tests, LOT and SLOT, conform to our intuition about modal logics being logically omniscient.

Theorem 6. *Let a modal epistemic logic* ML *satisfy (1) and be conservative over classical propositional calculus (CPC).*

1. *There is no proof system for* ML *that passes LOT unless NP=coNP.*
2. *There is no proof system for* ML *that passes SLOT unless P=NP.*

In particular, common epistemic modal logics cannot pass either of the tests modulo the stated complexity theory assumptions. Thus, these modal logics are inherently logically omniscient with respect to LOT and SLOT.

Proof. If a proof system E for ML passes LOT, then the derivability of F is equivalent to an existence of an E-proof of F, polynomial in $|OK^{-1}(F)|$, which is the same as being polynomial in $|F|$ since $|OK^{-1}(F)| = |\Box F| = |F|+1$ for modal epistemic systems. If, in addition, E passes SLOT, this proof can then be found by a deterministic algorithm, polynomial in $|F|$.

Guessing a polynomial-size proof and verifying it works as an NP decision procedure, whereas finding the proof and verifying it is a P algorithm for deciding ML. It remains to note that any modal logic conservative over CPC is coNP-hard. □

2.2 JUSTIFICATION LOGICS

2.2.1 Axiom Systems

The first justification logic, the Logic of Proofs LP, was introduced in (Artemov, 1995) to provide a provability semantics for the modal logic S4; see also (Artemov, 2001). The language $\mathcal{JL}$ of LP and other justification logics

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

[5] We thank an anonymous referee for pointing this out.

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, with a countable supply of each of the three.

Statements $t\!:\!F$ can be seen as refinements of modal statements $\Box F$: the latter state 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:

$$\begin{aligned} (t\!:\!F)^\circ &= \Box F^\circ \ , & p^\circ &= p \ , \\ (F \to G)^\circ &= F^\circ \to G^\circ \ , & \bot^\circ &= \bot \ . \end{aligned}$$

Axioms and rules of LP:

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

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

A3. *Monotonicity Axiom* $s\!:\!F \to (s + t)\!:\!F$, $t\!:\!F \to (s + t)\!:\!F$;

A4. *Factivity Axiom* $t\!:\!F \to F$;

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

R4. *Axiom Internalization Rule*: $\dfrac{}{c\!:\!A}$,
where A is an axiom and c is a justification constant.

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

Theorem 7 (Realization Theorem. Artemov, 1995, 2001)**.** $\mathsf{LP}^\circ = \mathsf{S4}$.

Other epistemic modal logics have their own justification counterparts in the same sense. Counterparts of K, D, T, K4, and D4 were developed in (Brezhnev, 2000). 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:

A7. *Consistency* $t\!:\!\bot \to \bot$,

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

Finally, rule R4 for J4 and JD4 is written the same way as for LP, but its scope changes from logic to logic along with the set of axioms. The logics without 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*: $\dfrac{}{\underbrace{!!\cdots!}_{n}c\!:\!\ldots\!:\!!!c\!:\!!c\!:\!c\!:\!A}$,
where A is an axiom, 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.

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

Theorem 8 (Realization Theorem. Brezhnev, 2000)**.**

$$\begin{aligned} &\mathsf{J}^\circ = \mathsf{K}, \quad && \mathsf{JD}^\circ = \mathsf{D}, \quad && \mathsf{JT}^\circ = \mathsf{T}, \\ &\mathsf{J4}^\circ = \mathsf{K4}, && \mathsf{JD4}^\circ = \mathsf{D4}. \end{aligned}$$

Justification counterparts also exist for epistemic modal logics with negative introspection; see (Artemov, 2008; Pacuit, 2005; Rubtsova, 2006). But too little is known about these justification systems to try to answer questions about their logical omniscience.

2.2.2 Constant Specifications

Section 2.2.3 discusses what should be considered the proper reflected fragment for justification logics. This question turns out to present more interest than for modal logics, and it requires a certain flexibility in controlling how the constants are used. This flexibility is achieved through the mechanism of *constant specifications*:

Definition 9. A *constant specification* $\mathcal{CS}$ (for a justification logic JL) is a set of instances of rule R4:

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

With each constant specification $\mathcal{CS}$ we will associate a function from constants to sets of axioms. We will use the same name for the constant specification and its associated function: for each constant c,

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

Given a constant specification $\mathcal{CS}$, the logic $\mathsf{JL}_{\mathcal{CS}}$ is the result of replacing R4 or R4$^!$ by their relativized versions, i.e., respectively by

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

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.

