Theory Comput Syst (2008) 43: 272–293
DOI 10.1007/s00224-007-9057-y

Interacting Explicit Evidence Systems

Tatiana Yavorskaya (Sidon)

Published online: 25 October 2007

Abstract Logic of proofs LP, introduced by S. Artemov, originally designed for describing properties of formal proofs, now became a basis for the theory of knowledge with justification (cf. S. Artemov, *Evidence-based common knowledge*, Technical report TR–2004018, CUNY Ph.D. Program in Computer Science, 2005). So far, in epistemic systems with justification the corresponding "evidence part", even for multi-agent systems, consisted of a single explicit evidence logic. In this paper we introduce logics describing two interacting explicit evidence systems. We find an appropriate formalization of the intended semantics and prove the completeness of these logics with respect to both symbolic and arithmetical models. Also, we find the forgetful projections for the two-agent justification logics which are extensions of the bimodal logic $\mathsf{S4}^2$.

Keywords Justification logic · Explicit evidence · Logic of proofs · Multi-modal logic · Epistemic logic

1 Introduction

The Logic of Proofs LP introduced by S. Artemov in 1995 (see the detailed description in [1, 2]) was originally designed to express in logic the notion of a proof. It is formulated in the propositional language enriched by new atoms $[\![t]\!]F$ with the intended meaning *"t is a proof of F"*. Proofs are represented by *proof terms* constructed from *proof variables* and *proof constants* by means of three elementary computable

T. Yavorskaya (Sidon) (✉)
Department of Mathematical Logic and Theory of Algorithms, Faculty of Mechanics and Mathematics, Moscow State University, Moscow, 119992, Russia
e-mail: tanya@lpcs.math.msu.su

operations: binary $\cdot$, $+$ and unary ! specified by the axioms:

$$\begin{array}{lll} [\![t]\!](A \to B) \to ([\![s]\!]A \to [\![t \cdot s]\!]B) & & \textit{application}, \\ [\![t]\!]A \to [\![t + s]\!]A, & [\![s]\!]A \to [\![t + s]\!]A & \textit{nondeterministic choice}, \\ [\![t]\!]A \to [\![!t]\!][\![t]\!]A & & \textit{positive checker}. \end{array}$$

LP is axiomatized over propositional calculus by the above axioms and the principle

$$[\![t]\!]A \to A \quad \textit{weak reflexivity}.$$

The rules of inference are *modus ponens* and *axiom necessitation rule*. The latter allows to specify proof constants as proofs of the concrete axioms

$$\frac{}{[\![a]\!]\mathsf{A}}, \quad \text{where } a \text{ is an axiom constant, A is an axiom of LP.}$$

The intended semantics for LP is given by formal proofs in Peano Arithnmetic PA: proof variables are interpreted by codes of PA-derivations, $[\![t]\!]F$ stands for the arithmetical proof predicate "t is a code of a derivation proving F". It is proven in [2] that LP is arithmetically complete with respect to the class of all proof systems. Furthermore, LP suffices to realize Gödel's provability logic S4 and thus provides S4 and intuitionistic logic with the exact provability semantics.

In [3] it was suggested to treat $[\![t]\!]F$ as a new type of knowledge operator called *evidence-based knowledge* with the meaning "t is an evidence (justification) for F." Evidence based knowledge (*EBK*) systems are obtained by augmenting a multi–agent logic of knowledge with a system of evidence assertions $[\![t]\!]F$. Three main cases of *EBK*-systems were introduced in [3] in which the base knowledge logic is T_n, $\mathsf{S4}_n$ or $\mathsf{S5}_n$. The evidence part for all of them consists of a single logic of proofs LP.

In this paper we study multiple interacting *EBK*-systems, namely, we study logics that describe the behavior of two reasoning agents $\mathcal{P}_1$ and $\mathcal{P}_2$ which somehow communicate to each other. We develop a language with two types of evidence assertions: $[\![t]\!]_1 A$ with the meaning *"$\mathcal{P}_1$ accepts t as an evidence for A"* and $[\![s]\!]_2 B$ with the meaning *"$\mathcal{P}_2$ accepts s as an evidence for B"* In general, evidences accepted by these two systems are distinct, so evidence terms for each $\mathcal{P}_i$ $(i = 1, 2)$ are constructed from its own atomic evidences represented by variables p_k^i and constants c_k^i. We suppose that both $\mathcal{P}_1$ and $\mathcal{P}_2$ has all the power of LP, so we reserve a copy of LP-operations $\times_i$, $+_i$ and $!_i$ for application, nondeterministic choice and positive checker in $\mathcal{P}_i$ $(i = 1, 2)$.

For the minimal two-agent justification logic denoted by LP^2 we assume that there is no communication between the agents, except that all axioms are common knowledge, so we extend the axiom necessitation rule and allow it to derive all the formulas

$$[\![c_{j_1}^{k_1}]\!]_{k_1}[\![c_{j_2}^{k_2}]\!]_{k_2} \ldots [\![c_{j_n}^{k_n}]\!]_{k_n} A, \quad \text{where all } k_i \in \{1, 2\},\ A \text{ is an axiom.}$$

Going further, we may assume that the two systems $\mathcal{P}_1$ and $\mathcal{P}_2$ are allowed to communicate, that is, one of these systems is able to derive something about the other. We study two types of communications.

Evidence verification We assume that $\mathcal{P}_2$ can verify all evidences of $\mathcal{P}_1$. To express that, we introduce a unary operation $!_1^2$ specified by the axiom

$$[\![t]\!]_1 A \to [\![!_1^2 t]\!]_2 [\![t]\!]_1 A.$$

Further, we can consider the case when both $\mathcal{P}_1$ and $\mathcal{P}_2$ are able to verify each other, then we add the dual operation $!_2^1$ with the specification

$$[\![t]\!]_2 A \to [\![!_2^1 t]\!]_1 [\![t]\!]_2 A.$$

The resulting logics are denoted by $\mathsf{LP}_!^2$ and $\mathsf{LP}_{!!}^2$ respectively.

Evidence conversion Here we suppose that all evidences accepted by $\mathcal{P}_1$ can be converted to $\mathcal{P}_2$-evidences; this is done by the operation $\uparrow_1^2$ specified by the principle

$$[\![t]\!]_1 A \to [\![\uparrow_1^2 t]\!]_2 A.$$

If $\mathcal{P}_1$ can also adopt $\mathcal{P}_2$–evidences, we add a converse operation $\uparrow_2^1$ with the specification

$$[\![t]\!]_2 A \to [\![\uparrow_2^1 t]\!]_1 A.$$

We denote the resulting logics by $\mathsf{LP}_{\uparrow}^2$ and $\mathsf{LP}_{\uparrow\uparrow}^2$.

In this paper for all the logics L mentioned above we do the following:

- find the forgetful projection of L, i.e. a bimodal logic obtained from L by replacing all occurrences of $[\![t]\!]_i$ by $\Box_i$ for $i = 1, 2$;
- describe symbolic semantics and prove completeness of L;
- describe arithmetical interpretation and prove completeness of L.

The structure of the paper is the following. In Sect. 2 we give a precise description of the language and the logics we are dealing with. In Sect. 3 the modal counterparts of all the described logics are found. It turned out that the forgetful projections of $\mathsf{LP}_!^2$ and $\mathsf{LP}_{!!}^2$ coincide with the projections of $\mathsf{LP}_{\uparrow}^2$ and $\mathsf{LP}_{\uparrow\uparrow}^2$ respectively. Sections 4 and 5 are devoted to symbolic and arithmetical semantics.

2 Two-Agent Explicit Evidence Logics: Definitions

Definition 2.1 The minimal language L of the bimodal explicit evidence logic is denoted by LP^2. It contains:

- propositional variables $SVar = \{S_1, S_2, \ldots\}$;
- two disjoint sets of variables $PVar^i = \{p_1^i, p_2^i, \ldots\}$ and two disjoint sets of constants $\{c_1^i, c_2^i, \ldots\}$ where $i = 1, 2$;
- two copies of every operation from LP: binary $\times_1, +_1, \times_2, +_2$ and unary $!_1$ and $!_2$;
- Boolean connectives and two operational symbols $[\![\cdot]\!]_1(\cdot)$ and $[\![\cdot]\!]_2(\cdot)$ of the type evidence $\to$ (proposition $\to$ proposition)

We also consider extensions of L. The first option is to add one or both of the unary functional symbols $!_1^2$ and $!_2^1$; we denote the result by $\mathsf{LP}_!^2$, $\mathsf{LP}_{!!}^2$ respectively. Another option is to add one or both of the unary functional symbols $\uparrow_1^2$ and $\uparrow_2^1$; the result is denoted by $\mathsf{LP}_\uparrow^2$, $\mathsf{LP}_{\uparrow\uparrow}^2$ respectively.

For every language L from the definition above and for $i = 1, 2$ we define the set of terms $Tm_i(L)$. For $L = \mathsf{LP}^2$ the set $Tm_i(L)$ consists of all terms constructed from variables and constants labelled with sup-i by operations labelled by sub-i. Namely, for $i = 1, 2$, every variable p_j^i or constant c_j^i is an element of $Tm_i(L)$ and if $t, s \in Tm_i(L)$, then $t \times_i s$, $t +_i s$ and $!_i t$ belong to $Tm_i(L)$ too. For the extensions of the minimal language we add the following clauses to the definition of terms:

- for $L = \mathsf{LP}_!^2$, if $t \in Tm_1(L)$ then $!_1^2 t \in Tm_2(L)$;
- for $L = \mathsf{LP}_{!!}^2$, if $t \in Tm_1(L)$ then $!_1^2 t \in Tm_2(L)$ and if $t \in Tm_2(L)$ then $!_2^1 t \in Tm_1(L)$;
- for $L = \mathsf{LP}_\uparrow^2$, if $t \in Tm_1(L)$ then $\uparrow_1^2 t \in Tm_2(L)$;
- for $L = \mathsf{LP}_{\uparrow\uparrow}^2$, if $t \in Tm_1(L)$ then $\uparrow_1^2 t \in Tm_2(L)$ and if $t \in Tm_2(L)$ then $\uparrow_2^1 t \in Tm_1(L)$.

Formulas of the language L are constructed from sentence variables by boolean connectives and according to the rule: for $i = 1, 2$ if $t \in Tm_i(L)$ and F is a formula of L then $[\![t]\!]_i F$ is a formula of L too. The set of all formulas is denoted by $Fm(L)$. Formulas of the form $[\![t]\!]_i F$ are called *q-atomic*, the set of such formulas is denoted by $QFm_i(L)$. We write $QFm(L)$ for $QFm_1(L) \cup QFm_2(L)$.

