

WEAK ARITHMETICAL INTERPRETATIONS FOR THE LOGIC OF PROOFS

ROMAN KUZNETS AND THOMAS STUDER

ABSTRACT. Artemov established an arithmetical interpretation for the Logics of Proofs LP_{CS} , which yields a classical provability semantics for the modal logic S4 . The Logics of Proofs are parameterized by so-called constant specifications CS , stating which axioms can be used in the reasoning process, and the arithmetical interpretation relies on constant specifications being finite. In this paper, we remove this restriction by introducing weak arithmetical interpretations that are sound and complete for a wide class of constant specifications, including infinite ones. In particular, they interpret the full Logic of Proofs LP .

1. INTRODUCTION

The original motivation for developing the first justification logic, the Logic of Proofs, was to provide intuitionistic logic with an adequate provability semantics, i.e., a semantics that respects Brouwer’s fundamental idea, see, e.g., [TvD88a, TvD88b], that

- (1) intuitionistic truth means provability.

Heyting and Kolmogorov [Hey31, Hey34, Kol32] gave an explicit (but informal) definition of this notion of intuitionistic truth, which nowadays is known as Brouwer–Heyting–Kolmogorov (BHK) semantics for intuitionistic logic.

The BHK semantics is widely accepted as the intended semantics for intuitionistic logic. However, it is purely informal and does not provide a precise definition of intuitionistic truth. This was tackled by Gödel [Göd33] who introduced a modal calculus of classical provability (equivalent to S4) with the intended reading of $\Box F$ as *F is provable*. Gödel defined a translation $t(\cdot)$ from IL to S4 where the translation $t(F)$ of an intuitionistic formula F is given by

prefix each subformula of F with a \Box -modality.

He apparently considered this to be an appropriate formalization of Brouwer’s thesis (1). Gödel established that

$$\text{IL} \vdash F \quad \text{implies} \quad \text{S4} \vdash t(F) \text{ .}$$

He conjectured that the converse direction also holds, which was later shown by McKinsey and Tarski [MT48].

Both authors were partially supported by the Swiss National Science Foundation grant 200020-153169.

On the later stages of research, the first author was supported by the Austrian Science Fund (FWF) grants: Lise Meitner M 1770-N25, START Y 544-N23, and P 25417-G15.

However, the ultimate goal of providing a classical provability semantics for IL is not achieved by Gödel's translation because no precise semantics is given for the provability operator \Box . The situation can be depicted as follows:

$$\text{IL} \hookrightarrow \text{S4} \rightarrow \dots ??? \dots \rightarrow \text{classical proofs} .$$

Artemov [Art95, Art01] was able to give a formal provability semantics for S4 . He introduced the Logics of Proofs LP_{CS} , which are systems in the spirit of S4 but with explicit proof terms.

Artemov established a realization theorem, which provides an embedding $r(\cdot)$ of S4 into LP_{CS} (we describe formal conditions imposed on $r(\cdot)$ in Definition 6.3). Further, he developed a formal provability semantics for LP_{CS} , which gives us the following chain of faithful embeddings:

$$\text{IL} \hookrightarrow \text{S4} \rightarrow \text{LP}_{\text{CS}} \rightarrow \text{classical proofs} .$$

Hence, the Logics of Proofs LP_{CS} can be viewed as a formalization of the BHK semantics for intuitionistic propositional logic.

It is important that LP_{CS} is not one single logic but rather a family of logics parameterized by a so-called constant specification CS . Roughly speaking, the purpose of this constant specification is to state which axioms are available for the internalized reasoning process. Note that there exists the largest constant specification and LP denotes its corresponding Logic of Proofs.

Artemov's arithmetical semantics only works for finite CS , which, in a sense, is enough since each proof can only refer to finitely many axioms. In other words, for each formula F of the Logic of Proofs, there exists a finite constant specification CS such that

$$\text{LP} \vdash F \quad \implies \quad \text{LP}_{\text{CS}} \vdash F .$$

Since different proofs may require different finite constant specifications, we actually have the following statement about arithmetic validity (where \mathcal{L}_{\Box} and \mathcal{L}_{J} are the languages of modal logic and the Logic of Proofs respectively):

Theorem 1.1 ([Art95, Art01]). There exists a realization $r: \mathcal{L}_{\Box} \rightarrow \mathcal{L}_{\text{J}}$ such that for each \mathcal{L}_{\Box} -formula F

$$\text{S4} \vdash F \quad \iff \quad \begin{array}{l} r(F) \text{ is arithmetically CS-valid} \\ \text{for some finite constant specification CS.} \end{array}$$

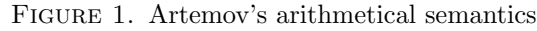
In other words, the constant specification and, hence, the notion of validity depend on the formula F . This is depicted in Figure 1, where two different formulas F and G lead to two arithmetical embeddings $*_{\text{CS}_F}$ and $*_{\text{CS}_G}$ that are based on different constant specifications.¹

It is the aim of this paper to establish an arithmetical interpretation result where CS does not depend on the formula. Namely, we will show the following:

Theorem 1.2. Let CS be a primitive recursive, axiomatically appropriate, and schematic constant specification. There exists a realization r such that for each \mathcal{L}_{\Box} -formula F

$$\text{S4} \vdash F \quad \iff \quad r(F) \text{ is weakly arithmetically CS-valid.}$$

¹Strictly speaking, it is always possible to use the same constant specification and, hence, the same arithmetical embedding for any pair of formulas or, indeed, for any finite set of formulas. However, infinitely many theorems of IL require the use of infinitely many arithmetical embeddings.


$$\text{IL} \vdash F \hookrightarrow t \rightarrow \text{S4} \vdash t(F) \multimap r \rightarrow \text{LP} \vdash r(t(F)) \multimap * \rightarrow \text{PA}$$

Of course, there is a price to pay for removing the dependence on the constant specification. Artemov’s arithmetical semantics interprets the operations on evidence terms by computable functions on codes for proofs. It is open whether this is also possible in our semantics, and, hence, we call it *weak*.

$$t ::= c_i \mid x_i \mid (t \cdot t) \mid (t + t) \mid !t \quad .$$
$$F ::= p_i \mid \perp \mid (F \rightarrow F) \mid t:F \quad .$$

The axioms of LP consist of all instances of the following schemes:

A *constant specification* CS for LP is any subset

For a constant specification CS the deductive system LP_{CS} is the Hilbert system given by the axioms above and by the rules modus ponens and axiom necessitation:

$$\frac{A \quad A \rightarrow B}{B} \text{ (MP) } , \quad \frac{(c, A) \in \mathbf{CS}}{c:A} \text{ (AN) } .$$

Definition 2.1 (Axiomatic appropriateness, schematicness, almost schematicness). A constant specification CS is called

- (i) *axiomatically appropriate* if, for each axiom A of LP , there is a constant c such that $(c, A) \in \text{CS}$;
- (ii) *schematic* if it satisfies the following property: for each constant c , the set of axioms $\{A \mid (c, A) \in \text{CS}\}$ consists of all instances of one or several (possibly zero) axiom schemes of LP ;
- (iii) *almost schematic* if it is the union of a schematic and a finite constant specifications.

