2011

TR-2011005: First-Order Logic of Proofs

Sergei N. Artemov

Tatiana Yavorskaya (Sidon)

First-Order Logic of Proofs

Sergei N. Artemov*
CUNY Graduate Center
365 Fifth Avenue, rm. 4319
New York City, NY 10016, USA
sartemov@gc.cuny.edu

Tatiana Yavorskaya (Sidon)†
Department of Mathematics
Moscow University
Moscow, 119899, Russia
tanya@lpcs.math.msu.su

May 27, 2011

Abstract

The propositional logic of proofs LP revealed an explicit provability reading of modal logic S4 which provided an indented provability semantics for the propositional intuitionistic logic IPC and led to a new area, Justification Logic. In this paper, we find the *first-order logic of proofs* FOLP capable of realizing first-order modal logic S4 and, therefore, the first-order intuitionistic logic HPC. FOLP enjoys a natural provability interpretation; this provides a semantics of explicit proofs for first-order S4 and HPC compliant with Brouwer-Heyting-Kolmogorov requirements. FOLP opens the door to a general theory of first-order justification.

1 Introduction

The propositional logic of proofs LP revealed an explicit proof reading of modal logic S4 ([1, 2]). This completed Gödel's effort ([9, 10]) to provide the intended provability semantics for the propositional intuitionistic logic IPC and eventually led to a new field, Justification Logic (cf. [3, 8]).

The intended semantics of intuitionistic logic is the semantics of proofs, also known as *Brouwer-Heyting-Kolmogorov (BHK) semantics*. It starts from Brouwer's thesis that intuitionistic truth is provability, i.e., a proposition is true if it has a proof (cf. [19], p. 4). Proofs should reflect the logical structure of a sentence. In particular,

- a proof of $A \to B$ is a construction which, given a proof of A, returns a proof of B;
- a proof of $A \wedge B$ consists of a proof of A and a proof of B;

*Supported by the National Science Foundation.
†Partially supported by the Russian Foundation for Basic Research.

- a proof of $A \vee B$ is given by presenting either a proof of A or a proof of B;
- a proof of $\forall x A(x)$ is a function converting c into a proof of $A(c)$;
- a proof of $\exists x A(x)$ is a pair (c, d) where d is a proof of $A(c)$.

Kolmogorov [11], and later Gödel [9, 10], viewed BHK-proofs as proofs[1] in classical mathematics. Moreover, Gödel offered a method of connecting classical provability with intuitionistic logic ([9]) in a way that respects the informal provability reading of IPC:

$$\mathsf{IPC} \vdash F \quad \textit{iff} \quad \mathsf{S4} \vdash tr(F),^{2}$$

where $tr(F)$ is obtained from F by prefixing each subformula of F with $\Box$. When parsing Gödel's translation $tr(F)$ of some formula F, we encounter a provability modality before each subformula, which forces us to read said subformula as provable rather than true. Therefore, Gödel's translation reflects the fundamental intuitionistic paradigm that intuitionistic truth is provability. Moreover, the classical version of BHK is that which provides a non-circular semantics for intuitionistic logic. A similar position was taken by P.S. Novikov in [15].

At that stage, the problem of finding a BHK-type semantics of proof for IPC was reduced to developing such a semantics for S4 which proved to be quite elusive; cf. [2, 20] for more on the history of this problem. In this context, Gödel in [10] discusses the possibility of finding a classical logic of proofs which could provide a semantics of proofs for S4, hence for IPC.

The next step was taken in [1, 2], where the propositional logic of proofs LP with new atoms $t{:}F$ for

$$t \textit{ is a proof of } F$$

was introduced. The Realization Theorem from [1, 2] demonstrated that each S4 theorem conceals an explicit statement about proofs, e.g.,

$$\Box F \to \Box G$$

reads as

$$u{:}F \to t(u){:}G$$

i.e., *if u is a proof of F, then $t(u)$ is a proof of G*, for an appropriate proof term $t(u)$. The Realization Theorem allows for the extension of this kind of explicit reading of modalities to all theorems of S4, so S4 has a semantics via LP as anticipated by Gödel in [10]. Since proof terms in LP can be naturally interpreted as mathematical proofs, e.g., in Peano Arithmetic PA, S4 and IPC received an exact provability semantics consistent with BHK-requirements.

There is a tradition to regard computational semantics for intuitionistic logic, e.g., provided by Kleene realizability, or the computational interpretation of Martin-Löf Type Theory as a variant of BHK-semantics (cf. [19]) in which 'proofs' are read as 'computational programs.' Indeed, BHK conditions are mainly satisfied under computational interpretation

[1] Kolmogorov in [11] wrote about "problem solutions."

[2] This statement follows from [9, 14].

with a principal exception: the original BHK cause for disjunction was intended for provability semantics and is not valid for the aforementioned computational reading. To fit the computational model, this item has been modified by adding a new requirement

a proof of a disjunction should also specify which disjunct it proves.

This extra condition is redundant for the provability semantics since the predicate *p is a proof of F* is decidable and given a proof p, we always 'know' which of the disjuncts it proves.

The *provability BHK semantics*, the logic of proofs, turned out to be quite different from the *computational BHK semantics*. Both have applications far beyond the original foundational area. The computational BHK semantics has strong links with Computer Science, e.g., typed programming languages, whereas the provability BHK semantics is contributing to epistemology, where it has led to a mathematical theory of justification (cf. [3, 8]).

1.1 Propositional logic of proofs

Let us discuss the propositional case in more detail. Proofs are represented in LP by *proof terms* constructed from *proof variables* and *proof constants* by means of functional symbols for elementary computable operations on proofs, binary $\cdot$, $+$, and unary $!$. The formulas of LP are built by Boolean connectives from propositional atoms and those of the form $t{:}F$ where t is a *proof term* and F is a formula. The operations of LP are specified by the following schemas (here A, B are arbitrary formulas and t, s are arbitrary terms):

$t{:}(A \to B) \to (s{:}A \to (t \cdot s){:}B)$ *application*

$t{:}A \to (t + s){:}A, \quad s{:}A \to (t + s){:}A$ *sum*

$t{:}A \to\, !t{:}t{:}A$ *proof checker.*

LP is axiomatized over the classical propositional calculus by the above schemas, the principle

$t{:}A \to A$ *reflexivity*

and the *axiom necessitation rule*, which allows for the specification of proof constants as proofs of the concrete axioms

$$\vdash c{:}A, \quad \text{where } c \text{ is an axiom constant, } A \text{ is an axiom of LP.}$$

The intended semantics for LP is provided by proof predicates in Peano Arithmetic PA. The proof terms of LP-language are interpreted by codes of arithmetical derivations. Operations $\cdot$, $+$, and unary $!$ become total recursive functions on such codes. Formulas of LP are interpreted by closed arithmetical formulas; interpretations commute with Boolean connectives and $t{:}F$ is interpreted by an arithmetical proof predicate which numerates theorems of PA. It is established in [1, 2] that LP is complete with respect to the class of all arithmetical proof predicates[3].

The following Realization Theorem ([1, 2]) shows that LP is an exact counterpart of Gödel's provability logic S4.

[3]The proof predicates considered here are multi-conclusion, e.g., a proof may be a proof of more than one proposition. Single-conclusion proof predicates have their own logic of proofs axiomatized in [12, 13].

A modal formula F is provable in S4 *iff there exists an assignment (called a realization) of proof terms to all occurrences of $\Box$ in F such that the resulting formula is provable in* LP*.*

The proof of the Realization Theorem treats $\Box$ in the style of Skolem as the existential quantifier on proofs. Negative occurrences of $\Box$'s are assumed to hide universal quantifiers and hence are realized by proof variables, and positive occurrences of $\Box$'s are realized as existential quantifiers, i.e., by proof terms depending on these variables.

The Realization Theorem provides S4, and therefore intuitionistic logic, with an exact BHK-style provability semantics, thus completing Gödel's project of formalizing the intended provability semantics for IPC.

1.2 First-order case

Similar questions for first-order S4, FOS4, and the first-order intuitionistic logic HPC remained open and are answered in this paper. Studies of the first-order logic of proofs were initiated in [16] and [4] where it was established that some natural axiomatizability questions for the first-order logic of proofs have negative answers. Paper [21] considered so-called *binding interpretations* where formulas $p{:}F$ were interpreted by $Prf(p^*, \ulcorner F^* \urcorner)$ and thus had no parameters. A complete axiomatization of the corresponding logic of proofs was found, which, however, does not help to realize first-order modal logic.

An extension of LP by quantifiers on proofs, QLP, was considered in [7].

In this paper, we suggest a first-order logic of proofs FOLP and its exact Gödel-style interpretation in mathematical proofs (e.g., in Peano Arithmetic). We show that FOLP is capable of realizing FOS4; this provides FOS4 with an exact provability semantics. Since HPC can be faithfully embedded in FOS4 via Gödel's transation, this also provides HPC with a BHK-style semantics of proofs.

2 First-order logic of proofs

Let $A(x)$ be a formula with a parameter x. Then, in FOS4, $\Box A(x)$ also has x as a parameter. In the provability reading of $\Box$, this reflects the reading of $\Box A(x)$ as

given a natural parameter $x = n$, formula $A(n)$ is provable.

On the other hand, FOLP cannot express the notion

formula $A(x)$ with a free variable x is provable.

In FOLP, there are tools for representing both of the aforementioned readings of individual variables: as 'global parameters' or 'local' variables not accessible from outside the proof/provability operator. Suppose p is being interpreted as a specific proof (in PA) and A as a specific formula (of PA) with a free variable x. We can read $p{:}A$ in two ways:

- p proves A for a given value of the parameter x and thus the formula $p{:}A$ and its truth value depend on x. For example, p is $\{0 = 0\}$, and A is $x = x$; then $p{:}A$ holds only when x is substituted by 0.

- p is a proof of a formula with a free variable x and so $p{:}A$ does not depend on x. For example, p is $\{x = x\}$ and A is, as before, $x = x$.

In the language of FOLP, the proof predicate is represented by formulas of the form

$$t{:}_X A$$

where X is a finite set of individual variables that are considered global parameters and free variables of this formula. All occurrences of variables from X that are free in A are also free in $t{:}_X A$. All other free variables of A are considered local and hence bound in $t{:}_X A$. For example, if $A(x, y)$ is an atomic formula, then in $p{:}_{\{x\}}A(x, y)$, variable x is free and variable y is bound. Likewise, in $p{:}_{\{x,y\}}A(x, y)$ both variables are free and in $p{:}_{\emptyset}A(x, y)$, neither x nor y is free.

Proofs are represented by proof terms which do not contain individual variables. On the level of FOLP, this makes presentation of proofs light and manageable without sacrificing generality.

Definition 1 Let $\mathcal{L}$ denote the first-order language that contains a countable set of predicate symbols of any arity, without functional symbols and equality. *The language* FOLP is the extension of $\mathcal{L}$ by special means to represent proofs and proof assertions, namely, the language FOLP contains individual variables $x_0, x_1, x_2, \ldots$, Boolean connectives, quantifiers over individual variables, predicate symbols Q_i^n of any arity n $(i, n = 0, 1, 2, \ldots)$ and

- proof variables p_k, $k = 0, 1, 2 \ldots$, and proof constants $c_0, c_1, c_2, \ldots$;

- functional symbols for operations on proofs:

  - those of LP: binary $+$, $\cdot$, and unary $!$,
  - unary gen_x for each individual variable x;

- an operational symbol $(\cdot){:}_X(\cdot)$ for each finite set X of individual variables.

Definition 2 *Proof terms* are constructed as follows:

- each proof constant and proof variable is a proof term;

- if t, s are proof terms, then $t \cdot s$, $!t$, $t + s$, and $\mathsf{gen}_x(t)$ are proof terms.

Notation. 1 By X, Y, etc., we denote finite sets of individual variables. If y is an individual variable, then we will write Xy for $X \cup \{y\}$. An additional convention: notation Xy means, in part, that $y \notin X$. Note also that in gen_x, variable x is merely a syntactic label of this operation and is not considered an occurrence of a variable. So terms in FOLP do not contain individual variables.

Definition 3 *Formulas* are defined in the standard way with an additional clause for the proof operator. Namely,

- If Q_i^n is a predicate symbol of arity n and $x_1, \ldots, x_n$ are individual variables, then $Q_i^n(x_1, \ldots, x_n)$ is an atomic formula; all occurrences of individual variables are free.
- If A, B are formulas, then $\neg A$, $A \alpha B$ (α being a Boolean connective) are formulas; Boolean connectives preserve free and bound occurrences of variables.
- If A is a formula and x is an individual variable, then $\forall x A$ is a formula; $\forall x$ binds all occurrences of x and preserves free and bound occurrences of all other variables.
- If t is a proof term and A is a formula, then $t:_X A$ is a formula. In this formula, all variables from X, and only from X, are free. All free occurrences of variables from X in A are also free.

The set of free variables of a formula A is denoted by $FVar(A)$. We use the abbreviation $t{:}A$ for $t:_\emptyset A$.

Definition 4 There are two types of substitutions in the language of FOLP: individual variables and proof variables.

- Substitution of individual variables. If x, y are variables and A is a formula, then by $A(y/x)$ we denote the result of substitution of y for all free occurrences of x in A. We always assume that substitution is correct, namely, there is no free occurrence of x in A within the scope of any binding of y.
- Substitution of proof terms. Let p be a proof variable and t be a proof term. By $A(t/p)$ we denote the result of substitution of t for all occurrences of p in A.

Definition 5 The first-order logic of proofs FOLP is axiomatized by the following schemas. Here A, B are formulas, s, t are terms, X is a set of individual variables, and y is an individual variable.