Definition 2.2 Operations on proofs are specified by the following schemas (t and s are terms, A and B are formulas):

$$
\begin{array}{ll}
\mathrm{Ax}(\times_i) & [\![t]\!]_i(A \to B) \to ([\![s]\!]_i A \to [\![t \times_i s]\!]_i B), \\
\mathrm{Ax}(+_i) & [\![t]\!]_i A \to [\![t +_i s]\!]_i A, \qquad [\![s]\!]_i A \to [\![t +_i s]\!]_i A, \\
\mathrm{Ax}(!_i) & [\![t]\!]_i A \to [\![!_i t]\!]_i [\![t]\!]_i A, \\
\mathrm{Ax}(!_1^2) & [\![t]\!]_1 A \to [\![!_1^2 t]\!]_2 [\![t]\!]_1 A, \\
\mathrm{Ax}(!_2^1) & [\![t]\!]_2 A \to [\![!_2^1 t]\!]_1 [\![t]\!]_2 A, \\
\mathrm{Ax}(\uparrow_1^2) & [\![t]\!]_1 A \to [\![\uparrow_1^2 t]\!]_2 A, \\
\mathrm{Ax}(\uparrow_2^1) & [\![t]\!]_2 A \to [\![\uparrow_2^1 t]\!]_1 A.
\end{array}
$$

Definition 2.3 For every language L from Definition 2.1 we define the corresponding bimodal justification logic L. It is axiomatized by the following schemas:

$$
\begin{array}{ll}
\mathrm{A0} & \textit{classical propositional axioms}, \\
\mathrm{A1} & [\![t]\!]_i A \to A, \quad i = 1, 2, \\
\mathrm{A2}\ldots & \textit{axioms for all operations of } L.
\end{array}
$$

The rules of inference are *modus ponens* and axiom necessitation rule

$$
[\![c_{j_1}^{k_1}]\!]_{k_1} [\![c_{j_2}^{k_2}]\!]_{k_2} \ldots [\![c_{j_n}^{k_n}]\!]_{k_n} A, \qquad \text{where all } k_i \in \{1, 2\},\ A \text{ is an axiom of } L.
$$

Informally speaking, the language LP^2 describes the structure consisting of objects of three types: *propositions* represented by formulas and two types of evidences, *evidences*$_1$ and *evidences*$_2$, represented by terms. We suppose that there are two intelligent systems $\mathcal{P}_1$ and $\mathcal{P}_2$; the system $\mathcal{P}_i$ can accept $t \in \mathit{evidences}_i$ as a justification for a proposition $A \in \mathit{propositions}$. We will call $\mathcal{P}_i$ systems with justification or merely *justification systems*.

The structure is supplied with two predicates, $[\![t]\!]_1 A$ and $[\![t]\!]_2 A$, where $[\![t]\!]_i A$ means that $\mathcal{P}_i$ accepts t as an evidence for A. The above predicates are supposed to be recursive. For every $p \in \mathit{evidences}_i$ the set of propositions justified by p in $\mathcal{P}_i$ is finite and the function that maps evidences to the corresponding sets is total recursive.

In the provability context, one may think about $\mathcal{P}_1$ and $\mathcal{P}_2$ as two deductive systems; then evidences are proofs (derivations) in these systems and formulas $[\![t]\!]_i A$ denote a proof–theorem relation for the system $\mathcal{P}_i$, namely, $[\![t]\!]_i A$ means that t is a (code of a) derivation of A in the system $\mathcal{P}_i$. One may take for $\mathcal{P}_1$ and $\mathcal{P}_2$ two different style deductive systems for the same theory, say, Hilbert-style and Gentsen-style derivations. Another option is to take for $\mathcal{P}_1$ and $\mathcal{P}_2$ two theories in the same language which are not equally strong, like Peano Arithmetic and $I\Sigma_1$ and so on. Section 5 is devoted to the arithmetical reading of our logics.

In this paper we consider the case when both $\mathcal{P}_1$ and $\mathcal{P}_2$ are supplied with operations on evidences taken from LP, thus, they are capable of internalizing there own derivations. The minimal language LP^2 corresponds to the situation when two justification systems do not communicate. The only information about $\mathcal{P}_1$ which is available to $\mathcal{P}_2$ and vice versa, is transferred via the axiom necessitation rule. For example, $\mathcal{P}_2$ knows that $!_1$ checks evidences in $\mathcal{P}_1$ since we can derive $[\![c^2]\!]_2([\![t]\!]_1 A \to [\![!t]\!]_1[\![t]\!]_1 A)$. Externally we can prove that something is provable in $\mathcal{P}_1$ iff it is provable in $\mathcal{P}_2$, that is, the following two assertions are equivalent:

(1) there exists a term $t \in Tm_1(\mathsf{LP}^2)$ such that $\mathsf{LP}^2 \vdash [\![t]\!]_1 A$, and

(2) there exists a term $s \in Tm_2(\mathsf{LP}^2)$ such that $\mathsf{LP}^2 \vdash [\![s]\!]_2 A$.

This fact immediately follows from lemma 2.6. However, it cannot be derived in LP^2, that is, there is no term $t \in Tm_2(\mathsf{LP}^2)$ such that $\mathsf{LP}^2 \vdash [\![p^1]\!]_1 S \to [\![t]\!]_2 S$ (this last fact can be easily proven using symbolic semantics from Sect. 4). So, neither $\mathcal{P}_1$ nor $\mathcal{P}_2$ is able to formalize or prove the equivalence just mentioned.

The communication between the two justification systems becomes possible in the extensions of LP^2. In $\mathsf{LP}^2_!$ and $\mathsf{LP}^2_\uparrow$ it is one-way: $\mathcal{P}_2$ can derive some facts about $\mathcal{P}_1$. In $\mathsf{LP}^2_{!!}$ in $\mathsf{LP}^2_{\uparrow\uparrow}$ information can be transferred symmetrically both-ways. Operations $!^2_1$ and $!^1_2$ are checkers. $\mathsf{LP}^2_!$ corresponds to the case when $\mathcal{P}_2$ is able to check evidences of $\mathcal{P}_1$; in $\mathsf{LP}^2_{!!}$ we suppose that both of $\mathcal{P}_i$ can check each other. Operations $\uparrow^2_1$ or $\uparrow^1_2$ appear if one of the systems can adopt evidences of another one.

Operations $!^2_1$ and $\uparrow^2_1$ can imitate each other in the following sense.

Lemma 2.4

1. *For every term* $t \in Tm_1(\mathsf{LP}^2_!)$ *and formula* $F \in Fm(\mathsf{LP}^2_!)$, *there is a term* $s \in Tm_2(\mathsf{LP}^2_!)$ *such that* $\mathsf{LP}^2_! \vdash [\![t]\!]_1 F \to [\![s]\!]_2 F$.

2. *For every term* $t \in Tm_1(\mathsf{LP}^2_{\uparrow})$ *and formula* $F \in Fm(\mathsf{LP}^2_{\uparrow})$, *there is a term* $s \in Tm_2(\mathsf{LP}^2_{\uparrow})$ *such that* $\mathsf{LP}^2_{\uparrow} \vdash [\![t]\!]_1 F \to [\![s]\!]_2[\![t]\!]_1 F$.

Proof To prove part 1, derive in $\mathsf{LP}^2_!$:

(1) $[\![t]\!]_1 F \to [\![!^2_1 t]\!]_2[\![t]\!]_1 F$, *axiom* $\mathrm{Ax}(!^2_1)$,
(2) $[\![t]\!]_1 F \to F$, *axiom* A1,
(3) $[\![c^2]\!]_2([\![t]\!]_1 F \to F)$, *by axiom necessitation*,
(4) $[\![c^2]\!]_2([\![t]\!]_1 F \to F) \to ([\![!^2_1 t]\!]_2[\![t]\!]_1 F \to [\![c^2 \times_2 (!^2_1 t)]\!]_2 F)$, *axiom* $\mathrm{Ax}(\times_2)$,
(5) $[\![t]\!]_1 F \to [\![c^2 \times_2 (!^2_1 t)]\!]_2 F$, *by propositional logic from* (4), (3) *and* (1).

Take $s = c^2 \times_2 (!^2_1 t)$. To prove part 2, derive in $\mathsf{LP}^2_{\uparrow}$:

(1) $[\![t]\!]_1 F \to [\![!_1 t]\!]_1[\![t]\!]_1 F$ *axiom* $\mathrm{Ax}(!_1)$
(2) $[\![!_1 t]\!]_1[\![t]\!]_1 F \to [\![\uparrow^2_1 (!_1 t)]\!]_2[\![t]\!]_1 F$ *axiom* $\mathrm{Ax}(\uparrow^2_1)$
(3) $[\![t]\!]_1 F \to [\![\uparrow^2_1 (!_1 t)]\!]_2[\![t]\!]_1 F$ *by propositional logic*

Take $s = \uparrow^2_1 (!_1 t)$. □

Note that the ability of $!^2_1$ to imitate $\uparrow^2_1$ is due to the fact that $\mathcal{P}_2$ has an evidence for reflexivity of $\mathcal{P}_1$. On the other side, $\uparrow^2_1$ is able to simulate $!^2_1$ because $\mathcal{P}_1$ can proof-check itself. So, if one of the proof systems $\mathcal{P}_i$ or both of them would be weaker than LP, then operations $!^2_1$ and $\uparrow^2_1$ would be different (possibly incomparable).

One can easily prove that all the logics from Definition 2.3 enjoy the standard deduction and substitution theorems. For a formula A, by $A[B/S]$ we denote the result of substitution a formula B for a sentence variable S in A. Similarly, by $A[t/p^i]$ we denote the result of substitution a term $t \in Tm_i(L)$ for an evidence variable p^i in A.

Lemma 2.5 *Let* L *be any logic from Definition* 2.3.

1. Substitution theorem *If* $L \vdash A$ *then for all formulas* B *and terms* $t \in Tm_i(L)$, $L \vdash A[B/S]$ *and* $L \vdash A[t/p^i]$.
2. Deduction theorem *If* $\Gamma, A \vdash B$ *in* L, *then* $\Gamma \vdash A \to B$ *in* L.

The lemma below says that any logic defined above is capable to internalize its own derivations as evidence terms.

Lemma 2.6 (Internalization property) *Let* L *be one of the logics from Definition* 2.3. *If* $L \vdash F$, *then for* $i = 1, 2$ *there exists a term* t_i *constructed from constants with the help of operations* $\times_i$ *and* $!_i$ *such that* $L \vdash [\![t_i]\!]_i F$.