In this paper, justification terms represent formal mathematical proofs. There are, however, many more forms of justifications that can be considered in a general epistemic setting, such as direct observation, public announcements, or private communication. Systems of explicit justifications, together with a corresponding possible world semantics, make it possible to analyze (dynamic) epistemic situations in a fine-grained way and to formalize and discuss many epistemic problems and puzzles [Art08, Art12, BKR⁺10, BKS11, BRS14, DK14, KS12, KS13, Ren11].

3. DECIDABILITY FOR LP_{CS}

Generated models are models for LP_{CS} where the evidence relation is generated by the least fixed point construction. To inductively construct this least fixed point, we need a monotone operator, which is given as follows.

Definition 3.1 (Evidence closure). Let $\mathcal{B} \subseteq \text{Tm} \times \mathcal{L}_{\text{J}}$. For a set $X \subseteq \text{Tm} \times \mathcal{L}_{\text{J}}$ we define $\text{cl}_{\mathcal{B}}(X) \subseteq \text{Tm} \times \mathcal{L}_{\text{J}}$ as follows:

- (i) if $(t, A) \in \mathcal{B}$, then $(t, A) \in \text{cl}_{\mathcal{B}}(X)$;
- (ii) if $(s, A) \in X$ or $(t, A) \in X$, then $(s + t, A) \in \text{cl}_{\mathcal{B}}(X)$;
- (iii) if $(s, A) \in X$ and $(t, A \rightarrow B) \in X$, then $(t \cdot s, B) \in \text{cl}_{\mathcal{B}}(X)$;
- (iv) if $(t, A) \in X$, then $(!t, t:A) \in \text{cl}_{\mathcal{B}}(X)$.

Note that $\text{cl}_{\mathcal{B}}$ is a monotone operator on $\text{Tm} \times \mathcal{L}_{\text{J}}$. Hence, it has the least fixed point $\mathcal{E}(\mathcal{B})$, which we call the *minimal evidence relation* over \mathcal{B} .

Definition 3.2 (Generated Model). A *generated model* is a pair $\mathcal{M} = (\text{val}, \mathcal{B})$ where $\text{val} \subseteq \text{Prop}$ and $\mathcal{B} \subseteq \text{Tm} \times \mathcal{L}_{\text{J}}$. For a constant specification CS , the generated model \mathcal{M} is called a *generated CS-model* if $\text{CS} \subseteq \mathcal{B}$.

Definition 3.3 (Truth in generated models). Let $\mathcal{M} = (\text{val}, \mathcal{B})$ be a generated model and D be a formula. We define the relation $\mathcal{M} \Vdash D$ as follows:

- (i) $\mathcal{M} \not\Vdash \perp$;
- (ii) $\mathcal{M} \Vdash p_i$ iff $p_i \in \text{val}$;
- (iii) $\mathcal{M} \Vdash A \rightarrow B$ iff $\mathcal{M} \not\Vdash A$ or $\mathcal{M} \Vdash B$;
- (iv) $\mathcal{M} \Vdash t:A$ iff $(t, A) \in \mathcal{E}(\mathcal{B})$ and $\mathcal{M} \Vdash A$.

Definition 3.4 (Finitary model). Let CS be an almost schematic constant specification. Let $\mathcal{C} \subseteq \text{Tm} \times \mathcal{L}_{\text{J}}$ be finite and set $\mathcal{B} := \text{CS} \cup \mathcal{C}$. Further let val be a finite valuation, i.e., a finite subset of Prop . Then we call the generated CS -model $\mathcal{M} = (\text{val}, \mathcal{B})$ a *finitary CS-model*.

A formula D is *valid with respect to finitary CS-models* if $\mathcal{M} \Vdash D$ for all finitary CS -models \mathcal{M} .

We have soundness and completeness of LP_{CS} with respect to finitary CS -models:

Theorem 3.5 ([Kuz08, BKS13]). Let CS be an almost schematic constant specification. For each formula $F \in \mathcal{L}_J$,

$$\text{LP}_{\text{CS}} \vdash F \quad \text{iff} \quad F \text{ is valid with respect to finitary CS-models.}$$

Remark 3.6. CS is restricted to be almost schematic only because this restriction is present in the definition of finitary models, which are the key to establishing decidability for the Logics of Proofs. Soundness holds for arbitrary CS for an extended class of models.

Theorem 3.7. Let CS be a primitive recursive and almost schematic constant specification. For each finitary CS -model, its satisfaction relation is primitive recursive.

Proof. By a careful examination of the decision algorithm for the satisfaction relation from [Kuz08, Corollary 4.4.8]. \square

4. PEANO ARITHMETIC

In this section, we introduce all notions and concepts of Peano Arithmetic PA that will be needed later in order to present arithmetical interpretations for the Logics of Proofs. We employ a formulation of PA that includes symbols for all primitive recursive functions and relations.

The *language \mathcal{L}_{PA} of arithmetic* is the language of first-order logic with countably many (individual) variables, with the logical symbols \perp , \rightarrow , and \forall , and with the following non-logical symbols:

- (i) an n -ary function symbol \underline{f} for each n -ary primitive recursive function f ;
- (ii) an n -ary relation symbol \underline{R} for each n -ary primitive recursive relation R .

We use x, y, z, \dots to denote individual variables of \mathcal{L}_{PA} and hope the reader is able to distinguish them from the justification variables of \mathcal{L}_J . Further, we denote formulas of \mathcal{L}_{PA} by ϕ, ψ, \dots . A *sentence* is a formula without free occurrences of variables.

For each natural number n , we use its standard representation $\underbrace{s(s(\dots s(0)\dots))}_n$

in the language \mathcal{L}_{PA} , call it a *numeral*, and denote it by \underline{n} . When working in \mathcal{L}_{PA} , we will often use \underline{f} for \underline{f} , \underline{R} for \underline{R} , and \underline{n} for \underline{n} whenever the exact typification can be inferred from the context. In particular, we often write $=$ for \equiv .

Peano Arithmetic PA is given in the language \mathcal{L}_{PA} . It comprises the axioms and rules of first-order predicate logic, equality axioms for the primitive recursive relation $=$, defining axioms for all primitive recursive functions and relations, the standard arithmetical axioms for the successor function, and the induction axioms. As usual, we write $\text{PA} \vdash \phi$ if the formula ϕ is provable in PA .

If ϕ is a sentence, we write $\mathbb{N} \models \phi$ to say that ϕ is true in the standard model \mathbb{N} of the natural numbers. In the following, we assume that PA is sound with respect to \mathbb{N} : for all \mathcal{L}_{PA} -sentences ϕ ,

$$(2) \quad \text{PA} \vdash \phi \quad \text{implies} \quad \mathbb{N} \models \phi .$$

Remark 4.1. This assumption is standard (and, indeed, mundane) for both the Logic of Proofs and Provability Logic to the point of not being formulated as a stand-alone lemma but being used implicitly or mentioned in passing in an introduction, cf., for instance, statements

- “It is obvious that *provably valid* yields *valid*” ([Art01, p. 13]);
- “[A]ll sentences provable in Peano Arithmetic are indeed true about the natural numbers [...]” ([Ver14, Sect. 1]).