A1 classical axioms of first-order logic
A2 $t:_{Xy} A \to t:_X A, \quad y \notin FVar(A)$
A3 $t:_X A \to t:_{Xy} A$
B1 $t:_X A \to A$
B2 $s:_X (A \to B) \wedge t:_X A \to (s \cdot t):_X B$
B3 $t:_X A \to (t+s):_X A, \quad s:_X A \to (t+s):_X A$
B4 $t:_X A \to !t:_X t:_X A$
B5 $t:_X A \to \mathsf{gen}_x(t):_X \forall x A,\ x \notin X$

FOLP has the following inference rules:

R1 $\vdash A, A \to B \ \Rightarrow\ \vdash B$ *Modus Ponens*
R2 $\vdash A \ \Rightarrow\ \vdash \forall x A$ *generalization*
R3 $\vdash c{:}A$, where A is an axiom, c is a proof constant *axiom necessitation.*

We define derivations from the hypothesis in FOLP in the standard way. Let us recall that in a derivation from the set of hypotheses Γ, the generalization rule may not be applied to variables that are free in Γ.

Example 1 Let us derive (in FOLP) an explicit counterpart of the converse Barcan Formula

$$\Box\forall xA \to \forall x\Box A.$$

1. $\forall xA \to A$ - logical axiom;
2. $c{:}(\forall xA \to A)$ - axiom necessitation;
3. $c{:}_{\{x\}}(\forall xA \to A)$ - from 2, by axiom A3;
4. $c{:}_{\{x\}}(\forall xA \to A) \to (u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A)$ - axiom B2;
5. $u{:}_{\{x\}}\forall xA \to (c\cdot u){:}_{\{x\}}A$ - from 3, 4, by Modus Ponens;
6. $u{:}\forall xA \to u{:}_{\{x\}}\forall xA$ - by axiom A3;
7. $u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A$ - from 5, 6;
8. $\forall x[u{:}\forall xA \to (c\cdot u){:}_{\{x\}}A]$ - from 7, by generalization;
9. $u{:}\forall xA \to \forall x(c\cdot u){:}_{\{x\}}A$ - from 8, since the antecedent of 8 does not contain x free.

Lemma 1 [Substitution] *If* FOLP $\vdash F$, *p is a proof variable, and t is a proof term, then* FOLP $\vdash F(t/p)$.

Lemma 2 [Deduction] *If* $\Gamma, A \vdash F$ *in* FOLP, *then* $\Gamma \vdash A \to F$.

Theorem 1, which follows, establishes the internalization property for FOLP (though not in its most general form, but nonetheless sufficient for purposes of this paper).

Theorem 1 [Internalization] *Let $p_0, \ldots, p_k$ be proof variables, $X_0, \ldots, X_k$ be sets of individual variables, and $X = X_0 \cup X_1 \cup \ldots \cup X_k$. Suppose that in* FOLP

$$p_0{:}_{X_0}A_0, \ldots, p_k{:}_{X_k}A_k \vdash F.$$

Then there exists a proof term $t(p_0, p_1, \ldots, p_k)$ such that

$$p_0{:}_{X_0}A_0, \ldots, p_k{:}_{X_k}A_k \vdash t{:}_X F.$$

Proof. Induction on derivation of F from $p_0{:}_{X_0}A_0, \ldots, p_k{:}_{X_k}A_k$.

Case 1. F is an axiom of FOLP. By the axiom necessitation rule, $c{:}F$ is derivable for a proof constant c. By A3, $c{:}_X F$ as well. Take $t = c$.

Case 2. F is $p_i{:}_{X_i}A_i$ for some i. By B4, $!p_i{:}_{X_i}p_i{:}_{X_i}A_i$. By A3, $!p_i{:}_X p_i{:}_{X_i}A_i$. Take $t = !p_i$.

Case 3. F follows by *Modus Ponens*. Then, by the Induction Hypothesis, from the given set of hypotheses it is derivable that $s_1{:}_X(G \to F)$ and $s_2{:}_X G$ for some G, s_1, and s_2. By B2, $(s_1 \cdot s_2){:}_X F$. Take $t = s_1 \cdot s_2$.

Case 4. F follows by *generalization*, i.e., $F = \forall xG$ for some x not occurring free in the set of hypotheses. In particular, $x \notin X$. By IH, $s{:}_X G$ for some s. By B5, $s{:}_X G \to \mathsf{gen}_x(s){:}_X \forall xG$, hence $\mathsf{gen}_x(s){:}_X \forall xG$. Take $t = \mathsf{gen}_x(s)$.

Case 5. F follows by *axiom necessitation*, i.e., $F = c{:}A$ for some axiom A and constant c. By B4, $!c{:}c{:}A$. By A3, $!c{:}_X c{:}A$. Take $t = !c$. □

In particular, given $\vdash F$, there is a proof term t containing no proof or individual variables such that $\vdash t{:}F$. Such t can be chosen +-free.

3 Realization of FOS4

Definition 6 Let A be a first-order modal formula. By *realization* of a formula A we mean a formula A^r of the language of FOLP that is obtained from A by replacing all occurrences of subformulas of A of the form $\Box B$ by $t{:}_X B$ for some proof terms t and such that $X = FVar(B)$. To avoid unnecessary formalism, we suggest considering a realization the result of an iterated procedure that always replaces an innermost $\Box B$ by $t:_X B$. A realization is *normal* if all negative occurrences of $\Box$ are assigned proof variables.

Remark 1 If A^r is a realization of A, then for every subformula B of A,

$$FVar(B^r) = FVar(B).$$

As in the propositional case [1, 2], we define a forgetful projection $(\cdot)^0$ of FOLP to modal logic. The straightforward definition of $t{:}_X F$ as $\Box F$ does not work in first-order logic. This is due to the fact that $\Box$ does not bind individual variables, while the proof operator can bind them. For example, the 'naive' projection of an instance of axiom B5

$$t{:}A(x) \to \mathsf{gen}_x(t){:}\forall x A(x) \tag{1}$$

is

$$\Box A(x) \to \Box \forall x A(x). \tag{2}$$

This yields Barcan formula $\forall x \Box A(x) \to \Box \forall x A(x)$ which is not provable in FOS4 and not valid under the intuitive provability interpretation of $\Box$ (cf. Section 7). We argue that (1) states something different than (2). Since formula $t{:}A(x)$ does not contain global parameters (open variables), it asserts that t is a proof of $A(x)$ with a free variable x. In (2), $\Box A(x)$ states the weaker condition that $A(x)$ is provable for each value of x, which makes all or (2) stronger than was suggested by (1).

A more adequate definition of the forgetful projection should not change the status of individual variables. A possible way to meet this condition is to require that forgetful projection binds local variables by the universal quantifier. Provability of a formula F with a free variable y is equivalent to the provability of $\forall y F$. So a forgetful projection of $t{:}_{\{x\}} F(x, y)$ that respects binding could be $\Box \forall y F(x, y)$. This example leads us to the following definition.

Definition 7 We define the forgetful projection $(\cdot)^0$ of FOLP to first-order modal language by induction on an FOLP-formula. For atomic formulas, we stipulate $F^0 = F$, forgetful projection commutes with Boolean connectives and quantifiers, and for proof assertions,

$$(t{:}_X F)^0 = \Box \forall y_0 \ldots \forall y_k F^0, \text{ where } \{y_0, \ldots, y_k\} = FVar(F) \setminus X.$$

Lemma 3 *If* FOLP $\vdash F$*, then* F^0 *is derivable in* FOS4*.*

Proof. By straightforward induction on derivations in FOLP. $\Box$

Theorem 2 [Realization Theorem] *If* $\mathsf{FOS4} \vdash A$*, then there is a normal realization* A^r *such that* $\mathsf{FOLP} \vdash A^r$*.*

Proof. The proof is similar to that in [2] with additional consideration given to individual variables. We consider the Gentzen-style calculus for FOS4 and prove that for every sequent $\Gamma \Rightarrow \Delta$ that is provable in FOS4, there exists a realization r of all formulas from Γ and Δ such that $\mathsf{FOLP} \vdash (\wedge\Gamma^r \rightarrow \vee\Delta^r)$. For this purpose, we take a cut-free derivation of $\Gamma \Rightarrow \Delta$ and construct realization for the entire derivation.

According to [18], Section 9.1.3, in addition to structural rules, the sequential calculus for FOS4 has the following axioms:

$$\bot \Rightarrow \qquad \text{and} \qquad P(X) \Rightarrow P(X)$$

and logical rules:

$$\frac{\Gamma, A \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \rightarrow B}\ (R\rightarrow), \qquad \frac{\Gamma \Rightarrow \Delta, A, \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \rightarrow B \Rightarrow \Delta}\ (L\rightarrow),$$

$$\frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, \forall x A}\ (R\forall), \qquad \frac{\Gamma, A(y/x) \Rightarrow \Delta}{\Gamma, \forall x A \Rightarrow \Delta}\ (L\forall),$$

$$\frac{\Box\Gamma \Rightarrow A}{\Box\Gamma \Rightarrow \Box A}\ (R\Box), \qquad \frac{\Gamma, A \Rightarrow \Delta}{\Gamma, \Box A \Rightarrow \Delta}\ (L\Box).$$

In $R\forall$, we suppose that $x \notin FVar(\Gamma, \Delta)$.

The following connection between FOS4 and its Gentzen-style version $\mathcal{G}$FOS4 takes place:

$$\mathcal{G}\mathsf{FOS4} \vdash \Gamma \Rightarrow \Delta \quad \text{iff} \quad \mathsf{FOS4} \vdash \bigwedge \Gamma \rightarrow \bigvee \Delta.$$

Cut-elimination holds in $\mathcal{G}$FOS4 ([18]): if $\mathcal{G}\mathsf{FOS4} \vdash \Gamma \Rightarrow \Delta$, then $\Gamma \Rightarrow \Delta$ can be derived in $\mathcal{G}$FOS4 without using the cut rule.

Lemma 4 *If* $\mathcal{G}\mathsf{FOS4} \vdash \Gamma \Rightarrow \Delta$*, then there exists a normal realization* r *such that* $\mathsf{FOLP} \vdash (\bigwedge \Gamma^r \rightarrow \bigvee \Delta^r)$.

Proof. Suppose that $\mathcal{D}$ is a cut-free derivation in FOS4. We will construct a realization for each sequent $\Gamma \Rightarrow \Delta$ in $\mathcal{D}$ in such a way that the formula $\bigwedge \Gamma^r \rightarrow \bigvee \Delta^r$ is provable in FOLP.

Following [2], we split all occurrences of $\Box$'s in derivation $\mathcal{D}$ into families of related ones. Namely, two occurrences of $\Box$ are related if they occur in related subformulas of premises and conclusions of rules; we extend this relationship by reflexivity and transitivity. All rules of $\mathcal{G}$FOS4 respect polarities, hence all $\Box$'s in every family have the same polarity, so we can speak of *positive and negative families of* $\Box$*'s*. If f is a positive family, then all $\Box$'s from f are introduced either by weakening on the right or by the rule $(R\Box)$. If at least one $\Box$ in f is introduced by $(R\Box)$, then we call f an *essential* family, otherwise f is called *inessential family.*

Step 1. Initialization. To every negative or inessential positive family f we assign a fresh proof variable p. Replace all $\Box A$, where $\Box$ is from f, by $p{:}_X A$ with $X = FVar(A)$.

Suppose that f is an essential positive family. We enumerate the rules $(R\Box)$ which introduce $\Box$'s from the family f. Let $n(f)$ be the total number of such rules for the family f. For the $(R\Box)$–rule number k in a family f where $k = 1, \ldots, n(f)$, we take a fresh proof variable u_k called *a provisional variable*. Finally, replace all $\Box A$ from the family f by

$$[u_1 + \ldots + u_{n(f)}]{:}_X A$$

with $X = FVar(A)$.

After initialization is completed, all nodes in the resulting tree $\mathcal{D}'$ are assigned formulas of the logic FOLP.

Step 2. Realization. We now travel along derivation $\mathcal{D}'$ from leaves to root and replace all provisional variables by FOLP–terms. We retain the notation u_j for both provisional variables and terms substituted for them. The resulting tree is denoted by $\mathcal{D}^r$. By induction on the depth of a node in $\mathcal{D}'$, we prove that after the process passes the node $\Gamma \Rightarrow \Delta$ in $\mathcal{D}'$ and replaces it by $\Gamma^r \Rightarrow \Delta^r$,

1. sequent $\Gamma^r \Rightarrow \Delta^r$ is derivable in FOLP[4];
2. for every subformula B occurring in Γ, Δ, we have $FVar(B^r) = FVar(B)$.

We do not change the realization when the process passes sequents that are not conclusions of a $(R\Box)$–rule. All rules except $(R\Box)$ are admissible in FOLP, therefore the conclusions of those rules are derivable in FOLP as long as the premises are derivable.

The only case in which we alter realization is in rule $(R\Box)$. Suppose that $\Gamma \Rightarrow \Delta$ is obtained by rule $(R\Box)$:

$$\frac{\Box A_1, \ldots, \Box A_k \Rightarrow A}{\Box A_1, \ldots, \Box A_k \Rightarrow \Box A}.$$

The symbol $\Box$ introduced by this rule belongs to an essential positive family f. Let this rule have the number i among rules $(R\Box)$ which introduce $\Box$'s from this family f, and $n = n(f)$.

Currently in $\mathcal{D}^r$, the node corresponding to the premise of this rule is assigned a sequent $q_1{:}_{X_1} B_1, \ldots, q_k{:}_{X_k} B_k \Rightarrow B$ which, by the Induction Hypothesis, is provable in FOLP. The node corresponding to the conclusion is assigned a sequent

$$q_1{:}_{X_1} B_1, \ldots, q_k{:}_{X_k} B_k \Rightarrow [u_1 + \ldots + u_i + \ldots + u_n]{:}_X B$$

where all q_j are proof variables, all u_j are either provisional variables or terms, u_i is a provisional variable, and $X = FVar(B)$.