Proof Induction on derivation of F. If F is an axiom then apply the axiom necessitation rule to derive $[\![c^i]\!]_i F$ for any constant c^i. If F is obtained by modus ponens from G and $G \to F$ then by the induction hypothesis L derives $[\![u]\!]_i G$ and $[\![v]\!]_i (G \to F)$ for some terms $u, v \in Tm_i(L)$. Then by $\mathrm{Ax}(\times_i)$ we derive

$L \vdash [\![v \times_i u]\!]F$. Finally, if F is obtained by the axiom necessitation rule, then it has the form $[\![c_{l_1}^{k_1}]\!]_{k_1} \ldots [\![c_{l_n}^{k_n}]\!]_{k_n} A$ where A is an axiom of L. We pick a fresh constant c^i and derive $[\![c^i]\!]_i [\![c_{l_1}^{k_1}]\!]_{k_1} \ldots [\![c_{l_n}^{k_n}]\!]_{k_n} A$ by the same rule. □

Internalization lemma is the property we would like all justification systems to have. In particular, this is the reason why we choose the form of axiom necessitation rule stronger than in LP. In LP^2, if we replace the axiom necessitation rule by its standard form ($\vdash [\![c^i]\!]_i A$, where A is an axiom), this does not allow to derive $[\![t]\!]_2 [\![c^1]\!]_1 A$ for any term $t \in Tm_2(\mathsf{LP}^2)$. However, the lemma below shows that in order to preserve internalization it is sufficient to take more modest version of the rule.

Lemma 2.7 *Internalization property remains true in the following cases*:

1. *If in any of the logics from Definition 2.3 the axiom necessitation rule is replaced by a weaker version*: $\vdash [\![c_{i_1}^1]\!]_1 [\![c_{i_2}^2]\!]_2 [\![c_{i_3}^1]\!]_1 \ldots A$, *A is an axiom.*
2. *If in* $\mathsf{LP}^2_{!!}$ *or* $\mathsf{LP}^2_{\uparrow\uparrow}$ *the of the axiom necessitation rule is replaced by its standard version*: $\vdash [\![c^i]\!]_i A$, *A is an axiom*, $i = 1, 2$.

Proof Similarly with Lemma 2.6, we prove that if $L \vdash F$, then for $i = 1, 2$ there exists a term t_i such that $L \vdash [\![t_i]\!]_i F$ by induction on the derivation of F. Let us consider the only different case, when F is obtained by the axiom necessitation rule.

For part 1, F has the form $[\![c_{i_1}^1]\!]_1 [\![c_{i_2}^2]\!]_2 [\![c_{i_3}^1]\!]_1 \ldots A$, A is an axiom. For $i = 1$ apply $\mathrm{Ax}(!_1)$ to obtain $L \vdash [\![!_1 c_{i_1}^1]\!]_1 F$. For $i = 2$ take some constants c^1 and c^2 and by axiom necessitation rule derive $L \vdash [\![c^1]\!]_1 [\![c^2]\!]_2 F$, hence $L \vdash [\![c^2]\!]_2 F$ by the reflexivity axiom A1.

For part 2, F has the form $[\![c_n^j]\!]_j A$, where A is an axiom, and $j = 1, 2$. Suppose that $j = 1$ (the case $j = 2$ is symmetrical). If $i = 1$, then apply $\mathrm{Ax}(!_1)$ to obtain $L \vdash [\![!_1 c_j^1]\!]_1 F$. If $i = 2$, then $\mathsf{LP}^2_{!!} \vdash [\![!_1^2 c_j^1]\!]_2 F$ and $\mathsf{LP}^2_{\uparrow\uparrow} \vdash [\![\uparrow_1^2 (!_1 c_j^1)]\!]_2 F$. □

Lemma 2.8 *Let L be one of the logics from Definition 2.3. For* $i = 1, 2$ *let* δ_i *be a* $\wedge,\vee$*-combination of q-atoms from* $QFm_i(L)$. *Let* δ *stand for a* $\wedge,\vee$*-combination of q-atoms from* $QFm_1(L) \cup QFm_2(L)$:

1. *There exists a term* t_i *such that* $L \vdash \delta_i \to [\![t_i]\!]_i \delta_i$.
2. *If L contains either* $!_1^2$ *or* $\uparrow_1^2$, *then there exists a term* $t \in Tm_2(L)$ *such that* $L \vdash \delta \to [\![t]\!]_2 \delta$.
3. *If L contains either* $!_2^1$ *or* $\uparrow_2^1$, *then there exists a term* $t \in Tm_1(L)$ *such that* $L \vdash \delta \to [\![t]\!]_1 \delta$.

Proof 1. Induction on the construction of δ_i. If $\delta_i = [\![u]\!]_i F$, then apply $\mathrm{Ax}(!_i)$ to obtain $L \vdash \delta_i \to [\![!_i u]\!]_i \delta_i$. If $\delta_i = \alpha_i \wedge \beta_i$ or $\delta_i = \alpha_i \vee \beta_i$ then, by the induction hypothesis, there exist terms u and v such that $L \vdash \alpha_i \to [\![u]\!]_i \alpha_i$ and $L \vdash \beta_i \to [\![v]\!]_i \beta_i$. By the axiom necessitation rule, $L \vdash [\![c^i]\!]_i (\alpha_i \to (\beta_i \to (\alpha_i \wedge \beta_i))$. Using $\mathrm{Ax}(\times_i)$, we derive

$$L \vdash \alpha_i \wedge \beta_i \to [\![c^i \times_i u \times_i v]\!]_i (\alpha_i \wedge \beta_i).$$

By axiom necessitation we also have $L \vdash [\![c_1^i]\!]_i(\alpha_i \to \alpha_i \vee \beta_i)$ and $L \vdash [\![c_2^i]\!]_i(\beta_i \to \alpha_i \vee \beta_i)$. Hence $L \vdash \alpha_i \to [\![c_1^i \times_i u]\!]_i(\alpha_i \vee \beta_i)$ and $L \vdash \beta_i \to [\![c_2^i \times_i v]\!]_i(\alpha_i \vee \beta_i)$. Therefore $L \vdash \alpha_i \vee \beta_i \to [\![(c_1^i \times_i u) +_i (c_2^i \times_i v)]\!]_i(\alpha_i \vee \beta_i)$.

2. The induction step is similar to the previous case. For the induction base we consider two options when δ is either $[\![u]\!]_1 F$ or $[\![v]\!]_2 F$. The second option is treated similarly with the previous case. For $\delta = [\![u]\!]_1 F$ we have $\mathsf{LP}_!^2 \vdash [\![u]\!]_1 F \to [\![!_1^2 u]\!]_2 [\![u]\!]_1 F$. In $\mathsf{LP}_\uparrow^2$ we reason as follows: $\mathsf{LP}_\uparrow^2 \vdash [\![u]\!]_1 F \to [\![!_1 u]\!]_1 [\![u]\!]_1 F$, that is, $\mathsf{LP}_\uparrow^2 \vdash \delta \to [\![!_1 u]\!]_1 \delta$ and $\mathsf{LP}_\uparrow^2 \vdash [\![!_1 u]\!]_1 \delta \to [\![\uparrow_1^2 (!_1 u)]\!]_2 \delta$. Therefore, $\mathsf{LP}_\uparrow^2 \vdash \delta \to [\![\uparrow_1^2 (!_1 u)]\!]_2 \delta$ and $\uparrow_1^2 (!_1 u)$ is the desired t.

3. Similar to part 2. □

3 Realization of Bimodal Logics

In [2] it is proven that LP is able to realize all derivations in the modal logic S4, namely, if A is a theorem of S4 then there is an assignment of LP–terms to all occurrences of $\Box$'s in A such that the resulting formula is a theorem in LP. In this section we describe the modal counterparts of the logics LP^2, $\mathsf{LP}_!^2$ and $\mathsf{LP}_\uparrow^2$.

Let us describe the bimodal logic $\mathsf{S4}^2$ and its extensions $\mathsf{S4}^2_{\mathsf{mon}}$ and $\mathsf{S4}^2_{\mathsf{triv}}$. $\mathsf{S4}^2$ is given by the following axioms and rules of inference, for $i = 1, 2$:

A1 propositional tautologies

A2 $\Box_i A \to A$

A3 $\Box_i (A \to B) \to (\Box_i A \to \Box_i B)$

A4 $\Box_i A \to \Box_i \Box_i A$

R1 Modus ponens: $A, A \to B \vdash B$

R2 Necessitation: if $\vdash A$ then $\vdash \Box_i A$.

$\mathsf{S4}^2_{\mathsf{mon}}$ is an extension of $\mathsf{S4}^2$ by the principle

A5 $\Box_1 F \to \Box_2 F$.

$\mathsf{S4}^2_{\mathsf{triv}}$ is obtained by adding to $\mathsf{S4}^2_{\mathsf{mon}}$ the dual principle

A6 $\Box_2 F \to \Box_1 F$.

We prove that the analog of the realization theorem for S4 and LP holds for the following pairs of logics: $\mathsf{S4}^2$ and LP^2, $\mathsf{S4}^2_{\mathsf{mon}}$ and $\mathsf{LP}_!^2$, $\mathsf{S4}^2_{\mathsf{mon}}$ and $\mathsf{LP}_\uparrow^2$, $\mathsf{LP}_{!!}^2$ and $\mathsf{S4}^2_{\mathsf{triv}}$, $\mathsf{LP}_{\uparrow\uparrow}^2$ and $\mathsf{S4}^2_{\mathsf{triv}}$. We need the following definition.

Definition 3.1 Let L be one of the languages from Definition 2.1. Suppose that A is a formula with two modalities. A realization of A in the language L is a formula $A^r \in Fm(L)$ which is obtained from A by substitution of terms from $Tm_i(L)$ for all occurrences of $\Box_i$ in A, $i = 1, 2$. A realization is normal if all negative occurrences of modalities are assigned proof variables.

Theorem 3.2

1. $\mathsf{S4}^2 \vdash A$ *iff there exists a normal realization* A^r *in the language* LP^2 *such that* $\mathsf{LP}^2 \vdash A^r$.
2. *For* $L \in \{\mathsf{LP}^2_!, \mathsf{LP}^2_\uparrow\}$, $\mathsf{S4}^2_{\mathsf{mon}} \vdash A$ *iff there exists a normal realization* A^r *in the language* L *such that* $L \vdash A^r$.
3. *For* $L \in \{\mathsf{LP}^2_{!!}, \mathsf{LP}^2_{\uparrow\uparrow}\}$, $\mathsf{S4}^2_{\mathsf{triv}} \vdash A$ *iff there exists a normal realization* A^r *in the language* L *such that* $L \vdash A^r$.

The proof of this theorem goes along the lines of the proof of realization of S4 in LP (cf. [2]).

Step 1. First of all, we need the cut-free Gentzen-style versions of $\mathsf{S4}^2$, $\mathsf{S4}^2_{\mathsf{mon}}$ and $\mathsf{S4}^2_{\mathsf{triv}}$.

For the sake of briefness, let us consider bimodal logics in the language with two modalities $\Box_1$ and $\Box_2$ and $\bot$, $\to$ for propositional logic. A sequent has the form $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite multi-sets of formulas. The standard translation of this sequent is the formula $\wedge\Gamma \to \vee\Delta$, where by $\wedge\Gamma$ and $\vee\Delta$ we denote the conjunction of all formulas from Γ and the disjunction of all formulas from Δ respectively. Positive and negative occurrences of subformulas in a sequent are defined in accordance with its translation.