Definition 4.2 (Provable equivalence). \mathcal{L}_{PA} -formulas ϕ and ψ are called *provably equivalent* if

$$\text{PA} \vdash \phi \leftrightarrow \psi .$$

Now we can define several important classes of \mathcal{L}_{PA} -formulas.

Definition 4.3 (Primitive recursive, Σ_1 -, and Δ_1 -formulas).

- (i) A *standard primitive recursive formula* is an \mathcal{L}_{PA} -formula of the form

$$\underline{R}(t_0, \dots, t_{n-1})$$

where R is an n -ary primitive recursive relation.

- (ii) A *standard Σ_1 -formula* is an \mathcal{L}_{PA} -formula of the form

$$\exists x \phi$$

where ϕ is a standard primitive recursive formula.

- (iii) An \mathcal{L}_{PA} -formula ϕ is *provably Σ_1* if there exists a standard Σ_1 -formula ψ such that ϕ and ψ are provably equivalent.
- (iv) An \mathcal{L}_{PA} -formula ϕ is *provably Δ_1* if both ϕ and $\neg\phi$ are provably Σ_1 .

Lemma 4.4 ([Boo93]). The class of provably Δ_1 formulas is closed under Boolean connectives and under substitutions of terms for variables. All standard primitive recursive formulas are provably Δ_1 .

It is a key fact that PA is complete for provably Σ_1 sentences.

Lemma 4.5 ([Boo93]).

- (i) Let ϕ be a provably Σ_1 sentence. If $\mathbb{N} \models \phi$, then $\text{PA} \vdash \phi$.
- (ii) Let ϕ be a sentence such that $\neg\phi$ is provably Σ_1 . If $\mathbb{N} \not\models \phi$, then $\text{PA} \vdash \neg\phi$.

Formulating this lemma for provably Δ_1 sentences yields the following corollary, which we are going to apply often.

Theorem 4.6. Let ϕ be a provably Δ_1 sentence.

- (i) If $\mathbb{N} \models \phi$, then $\text{PA} \vdash \phi$.
- (ii) If $\mathbb{N} \not\models \phi$, then $\text{PA} \vdash \neg\phi$.

To be able to talk within PA about formulas and proofs of PA, as well as about formulas and terms of LP, we need a so-called Gödel numbering of $\mathcal{L}_{\text{PA}} \cup \mathcal{L}_{\text{J}}$, i.e., an assignment of a numerical code $\ulcorner \phi \urcorner \in \mathbb{N}$ to each formula $\phi \in \mathcal{L}_{\text{PA}} \cup \mathcal{L}_{\text{J}}$ and of a numerical code $\ulcorner t \urcorner \in \mathbb{N}$ to each term $t \in \mathcal{L}_{\text{PA}} \cup \mathcal{L}_{\text{J}}$. As mentioned above, when working in \mathcal{L}_{PA} , we often use m for \underline{m} . Accordingly, whenever $\ulcorner \phi \urcorner$ occurs within an \mathcal{L}_{PA} -formula, what we mean is, of course, the \mathcal{L}_{PA} -term $\underline{\ulcorner \phi \urcorner}$.

Making use of the Gödel numbering, we can state the Diagonalization lemma, which is crucial for defining arithmetical interpretations for LP_{CS} .

Lemma 4.7 (Diagonalization). Let $\psi(y, x_0, \dots, x_{n-1})$ be an \mathcal{L}_{PA} -formula. There exists an \mathcal{L}_{PA} -formula $\phi(x_0, \dots, x_{n-1})$ such that

- $\text{PA} \vdash \phi(x_0, \dots, x_{n-1}) \leftrightarrow \psi(\ulcorner \phi(x_0, \dots, x_{n-1}) \urcorner, x_0, \dots, x_{n-1})$;
- $\phi(x_0, \dots, x_{n-1})$ is provably Δ_1 if $\psi(y, x_0, \dots, x_{n-1})$ is; and

- $\phi(x_0, \dots, x_{n-1})$ contains a relation symbol \underline{R} if $\psi(y, x_0, \dots, x_{n-1})$ does.

Proof. The first two conditions form the classical formulation of the diagonalization lemma (see, e.g., [Boo93]). The third condition easily follows from a close observation of the proof in [Boo93]. \square

Last but not least we will use the notion of a proof predicate:

Definition 4.8 (Proof predicate). A *proof predicate* is a provably Δ_1 formula $\text{Prf}(x, y)$ with no free occurrences of variables other than x and y such that for every \mathcal{L}_{PA} -sentence ϕ we have

$$(3) \quad \text{PA} \vdash \phi \iff \mathbb{N} \models \text{Prf}(\underline{n}, \ulcorner \phi \urcorner) \text{ for some natural number } n.$$

We will not formally establish the existence of proof predicates for PA. A detailed formal construction of a proof predicate is presented, e.g., in [Gir87]. For the rest of this paper, we simply assume that we are given the usual primitive recursive proof predicate $\text{Proof}(x, y)$ for Peano Arithmetic.

5. WEAK ARITHMETICAL INTERPRETATION

Definition 5.1 (Weak arithmetical interpretation). A *weak arithmetical interpretation* is a pair $(*, \text{Prf})$ such that

- (i) $*$ maps atomic propositions of \mathcal{L}_{J} to sentences of \mathcal{L}_{PA} ;
- (ii) $*$ maps evidence terms of \mathcal{L}_{J} to numerals of \mathcal{L}_{PA} ;
- (iii) Prf is a proof predicate; and
- (iv) for all evidence terms s and t ,

$$(4) \quad \mathbb{N} \models \text{Prf}(s^*, \ulcorner \phi \rightarrow \psi \urcorner) \wedge \text{Prf}(t^*, \ulcorner \phi \urcorner) \rightarrow \text{Prf}((s \cdot t)^*, \ulcorner \psi \urcorner),$$

$$(5) \quad \mathbb{N} \models \text{Prf}(s^*, \ulcorner \phi \urcorner) \vee \text{Prf}(t^*, \ulcorner \phi \urcorner) \rightarrow \text{Prf}((s + t)^*, \ulcorner \phi \urcorner), \text{ and}$$

$$(6) \quad \mathbb{N} \models \text{Prf}(s^*, \ulcorner \phi \urcorner) \rightarrow \text{Prf}(!s^*, \ulcorner \text{Prf}(s^*, \ulcorner \phi \urcorner) \urcorner).$$

We extend the mapping $*$ to all formulas of \mathcal{L}_{J} by setting

$$(t:F)^* := \text{Prf}(t^*, \ulcorner F^* \urcorner), \quad \perp^* := \perp, \quad \text{and} \quad (F \rightarrow G)^* := F^* \rightarrow G^*.$$

If there is no need to explicitly mention the proof predicate Prf , we denote the weak arithmetical interpretation $(*, \text{Prf})$ by $*$.

Given a CS, a weak arithmetical interpretation $*$ is called a *weak arithmetical CS-interpretation* if for each $(c, A) \in \text{CS}$ we have $\mathbb{N} \models (c:A)^*$. An \mathcal{L}_{J} -formula F is *weakly arithmetically CS-valid* if $\text{PA} \vdash F^*$ for all weak arithmetical CS-interpretations $*$.

