Logic of Proofs: a Unified Semantics for Modality and λ-terms *

Sergei N. Artemov †

March, 1998

Abstract

In 1933 Gödel introduced a modal logic of provability ($\mathcal{S}4$) and left open the problem of a formal provability semantics for this logic. We give a complete solution to this problem. We find a sound and complete axiom system for the logic ($\mathcal{LP}$) with atomic propositions for explicit proofs "t is a proof of F", and construct an exact realization of $\mathcal{S}4$ in $\mathcal{LP}$. In $\mathcal{LP}$ the reflection principle is valid, which circumvents some of the restrictions imposed on the provability operator by Gödel's second incompleteness theorem.

$\mathcal{LP}$ formalizes the Kolmogorov calculus of problems and proves the Kolmogorov conjecture (1932) that intuitionistic logic coincides with the classical calculus of problems. $\mathcal{LP}$ has revealed the relationship between proofs and types, and subsumes the λ-calculus, modal λ-calculus and combinatory logic.

Introduction

The intended meaning of the intuitionistic logic was informally explained first in terms of operations on proofs due to Brouwer, Heyting and Kolmogorov (cf. [48],[49],[13]). This interpretation is widely known as the *BHK* semantics of intuitionistic logic. However, despite some similarities in the informal description of the functions assigned to the intuitionistic connective, the Brouwer-Heyting semantics and the Kolmogorov semantics have fundamentally different objectives. The Brouwer-Heyting semantics explaines intuitionistic logic in terms of an undefined notion of intuitionistic proof. The Kolmogorov interpretation of $\mathcal{I}nt$ as a calculus of problems [23], along with related papers by Gödel [17],[18] intended to interpret $\mathcal{I}nt$ on the basis of classical proofs, thus providing an independent definition of intuitionistic logic within the classical mathematics.

*Technical Report CFIS 98-06, Cornell University

†627 Rhodes Hall, Cornell University, Ithaca NY, 14853 U.S.A. email:artemov@hybrid.cornell.edu and Moscow University, Russia.

Kleene realizability [22], Medvedev finite problems [33] and its variants ([50], [51]) are regarded (cf. [48],[13],[50],[51]) as formalizations of Kolmogorov's calculus of problems. However, they give only necessary conditions for $\mathcal{I}nt$, each of them realizes some formulas not derivable in $\mathcal{I}nt$. A formalization of the BHK semantics suggested by Kreisel in [25] turned out to be based on an inconsistent theory (cf. [52], [41]). For more discussion on Kolmogorov's semantics of $\mathcal{I}nt$ see Sections 6 and 10.

In 1933 Gödel ([17]) defined $\mathcal{I}nt$ on the basis of the notion of *proof* in a classical mathematical system, where "proof" may be regarded as a special case of Kolmogorov's "problem solution". Namely, Gödel introduced the *logic of provability* (coinciding with the modal logic $\mathcal{S}4$) and constructed an embedding of $\mathcal{I}nt$ into $\mathcal{S}4$. $\mathcal{S}4$ has all axioms and rules of classical logic in the modal propositional language along with the axioms $\Box F \rightarrow F$, $\Box(F \rightarrow G) \rightarrow (\Box F \rightarrow \Box G)$, $\Box F \rightarrow \Box\Box F$, and the *necessitation rule* $F \vdash \Box F$. In [17] no formal provability semantics for $\mathcal{S}4$ was suggested. The straightforward interpretation of $\Box F$ as the arithmetical formula $Provable(F)$

"there exists a number x which is the code of a proof of F".

leads to logics of formal provability incompatible with $\mathcal{S}4$ (cf.[9],[10]).

> Let us consider, for example, the first order arithmetic $\mathcal{PA}$. If F is the boolean constant *false*, then the $\mathcal{S}4$-axiom $\Box F \rightarrow F$ becomes a statement *Consis* $\mathcal{PA}$, expressing the consistency of $\mathcal{PA}$. By necessitation, $\mathcal{S}4$ derives $\Box(\Box F \rightarrow F)$. The latter formula expresses the assertion that *Consis* $\mathcal{PA}$ is provable in $\mathcal{PA}$, which contradicts the Second Gödel Incompleteness Theorem.

The issue of a provability model for $\mathcal{S}4$ was studied by Gödel [18], Lemmon [29], Myhill [39],[40], Kripke [26], Montague [38], Mints [35], Kuznetsov & Muravitskii [28], Goldblatt [19], Boolos [9],[10] Shapiro [43],[44], Buss [12], Artemov [1], and many others. However, the problem of a formal provability semantics for $\mathcal{S}4$ has remained open.

A principal difficulty here is caused by the existential quantifier over proofs in $Provable(F)$. Indeed, the interpretation of the formula $\Box(\Box F \rightarrow F)$ is

'it is provable that "$Provable(F)$ implies F" '

Provability in $\mathcal{PA}$ can be characterized as "true in all models of $\mathcal{PA}$", including the nonstandard ones. In a given model of $\mathcal{PA}$ an element that instantiates the variable x from the existential quantifier for the code of a proof of F in $Provable(F)$ may be nonstandard. In such a case $Provable(F)$ is true in this model, but there is no "real" $\mathcal{PA}$-derivation behind such an x. So, $\mathcal{PA}$ is not able to conclude that F is true from $Provable(F)$ is true since the latter formula does not necessarily deliver a proof of F.

This consideration suggests replacing the provability formula $Provable(F)$ by the formula for proofs *Proof(t,F)* and the existential quantifier on proofs in the former by Skolem style operations on proofs in the latter. Such a conversion helps to avoid evaluation of proofs by nonstandard numbers. Some of these operations come from the proof of Gödel's second incompleteness theorem. Within that proof it was established that

$$\mathcal{PA} \vdash \mathit{Provable}(F \to G) \wedge \mathit{Provable}(F) \;\to\; \mathit{Provable}(G).$$

This formula is a "forgetful" version of the following theorem.

For some computable function $m(x, y)$

$$\mathcal{PA} \vdash \mathit{Proof}(s, F \to G) \wedge \mathit{Proof}(t, F) \;\to\; \mathit{Proof}(m(s,t), G).$$

A similar decoding can be done for another lemma from Gödel's second incompleteness theorem $\mathcal{PA} \vdash \mathit{Provable}(F) \to \mathit{Provable}(\mathit{Provable}(F))$.

For some computable function $c(x)$

$$\mathcal{PA} \vdash \mathit{Proof}(t, F) \to \mathit{Proof}(c(t), \mathit{Proof}(t, F)).$$

In his Lecture at Zilsel's, 1938, (published in 1995 in [18], see also [41]) Gödel sketched a constructive version of $\mathcal{S}4$ with the basic proposition "t is a proof of F" and operations similar to $m(x, y)$ and $c(x)$. This Gödel's suggestion suffices to justify the reflexivity principle along with the necessitation rule. However, the question about a complete set of axioms for a logic of proofs, as well as the question about its ability to realize the entire $\mathcal{S}4$ have remained unanswered. It turned out that Gödel's sketch of 1938 lacks the operation "+", without which a realization of $\mathcal{S}4$ cannot be completed (see Comment 6.6).

In this paper we implement Gödel's suggestion of 1938 and find the *Logic of Proofs* ($\mathcal{LP}$)[1] in the propositional language with an extra basic proposition $t : F$ for "t is a proof of F". We find axiom systems for $\mathcal{LP}$ in Hilbert, Gentzen and natural deduction format, establish its soundness and completeness with respect to the standard provability semantics. We demonstrate that any invariant operation on proofs which can be specified by a propositional condition is represented by an $\mathcal{LP}$-term. By assigning explicit proof terms to the modalities in every $\mathcal{S}4$-derivation, we establish that $\mathcal{LP}$ realizes the entire $\mathcal{S}4$. This yields a positive solution to the problem of finding the intended provability semantics for $\mathcal{S}4$ which in turn proves Kolmogorov's conjecture of 1932 that intuitionistic logic $\mathcal{I}nt$ is nothing but the calculus of problems for systems based on classical logic.

[1] The Logic of Proofs $\mathcal{LP}$ was found by the author independently of Gödel's paper [18]. The first presentations of $\mathcal{LP}$ took place at the author's talks at the conferences in Münster and Amsterdam in 1994. Preliminary versions of $\mathcal{LP}$ along with the completeness theorem, realization of $\mathcal{S}4$ and λ-calculi in $\mathcal{LP}$ appeared in Technical Reports [4] and [5], cf. also a survey [21]. Note that despite its title the paper [3] does not introduce $\mathcal{LP}$.

In $\mathcal{LP}$ the notion *term t is a proof of F* and *term t has type F* are subsumed by the basic $\mathcal{LP}$ proposition $t:F$. We demonstrate that under these interpretations $\mathcal{LP}$ naturally encompasses λ-calculi corresponding to intuitionistic and modal logics, and combinatory logic. In addition, $\mathcal{LP}$ is strictly more expressive because it admits arbitrary combinations of ":" and propositional connectives. By treating ":" uniformly, $\mathcal{LP}$ unifies the semantics of modality, combinatory, and λ-terms. All these objects are realized as proof terms in $\mathcal{LP}$.

Among the related works there is [15], where Gabbay's Labelled Deductive Systems may serve as a natural framework for $\mathcal{LP}$. The Logic of Proofs may also be regarded as a basic epistemic logic with explicit justifications; a problem of finding such systems was raised by van Benthem in [7]. Intuitionistic Type Theory by Martin-Löf [31], [32] also makes use of the format $t{:}F$ with its informal provability reading.

1 Logic of Proofs

1.1 Definition. The language of Logic of Proofs ($\mathcal{LP}$) contains

the usual language of classical propositional logic
proof variables $x_0,\ldots,x_n,\ldots$, proof constants $a_0,\ldots,a_n,\ldots$
function symbols: monadic !, binary $\cdot$ and $+$
operator symbol of the type "*term : formula*".

We will use $a,b,c,\ldots$ for proof constants, $u,v,w,x,y,z,\ldots$ for proof variables, i,j,k,l,m,n for natural numbers. Terms are defined by the grammar

$$p ::= x_i \mid a_i \mid\, !p \mid p_1\cdot p_2 \mid p_1+p_2$$

We call these terms *proof polynomials* and denote them by $p,r,s,t\ldots$. By analogy we refer to constants as coefficients. Constants correspond to proofs of a finite fixed set of propositional schemas. We will also omit $\cdot$ whenever it is safe. We also assume that $(a\cdot b\cdot c)$, $(a\cdot b\cdot c\cdot d)$, *etc.* should be read as $((a\cdot b)\cdot c)$, $(((a\cdot b)\cdot c)\cdot d)$, *etc.*

Using t to stand for any term and S for any propositional letter, the formulas are defined by the grammar

$$\sigma ::= S \mid \sigma_1\!\rightarrow\!\sigma_2 \mid \sigma_1\!\wedge\!\sigma_2 \mid \sigma_1\!\vee\!\sigma_2 \mid \neg\sigma \mid t{:}\sigma$$

We will use A,B,C,F,G,H,X,Y,Z for the formulas in this language, and $\Gamma,\Delta,\ldots$ for the finite sets (also finite multisets, or finite lists) of formulas unless otherwise explicitly stated. We will also use $\vec{x},\vec{y},\vec{z},\ldots$ and $\vec{p},\vec{r},\vec{s},\ldots$ for vectors of proof variables and proof polynomials respectively. If $\vec{s}=(s_1,\ldots,s_n)$ and $\Gamma=(F_1,\ldots,F_n)$, then $\vec{s}{:}\Gamma$ denotes $(s_1:F_1,\ldots,s_n:F_n)$, $\bigvee\Gamma = F_1\vee\ldots\vee F_n$, $\bigwedge\Gamma = F_1\wedge\ldots\wedge F_n$. We assume the following precedences from highest to lowest: $!,\cdot,+,:,\neg,\wedge,\vee,\rightarrow$. We will use the symbol $=$ in different situations, both formal and informal. Symbol $\equiv$ denotes syntactical identity, $\ulcorner E\urcorner$ is the Gödel number of E.

The intended semantics for $p:F$ is "p is a proof of F", which will be formalized in the next section. Note that proof systems which provide a formal semantics for $p:F$ are *multi-conclusion* ones, i.e. p may be a proof of several different F's (see Comment 1.7). Within this paper we will call multi-conclusion proof systems *non-deterministic.*

1.2 Definition. The system $\mathcal{LP}_0$. Axioms:

A0. Finite set of axiom schemes of classical propositional logic in the language of $\mathcal{LP}$

A1. $t:F \to F$ *"verification"*

A2. $t:(F \to G) \to (s:F \to (t\cdot s):G)$ *"application"*

A3. $t:F \to\ !t:(t:F)$ *"proof checker"*

A4. $s:F \to (s+t):F, \quad t:F \to (s+t):F$ *"choice"*

Rule of inference:

R1. $$\frac{F \to G \qquad F}{G}$$ *"modus ponens".*

The system $\mathcal{LP}$ is $\mathcal{LP}_0$ plus the rule

R2. $$\frac{}{c:\mathbf{A}}\,,$$

if $\mathbf{A}$ *is an axiom A0 – A4, and c a proof constant* *"axiom necessitation".*

A *Constant Specification* ($\mathcal{CS}$) is a finite set of formulas $c_1 : A_1, \dots, c_n : A_n$ such that c_i is a constant, and F_i an axiom *A0 – A4*. Each derivation in $\mathcal{LP}$ naturally generates the $\mathcal{CS}$ consisting of all formulas introduced in this derivation by the *necessitation* rule.

1.3 Comment. Proof constants in $\mathcal{LP}$ stand for proofs of "simple facts", namely propositional axioms and axioms *A1 – A4*. In a way the proof constants resemble atomic constant terms (*combinators*) of typed combinatory logic (cf. [49]). A constant c_1 specified as $c_1 : (A \to (B \to A))$ can be identified with the combinator $\mathbf{k}^{A,B}$ of the type $A \to (B \to A)$. A constant c_2 such that $c_2 : [(A \to (B \to C)) \to ((A \to B) \to (A \to C))]$ corresponds to the combinator $\mathbf{s}^{A,B,C}$ of the type $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$. The proof variables may be regarded as term variables of combinatory logic, the operation "$\cdot$" as the application of terms. In general an $\mathcal{LP}$-formula $t : F$ can be read as a combinatory term t of the type F. Typed combinatory logic $\mathbf{CL}_{\to}$ thus corresponds to a fragment of $\mathcal{LP}$ consisting only of formulas of the sort $t:F$ where t contains no operations other than "$\cdot$" and F is a formula built from the propositional letters by "$\to$" only.

There is no restriction on the choice of a constant c in *R2* within a given derivation. In particular, *R2* allows to introduce a formula $c:A(c)$, or to specify a constant several times as a proof of different axioms from *A0* – *A4*. One might restrict $\mathcal{LP}$ to injective constant specifications, i.e. only allowing each constant to serve as a proof of a single axiom $\mathbf{A}$ within a given derivation (although allowing constructions $c{:}\mathbf{A}(c)$, as before). Such a restriction would not change the ability of $\mathcal{LP}$ to emulate classical modal logic, or the functional and arithmetical completeness theorems for $\mathcal{LP}$ (below), though it will provoke an excessive renaming of the constants.

Both $\mathcal{LP}_0$ and $\mathcal{LP}$ enjoy the deduction theorem

$$\Gamma, A \vdash B \quad \Rightarrow \quad \Gamma \vdash A \to B,$$

and the substitution lemma: *If* $\Gamma(x,P) \vdash B(x,P)$ *for a propositional variable* P *and a proof variable* x*, then for any proof polynomial* t *and any formula* F

$$\Gamma(x/t, P/F) \vdash B(x/t, P/F).$$

For a given constant specification $\mathcal{CS}$ under $\mathcal{LP}(\mathcal{CS})$ we mean $\mathcal{LP}_0$ plus $\mathcal{CS}$. Obviously,

F is derivable in $\mathcal{LP}$ *with a constant specification* $\mathcal{CS} \;\Leftrightarrow\; \mathcal{LP}(\mathcal{CS}) \vdash F \;\Leftrightarrow\; \mathcal{LP}_0 \vdash \bigwedge \mathcal{CS} \to F$.

1.4 Lemma. (Lifting Lemma) *Given a derivation* $\mathcal{D}$ *of the type*

$$\vec{s}{:}\Gamma, \Delta \vdash_{\mathcal{LP}} F,$$

one can construct a proof polynomial $t(\vec{x},\vec{y})$ *such that*

$$\vec{s}{:}\Gamma, \vec{y}{:}\Delta \vdash_{\mathcal{LP}} t(\vec{s},\vec{y}){:}F.$$

Proof. By induction on the derivation $\vec{s}{:}\Gamma, \Delta \vdash F$. If $F = s_i{:}G_i \in \vec{s}{:}\Gamma$, then put $t := !s_i$ and use *A3*. If $F = D_j \in \Delta$, then put $t := y_j$. If F is an axiom *A0* – *A4*, then pick a fresh proof constant c and put $t := c$; by *R2*, $F \vdash c{:}F$. Let F be introduced by *modus ponens* from $G \to F$ and G. Then, by the induction hypothesis, there are proof polynomials $u(\vec{s},\vec{y})$ and $v(\vec{s},\vec{y})$ such that $u{:}(G \to F)$ and $v{:}G$ are both derivable in $\mathcal{LP}$ from $\vec{s}{:}\Gamma, \vec{y}{:}\Delta$. By *A2*, $\vec{s}{:}\Gamma, \vec{y}{:}\Delta \vdash (uv){:}F$, and we put $t := uv$. If F is introduced by *R2*, then $F = c{:}A$ for some axiom A. Use the same *R2* followed by *A3*: $c{:}A \to !c{:}c{:}A$, to get $\vec{s}{:}\Gamma, \vec{y}{:}\Delta \vdash !c{:}F$, and put $t := !c$.
◀

Note that if $\Delta \vdash_{\mathcal{LP}_0} F$, then one can construct $t(\vec{y})$ which is a product of proof constants and variables from $\vec{y}$ such that $\vec{y}{:}\Delta \vdash_{\mathcal{LP}_0} t(\vec{y}){:}F$. It is easy to see from the proof that the lifting polynomial $t(\vec{x},\vec{y})$ is nothing but a blueprint of $\mathcal{D}$. Thus $\mathcal{LP}$ accommodates its own proofs as terms. The necessitation rule

$$\vdash F \quad \Rightarrow \ \vdash p{:}F \ \ \textit{for some proof polynomial } p,$$

where p is a blueprint of the proof $\vdash F$ is a special case of Lifting. As we can see in Section 6 $\mathcal{LP}$ suffices to emulate all $\mathcal{S}4$-derivations.

1.5 Example. $\mathcal{S}4 \vdash (\Box A \wedge \Box B) \rightarrow \Box(A \wedge B)$

In $\mathcal{LP}$ the corresponding derivation is

1. $A \rightarrow (B \rightarrow A \wedge B)$, by *A0*,
2. $c{:}(A \rightarrow (B \rightarrow A \wedge B))$, from 1, by *R2*,
3. $x{:}A \rightarrow (c \cdot x){:}(B \rightarrow A \wedge B)$, from 2, by *A2*,
4. $x{:}A \rightarrow (y{:}B \rightarrow (c \cdot x \cdot y){:}(A \wedge B))$, from 3, by *A2* and propositional logic,
5. $x{:}A \wedge y{:}B \rightarrow (c \cdot x \cdot y){:}(A \wedge B))$, from 4, by propositional logic.

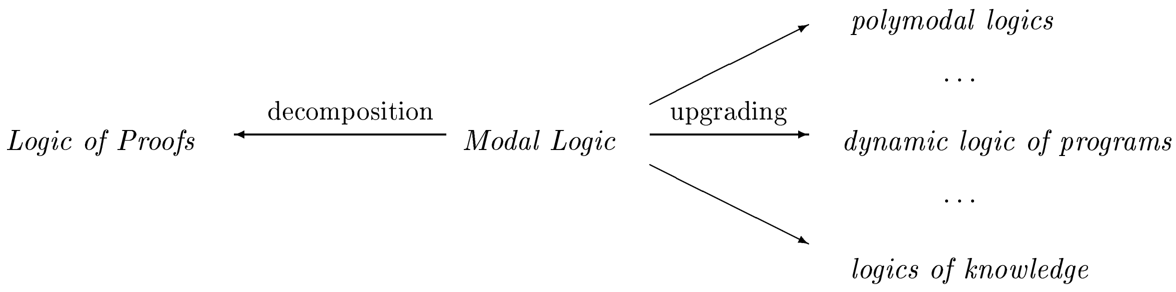
1.6 Example. $\mathcal{S}4 \vdash (\Box A \vee \Box B) \rightarrow \Box(A \vee B)$.

In $\mathcal{LP}$ the corresponding derivation is

1. $A \rightarrow A \vee B, \quad B \rightarrow A \vee B$, by *A0*,
2. $a{:}(A \rightarrow A \vee B), \quad b{:}(B \rightarrow A \vee B)$, by *R2*,
3. $x{:}A \rightarrow (a \cdot x){:}(A \vee B), \quad y{:}B \rightarrow (b \cdot y){:}(A \vee B)$, from 2 by *A2*
4. $(a \cdot x){:}(A \vee B) \rightarrow (a \cdot x + b \cdot y){:}(A \vee B), \quad (b \cdot y){:}(A \vee B) \rightarrow (a \cdot x + b \cdot y){:}(A \vee B)$, by *A4*
5. $(x{:}A \vee y{:}B) \rightarrow (a \cdot x + b \cdot y){:}(A \vee B)$, from 4, by propositional logic.