By the Internalization Lemma, it follows that there exists a term t such that FOLP derives

$$q_1{:}_{X_1} B_1, \ldots, q_k{:}_{X_k} B_k \Rightarrow t{:}_Y B$$

[4]Which means that $\mathsf{FOLP} \vdash \wedge\Gamma^r \rightarrow \vee\Delta^r$, or equivalently, $\Gamma^r \vdash \vee\Delta^r$.

where $Y = X_1 \cup X_2 \cup \ldots \cup X_n$. Using axiom A2, we remove from Y all variables that are not in $FVar(B)$ and obtain $Y' = Y \cap FVar(B)$. Then, by A3, add to Y' all free variables of B that were not yet there and obtain X. The resulting sequent

$$q_1{:}_{X_1}B_1, \ldots, q_k{:}_{X_k}B_k \Rightarrow t{:}_X B$$

is provable in FOLP. Therefore, by B3,

$$q_1{:}_{X_1}B_1, \ldots, q_k{:}_{X_k}B_k \Rightarrow [u_1 + \ldots + u_{i-1} + t + u_{i+1} + \ldots + u_n]{:}_X B$$

is also provable in FOLP. Replace provisional variable u_i by t everywhere in $\mathcal{D}^r$. By the Substitution Lemma, this respects provability in FOLP. □

Proof of the Theorem is now immediate. Since $\mathsf{FOS4} \vdash A$, there is a cut-free proof of sequent $\Rightarrow A$ in $\mathcal{G}$FOS4. By Lemma 4, there exists its normal realization $\Rightarrow A^r$ provable in FOLP, i.e., $\mathsf{FOLP} \vdash A^r$. □

Corollary 1 FOS4 *is the forgetful projection of* FOLP.

Corollary 2 *F is derivable in* HPC *if and only if its Gödel translation is realizable in* FOLP.

4 Parametric arithmetical semantics for FOLP

In this section, we describe the arithmetical interpretation of FOLP via proofs in Peano Arithmetic PA. The definition of PA and related topics are given in detail, e.g., in [5, 17]. As usual, for a natural number n by $\overline{n}$ we denote the numeral $\underbrace{SS\ldots S}_{n}0$ that represents n in PA. We will simply write n for this numeral in case it is not misleading.

According to the classical Gödel approach ([9, 10]), the main idea is to interpret FOLP formulas as arithmetical formulas, proof terms as codes of proofs in PA, and proof assertions $t{:}_X F$ as arithmetical proof predicates *t is a proof of F in* PA. The main technical problem here is how to properly handle individual variables.

4.1 Open variables in first-order derivations

The role of X in $t{:}_X F$ is to provide a substitutional access to derivation t and formula F for all variables from X. For this we have to define 'free variables of a derivation' in such a way that

> *if a derivation $\mathcal{D}(x)$ with a 'free variable x' is a proof of a formula $F(x)$, then for each n, $\mathcal{D}(n)$ is a derivation of $F(n)$.* (3)

Surprisingly, making sense of the above-stated notion of 'free variables of a derivation' is not immediate. Here is a relevant example. Let $F(x)$ be a logical axiom with a free variable x. Then the following is a Hilbert-style derivation $\mathcal{D}$:

$F(x)$ - axiom;
$\forall x F(x)$ - generalization;
$F(x) \to (\forall x F(x) \to F(x) \wedge \forall x F(x))$ - conjunction axiom;
$\forall x F(x) \to F(x) \wedge \forall x F(x)$ - Modus Ponens;
$F(x) \wedge \forall x F(x)$ - Modus Ponens.

Question: is the very first occurrence of x free in this derivation?

Answer 1: x is free, since it is free in $F(x)$.

Answer 2: x is not free, since substitution $(0/x)$ ruins the derivation. The generalization step is no longer legitimate: one cannot conclude $\forall x F(x)$ from $F(0)$.

The reason for this confusion lies in the fact that Hilbert derivations are not trees and reuse the same formulas. The true structure of this derivation is revealed by its tree-style presentation

$$\dfrac{\dfrac{\dfrac{F(x) \to (\forall x F(x) \to F(x) \wedge \forall x F(x)) \qquad F(x)}{\forall x F(x) \to F(x) \wedge \forall x F(x)} \qquad \dfrac{F(x)}{\forall x F(x)}}{F(x) \wedge \forall x F(x)}\,. \tag{4}$$

As we can see, axiom $F(x)$ appears twice in this derivation in quite different substitutional contexts. In the left branch, variable x from $F(x)$ remains free until the root of the derivation. In the right branch, $F(x)$ was subjected to generalization and binding of variable x. Which occurrences of x in this derivation are open to substitution in the way described in (3)? The answer is given by the boldface occurrences $\boldsymbol{x}$:

$$\dfrac{\dfrac{\dfrac{F(\boldsymbol{x}) \to (\forall x F(x) \to F(\boldsymbol{x}) \wedge \forall x F(x)) \qquad F(\boldsymbol{x})}{\forall x F(x) \to F(\boldsymbol{x}) \wedge \forall x F(x)} \qquad \dfrac{F(x)}{\forall x F(x)}}{F(\boldsymbol{x}) \wedge \forall x F(x)}\,. \tag{5}$$

Indeed, if we denote this derivation as $\mathcal{D}(\boldsymbol{x})$, and the root formula as $A(\boldsymbol{x})$, then

1. $\mathcal{D}(\boldsymbol{x})$ is a proof of $A(\boldsymbol{x})$;
2. $\mathcal{D}(n/\boldsymbol{x})$ is a proof of $A(n/\boldsymbol{x})$ for each n.

This observation leads us to the following convention about open occurrences of variables in derivations. By a PA-proof d, we mean any finite collection of tree-like PA-derivations which do not overlap, and d proves all root formulas of those derivations.

We define *open occurrences of a variable in a given proof tree* in PA-proofs. The idea of an 'open occurrence' is that it is open for substituting a number without destroying the proof tree: if x is open in a proof tree $\mathcal{T}(x)$ of a formula $A(x)$, then $\mathcal{T}(n)$ is a proof of $A(n)$. The following definition declares as open in a proof tree only those occurrences of variables which are safe to substitute throughout the whole tree.

1. All free occurrences of variables of root formulas are open.

2. If x is open in F, then all free occurrences of x are open in the immediate predecessors of F in the proof tree.

After the simultaneous substitution of n for an open variable x, the proof tree $\mathcal{T}(x)$ becomes a proof tree $\mathcal{T}(n)$, since each node in $\mathcal{T}(n)$ represents the same rule as in $\mathcal{T}(x)$. Moreover, if $\mathcal{T}(x)$ has a formula $A(x)$ at the root node, then $\mathcal{T}(n)$ has $A(n)$ at the root node.

Given a PA-proof d, we write $d(X)$ for d with marked open occurrences of variables from X.

Now let us see how open variables can be defined in a standard multi-conclusion derivation $\mathcal{D}$ which proves all formulas occurring in it. First, we assume that all derivations are presented in a *regular form*, which we define as follows.

1. Derivations are supplied with a tree-like proof of each of its formulas, and

2. These trees do not overlap, i.e., each occurrence of a formula in such belongs only to one of the trees.

It is obvious that each Hilbert-style proof can be presented in a regular form which proves exactly the same formulas as the original derivation. The specifics of converting a Hilbert-style derivation into its regular form do not matter.

4.2 Provability interpretation of FOLP

Let us fix a natural multi-conclusion Gödel proof predicate

Proof$(x, y) \rightleftharpoons$ *x is the Gödel number of a finite set of tree-like* PA*-derivations, y is the Gödel number of a root formula of one of those derivations.*

In what follows, by a proof in PA we mean a multi-conclusion proof in the sense of predicate *Proof*, that is, a finite set of tree-like PA-derivations, which proves root formulas of those derivations.

Let X be a set of individual variables. For each arithmetical formula F, by $[F(\underline{X})]$ we understand a natural arithmetical term for the primitive recursive function that for each value $K = (k_1, \ldots, k_n)$ of X returns the Gödel number of the result of substituting K for X in all free occurrences of $x_1, \ldots, x_n$ in F. Similarly, for each arithmetical derivation d, by $[d(\underline{X})]$ we mean an arithmetical term for the primitive recursive function that for each value $K = (k_1, \ldots, k_n)$ of X returns the Gödel number of the result of substituting K for X in all open occurrences of $x_1, \ldots, x_n$ in d. Formally,

$$[F(\underline{X})] \text{ is a natural arithmetical term for } \lambda K \ulcorner F(K) \urcorner,$$

and

$$[d(\underline{X})] \text{ is a term for } \lambda K \ulcorner d(K) \urcorner.$$

In particular, if $X = \emptyset$, then $[F(\underline{X})] = \ulcorner F \urcorner$ and $[d(\underline{X})] = \ulcorner d \urcorner$. We assume that if variable y is not free in F, then $[F(\underline{Xy})]$ and $[F(\underline{X})]$ graphically coincide, and similarly for d. We will also systematically skip brackets '[ ]' in $[F(\underline{X})]$ and $[d(\underline{X})]$ whenever it is safe.

As a notational example, consider arithmetical formulas $F(x,y)$ and $G(x)$. A natural arithmetical term for

$$\lambda n^{\ulcorner} F(n, {}^{\ulcorner} G(n)^{\urcorner})^{\urcorner}$$

in full notation will be

$$[F(\underline{x}, [G(\underline{\underline{x}})])],$$

and in simplified notation

$$F(\underline{x}, G(\underline{\underline{x}})).$$

Lemma 5 *For each* PA*-proof* d*, arithmetical formula* F*, and set of variables* X*, the following formulas are provable in* PA*:*

1. $\mathit{Proof}(d(\underline{X}), F(\underline{X})) \to F(X)$*;*
2. $\mathit{Proof}(d(\underline{Xy}), F(\underline{Xy})) \to \mathit{Proof}(d(\underline{X}), F(\underline{X}))$*,* $y \notin FVar(F)$*;*
3. $\mathit{Proof}(d(\underline{X}), F(\underline{X})) \to \mathit{Proof}(d(\underline{Xy}), F(\underline{Xy}))$.

Proof. To prove (1), reason in PA. Given X and the fact that $d(X)$ is a proof of $F(X)$, we conclude that $F(X)$ is nothing but a substitutional example of F such that d is a proof of F. Since d is a specific derivation, F follows by the standard parameter-free argument from the proof of correctness of the propositional Logic of Proofs [1, 2].

To verify (2), note that if d is a derivation of a formula F and y is not free in F, then y is not open in the tree-like proof of F provided by d. Hence, removing y from the set of global parameters of d does not alter the proof of F.

To prove (3), we need some straightforward combinatorics on derivations, naturally formalizable in PA. In particular, it reflects property (3) that a substitution of any number y for an open variable in a derivation d of a formula F is a legitimate derivation $d(y)$ of $F(y)$. □

Given *Proof*, we define natural recursive operations on PA-derivations that correspond to the functional symbols $\cdot$, $+$, !, and gen_x of FOLP.

Lemma 6 *There exist total recursive operations on proofs* $\cdot$*,* $+$*,* !*, and* gen_x *such that for any proofs* d *and* e*, formulas* F *and* G*, and a set of individual variables* X*, the following formulas are provable in* PA*:*

1. $\mathit{Proof}(d(\underline{X}), (F \to G)(\underline{X})) \to (\mathit{Proof}(e(\underline{X}), F(\underline{X})) \to \mathit{Proof}((d \cdot e)(\underline{X}), G(\underline{X})))$*;*
2. $\mathit{Proof}(d(\underline{X}), F(\underline{X})) \vee \mathit{Proof}(e(\underline{X}), F(\underline{X})) \to \mathit{Proof}((d{+}e)(\underline{X}), F(\underline{X}))$*;*
3. $\mathit{Proof}(d(\underline{X}, F(\underline{X})) \to \mathit{Proof}(!d(\underline{X}), \mathit{Proof}(d(\underline{\underline{X}}), F(\underline{\underline{X}})))$*;*
4. $\mathit{Proof}(d(\underline{X}), F(\underline{X})) \to \mathit{Proof}(\mathsf{gen}_x(d)(\underline{X}), \forall x F(\underline{X}))$*,* $x \notin X$.

Proof. In order to find $d \cdot e$, we take all formulas $F \to G$ proved by some proof tree $\mathcal{T}_1 \in d$ and F proved by a proof tree $\mathcal{T}_2 \in e$, and construct a new proof tree consisting of $\mathcal{T}_1$ and $\mathcal{T}_2$ followed by G obtained by *Modus Ponens*. For $d \cdot e$, we take the set of all proof trees obtained in this way.

By $d{+}e$ we understand the disjoint union of d and e.

Given a variable x, by $\mathsf{gen}_x(d)$ we mean the collection of tree-like arithmetical derivations each of which is a tree $\mathcal{T} \in d$ with a new root node $\forall x A$ where A is an 'old' root node of $\mathcal{T}$.

The only nontrivial case is 'verifier' ! which works as follows. Given a proof d, it recovers the set X of all parameters open in d. Then for each subset Y of X and for each formula F proved by d, it reconstructs the formula $\mathit{Proof}(d(\underline{Y}), F(\underline{Y}))$ with free variables Y. Since d is a proof of F, these formulas are all provable in PA. Operation '!' first finds a tree-like derivation for each of those formulas and finally, for $!d$, takes the set of all such derivations. The main purpose of $!d$ is to provide the proof of $\mathit{Proof}(d(\underline{Y}), F(\underline{Y}))$ for any $Y \subseteq X$.

It is clear that our definition of operations $\cdot$, $+$, !, and gen_x renders all formulas 1–4 true. Now let us show that they are provable in PA. All cases except, perhaps, the verifier are straightforward and amount to formalizing in PA some routine combinatorial arguments.