Theorem 5.2 (Weak Arithmetical Soundness). Let CS be a constant specification and F be an \mathcal{L}_{J} -formula. Then

$$(7) \quad \text{LP}_{\text{CS}} \vdash F \text{ implies } F \text{ is weakly arithmetically CS-valid.}$$

Proof. Suppose $*$ is a weak arithmetical CS-interpretation. We show (7) by induction on the LP_{CS} -derivation of F . We use the fact that F^* is always a sentence.

If F is a classical tautology, then so is F^* and we trivially have $\text{PA} \vdash F^*$.

If F is an instance of **j**, **j+**, or **j4**, then $\mathbb{N} \models F^*$ follows from (4)–(6). Since F is a Boolean combination of $s:G$ -type formulas, F^* is a Boolean combination of substitution instances of provably Δ_1 formulas and is itself provably Δ_1 by Lemma 4.4. It follows by Theorem 4.6 that $\text{PA} \vdash F^*$.

If F is the conclusion of an instance of axiom necessitation, then F has the form $c:A$ where $(c, A) \in \text{CS}$. We have $\mathbb{N} \models (c:A)^*$ because $*$ is a CS-interpretation. Since $(c:A)^*$ is a substitution instance of a provably Δ_1 formula, by Lemma 4.4 it is provably Δ_1 itself, and $\text{PA} \vdash (c:A)^*$ by Theorem 4.6.

The only axiom instances F where we cannot be sure that F^* is provably Δ_1 are instances of **jt**. If $F = t:A \rightarrow A$, then $F^* = \text{Prf}(t^*, \ulcorner A^* \urcorner) \rightarrow A^*$. We distinguish two cases, depending on whether the sentence $\text{Prf}(t^*, \ulcorner A^* \urcorner)$ is true or false:

- $\mathbb{N} \models \text{Prf}(t^*, \ulcorner A^* \urcorner)$. By (3), we find $\text{PA} \vdash A^*$ and, thus,

$$\text{PA} \vdash \text{Prf}(t^*, \ulcorner A^* \urcorner) \rightarrow A^* .$$

- $\mathbb{N} \not\models \text{Prf}(t^*, \ulcorner A^* \urcorner)$. In this case, $\text{Prf}(t^*, \ulcorner A^* \urcorner)$ is a false provably Δ_1 sentence, meaning that $\text{PA} \vdash \neg \text{Prf}(t^*, \ulcorner A^* \urcorner)$ by Theorem 4.6. Therefore, again

$$\text{PA} \vdash \text{Prf}(t^*, \ulcorner A^* \urcorner) \rightarrow A^* .$$

Finally, if F is the conclusion of an instance of modus ponens, the claim follows by the induction hypothesis and the fact that $*$ distributes through implication. \square

In the remainder of this section we show completeness of LP_{CS} with respect to any weak arithmetical CS-interpretation where CS is a primitive recursive and almost schematic constant specification. In order to obtain this result, we will establish the following property:

Lemma 5.3. For each finitary CS-model \mathcal{M}_{fin} , there exists a weak arithmetical CS-interpretation $*$ such that for all \mathcal{L}_J -formulas G

$$(8) \quad \mathcal{M}_{\text{fin}} \models G \quad \text{implies} \quad \mathbb{N} \models G^* .$$

Weak arithmetical completeness easily follows from Lemma 5.3.

Theorem 5.4 (Weak arithmetical completeness). Let CS be a primitive recursive and almost schematic constant specification. For any formula F of \mathcal{L}_J we have

$$F \text{ is weakly arithmetically CS-valid} \quad \text{implies} \quad \text{LP}_{\text{CS}} \vdash F .$$

Proof. Assume that $\text{LP}_{\text{CS}} \not\vdash F$. By Theorem 3.5, there exists a finitary CS-model \mathcal{M}_{fin} with $\mathcal{M}_{\text{fin}} \not\models F$. Thus, $\mathcal{M}_{\text{fin}} \models \neg F$. By Lemma 5.3, there is a weak arithmetical CS-interpretation $*$ such that $\mathbb{N} \models (\neg F)^*$, i.e., $\mathbb{N} \models \neg(F^*)$. Therefore, $\mathbb{N} \not\models F^*$, which implies $\text{PA} \not\vdash F^*$ by soundness (2) of PA. Hence, F is not weakly arithmetically CS-valid. \square

The complicated part is to establish (8). For the rest of this section, we assume that we are given

- (i) a primitive recursive and almost schematic constant specification CS and
- (ii) a finitary CS-model \mathcal{M}_{fin} .

Further, we assume that the Gödel numbering of the union of \mathcal{L}_{PA} and \mathcal{L}_J is injective, i.e.,

$$\ulcorner E_1 \urcorner = \ulcorner E_2 \urcorner \quad \text{if and only if} \quad E_1 \equiv E_2$$

for arbitrary expressions E_1 and E_2 .

We first have to decide what objects should serve as ‘proofs’ in our arithmetical interpretation. There will be two sources of ‘proofs’:

- (i) To begin with, all usual proofs will be ‘proofs.’ This guarantees that the direction from left to right in (3) is satisfied.

- (ii) The second source of ‘proofs’ are the evidence terms of \mathcal{L}_J . Every term t is a ‘proof’ for all formulas B for which $\mathcal{M}_{\text{fin}} \Vdash t:B$.

To take care of the usual proofs, we make use of the usual primitive recursive proof predicate $\text{Proof}(x, y)$ for Peano Arithmetic. Without loss of generality we assume that $\mathbb{N} \not\models \text{Proof}(\ulcorner s \urcorner, \underline{k})$ for any evidence term s of \mathcal{L}_J and any natural number k .

In order to deal with the evidence terms, we denote by $\text{Prf}(x, y)$ a formula with no free variables other than x and y that will be chosen later based on its desired properties that we are going to discuss now. For any such $\text{Prf}(x, y)$, we can define an auxiliary translation † from \mathcal{L}_J -formulas to \mathcal{L}_{PA} -sentences as follows:

$$\begin{aligned} p^\dagger &:= \begin{cases} \ulcorner p \urcorner = \ulcorner p \urcorner & \text{if } \mathcal{M}_{\text{fin}} \Vdash p, \\ \neg(\ulcorner p \urcorner = \ulcorner p \urcorner) & \text{otherwise} \end{cases} \quad \text{for any atomic proposition } p; \\ (t:F)^\dagger &:= \text{Prf}(\ulcorner t \urcorner, \ulcorner F^\dagger \urcorner); \\ \perp^\dagger &:= \perp; \\ (F \rightarrow G)^\dagger &:= F^\dagger \rightarrow G^\dagger. \end{aligned}$$

Obviously, atomic propositions that hold in \mathcal{M}_{fin} are translated to true provably Δ_1 sentences and atomic propositions that do not hold in \mathcal{M}_{fin} are translated to false provably Δ_1 sentences. The former are provable and the latter are refutable in PA.

Remark 5.5. We need the translation † to be injective. Therefore, simply putting

$$p^\dagger := \begin{cases} 0 = 0 & \text{if } \mathcal{M}_{\text{fin}} \Vdash p, \\ 0 = 1 & \text{otherwise} \end{cases}$$

would not be sufficient.