1.7 Comment. Operations "$\cdot$" and "!" are present for deterministic as well as on non-deterministic proof systems. In turn, "+" is an operation for non-deterministic proof systems only. Indeed, by *A4* we have $s : F \wedge t : G \rightarrow \ (s+t) : F \wedge (s+t) : G$, thus $s+t$ proves different formulas. The differences between deterministic and non-deterministic proof systems are mostly cosmetic. Usual proof systems (Hilbert or Gentzen style) may be considered as deterministic, e.g. a proof derives only the end formula (sequent) of a proof tree. On the other hand, the same systems may be regarded as non-deterministic by assuming that a proof derives all formulas assigned to the nodes of the proof tree. The logic of strictly deterministic proof systems was studied in [2], [3],[27], where it meets a complete axiomatization (system $\mathcal{FLP}$). $\mathcal{FLP}$ is not compatible with any modal logic (cf. Comment 6.6) and thus is not directly relevant to the problem of finding an intended semantics for the modal logic of provability. Therefore, provability as a modal operator corresponds to non-deterministic proof systems.

No single operator "$t:$" in $\mathcal{LP}$ is a normal modality since none of them satisfies the property $t:(P \to Q) \to (t:P \to t:Q)$. This makes $\mathcal{LP}$ essentially different from polymodal logics, e.g. the dynamic logic of programs ([24]), where the modality is upgraded by some additional features. In the Logic of Proofs the modality is decomposed into a family of proof polynomials (see Section 6)



2 Standard provability interpretation of $\mathcal{LP}$

The Logic of Proofs is meant to play for a notion of proof a role similar to that played by the boolean propositional logic for the notion of statement. It is shown in Sections 2 and 4 of this paper that $\mathcal{LP}$ enjoys the soundness/completeness property:

$$\mathcal{LP} \vdash F \qquad \Leftrightarrow \qquad F \textit{ is true under any interpretation .}$$

Any system of proofs with a proof checker operation capable of internalizing its own proofs as terms (cf. [46]) may be in the scope of $\mathcal{LP}$. In particular, any proof system for the first order Peano Arithmetic $\mathcal{PA}$ (cf. [9], [10], [34], [47]) provides a model for $\mathcal{LP}$ with Gödel numbers of proofs being a instrument of internalizing proofs as terms. The soundness ($\Rightarrow$) does not necessarily refer to the arithmetical models. However, $\mathcal{PA}$ is convenient for establishing the completeness ($\Leftarrow$) of $\mathcal{LP}$: given $\mathcal{LP} \not\vdash F$ one can always find a proof system for $\mathcal{PA}$ along with an evaluation of variables in F which makes F false (Theorem 5.1).

Within Sections 2 and 5 of this paper under Δ_1 and Σ_1 we mean the corresponding classes of arithmetical predicates. We will use φ, ψ to denote arithmetical formulas, f, g, h to denote arithmetical terms, i, j, k, l, n to denote natural numbers unless stated otherwise. We will use the letters u, v, w, x, y, z to denote individual variables in arithmetic and hope that a reader is able to distinguish them from the proof variables. If n is a natural number, then $\overline{n}$ will denote a numeral corresponding to n, i.e. a standard arithmetical term $0'''^{\cdots}$ where $'$ is a successor functional symbol and the number of $'$'s equals n. We will use the simplified notation n for a numeral $\overline{n}$ when it is safe.

2.1 Definition. We assume that $\mathcal{PA}$ contains terms for all primitive recursive functions (cf. [47]), called *primitive recursive terms.* Formulas of the form $f(\vec{x})=0$ where $f(\vec{x})$ is a primitive recursive term are *standard primitive recursive formulas. A standard* Σ_1 *formula* is a formula $\exists x\varphi(x,\vec{y})$ where $\varphi(x,\vec{y})$ is a standard primitive recursive formula. An arithmetical formula φ is *provably* Σ_1 if it is provably equivalent in $\mathcal{PA}$ to a standard Σ_1 formula; φ is *provably* Δ_1 iff both φ and $\neg\varphi$ are provably Σ_1.

2.2 Definition. A *proof predicate* is a provably Δ_1-formula $\mathit{Prf}(x,y)$ such that for every arithmetical sentence φ

$$\mathcal{PA}\vdash\varphi \quad\Leftrightarrow\quad \text{for some } n\in\omega \quad \mathit{Prf}(n,\ulcorner\varphi\urcorner) \text{ holds}^2.$$

A proof predicate $\mathit{Prf}(x,y)$ is *normal* if the following conditions are fulfilled:

1) (*finiteness of proofs*) For every proof k the set $T(k)=\{l \mid \mathit{Prf}(k,l)\}$ is finite. The function from k to the canonical number of $T(k)$ is computable.

2) (*conjoinability of proofs*) For any natural numbers k and l there is a natural number n such that

$$T(k)\cup T(l)\subseteq T(n).$$

2.3 Comment. Every non-deterministic normal proof predicate can be transformed into a deterministic one by changing from

$$\text{“}p \text{ proves } F_1,\ldots,F_n\text{”} \qquad \text{to} \qquad \text{“}(p,i) \text{ proves } F_i,\ i=1,\ldots n\text{”}.$$

In turn, every deterministic proof predicate may be regarded as non-deterministic by reading

$$\text{“}p \text{ proves } F_1\wedge\ldots\wedge F_n\text{”} \qquad \text{as} \qquad \text{“}p \text{ proves each of } F_i,\ i=1,\ldots n\text{”}.$$

2.4 Lemma. *For every normal proof predicate* Prf *there are computable functions* $m(x,y)$, $a(x,y)$, $c(x)$ *such that for all arithmetical formulas* φ,ψ *and all natural numbers* k,n *the following formulas are valid:*

[2] We have omitted bars over numerals for natural numbers $n,\ulcorner\varphi\urcorner$ in the formula Prf and will do it throughout this paper.

$$Prf(k,\ulcorner\varphi\!\to\!\psi\urcorner)\wedge Prf(n,\ulcorner\varphi\urcorner)\to Prf(m(k,n),\ulcorner\psi\urcorner)$$
$$Prf(k,\ulcorner\varphi\urcorner)\to Prf(a(k,n),\ulcorner\varphi\urcorner),\qquad Prf(n,\ulcorner\varphi\urcorner)\to Prf(a(k,n),\ulcorner\varphi\urcorner)$$
$$Prf(k,\ulcorner\varphi\urcorner)\to Prf(c(k),\ulcorner Prf(k,\ulcorner\varphi\urcorner)\urcorner).$$

Proof. The following function can be taken as m:

> *Given k,n put $m(k,n)=\mu z$"$Prf(z,\ulcorner\psi\urcorner)$ for all ψ such that there are $\ulcorner\varphi\!\to\!\psi\urcorner\in T(k)$ and $\ulcorner\varphi\urcorner\in T(n)$" .*

Likewise, for a one could take

> *Given k,n put $a(k,n)=\mu z$ "$T(k)\cup T(n)\subseteq T(z)$".*

Finally, c may be put

> *Given k put $c(k)=\mu z$"$Prf(z,\ulcorner Prf(k,\ulcorner\varphi\urcorner)\urcorner)$ for all $\ulcorner\varphi\urcorner\in T(k)$". Such z always exists. Indeed, $Prf(k,\ulcorner\varphi\urcorner)$ are true Δ_1 formulas for every $\ulcorner\varphi\urcorner\in T(k)$, therefore they all are provable in $\mathcal{PA}$. Use conjoinability to find a uniform proof of all of them.*

◀

Note, that the natural arithmetical proof predicate *PROOF(x,y)*

> "*x is the code of a derivation* **containing** *a formula with the code y*".

is an example of a normal proof predicate.

2.5 Definition. An arithmetical *interpretation* $*$ of the $\mathcal{LP}$-language has the following parameters:

- a normal proof predicate Prf with the functions $m(x,y)$, $a(x,y)$, $c(x)$ as in Lemma 2.4,
- an evaluation of propositional letters by sentences of arithmetic, and
- an evaluation of proof variables and proof constants by natural numbers.

Let $*$ commute with boolean connectives,

$$(t\cdot s)^*=m(t^*,s^*),\quad (t+s)^*=a(t^*,s^*),\quad (!t)^*=c(t^*),$$
$$(t\!:\!F)^*=Prf(\overline{t^*},\overline{\ulcorner F^*\urcorner}).$$

Under an interpretation $*$ a proof polynomial t becomes a natural number t^*, an $\mathcal{LP}$-formula F becomes an arithmetical sentence F^*. A formula $(t{:}F)^*$ is always provably Δ_1. Note, that $\mathcal{PA}$ (as well as any theory containing certain finite number of arithmetical axioms, e.g. Robinson's arithmetic) is able to derive any true Δ_1 formula, and thus to derive a negation of any false Δ_1 formula (cf. [34]). For a set X of $\mathcal{LP}$-formulas under X^* we mean the set of all F^*'s such that $F \in X$. Given a constant specification $\mathcal{CS}$, an arithmetical interpretation $*$ is a *$\mathcal{CS}$-interpretation* if all formulas from $\mathcal{CS}^*$ are true (equivalently, are provable in $\mathcal{PA}$). An $\mathcal{LP}$-formula F is *valid* (with respect to the arithmetical semantics) if the arithmetical formula F^* is true under all interpretations $*$. F is *$\mathcal{CS}$-valid* if F^* is true under all $\mathcal{CS}$-interpretations $*$.

In Section 3 we will need the definition of $*$ to be extended to the language with $\Box$. Then we assume that $(\Box F)^* = \exists y Prf(y, \ulcorner F^* \urcorner)$.

2.6 Theorem. (Arithmetical soundness of $\mathcal{LP}_0$)

1. *If $\mathcal{LP}_0 \vdash F$ then F is valid.*
2. *If $\mathcal{LP}_0 \vdash F$ then $\mathcal{PA} \vdash F^*$ for any interpretation $*$.*

Proof. A straightforward induction on the derivation in $\mathcal{LP}_0$. Let us check 2. for the axiom $t : F \to F$. Under an interpretation $*$ $(t{:}F \to F)^* \equiv Prf(t^*, \ulcorner F^* \urcorner) \to F^*$. Consider two possibilities. Either $Prf(t^*, \ulcorner F^* \urcorner)$ is true, in which case t^* is indeed a proof of F^*, thus $\mathcal{PA} \vdash F^*$ and $\mathcal{PA} \vdash (t{:}F \to F)^*$. Otherwise $Prf(t^*, \ulcorner F^* \urcorner)$ is false, in which case being a false Δ_1 formula it is refutable in $\mathcal{PA}$, i.e. $\mathcal{PA} \vdash \neg Prf(t^*, \ulcorner F^* \urcorner)$ and again $\mathcal{PA} \vdash (t{:}F \to F)^*$.
◀

2.7 Corollary. (Arithmetical soundness of $\mathcal{LP}$)

If $\mathcal{LP}(\mathcal{CS}) \vdash F$, then F is $\mathcal{CS}$-valid.

2.8 Comment. The standard provability semantics for $\mathcal{LP}$ above may be characterized as a *call-by-value* semantics, since the evaluation F^* of a given $\mathcal{LP}$-formula F depends upon the value of participating functions. A *call-by-name* provability semantics for $\mathcal{LP}$ was introduced in [4] and then used in [27], [45]. In the latter semantics F^* depends upon the particular programs for the functions participating in $*$.

In order to define the *call-by-name* provability semantics for $\mathcal{LP}$ we assume that $\mathcal{PA}$ has the standard set of tools to introduce so-called ι-terms. We use a new functional symbol $\iota z \varphi(z)$ for each arithmetical formula $\varphi(z)$ and assume that ι-terms could be eliminated by using the small scope convention (cf. [14]). The term $\iota z \varphi(z)$ is called *computable* if $\varphi(z)$ is provably

Σ_1. A computable term represents some computable function, every computable function is represented by a computable term (cf. [34]).

The term $\iota z\varphi(z)$ is *provably total*, if $\mathcal{PA} \vdash \exists_1 z\varphi(z)$, i.e. $\mathcal{PA}$ proves that there exists a unique z such that $\varphi(z)$. In particular, every arithmetical term in a narrow sense, i.e. a term built from 0 by $', +, \times$ may be regarded as a provably total computable term. *A closed computable term* is a computable provably total term $\iota z\varphi(z)$ such that $\varphi(z)$ contains no free variables other than z.

The set of computable terms is closed under substitution. The result of substituting a closed computable term into a Δ_1 formula is again a Δ_1 formula. Closed computable terms stand for all computable "names" for natural numbers. There is an algorithm which for any closed computable term f calculates its *value*, i.e. the numeral $\overline{n}$ such that $\mathcal{PA} \vdash f = \overline{n}$.

An analog of Lemma 2.4 can be established that for every normal proof predicate Prf there are computable terms $m(x,y)$, $a(x,y)$, $c(x)$ such that if f, g are closed computable terms, then $m(f,g)$, $a(f,g)$, $c(\ulcorner f \urcorner)$ are again closed computable terms and for all arithmetical formulas φ, ψ the following formulas are valid:

$$Prf(f, \ulcorner \varphi \to \psi \urcorner) \wedge Prf(g, \ulcorner \varphi \urcorner) \to Prf(m(f,g), \ulcorner \psi \urcorner)$$

$$Prf(f, \ulcorner \varphi \urcorner) \to Prf(a(f,g), \ulcorner \varphi \urcorner), \qquad Prf(g, \ulcorner \varphi \urcorner) \to Prf(a(f,g), \ulcorner \varphi \urcorner)$$

$$Prf(f, \ulcorner \varphi \urcorner) \to Prf(c(\ulcorner f \urcorner), \ulcorner Prf(f, \ulcorner \varphi \urcorner) \urcorner).$$

Note that $c(\ulcorner f \urcorner)$ depends on the code of f rather than on the value of f. In particular, it may be the case that the values of f and g are equal, but $c(\ulcorner f \urcorner) \neq c(\ulcorner g \urcorner)$.

An interpretation $*$ is defined by the choice of a normal proof predicate Prf with the terms $m(x,y)$, $a(x,y)$, $c(x)$, an evaluation of propositional letters by sentences of arithmetic, and an evaluation of proof variables and proof constants by closed computable terms. As before $*$ commutes with boolean connectives, $(t \cdot s)^* = m(t^*, s^*)$, $(t+s)^* = a(t^*, s^*)$, $(!t)^* = c(\ulcorner t^* \urcorner)$, $(t\!:\!F)^* = Prf(t^*, \ulcorner F^* \urcorner)$. Note that unlike the standard call-by-value interpretation above in this case we plug not the numeral of the value of f for the variable x in $Prf(x,y)$ but a term f itself. Under any interpretation $*$ a proof polynomial t becomes a closed computable term t^*, an $\mathcal{LP}$-formula F becomes an arithmetical sentence F^*. A formula $(t\!:\!F)^*$ is always provably Δ_1.

As it was established in [4] $\mathcal{LP}$ is sound and complete with respect to this call-by-name provability interpretation. In fact the soundness in this case can be shown by an easy modification of the soundness proof for the standard call-by-name interpretation above. In Section 5 (Comment 5.13) we will discuss how to establish the completeness of $\mathcal{LP}$ in the call-by-name case.

3 Finite basis theorem

In this section we show functional completeness for the system of proof polynomials in $\mathcal{LP}$, as another justification for the basic set of operations $\cdot, !, +$. This result is not pertinent to the rest of the paper so the reader may choose to skip this section.

Operations on proofs invariant with respect to the choice of a proof system naturally arise from the notion of admissible rule in arithmetic.

3.1 Definition. Let $\mathcal{L}$ be a logical language (propositional, modal, first order) with a class of its arithmetical interpretations such that for any interpretation $*$ and any formula F from $\mathcal{L}$ an arithmetical formula F^* is defined. A *deterministic admissible rule in $\mathcal{PA}$ over $\mathcal{L}$* is a figure

$$\frac{\vdash C_1, \ldots, \vdash C_n}{\vdash G} \qquad (1)$$

where $C_1, \ldots, C_n, G$ are $\mathcal{L}$-formulas such that for every interpretation $*$, G^* is provable in $\mathcal{PA}$ whenever ${C_1}^*, \ldots, {C_n}^*$ are. A *(non-deterministic) admissible rule in $\mathcal{PA}$* is a rule of the kind

$$\frac{\vdash C_1^1, \ldots, \vdash C_1^{k_1} \quad or \quad \ldots \quad or \quad \vdash C_n^1, \ldots, \vdash C_n^{k_n}}{\vdash G}\,. \qquad (2)$$

The meaning of (2) is that for every interpretation $*$ G^* is derivable in $\mathcal{PA}$ whenever for some i, $0 \leq i \leq n$, all $(C_i^1)^*, \ldots, (C_i^{k_i})^*$ are derivable in $\mathcal{PA}$. Every such rule may be regarded as an implicit specification of a proof y of G as a function of proofs x_i^j's of C_i^j's. An *invariant operation on proofs* is a valid formula

$$\bigvee_i \bigwedge_j x_i^j : C_i^j \rightarrow \Box G, \qquad (3)$$

where all C_i^j's and G are formulas in the language of $\mathcal{LP}$. Since $(\Box G)^* = \exists y Prf(y, \ulcorner G^* \urcorner)$, formula (3) is a straight formalization of (2) where the existential quantifier over proofs in $(\Box G)^*$ is an implicit specification of a proof y of G^* as a function of x_i^j's.

3.2 Example. Some examples of invariant operations on proofs:

i) $x : (F \rightarrow G) \wedge y : F \rightarrow \Box G$,
ii) $x : F \rightarrow \Box x : F$,
iii) $x : F \wedge y : G \rightarrow \Box(F \wedge G)$,
iv) $x : F \vee y : G \rightarrow \Box(F \vee G)$.

For each of these examples there is a proof polynomial p realizing the operator $\Box$ in such a way that instantiating p inside $\Box$ gives a formula derivable in $\mathcal{LP}$:

i) $\mathcal{LP} \vdash x\!:\!(F \to G) \wedge y\!:\!F \to (x \cdot y)\!:\!G$,
ii) $\mathcal{LP} \vdash x\!:\!F \to !x\!:\!x\!:\!F$,
iii) $\mathcal{LP} \vdash x\!:\!F \wedge y\!:\!G \to t(x,y)\!:\!(F \wedge G)$, (Example 1.5)
iv) $\mathcal{LP} \vdash x\!:\!F \vee y\!:\!G \to (ax+by)\!:\!(F \vee G)$. (Example 1.6)

The following theorem demonstrates that proof existence in any invariant operation on proofs can be instantiated with a specific proof polynomial.

3.3 Theorem. *For any invariant operation on proofs*

$$\bigvee_i \bigwedge_j x_i^j\!:\!C_i^j \to \Box G$$

one can construct a proof polynomial $p(\vec{x})$ *such that*

$$\mathcal{LP} \vdash \bigvee_i \bigwedge_j x_i^j\!:\!C_i^j \to p(\vec{x})\!:\!G.$$

Proof. Let (2) be an invariant operation on proofs, and let us denote

$$\bigvee_i \bigwedge_j x_i^j\!:\!C_i^j$$

by C. Since $\Box G \to G$ is valid, the formula $C \to G$ is also valid. By the completeness theorem for $\mathcal{LP}$ (see Section 5), $\mathcal{LP} \vdash C \to G$. By Lifting Lemma 1.4 and Deduction, for a fresh variable y there is a proof polynomial $t(y)$ such that

$$\mathcal{LP} \vdash y\!:\!C \to t(y)\!:\!G. \qquad (4)$$

3.4 Lemma. *For any formulas* A, B *one can construct a proof polynomial* $u(x,y)$ *such that*

$$\mathcal{LP} \vdash x\!:\!A \wedge y\!:\!B \to u(x,y)\!:\!(x\!:\!A \wedge y\!:\!B).$$

Indeed, $\mathcal{LP} \vdash x\!:\!A \to !x\!:\!x\!:\!A$, $\mathcal{LP} \vdash y\!:\!B \to !y\!:\!y\!:\!B$. By 1.5,

$$\mathcal{LP} \vdash !x\!:\!x\!:\!A \wedge !y\!:\!y\!:\!B \to t(!x,!y)\!:\!(x\!:\!A \wedge y\!:\!B)$$

for an appropriate t, thus

$$\mathcal{LP} \vdash x\!:\!A \wedge y\!:\!B \to t(!x,!y)\!:\!(x\!:\!A \wedge y\!:\!B).$$

3.5 Lemma. *For all formulas A, B there exists a proof polynomial $v(x,y)$ such that*

$$\mathcal{LP} \vdash x\!:\!A \vee y\!:\!B \to v(x,y)\!:\!(x\!:\!A \vee y\!:\!B).$$

Similar to the proof of 3.4, using 1.6.

3.6 Lemma. *One can construct a proof polynomial $s(\vec{x})$ such that*

$$\mathcal{LP} \vdash C \to s(\vec{x})\!:\!C.$$

Proof. This lemma is proved by a straightforward induction on the formula

$$C = \bigvee_i \bigwedge_j x_i^j\!:\!C_i^j$$

with the use of 3.4 and 3.5.
◀

From (4), we have $\mathcal{LP} \vdash s : C \to t(s) : G$, and thus from 3.6 we get the desired result $\mathcal{LP} \vdash C \to t(s)\!:\!G$. This concludes the proof of 3.3.
◀

3.7 Comment. Whereas (2) requires all three proof connectives, the realization of deterministic operations of the form (1) does not require "+".

3.8 Comment. We demonstrated that every invariant propositional operation on proofs can be represented by a corresponding proof polynomial. Thus the basic operations $\cdot, !, +$ to some extent play for the proofs a role similar to that played by the boolean connectives in classical logic where every truth function (total function from $\{0,1\}^n$ to $\{0,1\}$) is represented by a boolean polynomial.

4 A sequent formulation of Logic of Proofs

Throughout Sections 4 - 7 by *sequent* we mean a pair $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite sets of $\mathcal{LP}$-formulas. By Γ, F we understand $\Gamma \cup \{F\}$.