For the Realization Theorem to hold, i.e., for $(\mathsf{JL}_{\mathcal{CS}})^\circ = \mathsf{ML}$ for the corresponding modal logic $\mathsf{ML}=\mathsf{JL}^\circ$, it is necessary and sufficient that $\mathcal{CS}$ be *axiomatically appropriate*:

Definition 10. A constant specification $\mathcal{CS}$ is called:

- *axiomatically appropriate*[6] if
$$\bigcup_c \mathcal{CS}(c) = \{A \mid A \text{ is an axiom of } \mathsf{JL}\};$$
- *finite* if $\mathcal{CS}$ is finite as a set;
- *schematic*[7] if set $\mathcal{CS}(c)$ consists of several (maybe zero) axiom schemes for each constant c;
- *schematically injective*[8] if it is schematic and $\mathcal{CS}(c)$ consists of at most one axiom scheme for each constant c.

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

Lemma 11 (Constructive Necessitation. Artemov, 1995, 2001)**.** *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*[9] *justification term s such that* $\mathsf{JL}_{\mathcal{CS}}\vdash s\!:\!F$.

2.2.3 Reflected Fragments of Justification Logics

We will now try to answer the question of what is the right form of knowledge assertions in justification language. The first answer that comes to mind is, by analogy with modal logics,

$$r\mathcal{JL} = \{t\!:\!F \mid t \text{ is a term, } F \text{ is a formula}\}$$

with the associated object of knowledge extraction function

$$OK(t\!:\!F) = F \enspace .$$

In this case, the reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$ is complete iff $\mathcal{CS}$ is axiomatically appropriate. So it seems that any such $\mathsf{JL}_{\mathcal{CS}}$ can be considered an epistemic system.

As was argued in (Artemov and Kuznets, 2006), justification terms are intended to serve as a persuasive reason for the knowledge of a formula. In this respect, knowledge assertions of type $t\!:\!F$ are not quite satisfactory since both t and F may contain justification constants, the meanings of which are given only in the corresponding constant specification. But an infinite constant specification may contain an infinite amount of information, and so can each knowledge assertion $t\!:\!F$. Naturally, infinite information in one formula can cause logical omniscience. Hence, the following

Definition 12. The set of *comprehensive knowledge assertions* is defined by

$$cr\mathcal{JL} = \{\textstyle\bigwedge\mathcal{CS}\to t\!:\!F \mid t \text{ is a term, } F \text{ is a formula,}\\ \mathcal{CS} \text{ is a finite constant specification}\}$$

with the associated knowledge extraction function

$$OK\left(\bigwedge\mathcal{CS}\to t\!:\!F\right) = F \enspace .$$

In this case, the *comprehensive reflected fragment* for a justification logic JL is defined as

$$\mathsf{crJL} = \left\{\bigwedge\mathcal{CS}\to t\!:\!F \mid \mathsf{JL}_0\vdash\bigwedge\mathcal{CS}\to t\!:\!F\right\} \enspace ,$$

where $\mathsf{JL}_0=\mathsf{JL}_\emptyset$ is JL with the empty constant specification.

Note that comprehensive reflected fragments are $\mathcal{CS}$-independent. Since each JL-derivation uses the Axiom Internalization Rule at most finitely many times, each JL-derivation of $t : F$ can be turned into a JL_0-derivation of $\bigwedge\mathcal{CS}\to t\!:\!F$ for some finite $\mathcal{CS}$:

Lemma 13. *Let* $\mathsf{JL}\in\{\mathsf{J},\mathsf{JD},\mathsf{JT},\mathsf{J4},\mathsf{JD4},\mathsf{LP}\}$. *Then,* $\mathsf{JL}\vdash t\!:\!F$ *iff* $\mathsf{JL}_0\vdash\bigwedge\mathcal{CS}\to t\!:\!F$ *iff* $\mathsf{rJL}_{\mathcal{CS}}\vdash t\!:\!F$ *for some finite constant specification $\mathcal{CS}$.*

As a consequence of this lemma and Constructive Necessitation (for JL), comprehensive reflected fragments are always complete.