Lemma 5.6. If the formula $\text{Prf}(x, y)$ contains some relation symbol \underline{R} other than $=$, i.e., some relation symbol not occurring in the † -translation outside of Prf , then this translation is injective, in other words,

$$(9) \quad F^\dagger \equiv G^\dagger \quad \text{implies} \quad F \equiv G.$$

Proof. We assume $F^\dagger \equiv G^\dagger$ and show (9) by induction on the structure of the \mathcal{L}_J -formula F .

- (i) F is an atomic proposition. By the definition of † , G must also be an atomic proposition and, by the injectivity of the Gödel numbering, G must be the same atomic proposition as F .
- (ii) F is \perp . By the definition of † , it is clear that $G \equiv \perp$.
- (iii) F is a formula $s:F_1$. Then G must be of the form $t:G_1$. Indeed, by the definition of † , it is obvious that G can be neither \perp nor an atomic proposition. Suppose towards a contradiction the only remaining possibility, i.e., that $G \equiv G_1 \rightarrow G_2$. Since $(s:F_1)^\dagger \equiv \text{Prf}(\underline{k}, \underline{n})$ for suitable k and n , the sentence $(s:F_1)^\dagger \equiv G_1^\dagger \rightarrow G_2^\dagger$ would contain the symbol \underline{R} , meaning that G_1^\dagger or G_2^\dagger would contain a subformula of the form $\text{Prf}(\underline{k}_1, \underline{n}_1)$. Let l be the number of occurrences of \rightarrow in $\text{Prf}(x, y)$. Substitutions of terms for variables do not affect Boolean connectives, hence both $(s:F_1)^\dagger$ and $\text{Prf}(\underline{k}_1, \underline{n}_1)$ would contain exactly l occurrences of \rightarrow each. But then $G_1^\dagger \rightarrow G_2^\dagger$ would contain at least $l + 1$ occurrences of \rightarrow . This contradiction shows the impossibility

- of $(s:F_1)^\dagger \equiv G_1^\dagger \rightarrow G_2^\dagger$. Therefore, $G \equiv t:G_1$. By the induction hypothesis and injectivity of the Gödel numbering we conclude $s \equiv t$ and $F_1 \equiv G_1$.
- (iv) F is $F_1 \rightarrow F_2$. By the same argument as in (iii), we have $G \equiv G_1 \rightarrow G_2$. By the induction hypothesis, $F_1 \equiv G_1$ and $F_2 \equiv G_2$. \square

For any formula $\text{Prf}(x, y)$ that yields an injective \dagger , it can be shown by using the standard techniques for Gödel numbering that binary functions $\text{dag}(x, y)$ and $\text{undag}(x, y)$ such that

$$\text{dag}(\ulcorner B^\neg, \ulcorner \text{Prf}(x, y)^\neg \urcorner) = \ulcorner B^\dagger^\neg \urcorner \quad \text{and} \quad \text{undag}(\ulcorner B^\dagger^\neg, \ulcorner \text{Prf}(x, y)^\neg \urcorner) = \ulcorner B^\neg \urcorner$$

are primitive recursive and our language contains the corresponding function symbols dag and undag (it does not matter much how these functions are defined on inputs that are not Gödel numbers of such formulas, e.g., they can be assumed to be constant on all other inputs). Note that the functions dag and undag are supposed to take the Gödel number of $\text{Prf}(x, y)$ as a parameter. Hence, unlike the translation \dagger , these functions do not depend on $\text{Prf}(x, y)$. This means, in particular, that the way undag is defined does not depend on whether \dagger is injective. The above property, however, is only guaranteed for injective \dagger 's. Note also that dag and undag do depend on the chosen model \mathcal{M}_{fin} .

By Theorem 3.7, the satisfaction relation for \mathcal{M}_{fin} is primitive recursive. Therefore, there is a binary relation symbol $\underline{\text{Jus}}$ such that

$$\mathbb{N} \models \underline{\text{Jus}}(\underline{n}, \underline{k}) \quad \Longleftrightarrow \quad \begin{array}{l} \text{there is a term } s \text{ and a formula } F \text{ such that} \\ n = \ulcorner s^\neg \urcorner, \text{ and } k = \ulcorner F^\neg \urcorner, \text{ and } \mathcal{M}_{\text{fin}} \models s:F. \end{array}$$

As mentioned in Lemma 4.4, $\underline{\text{Jus}}(x, y)$ is a provably Δ_1 formula. Using Lemma 4.7, we now define the desired formula $\text{Prf}(x, y)$ to satisfy

$$(10) \quad \text{PA} \vdash \text{Prf}(x, y) \leftrightarrow \text{Proof}(x, y) \vee \underline{\text{Jus}}(x, \text{undag}(y, \ulcorner \text{Prf}(x, y)^\neg \urcorner)) .$$

Moreover, since $\text{Proof}(x, y) \vee \underline{\text{Jus}}(x, \text{undag}(y, z))$ is clearly provably Δ_1 , so is our $\text{Prf}(x, y)$ by Lemma 4.7. It follows by Lemma 4.4 that F^\dagger is provably Δ_1 for each F .

Thus, by soundness (2) of PA, for the universal closure of (10),

$$(11) \quad \mathbb{N} \models \forall x \forall y \left(\text{Prf}(x, y) \leftrightarrow \text{Proof}(x, y) \vee \underline{\text{Jus}}(x, \text{undag}(y, \ulcorner \text{Prf}(x, y)^\neg \urcorner)) \right) .$$

Further, (9) holds by Lemma 4.7 because the formula $\text{Prf}(x, y)$ contains the relation symbol $\underline{\text{Jus}}$. It follows that undag really performs the inverse translation, i.e., informally we have

$$\text{Prf}(x, y) \quad \Longleftrightarrow \quad \begin{array}{l} \text{Proof}(x, y) \\ \text{or} \\ \text{there is a term } s \text{ and a formula } F \text{ such that} \\ x = \ulcorner s^\neg \urcorner, \text{ and } y = \ulcorner F^\dagger^\neg \urcorner, \text{ and } \mathcal{M}_{\text{fin}} \models s:F. \end{array}$$

The key property of the translation \dagger that is based on the chosen Prf is that \mathcal{L}_J -formulas that hold (do not hold) in \mathcal{M}_{fin} are translated to true (false) \mathcal{L}_{PA} -sentences, as stated formally in the following lemma:

Lemma 5.7. For each formula F of \mathcal{L}_J ,

- (i) $\mathcal{M}_{\text{fin}} \models F$ implies $\mathbb{N} \models F^\dagger$;
- (ii) $\mathcal{M}_{\text{fin}} \not\models F$ implies $\mathbb{N} \not\models F^\dagger$.