Axioms of $\mathcal{LPG}_0$ are sequents of the form $F \Rightarrow F$ and $\perp \Rightarrow$. Along with the usual Gentzen sequent rules of classical propositional logic, including the cut rule (cf. the system **G1c** from [49]), the system $\mathcal{LPG}_0$ contains the rules

$$\frac{A, \Gamma \Rightarrow \Delta}{t\!:\!A, \Gamma \Rightarrow \Delta}\ (:\ \Rightarrow) \qquad\qquad \frac{\Gamma \Rightarrow \Delta, t\!:\!A}{\Gamma \Rightarrow \Delta, !t\!:\!t\!:\!A}\ (\Rightarrow !)$$

$$\frac{\Gamma \Rightarrow \Delta, t\!:\!A}{\Gamma \Rightarrow \Delta, (t+s)\!:\!A}\ (\Rightarrow +) \qquad\qquad \frac{\Gamma \Rightarrow \Delta, t\!:\!A}{\Gamma \Rightarrow \Delta, (s+t)\!:\!A}\ (\Rightarrow +)$$

$$\frac{\Gamma \Rightarrow \Delta, s\!:\!(A \to B) \qquad \Gamma \Rightarrow \Delta, t\!:\!A}{\Gamma \Rightarrow \Delta, (s \cdot t)\!:\!B}\quad (\Rightarrow \cdot)$$

As will follow from the proof of 5.1 the rule $(\Rightarrow \cdot)$ for $\mathcal{LPG}_0$ can in fact be limited by the condition that $A \to B$ must occur in Γ, Δ, without losing any provable sequents.

The system $\mathcal{LPG}$ is $\mathcal{LPG}_0$ plus the rule

$$\frac{}{\Rightarrow c\!:\!\mathbf{A}}\quad (\Rightarrow c),$$

where $\mathbf{A}$ is an axiom *A0* – *A4* from Section 1, and c is a proof constant.

Under $\mathcal{LPG}^-$ and $\mathcal{LPG}_0^-$ we mean the cut free fragments of $\mathcal{LPG}$ and $\mathcal{LPG}_0^-$ respectively.

4.1 Theorem. $\mathcal{LPG}_0 \vdash \Gamma \Rightarrow \Delta$ *iff* $\mathcal{LP}_0 \vdash \bigwedge \Gamma \to \bigvee \Delta$, $\mathcal{LPG} \vdash \Gamma \Rightarrow \Delta$ *iff* $\mathcal{LP} \vdash \bigwedge \Gamma \to \bigvee \Delta$.

The proof proceeds by a straightforward induction.

4.2 Corollary. $\mathcal{LP}(\mathcal{CS}) \vdash F$ *iff* $\mathcal{LPG}_0 \vdash \mathcal{CS} \Rightarrow F$

4.3 Definition. The sequent $\Gamma \Rightarrow \Delta$ is *saturated* if

1. $A \to B \in \Gamma$ implies $B \in \Gamma$ or $A \in \Delta$,

2. $A \to B \in \Delta$ implies $A \in \Gamma$ and $B \in \Delta$[3],

3. $t{:}A \in \Gamma$ implies $A \in \Gamma$,

4. $!t{:}t{:}A \in \Delta$ implies $t{:}A \in \Delta$,

5. $(s+t){:}A \in \Delta$ implies $s{:}A \in \Delta$ and $t{:}A \in \Delta$

6. $(s \cdot t) : B \in \Delta$ implies *for each* $X \to B$ *occurring as a subformula in* Γ, Δ *either* $s{:}(X \to B) \in \Delta$ *or* $t{:}X \in \Delta$.

4.4 Lemma. (Saturation lemma) *Suppose* $\mathcal{LPG}_0^- \nvdash \Gamma \Rightarrow \Delta$. *Then there exists a saturated sequent* $\Gamma' \Rightarrow \Delta'$ *such that*

1. $\Gamma \subseteq \Gamma'$, $\Delta \subseteq \Delta'$,

2. $\Gamma' \Rightarrow \Delta'$ *is not derivable in* $\mathcal{LPG}_0^-$.

Proof. A saturated sequent is obtained by the following *Saturation Algorithm* $\mathcal{SA}$. Given $\Gamma \Rightarrow \Delta$, for each undischarged formula S from $\Gamma \cup \Delta$ non-deterministically try to perform one of the following steps. At the moment 0 all formulas from $\Gamma \cup \Delta$ are available. After a step is performed discharge S (make it unavailable). If none of the clauses 1 - 6 is applicable terminate with success.

1. if $S = (A \to B) \in \Gamma$, then put A into Δ or B into Γ,
2. if $S = (A \to B) \in \Delta$, then put A into Γ and B into Δ,
3. if $S = t{:}A \in \Gamma$, then put A into Γ,
4. if $S = !t{:}t{:}A \in \Delta$, then put $t{:}A$ into Δ,
5. if $S = (s+t){:}A \in \Delta$, then put both $s{:}A$ and $t{:}A$ into Δ,
6. if $S = (s \cdot t){:}B \in \Delta$, then for each $X_1, \ldots, X_n$ such that $X_i \to B$ is a subformula in Γ, Δ put either $s{:}(X_i \to B)$ or $t{:}X_i$ into Δ,
7. if $\Gamma \cap \Delta \neq \emptyset$ or $\bot \in \Gamma$, then backtrack. If backtracked to the root node terminate with failure. When backtracking to a given node make available again all the formulas discharged after leaving this node previous time.

The Saturation Algorithm $\mathcal{SA}$ terminates. Indeed, $\mathcal{SA}$ is finitely branching and each non-backtracking step breaks either a subformula of $\Gamma \Rightarrow \Delta$ or a formula of the type $t{:}F$, where both t and F occur in $\Gamma \Rightarrow \Delta$. There are only finitely many of those formulas, which guarantees termination. Moreover, $\mathcal{SA}$ terminates with success. Indeed, otherwise $\mathcal{SA}$ terminates at the root node $\Gamma \Rightarrow \Delta$ of the computation tree with all the possibilities exhausted and no way to backtrack. Then the computation tree $\mathcal{T}$ of $\mathcal{SA}$ contains the sequent $\Gamma \Rightarrow \Delta$ at the root, and $\mathcal{LPG}_0$ axioms at the leaf nodes. By a standard induction on the depth of a node in $\mathcal{T}$ one can prove, that every sequent in $\mathcal{T}$ is derivable in $\mathcal{LPG}_0^-$, which contradicts the assumption that $\mathcal{LPG}_0^- \nvdash \Gamma \Rightarrow \Delta$. The nodes corresponding to the steps 1 – 5 and 7 are trivial. Let us consider a node, which corresponds to 6. Such a node is labelled by a sequent $\Pi \Rightarrow \Theta, st{:}B$, and its

[3] The clauses concerning other boolean connectives are optional.

children are 2^n sequents of the form $\Pi \Rightarrow \Theta, st\!:\!B, Y_1^\sigma, \ldots, Y_n^\sigma$, where $\sigma = (\sigma_1 \ldots, \sigma_n)$ is an n-tuple of 0's and 1's, and

$$Y_i^\sigma = \begin{cases} s\!:\!(X_i \to B), & \text{if } \sigma_i = 0 \\ t\!:\!X_i, & \text{if } \sigma_i = 1. \end{cases}$$

Here $X_1, \ldots, X_n$ is the list of all formulas such that $X_i \to B$ is a subformula of $\Gamma \Rightarrow \Delta$. By the induction hypothesis all the child sequents are derivable in $\mathcal{LPG}_0^-$. In particular, among them there are 2^{n-1} pairs of sequents of the form $\Pi \Rightarrow \Theta', s\!:\!(X_1 \to B)$ and $\Pi \Rightarrow \Theta', t\!:\!X_1$. To every such pair apply the rule $(\Rightarrow \cdot)$ to obtain $\Pi \Rightarrow \Theta'$ (we assume, that $st : B \in \Theta'$. The resulting 2^{n-1} sequents are of the form $\Pi \Rightarrow \Theta, st\!:\!B, Y_2^\sigma, \ldots, Y_n^\sigma$. After we repeat this procedure $n-1$ more times we end up with the sequent $\Pi \Rightarrow \Theta, st\!:\!B$, which is thus derivable in $\mathcal{LPG}_0^-$.
◀

Note, that in a saturated sequent $\Gamma \Rightarrow \Delta$ which is not $\mathcal{LPG}_0^-$ derivable the set Γ is closed under the rules $t\!:\!X/X$ and $X \to Y, X/Y$.

4.5 Lemma. *For each saturated sequent $\Gamma \Rightarrow \Delta$ not derivable in $\mathcal{LPG}_0^-$ there is a set of $\mathcal{LP}$-formulas $\widetilde{\Gamma}$ (a completion of $\Gamma \Rightarrow \Delta$) such that*

1. *$\widetilde{\Gamma}$ is a provably decidable set, for each term t the set $I(t) = \{X \mid t\!:\!X \in \widetilde{\Gamma}\}$ is finite and a function from a code[4] of t to a code[5] of $I(t)$ is provably computable,*
2. *$\Gamma \subseteq \widetilde{\Gamma}$, $\Delta \cap \widetilde{\Gamma} = \emptyset$,*
3. *if $t\!:\!X \in \widetilde{\Gamma}$, then $X \in \widetilde{\Gamma}$,*
4. *if $s\!:\!(X \to Y) \in \widetilde{\Gamma}$ and $t\!:\!X \in \widetilde{\Gamma}$, then $(s \cdot t)\!:\!Y \in \widetilde{\Gamma}$,*
5. *if $t\!:\!X \in \widetilde{\Gamma}$, then $!t\!:\!t\!:\!X \in \widetilde{\Gamma}$,*
6. *if $t\!:\!X \in \widetilde{\Gamma}$ and s is a proof polynomial, then $(t + s)\!:\!X \in \widetilde{\Gamma}$ and $(s + t)\!:\!X \in \widetilde{\Gamma}$.*

Proof. We describe a *completion algorithm* $\mathcal{COM}$ that produces a series of finite sets of $\mathcal{LP}$-formulas $\Gamma_0, \Gamma_1, \Gamma_2 \ldots$. Let $\Gamma_0 = \Gamma$.

For each natural number $i > 1$ let $\mathcal{COM}$ do the following:

if $i = 3k$, then $\mathcal{COM}$ puts

$$\Gamma_{i+1} = \Gamma_i \bigcup_{s,t} \{(s \cdot t)\!:\!Y \mid s\!:\!(X \to Y), t\!:\!X \in \Gamma_i\},$$

if $i = 3k + 1$, then $\mathcal{COM}$ puts

$$\Gamma_{i+1} = \Gamma_i \bigcup_{t} \{!t\!:\!t\!:\!X \mid t\!:\!X \in \Gamma_i\},$$

[4] For example, the Gödel number of t.

[5] For example, the canonical number of the finite set of Gödel numbers of the formulas from $I(t)$.

if $i = 3k+2$, then $\mathcal{COM}$ puts

$$\Gamma_{i+1} = \Gamma_i \bigcup_{s,t} \{(s+t):X, (t+s):X \mid t:X \in \Gamma_i, |s| < i.\}^6$$

Put

$$\widetilde{\Gamma} = \bigcup_i \Gamma_i.$$

By definition, $\Gamma_i \subseteq \Gamma_{i+1}$.

It is easy to see that at step $i > 0$ $\mathcal{COM}$ produces either a formula from Γ or formulas of the form $t:X$ with the length of t greater than i. This observation secures the decidability of $\widetilde{\Gamma}$. Indeed, given a formula F of length n wait until step $i = n$ of $\mathcal{COM}$; $F \in \Gamma_n$ iff $F \in \widetilde{\Gamma}$. Similar argument establishes the decidability of $I(t)$ from which one can construct the desired provable computable arithmetical term for $I(t)$.

In order to establish 2 and 3 we prove by induction on i that for all $i = 0, 1, 2, \ldots$

1. $\Gamma_i \cap \Delta = \emptyset$,
2. $t:X \in \Gamma_i \quad \Rightarrow \quad X \in \Gamma_i$,
3. $X \to Y, X \in \Gamma_i \quad \Rightarrow \quad Y \in \Gamma_i$.

Base case $i = 0$ holds because of the saturation properties of $\Gamma_0 = \Gamma$.

The step of induction. Assume the induction hypothesis that the properties 1,2,3 hold for i and consider Γ_{i+1}.

1. Suppose there is $F \in \Gamma_{i+1} \cap \Delta$ but $F \notin \Gamma_i$. There are three possibilities. If $i-1 = 3k$ then F is $(s \cdot t):Y$ such that $s:(X \to Y), t:X \in \Gamma_i$ for some X. From the description of $\mathcal{COM}$ and from the remark before Lemma 4.5 it follows that $(X \to Y) \in \Gamma$. By the saturation properties of $\Gamma \Rightarrow \Delta$, since $(s \cdot t):Y \in \Delta$ and $X \to Y$ occurs in Γ either $s:(X \to Y) \in \Delta$ or $t:X \in \Delta$. In either case $\Gamma_i \cap \Delta \neq \emptyset$ which is impossible by the induction hypothesis.

If $i-1 = 3k+1$ then F is $!t:t:X$ such that $t:X \in \Gamma_i$. By the saturation properties of Δ, $t:X \in \Delta$. Again $\Gamma_i \cap \Delta \neq \emptyset$ which is impossible by the induction hypothesis.

If $i-1 = 3k+2$ then F is $(t+s):X$ such that either $t:X \in \Gamma_i$ or $s:X \in \Gamma_i$. By the saturation properties, from $(t+s):X \in \Delta$ conclude that both $t:X \in \Delta$ and $s:X \in \Delta$. Again $\Gamma_i \cap \Delta \neq \emptyset$ which is impossible by the induction hypothesis.

Thus $\Gamma_{i+1} \cap \Delta = \emptyset$.

2. Suppose $p:B \in \Gamma_{i+1}$ and $p:B \notin \Gamma_i$. We conclude that $B \in \Gamma_{i+1}$. Indeed, there are three possibilities again.

If If $i-1 = 3k$ then $p:B$ is $(s \cdot t):Y$ such that $s:(X \to Y), t:X \in \Gamma_i$ for some X. By the induction hypothesis for Γ_i, $(X \to Y), X \in \Gamma_i$ and thus $Y \in \Gamma_i$. By the inclusion $\Gamma_i \subseteq \Gamma_{i+1}$, $Y \in \Gamma_{i+1}$.

[6] $|s|$ is the length of s, i.e. the total number of variables, constants, and functional symbols in s.

If $i-1=3k+1$ then $p{:}B$ is $!t{:}t{:}X$ such that $t{:}X \in \Gamma_i$. Then $t{:}X \in \Gamma_{i+1}$.

If $i-1=3k+2$ then $p{:}B$ is $(t+s){:}B$ such that either $t{:}B \in \Gamma_i$ or $s{:}B \in \Gamma_i$. By the induction hypothesis, in either case $B \in \Gamma_i$, therefore $B \in \Gamma_{i+1}$.

3. Suppose $X \to Y, X \in \Gamma_{i+1}$. From the description of $\mathcal{COM}$ it follows that $(X \to Y) \in \Gamma$. By the saturation properties of $\Gamma \Rightarrow \Delta$, either $Y \in \Gamma$ or $X \in \Delta$. In the former case we are done. If $X \in \Delta$ then $\Gamma_{i+1} \cap \Delta \neq \emptyset$ which is impossible by the item 1 of the induction step.

Items 4., 5., 6. of Lemma 4.5 are guaranteed by the definition of $\mathcal{COM}$. Indeed, if some *if* condition is fulfilled, then it occurs at step i and $\mathcal{COM}$ necessarily puts the *then* formula into Γ_{i+3} at the latest.
◀

5 Consolidated completeness theorem

In this section we establish completeness and cut elimination theorems for the Logic of Proofs.

5.1 Theorem. *The following are equivalent*

1. $\mathcal{LPG}_0^- \vdash \Gamma \Rightarrow \Delta$,
2. $\mathcal{LPG}_0 \vdash \Gamma \Rightarrow \Delta$,
3. $\mathcal{LP}_0 \vdash \bigwedge \Gamma \to \bigvee \Delta$,
4. *for every interpretation* $*$ $\mathcal{PA} \vdash (\bigwedge \Gamma \to \bigvee \Delta)^*$,
5. *for every interpretation* $*$ *the formula* $(\bigwedge \Gamma \to \bigvee \Delta)^*$ *is true.*

Proof. The steps from 1 to 2 and from 4 to 5 are trivial. The step from 2 to 3 follows from 4.1, from 3 to 4 follows from 2.6. The only remaining step is thus from 5 to 1. We assume "not 1" and establish "not 5". Suppose $\mathcal{LPG}_0^- \nvdash \Gamma \Rightarrow \Delta$. Our aim now will be to construct an interpretation $*$ such that $(\bigwedge \Gamma \to \bigvee \Delta)^*$ is false (in the standard model of arithmetic).

Perform a saturation procedure to get a saturated sequent $\Gamma' \Rightarrow \Delta'$ (4.4), and then make a completion to get a set of formulas $\widetilde{\Gamma'}$ (4.5).

We define the desired interpretation $*$ on propositional letters S_i, proof variables x_j and proof constants a_j first. We assume that Gödel numbering of the joint language of $\mathcal{LP}$ and $\mathcal{PA}$ is injective, i.e.

$$\ulcorner E_1 \urcorner = \ulcorner E_2 \urcorner \quad \leftrightarrow \quad E_1 \equiv E_2,$$

and that 0 is not a Gödel number of any expression. For a propositional letter S, proof variable x and proof constant a let

$$S^* = \begin{cases} \ulcorner S \urcorner = \ulcorner S \urcorner, & \text{if } S \in \widetilde{\Gamma'} \\ \ulcorner S \urcorner = 0, & \text{if } S \notin \widetilde{\Gamma'}, \end{cases} \qquad x^* = \ulcorner x \urcorner, \qquad a^* = \ulcorner a \urcorner.$$

The remaining parts of $*$ are constructed by a multiple arithmetical fixed point equation below.

For any arithmetical formula $\mathit{Prf}(x,y)$ define an auxiliary translation † of $\mathcal{LP}$-terms to numerals and $\mathcal{LP}$-formulas to $\mathcal{PA}$-formulas such that $S^\dagger = S^*$ for any propositional letter S, $t^\dagger = \ulcorner t \urcorner$ for any $\mathcal{LP}$-term t, $(t\!:\!F)^\dagger = \mathit{Prf}(t^\dagger, \ulcorner F^\dagger \urcorner)$, and † commutes with the propositional connectives.

It is clear that if $\mathit{Prf}(x,y)$ contains quantifiers, then † is injective, i.e. $F^\dagger \equiv G^\dagger$ yields $F \equiv G$. Indeed, from $F^\dagger \equiv G^\dagger$ it follows that the principal connectives in F and G coincide. We consider one case: $(F_1 \to F_2)^\dagger \equiv (s\!:\!G)^\dagger$ is impossible. Since $(s\!:\!G)^\dagger \equiv \mathit{Prf}(k,n)$ for the corresponding numerals k and n, this formula contains quantifiers. Therefore the formula $(F_1 \to F_2)^\dagger \equiv {F_1}^\dagger \to {F_2}^\dagger$ also contains quantifiers and thus contains a subformula of the form $\mathit{Prf}(k_1,n_1)$. However, $E(\mathit{Prf}(k_1,n_1)) \equiv \mathit{Prf}(k,n)$ is impossible for a nontrivial E since the number of logical connectives and quantifiers in $E(\mathit{Prf}(k_1,n_1))$ and $\mathit{Prf}(k,n)$ is different. Now the injectivity of † can be shown by an easy induction on the construction of an $\mathcal{LP}$-formula. Moreover, one can construct primitive recursive functions f and g such that

$$f(\ulcorner B \urcorner, \ulcorner \mathit{Prf} \urcorner) = \ulcorner B^\dagger \urcorner, \quad g(\ulcorner B^\dagger \urcorner, \ulcorner \mathit{Prf} \urcorner) = \ulcorner B \urcorner.$$

Let *(PROOF,* $\otimes$, $\oplus$, $\Uparrow$*)* be the standard non-deterministic proof predicate from Section 2, with $\otimes$ standing for application, $\oplus$ for choice and $\Uparrow$ for proof checker operations on proofs associated with *PROOF*. In particular, for any arithmetical formulas φ, ψ and any natural numbers k, n the following formulas are true:

$$\mathit{PROOF}(k, \ulcorner \varphi \to \psi \urcorner) \wedge \mathit{PROOF}(n, \ulcorner \varphi \urcorner) \to \mathit{Prf}(k \otimes n, \ulcorner \psi \urcorner)$$

$$\mathit{PROOF}(k, \ulcorner \varphi \urcorner) \to \mathit{PROOF}(k \oplus n, \ulcorner \varphi \urcorner), \quad \mathit{PROOF}(n, \ulcorner \varphi \urcorner) \to \mathit{PROOF}(k \oplus n, \ulcorner \varphi \urcorner)$$

$$\mathit{PROOF}(k, \ulcorner \varphi \urcorner) \to \mathit{PROOF}(\Uparrow\! k, \ulcorner \mathit{PROOF}(k, \ulcorner \varphi \urcorner) \urcorner).$$

For technical convenience and without loss of generality we assume that $\mathit{PROOF}(\ulcorner t \urcorner, k)$ is false for any $\mathcal{LP}$-term t and any $k \in \omega$.