In light of Lemma 13, a comprehensive reflected fragment can also be seen as a combination of $\mathsf{rJL}_{\mathcal{CS}}$ for all possible finite $\mathcal{CS}$. None of them are complete by themselves, but their combination is.

It should also be noted that whenever reflected fragment $\mathsf{rJL}_{\mathcal{CS}}$ is complete, it has a smaller complete subfragment:

$$\mathsf{rJL}^-_{\mathcal{CS}} = \{t\!:\!F\in\mathsf{rJL}_{\mathcal{CS}} \mid t \text{ does not contain } +\} \enspace .$$

The completeness of such $+$-free reflected fragments follows from the completeness of $\mathsf{rJL}_{\mathcal{CS}}$ and Constructive Necessitation (Lemma 11).

3 COMPLEXITY GUIDE TO REFLECTED FRAGMENTS OF JL

In this section, we will discuss complexity of various reflected fragments from Sect. 2.2.3. As will be shown in

[6] The term is due to Fitting.

[7] The term is due to Milnikel although the idea goes back to Mkrtychev.

[8] The term is due to Milnikel.

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

Table 2: $*$-Calculi

Calculus	Axioms and rules	Used for
$*_{\mathcal{CS}}$	$*!\mathcal{CS}$, $*$A2, $*$A3	rJ, rJD, rJT
$*!_{\mathcal{CS}}$	$*\mathcal{CS}$, $*$A2, $*$A3, $*$A5	rJ4, rJD4, rLP

Sect. 4, knowing their complexity is the key to determining whether an epistemic proof system is logically omniscient.

The study of reflected fragments was pioneered in (Krupski, 2006), where rLP was studied, axiomatized, and shown to be in NP. His method was extended to rJ, rJD, rJT, rJ4, and rJD4 in (Kuznets, 2008). The decision procedure for these reflected fragments is based on their axiomatizations by the so-called $*$-calculi of the following two types: $*_{\mathcal{CS}}$ and $*!_{\mathcal{CS}}$. The difference between them stems from the presence/absence of general positive introspection in the corresponding justification logic. Both calculi share the following rules:

$*$A2. *Application Rule*
$$\frac{s\!:\!(F \rightarrow G) \qquad t\!:\!F}{s \cdot t\!:\!G}\;;$$

$*$A3. *Sum Rule*
$$\frac{s\!:\!F}{(s+t)\!:\!F}, \qquad \frac{t\!:\!F}{(s+t)\!:\!F}.$$

The $*_{\mathcal{CS}}$-calculus for a given $\mathcal{CS}$ is obtained by adding axioms

$*!\mathcal{CS}$. $\underbrace{!!\cdots!}_{n}c\!:\!\ldots\!:\!!!c\!:\!!c\!:\!c\!:\!A$,

where $c\!:\!A \in \mathcal{CS}$ and $n \geq 0$ is an integer.

Alternatively, for the logics with positive introspection, $\mathsf{J4}_{\mathcal{CS}}$, $\mathsf{JD4}_{\mathcal{CS}}$, and $\mathsf{LP}_{\mathcal{CS}}$, an additional rule is required:

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

while the axioms have a simpler form:

$*\mathcal{CS}$. $c\!:\!A$, where $c\!:\!A \in \mathcal{CS}$.

The resulting calculus is called $*!_{\mathcal{CS}}$-calculus. Both calculi are summarized in Table 2.

Theorem 14 (Krupski, 2006; Kuznets, 2008)**.** *Let $\mathcal{CS}$ be a constant specification for a justification logic* JL*.*
1. If JL $\in \{$J, JD, JT$\}$*,* $\mathsf{rJL}_{\mathcal{CS}} \vdash t\!:\!F \iff *_{\mathcal{CS}} \vdash t\!:\!F$*;*
2. If JL $\in \{$J4, JD4, LP$\}$*,* $\mathsf{rJL}_{\mathcal{CS}} \vdash t\!:\!F \iff *!_{\mathcal{CS}} \vdash t\!:\!F$*.*

Thus, to determine whether $t : F$ is valid, it is necessary and sufficient to consider all derivations in the respective $*$-calculus that can yield $t : F$ as their conclusion. The $*$-calculi have a useful property: each rule strictly increases the size of the outer term. Moreover, each outer term in any $*$-derivation of $t : F$ is a subterm of t. Note also that the rules $*$A2 and $*$A5 are deterministic, i.e., their premises uniquely determine their conclusions. Although the same does not hold for the rule $*$A3 (premise $s : G$ may yield infinitely many conclusions $(s + s') : G$ and $(s' + s) : G$), given a target formula $t : F$, it is sufficient to consider only subterms of t, which effectively makes $*$A3 quasi-deterministic.