Proof. By simultaneous induction on the structure of F . We distinguish the following cases:

- (i) Let F be an atomic proposition. If $\mathcal{M}_{\text{fin}} \Vdash F$, then F^\dagger is $\ulcorner F^\neg \urcorner = \ulcorner F^\neg \urcorner$, which clearly is true. If $\mathcal{M}_{\text{fin}} \nVdash F$, then F^\dagger is $\neg(\ulcorner F^\neg \urcorner = \ulcorner F^\neg \urcorner)$, which clearly is false.
- (ii) If $F = \perp$, then trivially we have $\mathcal{M}_{\text{fin}} \nVdash \perp$ and $\mathbb{N} \nVdash \perp$.
- (iii) The case of $F = G \rightarrow H$ is immediate by the induction hypothesis.
- (iv) Let $F = s:G$. If $\mathcal{M}_{\text{fin}} \Vdash s:G$, then $\mathbb{N} \models \text{Jus}(\ulcorner s^\neg \urcorner, \ulcorner G^\neg \urcorner)$. Given that

$$(12) \quad \ulcorner G^\neg \urcorner = \text{undag}(\ulcorner G^\dagger \urcorner, \ulcorner \text{Prf}(x, y)^\neg \urcorner) \quad ,$$

we have by (11) that $\mathbb{N} \models \text{Prf}(\ulcorner s^\neg \urcorner, \ulcorner G^\dagger \urcorner)$, i.e., $\mathbb{N} \models (s:G)^\dagger$.

If $\mathcal{M}_{\text{fin}} \nVdash s:G$, then $\mathbb{N} \nVdash \text{Jus}(\ulcorner s^\neg \urcorner, \ulcorner G^\neg \urcorner)$. Moreover, $\mathbb{N} \nVdash \text{Proof}(\ulcorner s^\neg \urcorner, \ulcorner G^\dagger \urcorner)$ since by our assumptions $\mathbb{N} \nVdash \text{Proof}(\ulcorner s^\neg \urcorner, \ulcorner k \urcorner)$ for any k . Thus, by (11) and (12), we have $\mathbb{N} \nVdash \text{Prf}(\ulcorner s^\neg \urcorner, \ulcorner G^\dagger \urcorner)$, i.e., $\mathbb{N} \nVdash (s:G)^\dagger$. \square

Next, we show that $\text{Prf}(x, y)$ is a proof predicate.

Lemma 5.8. For every \mathcal{L}_{PA} -sentence ϕ ,

$$\text{PA} \vdash \phi \quad \Longleftrightarrow \quad \mathbb{N} \models \text{Prf}(\underline{n}, \ulcorner \phi^\neg \urcorner) \text{ for some natural number } n.$$

Proof. From left to right. Suppose $\text{PA} \vdash \phi$. Then there is a natural number n such that $\mathbb{N} \models \text{Proof}(\underline{n}, \ulcorner \phi^\neg \urcorner)$. By (11), we conclude $\mathbb{N} \models \text{Prf}(\underline{n}, \ulcorner \phi^\neg \urcorner)$.

From right to left. Suppose that $\mathbb{N} \models \text{Prf}(\underline{n}, \ulcorner \phi^\neg \urcorner)$. Then, by (11), either $\mathbb{N} \models \text{Proof}(\underline{n}, \ulcorner \phi^\neg \urcorner)$, in which case $\text{PA} \vdash \phi$ follows immediately, or $n = \ulcorner s^\neg \urcorner$ for some evidence term s and some \mathcal{L}_J -formula F such that $\ulcorner F^\neg \urcorner = \text{undag}(\ulcorner \phi^\neg \urcorner, \ulcorner \text{Prf}(x, y)^\neg \urcorner)$ and $\mathcal{M}_{\text{fin}} \Vdash s:F$. Therefore, $\phi \equiv F^\dagger$ and $\mathcal{M}_{\text{fin}} \Vdash F$. By the previous lemma, $\mathbb{N} \models F^\dagger$. Since F^\dagger is a provably Δ_1 sentence, we find $\text{PA} \vdash F^\dagger$, i.e., $\text{PA} \vdash \phi$. \square

Now we obtain a weak arithmetical CS-interpretation as follows.

Lemma 5.9. Let $*$ be a mapping such that $s^* := \ulcorner s^\neg \urcorner$ for each evidence term s and $p^* := p^\dagger$ for each atomic proposition p . Then the pair $(*, \text{Prf})$ is a weak arithmetical CS-interpretation. Moreover,

$$(13) \quad F^* = F^\dagger$$

for any \mathcal{L}_J -formula F .

Proof. We start with showing (13) by induction on the structure of F . We distinguish the following cases.

- (i) If F is an atomic proposition, then $F^* = F^\dagger$ by definition.
- (ii) If $F = t:G$, then $t^* = \ulcorner t^\neg \urcorner$. By the induction hypothesis, $G^* = G^\dagger$. Thus,
$$(t:G)^* = \text{Prf}(t^*, \ulcorner G^* \urcorner) = \text{Prf}(\ulcorner t^\neg \urcorner, \ulcorner G^\dagger \urcorner) = (t:G)^\dagger .$$
- (iii) If $F = \perp$, then $\perp^* = \perp^\dagger$ by definition.
- (iv) If $F = G \rightarrow H$, then $G^* = G^\dagger$ and $H^* = H^\dagger$ by the induction hypothesis. Thus, $(G \rightarrow H)^* = G^* \rightarrow H^* = G^\dagger \rightarrow H^\dagger = (G \rightarrow H)^\dagger$.

This finishes the proof of (13).

We show that $(*, \text{Prf})$ is indeed a weak arithmetical CS-interpretation. The mapping $*$ maps atomic propositions of \mathcal{L}_J to sentences of \mathcal{L}_{PA} and evidence terms to numerals. Further, Prf is a proof predicate by the previous lemma. It remains to establish (4)–(6) from Definition 5.1. We only present a proof of (4). The other proofs are similar.

Assume that $\mathbb{N} \models \text{Prf}(s^*, \ulcorner \phi \rightarrow \psi^\neg \urcorner)$ and $\mathbb{N} \models \text{Prf}(t^*, \ulcorner \phi^\neg \urcorner)$, i.e., in other words, $\mathbb{N} \models \text{Prf}(\ulcorner s^\neg \urcorner, \ulcorner \phi \rightarrow \psi^\neg \urcorner)$ and $\mathbb{N} \models \text{Prf}(\ulcorner t^\neg \urcorner, \ulcorner \phi^\neg \urcorner)$. By our assumptions, $\mathbb{N} \nVdash \text{Proof}(\ulcorner r^\neg \urcorner, \underline{k})$

for any evidence term r and any natural number k . Therefore, by (11), we find $\phi \equiv F^\dagger$ and $\psi \equiv G^\dagger$ for some \mathcal{L}_J -formulas F and G such that

$$\mathcal{M}_{\text{fin}} \Vdash s:(F \rightarrow G) \quad \text{and} \quad \mathcal{M}_{\text{fin}} \Vdash t:F .$$

Hence, $\mathcal{M}_{\text{fin}} \Vdash (s \cdot t):G$. Therefore, by (11), we obtain $\mathbb{N} \models \text{Prf}(\ulcorner s \cdot t \urcorner, \ulcorner G^\dagger \urcorner)$, which is $\mathbb{N} \models \text{Prf}((s \cdot t)^*, \ulcorner \psi \urcorner)$.