Under $\mu x \varphi(x, \vec{y})$ we mean a function that calculates x such that

$$\varphi(x, \vec{y}) \wedge \forall z < x \neg \varphi(z, \vec{y})).$$

It is clear that $\mu x \varphi(x, \vec{y})$ is computable if $\varphi(x, \vec{y}) \wedge \forall z < x \neg \varphi(z, \vec{y})$ is provably Σ_1. There are two convenient sufficient conditions under each of which $\mu x \varphi(x, \vec{y})$ is computable:

$\varphi(x, \vec{y})$ is provably Δ_1,

$\varphi(x, \vec{y})$ is provably Σ_1 and functional with respect to x, i.e. $\varphi(k_1, \vec{n}) \wedge \varphi(k_2, \vec{n}) \to k_1 = k_2$ is true for all $k_1, k_2, \vec{n}$.

By the arithmetical fixed point argument we construct the formula $\mathit{Prf}(x,y)$ such that $\mathcal{PA}$ proves the following *fixed point equation (FPE)*:

$$
\begin{aligned}
Prf(x,y) \quad \leftrightarrow \quad & PROOF(x,y) \quad \vee \\
& (\text{“}x = \ulcorner t \urcorner \textit{ for some } \mathcal{LP}\textit{-term } t \textit{ and} \\
& y = \ulcorner B^{\dagger} \urcorner \textit{ for some } \mathcal{LP}\textit{-formula } B \textit{ such that } B \in I(t)\text{”})
\end{aligned}
$$

Here the arithmetical formula "..." describes a primitive recursive procedure: given x and y recover t and B such that $x = \ulcorner t \urcorner$ and $y = \ulcorner B^{\dagger} \urcorner$, then verify $B \in I(t)$. From *FPE* it is immediate that *Prf* is a provably Δ_1-formula, since $PROOF(x,y)$ is provably Δ_1. It also follows from *FPE* that $\mathcal{PA} \vdash \psi$ yields $Prf(k, \ulcorner \psi \urcorner)$ for some $k \in \omega$.

We define the arithmetical formulas $M(x,y,z)$, $A(x,y,z)$, $C(x,z)$ as follows

$$
\begin{aligned}
M(x,y,z) \;\leftrightarrow\; & (\text{“}x = \ulcorner s \urcorner \textit{ and } y = \ulcorner t \urcorner \textit{ for some } \mathcal{LP}\textit{-terms } s \textit{ and } t\text{”} \wedge z = \ulcorner s \cdot t \urcorner) \vee \\
& (\text{“}x = \ulcorner s \urcorner \textit{ for some } \mathcal{LP}\textit{-term } s \textit{ and } y \neq \ulcorner t \urcorner \textit{ for any } \mathcal{LP}\textit{-term } t\text{”} \wedge \\
& \exists v[\text{“}v = \mu w(\textstyle\bigwedge\{PROOF(w, \ulcorner B^{\dagger} \urcorner) \mid B \in I(s)\})\text{”} \wedge z = v \otimes y]) \vee \\
& (\text{“}x \neq \ulcorner s \urcorner \textit{ for any } \mathcal{LP}\textit{-term } s \textit{ and } y = \ulcorner t \urcorner \textit{ for some } \mathcal{LP}\textit{-term } t\text{”} \wedge \\
& \exists v[\text{“}v = \mu w(\textstyle\bigwedge\{PROOF(w, \ulcorner B^{\dagger} \urcorner) \mid B \in I(t)\})\text{”} \wedge z = x \otimes v]) \vee \\
& (\text{“}x \neq \ulcorner s \urcorner \textit{ and } y \neq \ulcorner t \urcorner \textit{ for any } \mathcal{LP}\textit{-terms } s \textit{ and } t\text{”} \wedge z = x \otimes y)
\end{aligned}
$$

$$
\begin{aligned}
A(x,y,z) \;\leftrightarrow\; & (\text{“}x = \ulcorner s \urcorner \textit{ and } y = \ulcorner t \urcorner \textit{ for some } \mathcal{LP}\textit{-terms } s \textit{ and } t\text{”} \wedge z = \ulcorner s + t \urcorner) \vee \\
& (\text{“}x = \ulcorner s \urcorner \textit{ for some } \mathcal{LP}\textit{-term } s \textit{ and } y \neq \ulcorner t \urcorner \textit{ for any } \mathcal{LP}\textit{-term } t\text{”} \wedge \\
& \exists v[\text{“}v = \mu w(\textstyle\bigwedge\{PROOF(w, \ulcorner B^{\dagger} \urcorner) \mid B \in I(s)\})\text{”} \wedge z = v \oplus y]) \vee \\
& (\text{“}x \neq \ulcorner s \urcorner \textit{ for any } \mathcal{LP}\textit{-term } s \textit{ and } y = \ulcorner t \urcorner \textit{ for some } \mathcal{LP}\textit{-term } t\text{”} \wedge \\
& \exists v[\text{“}v = \mu w(\textstyle\bigwedge\{PROOF(w, \ulcorner B^{\dagger} \urcorner) \mid B \in I(t)\})\text{”} \wedge z = x \oplus v]) \vee \\
& (\text{“}x \neq \ulcorner s \urcorner \textit{ and } y \neq \ulcorner t \urcorner \textit{ for any } \mathcal{LP}\textit{-terms } s \textit{ and } t\text{”} \wedge z = x \oplus y)
\end{aligned}
$$

$$
\begin{aligned}
C(x,z) \leftrightarrow\; & (\text{“}x = \ulcorner t \urcorner \textit{ for some } \mathcal{LP}\textit{-term } t\text{”} \wedge z = \ulcorner !t \urcorner) \vee \\
& (\text{“}x \neq \ulcorner t \urcorner \textit{ for any } \mathcal{LP}\textit{-term } t\text{”} \wedge \\
& \exists v[\text{“}v = \mu w(\textstyle\bigwedge\{PROOF(w, \ulcorner PROOF(t, \ulcorner \varphi \urcorner) \rightarrow Prf(t, \ulcorner \varphi \urcorner) \urcorner) \mid \varphi \in T(t)\})\text{”} \wedge \\
& z = v \otimes \Uparrow x])
\end{aligned}
$$

Here "..." denotes a natural arithmetical formula representing in $\mathcal{PA}$ the condition '...', "$v = \mu w \psi$" is a natural formula representing in $\mathcal{PA}$ the function $\mu w \psi$. Note, that in the definitions above all these functions are computable since all the corresponding ψ's are provably Δ_1. Therefore $M(x,y,z)$, $A(x,y,z)$ and $C(x,z)$ are provably Σ_1. Moreover, these formulas are functional with respect to z. By the necessary conditions above the functions $m(x,y)$, $a(x,y)$ and $c(x)$ are computable.

We continue defining the interpretation $*$. Let *Prf* for $*$ be the one from *FPE*,

$$m(x,y) := \mu z M(x,y,z), \quad a(x,y) := \mu z A(x,y,z), \quad c(x) := \mu z C(x,z).$$

5.2 Lemma.

a) $t^* = t^\dagger$ *for any* $\mathcal{LP}$*-term* t,

b) $B^* \equiv B^\dagger$ *for any* $\mathcal{LP}$*-formula* B.

Proof. a) Induction on the construction of an $\mathcal{LP}$-term. Base cases are covered by the definition of the interpretation $*$. The induction step. According to the definitions, the following equalities are provable in $\mathcal{PA}$:

$$(s \cdot t)^* = m(s^*, t^*) = m(\ulcorner s \urcorner, \ulcorner t \urcorner) = \ulcorner s \cdot t \urcorner = (s \cdot t)^\dagger,$$

$$(s + t)^* = a(s^*, t^*) = a(\ulcorner s \urcorner, \ulcorner t \urcorner) = \ulcorner s + t \urcorner = (s + t)^\dagger,$$

$$(!t)^* = c(t^*) = c(\ulcorner t \urcorner) = \ulcorner !t \urcorner = (!t)^\dagger.$$

b) By an induction on B we prove that B^* and $B^\dagger$ coincide. Atomic case when B is a propositional letter holds by the definitions. If B is $t\!:\!F$, then $(t\!:\!F)^* = Prf(t^*, \ulcorner F^* \urcorner)$. By a), $t^* = t^\dagger$. By the induction hypothesis, $F^* \equiv F^\dagger$ which yields $\ulcorner F^* \urcorner = \ulcorner F^\dagger \urcorner$. Therefore $Prf(t^*, \ulcorner F^* \urcorner) = Prf(t^\dagger, \ulcorner F^\dagger \urcorner) = (t\!:\!F)^\dagger$. The steps of the induction are trivial.

◀

5.3 Corollary. *The mapping* $*$ *is injective on terms and formulas of* $\mathcal{LP}$*. In particular, for all expressions* E_1 *and* E_2

$$E_1{}^* = E_2{}^* \quad \Rightarrow \quad E_1 \equiv E_2.$$

5.4 Corollary. X^* *is provably* Δ_1 *for any* $\mathcal{LP}$*-formula* X *.*

Indeed, if X is atomic, then X is provably Δ_1 by the definition of $*$. If X is $t\!:\!Y$, then $(t\!:\!Y)^*$ is $Prf(t^*, \ulcorner Y^* \urcorner)$. By Lemma 5.2,

$$\mathcal{PA} \vdash Prf(t^*, \ulcorner Y^* \urcorner) \leftrightarrow Prf(\ulcorner t \urcorner, \ulcorner Y^* \urcorner).$$

The latter formula is provably Δ_1, therefore $(t\!:\!Y)^*$ is provably Δ_1. Since the class of provably Δ_1 formulas is closed under boolean connectives X^* is provably Δ_1 for each X.

5.5 Lemma. *If $X \in \widetilde{\Gamma}'$, then $\mathcal{PA} \vdash X^*$, if $X \in \Delta'$, then $\mathcal{PA} \vdash \neg X^*$.*

Proof. By induction on the length of X. Base case, i.e. X is atomic or $X = t:Y$. Let X be atomic. By the definition of $*$, X^* is true iff $X \in \widetilde{\Gamma}'$. Let $X = t:Y$ and $t:Y \in \widetilde{\Gamma}'$. Then $\mathcal{PA} \vdash$ "$Y \in I(t)$". By *FPE*, $\mathcal{PA} \vdash Prf(\ulcorner t \urcorner, \ulcorner Y^\dagger \urcorner)$. By Lemma 5.2, $\mathcal{PA} \vdash Prf(t^*, \ulcorner Y^* \urcorner)$. Therefore $\mathcal{PA} \vdash (t:Y)^*$.

If $t:Y \in \Delta'$, then $t:Y \notin \widetilde{\Gamma}'$ and "$Y \in I(t)$" is false. The formula $PROOF(t^*, \ulcorner Y^* \urcorner)$ is also false since t^* is $\ulcorner t \urcorner$ (by Lemma 5.2) and $PROOF(\ulcorner t \urcorner, k)$ is false for any k by assumption. By *FPE*, $(t:Y)^*$ is false. Since $(t:Y)^*$ is provably Δ_1 (Lemma 5.4) $\mathcal{PA} \vdash \neg(t:Y)^*$.

The induction steps corresponding to boolean connectives are standard and based on the saturation properties of $\Gamma' \Rightarrow \Delta'$. For example, let $X = Y \rightarrow Z \in \widetilde{\Gamma}'$. Then $Y \rightarrow Z \in \Gamma'$, and by definition 4.3, $Y \in \Gamma'$ or $Z \in \Delta'$. By the induction hypothesis, Y^* is true or Z^* is false, and thus $(Y \rightarrow Z)^*$ is true, etc.
◀

5.6 Lemma. $\mathcal{PA} \vdash \varphi \quad \Leftrightarrow \quad Prf(n, \ulcorner \varphi \urcorner)$ *for some* $n \in \omega$.

Proof. It remains to establish ($\Leftarrow$). Let $Prf(n, \ulcorner \varphi \urcorner)$ for some $n \in \omega$. By *FPE* ,

$$Prf(n, \ulcorner \varphi \urcorner) \;\Rightarrow\; PROOF(n, \ulcorner \varphi \urcorner) \;\text{ or }\; \ulcorner \varphi \urcorner = \ulcorner B^\dagger \urcorner \;\text{ for some } B \text{ such that } t:B \in \widetilde{\Gamma}'.$$

In the latter case by the saturation property of $\widetilde{\Gamma}'$, $B \in \widetilde{\Gamma}'$. By Lemma 5.5, $\mathcal{PA} \vdash B^*$. By the injectivity of the Gödel numbering, $\varphi \equiv B^\dagger$. By Lemma 5.2, $\varphi \equiv B^*$. Therefore $\mathcal{PA} \vdash \varphi$.
◀

5.7 Lemma. *For all arithmetical formulas φ, ψ and natural numbers k, n the following is true*

a) $Prf(k, \ulcorner \varphi \rightarrow \psi \urcorner) \wedge Prf(n, \ulcorner \varphi \urcorner) \rightarrow Prf(m(k,n), \ulcorner \psi \urcorner)$

b) $Prf(k, \ulcorner \varphi \urcorner) \rightarrow Prf(a(k,n), \ulcorner \varphi \urcorner), \qquad Prf(n, \ulcorner \varphi \urcorner) \rightarrow Prf(a(k,n), \ulcorner \varphi \urcorner)$

c) $Prf(k, \ulcorner \varphi \urcorner) \rightarrow Prf(c(k), \ulcorner Prf(k, \ulcorner \varphi \urcorner) \urcorner)$.

Proof. *a)* Assume $Prf(k, \ulcorner \varphi \rightarrow \psi \urcorner)$ and $Prf(n, \ulcorner \varphi \urcorner)$. There are four possibilities.

i) Neither of k, n equals to a Gödel number of an $\mathcal{LP}$-term. By *FPE*, both $PROOF(n, \ulcorner \varphi \urcorner)$ and $PROOF(k, \ulcorner \varphi \rightarrow \psi \urcorner)$ hold so does $PROOF(k \otimes n, \ulcorner \psi \urcorner)$.

ii) Both k and n equal to Gödel numbers of some $\mathcal{LP}$-terms, say $k = \ulcorner s \urcorner$ and $n = \ulcorner t \urcorner$. By *FPE*, φ is F^* and ψ is G^* for some $\mathcal{LP}$-formulas F, G such that $F \rightarrow G \in I(s)$ and $F \in I(t)$.

By the closure property of $\widetilde{\Gamma}'$ (Lemma 4.5(4)), $G \in I(s \cdot t)$. By *FPE*, $Prf(\ulcorner s \cdot t \urcorner, \ulcorner G^* \urcorner)$. By Lemma 5.2 and by definitions, $\mathcal{PA}$ proves that

$$\ulcorner s \cdot t \urcorner = (s \cdot t)^* = m(s^*, t^*) = m(\ulcorner s \urcorner, \ulcorner t \urcorner) = m(k, n).$$

Thus $m(k,n) = \ulcorner s \cdot t \urcorner$ and $Prf(m(k,n), \ulcorner \psi \urcorner)$ is true.

iii) k is not equal to a Gödel number of an $\mathcal{LP}$-term, $n = \ulcorner t \urcorner$ for some $\mathcal{LP}$-term t. By *FPE*, $PROOF(k, \ulcorner \varphi \to \psi \urcorner)$ and $\varphi \equiv F^\dagger$ for some $\mathcal{LP}$-formula F such that $F \in I(t)$. Compute the number

$$l = \mu w (\bigwedge \{PROOF(w, \ulcorner B^\dagger \urcorner) \mid B \in I(t)\})$$

by the following way. Take $I(t) = \{B_1, \ldots, B_l\}$. By definition, $B_i \in \widetilde{\Gamma}'$, $i = 1, \ldots, l$. By Lemma 5.5, $\mathcal{PA} \vdash B_i{}^*$ for all $i = 1, \ldots, l$. By Lemma 5.2, $\mathcal{PA} \vdash B_i{}^\dagger$ for all $i = 1, \ldots, l$. By the conjoinability property of *PROOF* there exists w such that $PROOF(w, \ulcorner B_i{}^\dagger \urcorner)$ for all $i = 1, \ldots, l$. Let j be the least such w. In particular, $PROOF(j, \ulcorner F^\dagger \urcorner)$. By the definition of $\otimes$, $PROOF(k \otimes j, \ulcorner \psi \urcorner)$. By the definition of M, $\mathcal{PA} \vdash m(k,n) = k \otimes j$, therefore $PROOF(m(k,n), \ulcorner \psi \urcorner)$.

The case iv): "s is a Gödel number of an $\mathcal{LP}$-term but t is not a Gödel number of any $\mathcal{LP}$-term" is similar to (iii).

The case (b) can be checked in the same way as (a).

c) Given $Prf(k, \ulcorner \varphi \urcorner)$ there are two possibilities.

i) $k = \ulcorner t \urcorner$ for some $\mathcal{LP}$-term t. By *FPE*, $\varphi \equiv F^\dagger$ for some F such that $F \in I(t)$. By the closure property 4.5(5) of $\widetilde{\Gamma}'$, $!t{:}t{:}F \in \widetilde{\Gamma}'$. By Lemma 5.5, $(!t{:}t{:}F)^*$ holds. By definitions,

$$(!t{:}t{:}F)^* \equiv Prf(c(t^*), \ulcorner Prf(t^*, \ulcorner F^* \urcorner) \urcorner).$$

By Lemma 5.2, $t^* = \ulcorner t \urcorner$ and $F^* \equiv F^\dagger$. Therefore $t^* = k$, $F^* \equiv \varphi$ and

$$Prf(c(k), \ulcorner Prf(k, \ulcorner \varphi \urcorner) \urcorner).$$

ii) $k \neq \ulcorner t \urcorner$ for any $\mathcal{LP}$-term t. By *FPE*, $PROOF(k, \ulcorner \varphi \urcorner)$ holds. By definition of the proof checking operation $\Uparrow$ for *PROOF*,

$$PROOF(\Uparrow k, \ulcorner PROOF(k, \ulcorner \varphi \urcorner) \urcorner).$$

By the definition of C, in this case $\mathcal{PA} \vdash c(k) = l \otimes \Uparrow k$ where

$$l = \mu w \bigwedge \{PROOF(w, \ulcorner PROOF(k, \ulcorner \psi \urcorner) \to Prf(k, \ulcorner \psi \urcorner) \urcorner) \mid PROOF(k, \ulcorner \psi \urcorner)\}.$$

By the definition of l,

$$PROOF(l, \ulcorner PROOF(k, \ulcorner \varphi \urcorner) \to Prf(k, \ulcorner \varphi \urcorner) \urcorner).$$

Therefore

$$PROOF(l \otimes \Uparrow k, \ulcorner Prf(k, \ulcorner\varphi\urcorner)\urcorner).$$

By *FPE*,

$$Prf(l \otimes \Uparrow k, \ulcorner Prf(k, \ulcorner\varphi\urcorner)\urcorner),$$

therefore

$$Prf(c(k), \ulcorner Prf(k, \ulcorner\varphi\urcorner)\urcorner).$$

◀

5.8 Lemma. *The normality conditions for Prf are fulfilled.*

Proof. By *FPE*, *Prf* is provably Δ_1. It follows from *FPE* and 5.6 that for any arithmetical sentence φ

$$\mathcal{PA} \vdash \varphi \text{ if and only if } Prf(n, \ulcorner\varphi\urcorner) \text{ holds for some natural } n.$$

Finiteness of proofs. For each n the set

$$T(k) = \{l \mid Prf(k, l)\}$$

is finite. Indeed, either k is a number of an $\mathcal{LP}$-term, then use the finiteness of $I(t)$. Otherwise use the normality of *PROOF*. An algorithm for the function from k to the canonical number of $T(k)$ for *Prf* can be constructed from those for *PROOF*, and from the decision algorithm for $I(t)$, Lemma 4.5(1).

Conjoinability of proofs for *Prf* is realized by the function $a(x, y)$ since by Lemma 5.7,

$$T(k) \cup T(n) \subseteq T(a(k, n)).$$

◀

Let us finish the proof of the concluding "not 1 implies not 5" part of 5.1. Given a sequent $\Gamma \Rightarrow \Delta$ not provable in $\mathcal{LPG}_0^-$ we have constructed an interpretation $*$ such that Γ^* are all true, and Δ^* are all false in the standard model of arithmetic (5.5). Therefore, $(\bigwedge \Gamma \rightarrow \bigvee \Delta)^*$ is false.

◀

5.9 Corollary. *$\mathcal{LP}_0$ is decidable.*

Indeed, given an $\mathcal{LP}$-formula F run the saturation algorithm $\mathcal{SA}$ on a sequent $\Rightarrow F$. If $\mathcal{SA}$ fails, then $\mathcal{LP}_0 \vdash F$. Otherwise, $\mathcal{LP}_0 \nvdash F$.