This observation suggests a method for exhausting all potential derivations of $t : F$, namely: 1) consider all possible $*$-derivation axioms that assign axioms of JL to occurrences of constants in t; 2) for each such initial assignment, build up a derivation from these axioms until one or several formulas are assigned to t itself; and 3) compare these formulas with F. In general, there may be infinitely many axioms assigned to a constant. So the efficiency of this algorithm, as well as whether it decides $\mathsf{rJL}_{\mathcal{CS}}$ or just recursively enumerates it, depends on the ability to bundle all axioms assigned to a constant into finitely many constructive sets, which allows for an effective application of the deterministic rules of the $*$-calculus to these "bundles." In particular, the following theorem is the result of bundling all axioms into finitely many axiom schemes.

Theorem 15 (Krupski, 2006; Kuznets, 2008)**.** *Let $\mathcal{CS}$ be a schematic*[10] *constant specification for a justification logic* JL $\in \{$J, JD, JT, J4, JD4, LP$\}$*. Then,* $\mathsf{rJL}_{\mathcal{CS}}$ *is in NP.*

This bundling works because each rule of $*$-calculi applied to scheme(s) produces a scheme that can be efficiently computed.

In some cases, see (Kuznets, 2005), $\mathsf{rJL}_{\mathcal{CS}}$ is undecidable; therefore, bundling cannot always be achieved efficiently.

As proved in (Buss and Kuznets, 2009), for certain schematic $\mathcal{CS}$, namely for axiomatically appropriate and schematically injective ones, the reflected fragments $\mathsf{rJL}_{\mathcal{CS}}$ are NP-complete. But in other cases, the complexity can be lowered, which will prove instrumental in finding epistemic systems that pass SLOT.

A good model of $\mathsf{rJL}_{\mathcal{CS}}$ that is in P is presented by the case of finite $\mathcal{CS}$. To facilitate a concise description of the decision algorithm, we extend function $\mathcal{CS}(\cdot)$ to all terms as follows:

$$\mathcal{CS}(t) = \{F \mid \;\; \vdash_{*} t\!:\!F\} \;\; ,$$

where $\vdash_{*}$ stands for derivation in the respective $*$-calculus. Note that $\mathcal{CS}(c) = \{A \mid c\!:\!A \in \mathcal{CS}\} = \{A \mid \;\; \vdash_{*} c\!:\!A\}$.

Theorem 16. *Let $\mathcal{CS}$ be a finite constant specification for a justification logic* JL $\in \{$J, JD, JT, J4, JD4, LP$\}$*. Then* $\mathsf{rJL}_{\mathcal{CS}}$ *is in P.*

Proof. We will first consider the case of $*!_{\mathcal{CS}}$-calculus. Let $t\!:\!F$ be the knowledge assertion given to prove or disprove. There are only finitely many initial assignments of axioms to constants that occur in t. The number and size of axioms assigned to each constant do not depend on t: they

[10] Here and in Theorem 18, it is assumed that $\mathcal{CS}(\cdot)$ is computable in polynomial time.

are bounded by the size of $\mathcal{CS}$. Therefore, it is possible to bundle together all axioms from $\mathcal{CS}(c)$ and assign them to c in a single initial assignment. This initial assignment will have $O(1)$ formulas of size $O(1)$ assigned to each constant c. The deterministic rules of the $*$-calculi applied to finite sets of formulas can be formulated as:

$$\mathcal{CS}(s_1 \cdot s_2) = \{G \mid (\exists F)(F \in \mathcal{CS}(s_2) \wedge F \to G \in \mathcal{CS}(s_1))\} \ , \qquad (3)$$

$$\mathcal{CS}(s_1 + s_2) = \mathcal{CS}(s_1) \cup \mathcal{CS}(s_2) \ , \qquad (4)$$

$$\mathcal{CS}(!s) = \{s{:}F \mid F \in \mathcal{CS}(s)\} \ . \qquad (5)$$

Note that cases (3) and (4) do not increase the size of assigned formulas: in the former case, each $G \in \mathcal{CS}(s_1 \cdot s_2)$ is smaller than some $F \to G \in \mathcal{CS}(s_1)$. Thus, the size of assigned formulas is only increased in (5) by $|s|+1 \leq |t|+1$ each. Since there are at most $|t|$ such steps, for any subterm s of t, the size of all formulas in $\mathcal{CS}(s)$ is bounded by $O(1) + |t|(|t|+1) = O(|t|^2)$.