It remains to show that the constant specification is respected. Let $(c, A) \in \text{CS}$. Then $\mathcal{M}_{\text{fin}} \Vdash c:A$. Thus, by Lemma 5.7, we have $\mathbb{N} \models (c:A)^\dagger$. Hence, by (13), we have $\mathbb{N} \models (c:A)^*$. \square

Now, Lemma 5.3 follows easily. First, observe that by Lemma 5.9 the pair $(*, \text{Prf})$ is a weak arithmetical CS-interpretation. Suppose $\mathcal{M}_{\text{fin}} \Vdash G$. By Lemma 5.7, we find $\mathbb{N} \models G^\dagger$, which is $\mathbb{N} \models G^*$ by (13). This completes the proof of weak arithmetical completeness.

6. A SEMANTICS OF PROOFS FOR INTUITIONISTIC LOGIC

Definition 6.1 (Forgetful projection). The mapping $\circ : \mathcal{L}_J \rightarrow \mathcal{L}_\square$ is defined as follows:

$$\begin{aligned} p^\circ &:= p \quad \text{for } p \in \text{Prop} , \\ \perp^\circ &:= \perp , \\ (A \rightarrow B)^\circ &:= A^\circ \rightarrow B^\circ , \\ (t:A)^\circ &:= \Box A^\circ . \end{aligned}$$

Lemma 6.2 ([Art01]). For any constant specification CS and any formula $F \in \mathcal{L}_J$,

$$\text{LP}_{\text{CS}} \vdash F \quad \text{implies} \quad \text{S4} \vdash F^\circ .$$

Definition 6.3 (Realization). A *realization* is a mapping $r : \mathcal{L}_\square \rightarrow \mathcal{L}_J$ such that $(r(A))^\circ = A$.

The realization theorem provides an embedding of S4 into LP_{CS} :

Theorem 6.4 (Realization, [Art01, Fit05, GK12]). Let CS be an axiomatically appropriate and schematic constant specification. There exists a realization r such that for each \mathcal{L}_\square -formula F

$$\text{S4} \vdash F \quad \text{implies} \quad \text{LP}_{\text{CS}} \vdash r(F) .$$

Theorem 6.5. Let CS be a primitive recursive, axiomatically appropriate, and schematic constant specification. There exists a realization r such that, for each \mathcal{L}_\square -formula F ,

$$\text{S4} \vdash F \quad \text{if and only if} \quad r(F) \text{ is weakly arithmetically CS-valid.}$$

Proof. First we show the direction from left to right. By the realization theorem, there exists a realization r such that for each \mathcal{L}_\square -formula F

$$\text{S4} \vdash F \quad \text{implies} \quad \text{LP}_{\text{CS}} \vdash r(F) .$$

Combining this with Theorem 5.2, we obtain, for each \mathcal{L}_\square -formula F ,

$$\text{S4} \vdash F \quad \text{implies} \quad r(F) \text{ is weakly arithmetically CS-valid.}$$

For the direction from right to left, let r be an arbitrary realization and suppose that $r(F)$ is weakly arithmetically CS-valid. By Theorem 5.4, we obtain $\text{LP}_{\text{CS}} \vdash r(F)$. Hence, by Lemma 6.2, we find $\text{S4} \vdash F$. \square

To obtain a provability semantics for intuitionistic logic IL , we combine the previous result with the Gödel translation from IL to S4 . By \mathcal{L}_{ip} we understand the propositional language of intuitionistic logic IL . Note that it must include all Boolean connectives \wedge , \vee , and \rightarrow because these are intuitionistically independent of each other. Let the translation $t(\cdot)$ from IL to S4 be such that, for each formula F of the language \mathcal{L}_{ip} of IL ,

$$\text{IL} \vdash F \quad \text{if and only if} \quad \text{S4} \vdash t(F) .$$

Corollary 6.6. Let CS be a primitive recursive, axiomatically appropriate, and schematic constant specification. There exists a realization r such that, for each \mathcal{L}_{ip} -formula F ,

$$\text{IL} \vdash F \quad \text{if and only if} \quad r(t(F)) \text{ is weakly arithmetically } \text{CS}\text{-valid}.$$

7. RELATED WORK

The construction of the proof predicate $\text{Prf}(x, y)$ presented here is essentially taken from Artemov's original proof of arithmetical completeness for LP_{CS} , see [Art95, Art01, AB05]. Goris [Gor08] used a similar construction to provide LP_{CS} with a semantics of proofs in Buss's system S_2^1 .

Our result relies on the fact that LP_{CS} is decidable not only for finite constant specifications but, more generally, for almost schematic ones. The first general decidability proof for LP_{CS} with non-finite CS was done by Mkrtychev [Mkr97]. The notion of an *almost schematic* constant specification goes back to Kuznets [Kuz08]. For recent presentations of various decidability results, see [BKS13, Stu12, Stu13].

Gödel [Göd95] suggested using a system with explicit proofs for the interpretation of S4 in a lecture already in 1938, but the transcript of the lecture only appeared in 1995. Even before the publication of Gödel's work, Artemov [Art95] came up with the Logic of Proofs LP_{CS} and established the realization theorem, as well as completeness with respect to arithmetical interpretations.

The first systems for logics of proofs featuring formulas of the form $t:F$ with the intended meaning t is a proof of F appear in the work of Artemov and Straßen [AS93a, AS93b], who investigate arithmetical interpretations for these logics. However, these ancestors of LP_{CS} had no operations on proof terms and were too weak to capture the \Box -modality of S4 in full.

8. CONCLUSION

What is new in our work is the observation that, if one considers weak arithmetical interpretations, the construction of the proof predicate for them can be based on the largest constant specification and, thereby, be suitable for all constant specifications. Our Corollary 6.6 provides a uniform arithmetical provability semantics for intuitionistic logic, a semantics that can be based on any of a wide class of constant specifications. This strengthens the previously known result by Artemov that each intuitionistically valid formula has a constant specification that provides a provability interpretation for this formula. In particular, for the first time, the Logic of Proofs LP itself, i.e., LP_{CS} with the total constant specification CS where each constant proves every axiom, provides a provability semantics for intuitionistic logic.

It might be useful to point out exactly how Artemov's semantics was weakened to obtain this result. The obvious change was that the *finiteness of proofs* property

had to be dropped. According to Artemov’s definition of the proof predicate, each ‘proof’ can only prove finitely many formulas, whereas in our case the interpretation of a constant generally needs to prove infinitely many axioms from a given axiom scheme(s). Secondly, while we define how to interpret arithmetically the \cdot , $+$, and $!$ operations on proof terms, we do not extend the corresponding arithmetical functions to the (Gödel numbers of) proofs obtained from the standard Gödel proof predicate **Proof**.

Acknowledgments. We are grateful to Sergei Artemov for his expert advice on the finer points of the original arithmetical semantics for LP. The first author would like to thank Anna Zamansky for her amazing hospitality at and for the impeccable organization of ISRALOG’14, resulting in particular, in the publication of this special issue. The authors would like to thank the audiences of ISRALOG’14, Logic Colloquium 2014, and of the Researchers’ Seminar of the Theory and Logic Group at TU Wien for engaging discussions. We also thank the anonymous reviewers for raising the relevant foundational questions.