Sequential calculus for $\mathsf{S4}^2$, denoted by $\mathsf{GS4}^2$, has the same axioms and rules as sequential calculus for classical propositional logic (see [9]) plus four modal rules (two for each modality):

$$(\text{Left}\Box_i)\quad \frac{A, \Gamma \Rightarrow \Delta}{\Box_i A, \Gamma \Rightarrow \Delta} \qquad (\text{Right}\Box_i)\quad \frac{\Box_i \Gamma \Rightarrow A}{\Box_i \Gamma \Rightarrow \Box_i A} \quad (i = 1, 2).$$

In the Gentzen-style version of $\mathsf{S4}^2_{\mathsf{mon}}$ denoted by $\mathsf{GS4}^2_{\mathsf{mon}}$ the rule (Right$\Box_2$) is replaced by a stronger version

$$\frac{\Box_1\Gamma_1, \Box_2\Gamma_2 \Rightarrow A}{\Box_1\Gamma_1, \Box_2\Gamma_2 \Rightarrow \Box_2 A}.$$

In the Gentzen-style version of $\mathsf{S4}^2_{\mathsf{triv}}$ denoted by $\mathsf{GS4}^2_{\mathsf{triv}}$ we also replace the rule (Right$\Box_1$) by

$$\frac{\Box_1\Gamma_1, \Box_2\Gamma_2 \Rightarrow A}{\Box_1\Gamma_1, \Box_2\Gamma_2 \Rightarrow \Box_1 A}.$$

Theorem 3.3 *For any logic* $L \in \{\mathsf{S4}^2, \mathsf{S4}^2_{\mathsf{mon}}, \mathsf{S4}^2_{\mathsf{triv}}\}$ *the following connection between* L *and its Gentzen-style version* $\mathcal{G}$ *holds*:

$$\mathcal{G} \vdash \Gamma \Rightarrow \Delta \quad \textit{iff} \quad L \vdash \wedge\Gamma \to \vee\Delta.$$

Proof Standard induction on the derivation. □

Using standard techniques (see [4]) one can prove cut-elimination for all the above systems. We skip the proof in this paper.

Theorem 3.4 *Any logic* $\mathcal{G} \in \{\mathsf{GS4}^2, \mathsf{GS4}^2_{\mathsf{mon}}, \mathsf{GS4}^2_{\mathsf{triv}}\}$ *enjoys cut-elimination*: *if* $\mathcal{G} \vdash \Gamma \Rightarrow \Delta$, *then* $\Gamma \Rightarrow \Delta$ *can be derived in* $\mathcal{G}$ *without using of the Cut-rule*.

Step 2. By a (normal) realization of a modal sequent $\Gamma \Rightarrow \Delta$ in a justification logic L we mean a (normal) realization of its standard translation $\wedge\Gamma \to \vee\Delta$. Theorem 3.2 immediately follows from the lemma below.

Lemma 3.5

1. $\mathsf{GS4}^2 \vdash \Gamma \Rightarrow \Delta$ *iff there exists a normal realization such that* $\mathsf{LP}^2 \vdash (\wedge\Gamma \to \vee\Delta)^r$.
2. *For* $L \in \{\mathsf{LP}^2_!, \mathsf{LP}^2_\uparrow\}$, $\mathsf{GS4}^2_{\mathsf{mon}} \vdash \Gamma \Rightarrow \Delta$ *iff there exists a normal realization in the language* L *such that* $L \vdash (\wedge\Gamma \to \vee\Delta)^r$.
3. *For* $L \in \{\mathsf{LP}^2_{!!}, \mathsf{LP}^2_{\uparrow\uparrow}\}$, $\mathsf{GS4}^2_{\mathsf{triv}} \vdash \Gamma \Rightarrow \Delta$ *iff there exists a normal realization in the language* L *such that* $L \vdash (\wedge\Gamma \to \vee\Delta)^r$.

Proof Similar to the proof of the realization theorem for LP (see [2]) and goes by induction on the cut-free derivation of $\Gamma \Rightarrow \Delta$. We construct normal realizations provable in L for all sequent in this derivation.

Let $\mathcal{D}$ be a cut-free derivation of $\Gamma \Rightarrow \Delta$ in any of the Gentzen-style modal logics $\mathcal{G}$ mentioned in this theorem. We denote the explicit justification counterpart of $\mathcal{G}$ by L. Following [2], we split all occurrences of $\Box_i$'s in the derivation $\mathcal{D}$ into the families of related ones. Namely, two occurrences of $\Box_i$'s are related if they occur in related subformulas of premises and conclusions of rules; we extend this relationship by transitivity. All the rules of $\mathcal{G}$ respect polarities, hence all $\Box_i$'s in every family have the same polarity. So, we can speak about positive and negative families of $\Box_i$'s.

If f is a positive family, then all $\Box_i$'s from f are introduced either by weakening on the right or by the rule Right($\Box_i$). If at least one of $\Box_i$'s in f is introduced by Right($\Box_i$), then we call f an *essential* family, otherwise f is called *inessential family*.

Initialization To every negative or inessential positive family f of $\Box_i$ we assign a distinct fresh evidence variable p^i. Replace all boxes in the family f by p^i.

Suppose that f is an essential positive family. We enumerate the rules Right($\Box_i$) which introduce $\Box_i$'s from the family f. Let $n(f)$ be the total number of such rules for the family f. For the Right($\Box_i$)-rule number k in a family f where $k = 1, \ldots, n_f$, we take a fresh evidence variable u^i_k called *a provisional variable*. Finally, replace all $\Box_i$'s from the family f by the term $u^i_1 +_i \cdots +_i u^i_{n(f)}$.

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

Realization We replace provisional variables in the tree $\mathcal{D}'$ by evidence terms proceeding from leaves to the root. We chose these terms in such a way that after the process passes the node the sequent assigned to this node becomes derivable in L; we prove this last fact by induction on the depth of a node in $\mathcal{D}'$. We keep the notation u^i_j for both provisional variables and terms substituted for them.

We do not change realization of formulas for the axioms and all the sequents which are not conclusions of a Right($\Box_i$)-rule. Note that all rules except from Right($\Box_i$) are admissible in L, therefore these sequents are derivable in L.

If a sequent $\Gamma \Rightarrow \Delta$ assigned to the current node of $\mathcal{D}'$ is introduced by a Right($\Box_i$)-rule, then we replace the corresponding provisional variable by a term.

We consider the case, when $\mathcal{G} = \mathsf{S4}^2_{\mathsf{mon}}$ and $L = \mathsf{LP}^2_{\uparrow}$. The other cases are similar.

Suppose that $\Gamma \Rightarrow \Delta$ was introduced by the rule Right($\Box_i$). Then the right box introduced by this rule belongs to an essential family f and has number k in f.

For $i = 1$, the corresponding node in $\mathcal{D}'$ has the form

$$\frac{[\![x^1_1]\!]_1 A_1, \ldots, [\![x^1_n]\!]_1 A_n \Rightarrow F}{[\![x^1_1]\!]_1 A_1, \ldots, [\![x^1_n]\!]_1 A_n \Rightarrow [\![u^1_1 +_1 \cdots +_1 u^1_{n(f)}]\!] F},$$

where x^1_j are evidence variables, u^1_j are evidence terms and u^1_k is an evidence variable. By the induction hypothesis,

$$\mathsf{LP}^2_{\uparrow} \vdash [\![x^1_1]\!]_1 A_1 \wedge \cdots \wedge [\![x^1_n]\!]_1 A_n \to F.$$

Then by Lemma 2.6 there exists a term $t \in Tm_1(\mathsf{LP}^2_{\uparrow})$ such that

$$\mathsf{LP}^2_{\uparrow} \vdash [\![t]\!]_1([\![x^1_1]\!]_1 A_1 \wedge \cdots \wedge [\![x^1_n]\!]_1 A_n \to F).$$

Also, by Lemma 2.8, for $\delta_1 = [\![x^1_1]\!]_1 A_1 \wedge \cdots \wedge [\![x^1_n]\!]_1 A_n$ there exists a term $s \in Tm_1(\mathsf{LP}^2_{\uparrow})$ such that $\mathsf{LP}^2_{\uparrow} \vdash \delta_1 \to [\![s]\!]_1 \delta_1$, hence

$$\mathsf{LP}^2_{\uparrow} \vdash [\![x^1_1]\!]_1 A_1 \wedge \cdots \wedge [\![x^1_n]\!]_1 A_n \to [\![t \times_1 s]\!]_1 F.$$

and by axiom Ax($+_1$) $\mathsf{LP}^2_{\uparrow}$ derives

$$\begin{aligned}&[\![x^1_1]\!]_1 A_1 \wedge \cdots \wedge [\![x^1_n]\!]_1 A_n\\ &\quad \to [\![u^1_1 +_1 \cdots +_1 u^1_{k-1} +_1\ t \times_1 s\ +_1 u^1_{k+1} +_1 \cdots +_1 u^1_{n(f)}]\!]_1 F.\end{aligned}$$

We replace the provisional variable u^1_k by $t \times_1 s$ everywhere in $\mathcal{D}'$.

Similarly, for $i = 2$ the corresponding node in $\mathcal{D}'$ has the form

$$\frac{[\![x^1_1]\!]_1 A_1, \ldots, [\![x^1_n]\!]_1 A_n, [\![y^2_1]\!]_2 B_1, \ldots, [\![y^2_l]\!]_2 B_l \Rightarrow F}{[\![x^1_1]\!]_1 A_1, \ldots, [\![x^1_n]\!]_1 A_n, [\![y^2_1]\!]_2 B_1, \ldots, [\![y^2_l]\!]_2 B_l \Rightarrow [\![u^2_2 +_2 \cdots +_2 u^2_{n(f)}]\!] F},$$

where x^1_j, y^2_j are evidence variables, u^2_j are evidence terms and u^2_k is an evidence variable. By the induction hypothesis,

$$\mathsf{LP}^2_{\uparrow} \vdash [\![x^1_1]\!]_1 A_1 \wedge \cdots \wedge [\![x^1_n]\!]_1 A_n \wedge [\![y^2_1]\!]_2 B_1 \wedge \cdots \wedge [\![y^2_l]\!]_2 B_l \to F.$$

Then by Lemma 2.6 there exists a term $t \in Tm_2(\mathsf{LP}^2_{\uparrow})$ such that

$$\mathsf{LP}^2_{\uparrow} \vdash [\![t]\!]_2([\![x^1_1]\!]_1 A_1 \wedge \cdots \wedge [\![x^1_n]\!]_1 A_n \wedge [\![y^2_1]\!]_2 B_1 \wedge \cdots \wedge [\![y^2_l]\!]_2 B_l \to F).$$

Also, by Lemma 2.8, for $\delta = [\![x_1^1]\!]_1 A_1 \wedge \cdots \wedge [\![x_n^1]\!]_1 A_n \wedge [\![y_1^2]\!]_2 B_1 \wedge \cdots \wedge [\![y_l^2]\!]_2 B_l$ there exists a term $s \in Tm_2(\mathsf{LP}^2_{\uparrow})$ such that $\mathsf{LP}^2_{\uparrow} \vdash \delta \to [\![s]\!]_2 \delta$, hence

$$\mathsf{LP}^2_{\uparrow} \vdash [\![x_1^1]\!]_1 A_1 \wedge \cdots \wedge [\![x_n^1]\!]_1 A_n \wedge [\![y_1^2]\!]_2 B_1 \wedge \cdots \wedge [\![y_l^2]\!]_2 B_l \to [\![t \times_2 s]\!]_2 F.$$

and by axiom $\mathrm{Ax}(+_2)$ $\mathsf{LP}^2_{\uparrow}$ derives

$$[\![x_1^1]\!]_1 A_1 \wedge \cdots \wedge [\![x_n^1]\!]_1 A_n \wedge [\![y_1^2]\!]_2 B_1 \wedge \cdots \wedge [\![y_l^2]\!]_2 B_l \to [\![u_1^2 +_2 \cdots +_2 u_{k-1}^2 +_2 \ t \times_2 s \ +_2 u_{k+1}^2 +_2 \cdots +_2 u_{n(f)}^2]\!]_2 F.$$

We replace the provisional variable u_k^2 by $t \times_2 s$ everywhere in $\mathcal{D}'$.