In the case of verifier '!,' we first prove (3) for the empty X: the formula

$$\mathit{Proof}([d], [F]) \to \mathit{Proof}([!d], [\mathit{Proof}([d], [F])])$$

is a true Δ_1-sentence, therefore it is provable in PA.

Now in the case $X \neq \emptyset$, reason in PA. Let us consider a representative example which will make the general argument rather clear. Let d be a PA-derivation $\{x = x\}$; d has one open variable x which determines that there are only two choices for Y from the definition of '!,' i.e., $Y = \emptyset$ and $Y = \{x\}$. Verifier '!' works as follows.

- Case $Y = \emptyset$. Find all formulas proved by d, in this case formula $x = x$, and find the derivation d_1 of formula $\mathit{Proof}(\ulcorner d \urcorner, \ulcorner x = x \urcorner)$.
- Case $Y = \{x\}$. In this case, $d(Y)$ is encoded by term $d(\underline{x})$ which for each $x = n$ returns $\ulcorner \{n = n\} \urcorner$. The only formula proved by it is $n = n$. This admits a direct formalization in PA and hence $\mathit{Proof}(d(\underline{x}), [\underline{x} = \underline{x}])$ is provable in PA with free variable x. Let d_2 be a derivation of this formula in PA.

For $!d$ we take the disjoint union of d_1 and d_2. Let us prove that this operation ! does what it should, i.e., that for all F and X,

$$\mathit{Proof}(d(\underline{X}), F(\underline{X})) \to \mathit{Proof}(!d(\underline{X}), \mathit{Proof}(d(\underline{\underline{X}}), F(\underline{\underline{X}}))).$$

Consider two cases of X and reason in PA.

- $x \notin X$ (e.g., $X = \emptyset$). In this case, $d(\underline{X})$ is just $\ulcorner d \urcorner$. The only formula for which d is a proof is $x = x$. By construction, $!d$ proves $\mathit{Proof}(\ulcorner d \urcorner, \ulcorner x = x \urcorner)$.
- $x \in X$. In this case, $d(\underline{X})$ is just $d(\underline{x})$, and the rest of the variables in X are of no consequence. The only formula proved by $d(\underline{x})$ is $[\underline{x} = \underline{x}]$, and, by construction, $!d(X)$ proves $\mathit{Proof}(d(\underline{x}), [\underline{x} = \underline{x}])$.

□

Definition 8 A *parametric arithmetical interpretation* for the language FOLP is defined by operations $+$, $\cdot$, $!$, and gen_x which satisfy Lemma 6 and an evaluation $*$ that maps

- proof variables and constants to multi-conclusion arithmetical proofs and
- predicate symbols of arity n to arithmetical formulas with n free variables. We suppose that $*$ commutes with the renaming of individual variables.

Now we can define the interpretation t^* of an FOLP-term t as follows: for proof variables and constants, t^* is given by the evaluation $*$, and we take $(s{\cdot}t)^*$ to be $s^*{\cdot}t^*$, $(s{+}t)^* = s^*{+}t^*$, $(\mathsf{gen}_x(t))^* = \mathsf{gen}_x(t^*)$, and $(!t)^* = !(t^*)$.

For formulas, $*$ commutes with Boolean connectives and quantifiers and

$$(t{:}_X F)^* = \mathit{Proof}(t^*(\underline{X}), F^*(\underline{X})),$$

i.e., $(t{:}_X F)^*$ is evaluated by the natural arithmetical formula asserting that t is a proof of F with global variables X.

Each derivation in FOLP generates *constant specification*, which is a (finite) set of formulas $c{:}A$ introduced by the axiom necessitation rule R3. We say that interpretation $*$ respects constant specification CS if all formulas from CS are true (hence provable in PA).

Theorem 3 [Arithmetical soundness] *If* $\mathsf{FOLP} \vdash A$ *with a constant specification* CS*, then for every parametric arithmetical interpretation* $*$ *respecting* CS*,* $\mathsf{PA} \vdash A^*$.

Proof. Induction on the proof of A in FOLP.

All instances of logical axioms A1 and rules R1 and R2 are trivially provable in PA. Validity of axioms B2–B5 is secured by the definition of operations $\cdot$, $+$, $!$, and gen_x; see Lemma 6. Rule R3 holds once we assume that constant specification CS is respected. Provability of A2, A3, and B1 is guaranteed by Lemma 5. □

The following Corollaries 3 and 4 provide a provability semantics for first-order modal logic $\mathsf{FOS4}$ and first-order intuitionistic logic HPC.

Corollary 3 *If* $\mathsf{FOS4}$ *proves* F*, then there exists a realization of* F *in* FOLP *which is a parametric provability tautology.*

Corollary 4 *If* HPC *proves* F*, then*

a) the Gödel translation of F*,* $tr(F)$*, is provable in* $\mathsf{FOS4}$*,*

b) there exists a realization of $tr(F)$ *in* FOLP *which is a parametric provability tautology.*

Example 2 Consider intuitionistic theorem

$$\exists x A(x) \to \neg\forall x \neg A(x) \quad \text{(where } A(x) \text{ is atomic)}. \tag{6}$$

Its simplified Gödel translation $()^\circ$ ([18], Section 9.2.1), is

$$\Box\exists x \Box A(x) \to \neg\Box\forall x \neg\Box A(x), \tag{7}$$

which is provable in FOS4: it is easy to guess its realization in FOLP:

$$u{:}\exists x v{:}_{\{x\}} A(x) \to \neg w{:}\forall x \neg v{:}_{\{x\}} A(x). \tag{8}$$

Indeed, (8) is a normal realization of (6) and it remains to check that (6) is provable in FOLP. Indeed, with $F = \forall x \neg v{:}_{\{x\}} A(x)$, formula (8) states $u{:}\neg F \to \neg w{:}F$ which is obviously provable in FOLP. By Theorem 3, any arithmetical interpretation of (8) is provable in PA.

5 Invariant parametric semantics

Arithmetical interpretation based on a specific proof predicate may yield provability tautologies that appear as a result of the specifics of proof numbering and as such do not represent 'true' provability tautologies that are invariant with respect to the choice of a proof predicate. This invariance issue has been discussed in the context of propositional logic of proofs in [2].

Example 3 Consider the formula

$$\neg u{:}\neg u{:}\bot. \tag{9}$$

Informally, it states that no u can be proof of a (valid) statement that u is not a proof of a contradiction. Intuitively, this does not seem right, since the arithmetical translation of $\neg u{:}\bot$ is a true decidable statement clearly provable in PA, and there is no reason to rule out a sophisticated u that can prove $\neg u{:}\bot$. Indeed, the propositional logic of proofs LP suggests that such a u exists: LP does not prove (9), hence, by the arithmetical completeness theorem for LP (cf. [2]), there is a proof predicate and evaluation of u under which (9) is false, hence $u{:}\neg u{:}\bot$ is true.

However, this intuition is not supported by the parametric provability semantics from Section 4 in which (9) is vacuously valid. Indeed, the standard Gödel numbering of formulas and proofs is monotonic and the code of the whole is strictly greater than the code of its proper part. Therefore, if $(u{:}\neg u{:}\bot)^*$ were true, then the code of u^* would be less than the code of $(\neg u{:}\bot)^*$ which is less than the code of u^* – a contradiction. This is a typical example of an 'accidental identity' that is based on the specifics of coding rather than on essence.

In order to avoid such 'identities,' we introduce the notion of *invariant parametric interpretation* which accepts as valid only those principles that hold for all legitimate numerations of proofs.

We consider the class of all proof predicates that are provably equivalent to the standard proof predicate but allow different numeration of proofs.

A *proof predicate* is a provably Δ_1-formula *Prf(x,y)* for which there are provably total recursive functions $\alpha(n)$ and $\beta(n)$ such that

PA $\vdash \forall x, y(\mathit{Proof}(x, y) \leftrightarrow \mathit{Prf}(\alpha(x), y))$, and

PA $\vdash \forall x, y(\mathit{Proof}(\beta(x), y) \leftrightarrow \mathit{Prf}(x, y))$.

Informally, α and β are computable translators from proofs in *Prf* to proofs of the same theorems in *Proof*, and vice versa.

It is convenient to not distinguish between derivations and their Gödel numbers. In what follows, if n is the Gödel number of the derivation d we write $[n](\underline{X})$, or even $n(\underline{X})$, for $[d(\underline{X})]$ and, equivalently, $d(\underline{X})$.

For each *Prf*-proof d and each set X of individual variables, $d(X)$ is a natural arithmetical term for a primitive recursive function that, for each value N of X, recovers $\beta(d)$ - the Gödel number of a regular *Proof*-derivation corresponding to d, substitutes N for X in $\beta(d)$, and computes back the *Prf*-number of the resulted derivation:

$$d(X) = \alpha(\beta(d)(\underline{X})). \tag{10}$$

An analogue of Lemma 5 holds for this notion of $d(X)$. In particular, for item 2 of the lemma, suppose that $Prf(d(Xy), F(\underline{Xy}))$. This last formula graphically coincides with $Prf(\alpha(\beta(d)(\underline{Xy})), F(\underline{Xy}))$, from which we obtain $Proof(\beta(d)(\underline{Xy}), F(\underline{Xy}))$. Since y is not free in F, by Lemma 5, $Proof(\beta(d)(\underline{X}), F(\underline{X}))$, from which $Prf(\alpha(\beta(d)(\underline{X})), F(\underline{X}))$, i.e., $Prf(d(X), F(\underline{X}))$.

One can define operations on proofs $\cdot$, $+$, !, and gen_x in such a way that Lemma 6 holds directly.

Definition 9 For any *Prf*-proofs a and b, and each finite set X of individual variables,

- $(a \cdot b)(X) = \alpha((\beta(a) \cdot \beta(b))(\underline{X}))$;
- $(a + b)(X) = \alpha((\beta(a) + \beta(b))(\underline{X}))$;
- $(\mathsf{gen}_x(a))(X) = \alpha((\mathsf{gen}_x(\beta(a)))(\underline{X}))$;
- $(!a)(X) = \alpha(!_0\beta(a)(\underline{X}))$

where functions $+$, $\cdot$, and gen_x are defined for *Proof* in the previous section, and $!_0$ is defined similarly with ! in such a way that it satisfies the following condition: for every derivation d, arithmetical formula F, and set of variables X,

$$Prf(\alpha(d(\underline{X})), F(\underline{X})) \to Proof(!_0 d(\underline{X}), Prf(\alpha(d(\underline{\underline{X}})), F(\underline{\underline{X}}))).$$

Remark 2 Let us show that such a function $!_0$ exists. Briefly speaking, it works as follows. Given a derivation d, it recovers the set of formulas proved by d and the set X of open variables of d. For each F of these formulas and each $Y \subseteq X$, we have $Proof(d(\underline{Y}), F(\underline{Y}))$ which is equivalent to $Prf(\alpha(d(\underline{Y})), F(\underline{Y}))$. Formula $Proof(d(\underline{Y}), F(\underline{Y}))$ and, therefore, $Prf(\alpha(d(\underline{Y})), F(\underline{Y}))$ is provable in PA with free variables Y; let e be the proof of the latter. As $!_0 d$ we take the disjoint union of such proofs e for all formulas F proved by d and all subsets Y of open variables in d.

The direct analogue of Lemma 6 holds for operations from Definition 9. For '$\cdot$' we reason as follows. If $Prf(a(X), (F \to G)(\underline{X}))$ and $Prf(b(X), F(\underline{X}))$, which is the same as $Prf(\alpha(a'(\underline{X})), (F \to G)(\underline{X}))$, and $Prf(\alpha(b'(\underline{X})), F(\underline{X}))$, then

$$Proof(a'(\underline{X}), (F \to G)(\underline{X})) \text{ and } Proof(b'(\underline{X}), F(\underline{X})).$$

Therefore, by Lemma 6, $Proof((a'\cdot b')(\underline{X}), G(\underline{X}))$, from which $Prf(\alpha((a'\cdot b')(\underline{X})), G(\underline{X}))$. But $\alpha((a'\cdot b')(\underline{X}))$ coincides with $(a\cdot b)(X)$, and we conclude the desired $Prf((a\cdot b)(X), G(\underline{X}))$.

We reason similarly in the case of '$+$' and 'gen_x.'

For '!,' assume that $Prf(a(X), F(\underline{X}))$, where $a(X) = \alpha(a'(\underline{X}))$. Then, by definition of $!_0$, we obtain $Proof(!_0 a'(\underline{X}), Prf(a(\underline{X}), F(\underline{\underline{X}})))$, therefore

$$Prf(\alpha(!_0 a'(\underline{X})), Prf(a(\underline{X}), F(\underline{\underline{X}}))),$$

i.e.,

$$Prf((!a)(X), Prf(a(\underline{X}), F(\underline{\underline{X}}))).$$

Definition 10 Given a proof predicate Prf and operations on proofs $\cdot$, $+$, !, and gen_x that satisfy Lemma 6, *invariant parametric interpretation* for the language FOLP is an evaluation $*$ that maps

- proof variables and constants to natural numbers (which should be thought of as Prf-numbers of regular arithmetical derivations) and
- predicate symbols of arity n to arithmetical formulas with n free variables. We suppose that $*$ commutes with renaming of individual variables.

For terms, $*$ commutes with operations $\cdot$, $+$, !, and gen_x. For formulas, $*$ commutes with the Boolean connectives and quantifiers and

$$(t{:_X}F)^* = Prf(t^*(X), F^*(\underline{X})).$$

Theorem 4 [Arithmetical soundness] *If* $\mathsf{FOLP} \vdash A$ *with a constant specification CS, then for every invariant parametric interpretation* $*$ *respecting CS,* $\mathsf{PA} \vdash A^*$.