5.10 Corollary. (Completeness of $\mathcal{LP}$ with respect to the provability semantics)

$$\begin{array}{rcl} \mathcal{LP}(\mathcal{CS}) \vdash F & \Leftrightarrow & \mathcal{PA} \vdash F^* \textit{ for any } \mathcal{CS}\textit{-interpretation } *. \\ & \Leftrightarrow & F^* \textit{ is true for any } \mathcal{CS}\textit{-interpretation } *. \end{array}$$

5.11 Corollary. (Cut elimination in $\mathcal{LP}$.) *Every sequent derivable in $\mathcal{LPG}$ can be derived without the cut rule.*

Proof. Let $\mathcal{LPG} \vdash \Gamma \Rightarrow \Delta$. For each use of the rule $\overline{\Rightarrow c\!:\!\mathbf{A}}$ replace it by the axiom sequent $c\!:\!\mathbf{A} \Rightarrow c\!:\!\mathbf{A}$, and add $c\!:\!\mathbf{A}$ to the antecedents of all the sequents below the given node. The result will be a legitimate derivation in $\mathcal{LPG}_0$ of the sequent $\mathcal{CS}, \Gamma \Rightarrow \Delta$ for a constants specification $\mathcal{CS}$ consisting of all those $c:\mathbf{A}$'s. By 5.1 there exists a cut-free derivation $\mathcal{D}$ of $\mathcal{CS}, \Gamma \Rightarrow \Delta$ in $\mathcal{LPG}_0$. Without loss of generality we may assume that every formula $c\!:\!\mathbf{A}$ from $\mathcal{CS}$ there has been introduced in $\mathcal{D}$ by an axiom $c\!:\!\mathbf{A} \Rightarrow c\!:\!\mathbf{A}$. Two other options: the weakening rule, or the rule $(:\ \Rightarrow)$ can be eliminated. The weakenings can be easily removed without any harm to a derivation. By induction on the cardinality of $\mathcal{CS}$ we show that there exists a cut-free derivation of $\mathcal{CS}', \Gamma \Rightarrow \Delta$ in $\mathcal{LPG}_0$ such that $\mathcal{CS}' \subseteq \mathcal{CS}$ and no occurrence of a formula $c\!:\!\mathbf{A}$ from $\mathcal{CS}'$ has been introduced in this derivation by the rule $(:\ \Rightarrow)$. The base case corresponds to an axiom and is trivial. Every use of the rule $(:\ \Rightarrow)$

$$\dfrac{\begin{array}{c}\mathcal{D}' \\ \mathbf{A}, \Gamma' \Rightarrow \Delta'\end{array}}{c\!:\!\mathbf{A}, \Gamma' \Rightarrow \Delta'} \qquad \text{transforms into} \qquad \dfrac{\begin{array}{c}\mathcal{D}'_0 \\ \Rightarrow \mathbf{A}\end{array} \qquad \begin{array}{c}\mathcal{D}' \\ \mathbf{A}, \Gamma' \Rightarrow \Delta'\end{array}}{\Gamma' \Rightarrow \Delta'}$$

The end sequent $\Gamma' \Rightarrow \Delta'$ of the latter derivation is one formula from $\mathcal{CS}$ less and, by the induction hypothesis admits a cut-free derivation without an introduction of formulas from $\mathcal{CS}$ by the rule $(:\ \Rightarrow)$. Finally, in a cut-free derivation of $\mathcal{CS}, \Gamma \Rightarrow \Delta$ in $\mathcal{LPG}_0$ without weakenings and $(:\ \Rightarrow)$ for formulas from $\mathcal{CS}$ replace each axiom $c\!:\!\mathbf{A} \Rightarrow c\!:\!\mathbf{A}$ for $c\!:\!\mathbf{A} \in \mathcal{CS}$ by the rule $(\Rightarrow c)$ introducing $\Rightarrow c\!:\!\mathbf{A}$. The result will be a derivation in $\mathcal{LP}$ of the sequent $\Gamma \Rightarrow \Delta$.
◀

Of course, cut elimination for $\mathcal{LP}$ can be also established by a direct system of reductions (cf. Section 8 of this paper). However, here we have got the cut elimination theorem as a side product of the arithmetical completeness theorem.

5.12 Comment. Decidability of $\mathcal{LP}$ follows from the results of [37]. This fact can also be easily obtained from the cut elimination property of $\mathcal{LP}$ (Corollary 5.10).

5.13 Comment. The straightforward analogue of Theorem 5.1 holds for the call-by-name semantics (cf. Comment 2.8) as well. Some minor modifications are needed to adopt the proof of 5.1 to this new case. First, we redefine $\mu x\varphi(x,\vec{y})$ as an arithmetical ι-term

$$\iota z[\varphi(x,\vec{y}) \wedge \forall z < x \neg\varphi(z,\vec{y}))].$$

Then we write down a *Fixed Point Equation* that is similar to *FPE* from 5.1 with some adjustments corresponding to the understanding of $*$ as the call-by-name interpretation, and the new reading of $\mu x\varphi(x,\vec{y})$ as an arithmetical ι-term (cf. [4], [27], [45]).

5.14 Comment. The paper [45] finds a complete axiomatization of the joint logic of proofs with its call-by-name semantics and the formal provability thus combining $\mathcal{LP}$ as it was presented in [4] with the logic of formal provability $\mathcal{GL}$ (cf.[9],[10]).

6 Realization of modal and intuitionistic logics

It is easy to see that a forgetful projection of $\mathcal{LP}$ is correct with respect to $\mathcal{S}4$. Let F^o be the result of substituting $\Box X$ for all occurrences of $t\!:\!X$ in F, and $\Gamma^o = \{F^o \mid F \in \Gamma\}$ for any set Γ of $\mathcal{LP}$-formulas.

6.1 Lemma. *If $\mathcal{LP} \vdash F$, then $\mathcal{S}4 \vdash F^o$.*

Proof. This is a straightforward induction on a derivation in $\mathcal{LP}$.
◀

The goal of the current section is to establish the converse, namely that $\mathcal{LP}$ suffices to realize any $\mathcal{S}4$ theorem. By an *$\mathcal{LP}$-realization* of a modal formula F we mean an assignment of proof polynomials to all occurrences of the modality in F, Let F^r be the image of F under a realization r. Positive and negative occurrences of modality in a formula and a sequent are defined in the usual way. Namely

1. an indicated occurrence of $\Box$ in $\Box F$ is positive;
2. any occurrence of $\Box$ in $G\to F$, $G\wedge F$, $F\wedge G$, $G\vee F$, $F\vee G$, $\Box F$ and $\Gamma \Rightarrow \Delta, F$ has the same polarity as the corresponding occurrence of $\Box$ in F;
3. any occurrence of $\Box$ in $\neg F$, $F\to G$ and $F,\Gamma \Rightarrow \Delta$ has a polarity opposite to that of the corresponding occurrence of $\Box$ in F.

In a provability context $\Box F$ is intuitively understood as "*there exists a proof x of F*". After a skolemization, all negative occurrences of $\Box$ produce arguments of Skolem functions, while positive ones give functions of those arguments. For example, $\Box A \to \Box B$ should be read informally as

$$\exists x \text{ “ } x \textit{ is a proof of } A\text{” } \to \exists y \text{ “ } y \textit{ is a proof of } B\text{”},$$

with the Skolem form

$$\text{“ } x \textit{ is a proof of } A\text{”} \rightarrow \text{“ } f(x) \textit{ is a proof of } B\text{”}.$$

The following definition captures this feature. A realization r is *normal* if all negative occurrences of $\Box$ are realized by proof variables.

6.2 Theorem. *If $\mathcal{S}4 \vdash F$, then $\mathcal{LP} \vdash F^r$ for some normal realization r.*

Proof. Consider a cut-free sequent formulation of $\mathcal{S}4$, with sequents $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite sets of modal formulas. Axioms are sequents of the form $S \Rightarrow S$, where S is a propositional letter, and the sequent $\bot \Rightarrow$. Along with the usual structural rules and rules introducing boolean connectives there are two proper modal rules (cf.[49]):

$$\frac{A, \Gamma \Rightarrow \Delta}{\Box A, \Gamma \Rightarrow \Delta}\,(\Box \Rightarrow) \qquad \text{and} \qquad \frac{\Box\Gamma \Rightarrow A}{\Box\Gamma \Rightarrow \Box A}\,(\Rightarrow \Box)$$

$(\Box\{A_1, \ldots, A_n\} = \{\Box A_1, \ldots, \Box A_n\})$.

If $\mathcal{S}4 \vdash F$, then there exists a cut-free derivation $\mathcal{T}$ of a sequent $\Rightarrow F$. It suffices now to construct a normal realization r such that $\mathcal{LP} \vdash \bigwedge \Gamma^r \rightarrow \bigvee \Delta^r$ for any sequent $\Gamma \Rightarrow \Delta$ in $\mathcal{T}$. We will also speak about a sequent $\Gamma \Rightarrow \Delta$ being derivable in $\mathcal{LP}$ meaning $\mathcal{LP} \vdash \bigwedge \Gamma \rightarrow \bigvee \Delta$, or, equivalently, $\Gamma \vdash_{\mathcal{LP}} \bigvee \Delta$, or $\mathcal{LPG} \vdash \Gamma \Rightarrow \Delta$. Note that in a cut-free derivation $\mathcal{T}$ the rules respect polarities, all occurrences of $\Box$ introduced by $(\Rightarrow \Box)$ are positive, and all negative occurrences are introduced by $(\Box \Rightarrow)$ or by weakening. Occurrences of $\Box$ are *related* if they occur in related formulas of premises and conclusions of rules; we extend this relationship by transitivity. All occurrences of $\Box$ in $\mathcal{T}$ are naturally split into disjoint *families* of related ones. We call a family *essential* if it contains at least one case of the $(\Rightarrow \Box)$ rule.

Now the desired r will be constructed by steps 1 – 3 described below. We reserve a large enough set of proof variables as *provisional variables*.

Step 1. For every negative family and non essential positive family we replace all occurrences of $\Box$ by “x:” for a fresh proof variable x.

Step 2. Pick an essential family f, enumerate all the occurrences of rules $(\Rightarrow \Box)$, which introduce boxes of this family. Let n_f be the total number of such rules for the family f. Replace all boxes of the family f by the term

$$(v_1 + \ldots + v_{n_f}),$$

where v_i's are fresh provisional variables. The resulting tree $\mathcal{T}_0$ is labelled by $\mathcal{LP}$ formulas, since all occurrences of the kind $\Box X$ in $\mathcal{T}$ are replaced by $t{:}X$ for the corresponding t.

Step 3. Replace the provisional variables by proof polynomials as follows. Proceed from the leaves of the tree to its root. By induction on the depth of a node in $\mathcal{T}_0$ we establish that after the process passes a node, a sequent assigned to this node becomes derivable in $\mathcal{LP}$. The axioms $S \Rightarrow S$ and $\perp \Rightarrow$ are derivable in $\mathcal{LP}$. For every rule other than $(\Rightarrow \Box)$ we do not change the realization of formulas, and just establish that the concluding sequent is provable in $\mathcal{LP}$ given that the premises are. Moreover, every move down in the tree $\mathcal{T}_0$ other than $(\Rightarrow \Box)$ is a rule of the system $\mathcal{LPG}$, therefore, the induction steps corresponding to these moves follow easily from the equivalence of $\mathcal{LP}$ and $\mathcal{LPG}$.

Let an occurrence of the rule $(\Rightarrow \Box)$ have number i in the numbering of all rules $(\Rightarrow \Box)$ from a given family f. This rule already has a form

$$\frac{y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow Y}{y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow (u_1 + \ldots + u_{n_f}){:}Y} \ ,$$

where $y_1, \ldots, y_k$ are proof variables, $u_1, \ldots, u_{n_f}$ are proof polynomials, and u_i is a provisional variable. By the induction hypothesis, the premise sequent $y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow Y$ is derivable in $\mathcal{LP}$, which yields a derivation of

$$y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow Y.$$

By Lifting Lemma (1.4), construct a proof polynomial $t(y_1, \ldots, y_n)$ such that

$$y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow t(y_1, \ldots, y_n){:}Y$$

is derivable in $\mathcal{LP}$. Since

$$\mathcal{LP} \vdash t{:}Y \rightarrow (u_1 + \ldots + u_{i-1} + t + u_{i+1} + \ldots + u_{n_f}){:}Y$$

we have

$$\mathcal{LP} \vdash y_1{:}Y_1, \ldots, y_k{:}Y_k \Rightarrow (u_1 + \ldots + u_{i-1} + t + u_{i+1} + \ldots + u_{n_f}){:}Y.$$

Now substitute $t(y_1, \ldots, y_n)$ for u_i everywhere in the tree $\mathcal{T}_0$. Note, that $t(y_1, \ldots, y_n)$ has no provisional variables, there is one provisional variable (namely u_i) less in the entire $\mathcal{T}_0$. All sequents derivable in $\mathcal{LP}$ remain derivable in $\mathcal{LP}$, the conclusion of the given rule $(\Rightarrow \Box)$ became derivable, and the induction step is complete.

Eventually, we substitute terms of non-provisional variables for all provisional variables in $\mathcal{T}_0$ and establish that the corresponding root sequent of $\mathcal{T}_0$ is derivable in $\mathcal{LP}$. Note that the realization of $\Box$'s built by this procedure is normal.
◄

6.3 Corollary. (Arithmetical completeness of $\mathcal{S}4$.) *$\mathcal{S}4 \vdash F$ iff there is a realization r and a constant specification $\mathcal{CS}$ such that F^r is $\mathcal{CS}$-valid.*

6.4 Comment. It follows from 6.1 and 6.2 that $\mathcal{S}4$ is nothing but a lazy version of $\mathcal{LP}$ when we don't keep track on the proof polynomials assigned to the occurrences of $\Box$. Each theorem of $\mathcal{S}4$ admits a decoding via $\mathcal{LP}$ as a statement about specific proofs. The language of $\mathcal{LP}$ is more rich than the one of $\mathcal{S}4$. In particular, $\mathcal{S}4$ theorems admit essentially different realizations in $\mathcal{LP}$. For example, consider two theorems of $\mathcal{LP}$ having the same modal projection:

$$x\!:\!F \vee y\!:\!F \rightarrow (x+y)\!:\!F \quad \text{and} \quad x\!:\!F \vee x\!:\!F \rightarrow x\!:\!F.$$

The former of these formulas is a meaningful specification of the operation "+". In a contrast, the latter one is a trivial tautology.

So, $\mathcal{LP}$ is the right logic of provability, and $\mathcal{S}4$ should be considered as a lazy higher level language on top of $\mathcal{LP}$. A general recipe for using $\mathcal{S}4$ as a provability logic might be the following: derive in $\mathcal{S}4$ or reason about $\mathcal{S}4$ using a conventional modal logic technique as before, translate the results into $\mathcal{LP}$ to recover their true provability meaning.

Let $k(F)$ denote a translation of an intuitionistic formula F into the plain modal language which puts the prefix $\Box$ in front of all atoms and implications in F. It is well-known that $\mathcal{I}nt \vdash F$ *iff* $\mathcal{S}4 \vdash k(F)$ (see, for example, [13]).

6.5 Corollary. (Realization of intuitionistic logic) *For any $\mathcal{I}nt$-formula F*

$$\mathcal{I}nt \vdash F \quad \Leftrightarrow \quad \mathcal{LP} \vdash (k(F))^r \textit{ for some realization } r.$$

6.6 Comment. As it was noticed by A. Kopylov, the example from 6.4 can be generalized: $\mathcal{S}4$ also admits a degenerated realization in the "+"free fragment of $\mathcal{LP}$, under which all agruments of proof polynomilas are denoted by the same proof variable and only one universal constant is used as a coefficient.

For example, the $\mathcal{S}4$-theorem $(\Box A \vee \Box B) \rightarrow \Box(A \vee B)$ (cf. Example 1.6) can be realized in $\mathcal{LP}$ as $(x:A \vee x:B) \rightarrow (c\cdot x):(A \vee B)$ with the constant specification $c:(A \rightarrow A \vee B)$, $c:(B \rightarrow A \vee B)$. As one can see, this realization cripples the provability content of modal logic. Namely, it presupposes that the constant c stands for the proof of two different axioms, which is inconsistent with an injective assignment of proof constants to propositional axioms in the rule *R2* of $\mathcal{LP}$. The assumption, that A and B have the same proof contradicts the intended provability reading of the original modal formula $(\Box A \vee \Box B) \rightarrow \Box(A \vee B)$ as *if there is a proof of A, and there is a proof of B, then there is a proof of $A \vee B$*. Indeed, the Skolem style conversion of this formula from the language with quantifiers into the quantifier-free language with Skolem functions is *if x is a proof of A and y is a proof of B, then $t(x,y)$ is a proof of $A \vee B$*. One can show that such $t(x,y)$ cannot be taken "+"free provided x and y are distinct proof variables.

Indeed, let S_1 and S_2 be propositional letters. Suppose

$$\mathcal{LP} \vdash x\!:\!S_1 \vee y\!:\!S_2 \to t\!:\!(S_1 \vee S_2)$$

for some "+"free term t. Then

$$\mathcal{LP} \vdash x\!:\!S_1 \to t\!:\!(S_1 \vee S_2)$$

and

$$\mathcal{LP} \vdash y\!:\!S_2 \to t\!:\!(S_1 \vee S_2).$$

Consider a cut-free derivation $\mathcal{D}$ of $x\!:\!S_1 \Rightarrow t\!:\!(S_1 \vee S_2)$ in $\mathcal{LPG}$. A routine analysis of $\mathcal{D}$ rules out the use of $(:\ \Rightarrow)$, $(\Rightarrow !)$, $(\Rightarrow +)$ and $(+\Rightarrow)$. The weakenings may be pushed down by the use of obvious conversions of the type

$$\dfrac{\dfrac{x\!:\!S_1 \Rightarrow \Delta}{x\!:\!S_1 \Rightarrow \Delta, t_1\!:\!Y} \qquad x\!:\!S_1 \Rightarrow \Delta, t_2\!:\!(Y \to Z)}{x\!:\!S_1 \Rightarrow \Delta, (t_1 \cdot t_2)\!:\!Z} \qquad \text{converts to} \qquad \dfrac{x\!:\!S_1 \Rightarrow \Delta}{x\!:\!S_1 \Rightarrow \Delta, (t_1 \cdot t_2)\!:\!Z}\ .$$

Eventually we may assume without a loss of generality that a weakening is the last rule of $\mathcal{D}$, if any. Since $\mathcal{LPG} \not\vdash x\!:\!S_1 \Rightarrow$, we may rule out the weakenings in the succedent too. So, $\mathcal{D}$ contains no rules other than $(\Rightarrow \cdot)$, $(\Rightarrow c)$ and possibly a weakening in the antecedent as the last rule in $\mathcal{D}$. Therefore t is a product of some proof constants and the variable x. Similarly, from $\mathcal{LP} \vdash y\!:\!S_2 \to t\!:\!(S_1 \vee S_2)$ we conclude that t is a product of some proof constants and the variable y. Therefore, t is a product of some proof constants, $\mathcal{D}$ does not contain axioms of the sort $x\!:\!S_1 \Rightarrow x\!:\!S_1$, the formula $x\!:\!S_1$ is introduced by the weakening as the last rule of $\mathcal{D}$. This yields $\mathcal{LP} \vdash\ \Rightarrow t\!:\!(S_1 \vee S_2)$ which does not take place.

The "+"free fragment of $\mathcal{LP}$ is not complete with respect to the class of all deterministic proof predicates. It can be made complete by adding the functionality principle from [2]. The completeness of the resulting system $\mathcal{FLP}$ with respect to deterministic proof systems was established by V. Krupski in ([27]). $\mathcal{FLP}$ does not have a modal counterpart. For example, $\mathcal{FLP}$ derives a principle $\neg(x:A \wedge x:(A \to A))$, which has the forgetful projection $\neg(\Box A \wedge \Box(A \to A))$. The latter is false in any normal modal logic. So, provability as a modal operator corresponds to the non-deterministic understanding of proof.

6.7 Corollary. (Arithmetical completeness of $\mathcal{I}nt$.) *$\mathcal{I}nt \vdash F$ iff there is a realization r and constant specification $\mathcal{CS}$ such that $k(F)^r$ $\mathcal{CS}$-valid.*

Kolmogorov's interpretation of intuitionistic logic $\mathcal{I}nt$ as a "*calculus of problems*" ([23]) can be made explicit via $\mathcal{LP}$.

6.8 Definition. Let F be a formula in the intuitionistic propositional language. A formula F is *Kolmogorov realizable* if $\mathcal{LP} \vdash [k(F)]^r$ for some realization r of modalities in $k(F)$ by proof polynomials. *Kolmogorov logic* ($\mathcal{K}$) is the set of all Kolmogorov realizable propositional formulas. Note that the Kolmogorov relizability may be regarded as a direct formalization of the Kolmogorov calculus of problems from [23] under reading "problem solutions" as "proofs".

6.9 Theorem. (Completeness of $\mathcal{I}nt$ with respect to the Kolmogorov realization)

$$\mathcal{I}nt \;=\; \mathcal{K}.$$

Proof. Either $\mathcal{I}nt \vdash F$, then $\mathcal{S}4 \vdash k(F)$ and, by theorem 6.2, there exists a proof realization r such that $\mathcal{LP} \vdash [k(F)]^r$. Therefore, $F \in \mathcal{K}$. Or $\mathcal{I}nt \not\vdash F$, then $\mathcal{S}4 \not\vdash k(F)$ and, by lemma 6.1, $\mathcal{LP} \not\vdash [k(F)]^r$ for any realization r. Hence $F \notin \mathcal{K}$.
◀

6.10 Comment. Definition 6.8 gives a formalization of Kolmogorov's calculus of problems [23]. Propositional atoms are interpreted as *atomic problems*, namely statements of the sort $t{:}S$ meaning "*t is a proof of S*". Intuitionistic connectives are given precise meaning according to inductive definition from [23]. Under this formalization of $\mathcal{K}$ the Kolmogorov conjecture of 1932 holds that $\mathcal{I}nt \;=\; \mathcal{K}$ (Theorem 6.9).

7 On Intuitionistic Logic of Proofs

The Intuitionistic logic of proofs $\mathcal{ILP}$ is defined as a system 1.2 with $A0$ being a list of axiom scheme for the propositional intuitionistic logic. $\mathcal{ILP}$ is correct with respect to the natural provability interpretation as a calculus of proofs for either $\mathcal{PA}$ or the intuitionistic arithmetic $\mathcal{HA}$. We do not address the issue of arithmetical completeness of $\mathcal{ILP}$ in this paper.

The Gentzen style system $\mathcal{ILPG}$ for the intuitionistic logic of proofs can be defined as follows (cf. the system **G2i** from [49]). Sequents in $\mathcal{ILPG}$ are all of the form $\Gamma \Rightarrow F$, where Γ is a multiset of $\mathcal{LP}$-formulas, and F is an $\mathcal{LP}$-formula.