For both cases $i = 1$ and $i = 2$, the formulas above the current node remain provable in $\mathsf{LP}^2_{\uparrow}$ due to Lemma 2.5. Further, the terms we substitute do not contain provisional variables, so the total number of provisional variables in $\mathcal{D}'$ decreases. The conclusion of the given rule Right($\Box_i$) becomes provable in $\mathsf{LP}^2_{\uparrow}$, so the induction step is completed.

Finally, we arrive at the root of $\mathcal{D}'$, replace all provisional variables by terms on non-provisional ones and prove derivabiliy of the root sequent in $\mathcal{D}'$ in $\mathsf{LP}^2_{\uparrow}$. The realization constructed in this way is normal. □

4 Symbolic Semantics

Models of multi-agent logics of explicit knowledge below are natural generalizations of Mkrtychev models for LP (cf. [7]).

Definition 4.1 Let L be any language from Definition 2.1. An *L-model* $\mathcal{M} = (\#, v)$ consists of two objects:

- $\#$ is a mapping from proof terms of L to sets of formulas of L, called *an evidence function*;
- v is a truth evaluation of sentence variables.

For every functional symbol from L the evidence function $\#$ should satisfy the corresponding closure condition from the list given below: suppose that t, s are in $Tm_i(L)$, $i = 1, 2$:

- if $(A \to B) \in \#(t)$ and $A \in \#(s)$, then $B \in \#(t \times_i s)$;
- if $A \in \#(t)$, then $A \in \#(t +_i s)$ and $A \in \#(s +_i t)$;
- if $A \in \#(t)$, then $[\![t]\!]_i A \in \#(!_i t)$;
- if $A \in \#(u)$ and $u \in Tm_1(L)$, then $[\![u]\!]_1 A \in \#(!_1^2 u)$;
- if $A \in \#(v)$ and $v \in Tm_2(L)$, then $[\![v]\!]_2 A \in \#(!_2^1 v)$;
- if $A \in \#(u)$ and $u \in Tm_1(L)$, then $A \in \#(\uparrow_1^2 u)$;
- if $A \in \#(v)$ and $v \in Tm_2(L)$, then $A \in \#(\uparrow_2^1 v)$.

Definition of the truth relation $\mathcal{M} \models A$ (A is valid in $\mathcal{M}$) is inductive: for propositional variables $\mathcal{M} \models S$ iff $v(S) = true$, $\models$ commutes with Boolean connectives and for $t_i \in Tm_i$

$$\mathcal{M} \models [\![t]\!]_i A \quad \rightleftharpoons \quad A \in \#(t) \text{ and } \mathcal{M} \models A.$$

An L-model $\mathcal{M} = (\#, v)$ is called *finitely generated*(or f.g. for short) if:

- for every term t the set $\#(t)$ is finite; the set $\{p \in PVar \mid \#(p) \neq \emptyset\}$ is finite;
- the set of terms, for which the converse of the conditions on the evidence function does not hold, is finite;
- the set $\{S \in SVar \mid v(S) = true\}$ is finite.

Definition 4.2 For any logic L from Definition 2.3 a *constant specification CS* is any finite set of formulas derived by the axiom necessitation rule. We say that $L \vdash A$ *meeting CS* if all axiom necessitation rules in the derivation of A introduce formulas from CS. We say that an L–model $\mathcal{M}$ *meets CS* if all formulas from CS are valid in $\mathcal{M}$, that is, $\mathcal{M} \models (\wedge CS)$.

Theorem 4.3 *Let L be any logic from Definition 2.3:*

1. *If $L \vdash A$ meeting CS, then $\mathcal{M} \models A$ for every L-model $\mathcal{M}$ meeting CS.*
2. *If $L \nvdash A$ meeting CS, then there exists a f.g. L-model $\mathcal{M}$ meeting CS such that $\mathcal{M} \not\models A$.*

Proof We give the proof for $L = \mathsf{LP}^2$; for the remaining systems the proof differs in saturation and completion algorithms (see below) to which the instructions corresponding to the additional operations should be added. We sketch these additions. It suffices to consider the case $CS = \emptyset$; the general case can be reduces to this one by deduction theorem (see Lemma 2.5).

Soundness can be easily proven by induction on the derivation of A. In order to prove completeness suppose that $\mathsf{LP}^2 \nvdash A$. We will construct a finitely generated LP^2-model $\mathcal{M} = (\#, v)$ such that $\mathcal{M} \not\models A$.

Step 1: Saturation Algorithm It constructs a finite set of formulas $Sat(A)$ which is called an *adequate set*; from now on we consider only formulas and terms containing only variables from $Var(A)$. We need the following definition: the *complexity of a proof term t* denoted by $|t|$ is the length of the longest branch in the tree representing this term. The saturation algorithm works as follows:

1. Initialization. Put $Sat_0(A) := SubFm(A)$. Calculate the maximal complexity of terms which occur in A; let N denote the result.
2. For every $l = 1, \ldots, N+1$ we calculate the set $Sat_l(A)$ as follows:
   - Initially $Sat_l(A) := Sat_{l-1}(A)$.
   - If $[\![t]\!]_i(F \to G), [\![s]\!]_i F \in Sat_{l-1}(A)$ for $i = 1$ or 2, then extend $Sat_l(A)$ by $[\![t \times_i s]\!]_i G$.
   - If $[\![t]\!]_i F \in Sat_{l-1}(A)$, where $i = 1$ or 2, and s is a term from $Tm_i(\mathsf{LP}^2)$ s.t. $|s| \leq l$, then extend $Sat_l(A)$ by $[\![t +_i s]\!]_i F$ and $[\![s +_i t]\!]_i F$.

- If $[\![t]\!]_i F \in Sat_{l-1}(A)$ then extend $Sat_l(A)$ by $[\![!_i t]\!]_i [\![t]\!]_i F$.

3. Put $Sat(A) := Sat_{N+1}(A)$.

For the extensions of LP^2, add to step 2 of the Saturation Algorithm the appropriate instructions from the list below:

- If $[\![t]\!]_1 F \in Sat_{l-1}(A)$, then extend $Sat_l(A)$ by $[\![!_1^2 t]\!]_2 [\![t]\!]_1 F$.
- If $[\![t]\!]_2 F \in Sat_{l-1}(A)$, then extend $Sat_l(A)$ by $[\![!_2^1 t]\!]_1 [\![t]\!]_2 F$.
- If $[\![t]\!]_1 F \in Sat_{l-1}(A)$, then extend $Sat_l(A)$ by $[\![\uparrow_1^2 t]\!]_2 F$.
- If $[\![t]\!]_2 F \in Sat_{l-1}(A)$, then extend $Sat_l(A)$ by $[\![\uparrow_2^1 t]\!]_1 F$.

Lemma 4.4 *For every* $l = 0, \ldots, N$:

1. $Sat_l(A)$ *is closed under subformulas, that is,* $SubFm(Sat_l(A)) \subseteq Sat_l(A)$.
2. *If* $G \in Sat_{l+1}(A) \setminus Sat_l(A)$ *then* G *has the form* $[\![t]\!]E$ *and* $|t| \geq l+1$.
3. *If* $[\![t]\!]_i (F \to G), [\![s]\!]_i F \in Sat_l(A)$ *then* $[\![t \times_i s]\!]_i G \in Sat_{l+1}(A)$.
   If $[\![t]\!]_i G \in Sat_l(A)$ *and* $|t +_i s| \leq l+1$, *then* $[\![t +_i s]\!]_i G, [\![s +_i t]\!]_i G \in Sat(A)$.
   If $[\![t]\!]_i G \in Sat_l(A)$ *then* $[\![!_i t]\!]_i [\![t]\!]_i G \in Sat_{l+1}(A)$.

Proof Joint induction on l. Item 3 is trivial.

The induction base for item 1 trivially follows from the definition of $Sat_0(A)$. For the induction step, assume that $B \in SubFm(Sat_{l+1}(A))$. Then either $B \in SubFm(Sat_l(A))$ or $B \in SubFm(C)$ where C is a formula from $Sat_{l+1}(A) \setminus Sat_l(A)$. In the former case $B \in Sat_l(A)$ by the induction hypothesis. In the later case C was added to $Sat_l(A)$ by some instruction of the saturation algorithm. There are three possibilities of how C was introduced. Let us consider the first one when $C = [\![t \times_i s]\!]_i G$ where formulas $[\![t]\!]_i (F \to G)$ and $[\![s]\!]_i F$ belong to $Sat_l(A)$ for some F. Then either B coincides with C and thus belongs to $Sat_{l+1}(A)$ or B is a subformula of G. In this later case B belongs to $Sat_l(A)$ by the induction hypothesis. The remaining two instructions of the saturation algorithm can be treated similarly.