Proof. Follows immediately from analogues of Lemma 5 and Lemma 6 adapted for the invariant setting. □

Let us reconsider formula (9) and show that it is not valid in the invariant parametric semantics. For this we have to find a proof predicate Prf and interpretation $*$ such that $(u{:}\neg u{:}\bot)^*$ holds (provable in PA).

In what follows, we assume that an injective numeration of the joint syntax of FOLP and PA is given. Consider the following fixed-point equation that defines an arithmetical predicate $Prf(x, y)$.

$$Prf(x, y) \;\leftrightarrow\; Proof(x, y) \vee (x = \ulcorner u \urcorner \;\wedge\; y = \ulcorner \neg Prf(\ulcorner u \urcorner, \ulcorner \bot \urcorner) \urcorner). \tag{11}$$

From (11), it immediately follows that $Prf(x, y)$ is provably Δ_1. Moreover, it is also clear from (11) that $\neg Prf(\ulcorner u \urcorner, \ulcorner \bot \urcorner)$ holds and let p be the Gödel number of its proof in PA. So, $Proof(p, \ulcorner \neg Prf(\ulcorner u \urcorner, \ulcorner \bot \urcorner) \urcorner)$. Let α and β be identity functions except for

$$\beta(\ulcorner u \urcorner) = p \quad \text{and} \quad \alpha(p) = \ulcorner u \urcorner.$$

From (11), it follows that the following are provable in PA with a free variable y:

$$Proof(p, y) \leftrightarrow Prf(\alpha(p), y)$$

and

$$Proof(\beta(\ulcorner u \urcorner), y) \leftrightarrow Prf(\ulcorner u \urcorner, y).$$

These facts are both provable in PA, hence

$$\mathsf{PA} \vdash \forall x, y(Proof(x, y) \leftrightarrow Prf(\alpha(x), y)) \text{ and } \mathsf{PA} \vdash \forall x, y(Proof(\beta(x), y) \leftrightarrow Prf(x, y)).$$

We conclude that $Prf(x, y)$ is a legitimate proof predicate. We now define the interpretation $*$ that interprets u as $\ulcorner u \urcorner$. From (11), $(u{:}\neg u{:}\bot)^*$ holds (provable in PA), hence formula (9) is not a valid provability principle in the invariant parametric semantics.

6 Further valid principles of proofs: Barcan formula

In this section, we show that a natural explicit version of the Barcan formula is valid under parametric and invariant parametric semantics.

Theorem 5 *$\forall y(t{:}_{Xy}A) \to t{:}_X A$ is valid in parametric and invariant parametric semantics.*

Proof. We first establish the validity of this principle in parametric semantics.

Lemma 7 *Let $A(x)$ and $B(x)$ be arithmetical formulas, and suppose for two distinct numerals n_1 and n_2, $A(n_i)$ syntactically coincides with $B(n_i)$. Then $A(x)$ coincides with $B(x)$.*

Proof. Actually, this is an exercise in unification. Run the unification algorithm on $A(x) = B(x)$. If unification succeeds, i.e., yields an empty set of equations, then $A(x)$ coincides with $B(x)$. Otherwise, if the unification yields an equation $s = t$ for syntactically different s and t, none of which is x, there is no way $A(n)$ syntactically coincides with $B(n)$ for any n which contradicts the assumptions. Suppose the unification algorithm reduces $A(x) = B(x)$ to a finite system of equations $x = t_1$, $x = t_2$,... where all t_i are pairwise distinct arithmetical terms, none of which contains x. If we substitute n for x in this algorithm, the equality $A(n) = B(n)$ will be reduced to $n = t_1$, $n = t_2, \ldots$. Since all t_i are distinct, there can be at most one valid equation in this system, say $n = t_1$. Since n is arbitrary, the equality $A(n) = B(n)$ can hold only when n coincides with t_1, which contradicts the assumptions. □

Lemma 8 *Let $p(x)$ be a derivation in PA, and $Q(x)$ an arithmetical formula. If for all $n = 0, 1, 2, ...$, $p(n)$ is a derivation for $Q(n)$, then p is a derivation for Q with x as a local variable.*

Proof. Suppose $p(x) = F_1(x), F_2(x), ..., F_k(x)$. From the assumptions, for each n there is an i such that $Q(n) = F_i(n)$. By the Pigeonhole Principle, there is an i such that $Q(n) = F_i(n)$ for two different n's. By Lemma 7, $Q(x) = F_i(x)$. □

Lemma 9 *Principle* $\forall y(t{:}_{Xy}A) \to t{:}_X A$ *is derivable in* PA *for each parametric evaluation* $*$.

Proof. Note that both Lemma 7 and Lemma 8 are formalizable in PA. Reason in PA. Suppose for all y, $t^*(X,y)$ is a proof of $A^*(X,y)$. By Lemma 8, $A^*(X,y)$ is in $t^*(X)$ where y is a local variable. Therefore, $t^*(X)$ is a proof of $A^*(X)$. □

This proves Theorem 5 for the parametric provability semantics. In the case of the invariant parametric semantics, the proof is as follows. Reason in PA. Suppose that

$$\forall y Prf(t^*(Xy), A^*(\underline{Xy})).$$

Then, by (10), $\forall y Prf(\alpha(d(\underline{Xy})), A^*(\underline{Xy}))$ for some *Proof*-derivation d. Then

$$\forall y Proof(d(\underline{Xy}), A^*(\underline{Xy})),$$

and, as in the parametric case, $Proof(d(\underline{X}), A^*(\underline{X}))$, hence $Prf(\alpha(d(\underline{X})), A^*(\underline{X}))$, i.e.,

$$Prf(t^*(X), A^*(\underline{X})).$$

□

Definition 11 First-order logic of proofs FOLPb is FOLP with $\forall y(t{:}_{Xy}A) \to t{:}_X A$ as an additional axiom.

Note that FOLPb derives an explicit version of the Barcan formula $\forall x \Box A \to \Box \forall x A$, namely

$$\forall x(t{:}_{Xx}A) \to \mathsf{gen}_x(t){:}_X \forall x A.$$

Corollary 5 FOLPb *is correct with respect to the parametric and invariant semantics.*

7 Generic provability semantics for FOLP

In (invariant) parametric semantics for FOLP, proof terms are interpreted as specific derivations with open variables. As a result, an explicit version of the Barcan formula holds. However, the intuitive provability semantics for first-order modal logic offers a somewhat different account of the Barcan formula $\forall x \Box A \to \Box \forall x A$. According to this intuition, if $A(x)$ is provable for each x, it does not guarantee that $\forall x A(x)$ is provable. In this section, we offer a generic provability semantics for FOLP that accommodates this kind of reasoning.

As in [2], we consider the class of all proof predicates. A *generic proof predicate* is a provably Δ_1-formula $Prf(x,y)$ such that for every arithmetical formula φ,

$$\mathsf{PA} \vdash \varphi \quad \Leftrightarrow \quad \textit{for some } n \in \omega,\ Prf(n, \ulcorner \varphi \urcorner) \textit{ holds.} \tag{12}$$

Here n is called a proof and φ a formula proved by n. One may think of n as a label for a set of provable formulas $\widehat{n} = \{\varphi \mid Prf(n, \ulcorner \varphi \urcorner)\}$. So $\cup_{n \in \omega} \widehat{n}$ is the set of all theorems of PA. Sets

$\widehat{n}$ are assumed finite, and a function from n to $\widehat{n}$ provably computable. For convenience, we allow $\widehat{n}$ to be empty for some n, e.g., let $\widehat{0} = \emptyset$. As in Section 5, we assume that there exist total recursive functions $\alpha(n)$ and $\beta(n)$ which translate proofs in *Prf* to proofs of the same theorems in *Proof*, and vice versa, i.e.,

$$\mathsf{PA} \vdash \forall x, y(\mathit{Proof}(x, y) \leftrightarrow \mathit{Prf}(\alpha(x), y)), \text{ and } \mathsf{PA} \vdash \forall x, y(\mathit{Proof}(\beta(x), y) \leftrightarrow \mathit{Prf}(x, y)).$$

We also make a simplifying assumption: *for each finite set Y of theorems of* PA*, there exists n such that $Y = \widehat{n}$ and functions from the standard derivation of Y to n and back are computable.* This property holds for the standard regular proof predicate: for any finite set of tree-like derivations, there is a proof (its disjoint union) which contains exactly their proof trees. This assumption is not really necessary and can be replaced by a weaker condition.

The following notion of a *proof function* is a generic analogue of the notion of a proof with a given set of global parameters.

Definition 12 Given a generic proof predicate *Prf* and a set of individual variables X, a *proof function* is a pair $(p(X), \mathcal{F})$ such that

1. $p(X)$ is a provably total recursive function from the set of values of X to *Prf*-proofs, fairly represented in PA by a term $p(X)$; $\mathcal{F} = (F_1, \ldots, F_n)$ is a finite set of arithmetical formulas (considered provable by this proof function).

2. PA 'knows' that for all values of X, $p(X)$ proves substitutional examples of formulas from $\mathcal{F}$ and only them, that is,

$$\mathsf{PA} \vdash \mathit{Prf}(p(X), y) \leftrightarrow \bigvee_{i=1}^{n} (y = [F_i(\underline{X})]); \tag{13}$$

3. PA 'knows' that each formula proved by $p(X)$ actually holds, i.e., for each formula F from $\mathcal{F}$,

$$\mathsf{PA} \vdash \mathit{Prf}(p(X), F(\underline{X})) \rightarrow F. \tag{14}$$

As a notational convention, we will speak about a proof function $p(X)$ and the set of formulas $\widehat{p}$ as $\mathcal{F}$ from the definition. Note that formulas from $\widehat{p}$ may have free variables other than from X, or do not have some variables from X; these nuances are automatically handled by notations $[F(\underline{X})]$.

Example 4 Each *Prf*-proof d may be regarded as a proof function for each X with the set of formulas $\mathcal{F}$ being the set of all formulas proven by d. In particular, Lemma 5 adapted for the invariant setting yields that (14) holds in this case, too.

There are proof functions that are not substitutional instances of specific derivations; we will see examples later in this section (i.e., in Examples 5 and 6).

Remark 3 Given $Z \subseteq X$, we sometimes need to regard a proof function $p(Z)$ as a proof function of X. Namely, for $U = X \setminus Z$ we state the existence of a proof function $p^U(X)$ for which $\widehat{p^U} = \widehat{p}$.

Given X, we calculate p^U as follows:

1. calculate $p(Z)$;
2. find a derivation $d = \beta(p(Z))$ which proves exactly the same formulas;
3. substitute given values of U in d to get $[\beta(p(Z))](U)$;
4. find a Prf-proof $\alpha([\beta(p(Z))](U))$.

So, we define the value of $p^U(X)$ as $\alpha([\beta(p(Z))](U))$. It is easy to derive in PA that

$$Prf(p(Z), A(\underline{Z})) \leftrightarrow Prf(p^U(X), A(\underline{X}))$$

using the following equivalences:

$$\begin{array}{lcll} Prf(p(Z), A(\underline{Z})) & \leftrightarrow & Proof(\beta(p(Z)), A(\underline{Z})) & \leftrightarrow \\ & \leftrightarrow & Proof([\beta(p(Z))(\underline{U})], A(\underline{ZU})) & \leftrightarrow \\ & \leftrightarrow & Prf(\alpha([\beta(p(Z))](\underline{U})), A(\underline{X})). & \end{array}$$

The reflexivity condition also holds. Argue in PA. Given X, Z, and $Prf(p^U(X), F(\underline{X}))$ we conclude $Prf(p(Z), F(\underline{Z}))$ whence F holds by reflexivity of p.

In what follows, in particular, the proof of Lemma 15, when speaking about a sum of $p(X)$ and $q(Z)$ with $Z \subseteq X$ we mean the sum of $p(X)$ and $q^U(X)$.

The notion of *proof form* from Definition 13 is a generic analogue of the notion of a proof as a finite family of derivations with global parameters X from a fixed set Y.

Definition 13 Fix a proof predicate Prf and a finite set of variables Y. By a *proof form* $\{p_X(X)\}$ we understand a family of proof functions $p_X(X)$, one for each $X \subseteq Y$ such that the following two properties are provable in PA for every arithmetical formula A:

- *Monotonicity* : $\widehat{p_X} \subseteq \widehat{p_{Xy}}$.
- *Coherence*: $\widehat{p_{Xy}} \setminus \widehat{p_X}$ consists only of formulas in which y occurs free.

Lemma 10 *Given a proof predicate Prf, finite set of variables Y, and a proof form $\{p_X(X)\}$,*

1. $\mathsf{PA} \vdash Prf(p_X(X), A(\underline{X})) \to Prf(p_{Xy}(Xy), A(\underline{Xy}))$.
2. $\mathsf{PA} \vdash Prf(p_{Xy}(Xy), A(\underline{X})) \to Prf(p_X(X), A(\underline{X}))$ *if y is not free in A.*

Proof. Argue in PA.

1) $Prf(p_X(X), A(\underline{X}))$ yields that $A(X)$ coincides with $F(X)$ for some formula $F \in \widehat{p_X}$. By monotonicity, $F \in \widehat{p_{Xy}}$. It remains to observe that once $A(X)$ syntactically coincides with $F(X)$, then $A(Xy)$ syntactically coincides with $F(Xy)$. Therefore $Prf(p_{Xy}(Xy), A(\underline{Xy}))$.

2) Let $Prf(p(Xy), A(X))$, y is not free in $A(X)$. Then $A(X)$ coincides with $F(Xy)$ for some formula $F \in \widehat{p_{Xy}}$. Since y is not free in $A(X)$, y is not free in $F(Xy)$ either and $[F(\underline{Xy})]$

is provably equal to $[F(\underline{X})]$. By coherence, $F \in \widehat{p_X}$. By assumptions, $Prf(p_X(X), F(\underline{X}))$, hence $Prf(p_X(X), A(\underline{X}))$. $\Box$

Note that for each Prf-proof p and each Y, the set of invariant Prf-proofs $\{p(X) \mid X \subseteq Y\}$ as defined in (10) is a legitimate proof form. This is a corollary of Lemma 5 adapted for the invariant interpretation.