As for the number of formulas, $|\mathcal{CS}(s)|$, assigned to each subterm s, it is not increased in cases (3) or (5): in the former case $|\mathcal{CS}(s_1 \cdot s_2)| \leq |\mathcal{CS}(s_1)|$. In case (4), clearly, $|\mathcal{CS}(s_1 + s_2)| \leq |\mathcal{CS}(s_1)| + |\mathcal{CS}(s_2)|$. Therefore, we have $|\mathcal{CS}(s)| = O(|s|)$.

Since each subterm s is assigned at most $O(|s|)$ formulas of size at most $O(|t|^2)$, the total size of all formulas in $\mathcal{CS}(s)$ is $O(|t|^3)$. This is the maximum size of the input for the subroutine that computes $\mathcal{CS}(s)$ for each s. Since each step of this subroutine is polynomial and the subroutine is applied $O(|t|)$ times, the overall complexity is also polynomial.

The only difference for the $*_{\mathcal{CS}}$-calculus is that case (5) must be restricted to terms of form $\underbrace{!\cdots!}_{n}c$, where c is a constant and $n \geq 0$ is an integer. Such checks can also be performed polynomially. □

It is interesting to observe what differentiates the finite number of axioms assigned to a constant in Theorem 16 from the finite number of schemes in Theorem 15 to cause a jump in complexity from P to NP. In both cases, the number of formulas/schemes assigned to s grows with the size of s. But the number of schemes grows exponentially, mainly due to case (3), which, as we have shown, does not increase the number of formulas at all. The reason is the behavior of the rule $*$A2: it generally fails when used on formulas and generally succeeds when applied to schemes. More importantly, in the case of formula assignment, the left premise of the rule together with the outer term of the right premise uniquely determine the successful conclusion, if any. On the contrary, in the case of scheme assignment, k schemes in the left premise and l schemes in the right in general, by way of unification, yield kl schemes assigned in the conclusion. This leads to an exponential blow-up and the necessity of non-deterministically choosing one scheme at a time.

As we will see in Sect. 4, P is the desired complexity for a reflected fragment. Unfortunately, reflected fragments $\mathsf{rJL}_{\mathcal{CS}}$ are not complete for any finite $\mathcal{CS}$. This is where comprehensive knowledge assertions come to the rescue. (They were already used in (Artemov and Kuznets, 2006) to prove that JL passes LOT with respect to the Hilbert proof system.)

Theorem 17. *Let* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$. *Then* crJL *is in P.*

Proof. Given the second equivalence in Lemma 13, the only difference from the situation in Theorem 16 is that here, $\mathcal{CS}$ is given as part of the input rather than being hardwired into the algorithm. Thus, given a knowledge assertion $\mathcal{A} = \bigwedge \mathcal{CS} \to t{:}F$, the number of axioms assigned by the algorithm to each constant occurring in t and their size are $O(|\mathcal{A}|)$ rather than $O(1)$. A careful scrutiny of the proof of Theorem 16 shows that in this case the size of formulas assigned to subterms of t and their number per subterm are both $O(|\mathcal{A}|^2)$. □

It is also possible to avoid an exponential blow-up while dealing with schemes.

Theorem 18. *Let* $\mathcal{CS}$ *be a schematically injective constant specification for a justification logic* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$. *Then,* $\mathsf{rJL}^-_{\mathcal{CS}}$ *is in P.*

Proof. It is sufficient to apply Krupski's original algorithm. Each branch of his non-deterministic algorithm results from choosing one axiom scheme for each constant and one of the two subterms for each occurrence of $s_1 + s_2$ in the given outer term t. Computation along each branch is then deterministic and polynomial. In our case, the constant specification is schematically injective, so there is only one axiom scheme assigned to each constant. In addition, there are no $+$'s in t. Thus, Krupski's algorithm run on any knowledge assertion from $\mathsf{rJL}^-_{\mathcal{CS}}$ is deterministic and polynomial. □

4 MAIN RESULTS

In this section, we will outline the relationship between the complexity of a reflected fragment of an epistemic system and the logical omniscience of this system.