Let us prove item 2. The induction base is obvious. For the induction step suppose that $l \geq 1$ and a formula $[\![t]\!]_i B$ belongs to $Sat_{l+1}(A) \setminus Sat_l(A)$. Then this formula was added by an instruction of the saturation algorithm. Let us consider the first one when $t = s \times_i u$ and formulas $[\![s]\!]_i (C \to B)$ and $[\![u]\!]_i C$ belong to $Sat_l(A)$. Since $[\![t]\!]_i B$ does not occur in $Sat_l(A)$, one of the formulas $[\![s]\!]_i (C \to B)$ and $[\![u]\!]_i C$ does not occur in $Sat_{l-1}(A)$ (otherwise $[\![t]\!]_i B$ would be added to $Sat_{l-1}(A)$ on the previous iteration of the saturation step). By the induction hypothesis this yields that either $|s| \geq l$ or $|u| \geq l$. Therefore, $|t| = \max(|s|, |u|) + 1 \geq l+1$. The remaining instructions of the saturation algorithm can considered in the similar way. □

Corollary 4.5 (Properties of adequate sets):

1. $Sat(A)$ *is closed under subformulas, that is,* $SubFm(Sat(A)) \subseteq Sat(A)$.
2. *If* $[\![t]\!]_i (F \to G), [\![s]\!]_i F \in Sat(A)$ *and* $|t \times_i s| \leq N$ *then* $[\![t \times_i s]\!]_i G \in Sat(A)$.
   If $[\![t]\!]_i G \in Sat(A)$ *and* $|t +_i s| \leq N$, *then* $[\![t +_i s]\!]_i G, [\![s +_i t]\!]_i G \in Sat(A)$.
   If $[\![t]\!]_i G \in Sat(A)$ *and* $|!_i t| \leq N$ *then* $[\![!_i t]\!]_i [\![t]\!]_i G \in Sat(A)$.

Proof Item 1 immediately follows from the previous lemma. Let us prove item 2. Let us remind that $Sat(A) = Sat_{N+1}A$.

If $[\![t]\!]_i(F \to G), [\![s]\!]_i F \in Sat(A)_{N+1}$ and $|t \times_i s| \le N$, then from item 2 of Lemma 4.4 we obtain that $[\![t]\!]_i(F \to G), [\![s]\!]_i F \in Sat\,(A)_N$. Hence $[\![t \times_i s]\!]_i G \in Sat_{N+1}(A)$ by item 3 of the same lemma. The remaining cases can be proved similarly. □

Step 2. Now we describe a translation of the language LP^2 into the pure propositional language. For every q-atom $[\![t]\!]_i B \in Sat(A)$ we reserve a fresh propositional variable $S_{t,i,B}$. For every formula G whose all q-atomic subformulas belong to $Sat(A)$ by G' we denote the result of substitution the corresponding propositional variables for all outermost occurrences of q-atomic subformulas in G. Namely, we define G' by induction on the construction of G: for propositional variables $S' \rightleftharpoons S$; $(\cdot)'$ commutes with boolean connectives and $([\![t]\!]_i B)' \rightleftharpoons S_{t,i,B}$.

Let $Ax(A)$ stand for the conjunction of all axioms of LP^2 except A0 whose q-atomic subformulas are from $Sat(A)$. Put $A_p \rightleftharpoons (Ax(A) \to A)'$. Since $\mathsf{LP}^2 \nvdash A$, we conclude that A_p is not provable in propositional logic (otherwise after the reverse substitution of $[\![t]\!]_i B$ for $S_{t,i,B}$ in the derivation of A_p in propositional calculus we get $\mathsf{LP}^2 \vdash Ax(A) \to A$, hence $\mathsf{LP}^2 \vdash A$). Therefore, there exists an evaluation w of propositional letters from A_p by $(true, false)$, such that $w(A_p) = false$. Define

$$\Gamma_0 \rightleftharpoons \{B \in Sat(A) \mid w(B') = true\},$$
$$\Delta_0 \rightleftharpoons \{B \in Sat(A) \mid w(B') = false\}.$$

Lemma 4.6 *The sets* Γ_0 *and* Δ_0 *has the following properties*:

1. $\Gamma_0 \cap \Delta_0 = \emptyset$ *and* $SubFm(\Gamma_0 \cup \Delta_0) \subseteq \Gamma_0 \cup \Delta_0$.
2. *If* $[\![t]\!]_i E_i \in \Gamma_0$, *then* $E \in \Gamma_0$.
3. *If* $[\![t]\!]_i(F \to G), [\![s]\!]_i F \in \Gamma_0$ *and* $|t \times_i s| \le N$, *then* $[\![t \times_i s]\!]_i G \in \Gamma_0$.
   If $[\![t]\!]_i G \in \Gamma_0$ *and* $|t +_i s| \le N$, *then* $[\![t +_i s]\!]_i G \in \Gamma_0$ *and* $[\![s +_i t]\!]_i G \in \Gamma_0$.
   If $[\![t]\!]_i G \in \Gamma_0$ *and* $|!_i t| \le N$, *then* $[\![!_i t]\!]_i [\![t]\!]_i G \in \Gamma_0$.
4. *If* $F \to G \in \Gamma_0$, *then either* $F \in \Delta_0$ *or* $G \in \Gamma_0$.
   If $F \to G \in \Delta_0$, *then* $F \in \Gamma_0$ *and* $G \in \Delta_0$.

Proof Items 1 and 4 are trivial. Let us prove item 2. If $[\![t]\!]_i E \in \Gamma_0$, then $[\![t]\!]_i E \in Sat(A)$ and $w(([\![t]\!]_i E)') = true$. By definition, $Ax(A)$ contains the formula $[\![t]\!]_i E \to E$, which is an instance of the axiom A1. Thus, $w(([\![t]\!]_i E \to E)') = true$. Therefore $w(E') = true$, hence $E \in \Gamma_0$.

Proof of item 3 is similar. Suppose that $[\![t]\!]_i(F \to G)$, $[\![s]\!]_i F \in \Gamma_0$ and $|t \times_i s| \le N$. Then $[\![t]\!]_i(F \to G), [\![s]\!]_i F \in Sat(A)$. By Corollary 4.5, $[\![t \times_i s]\!]_i G \in Sat(A)$. Therefore the formula $[\![t]\!]_i(F \to G) \to ([\![s]\!]_i F \to [\![t \times_i s]\!]_i G)$, which is an instance of axiom $\mathrm{Ax}(\times_i)$, belongs to $Ax(A)$. Thus, the $(\cdot)'$-translation of this formula is true under evaluation w. Since $w([\![t]\!]_i(F \to G))' = true$ and $w([\![s]\!]_i F)' = true$, we obtain that $w([\![t \times_i s]\!]_i G)' = true$. Hence $[\![t \times_i s]\!]_i G \in \Gamma_0$. The remaining two cases are similar. □

Step 3. Completion Algorithm It goes through infinite number of iterations; the l-th iteration produces the set Γ_l which is finite. Here again we restrict the language

by variables from $Var(A)$. Start with Γ_0. For every $l = 1, 2, \ldots$ on the l-th iteration construct the set Γ_l as follows:

- Initially $\Gamma_l := \Gamma_{l-1}$.
- If $[\![t]\!]_i(F \to G), [\![s]\!]_i F \in \Gamma_{l-1}$, then extend Γ_l by $[\![t \times_i s]\!]G$.
- If $[\![t]\!]_i F \in \Gamma_{l-1}$ and $|s| \le l$, then extend Γ_l by $[\![t +_i s]\!]_i F$ and $[\![s +_i t]\!]_i F$.
- If $[\![t]\!]_i F \in \Gamma_{l-1}$, then extend Γ_l by $[\![!_i t]\!]_i [\![t]\!]_l G$.
- Go to the next l.

For the extensions of LP^2 add the appropriate steps from the list below to the Completion Algorithm:

- If $[\![t]\!]_1 F \in \Gamma_{l-1}$, then extend Γ_l by $[\![!_1^2 t]\!]_2 [\![t]\!]_1 F$.
- If $[\![t]\!]_2 F \in \Gamma_{l-1}$, then extend Γ_l by $[\![!_2^1 t]\!]_1 [\![t]\!]_2 F$.
- If $[\![t]\!]_1 F \in \Gamma_{l-1}$, then extend Γ_l by $[\![\uparrow_1^2 t]\!]_2 F$.
- If $[\![t]\!]_2 F \in \Gamma_{l-1}$, then extend Γ_l by $[\![\uparrow_2^1 t]\!]_1 F$.

Put $\Gamma := \bigcup_l \Gamma_l$ and $\Delta := \Delta_0$.

Lemma 4.7 *For every $l = 0, 1, 2, \ldots$*

1. *The set Γ_l is finite and $\Gamma_l \cap \Delta = \emptyset$*
2. *If $E \in \Gamma_{l+1} \setminus \Gamma_l$, then E is of the form $[\![t]\!]G$ and $|t| \ge N + l + 1$.*
3. *$\Gamma_l \cup \Delta$ is closed under subformulas, that is, $SubFm(\Gamma_l \cup \Delta) \subseteq \Gamma_l \cup \Delta$ and if $[\![t]\!]_i G \in \Gamma_l$, then $G \in \Gamma_l$.*
4. *If $[\![t]\!]_i(F \to G), [\![s]\!]_i F \in \Gamma$, then $[\![t \times_i s]\!]_i G \in \Gamma$.*
   If $[\![t]\!]_i G \in \Gamma$, then $[\![t +_i s]\!]_i G \in \Gamma$ and $[\![s +_i t]\!]_i G \in \Gamma$.
   If $[\![t]\!]_i G \in \Gamma$, then $[\![!_i t]\!]_i [\![t]\!]_i G \in \Gamma$.
5. *For every term $t \in Tm_i(\mathsf{LP}^2)$ the set $I(t) = \{E \mid [\![t]\!]_i E \in \Gamma\}$ is finite and the function $\lambda t.I(t)$ is primitive recursive.*

Proof We prove items 1–3 by induction on l. Induction base for items 1 and 3 is due to items 1, 2 of Lemma 4.6. For item 2, suppose that $[\![t]\!]_i G \in \Gamma_1 \setminus \Gamma_0$ and $|t| < N + 1$. Then $[\![t]\!]_i G$ was introduced to Γ_1 by the Completion Algorithm. There are three possibilities: either $t = u \times_i v$, or $t = u +_i v$, or $t = !_i u$. Let us consider the first case. By the description of the Completion Algorithm there exists a formula F such that $[\![u]\!]_i(F \to G), [\![v]\!]_i F \in \Gamma_0$. Having in mind that $|u \times_i v| \le N$, by Lemma 4.6 we obtain $[\![t]\!]_i G \in \Gamma_0$, that yields a contradiction. The remaining cases are similar.

For the induction step for item 3 let us prove the second proposition; (note that the first proposition follows from it). Suppose that $[\![t]\!]_i G \in \Gamma_{l+1}$. Again, the most interesting case is when $t = u \times_i v$ and $[\![u]\!]_i(F \to G), [\![v]\!]_i F \in \Gamma_l$ for some formula F. Then by the induction hypothesis $F \to G, F \in \Gamma_l$. Since $F \to G$ is not q-atomic, by the induction hypothesis for item (2) we conclude that $F \to G \in \Gamma_0$. By Lemma 4.6 $F, G \in \Gamma_0 \cup \Delta$. By the induction hypothesis for item (1) $F \notin \Delta$, hence $F \in \Gamma_0$. Therefore by Lemma 4.6(4) $G \in \Gamma_0$.