In parametric semantics, proof terms are interpreted as real derivations with a mechanism of opening/closing variables, and operations on proof terms as operations on these derivations. In generic semantics, the analogue of a derivation is a proof form (rather than a proof function) and we need to define operations corresponding to functional symbols of FOLP on proof forms. In the future definition of generic arithmetical interpretation (Definition 14), operations, along with the proof predicate Prf, will be parameters of the definition. However, we will need a 'canonical' example of such operations.

We first define axiliary operations $+$, $\cdot$, !, and gen_x on proof functions, that is, for proof functions $p(X)$ and $q(X)$ we will construct proof functions $[p+q](X)$, $[p \cdot q](X)$, $[!p](X)$, and $[\mathsf{gen}_x(p)](X)$ in such a way that for each value of X

- $[p + q](X)$ is the Prf-proof of formulas from $\widehat{p} \cup \widehat{q}$ and only them;
- $[p \cdot q](X)$ is the Prf-proof of all G such that $F \to G \in \widehat{p}$ for some $F \in \widehat{q}$ and only them;
- $[!p](X)$ is the Prf-proof of $Prf(p(X), \ulcorner G \urcorner)$ for all $G \in \widehat{p}$ and only for them;
- $[\mathsf{gen}_x(p)](X)$ is the Prf-proof deriving $\forall x F$ for all $F \in \widehat{p}$ with $x \notin X$ and only for them.

As one can see, no irrelevant formulas occur in the compound proofs: we will call such operations 'tight.' Now we are going to establish two facts: first, that tight operations applied to proof functions result in a proof function, and that they satisfy specification axioms of FOLP.

Let $x \dot{\to} y$ denote a natural term for the primitive recursive function which calculates $\ulcorner F \to G \urcorner$ given $\ulcorner F \urcorner$ and $\ulcorner G \urcorner$. Similarly, let $\dot{\forall}_x y$ denote a natural term for the primitive recursive function which calculates $\ulcorner \forall x F \urcorner$ given $\ulcorner F \urcorner$ and $\ulcorner x \urcorner$.

Lemma 11 *Let $p(X)$ and $q(X)$ be provably recursive functions. Then there exist provably recursive functions $[p + q](X)$, $[p \cdot q](X)$, $[!p](X)$, and $[\mathsf{gen}_x(p)](X)$ such that the following formulas are provable in* PA*:*

- $Prf(p(X), y) \vee Prf(q(X), y) \to Prf([p + q](X), y)$;
  $Prf([p + q](X), y) \to Prf(p(X), y) \vee Prf(q(X), y)$;
- $Prf(p(X), y \dot{\to} z) \to (Prf(q(X), y) \to Prf([p \cdot q](X), z))$;
  $Prf([p \cdot q](X), z) \to \exists y\, (Prf(p(X), y \dot{\to} z) \wedge Prf(q(X), y))$;
- $Prf(p(X), y) \to Prf([!p](X), Prf(p(\underline{X}), \underline{y}))$;
  $Prf([!p](X), z) \to \exists y \left(z = Prf(p(\underline{X}), \underline{y}) \wedge Prf(p(X), y)\right)$;

- $Prf(p(X), y) \to Prf([\mathsf{gen}_x(p)](X), \dot\forall_x y)$, $x \notin X$;
  $Prf([\mathsf{gen}_x(p)](X), z) \to \exists y \Big(z = \dot\forall_x y \wedge Prf(p(X), y)\Big)$, $x \notin X$.

Proof. Specific formalizations of $+, \cdot, \mathsf{gen}_x$ are straightforward. For example, for $[p \cdot q](X)$ one can take $\alpha(\beta(p(X)) \cdot \beta(q(X)))$ where α and β are translators from *Proof*-proofs to *Prf*-proofs and vice versa. Formulas from $\widehat{[p \cdot q]}$ are exactly those G for which there is an F in $\widehat{q}$ such that $F \to G \in \widehat{p}$.

Let us describe how to build $[!p](X)$. We first recall a classical result:

Lemma 12 [Gödel's Lemma] (cf. [5, 17]) *For each provably Δ_1-formula $\sigma(X)$, there is a provably recursive function $g(X)$ such that*

$$\mathsf{PA} \vdash \sigma(X) \to Proof(g(X), \sigma(\underline{X})).$$

The key observation here is that Gödel's Lemma can be converted.

Lemma 13 [Two-way Gödel's Lemma] *For each provably Δ_1-formula $\sigma(X)$, there is a provably recursive function $t(X)$ such that*

$$\mathsf{PA} \vdash Proof(t(X), \sigma(\underline{X})) \leftrightarrow \sigma(X).$$

Proof. Direction '$\leftarrow$' is similar to the classical Gödel's Lemma. A tedious analysis of the proof of Gödel's Lemma shows that the converse implication is also provable:

$$\mathsf{PA} \vdash Proof(g(X), \sigma(\underline{X})) \to \sigma(X),$$

which yields the Lemma. However, we offer here an alternative shorter proof of Lemma 13. Given $g(X)$ from Gödel's Lemma, define $t(X)$ to be a natural arithmetical term for a provably recursive function that is equal to $g(X)$ if $\sigma(X)$ holds, and to 0 (which is a proof of nothing) otherwise. Therefore,

$$\mathsf{PA} \vdash \sigma(X) \to Proof(t(X), \sigma(\underline{X})).$$

We claim that

$$\mathsf{PA} \vdash \neg\sigma(X) \to \neg Proof(t(X), \sigma(\underline{X}))$$

as well. Reason in PA. If not $\sigma(X)$, then $t(X) = 0$, hence $t(X)$ is not a proof of $\sigma(X)$. □

Remark 4 In a private discussion of the preliminary draft of this paper, Lev Beklemishev offered the following alternative proof of Lemma 13. For a given Δ_1-formula $\sigma(X)$, consider a provably recursive term $g(X)$ constructed in the proof of Gödel's Lemma such that

$$\mathsf{PA} \vdash \sigma(X) \to Proof(g(X), \sigma(\underline{X}))$$

for some $g(X)$ that is the code of the particular proof in PA. Detailed analysis of the construction demonstrates that in fact $g(X)$ is the code of a derivation in Robinson's Arithmetic Q. Since this observation is purely syntactical, it can be formalized in PA, that is,

$$\mathsf{PA} \vdash Proof_{\mathsf{PA}}(g(X), y) \to Proof_{\mathsf{Q}}(g(X), y).$$

But PA proves reflection for Q, in particular,

$$\mathsf{PA} \vdash \mathit{Proof}_{\mathsf{Q}}(g(X), \sigma(\underline{X})) \to \sigma(X).$$

This gives us the desired

$$\mathsf{PA} \vdash \mathit{Proof}_{\mathsf{PA}}(g(X), \sigma(\underline{X})) \to \sigma(X).$$

We now proceed with building the proof checker operation '!.' Consider a proof function $(p(X), \mathcal{F})$. Since for each F, $\mathit{Prf}(p(X), F(\underline{X}))$ is a provably Δ_1-formula, using Lemma 13, one can build a provably recursive term $t(X)$ that provides a Proof-proof of $\mathit{Prf}(p(X), F(\underline{X}))$, i.e.,

$$\mathsf{PA} \vdash \mathit{Prf}(p(X), F(\underline{X})) \to \mathit{Proof}(t(X), \mathit{Prf}(p(\underline{X}), F(\underline{\underline{X}}))).$$

We can assume that $t(X)$ is a cumulative proof that fits all $F \in \mathcal{F}$ and hence depends only on the proof function. By assumptions about generic proof predicates, given a Proof-proof $t(X)$, we can construct a Prf-proof $s(X)$ of the same set of formulas, i.e.,

$$\mathsf{PA} \vdash \mathit{Prf}(p(X), F(\underline{X})) \to \mathit{Prf}(s(X), \mathit{Prf}(p(\underline{X}), F(\underline{\underline{X}}))).$$

We can now take this $s(X)$ as $[!p](X)$. In other words, operation '!' proceeds as follows: given specific values of X and p, it

- computes $n = p(X)$ and $k_i = F_i(\underline{X})$ for all $F_i \in \mathcal{F}$;
- builds formulas G_i which are $\mathit{Prf}(n, k_i)$;
- finds Proof-proofs $g(n, \ulcorner G_i \urcorner)$ of $\mathit{Prf}(n, \ulcorner G_i \urcorner)$'s, by two-way Gödel's Lemma;
- finds a Proof-proof of all of these $\mathit{Prf}(n, \ulcorner G_i \urcorner)$'s. There is a computable procedure which, for a given code of a finite set of Proof-proofs, builds a code of a Proof-proof that is their sum. This is our $t(X)$;
- and finally, given a Proof-proof $t(X)$, we build an equivalent Prf-proof $s(X)$ and take it as $[!p](X)$.

It is clear that all the functions constructed in this proof are total recursive. We assume that formalization of operations is natural, so, PA 'knows' that they satisfy the corresponding specifications. □

We will now prove that the tight operations defined above, when applied to proof functions, return proof functions.

Lemma 14 *Let $p(X)$ and $q(X)$ be proof functions. Then $[p+q](X)$, $[p \cdot q](X)$, $[!p](X)$, and $[\mathsf{gen}_x(p)](X)$ are proof functions.*

Proof. It is clear that property (13) is preserved by operations $+$, $\cdot$, $!$, and gen_x and that the corresponding sets $\mathcal{F}$ can be found effectively from the components of the operation. Given this, we will focus on proving (14). The case of '$+$' is straightforward. For $[p \cdot q](X)$, reason in PA. Assume that $Prf([p \cdot q](X), F(\underline{X}))$; then there exists y such that $Prf(p(X), y \dot{\rightarrow} F(\underline{X}))$ and $Prf(q(X), y)$. By definition of a proof function, from $Prf(q(X), y)$ we conclude that

$$\bigvee_{i=1}^{k} (y = G_i(\underline{X})),$$

therefore

$$\bigvee_{i=1}^{k} \left(Prf(p(X), G_i(\underline{X}) \rightarrow F(\underline{X})) \wedge Prf(q(X), G_i(\underline{X}))\right).$$

From property (14) for $p(X)$ and $q(X)$ we conclude

$$\bigvee_{i=1}^{k} ((G_i \rightarrow F) \wedge G_i).$$

By propositional logic, we derive F.

Let us check '$!$.' We have to prove that

$$\mathsf{PA} \vdash Prf([!p](X), Prf(p(\underline{X}), F(\underline{X}))) \rightarrow Prf(p(X), F(\underline{X})). \tag{15}$$

Argue in PA. Assume $Prf([!p](X), H(\underline{X}))$. Since operation '$!$' is tight, $H(X)$ is one of $Prf(p(X), G)$ where $G \in \widehat{p}$. By two-way Gödel's Lemma, $H(X)$ holds.

Checking gen_x. Argue in PA. Assume $Prf([\mathsf{gen}_x(p)](X), F(\underline{X}))$. By the definition of operation gen_x, $F(X)$ is $\forall x G(X)$ with $x \notin X$ and

$$Prf(p(X), G(\underline{X})).$$

Since $p(X)$ is a proof function, we have $Prf(p(X), G(\underline{X})) \rightarrow G$. Use generalization on x and take into account that x is not free in the antecedent. Conclude that

$$Prf(p(X), G(\underline{X})) \rightarrow \forall x G,$$

from which we obtain the desired

$$Prf([\mathsf{gen}_x(p)](X), \forall x G(\underline{X})) \rightarrow \forall x G.$$

□

We are now ready to define operations on proof forms.

Lemma 15 *Suppose that Y is a finite set of parameters and $\mathcal{P} = \{p_X \mid X \subseteq Y\}$, $\mathcal{Q} = \{q_X \mid X \subseteq Y\}$ are proof forms. Then one can effectively find proof forms $\mathcal{P} \cdot \mathcal{Q}$, $\mathcal{P} + \mathcal{Q}$, $!\mathcal{P}$ and $\mathsf{gen}_x\mathcal{P}$ such that for each $X \subseteq Y$, the following formulas are provable in PA:*

- $Prf(p_X(X), y) \vee Prf(q_X(X), y) \to Prf([\mathcal{P} + \mathcal{Q}]_X(X), y)$;
- $Prf(p_X(X), y \dot{\to} z) \to (Prf(q_X(X), y) \to Prf([\mathcal{P} \cdot \mathcal{Q}]_X(X), z))$;
- $Prf(p_X(X), y) \to Prf([!\mathcal{P}]_X(X), Prf(p_X(\underline{X}), \underline{y}))$;
- $Prf(p_X(X), y) \to Prf([\mathsf{gen}_x(\mathcal{P})]_X(X), \dot{\forall}_x y)$, $x \notin X$.

Proof. Sum '+' is defined as the usual 'tight' operation on corresponding proof functions $p_X(X)$; no coordination for different X's is needed, i.e., $[\mathcal{P} + \mathcal{Q}]_X = p_X + q_X$ in the sense of Lemma 14. It is easy to check that $\mathcal{P} + \mathcal{Q}$ is a proof form, that is, it is monotonic and coherent.

Application '$\cdot$' works as follows. For every $X \subseteq Y$ we first calculate $[p_Z \cdot q_Z](Z)$ for each $Z \supseteq X$. For every such Z, we then find the common Prf-proof $t_Z(Z)$ for all formulas F proven by $[p_Z \cdot q_Z](Z)$ which satisfy the condition $FVar(F) \cap (Z \setminus X) = \emptyset$. Then we define $[\mathcal{P} \cdot \mathcal{Q}]_X(X)$ as the tight sum of all such t_Z (such a proof can be effectively found by the definition of a generic proof predicate).