Axioms of $\mathcal{ILPG}$ are sequents of the form $P, \Gamma \Rightarrow P$, where P is either a propositional letter or a formula of the sort $t{:}F$, and sequents of the form $\bot, \Gamma \Rightarrow F$.

Rules of $\mathcal{ILPG}$ are

$$\frac{A, B, \Gamma \Rightarrow C}{A \wedge B, \Gamma \Rightarrow C}\,(L\wedge) \qquad\qquad \frac{\Gamma \Rightarrow A \qquad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B}\,(R\wedge)$$

$$\frac{A,\Gamma \Rightarrow C \qquad B,\Gamma \Rightarrow C}{A\vee B,\Gamma \Rightarrow C}\ (L\vee) \qquad \frac{\Gamma \Rightarrow A_i}{\Gamma \Rightarrow A_0\vee A_1}\ (R\vee)\quad (i=0,1)$$

$$\frac{\Gamma \Rightarrow A \qquad B,\Gamma \Rightarrow C}{A\to B,\Gamma \Rightarrow C}\ (L\to) \qquad \frac{A,\Gamma \Rightarrow B}{\Gamma \Rightarrow A\to B}\ (R\to)$$

$$\frac{A,A,\Gamma \Rightarrow C}{A,\Gamma \Rightarrow C}\ (LC) \qquad \frac{\Gamma \Rightarrow A \qquad A,\Gamma' \Rightarrow B}{\Gamma,\Gamma' \Rightarrow B}\ (Cut)$$

$$\frac{A,\Gamma \Rightarrow B}{t{:}A,\Gamma \Rightarrow B}\ (L:) \qquad \frac{\Gamma \Rightarrow t{:}A}{\Gamma \Rightarrow\ !t{:}t{:}A}\ (R!)$$

$$\frac{\Gamma \Rightarrow t{:}A}{\Gamma \Rightarrow \Delta,(t+s){:}A}\ (Rl+) \qquad \frac{\Gamma \Rightarrow t{:}A}{\Gamma \Rightarrow \Delta,(s+t){:}A}\ (Rr+)$$

$$\frac{\Gamma \Rightarrow s{:}(A\to B) \qquad \Gamma \Rightarrow t{:}A}{\Gamma \Rightarrow (s\cdot t){:}B}\ (R\cdot) \qquad \frac{\begin{matrix}\mathcal{D}\\ \Gamma \Rightarrow \mathbf{A}\end{matrix}}{\Gamma \Rightarrow c{:}\mathbf{A}}\ (Rc),$$

where in (Rc)-rule $\mathbf{A}$ is an axiom *A0 – A4* of the Hilbert style system for $\mathcal{ILP}$, c is a proof constant and $\mathcal{D}$ is the *standard derivation of* $\Gamma \Rightarrow \mathbf{A}$. Under the standard derivation here we mean the following. If $\mathbf{A}$ is *A0* (i.e. a propositional axiom), then $\mathcal{D}$ is the straightforward cut-free derivation of $\Gamma \Rightarrow \mathbf{A}$ in the Gentzen style system for $\mathcal{I}nt$. For axioms *A1 – A4* the standard derivations are respectively

$$\cfrac{\cfrac{\begin{matrix}\mathcal{D}'\\ F,\Gamma \Rightarrow F\end{matrix}}{t{:}F,\Gamma \Rightarrow F}}{\Gamma \Rightarrow t{:}F\to F}\ ,$$

where $\mathcal{D}'$ is the straightforward cut-free derivation of $F,\Gamma \Rightarrow F$ in the Gentzen style system for $\mathcal{I}nt$;

$$\cfrac{\cfrac{s{:}(F\to G),t{:}F,\Gamma \Rightarrow s{:}(F\to G) \qquad s{:}(F\to G),t{:}F,\Gamma \Rightarrow t{:}F}{s{:}(F\to G),t{:}F,\Gamma \Rightarrow (s\cdot t){:}G}}{\Gamma \Rightarrow s{:}(F\to G)\to(t{:}F\to(s\cdot t){:}G)}\ ;$$

$$\cfrac{\cfrac{t{:}F,\Gamma \Rightarrow t{:}F}{t{:}F,\Gamma \Rightarrow\ !t{:}t{:}F}}{\Gamma \Rightarrow t{:}F\to !t{:}t{:}F}\ ; \qquad \cfrac{\cfrac{t{:}F,\Gamma \Rightarrow t{:}F}{t{:}F,\Gamma \Rightarrow (t+s){:}F}}{\Gamma \Rightarrow t{:}F\to(t+s){:}F}\ .$$

Under $\mathcal{ILPG}^-$ we mean a cut-free fragment of $\mathcal{ILPG}$.

7.1 Theorem. *Cut elimination holds for $\mathcal{ILP}$.*

Proof. We shall deliver a syntactical proof that $\mathcal{ILPG} \vdash \Gamma \Rightarrow A$ yields $\mathcal{ILPG}^- \vdash \Gamma \Rightarrow A$.

7.2 Definition. A *level* of a cut is the sum of the depths of the deductions of the premises. The *rank* $rk(A, \mathcal{D})$ of a given occurrence of A in a derivation $\mathcal{D}$ is defined by the following induction on the depth of this occurrence in $\mathcal{D}$. For a term or a formula X by $|X|$ we denote the total number of occurrences of propositional, proof variables and constants, propositional and functional symbols in X. If $X \in \{P, \bot, \Gamma\}$ in a derivation $\mathcal{D}$ consisting of an axiom $P, \Gamma \Rightarrow P$ or $\bot, \Gamma \Rightarrow F$, then $rk(X, \mathcal{D}) = |X|$.

For all the rules of $\mathcal{ILPG}$ ranks of the corresponding occurrences of the side formulas coincide. For the rule(L$\wedge$)

$$rk(A \wedge B, \mathcal{D}) \;=\; rk(A, \mathcal{D}) + rk(B, \mathcal{D}) + 1.$$

Likewise for the rule (R$\wedge$), (L$\vee$) and (R$\to$).

For $(R\vee)$, case $j = 0$,

$$rk(A_0 \vee A_1, \mathcal{D}) \;=\; rk(A_0, \mathcal{D}) + |A_1| + 1,$$

similarly for $j = 1$.

For $(L \to)$

$$rk(A \to B, \mathcal{D}) \;=\; rk(A, \mathcal{D}) + rk(B, \mathcal{D}) + 1.$$

For $(L\,:)$

$$rk(t{:}A, \mathcal{D}) \;=\; rk(A, \mathcal{D}) + |t|.$$

For $(R!)$

$$rk(!t{:}t{:}A, \mathcal{D}) \;=\; rk(t{:}A, \mathcal{D}) + |!t|.$$

For $(Rl+)$

$$rk((t+s){:}A, \mathcal{D}) \;=\; rk(t{:}A, \mathcal{D}) + |s| + 1.$$

For $(Rr+)$

$$rk((t+s){:}A, \mathcal{D}) \;=\; rk(s{:}A, \mathcal{D}) + |t| + 1.$$

For $(R\cdot)$

$$rk((s\cdot t){:}B, \mathcal{D}) \;=\; rk(s{:}(A \to B), \mathcal{D}) + rk(t{:}A, \mathcal{D}) + |(s\cdot t){:}B|.$$

For (Rc)

$$rk(c{:}\mathbf{A}, \mathcal{D}) \;=\; rk(\mathbf{A}, \mathcal{D}) + |1|.$$

Note that $rk(\mathbf{A}, \mathcal{D}) = |\mathbf{A}|$.

For (LC) the rank of the occurrence of A in the conclusion of the rule is the maximum rank of the indicated occurrences of A in the premise sequent. The *rank of the cut rule*

$$\frac{\Gamma \Rightarrow A \qquad A, \Gamma' \Rightarrow B}{\Gamma, \Gamma' \Rightarrow B}\,(Cut)$$

is the maximum rank of the indicated occurrences of A in the premise sequents. The *cutrank* of the deduction $\mathcal{D}$ is the maximum of the ranks of the cuts occurring in $\mathcal{D}$.

From the definitions it follows easily that

1. $|A| \leq rk(A, \mathcal{D})$ and $rk(A, \mathcal{D}) = |A|$ if $\mathcal{D}$ does not use the rule (R$\cdot$).

2. $rk(A, \mathcal{D})$ monotonically increases for the related occurrences of A with the increase of depth.

3. Let $\Gamma \Rightarrow \Delta$ be an occurrence of a sequent in a derivation $\mathcal{D}$. Let $\mathcal{D}'$ be a a subderivation of $\Gamma \Rightarrow \Delta$ in $\mathcal{D}$. Suppose $\mathcal{D}''$ is another derivation of $\Gamma \Rightarrow \Delta$ such that

$$rk(X, \mathcal{D}'') \leq rk(X, \mathcal{D}')$$

for each occurrence of a formula X in $\Gamma \Rightarrow \Delta$. If we replace $\mathcal{D}'$ by $\mathcal{D}''$ in $\mathcal{D}$, then it will not increase the ranks of formulas in $\mathcal{D}$ outside $\mathcal{D}''$.

7.3 Lemma. (Rank- and depth-preserving invertibility of the rule (R$\rightarrow$)). *If $\mathcal{D}$ is a derivation of $\Gamma \Rightarrow A \rightarrow B$, then there is a derivation $\mathcal{D}'$ of $A, \Gamma \Rightarrow B$ such that*

1. the depth of $\mathcal{D}'$ is not greater, then the depth of $\mathcal{D}$,

2. the cutrank of $\mathcal{D}'$ equals to the cutrank of $\mathcal{D}$,

3. $rk(F, \mathcal{D}') \;=\; rk(F, \mathcal{D})$ for all formulas from Γ,

4. $rk(A, \mathcal{D}') + rk(B, \mathcal{D}') + 1 \;=\; rk(A \rightarrow B, \mathcal{D})$.

Proof. An induction on the depth of $\mathcal{D}$. The base case corresponds to an axiom. Since $A \rightarrow B$ is neither atomic nor of the form $t{:}F$ the case when $A \rightarrow B$ is a principal formula of an axiom is impossible. If $\mathcal{D}$ is an axiom $\perp, \Delta \Rightarrow A \rightarrow B$, then put $\mathcal{D}'$ to be $\perp, A, \Delta \Rightarrow B$. For the induction step consider two possibilities. If $A \rightarrow B$ is the side formula of the last rule in $\mathcal{D}$, then using induction hypothesis, replace $\Delta \Rightarrow A \rightarrow B$ in the premise(s) of the last rule by $A, \Delta \Rightarrow B$. If $A \rightarrow B$ is the principal formula of the last rule in $\mathcal{D}$, then the deduction ends with

$$\frac{\begin{matrix}\mathcal{D}_1\\ A, \Gamma \Rightarrow B\end{matrix}}{\Gamma \Rightarrow A \rightarrow B}\ .$$

In this case put $\mathcal{D}'$ to be $\mathcal{D}_1$.

◀

7.4 Lemma. (Stripping Lemma) *Let $\mathcal{D}$ be a cut-free derivation of $\Gamma \Rightarrow t{:}A$. Then there is a derivation $\mathcal{D}'$ of $\Gamma \Rightarrow A$ such that*

1. the cutrank of $\mathcal{D}'$ is less then the rank of the indicated occurrence of $t{:}A$ in $\mathcal{D}$,

2. $rk(F,\mathcal{D}') = rk(F,\mathcal{D})$ for all formulas from Γ of the end sequent,

3. $rk(A,\mathcal{D}') < rk(t{:}A,\mathcal{D})$ for A and $t{:}A$ being the antecedents of the end sequents of $\mathcal{D}'$ and $\mathcal{D}$ respectively.

Proof. Induction on the depth of $\mathcal{D}$. If $\mathcal{D}$ is an axiom $t{:}F,\Delta \Rightarrow t{:}F$, then let $\mathcal{D}'$ be the derivation

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ F,\Gamma \Rightarrow F\end{array}}{t{:}F,\Gamma \Rightarrow F}\,(\mathrm{L}{:})\ ,$$

where $\mathcal{D}_1$ is a standard cut-free derivation of $F,\Gamma \Rightarrow F$. Note, that such a derivation does not use the rule $(R\cdot)$, therefore $rk(X,\mathcal{D}') = |X| = rk(X,\mathcal{D})$ for all formulas from Γ. Likewise, $rk(F,\mathcal{D}') = |F| < rk(t{:}F,\mathcal{D})$.

The induction step. The case when $t{:}A$ is a side formula of the last rule in $\mathcal{D}$ is trivial. Let $t{:}A$ be the principal formula of the last rule $(R!)$ in $\mathcal{D}$, then the deduction ends with

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma \Rightarrow t{:}A\end{array}}{\Gamma \Rightarrow\, !t{:}t{:}A}\ .$$

In this case $\mathcal{D}_1$ is a cut-free derivations sutisfying also the requirements 2. and 3. of the lemma.

If the last rule in $\mathcal{D}$ is $(Rl+)$, then the deduction ends with

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma \Rightarrow t{:}A\end{array}}{\Gamma \Rightarrow (t+s){:}A}\ .$$

By the induction hypothesis, there exists a derivation $\mathcal{D}_1'$ of $\Gamma \Rightarrow A$ satisfying the lemma's conditions for the derivation $\mathcal{D}_1$. Put $\mathcal{D}'$ to be $\mathcal{D}_1'$. The case $(Rr+)$ can be treated similarly.

If the last rule in $\mathcal{D}$ is $(R\cdot)$, then the deduction ends with

$$\dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ \Gamma \Rightarrow s{:}(A\to B) & \Gamma \Rightarrow t{:}A\end{array}}{\Gamma \Rightarrow (s\cdot t){:}B}\ .$$

By the induction hypothesis, there exist derivations $\mathcal{D}_1'$ of $\Gamma \Rightarrow A \to B$ and $\mathcal{D}_2'$ of $\Gamma \Rightarrow A$ satisfying the lemma's conditions. Take the derivation $\mathcal{D}_1''$ of $A,\Gamma \Rightarrow B$ from the inversion lemma 7.3 and combine the new derivation $\mathcal{D}_3$

$$\dfrac{\begin{array}{cc} \mathcal{D}_2' & \mathcal{D}_1'' \\ \Gamma \Rightarrow A & A, \Gamma \Rightarrow B \end{array}}{\Gamma, \Gamma \Rightarrow B} \quad .$$

Using the contraction (LC) we get the desired derivation $\mathcal{D}'$ of $\Gamma \Rightarrow B$. It is easy to check that all the requirements of the lemma are met.

If the last rule in $\mathcal{D}$ is (Rc), then $\mathcal{D}$ is

$$\dfrac{\begin{array}{c} \mathcal{D}_1 \\ \Gamma \Rightarrow \mathbf{A} \end{array}}{\Gamma \Rightarrow c\!:\!\mathbf{A}} \; .$$

Let $\mathcal{D}'$ be $\mathcal{D}_1$.

◀

Now we return to the proof of theorem 7.1. Our strategy is to eliminate the uppermost cuts. In order to save expositions of some well known constructions we will refer to the corresponding steps of the proof of the cut elimination theorem 4.1.2 from [49] when convenient.

7.5 Lemma. *Let $\mathcal{D}$ be a derivation ending in a cut*

$$\dfrac{\begin{array}{cc} \mathcal{D}_1 & \mathcal{D}_2 \\ \Gamma \Rightarrow A & A, \Gamma' \Rightarrow B \end{array}}{\Gamma, \Gamma' \Rightarrow B}$$

such that $\mathcal{D}$ contains no other cuts. Then we can transform $\mathcal{D}$ into a derivation $\mathcal{D}'$ of the same sequent $\Gamma, \Gamma' \Rightarrow B$ such that cutrank($\mathcal{D}'$)<cutrank($\mathcal{D}$) $= \max\{rk(A,\mathcal{D}_1), rk(A,\mathcal{D}_2)\}$ without an increase of the ranks of the formulas from Γ, Γ', B.

Proof. An induction on the rank of the cut rule, with a subinduction on its level. There are then three possibilities:

1. at least one of $\mathcal{D}_1$, $\mathcal{D}_2$ is an axiom $P, \Gamma \Rightarrow P$ or $\bot, \Gamma \Rightarrow F$;
2. not 1. and the cutformula is not principal in at least one of the premises;
3. not 1. and the cutformula is principal on both sides.

Case 1. Cut can be eliminated by the standard reductions ([49]).

Case 2. We permute the cut upward in a stantard way (cf.[49]) without changing its rank as well as the ranks of all formulas in the end sequent of the derivation, until we find ourselves in situations number 1 or number 3.

Case 3. The cutformula is principal in both premises and neither of the premises an axiom. The induction hypothesis is that the claim of lemma has been shown for all cuts of rank less than $rk(A,\mathcal{D})$ and of rank equal $rk(A,\mathcal{D})$, but level less than the one of the given cut.

The rules corresponding to propositional connectives are treated in a usual way (cf.[49]). There is one additional concern here compared to [49]: we have to make sure that our reductions do not increase the ranks of the side formulas from Γ, Γ'. As an example, consider the case $(R \rightarrow)$. The original deduction is

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}_0\\ \Gamma, A \Rightarrow B\end{matrix}}{\Gamma \Rightarrow A \rightarrow B} \qquad \dfrac{\begin{matrix}\mathcal{D}_1\\ \Gamma' \Rightarrow A\end{matrix} \quad \begin{matrix}\mathcal{D}_2\\ \Gamma', B \Rightarrow C\end{matrix}}{\Gamma', A \rightarrow B \Rightarrow C}}{\Gamma, \Gamma' \Rightarrow C}$$

This is transformed into

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}_1\\ \Gamma' \Rightarrow A\end{matrix} \quad \begin{matrix}\mathcal{D}_0\\ \Gamma, A \Rightarrow B\end{matrix}}{\Gamma', \Gamma \Rightarrow B} \qquad \begin{matrix}\mathcal{D}_2\\ \Gamma', B \Rightarrow C\end{matrix}}{\Gamma', \Gamma', \Gamma, \ \Rightarrow C} \ .$$

We have eliminated the old cut but have got two new cuts instead, both having lower ranks. After a number of contractions in the end sequent we get the desired derivation.

The new cases emerge when the cutformula is of the form $t{:}F$. In all those cases the right premise is just introduced by (L:). So, we distinguish the cases by their left premises.

Case $(R!)$. The derivation is

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}_1\\ \Gamma \Rightarrow t{:}A\end{matrix}}{\Gamma \Rightarrow !t{:}t{:}A} \qquad \dfrac{\begin{matrix}\mathcal{D}_2\\ t{:}A, \Gamma' \Rightarrow C\end{matrix}}{!t{:}t{:}A, \Gamma \Rightarrow C}}{\Gamma, \Gamma' \Rightarrow C}$$

This is transformed into a derivation with a lower ranked cut

$$\dfrac{\begin{matrix}\mathcal{D}_1\\ \Gamma \Rightarrow t{:}A\end{matrix} \qquad \begin{matrix}\mathcal{D}_2\\ t{:}A, \Gamma' \Rightarrow C\end{matrix}}{\Gamma, \Gamma' \Rightarrow C} \ ,$$

Case (Rc) is treated is a similar way.

Case $(Rl+)$ (and $(Rr+)$ by a similar argument). The derivation

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma \Rightarrow t{:}A\end{array}}{\Gamma \Rightarrow (t+s){:}A} \qquad \dfrac{\begin{array}{c}\mathcal{D}_2\\ A,\Gamma' \Rightarrow C\end{array}}{(t+s){:}A,\Gamma \Rightarrow C}}{\Gamma,\Gamma' \Rightarrow C}$$

should be transformed into one with a lower cutrank

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma \Rightarrow t{:}A\end{array} \qquad \dfrac{\begin{array}{c}\mathcal{D}_2\\ A,\Gamma' \Rightarrow C\end{array}}{t{:}A,\Gamma \Rightarrow C}}{\Gamma,\Gamma' \Rightarrow C}$$

Case $(R\cdot)$:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_0\\ \Gamma \Rightarrow s{:}(A\to B)\end{array} \qquad \begin{array}{c}\mathcal{D}_1\\ \Gamma \Rightarrow t{:}A\end{array}}{\Gamma \Rightarrow (s\cdot t){:}B} \qquad \dfrac{\begin{array}{c}\mathcal{D}_2\\ B,\Gamma' \Rightarrow C\end{array}}{(s\cdot t){:}B,\Gamma' \Rightarrow C}}{\Gamma,\Gamma' \Rightarrow C}$$

We transform it into a derivation with a lower cutrank as follows. By Stripping Lemma (7.4) without a rank increase we get derivations

$$\begin{array}{c}\mathcal{D}'_0\\ \Gamma \Rightarrow A\to B\end{array} \qquad\qquad \begin{array}{c}\mathcal{D}'_1\\ \Gamma \Rightarrow A\end{array}$$

From $\mathcal{D}'_0$ by 7.3 without the rank increase of the side formulas we get a derivation

$$\begin{array}{c}\mathcal{D}''_0\\ \Gamma, A \Rightarrow B\end{array}$$

where

$$rk(A,\mathcal{D}''_0),\ rk(B,\mathcal{D}''_0) < rk(A\to B,\mathcal{D}'_0) < rk(s{:}(A\to B),\mathcal{D}_0) < rk((s\cdot t){:}B,\mathcal{D}).$$

The transformed derivation in this case will be

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}'_1\\ \Gamma \Rightarrow A\end{array} \quad \begin{array}{c}\mathcal{D}''_0\\ A,\Gamma \Rightarrow B\end{array}}{\Gamma,\Gamma \Rightarrow B} \qquad \begin{array}{c}\mathcal{D}_2\\ B,\Gamma' \Rightarrow C\end{array}}{\Gamma,\Gamma,\Gamma' \Rightarrow C}$$

Again, use some contractions in the end sequent to get the desired derivation. We have eliminated the old cut and have created two new ones and, may be, some more in $\mathcal{D}_1'$ and $\mathcal{D}_0''$ as a result of the Stripping. By 7.3 and 7.4 all new cuts have lower rank.