Theorem 19. *Let* rL *be a complete reflected fragment of an epistemic system* L.

1. *If* rL *is in NP, there exists a proof system for* L *for which it passes LOT.*
2. *If* rL *is in P, there exists a proof system for* L *for which it passes SLOT.*

Proof. In both cases, there must exist a (non-)deterministic Turing machine M for deciding the given complete reflected fragment.

1. In the case of the nondeterministic machine M, we construct the proof system as follows: each proof consists of an element $\mathcal{A} \in r\mathcal{L}$ followed by a sequence of choices made by M given $\mathcal{A}$ as its input. Clearly, a deterministic polynomial-time Turing machine E can emulate the non-deterministic M given the set of choices along one of M's branches. This deterministic machine E outputs $OK(\mathcal{A})$ if the corresponding branch of M's computation has been successful. Otherwise, E outputs a fixed valid formula. Since $OK(\cdot)$ is polynomially computable, the function computed by E is, too. Moreover, E is a function onto L due to completeness of rL. It remains to note that for each valid knowledge assertion $\mathcal{A} \in \mathsf{rL}$, there must exist at least one successful run of M, polynomial in $|\mathcal{A}|$ and, as such, involving only polynomially many choices. Therefore, there exists a proof of $OK(\mathcal{A})$ in the proof system we have constructed, polynomial in $|\mathcal{A}|$.

2. In the case of the deterministic Turing machine M, the proof system can be made even simpler. The proofs are just knowledge assertions from $r\mathcal{L}$. Given such a knowledge assertion $\mathcal{A} \in r\mathcal{L}$, E first runs the given M to determine whether $\mathcal{A}$ is valid. If M succeeds, E outputs $OK(\mathcal{A})$. Otherwise, E outputs a fixed valid formula. Again, this is clearly a proof system. Since a proof of $OK(\mathcal{A})$ is nothing but $\mathcal{A}$, finding the proof of $OK(\mathcal{A})$ given $\mathcal{A}$ is trivial, while verifying that it is a proof can be done in polynomial time by the given deterministic Turing machine M. □

Corollary 20.

1. *Each* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$ *as an epistemic system with comprehensive reflected fragment* crJL *passes SLOT (with respect to a certain proof system).*

2. *If* $\mathcal{CS}$ *is a schematically injective and axiomatically appropriate constant specification for* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$*, then* $\mathsf{JL}_{\mathcal{CS}}$ *as an epistemic system with* $+$*-free reflected fragment* $\mathsf{rJL}^{-}_{\mathcal{CS}}$ *passes SLOT (with respect to a certain proof system).*

3. *If* $\mathcal{CS}$ *is a schematic axiomatically appropriate constant specification for* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JD}, \mathsf{JT}, \mathsf{J4}, \mathsf{JD4}, \mathsf{LP}\}$*, then* $\mathsf{JL}_{\mathcal{CS}}$ *as an epistemic system with simple reflected fragment* $\mathsf{rJL}_{\mathcal{CS}}$ *passes LOT (with respect to a certain proof system).*

In the last two statements, we assume that $\mathcal{CS}(\cdot)$ *is computable in polynomial time.*

Proof. The first two statements follow from Theorem 19.2 combined with Theorem 17 for the first, or Theorem 18 for the second. The third statement follows from Theorems 19.1 and 15. □

The converse to Theorem 19 does not in general hold. The fact that an epistemic proof system passes (S)LOT enables us to guess/find proofs of $OK(\mathcal{A})$ feasible in $|\mathcal{A}|$. But guessing/computing $\mathcal{A} \in OK^{-1}(F)$ for a given formula F may not be feasible or even possible. Thus, given F and the non-logically omniscient proof system, it is unclear how to obtain a proof of F. However, the following partial converse holds, an instance of which is Theorem 6.

Theorem 21 (Partial converse to Theorem 19)**.** *Let* L *be an epistemic system in language* $\mathcal{L}$ *with a complete reflected fragment* rL.