Coherence follows immediately from the condition on free variables in the definition of application. Let us check monotonicity. Argue in PA. Suppose that $Prf([\mathcal{P} \cdot \mathcal{Q}]_X(X), F(\underline{X}))$. By the definition of $[\mathcal{P} \cdot \mathcal{Q}]_X$, there is $Z \supseteq X$ such that F does not contain free variables from $Z \setminus X$ and $Prf([\mathcal{P} \cdot \mathcal{Q}]_Z(Z), F(\underline{X}))$. If $y \in Z$, then $Z \supseteq Xy$, hence $FVar(F) \cap (Z \setminus Xy) = \emptyset$ and $Prf([\mathcal{P} \cdot \mathcal{Q}]_{Xy}(Xy), F(\underline{Xy}))$ by the definition of $[\mathcal{P} \cdot \mathcal{Q}]_{Xy}$.

Let $y \notin Z$. We have $Prf(p_Z(Z), (G \to F)(\underline{Z}))$ and $Prf(q_Z(Z), G(\underline{Z}))$ for some G. Then, by monotonicity of $\mathcal{P}$ and $\mathcal{Q}$, we obtain $Prf(p_{Zy}(Zy), (G \to F)(\underline{Zy}))$ and $Prf(q_{Zy}(Zy), G(\underline{Zy}))$, hence by definition of '$\cdot$' on proof functions, $Prf([p_{Zy} \cdot q_{Zy}](\underline{Zy}), F(\underline{Zy}))$. Now we have $Zy \supseteq Xy$ and F does not contain free variables from $Zy \setminus Xy$, thus, by definition of $\mathcal{P} \cdot \mathcal{Q}$, $Prf([\mathcal{P} \cdot \mathcal{Q}]_{Xy}(Xy), F(\underline{Xy}))$.

Proof checker. For every $X \subseteq Y$, we define $[!\mathcal{P}]_X(X)$ as the tight sum of $[!(p_Z)](Z)$ for all $Z \subseteq X$. Monotonicity is immediate since the definition of $[!\mathcal{P}]_X(X)$ is monotone with respect to X.

Coherence. Argue in PA. Suppose $[!\mathcal{P}]_{Xy}(Xy)$ is a proof of $A(X)$ and $A(X)$ does not contain y free. By definition of $[!\mathcal{P}]_{Xy}$, A has the form $Prf(p_Z(Z), F(\underline{Z}))$ for some $Z \subseteq Xy$, where $F(\underline{Z}) \in \widehat{p_Z}$. Since $FVar(A) = Z$ and Z does not contain y free, we conclude that $Z \subseteq X$. Then $[!\mathcal{P}]_X(X)$ is a proof of $A(\underline{X})$ by the definition of $[!\mathcal{P}]_X$.

Generalizer. Let us extend operation gen_x to all proof functions $p(X)$. If $x \notin X$, then $[\mathsf{gen}_x(p)](X)$ is defined in Lemma 11. If $x \in X$, then $[\mathsf{gen}_x(p)](X)$ is equal to 0 which does not prove anything, so $\widehat{\mathsf{gen}_x(p)} = \emptyset$. Reflexivity of $[\mathsf{gen}_x(p)](X)$ follows immediately from reflexivity of $p(X)$.

Given $X \subseteq Y$ and proof form $\mathcal{P}$, we define $[\mathsf{gen}_x(\mathcal{P})]_X$ as the tight sum of $[\mathsf{gen}_x(\mathcal{P}_Z)](Z)$ for all $Z \subseteq X$.

Checking monotonicity. Suppose $F(X) \in \widehat{[\mathsf{gen}_x(\mathcal{P})]_X}$. Then $F(X)$ is the result of opening parameters X in some formula $F \in \widehat{[\mathsf{gen}_x(\mathcal{P}_Z)]}$. By definition of the operation gen_x, and since $Z \subset \{Xy\}$, $F(Xy) \in \widehat{[\mathsf{gen}_x(\mathcal{P})]_{Xy}}$.

Let us check coherence. Suppose $F \in [\widehat{\mathsf{gen}_x(\mathcal{P})}]_{Xy}$ and $y \notin FVar(F)$. Then $F \in [\widehat{\mathsf{gen}_x(\mathcal{P}_Z)}]$ for some $Z \subseteq Xy$ such that $x \notin Z$.

Case 1: $Z \subseteq X$. Then $F \in [\widehat{\mathsf{gen}_x(\mathcal{P})}]_X$ by the definition of $[\mathsf{gen}_x(\mathcal{P})]_X$.

Case 2: $Z = Z'y$ where $Z' \subseteq X$ and $y \notin Z'$. Then F is $\forall x G$ for some $G \in \widehat{[\mathcal{P}_{Z'y}]}$. Since $x \notin Z$, we have that $x \notin Z'$ and x does not coincide with y. From the latter it follows that G does not contain y free, and by coherence of $\mathcal{P}$ we have $G \in \widehat{\mathcal{P}_{Z'}}$. Then $F \in [\widehat{\mathsf{gen}_x(\mathcal{P}_{Z'})}]$, hence $F \in \widehat{\mathsf{gen}_x(\mathcal{P}_X)}$.

This concludes the description of the canonical example of operations on proof forms; we will refer to these operations as 'tight cumulative operations.' □

Definition 14 A *generic arithmetical interpretation* of the language FOLP is

- a proof predicate Prf, finite set of variables Y, and operations $\{+, \cdot, !, \mathsf{gen}_x\}$ on proof forms which satisfy Lemma 15 for each $X \subseteq Y$;
- an evaluation $*$ that maps proof variables and constants p to proof forms $\{p_X(X)\}$ and predicate symbols of arity n to arithmetical formulas with n free variables. We suppose that $*$ commutes with renaming of individual variables.

For each X, interpretation $*$ commutes with Prf-operations on proofs, the Boolean connectives, and quantifiers. For proof assertions,

$$(t{:}_X F)^* = Prf(t^*(X), F^*(\underline{X})).$$

Theorem 6 [Soundness Theorem] *If* $\mathsf{FOLP} \vdash F$ *with a constant specification CS, then for every generic arithmetical interpretation* $*$ *respecting CS,* $\mathsf{PA} \vdash F^*$.

Proof. Validity of axioms A1–A3 of FOLP follows immediately from our assumptions concerning proof forms. Provability of B1 is guaranteed by the definition of a proof function. The arithmetical translation of the remaining axioms B2–B5 is provable due to the definition of a proof form. □

Generic proof forms provide yet another semantics of proofs for FOLP hence for HPC. Since each (invariant) parametric evaluation is generic, the generic provability semantics is the strongest, followed by the invariant, and then parametric semantics.

Example 5 Let us check that the explicit Barcan formula

$$\forall x(p{:}_{\{x\}} A(x)) \to \mathsf{gen}_x(p){:}\forall x A(x),$$

in which p is a proof variable and $A(x)$ a unary predicate letter is not valid in generic provability semantics. Fix the set of variables $Y = \{x\}$, the standard proof predicate $Proof$ with 'tight' operations from Lemma 15, and define $A^*(x)$ as $\neg Proof(x, \ulcorner \bot \urcorner)$ which is a

provably Δ_1-formula. By Lemma 13, there is a provably recursive term $g(x)$ such that for each x it returns the code of a proof of $A(x)$. Moreover,

$$\mathsf{PA} \vdash \mathit{Proof}(g(x), A^*(\underline{x})) \to A^*(x). \tag{16}$$

Consider a proof function $g(x)$ with $\widehat{g} = \{A^*(x)\}$, i.e., $A^*(x)$ is the only formula proved by the proof function $g(x)$; this ensures (13). It is a proof function, since (14) also holds. Indeed, argue in PA. If $F(\underline{x}) = A^*(\underline{x})$, then reflexivity for F holds by (16). Otherwise, by (13), $\neg \mathit{Prf}(p^*(x), F(\underline{x}))$, and reflexivity holds again.

Now we fix $Y = \{x\}$ and define a proof form $\mathcal{G}$ as the set of proof functions $g_\emptyset = 0$ and $g_{\{x\}}(x) = g(x)$. We will also need a 0-proof form $\mathcal{Z}$ in which $z_\emptyset = z_{\{x\}} = 0$.

Define p^* to be $\mathcal{G}$, and u^* to be $\mathcal{Z}$ for all other atomic proof terms u, and use tight cumulative operations to obtain a generic interpretation $*$.

We argue that under this interpretation $*$, the explicit Barcan formula is false. Indeed, its antecedent,

$$\forall x \mathit{Proof}(g(x), \neg \mathit{Proof}(\underline{x}, \ulcorner \bot \urcorner))$$

is true, by Gödel's Lemma, whereas its succedent,

$$\mathit{Proof}(f(v)^*, \ulcorner \forall x A^* \urcorner),$$

is false since $\forall x A$ is equivalent to the consistency of PA.

Example 6 Let us check that

$$\neg \forall x A(x) \to \exists x \neg A(x)$$

($A(x)$ is atomic here) is not valid with respect to the generic provability semantics. Its Gödel translation (in an equivalent simplified form $(\cdot)^\circ$, cf. [18], Section 9.2.1) is equivalent to

$$\Box \neg \Box \forall x A(x) \to \exists x \Box \neg \Box A(x). \tag{17}$$

Note that (17) is provable in PA if $\Box$ is interpreted as the **provability** operator '*there exists a proof that ...*.' Indeed, since $\mathsf{PA} \vdash \Box \neg \Box \varphi \to \Box \neg \Box \bot$, the antecedent of (17) implies $\Box \neg \Box \bot$, which, by the formalized Gödel's second incompleteness theorem, is equivalent to $\Box \bot$. In modal logic, $\Box \bot \to \exists x \Box \neg \Box A(x)$, which shows that (17) is provable in PA under the provability understanding of $\Box$. This observation demonstrates that the formal provability reading of modal operators does not conform to intuitionistic logic in terms of Gödel's translation.

Our goal now is to demonstrate that under any normal realization of (17), there is a generic arithmetical interpretation that renders its realization not provable in PA.

A normal realization of (17) has the form

$$u{:}\neg s(u,v){:}\forall x A(x) \to \exists x\ t(u,v){:}_{\{x\}} \neg v{:}_{\{x\}} A(x) \tag{18}$$

for some s and t. Given s and t, we will find an interpretation $*$ such that the antecedent of this formula, $[u{:}\neg s(u,v){:}\forall x A(x)]^*$ is true, but the succedent, $[\exists x\ t(u,v){:}_{\{x\}} \neg v{:}_{\{x\}} A(x)]^*$ is

false in the standard model of PA. This would yield that (18) is not provable in PA under interpretation $*$.

As in Example 5, $A(x)$ will be interpreted as $\neg Proof(x, \ulcorner \bot \urcorner)$. So $Proof(k, \ulcorner \forall x A^* \urcorner)$ is false for each k since $\forall x A^*$ is equivalent to the consistency of PA. Let $g(x)$ be a proof function from Example 5 in which $A^*(x)$ is the only formula proved. Define $v^*(x) = g(x)$, $v^* = 0$, $u^*(x) = u^* = \ulcorner u \urcorner$, and $p^* = p^*(x) = 0$ for all other atomic proof terms p.

Consider the following fixed-point equation that defines a new proof predicate $Prf(x, y)$.

$$Prf(x, y) \;\leftrightarrow\; Proof(x, y) \vee [x = \ulcorner u \urcorner \;\wedge\; y = \ulcorner \neg Prf([s(u, v)]^*, \ulcorner \forall x A^*(x) \urcorner) \urcorner]. \tag{19}$$

For the purposes of the fixed-point equation, what matters is that $[s(u, v)]^*$ can be effectively computed given $*$ and the Gödel number of Prf.

From (19) it immediately follows that $Prf(x, y)$ is provably Δ_1. Moreover, it is also provable from (19) that each $Proof$-proof is a Prf-proof of the same formulas. The only new Prf-proof is $\ulcorner u \urcorner$ and it proves formula $\neg Prf([s(u, v)]^*, \ulcorner \forall x A^*(x) \urcorner)$ which is a true Δ_1-formula and hence also $Proof$-provable. Therefore, Prf and $Proof$ prove the same formulas, and there is an easy computable translation from Prf-proofs to $Proof$-proofs and vice versa. In particular, $v^*(x) = g(x)$ is a proof function for both $Proof$ and Prf.

We can now define proof function f_u as follows. The first component is the constant function $\ulcorner u \urcorner$. The second component is a singleton

$$\{\neg Prf([s(u, v)]^*, \ulcorner \forall x A^*(x) \urcorner)\}.$$

Since $\neg Prf([s(u, v)]^*, \ulcorner \forall x A^*(x) \urcorner)$ is provable, f_u is reflexive and hence a legitimate proof function. Let us set $f_u(x)$ to coincide with f_u (in particular $f_u(x)$ does not really depend on x). Let $\mathcal{U}$ be the proof form $\{f_u, f_u(x)\}$.

Using tight cumulative operations, we obtain a generic interpretation $*$ which interprets v as $\mathcal{G}$, u as $\mathcal{U}$, and all other atomic proof terms p as $\mathcal{Z}$. It remains to observe that under interpretation $*$, the antecedent of (18) is true and the succedent of (18) is false. Indeed, by (19), u^* is a Prf-proof of $\neg Prf([s(u, v)]^*, \ulcorner \forall x A^*(x) \urcorner)$. On the other hand, for each $n \in \omega$, $v^*(n)$ as a $Proof$-proof of $A^*(n)$. Therefore,

$$Proof(v^*(n), \ulcorner A^*(n) \urcorner)$$

holds. Since Prf extends $Proof$, it follows that

$$Prf(v^*(n), \ulcorner A^*(n) \urcorner)$$

is true for each n. Therefore, its negation is never provable, hence $[\exists x\, t(u, v){:}_{\{x\}} \neg v{:}_{\{x\}} A(x)]^*$ is false.