To prove 2, suppose that $[\![t]\!]_i G \in \Gamma_{l+1} \setminus \Gamma_l$ and $|t| < N + l + 1$. Then $[\![t]\!]_i G$ was introduced to Γ_{l+1} by the Completion Algorithm and either $t = u \times_i v$, or $t = u +_i v$, or $t = !_i u$. Again, we consider the first case. By the description of the Completion Algorithm there exists a formula F such that $[\![u]\!]_i(F \to G), [\![v]\!]_i F \in \Gamma_l$. Moreover,

since $[\![u \times_i v]\!]G \notin \Gamma_l$, then either $[\![u]\!]_i(F \to G)$ or $[\![v]\!]_i F$ is not from Γ_{l-1}. By the induction hypothesis, this yields that either $|u| \geq N + l$ or $|v| \geq N + l$, hence $|u \times_i v| \geq N + l + 1$, contradiction. The remaining cases are similar.

In order to prove (1), suppose that $\Gamma_{l+1} \cap \Delta \neq \emptyset$. By the induction hypothesis, $\Gamma_l \cap \Delta \neq \emptyset$, therefore there exists a formula $[\![t]\!]_i G$ such that $[\![t]\!]_i G \in \Gamma_{l+1} \setminus \Gamma_l$ and $[\![t]\!]_i G \in \Delta$. From the first condition we obtain $|t| \geq N + l + 1$, while from the second one $[\![t]\!]_i G \in Sat(A)$, hence $|t| \leq N$. Contradiction.

Item 4 is easily seen from the description of the Completion Algorithm. Let us prove item 5. First, due to item 2

$$I(t) = \{E \mid [\![t]\!]_i E \in \Gamma\} = \{E \mid [\![t]\!]_i E \in \Gamma_{|t|}\}.$$

Since $\Gamma_{|t|}$ is finite, we conclude that $I(t)$ is finite. Then, the function which calculates the code of Γ_k given k, is primitive recursive. Therefore, $\lambda t.I(t)$ is primitive recursive too. □

Step 4. For every $t \in Tm_i$ and $S \in SVar$ put

$$\#(t) \rightleftharpoons \{E \mid [\![t]\!]_i E \in \Gamma\}, \qquad v(S) \rightleftharpoons w(S).$$

Lemma 4.8 *For every formula G*

$$\begin{array}{lcl} G \in \Gamma & \Rightarrow & \mathcal{M} \models G; \\ G \in \Delta & \Rightarrow & \mathcal{M} \not\models G. \end{array}$$

Proof Induction on G. If G is a sentence variable then

$$G \in \Gamma \iff G \in \Gamma_0 \iff w(G) = true \iff v(G) = true.$$

Here the first equivalence is due to Lemma 4.7(2) and the remaining are by definition of Γ_0 and v.

Suppose that G has the form $F \to H$. If $G \in \Gamma$, then by Lemma 4.7(2) $G \in \Gamma_0$. Hence by Lemma 4.6(4) either $F \in \Delta$ or $H \in \Gamma_0$. By the induction hypothesis we obtain that either $\mathcal{M} \not\models F$ or $\mathcal{M} \models H$, therefore $\mathcal{M} \models F \to H$. If $G \in \Delta$, then by Lemma 4.6(4) $F \in \Gamma_0$ and $H \in \Delta$. By the induction hypothesis we obtain that $\mathcal{M} \models F$ and $\mathcal{M} \not\models H$, therefore $\mathcal{M} \not\models F \to H$.

Suppose that G has the form $[\![t]\!]_i F$. If $G \in \Gamma$, then $F \in \Gamma$ by Lemma 4.7(3) and $F \in \#(t)$ by definition of #. Therefore $\mathcal{M} \models F$ by the induction hypothesis, hence $\mathcal{M} \models [\![t]\!]_i F$. If $G \in \Delta$ then by Lemma 4.7(1) $G \notin \Gamma$, hence $F \notin \#(t)$ by definition of #. Therefore $\mathcal{M} \not\models [\![t]\!]_i F$. □

From Lemmas 4.8 and 4.7 it follows that $\mathcal{M}$ is a finitely generated model for LP^2. Since $w(A') = false$, we conclude $A \in \Delta$. Therefore $\mathcal{M} \not\models A$. This completes the proof of the theorem. □

Corollary 4.9 *All the logics from Definition 2.3 are decidable.*

Proof The decision algorithm for any of these logics can be obtained by an appropriate modification of the decision algorithm for LP. See [6] fore more details on the last one. □

Epistemic semantics for LP^2 and its extensions is given by the following natural generalization of Fitting models (cf. [5]). For any language L from Definition 2.1 one could define *Fitting models* as follows. An *L-model* $\mathcal{M} = (W, R_1, R_2, \mathcal{E}, v)$ has the following parameters:

- a nonempty set of possible worlds W;
- two reflexive transitive accessibility relations on W denoted by R_1, R_2;
- an evidence function $\mathcal{E}$ which maps $W \times Tm(L)$ to sets of formulas of L;
- for every $x \in W$ a truth evaluation $v(x)$ maps propositional variables to $\{true, false\}$.

We require that for every node $x \in W$ the restriction of $\mathcal{E}$ to x satisfies all the conditions for # and $\mathcal{E}$ is monotone in the following sense: for $i = 1, 2$ if $x R_i y$ and $t \in Tm_i(L)$ then $\mathcal{E}(x, t) \subseteq \mathcal{E}(y, t)$.

The truth relation for every node $x \in W$ is defined in the standard way; we put $\mathcal{M}, x \models [\![t]\!]_i F$ iff $F \in \mathcal{E}(x, t)$ and $\mathcal{M}, y \models F$ for every $y \in W$ such that $x R_i y$.

Note that a model in the sense of Definition 4.1 is a Fitting model, namely, take W a singleton set and R_1, R_2 total relations on W. It is easy to prove that all the logics considered in this paper are sound and complete with respect to the models just described. In particular, the completeness with respect to Fitting semantics follows from Theorem 4.3 and the fact that aforementioned Mkrychev models are singleton versions of the corresponding Fitting models.

5 Arithmetical Semantics

In this section we consider evidences as proofs in concrete systems which are able to formalize there own proof predicate. Operations become operations on proofs, and operators $[\![\cdot]\!]_i(\cdot)$ represent two proof predicates. Let us describe the interpretation of our logics in Peano Arithmetic PA (the definition of PA and related topics can be found in [8]).

Definition 5.1 A normal proof predicate *Prf* is an arithmetical provably Δ_1 formula satisfying the following conditions:

1. for every arithmetical formula φ, $\mathsf{PA} \vdash \varphi$ iff there exists a natural number n such that $Prf(n, \ulcorner \varphi \urcorner)$ is true;
2. for every n the set $Th(n) \rightleftharpoons \{\varphi \mid Prf\ (n, \ulcorner \varphi \urcorner)\}$ is finite and the function $n \mapsto Th(n)$ is total recursive;
3. for every finite set of arithmetical theorems Γ there exists a natural number n such that $\Gamma \subseteq Th(n)$.

A typical example of such a predicate which will be useful later is a provably Δ_1-formula *Proof* which expresses the following relation

x is a code of a finite set of PA*-derivations,*
y is a code of the last formula in one of them.

Lemma 5.2 *For every pair of normal proof predicates* Prf_1, Prf_2 *there exist total recursive functions* app_i, check_i, choice_i $(i = 1, 2)$, check_1^2, check_2^1, $\mathsf{convert}_1^2$ *and* $\mathsf{convert}_2^1$ *such that for all natural numbers* k, n *for all arithmetical sentences* φ, ψ *the following formulas are derivable in* PA:

1. $Prf_i(k, \ulcorner \varphi \to \psi \urcorner) \to (Prf_i(n, \ulcorner \varphi \urcorner) \to Prf_i(\mathsf{app}_i(k, n), \ulcorner \psi \urcorner))$
2. $Prf_i(k, \ulcorner \varphi \urcorner) \to Prf_i(\mathsf{choice}_i(k, n), \ulcorner \varphi \urcorner) \wedge Prf_i(\mathsf{choice}_i(n, k), \ulcorner \varphi \urcorner)$
3. $Prf_i(k, \ulcorner \varphi \urcorner) \to Prf_i(\mathsf{check}_i(k), \ulcorner Prf_i(k, \ulcorner \varphi \urcorner) \urcorner)$
4. $Prf_1(k, \ulcorner \varphi \urcorner) \to Prf_2(\mathsf{check}_1^2(k), \ulcorner Prf_1(k, \ulcorner \varphi \urcorner) \urcorner)$
   $Prf_2(k, \ulcorner \varphi \urcorner) \to Prf_1(\mathsf{check}_2^1(k), \ulcorner Prf_2(k, \ulcorner \varphi \urcorner) \urcorner)$
5. $Prf_1(k, \ulcorner \varphi \urcorner) \to Prf_2(\mathsf{convert}_1^2(k), \ulcorner \varphi \urcorner)$
   $Prf_2(k, \ulcorner \varphi \urcorner) \to Prf_1(\mathsf{convert}_2^1(k), \ulcorner \varphi \urcorner)$

Proof Note that all formulas 1–5 are provably Δ_1, therefore their provability follows from their validity in the standard model of arithmetic.

Let us describe the algorithm which calculates function $\mathsf{convert}_1^2$. Given k, it reconstructs the set T of formulas φ, such that $Prf_1(k, \ulcorner \varphi \urcorner)$. Due to conditions 1 and 2 on normal proof predicates (see Definition 5.1), this operation is recursive, the resulting set T is finite and consists of PA-theorems. Therefore, by condition 3 of the same definition, there exists a natural number n such that $Prf_2(n, \ulcorner \varphi \urcorner)$ for any formula φ from T. The algorithm consequently tries $n = 0, 1, 2, \ldots$ until n meets the requirement above.

All the remaining algorithms work similarly. □

Definition 5.3 Let L be one of the languages from Definition 2.3. An *arithmetical interpretation* $*$ for the language L has the following parameters:

- two normal proof predicates Prf_1 and Prf_2;
- total recursive functions for operations of L which satisfy Lemma 5.2;
- an evaluation $(\cdot)^*$ that assigns natural numbers to proof variables and arithmetical sentences to propositional variables.

Arithmetical evaluation $(\cdot)^*$ can be extended to all L-terms and formulas in the following way. For terms, we put

$$\begin{aligned}
(t \times_i s)^* &\rightleftharpoons \mathsf{app}_i(t^*, s^*),\\
(t +_i s)^* &\rightleftharpoons \mathsf{choice}_i(t^*, s^*),\\
(!_i t)^* &\rightleftharpoons \mathsf{check}_i(t^*),\\
(!_1^2 t)^* &\rightleftharpoons \mathsf{check}_1^2(t^*), \quad \textit{similarly for } !_2^1,\\
(\uparrow_1^2 t)^* &\rightleftharpoons \mathsf{convert}_1^2(t^*), \quad \textit{similarly for } \uparrow_2^1.
\end{aligned}$$

Then, $*$ commutes with Boolean connectives and

$$([\![t]\!]_i A)^* \rightleftharpoons Prf_i(t^*, \ulcorner A^* \urcorner).$$

For a given constant specification CS we call $*$ a *CS-interpretation* if all formulas from CS^* are true in the standard model of arithmetic.