This ends the proof of lemma 7.5.

◀

7.6 Lemma. *Let a derivation $\mathcal{D}$ contains not more than one use of the cut rule. Then by a finite chain of reductions it can be transformed into a cut-free derivation $\mathcal{D}'$ of the same end sequent without changing the ranks of the formulas from the end sequent.*

Proof. An induction on the $n =$*cutrank*$(\mathcal{D})$. The base case $n = 0$. Then $\mathcal{D}$ is already cut-free. The induction step. Assume $n > 0$, thus, $\mathcal{D}$ contains a cut. Without loss of generality assume that the cut rule is the last rule in $\mathcal{D}$. By Lemma 7.5 transform $\mathcal{D}$ into $\mathcal{D}_1$ with *cutrank*$(\mathcal{D}_1)$ $<$*cutrank*$(\mathcal{D})$. Beginning with the uppermost cuts in $\mathcal{D}_1$ eliminate them all using the induction hypothesis.

◀

To conclude the proof of Theorem 7.1 use Lemma 7.6 to eliminate every cut in a given derivation beginning with the uppermost ones.

◀

8 Natural deduction system and λ-terms for Logic of Proofs

8.1 Definition. The *natural deduction system $\mathcal{LPN}$ for $\mathcal{LP}$* is obtained from a usual natural deduction system for propositional logic (cf. [14], [49]) in the language of $\mathcal{LP}$ extended by the following rules

$$\frac{s\colon(A \to B) \qquad t\colon A}{(s \cdot t)\colon B}\ (\cdot I) \qquad\qquad \frac{t\colon A}{A}\ (:E) \qquad\qquad \frac{t\colon A}{!t\colon t\colon A}\ (!I)$$

$$\frac{t\colon A}{(t+s)\colon A}\ (+I) \qquad\qquad \frac{t\colon A}{(s+t)\colon A}\ (+I) \qquad\qquad \frac{\begin{array}{c}\mathcal{D}\\ \mathbf{A}\end{array}}{c\colon \mathbf{A}}\ (cI)$$

where $\mathbf{A}$ is an axiom of the Hilbert version of $\mathcal{LP}$ (Section 1), c is a proof constant, and $\mathcal{D}$ is the *standard derivation* of $\mathbf{A}$. Under the standard derivation of $\mathbf{A}$ we mean the following.

If **A** is an axiom *A0*, then $\mathcal{D}$ is the straightforward normal derivation of **A** in the natural deduction system for $\mathcal{I}nt$. For other axioms *A1* – *A4* the standard derivations are

$$\dfrac{\dfrac{[t\!:\!A]}{A}}{t\!:\!A \to A} \qquad \dfrac{\dfrac{\dfrac{[s\!:\!(A\to B)] \quad [t\!:\!A]}{(s\cdot t)\!:\!B}}{t\!:\!A\to(s\cdot t)\!:\!B}}{s\!:\!(A\to B)\to(t\!:\!A\to(s\cdot t)\!:\!B)}$$

$$\dfrac{\dfrac{[t\!:\!A]}{!t\!:\!t\!:\!A}}{t\!:\!A\to\, !t\!:\!t\!:\!A} \qquad \dfrac{\dfrac{[t\!:\!A]}{(t+s)\!:\!A}}{t\!:\!A\to(t+s)\!:\!A}$$

Note that a standard derivation has no undischarged premises. Under $\mathcal{LPN} \vdash \Gamma \Rightarrow A$ we mean "*A is derivable in $\mathcal{LPN}$ from assumptions* Γ".

A standard theorem relating Hilbert, Gentzen and natural style derivations in $\mathcal{LP}$ holds. Namely the following are equivalent

1. $\Gamma \vdash_{\mathcal{LP}} A$
2. $\mathcal{LPG} \vdash \Gamma \Rightarrow A$
3. $\mathcal{LPN} \vdash \Gamma \Rightarrow A$.

This fact is established by the standard mutual simulations of derivations in all three systems (cf. section 3.3 in [49]). In fact for any source derivation of size s the simulation runs in polynomial time and produces a derivation of size $O(s)$.

The following analog of the Lifting lemma 1.4 holds for $\mathcal{LPN}$.

8.2 Corollary. *If* $\vdash \vec{s}\!:\!\Gamma, \Delta \Rightarrow A$, *then for any proof variables* $\vec{y}$ *one can construct a proof polynomial* $t(\vec{x}, \vec{y})$ *such that* $\mathcal{LPN} \vdash \vec{s}\!:\!\Gamma, \vec{y}\!:\!\Delta \Rightarrow t(\vec{s}, \vec{y})\!:\!A$.

Proof. Induction of the depth of a derivation. A straightforward natural deduction version of Lemma 1.4. The number of steps in the algorithm constructing $\mathcal{D}'$ is bounded by a polynomial of the length of $\mathcal{D}$.
◀

8.3 Definition. *Contractions* for $\mathcal{LPN}$ include all usual contractions for propositional logic ($\wedge-, \vee-, \to$*-contractions, permutation contractions*) (cf. section 6.1.3 in [49]), and the new contractions

$\cdot$*-contraction*

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}_1 \\ s:(A\to B)\end{matrix} \quad \begin{matrix}\mathcal{D}_2 \\ t:B\end{matrix}}{(s\cdot t):B}}{B} \qquad \textit{transforms into} \qquad \dfrac{\dfrac{\begin{matrix}\mathcal{D}_1 \\ s:(A\to B)\end{matrix}}{A\to B} \qquad \dfrac{\begin{matrix}\mathcal{D}_2 \\ t:A\end{matrix}}{A}}{B},$$

$+$*-contraction*

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D} \\ t:A\end{matrix}}{(t+s):A}}{A} \qquad \textit{transforms into} \qquad \dfrac{\begin{matrix}\mathcal{D} \\ t:A\end{matrix}}{A},$$

$!$*-contraction*

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D} \\ t:A\end{matrix}}{!t:t:A}}{t:A} \qquad \textit{transforms into} \qquad \begin{matrix}\mathcal{D} \\ t:A,\end{matrix}$$

c-contraction

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D} \\ \mathbf{A}\end{matrix}}{c:\mathbf{A}}}{\mathbf{A}} \qquad \textit{transforms into} \qquad \begin{matrix}\mathcal{D} \\ \mathbf{A}.\end{matrix}$$

An obvious new permutational contraction should also be added that allow "pulling" the $(:E)$ upwards through the $(\vee E)$ rule. An $\mathcal{LPN}$-derivation $\mathcal{D}$ is normal if no contraction is possible anywhere in $\mathcal{D}$.

8.4 Theorem. *Normalization holds for $\mathcal{LPN}$*

Proof. Given a derivation $\mathcal{D}$ in $\mathcal{LPN}$ of the type $\Gamma \Rightarrow A$ one can construct a normal derivation $\mathcal{D}'$ of the same type $\Gamma \Rightarrow A$. Indeed, for a given derivation of $\Gamma \Rightarrow A$ in $\mathcal{LPN}$ one can construct an $\mathcal{LPG}$-derivation of a sequent $\Gamma \Rightarrow A$. By 5.1, there exists a cut-free $\mathcal{LPG}$-derivation of $\Gamma \Rightarrow A$. A usual transformation of a cut-free $\mathcal{LPG}$-derivation into an $\mathcal{LPN}$-derivation

produces a normal $\mathcal{LPN}$-derivation (cf. section 6.3 in [49]). Since both of the transitions from $\mathcal{LPN}$ derivations and back are computable (and feasible), the entire efficiency of the process of proof normalization in $\mathcal{LPN}$ in this proof is determined by an efficiency of normalization of proofs in $\mathcal{LPG}$. An easy adaptation of a "constructive" normalization theorem 8.7 for $\mathcal{ILPN}$ (definition 8.5) independently establishes a direct algorithm of normalizing derivations in $\mathcal{LPN}$ as well.
◄

One can easily give a direct realization algorithm for the normal deduction system for $\mathcal{S}4$ in $\mathcal{LPN}$ based on 6.2.

8.5 Definition. Under $\mathcal{ILPN}$ we mean an intuitionistic version of of $\mathcal{LPN}$ which is obtained from $\mathcal{LPN}$ by omitting the double negation rule

$$\frac{\begin{array}{c}[\neg A]\\ \mathcal{D}\\ \bot\end{array}}{A}$$

8.6 Theorem. (From Gentzen to normal deductions in the intuitionistic $\mathcal{LP}$)

$$\mathcal{ILPG} \vdash \Gamma \Rightarrow A \qquad \textit{if and only if} \qquad \mathcal{ILPN} \vdash \Gamma \Rightarrow A.$$

Moreover, a cut-free derivation in $\mathcal{ILPG}$ transforms into a normal derivation in $\mathcal{ILPN}$.

Proof. A usual argument in the style of Section 6.3 from [49].
◄

8.7 Theorem. *Normalization holds for $\mathcal{ILPN}$*

Proof. Take a derivation of the sort $\Gamma \Rightarrow A$ in $\mathcal{ILPN}$, transform it into a derivation of $\Gamma \Rightarrow A$ in $\mathcal{ILPG}$, perform a cut elimination, and transform the resulting cut-free proof back into an $\mathcal{ILPN}$ derivation. By Theorem 8.6, the resulting derivation is normal. One could write down a direct algorithm of normalization of derivations in $\mathcal{ILPN}$ that will essentially repeat the reduction steps for cut elimination in $\mathcal{ILPG}$. Moreover, on the basis of the reductions from the proof of Theorem 7.1 one could establish a strong normalization property of $\mathcal{ILPG}$ and $\mathcal{ILPN}$.
◄

Extending the term calculus for the intuitionistic logic ([49]) we can identify the full $\mathcal{ILPN}$ with a system of typed λ-terms $\mathcal{ILPN}\lambda$ in a natural way. In $\mathcal{ILPN}\lambda$ λ-terms have the format $t\!:\!F$ where the type F is an $\mathcal{LP}$-formula, and the term t is built from the proof variables by specific operations (below).

8.8 Definition. We define a λ-term calculus $\mathcal{ILPN}\lambda$ for the full $\mathcal{ILPN}$. The language of $\mathcal{ILPN}\lambda$ has only formulas of the type $t\!:\!F$ where F is an $\mathcal{LP}$-formula, and t is a term built from the proof variables by atomic operations $\mathbf{p}$, $\mathbf{p}_j$, $\mathbf{k}_j$, $E^{\vee}_{u,v}$, $E^{\perp}_{A}$, App, $\mathbf{P}$, $\mathbf{U}$, $\mathbf{B}$, $\mathbf{S}_j$, $\mathbf{C}$, (j=0,1), and λ-abstraction. The arities of the operations will be made clear in the rules. The first eight clauses of come directly from the term calculus for $\mathcal{I}nt$ ([49], 2.2.2). We will omit an obvious description of free and bounded variables. As usual, $[A]$ denotes a discharged premise A. In the derivations denoted in this definition by

$$\begin{array}{c}[w\!:\!F]\\ \vdots\\ p\!:\!G\end{array}$$

the variable w occurs free neither in F, G nor in any undischarged premise of the derivation.

axiom $y\!:\!F$ (y is a variable)

$$\frac{t\!:\!\perp}{E^{\perp}_{A}(t)\!:\!A}\ (\perp Et)$$

$$\frac{s\!:\!A \qquad t\!:\!B}{\mathbf{p}(s,t)\!:\!(A\wedge B)}\ (\wedge It)$$

$$\frac{t\!:\!(A_0\wedge A_1)}{\mathbf{p_j}(t)\!:\!A_j}\ j\in\{0,1\},\ (\wedge Et)$$

$$\frac{t\!:\!A_j}{\mathbf{k}_j(t)\!:\!(A_0\vee A_1)}\ j\in\{0,1\},\ (\vee It)$$

$$\frac{t\!:\!(A\vee B) \qquad \begin{array}{c}[u\!:\!A]\\ \vdots\\ s\!:\!C\end{array} \qquad \begin{array}{c}[v\!:\!B]\\ \vdots\\ s'\!:\!C\end{array}}{E^{\vee}_{u,v}(t,s,s')\!:\!C}\ (\vee Et)$$

$$\frac{\begin{array}{c}[u\!:\!A]\\ \vdots\\ t\!:\!B\end{array}}{\lambda u.t\!:\!(A\rightarrow B)}\ (\rightarrow It)$$

$$\frac{s\!:\!(A\rightarrow B) \qquad t\!:\!A}{App(s,t)\!:\!B}\ (\rightarrow Et)$$

$$\frac{q\!:\!s\!:\!(A\to B)\qquad r\!:\!t\!:\!A}{\mathbf{P}(q,r)\!:\!(s\cdot t)\!:\!B}\ (\cdot It)\qquad\qquad \frac{q\!:\!t\!:\!A}{\mathbf{U}(q)\!:\!A}\ (:Et)$$

$$\frac{q\!:\!t\!:\!A}{\mathbf{B}(q)\!:\,!t\!:\!t\!:\!A}\ (!tI)\qquad\qquad \frac{q\!:\!t_j\!:\!A}{\mathbf{S}_j(q)\!:\!(t_0+t_1)\!:\!A}\ j\in\{0,1\},\ \ (+I)$$

Note that the list of rules above suffices to build a λ-term p without free variables which internalize in $\mathcal{ILPN}\lambda$ the standard $\mathcal{ILPN}$-derivation of an axiom $\mathbf{A}$. In particular, $\mathcal{ILPN}\lambda \vdash p\!:\!\mathbf{A}$. For example, the λ-term version of the standard derivation of axiom *A3* is

$$\cfrac{\cfrac{v\!:\!t\!:\!F}{\mathbf{B}(v)\ !t\!:\!t\!:\!F}}{\lambda v.\mathbf{B}(v)\!:\!(t\!:\!F\to\, !t\!:\!t\!:\!F)}$$

The last rule of $\mathcal{ILPN}\lambda$ is

$$\frac{\begin{array}{c}\widetilde{\mathcal{D}}\\ p\!:\!\mathbf{A}\end{array}}{\mathbf{C}(p)\!:\!c\!:\!\mathbf{A}}\ (cIt),$$

where $\widetilde{\mathcal{D}}$ is the λ-term version of the standard derivation of $\mathbf{A}$ in $\mathcal{ILPN}$.

8.9 Definition. In the term notation the *contractions* for the $\mathcal{ILPN}\lambda$ are

1. $\mathbf{p}_j(\mathbf{p}(t_0,t_1))$ *cont* t_j $(j\in\{0,1\})$,
2. $E^{\vee}_{x_0,x_1}(\mathbf{k}_j t, t_0, t_1)$ *cont* $t_j[x/t]$,
3. $App(\lambda x.t, s)$ *cont* $t[x/s]$,
4. $\mathbf{U}(\mathbf{B}(t))$ *cont* t,
5. $\mathbf{U}(\mathbf{C}(t))$ *cont* t,
6. $\mathbf{U}(\mathbf{P}(t_0,t_1))$ *cont* $App(\mathbf{U}(t_0),\mathbf{U}(t_1))$,
7. $\mathbf{U}(\mathbf{S}_j(t))$ *cont* $\mathbf{U}(t)$,
8. $f[E^{\vee}_{x_0,x_1}(\mathbf{k}_j t, t_0, t_1)]$ *cont* $E^{\vee}_{x_0,x_1}(\mathbf{k}_j t, f[t_0], f[t_1])$, where f is another eliminating operator (i.e. one of $\mathbf{p}_j$, *App*, $\mathbf{U}$).

The contractions 1 - 5 are the called *detour contractions*, 6 - 8 are *permutation contractions*. A λ-term t is *normal* if no contractions are possible in t.

It follows from the definitions that

$$\mathcal{ILPN} \vdash \Gamma \Rightarrow A \qquad \textit{iff} \qquad \mathcal{ILPN}\lambda \vdash \vec{x}{:}\Gamma \Rightarrow t(\vec{x}){:}A \quad \textit{for some } \mathcal{ILPN}\lambda\textit{-term } t.$$

8.10 Theorem. (Normalization of $\mathcal{ILPN}\lambda$-terms) *Every λ-term for $\mathcal{ILPN}\lambda$ is normalizing.*

Proof. Translate the proof of theorem 8.4 from the language of derivations into the language of λ-terms. In fact one can establish a strong normalization of $\mathcal{ILPN}\lambda$-terms with respect to the contractions 8.9.
◄

9 Abstraction in Logic of Proofs

In this section we show that $\mathcal{LP}$ provides a standard provability semantics for the operator of λ-abstraction. This matches our earlier observation (Section 6), that $\mathcal{S}4$-modality is realized by proof polynomials. Thus modality and λ-terms are objects of the same sort, namely, they are all proof polynomials. Through a realization in $\mathcal{LP}$ both modality and modal lambda terms receive a uniform provability semantics.

The defined abstraction operator λ^*x on proof polynomials below is a natural extension of the defined λ-abstraction operator λ^*x in combinatory logic (cf. [49]).

9.1 Definition. An $\mathcal{ILPN}$-derivation $\mathcal{D}$ is *pure* if it uses no rules other than $(\cdot I)$, (cI), and a *closed version of* $(!I)$ where the principal proof polynomial t contains no variables. It is clear that every pure derivation is normal since it has no elimination rules.

9.2 Lemma. (Definable abstraction) *Let $\mathcal{D}$ be a pure $\mathcal{ILPN}$-derivation of a type*

$$\vec{p}{:}\Gamma, x{:}A \Rightarrow t(x){:}B$$

such that x does not occur in $\vec{p}{:}\Gamma, A, B$. Then one may construct a proof polynomial $\lambda^*x.t(x)$ and a pure $\mathcal{ILPN}$-derivation $\mathcal{D}'$ of the type

$$\vec{p}{:}\Gamma \Rightarrow \lambda^*x.t(x){:}(A \rightarrow B).$$

Proof. The base case is the depth of $\mathcal{D}$ equals one. There are two possibilities.

1. $\mathcal{D}$ is $t(x){:}B$ and $t(x)$ contains an occurrence of x. Then $t(x){:}B = x{:}A$. Indeed, by the definition of a natural derivation of the depth 1, the formula $t(x){:}B$ should occur in $\vec{p}{:}\Gamma, x{:}A$. Since x does not occur in $\vec{p}{:}\Gamma, A, B$ the only remaining possibility is when $t(x){:}B$ coincides with $x{:}A$. Let $\mathcal{D}'$ be the pure derivation (without undischarged premises) of $(a\cdot b\cdot c){:}(A\rightarrow A)$ where a, b, c are proof constants specified by the conditions (cf. [49], section 1.3.6.)

$$\begin{array}{l} a{:}([A\rightarrow((A\rightarrow A)\rightarrow A)]\rightarrow[(A\rightarrow(A\rightarrow A))\rightarrow(A\rightarrow A)]) \\ b{:}[A\rightarrow((A\rightarrow A)\rightarrow A)] \\ c{:}[A\rightarrow(A\rightarrow A)]. \end{array}$$

Let $\lambda^*x.x = (a\cdot b\cdot c)$. In fact this case coincides with the presentation of $\lambda^*x^A.x$ as $\mathbf{s}^{A,A\rightarrow A,A}\mathbf{k}^{A,A\rightarrow A}\mathbf{k}^{A,A}$ in combinatory logic (cf. [49]).

2. $\mathcal{D}$ is $t{:}B$ and t does not contain an occurrence of x. Then $t{:}B \in \vec{p}{:}\Gamma$. Let $\mathcal{D}'$ be

$$\dfrac{\dfrac{\dfrac{\dfrac{[B]}{A\rightarrow B}}{B\rightarrow(A\rightarrow B)}}{b{:}(B\rightarrow(A\rightarrow B))}\,(cI) \qquad t{:}B}{(b\cdot t){:}(A\rightarrow B)}\,(\cdot I)\ .$$

Let $\lambda^*x.t = b\cdot t$. This is the well known equality $\lambda^*x^A.t^B = \mathbf{k}^{B,A}t^B$ of combinatory logic.

The induction step corresponding to the closed $(!I)$ rule is treated similarly to the case 2. Consider the case $(\cdot I)$. Let a derivation $\mathcal{D}$ from the premises $\vec{p}{:}\Gamma, x{:}A$ end with

$$\dfrac{s{:}(Y\rightarrow B) \qquad t{:}Y}{(s\cdot t){:}B}\ .$$

By the induction hypothesis, we have already built pure derivations from the premises $\vec{p}{:}\Gamma$ of $\lambda^*x.s{:}(A\rightarrow(Y\rightarrow B))$ and $\lambda^*x.t{:}(A\rightarrow Y)$. From them we construct a pure derivation $\mathcal{D}'$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1\\(A\rightarrow(Y\rightarrow B))\rightarrow((A\rightarrow Y)\rightarrow(A\rightarrow B))\end{array}}{c{:}((A\rightarrow(Y\rightarrow B))\rightarrow((A\rightarrow Y)\rightarrow(A\rightarrow B)))} \qquad \lambda^*x.s{:}(A\rightarrow(Y\rightarrow B))}{(c\cdot\lambda^*x.s){:}((A\rightarrow Y)\rightarrow(A\rightarrow B))} \qquad \lambda^*x.t{:}(A\rightarrow Y)}{(c\cdot\lambda^*x.s\cdot\lambda^*x.t){:}(A\rightarrow B)}\ ,$$

where $\mathcal{D}_1$ is the standard derivation of a propositional axiom. Let $\lambda^*x.(s\cdot t) = (c\cdot\lambda^*x.s\cdot\lambda^*x.t)$. In combinatory logic notations

$$\lambda^*x^A.s^{Y\rightarrow B}t^Y = \mathbf{s}^{A,Y,B}\lambda^*x.s\lambda^*x.t$$

◀

9.3 Comment. In $\mathcal{ILPN}$ λ-abstraction is presented by a set of proof polynomials depending on a context (e.g. an $\mathcal{ILPN}$-derivation). In this respect the realization from 9.2 of λ-abstraction by proof polynomials is similar the realization of $\mathcal{S}4$-modality which is decomposed in 6.2 into a set of proof polynomials depending on a context (there an $\mathcal{S}4$-derivation).