We have shown that $\neg \forall x A(x) \to \exists x \neg A(x)$ fails in the generic semantics of proofs.

8 To what extent is FOLP a BHK-semantics?

We argue that the first-order logic of proofs, in combination with Gödel's translation, conforms to BHK-semantics. First, the proof objects in FOLP have natural provability

interpretations as PA-proofs. Assuming a certain amount of good will from the reader[5], we will try to check that FOLP complies with BHK clauses.

- *A proof of $A \to B$ is a construction which, given a proof of A, returns a proof of B.* An intuitionistic implication $A \to B$ is realized in FOLP as $t{:}(\widetilde{A} \to \widetilde{B})$ where $\widetilde{A}$ and $\widetilde{B}$ are FOLP-versions of A and B respectively. By 'application' axiom B2,

$$t{:}(\widetilde{A} \to \widetilde{B}) \land u{:}\widetilde{A} \to [t{\cdot}u]{:}\widetilde{B},$$

  and t is indeed a construction that, given a proof u of the antecedent of the implication, returns a proof $t{\cdot}u$ of the succedent.

- *A proof of $A \land B$ consists of a proof of A and a proof of B.* An intuitionistic conjunction $A \land B$ is realized in FOLP as $t{:}(\widetilde{A} \land \widetilde{B})$ where $\widetilde{A}$ and $\widetilde{B}$ are, as before, FOLP-versions of A and B. This t contains sufficient information to recover both a proof of $\widetilde{A}$ and a proof of $\widetilde{B}$. Indeed, given such t and commonly known proofs a and b such that $a{:}((\widetilde{A} \land \widetilde{B}) \to \widetilde{A})$ and $b{:}((\widetilde{A} \land \widetilde{B}) \to \widetilde{B})$, one can find a proof of $\widetilde{A}$, $a{\cdot}t$, and a proof of $\widetilde{B}$, $b{\cdot}t$. Likewise, having a proof of $\widetilde{A}$ and a proof of $\widetilde{B}$, one can construct a proof of $\widetilde{A} \land \widetilde{B}$ within FOLP.

- *A proof of $A \lor B$ is given by presenting either a proof of A or a proof of B.* Argue in FOLP. Suppose $u{:}A$ or $u{:}B$. We have to construct a proof term $t(u)$ such that $t(u){:}(A \lor B)$. Consider the internalized disjunction principles $a{:}(A \to (A \lor B))$ and $b{:}(B \to (A \lor B))$, both obviously provable in FOLP. Using application axiom B2, we conclude that either $[a{\cdot}u]{:}(A \lor B)$ or $[b{\cdot}u]{:}(A \lor B)$. In any case,

$$[a{\cdot}u + b{\cdot}u]{:}(A \lor B),$$

  and we can set $t(u)$ to $[a{\cdot}u + b{\cdot}u]$.

- *A proof of $\forall x A(x)$ is a function converting c into a proof of $A(c)$.* An intuitionistic statement $\forall x A(x)$ is represented in FOLP as $u{:}\forall x \widetilde{A}(x)$.[6] By Example 1, we can conclude that

$$\forall x (d{\cdot}u){:}_{\{x\}} \widetilde{A}(x)$$

  with d such that $d{:}(\forall x \widetilde{A}(x) \to \widetilde{A}(x))$. A provability interpretation of the latter produces an arithmetical term (function) $f(x)$ such that

$$\forall x Prf(f(x), \widetilde{A}^*(\underline{x})).$$

  Therefore, a proof term u indeed yields a function $f(x)$ that converts any c into a proof $f(c)$ of $\widetilde{A}(c)$.

[5] which is always necessary when an informal requirement is compared to a formal condition

[6] Here, and in the next item, we suppress parameters other than x for brevity.

- *A proof of $\exists x A(x)$ is a pair (c, d) where d is a proof of $A(c)$.* An intuitionistic statement $\exists x A(x)$ is represented in FOLP as $u{:}\exists x \widetilde{A}(x)$. Since $\widetilde{A}(x)$ is a realization of a modalized FOS4 formula, it is itself a proof assertion $t{:}_{\{x\}}\widetilde{B}(x)$. From axiom B4,

$$t{:}_{\{x\}}B \to !t{:}_{\{x\}}(t{:}_{\{x\}}B).$$

By first-order logic,

$$\exists x t{:}_{\{x\}}B \to \exists x[!t]{:}_{\{x\}}(t{:}_{\{x\}}B).$$

By Internalization, there is a term s such that

$$s{:}(\exists x t{:}_{\{x\}}B \to \exists x[!t]{:}_{\{x\}}(t{:}_{\{x\}}B)),$$

i.e.,

$$s{:}(\exists x \widetilde{A} \to \exists x[!t]{:}_{\{x\}}\widetilde{A}).$$

By the application axiom B1,

$$[s\cdot u]{:}(\exists x[!t]{:}_{\{x\}}\widetilde{A}(x)),$$

which, under any arithmetical interpretation $*$, becomes an assertion that a specific derivation $(s\cdot u)^*$ is a proof of a Σ_1-formula

$$\exists x Prf((!t)^*(x), \widetilde{A}^*(\underline{x})).$$

Such proof assertion yields a specific number n such that

$$Prf((!t)^*(n), \widetilde{A}^*(n)),$$

i.e., that $(!t)^*(n)$ is a proof of $\widetilde{A}^*(n)$ which confirms to the corresponding BHK requirement.

9 Completeness is not attainable

In this section, to simplify formulations without a loss of generality, we consider the languages of LP and FOLP without proof constants and logics LP, FOLP without the axiom necessitation rule. Let PAR, INV, and GEN be sets of FOLP-formulas valid under the parametric, invariant parametric, and generic semantics correspondingly. Negative results from [4] rule out a complete axiom system for GEN. From what we have already learned, it follows that

$$\mathsf{FOLP} \subsetneq \mathsf{GEN} \subsetneq \mathsf{INV} \subsetneq \mathsf{PAR}. \tag{20}$$

We now show that neither PAR nor INV is recursively enumerable, hence neither can be effectively axiomatized.

Theorem 7 *Neither* GEN*,* PAR*, or* INV *is recursively enumerable.*

Proof. For simplicity, we present the proof for the parametric semantics; the same argument works in other cases as well.

Consider the set of FOLP-formulas $\mathcal{F}$ of the form $\neg p{:}F$, where F is a pure first-order formula and p is a proof variable. From the definition of (parametric) arithmetical interpretation, it follows that for each formula F,

$$\neg p{:}\neg F \in \mathsf{PAR} \text{ if and only if for all } *, \ \mathsf{PA} \nvdash \neg F^*.$$

To prove Theorem 7, it remains to demonstrate that the set

$$\mathcal{F} = \{F \mid F \text{ is a first-order formula and for all } *, \ \mathsf{PA} \nvdash \neg F^*\}.$$

is not recursively enumerable.

Let FO be the set of first-order formulas that are valid in all models (this set coincides with first-order logic). Let FIN be the set of formulas that are valid in all finite models and $\overline{\mathsf{FIN}}$ be its complement. It is obvious that $\mathsf{FO} \subset \mathsf{FIN}$, hence $\mathsf{FO} \cap \overline{\mathsf{FIN}} = \emptyset$. The following lemma holds (cf. [6]).

Lemma 16 [6] FO *and* $\overline{\mathsf{FIN}}$ *are recursively inseparable.*

Note that $\mathsf{FO} \subseteq \mathcal{F} \subseteq \mathsf{FIN}$. Indeed, if F is a first-order theorem, then, for each interpretation $*$, $\mathsf{PA} \vdash F^*$, hence $\mathsf{PA} \nvdash \neg F^*$; this secures $\mathsf{FO} \subseteq \mathcal{F}$. Suppose $F \notin \mathsf{FIN}$. Then F is false in some finite model $\mathcal{M}$. Since each finite model can be represented in PA, this yields an arithmetical interpretation $*$ for which $\mathsf{PA} \vdash \neg F^*$, hence $F \notin \mathcal{F}$. We conclude that $\mathcal{F} \subseteq \mathsf{FIN}$.

Therefore $\mathcal{F}$ separates FO and $\overline{\mathsf{FIN}}$ and, by Lemma 16, $\mathcal{F}$ is not decidable. It is easily seen from the definition of $\mathcal{F}$ that its complement $\overline{\mathcal{F}}$ is recursively enumerable. Therefore $\mathcal{F}$ is not recursively enumerable. □

The following corollary could be obtained from (20), but Theorem 7 offers a deeper analysis.

Corollary 6 FOLP *is not complete with respect to any of the aforementioned provability semantics: parametric, invariant parametric, or generic.*

10 Discussion

The arithmetical semantics of the propositional logic of proofs LP corresponds to the generic provability semantics of FOLP with $Y = \emptyset$[7]. This yields the following conservativity result: *for a formula F in the* LP*-language,* $\mathsf{LP} \vdash F$ *iff* $\mathsf{FOLP} \vdash F$. Indeed, $\mathsf{LP} \vdash F$ yields $\mathsf{FOLP} \vdash F$ since LP is subsumed by FOLP. Suppose $\mathsf{LP} \nvdash F$. By the arithmetical completeness theorem for LP (cf. [2]), there is an arithmetical (generic) interpretation $*$ such that F^* is false. By soundness of FOLP, Theorem 6, F is not derivable in FOLP.

[7]In the arithmetical semantics for LP from [2], operations $+$, $\cdot$, and ! are not tight but property (14) holds nevertheless because proof forms are all propositional.

Similar classes of tautologies can be considered for the propositional language: PARp, INVp, and GENp. GENp coincides with LP. INVp is strictly included into PARp (separated by Example 3). It is plausible that INVp also coincides with LP and that this can be established by a proper adaptation of the arithmetical completeness proof from [2].

FOLP may be viewed as a general-purpose justification logic; it opens the door to a general theory of first-order justification in which we anticipate a variety of FOLP-like systems equipped with appropriate epistemic semantics.

11 Acknowledgements

The authors are grateful to Lev Beklemishev, Mel Fitting, Vladimir Krupski, Elena Nogina, Bryan Renne, and Junhua Yu, whose advice helped with this paper. Many thanks to Karen Kletter for editing this text.

References

[1] S. Artemov. *Operational modal logic.* Technical Report MSI 95-29, Cornell University, 1995.

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

[3] S. Artemov. The Logic of Justification. *The Review of Symbolic Logic*, 1(4):477–513, 2008.

[4] S. Artemov and T. Yavorskaya (Sidon). On the first-order logic of proofs. *Moscow Mathematical Journal*, 1:475–490, 2001.

[5] G. Boolos. *The Logic of Provability.* Cambridge University Press, Cambridge, 1993.

[6] Yu. L. Ershov, E. A. Palyutin. *Mathematical logic.* Moskva, Nauka, 1979. In Russian.

[7] M. Fitting. A quantified logic of evidence. *Annals of Pure and Applied Logic*, 152(1–3):67–83, 2008.

[8] M. Fitting. Reasoning with Justifications. *Towards Mathematical Philosophy*, Trends in Logic, 28:107–123, 2009.

[9] K. Gödel. Eine Interpretation des intuitionistischen Aussagenkalkuls. *Ergebnisse Math. Kolloq.*, 4:39–40, 1933. English translation in: S. Feferman et al., editors, *Kurt Gödel Collected Works, Vol. 1*, pages 301–303. Oxford University Press, Oxford, Clarendon Press, New York, 1986.

[10] K. Gödel. Vortrag bei Zilsel, 1938. In S. Feferman, editor, *Kurt Gödel Collected Works. Volume III*, pages 86–113. Oxford University Press, 1995.

[11] A. Kolmogoroff. Zur Deutung der intuitionistischen Logik. *Mathematische Zeitschrift,* 35:5865, 1932. English translation in *Selected works of A.N. Kolmogorov. Volume I: Mathematics and Mechanics*, (V.M. Tikhomirov, editor), Kluwer, 1985.

[12] V.N. Krupski. Operational logic of proofs with functionality condition on proof predicate. In S. Adian and A. Nerode, editors, *Logical Foundations of Computer Science '97, Yaroslavl'*, volume 1234 of *Lecture Notes in Computer Science*, pages 167–177. Springer, 1997.

[13] V.N. Krupski. The single-conclusion proof logic and inference rules specification. *Annals of Pure and Applied Logic*, 113(1-3):181–206, 2001.

[14] J.C.C. McKinsey and A. Tarski. Some theorems about the sentential calculi of Lewis and Heyting. *The Journal of Symbolic Logic*, 13:1–15, 1948.

[15] P. S. Novikov. *Constructive mathematical logic from the viewpoint of the classical one.* Nauka, Moscow, 1977, (in Russian).

[16] T.L. Sidon. Non-axiomatizability of the predicate logics of proofs. *Vestnik Moskov. Univ. Ser. 1 Mat., Mekh.*, (6):18–22, 1998. In Russian. English translation in: *Moscow University Mathematics Bulletin*, v. 53, n. 6, pp. 17–21, 1998.

[17] C. Smoryński. *Self-Reference and Modal Logic.* Springer-Verlag, Berlin, 1985.

[18] A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory.* Cambridge University Press, Amsterdam, 1996.

[19] A.S. Troelstra and D. van Dalen. *Constructivism in mathematics. An introduction, vol. 1.* North-Holland, Amsterdam, 1988.

[20] D. van Dalen. *Intuitionistic logic.* In D. Gabbay and F. Guenther, editors. *Handbook of Philosophical Logic. vol. 3.* Dordrecht, The Netherlands: Reidelpp. 225–340, 1988.

[21] R. Yavorsky. On Arithmetical Completeness of First-Order Logics of Provability. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logic. Volume 3*, pages 1–16. CSLI Publications, Stanford University, 2003.