Theorem 5.4 (Arithmetical soundness and completeness) *Let L be one of the logics from Definition 2.3, CS be a constant specification for L. For every L-formula A the following three propositions are equivalent*:

1. $L \vdash A$ *meeting* CS;
2. *for every CS-interpretation* $*$, $\mathsf{PA} \vdash A^*$;
3. *for every CS-interpretation* $*$, A^* *is true*.

Proof Let us give the proof for $L = \mathsf{LP}_!^2$. For other systems it is similar. In view of the deduction theorem we may consider the case $CS = \emptyset$.

Trivially, $1 \Rightarrow 2 \Rightarrow 3$. Let us prove that $3 \Rightarrow 1$. Suppose that $\mathsf{LP}_!^2 \nvdash A$. By Theorem 4.3, there exists a finitely generated $\mathsf{LP}_!^2$-model $\mathcal{M} = (\#, v)$ such that $\mathcal{M} \not\models A$.

We are going to define an embedding of this model into Peano Arithmetic. We fix an injective Gödel numbering of the joint syntax of $\mathsf{LP}_!^2$ and Peano Arithmetic. We first consider an auxiliary evaluation $(\cdot)^@$: for every sentence variable S_i and proof term t put

$$t^@ \rightleftharpoons \ulcorner t \urcorner, \qquad S_i^@ \rightleftharpoons \begin{cases} i = i, & \text{if } \mathcal{M} \models S_i, \\ i = 0, & \text{otherwise.} \end{cases}$$

We find the appropriate proof predicates by the multiple fixed point equation. Since the model $\mathcal{M}$ is finitely generated, the relation "$\mathcal{M} \models [\![t]\!]F$" is primitive recursive, thus, it can be represented by a provably Δ_1 arithmetical formula. The function, which, being given Gödel numbers of arithmetical formulas Prf_1, Prf_2 and an $\mathsf{LP}_!^2$-formula B, calculates $B^@$, is primitive recursive too. Therefore, by the fixed point lemma for PA, there exist arithmetical Δ_1 formulas Prf_i, such that PA derives

$$\begin{aligned} Prf_i(x, y) \leftrightarrow\ & Proof(x, y) \vee \\ & \text{“}x = \ulcorner t \urcorner \text{ and } y = \ulcorner B^@ \urcorner \text{ and } \mathcal{M} \models [\![t]\!]_i B\text{”}, \end{aligned} \tag{FPE}$$

where *Proof* is a predicate defined before.

Note that the mapping $(\cdot)^@$ is injective, namely, for different formulas B and C we have $B^@ \neq C^@$.

Lemma 5.5 *For every formula C*

$$\begin{aligned} \mathcal{M} \models C \quad &\Rightarrow \quad \mathsf{PA} \vdash C^@; \\ \mathcal{M} \not\models C \quad &\Rightarrow \quad \mathsf{PA} \vdash \neg C^@. \end{aligned} \tag{1}$$

Proof Induction on the formula C. The induction base when C is a propositional variable is a direct corollary of the definition of $@$. The case of boolean connectives immediately follows from the induction hypothesis.

Suppose that C has the form $[\![t]\!]_i B$. If $\mathcal{M} \models C$, then by the definition of Prf_i we conclude that $Prf_i(\ulcorner t \urcorner, \ulcorner B^@ \urcorner)$ is true. Hence $\mathsf{PA} \vdash Prf(\ulcorner t \urcorner, \ulcorner B^@ \urcorner)$, that is $\mathsf{PA} \vdash C^@$.

If $\mathcal{M} \not\models C$, then $Prf_i(\ulcorner t \urcorner, \ulcorner B^{@} \urcorner)$ is false because of the injectivity of the evaluation $(\cdot)^{@}$. Therefore $\mathsf{PA} \vdash \neg Prf(\ulcorner t \urcorner, \ulcorner B^{@} \urcorner)$, that is, $\mathsf{PA} \vdash \neg C^{@}$. □

Lemma 5.6 *Prf is a normal proof predicate.*

Proof From the definitions of a finitely generated model and (FPE) it follows that for every proof term x the set $Th_i(\ulcorner x \urcorner) = \{\varphi \mid Prf_i(x, \ulcorner \varphi \urcorner)\}$ is finite and the functions $\lambda x.Th_i(x)$ are recursive.

It only remains to verify that if $Prf_i(n, \ulcorner \varphi \urcorner)$ is true then $\mathsf{PA} \vdash \varphi$. Suppose that $Prf_i(n, \ulcorner \varphi \urcorner)$. By the definition of Prf_i there are two possibilities. The first case when $Proof(n, \ulcorner \varphi \urcorner)$ is trivial. For the second case suppose that $n = \ulcorner t \urcorner$, $\varphi = B^{@}$ and $\mathcal{M} \models [\![t]\!]_i B$. In this case by definition of a model $\mathcal{M} \models B$ whence $\mathsf{PA} \vdash B^{@}$ by Lemma 5.5. □

Now we can define interpretation of operations on proofs. For every operation f of the language $\mathsf{LP}^2_!$, let $f^{@}$ be a total recursive function which satisfies the axiom $\mathrm{Ax}(f)$ for the standard predicate *Proof*. The following functions $cv_i(x)$ which convert Prf_i–proofs into *Proof*–proofs are total recursive:

$$cv_i(x) \rightleftharpoons \begin{cases} \mu y.\forall z(Prf_i(x,z) \leftrightarrow Proof(y,z)), & \text{if } x = \ulcorner t \urcorner \text{ for } t \in Tm(\mathcal{F}), \\ x, & \text{otherwise} \end{cases}$$

We define total recursive functions which interpret operations on proofs as follows:

$$\mathsf{app}_i(m,n) \rightleftharpoons \begin{cases} \ulcorner t \times_i s \urcorner, & \text{if } m = \ulcorner t \urcorner,\ n = \ulcorner s \urcorner \text{ for } t, s \in Tm_i, \\ \times_i^{@}(cv_i(m), cv_i(n)), & \text{otherwise} \end{cases}$$

$$\mathsf{choice}_i(m,n) \rightleftharpoons \begin{cases} \ulcorner t +_i s \urcorner, & \text{if } m = \ulcorner t \urcorner,\ n = \ulcorner s \urcorner \text{ for } t, s \in Tm_i, \\ +_i^{@}(cv_i(m), cv_i(n)), & \text{otherwise} \end{cases}$$

$$\mathsf{check}_i(m) \rightleftharpoons \begin{cases} \ulcorner !_i t \urcorner, & \text{if } m = \ulcorner t \urcorner \text{ for } t \in Tm_i, \\ !_i^{@}(m), & \text{otherwise} \end{cases}$$

$$\mathsf{check}^2_1(m) \rightleftharpoons \begin{cases} \ulcorner !^2_1 t \urcorner, & \text{if } m = \ulcorner t \urcorner \text{ for } t \in Tm_1, \\ !^{2\,@}_1(m), & \text{otherwise.} \end{cases}$$

Lemma 5.7 *Functions defined above satisfy conditions of Lemma 5.2.*

Proof For example, consider the axiom for $!^2_1$. We have to show that for every number m and arithmetical sentence φ the following formula is true:

$$Prf_1(m, \ulcorner \varphi \urcorner) \to Prf_2\big(\mathsf{check}^2_1(m), \ulcorner Prf_1(m, \ulcorner \varphi \urcorner) \urcorner\big).$$

Suppose that $Prf_1(m, \ulcorner \varphi \urcorner)$ is true. Then there are two possibilities: (1) $m = \ulcorner t \urcorner$ for some $t \in Tm_1$, $\varphi = \ulcorner B^{@} \urcorner$ for some $\mathsf{LP}^2_!$-formula B, and $\mathcal{M} \models [\![t]\!]_1 B$, or (2) $Proof(m, \ulcorner \varphi \urcorner)$. The second case is trivial. In the first case by the correctness

of $\mathsf{LP}^2_!$ with respect to symbolic models we conclude that $\mathcal{M} \models [\![!^2_1 t]\!]_2 [\![t]\!]_1 B$, hence $Prf_2(\ulcorner !^2_1 t \urcorner, \ulcorner ([\![t]\!]_1 B)^{@} \urcorner)$ by definition of Prf_2. □

We consider the interpretation $*$ which consists of the predicates Prf_i defined by (FPE), operations defined above and an evaluation $(\cdot)^*$ defined as follows: for every variable V

$$V^* \rightleftharpoons V^{@}.$$

By induction on the complexity of the term we can show that $t^* = t^{@}$ for every term t. Therefore, $F^* = F^{@}$ for every formula F. So, from Lemma 5.5 we deduce that for every formula C

$$\begin{array}{lcl} C \in \Gamma & \Rightarrow & \mathsf{PA} \vdash C^* \\ C \in \Delta & \Rightarrow & \mathsf{PA} \vdash \neg C^*. \end{array}$$

Since $A \in \Delta$, we conclude that $\mathsf{PA} \vdash \neg A^*$. □

Acknowledgements I would like to thank professor Sergei Artemov for his constant attention and encouragement. Also, I am thankful to Vladimir Krupski for numerous discussions and comments. This research was partially supported by RFBR, NWO and INTAS (grant No: 05–1000008–8144).

References

1. Artemov, S.: Uniform provability realization of intuitionistic logic, modality and λ-terms. Electron. Notes Theor. Comput. Sci. **23** (1999)
2. Artemov, S.: Explicit provability and constructive semantics. Bull. Symb. Log. **7**, 1–36 (2001)
3. Artemov, S.: Evidence–based common knowledge. Technical report TR–2004018, CUNY Ph.D. Program in Computer Science (2005)
4. Avron, A.: On Modal Systems having arithmetical interpretation. J. Symb. Log. **49**, 935–942 (1984)
5. Fitting, M.: The logic of proofs, semantically. Ann. Pure Appl. Log. **132**(1), 1–25 (2005)
6. Kuznets, R.: On the complexity of explicit modal logics. In: Computer Science Logic 2000. Lecture Notes in Computer Science, vol. 1862, pp. 371–383. Springer, Berlin (2000)
7. Mkrtychev, A.: Models for the logic of proofs. In: Logical Foundations of Computer Science '97, Yaroslavl'. Lecture Notes in Computer Science, vol. 1234, pp. 266–275. Springer, Berlin (1997)
8. Smoryński, C.: Self-reference and Modal Logic. Springer, New York (1985)
9. Troelstra, A.S., Schwichtenberg, H.: Basic Proof Theory. Cambridge University Press, Cambridge (1996)