1. *Let* L *pass LOT with respect to some proof system. If for some polynomial* P,
$$|\mathcal{A}| \leq P(|\,OK(\mathcal{A})|) \qquad (6)$$
for all $\mathcal{A} \in r\mathcal{L}$*, then* L *is in NP.*
2. *Let* L *pass SLOT with respect to some proof system. If there exists a function* $\mathsf{K}\colon \mathcal{L} \to r\mathcal{L}$*, computable by a deterministic polynomial algorithm such that, for any valid formula* $F \in \mathsf{L}$*, it outputs a valid knowledge assertion about* F*, i.e.,* $\mathsf{K}(F) \in OK^{-1}(F) \cap \mathsf{rL}$ *for any* $F \in \mathsf{rL}$*, then* L *is in P.*

Proof. 1. By completeness of rL, for every valid $F \in \mathsf{L}$, there must be a valid $\mathcal{A} \in \mathsf{rL}$ such that $OK(\mathcal{A}) = F$. Since L passes LOT, there must be a proof of F polynomial in the size of this $\mathcal{A}$. But according to (6), $|\mathcal{A}|$ itself is polynomial in $|F|$. Hence, there is a proof of F polynomial in $|F|$ that can be non-deterministically guessed and then deterministically verified by the given proof system.

2. Given $F \in \mathcal{L}$, first, $\mathcal{A} = \mathsf{K}(F)$ is computed. Then, the polynomial algorithm provided by SLOT is used to construct a proof of F from $\mathcal{A}$. If F is valid, then so is $\mathcal{A}$; therefore, the algorithm outputs an actual proof of F that can be verified by the given proof system. Thus, if the verification is not successful, it means that F is not valid. □

5 CONCLUSIONS

Both Logical Omniscience Test (LOT) and Strong Logical Omniscience Test (SLOT) label agents whose knowledge is described by epistemic modal logics as logically omniscient (given some widely accepted complexity conjectures), whereas agents corresponding to various systems of justification logic with natural constant specifications have been proven to be free of the logical omniscience defect. These results are consistent with intuition and provide an argument in favor of adopting LOT and SLOT as reasonable logical omniscience tests.

The proposed theory demonstrates where to look for practical solutions to the logical omniscience problem: find justification languages, perhaps model specific, with concise annotations of reasons that would yield feasible witnesses of knowledge. Consider systems that contain both justified and the usual modal-style presentation of knowledge; use modal epistemic operators for 'potential knowability' and justified knowledge assertions for 'real knowledge'; see (Artemov and Nogina, 2005).

The idea to view Logical Omniscience as a complexity problem can, perhaps, be applied to other presentations of knowledge as well.

Acknowledgments

S. Artemov is supported in part by National Science Foundation Grant 0830450; R. Kuznets is supported by Swiss National Science Foundation Grant 200021–117699. We thank Jonathan Zvesper and the audience of PALMYR VIII for their inspiring questions and remarks. We are grateful to the anonymous referees for concise but informative reviews. We thank Karen Kletter and Galina Savukova for proofreading and editing various versions of this paper.

References

Natasha Alechina and Brian Logan. Ascribing beliefs to resource bounded agents. In *Proceedings of the first international joint conference on Autonomous Agents and Multiagent Systems, AAMAS 2002, Bologna, Italy, July 15–19, 2002, Part 2*, pages 881–888. ACM Press, 2002.

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.

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

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

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

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

Robert J. Aumann. Reasoning about knowledge in economics. In Joseph Y. Halpern, editor, *Theoretical Aspects of Reasoning about Knowledge, Proceedings of the 1986 Conference, March 19–22, 1986, Monterey, California*, page 251. Morgan Kaufmann, 1986. Abstract of invited talk.

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

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.

Stephen Cook and Robert Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In *Conference Record of Sixth Annual ACM Symposium on Theory of Computing, Papers Presented at the Symposium, Seattle, Washington, April 30–May 2, 1974*, pages 135–148. ACM Press, 1974.

Jennifer J. Elgot-Drapkin, Michael Miller, and Donald Perlis. Memory, reason, and time: The step-logic approach. In Robert Cummins and John Pollock, editors, *Philosophy and AI, Essays at the Interface*, chapter 4, pages 79–104. MIT Press, 1991.

Ronald Fagin and Joseph Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34(1):39–76, December 1987.

Ronald Fagin, Joseph Y. Halpern, and Moshe Y. Vardi. A nonstandard approach to the logical omniscience problem. *Artificial Intelligence*, 79(2):203–240, December 1995.