REFERENCES

- [AB05] Sergei N. Artemov and Lev D. Beklemishev. Provability logic. In D. M. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic, 2nd Edition*, volume 13, pages 189–360. Springer, 2005.
- [Art95] Sergei N. Artemov. Operational modal logic. Technical Report MSI 95–29, Cornell University, December 1995.
- [Art01] Sergei N. Artemov. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic*, 7(1):1–36, March 2001.
- [Art08] Sergei [N.] Artemov. The logic of justification. *The Review of Symbolic Logic*, 1(4):477–513, December 2008.
- [Art12] Sergei N. Artemov. The ontology of justifications in the logical setting. *Studia Logica*, 100(1–2):17–30, April 2012.
- [AS93a] Sergei [N.] Artëmov and Tyko Straßen. The basic logic of proofs. In E. Börger, G. Jäger, H. Kleine Büning, S. Martini, and M. M. Richter, editors, *Computer Science Logic, 6th Workshop, CSL’92, San Miniato, Italy, September 28–October 2, 1992, Selected Papers*, volume 702 of *Lecture Notes in Computer Science*, pages 14–28. Springer, 1993.
- [AS93b] Sergei [N.] Artëmov and Tyko Straßen. The logic of the Gödel proof predicate. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, editors, *Computational Logic and Proof Theory, Third Kurt Gödel Colloquium, KGC’93, Brno, Czech Republic, August 24–27, 1993, Proceedings*, volume 713 of *Lecture Notes in Computer Science*, pages 71–82. Springer, 1993.
- [BKR⁺10] Samuel Bucheli, Roman Kuznets, Bryan Renne, Joshua Sack, and Thomas Studer. Justified belief change. In Xabier Arrazola and María Ponte, editors, *LogKCA-10, Proceedings of the Second ILLI International Workshop on Logic and Philosophy of Knowledge, Communication and Action*, pages 135–155. University of the Basque Country Press, 2010.
- [BKS11] Samuel Bucheli, Roman Kuznets, and Thomas Studer. Partial realization in dynamic justification logic. In Lev D. Beklemishev and Ruy de Queiroz, editors, *Logic, Language, Information and Computation, 18th International Workshop, WoLLIC 2011, Philadelphia, PA, USA, May 18–20, 2011, Proceedings*, volume 6642 of *Lecture Notes in Artificial Intelligence*, pages 35–51. Springer, 2011.
- [BKS13] Samuel Bucheli, Roman Kuznets, and Thomas Studer. Decidability for justification logics revisited. In Guram Bezhanishvili, Sebastian Löbner, Vincenzo Marra, and Frank Richter, editors, *Logic, Language, and Computation, 9th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2011, Kutaisi, Georgia, September 26–30, 2011, Revised Selected Papers*, volume 7758 of *Lecture Notes in Computer Science*, pages 166–181. Springer, 2013.

- [Boo93] George Boolos. *The Logic of Provability*. Cambridge University Press, 1993.
- [BRS14] Alexandru Baltag, Bryan Renne, and Sonja Smets. The logic of justified belief, explicit knowledge, and conclusive evidence. *Annals of Pure and Applied Logic*, 165(1):49–81, January 2014.
- [DK14] Walter Dean and Hidenori Kurokawa. The Paradox of the Knower revisited. *Annals of Pure and Applied Logic*, 165(1):199–224, January 2014.
- [Fit05] Melvin Fitting. The logic of proofs, semantically. *Annals of Pure and Applied Logic*, 132(1):1–25, February 2005.
- [Gir87] Jean-Yves Girard. *Proof Theory and Logical Complexity*, volume 1 of *Studies in Proof Theory*. Bibliopolis, 1987.
- [GK12] Remo Goetschi and Roman Kuznets. Realization for justification logics via nested sequents: Modularity through embedding. *Annals of Pure and Applied Logic*, 163(9):1271–1298, September 2012.
- [Göd33] Kurt Gödel. Eine Interpretation des intuitionistischen Aussagenkalküls. In *Ergebnisse eines Mathematischen Kolloquiums*, 4:39–40. 1933.
- [Göd95] Kurt Gödel. Vortrag bei Zilsel/Lecture at Zilsel’s (*1938a). In Solomon Feferman, John W. Dawson, Jr., Warren Goldfarb, Charles Parsons, and Robert M. Solovay, editors, *Unpublished essays and lectures*, volume III of *Kurt Gödel Collected Works*, pages 86–113. Oxford University Press, 1995.
- [Gor08] Evan Goris. Feasible operations on proofs: The Logic of Proofs for bounded arithmetic. *Theory of Computing Systems*, 43(2):185–203, August 2008.
- [Hey31] Arend Heyting. Die intuitionistische Grundlegung der Mathematik. *Erkenntnis*, 2(1):106–115, 1931.
- [Hey34] A[rend] Heyting. *Mathematische Grundlagenforschung. Intuitionismus. Beweistheorie*. Springer, 1934.
- [Kol32] A. Kolmogoroff. Zur Deutung der intuitionistischen Logik. *Mathematische Zeitschrift*, 35(1):58–65, 1932.
- [KS12] Roman Kuznets and Thomas Studer. Justifications, ontology, and conservativity. In Thomas Bolander, Torben Braüner, Silvio Ghilardi, and Lawrence Moss, editors, *Advances in Modal Logic, Volume 9*, pages 437–458. College Publications, 2012.
- [KS13] Roman Kuznets and Thomas Studer. Update as evidence: Belief expansion. In Sergei [N.] Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science, International Symposium, LFCS 2013, San Diego, CA, USA, January 6–8, 2013, Proceedings*, volume 7734 of *Lecture Notes in Computer Science*, pages 266–279. Springer, 2013.
- [Kuz08] Roman Kuznets. *Complexity Issues in Justification Logic*. PhD thesis, City University of New York, May 2008.
- [Mkr97] Alexey Mkrtychev. Models for the logic of proofs. In Sergei Adian and Anil Nerode, editors, *Logical Foundations of Computer Science, 4th International Symposium, LFCS’97, Yaroslavl, Russia, July 6–12, 1997, Proceedings*, volume 1234 of *Lecture Notes in Computer Science*, pages 266–275. Springer, 1997.
- [MT48] J.C.C. McKinsey and Alfred Tarski. Some theorems about the sentential calculi of Lewis and Heyting. *The Journal of Symbolic Logic*, 13(1):1–15, March 1948.
- [Ren11] Bryan Renne. Public communication in justification logic. *Journal of Logic and Computation*, 21(6):1005–1034, December 2011.
- [Stu12] Thomas Studer. Lectures on justification logic. Lecture notes, November 2012.
- [Stu13] Thomas Studer. Decidability for some justification logics with negative introspection. *The Journal of Symbolic Logic*, 78(2):388–402, June 2013.
- [TvD88a] A. S. Troelstra and D. van Dalen. *Constructivism in Mathematics: An Introduction, Volume I*, vol. 121 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1988.
- [TvD88b] A. S. Troelstra and D. van Dalen. *Constructivism in Mathematics: An Introduction, Volume II*, vol. 123 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1988.
- [Ver14] Rineke (L.C.) Verbrugge. Provability logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2014 edition, 2014.

INSTITUT FÜR COMPUTERSPRACHEN, TECHNISCHE UNIVERSITÄT WIEN
E-mail address: `roman@logic.at`

INSTITUT FÜR INFORMATIK, UNIVERSITÄT BERN
E-mail address: `tstuder@iam.unibe.ch`