Joseph Y. Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, April 1992.

Joseph Y. Halpern and Riccardo Pucella. Dealing with logical omniscience. In Dov Samet, editor, *Proceedings of the 11th conference on Theoretical Aspects of Rationality and Knowledge (TARK'07)*, pages 169–176, Brussels, Belgium, June 25–27, 2007. ACM.

Jaakko Hintikka. *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press, 1962.

Jaakko Hintikka. Impossible possible worlds vindicated. *Journal of Philosophical Logic*, 4(3):475–484, August 1975.

Kurt Konolige. *A Deduction Model of Belief*. Research Notes in Artificial Intelligence. Morgan Kaufmann, 1986.

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

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.

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

Hector J. Levesque. A logic of implicit and explicit belief. In Ronald J. Brachman, editor, *Proceedings of the Fourth National Conference on Artificial Intelligence, AAAI 1984, Austin, Texas, August 6–10, 1984*, pages 198–202. AAAI Press, 1984.

Richard Montague. Universal grammar. *Theoria*, 36(3):373–398, 1970.

Robert C. Moore. Reasoning about knowledge in artificial intelligence. In Joseph Y. Halpern, editor, *Theoretical Aspects of Reasoning about Knowledge*, Proceedings of the 1986 Conference, March 19–22, 1986, Monterey, California, page 81. Morgan Kaufmann, 1986. Abstract of invited talk.

Yoram Moses. Resource-bounded knowledge. In Moshe Y. Vardi, editor, *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge, March 7–9, 1988, Pacific Grove, California*, pages 261–275. Morgan Kaufmann, 1988.

Eric Pacuit. A note on some explicit modal logics. In *Proceedings of the 5th Panhellenic Logic Symposium*, pages 117–125, Athens, Greece, July 25–28, 2005. University of Athens.

Rohit Parikh. Knowledge and the problem of logical omniscience. In Zbigniew W. Ras and Maria Zemankova, editors, *Methodologies for Intelligent Systems, Proceedings of the Second International Symposium, ISMIS 1987, Charlotte, North Carolina, USA, October 14–17, 1987*, pages 432–439. North-Holland/Elsevier, 1987.

Rohit Parikh. Logical omniscience. In Daniel Leivant, editor, *Logic and Computational Complexity, International Workshop LCC '94, Indianapolis, IN, USA, October 13–16, 1994, Selected Papers*, volume 960 of *Lecture Notes in Computer Science*, pages 22–29. Springer, 1995.

Rohit Parikh. Logical omniscience and common knowledge; WHAT do we know and what do WE know? In Ron van der Meyden, editor, *Proceedings of the Tenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK'05)*, pages 62–77, Singapore, June 10–12, 2005. National University of Singapore.

Rohit Parikh. Sentences, belief and logical omniscience, or what does deduction tell us? *The Review of Symbolic Logic*, 1(4):459–476, December 2008.

Pavel Pudlák. The lengths of proofs. In Samuel R. Buss, editor, *Handbook of Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, chapter VIII, pages 547–637. Elsevier, 1998.

Veikko Rantala. Impossible worlds semantics and logical omniscience. *Acta Philosophica Fennica*, 35:106–115, 1982.

Natalia Rubtsova. Evidence reconstruction of epistemic modal logic S5. In Dima Grigoriev, John Harrison, and Edward A. Hirsch, editors, *Computer Science — Theory and Applications, First International Computer Science Symposium in Russia, CSR 2006, St. Petersburg, Russia, June 8–12, 2006, Proceedings*, volume 3967 of *Lecture Notes in Computer Science*, pages 313–321. Springer, 2006.

Dana Scott. Advice in modal logic. In K. Lambert, editor, *Philosophical Problems in Logic*, pages 143–173. Reidel, 1970.

Hyun Song Shin and Timothy Williamson. Representing the knowledge of Turing machines. *Theory and Decision*, 37(1):125–146, July 1994.

Moshe Y. Vardi. On epistemic logic and logical omniscience. In Joseph Y. Halpern, editor, *Theoretical Aspects of Reasoning about Knowledge, Proceedings of the 1986 Conference, March 19–22, 1986, Monterey, California*, pages 293–305. Morgan Kaufmann, 1986.

Heinrich Wansing. A general possible worlds framework for reasoning about knowledge and belief. *Studia Logica*, 49(4):523–539, December 1990.