The operation λ^* suffices to emulate the traditional λ-abstraction. In fact it cannot be easily extended from the pure to more general derivations without sacrificing some desired properties. We need to keep the format $\vec{p}:\Gamma, x:A \Rightarrow t(x):B$ throughout all the derivation $\mathcal{D}$ in order to preserve an inductive character of the definition. The restriction "x does not occur in $\vec{p}:\Gamma, A, B$" is needed to guarantee the correctness of β-conversion for λ^*-abstraction (below), though it rules out $(!I)$. Note, that the rule $(!I)$ does not admit abstraction anyway. Indeed, in $\mathcal{LPN}$ we have

$$x:A \Rightarrow\ !x:x:A,$$

but for no proof polynomial p

$$\Rightarrow p:(A \to x:A)$$

since $A \to x:A$ is not provable in $\mathcal{LP}$.

9.4 Definition. The dual operation to λ-abstraction is *β-conversion*

$$(\lambda x^A.t^B)s^A \quad \longrightarrow_\beta \quad t^B[x^A/s^A].$$

β-conversion is naturally presented as the following transformation of pure derivations in $\mathcal{ILPN}$:

$$\dfrac{\dfrac{\begin{array}{c}[x:A]\\ \vdots\\ t(x):B\end{array}}{\lambda^* x t(x):(A\to B)} \qquad \begin{array}{c}\mathcal{D}\\ s:A\end{array}}{(\lambda^* x t(x)\cdot s):B} \qquad \textit{transforms into} \qquad \begin{array}{c}\mathcal{D}\\ s:A\\ \vdots\\ t(s):B,\end{array}$$

The rule of *η-conversion*

$$(\lambda x^A.t^B)s^A \quad \longrightarrow_\eta \quad t \qquad \textit{if } x \textit{ is not free in } t$$

is treated in the same way. Finally, *α-conversion* corresponds to an obviously valid rule of renaming bounded variables in $\mathcal{ILPN}$-derivations with abstraction.

All other standard λ-term constructors for $\mathcal{I}nt$ can also be realized as operations on proof polynomials. This is a straightforward corollary of the fact that $\mathcal{I}nt$ is a fragment of $\mathcal{ILPN}$ and of the Lifting rule for $\mathcal{ILPN}$. Indeed, if $\mathcal{ILPN} \vdash \Gamma \Rightarrow B$, then by induction on the given proof one can construct a proof polynomial $p(\vec{y})$ such that $\mathcal{LPN} \vdash \vec{y}{:}\Gamma \Rightarrow p(\vec{y}){:}B$. However, for the sake of clear presentation of λ-terms as proof polynomials we will explicitly build the proof polynomials corresponding to standard λ-terms constructors.

9.5 Definition. We define a list of *standard translations* of term constructors from $\mathcal{ILPN}\lambda$ to corresponding derivations in $\mathcal{ILPN}$.

λ-term constructor	*corresponding derivation in $\mathcal{ILPN}$*
$y{:}F$	$y{:}F$
$\dfrac{s{:}A \quad t{:}B}{\mathbf{p}(s,t){:}(A\wedge B)}$	$\dfrac{\dfrac{\begin{matrix}\widetilde{\mathcal{D}}\\ c{:}(A\to(B\to A\wedge B))\end{matrix} \qquad s{:}A}{(c\cdot s){:}(B\to A\wedge B)} \quad t{:}B}{(c\cdot s\cdot t){:}(A\wedge B)}$
$\dfrac{t{:}(A_0\wedge A_1)}{\mathbf{p_j}(t){:}A_j}\ j\in\{0,1\}$	$\dfrac{\begin{matrix}\widetilde{\mathcal{D}}\\ c{:}(A_0\wedge A_1\to A_j)\end{matrix} \qquad t{:}(A_0\wedge A_1)}{(c\cdot t){:}A_j}$
$\dfrac{t{:}A_j}{\mathbf{k}_j(t){:}(A_0\vee A_1)}\ j\in\{0,1\}$	$\dfrac{\begin{matrix}\widetilde{\mathcal{D}}\\ c{:}(A_j\to A_0\vee A_1)\end{matrix} \qquad t{:}A_j}{(c\cdot t){:}(A_0\vee A_1)}$
$\dfrac{t{:}(A\vee B) \quad \begin{matrix}[u{:}A]\\ \vdots\\ s{:}C\end{matrix} \quad \begin{matrix}[v{:}B]\\ \vdots\\ s'{:}C\end{matrix}}{E^{\vee}_{u,v}(t,s,s'){:}C}$	$\dfrac{\dfrac{\dfrac{\mathcal{D}_1 \qquad \mathcal{D}_2}{(c\cdot\lambda^*u.s){:}((B\to C)\to(A\vee B\to C))} \quad \mathcal{D}_3}{(c\cdot\lambda^*u.s\cdot\lambda^*v.s'){:}(A\vee B\to C)} \qquad t{:}(A\vee B)}{(c\cdot\lambda^*u.s\cdot\lambda^*v.s'\cdot t){:}C}$

where $\mathcal{D}_1$ is

$$\begin{matrix}\widetilde{\mathcal{D}}\\ c{:}((A\to C)\to((B\to C)\to(A\vee B\to C)))\end{matrix}$$

$\mathcal{D}_2$ and $\mathcal{D}_3$ are

$$\dfrac{\begin{array}{c}[u\!:\!A]\\ \vdots\\ s\!:\!C\end{array}}{\lambda^{*}u.s\!:\!(A\to C)} \qquad\qquad \dfrac{\begin{array}{c}[v\!:\!B]\\ \vdots\\ s'\!:\!C\end{array}}{\lambda^{*}v.s'\!:\!(B\to C)}$$

$$\dfrac{\begin{array}{c}[u\!:\!A]\\ \vdots\\ t\!:\!B\end{array}}{\lambda u.t\!:\!(A\to B)} \qquad\qquad \dfrac{\begin{array}{c}[u\!:\!A]\\ \vdots\\ t\!:\!B\end{array}}{\lambda^{*}u.t\!:\!(A\to B)}$$

$$\dfrac{s\!:\!(A\to B) \qquad t\!:\!A}{App(s,t)\!:\!B} \qquad\qquad \dfrac{s\!:\!(A\to B) \qquad t\!:\!A}{(s\cdot t)\!:\!B}$$

$$\dfrac{t\!:\!\bot}{E^{\bot}_{A}(t)\!:\!A} \qquad\qquad \dfrac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c\!:\!(\bot\to A)\end{array} \qquad t\!:\!\bot}{(s\cdot t)\!:\!A}$$

$$\dfrac{u\!:\!s\!:\!(A\to B) \quad v\!:\!t\!:\!A}{\mathbf{P}(u,v)\!:\!(s\cdot t)\!:\!B} \qquad \dfrac{\dfrac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c\!:\!(s\!:\!(A\to B)\to(t\!:\!A\to(s\cdot t)\!:\!B))\end{array} \qquad u\!:\!s\!:\!(A\to B)}{(c\cdot u)\!:\!(t\!:\!A\to(s\cdot t)\!:\!B)} \qquad v\!:\!t\!:\!A}{(c\cdot u\cdot v)\!:\!(s\cdot t)\!:\!B}$$

$$\dfrac{v\!:\!t\!:\!A}{\mathbf{U}(v)\!:\!A} \qquad\qquad \dfrac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c\!:\!(t\!:\!A\to A)\end{array} \qquad v\!:\!t\!:\!A}{(c\cdot v)\!:\!A}$$

$$\frac{v\!:\!t\!:\!A}{\mathbf{B}(v)\!:\,!t\!:\!t\!:\!A} \qquad\qquad \frac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c\!:\!(t\!:\!A\!\to !t\!:\!t\!:\!A)\end{array} \qquad v\!:\!t\!:\!A}{(c\cdot v)\!:\,!t\!:\!t\!:\!A}$$

$$\frac{v\!:\!t_j\!:\!A}{\mathbf{S}_j(v)\!:\!(t_0+t_1)\!:\!A}\;j\in\{0,1\} \qquad\qquad \frac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c\!:\!(t_j\!:\!A\!\to\!(t_0+t_1)\!:\!A)\end{array} \qquad v\!:\!t_j\!:\!A}{(c\cdot v)\!:\!(t_0+t_1)\!:\!A}$$

$$\frac{\begin{array}{c}\mathcal{D}\\ p\!:\!\mathbf{A}\end{array}}{\mathbf{C}(p)\!:\!c\!:\!\mathbf{A}}\;(cIt) \qquad \frac{\dfrac{d\!:\!(c\!:\!\mathbf{A}\to(\mathbf{A}\to c\!:\!\mathbf{A})) \qquad \dfrac{\begin{array}{c}\mathcal{D}_1\\ c\!:\!\mathbf{A}\end{array}}{!c\!:\!c\!:\!\mathbf{A}}}{(d\cdot !c)\!:\!(\mathbf{A}\to c\!:\!\mathbf{A})} \qquad t\!:\!\mathbf{A}}{(d\cdot\, !c\cdot t)\!:\!\mathbf{A}}$$

In each case $\widetilde{\mathcal{D}}$ denotes a corresponding standard $\mathcal{ILPN}$-derivation with the end rule (cI).

9.6 Theorem. (Realization of $\mathcal{ILPN}\lambda$ into $\mathcal{ILPN}$) *Under standard translations from 9.5 an $\mathcal{ILPN}\lambda$-derivation $\vec{x}\!:\!\Gamma \Rightarrow t(\vec{x})\!:\!A$ becomes a pure derivation $\vec{x}\!:\!\Gamma^r \Rightarrow t^r(\vec{x})\!:\!A^r$ in the $\mathcal{ILPN}$.*

Proof. A straightforward induction on an $\mathcal{ILPN}\lambda$-derivation $\vec{x}:\Gamma \Rightarrow t(\vec{x}):A$. It is immediate from definition 9.5 and lemma 9.2 that each standard transformation returns a pure derivation.
◀

9.7 Corollary. (Realization of λ-calculus for $\mathcal{I}nt$ into $\mathcal{ILPN}$) *Let $\mathcal{D}$ be a λ-term derivation of the type $\vec{x}:\Gamma \Rightarrow t(\vec{x}):A$ in the term calculus for $\mathcal{I}nt$. Standard translations define an effective step by step realization "r" of $\mathcal{D}$ as a derivation $\mathcal{D}'$ of $\vec{x}\!:\!\Gamma \Rightarrow t^r(\vec{x})\!:\!A$ in the $\mathcal{ILPN}$.*

9.8 Comment. As it is easy to see that $\mathcal{ILPN}\lambda$ (as well as λ-calculus for $\mathcal{I}nt$) can be realized in a small fragment of $\mathcal{ILPN}$ consisting of pure derivations only.

We already have enough ingredients to demonstrate that the Logic of Proofs can emulate modal λ-calculi.

Under $\mathbf{IS4}_\Box$ we mean the intuitionistic modal logic on the basis of $\mathcal{S}4$, introduced in [8] (cf. also [30], [42]). An inspection of the proof of theorem 6.2 (realization of modal logic), shows that this theorem holds also for $\mathcal{ILP}$ instead of $\mathcal{LP}$ and $\mathbf{IS4}_\Box$ instead of $\mathcal{S}4$. In other words, the intuitionistic logic of proofs is an explicit version of $\mathbf{IS4}_\Box$ in the same sense that $\mathcal{LP}$ is an explicit version of $\mathcal{S}4$. We will show how $\mathcal{ILPG}$ naturally emulates the modal λ-calculus for $\mathbf{IS4}_\Box$ ([8], [30], [42], cf. also [11]) and thus supplies modal λ-terms with standard provability semantics.

9.9 Theorem. (Realization of modal λ-calculus). *There is an effective step by step realization "r" of any derivation $\vec{x}:\Gamma \Rightarrow t(\vec{x}):A$ in the λ-term calculus for $\mathbf{IS4}_\Box$ as a derivation of $\vec{x}:\Gamma^r \Rightarrow t^r(\vec{x}):A^r$ in $\mathcal{ILPN}$.*

Proof. As above all the usual steps of λ-terms formation can be emulated in the Logic of Proofs (here in $\mathcal{ILPN}$). The entire term assignment system for $\mathbf{IS4}_\Box$ is obtained from the usual one for intuitionistic logic by adding two new rules that correspond to "modal" operations on λ-terms "`box`" and "`unbox`":

$$\frac{\Delta \Rightarrow s_1:\Box A_1 \ \ldots\ \Delta \Rightarrow s_k:\Box A_k \quad x_1:\Box A_1,\ldots,x_k:\Box A_k \Rightarrow t(\vec{x}):B}{\Delta \Rightarrow \mathtt{box}(t(\vec{s})):\Box B}\ (\Box I)$$

and

$$\frac{\Gamma \Rightarrow t:\Box A}{\Gamma \Rightarrow \mathtt{unbox}(t):A}\ (\Box E)$$

Let $\vec{p}:\Gamma \Rightarrow t:A$ be a modal λ term, and let $\mathcal{D}$ be a natural derivation of A from the hypothesis Γ, which is represented by this λ-term. The construction of the desired realization r takes two rounds. First, we realize all the occurrences of $\Box$ in the derivation $\mathcal{D}$ of B from $A_1,\ldots,A_n$ by proof polynomials according to the algorithm from 6.2. As a result, we get a realization $*$ of modalities in $\mathcal{D}$ such that $\mathcal{ILPN} \vdash \Sigma^* \Rightarrow F^*$ holds for every intermediate derivation $\Sigma \Rightarrow F$ in $\mathcal{D}$. The second round produces the desired realization r and proceeds by an induction on the steps of the λ-term construction (i.e. on the construction of $\mathcal{D}$). Without loss of generality we assume, that the proof variables used in the first round for $*$ are all different from the ones we use in the second round.

At the nodes of $\mathcal{D}$ corresponding to intuitionistic connectives use standard translations from 9.5. At a $(\Box I)$ node perform a natural deduction step is performed:

$$\frac{\Delta \Rightarrow \Box A_1 \ \ldots\ \Delta \Rightarrow \Box A_k \qquad \Box A_1,\ldots,\Box A_k \Rightarrow B}{\Delta \Rightarrow \Box B}\ .$$

The corresponding step of the modal λ-term assigning process is

$$\frac{\vec{u}:\Delta \Rightarrow s_1(\vec{u}):\Box A_1 \ \ldots\ \vec{u}:\Delta \Rightarrow s_k(\vec{u}):\Box A_k \qquad x_1:\Box A_1,\ldots,x_k:\Box A_k \Rightarrow t(\vec{x}):B}{\vec{u}:\Delta \Rightarrow \mathtt{box}(t)(\vec{s}(\vec{u})):\Box B}$$

By the construction of $*$

$$\Delta^* \Rightarrow f_1\!:\!A_1^*,\ \ldots,\ \Delta^* \Rightarrow f_k\!:\!A_k^*, \quad y_1\!:\!A_1^*,\ldots,y_k\!:\!A_k^* \Rightarrow B^*(\vec{y})$$

for some proof polynomials $f_1,\ldots,f_k$. Also by the construction of $*$ from 6.2 we may assume, that the variables $y_1,\ldots,y_k$ do not occur in $A_1^*,\ldots,A_k^*$. By 8.2, find a proof polynomial $p(\vec{y})$ such that

$$y_1\!:\!A_1^*,\ \ldots,\ y_k\!:\!A_k^* \Rightarrow p(\vec{y})\!:\!B^*(\vec{y}).$$

By the substitution $[\vec{y}/\vec{f}]$ from the latter derivation we get

$$f_1\!:\!A_1^*,\ \ldots,\ f_k\!:\!A_k^* \Rightarrow p(\vec{f})\!:\!B^*(\vec{f}).$$

And by the induction hypothesis,

$$\vec{u}\!:\!\Delta^* \Rightarrow s_1^r(\vec{u})\!:\!f_1\!:\!A_1^*, \quad \ldots, \quad \vec{u}\!:\!\Delta^* \Rightarrow s_k^r(\vec{u})\!:\!f_k\!:\!A_k^*.$$

By 8.2, construct a proof polynomial $g(\vec{x})$ such that

$$x_1\!:\!f_1\!:\!A_1^*,\ldots,x_k\!:\!f_k\!:\!A_k^* \Rightarrow g(\vec{x})\!:\!p(\vec{f})\!:\!B^*(\vec{f}),$$

by substitution,

$$s_1^r\!:\!f_1\!:\!A_1^*,\ldots,s_k^r\!:\!f_k\!:\!A_k^* \Rightarrow g(\vec{s^r}(\vec{u}))\!:\!p(\vec{f})\!:\!B^*(\vec{f}),$$

and, by the transitivity of $\Rightarrow$,

$$\vec{u}\!:\!\Delta^* \Rightarrow g(\vec{s^r}(\vec{u}))\!:\!p(\vec{f})\!:\!B^*(\vec{f}).$$

Let $(\mathtt{box}(t))^r = g(\vec{s^r}(\vec{u}))$.

At a $(\Box E)$-node of $\mathcal{D}$ we have a figure

$$\frac{\vec{u}\!:\!\Gamma \Rightarrow t(\vec{u})\!:\!\Box A}{\vec{u}\!:\!\Gamma \Rightarrow \mathtt{unbox}(t)(\vec{u})\!:\!A},$$

which corresponds to a natural deduction step from $\Gamma \Rightarrow \Box A$ to $\Gamma \Rightarrow A$. By the realization $*$ we have $\Gamma^* \Rightarrow A^*$. Use the standard transformation $(:E)$ from 9.5. to construct $h(\vec{x})$ and a pure proof

$$\vec{u}\!:\!\Gamma^* \Rightarrow h(\vec{u})\!:\!A^*.$$

Put $(\mathtt{unbox}(t))^r = h$.

◄

10 Discussion

The Kolmogorov interpretation of $\mathcal{I}nt$ as a calculus of problems [23], along with related papers by Gödel [17],[18] intended to interpret $\mathcal{I}nt$ on the basis of **classical** proofs and to give an independent definition of intuitionistic logic within the classical mathematics. The realization of $\mathcal{I}nt$ into $\mathcal{LP}$ (Theorem 6.9), which specifies $\mathcal{I}nt$ via the notion of classical proof, may be considered as an implementation of that Kolmogorov-Gödel program.

There are several reasons why we chose the combinatory logic format for $\mathcal{LP}$ versus an essentially equivalent pure λ-style presentation of $\mathcal{LP}$ without proof constants but with extra operations on terms. The current combinatory style axiomatization of $\mathcal{LP}$ (definition 1.2) is more compact than a possible pure λ-style axiomatization. In addition, the former occupies a position in between its two major applications: the language of modal logic and the language of λ-calculi.

The Logic of Proofs has a solid provability semantics and a more expressive language than either modal logic or the λ-calculus. Modal logic and traditional λ-calculi cover only fractions of $\mathcal{LP}$ but instead enjoy nice symmetries, transparent models, normal forms, etc. In their own narrow areas modal and λ presentations of the same facts are usually shorter than the corresponding presentations via proof polynomials. Thus $\mathcal{S}4$ and λ-calculi may be regarded as higher level languages for corresponding fragments of $\mathcal{LP}$.

Proof polynomials reveal the dynamic character of modality. The realization of $\mathcal{S}4$ in $\mathcal{LP}$ provides a fresh look at modal logic and its applications in general. Such areas as modal λ-calculi, polymorphic second order λ-calculi, λ-calculi with types depending on terms, non-deterministic λ-calculi, etc., could gain from their semantics as proof polynomials delivered by $\mathcal{LP}$.

Logic of Proofs for the second order arithmetic naturally requires quantifiers over propositions. Such a logic will accomodate second order polymorphic λ-calculus (cf. [16], [36], [6]), and supply it with provability semantics.

11 Acknowledgements

This work has benefited from many interactions over the past several years with H. Barendregt, L. Beklemishev, J. van Benthem, G. Boolos, S. Buss, R. Constable, D. van Dalen, J. Diller, J.M. Dunn, E. Engeler, S. Feferman, D. Gabbay, J.-Y. Girard, W. Hodges, G. Jäger, D. de Jongh, D. Kozen, V. Krupski, S. MacLane, G. Mints, F. Montagna, A. Nerode, E. Nogina, R. Parikh, C. Parsons, W. Pholers, V. Pratt, A. Preller, J. Remmel, D. Roorda, A. Scedrov, R. Shore, T. Sidon, T. Strassen, A. Troelstra, V.A. Uspensky, A. Visser.

I am indebted to Lena Nogina, Volodya Krupski, Tanya Sidon and Fred Smith for a reading of this paper which led to valuable improvements.

The research described in this paper was supported in part by the Russian Foundation for Basic Research, grant 96-01-01395.

References

[1] S. Artemov. "Kolmogorov logic of problems and a provability interpretation of intuitionistic logic", *Theoretical Aspects of Reasoning about Knowledge - III Proceedings*, Morgan Kaufman Pbl., pp. 257-272, 1990

[2] S. Artemov and T. Strassen, "Functionality in the Basic Logic of Proofs", *Tech.Rep. IAM 92-004, Department for Computer Science*, University of Bern, Switzerland, 1993.

[3] S. Artemov, "Logic of Proofs", *Annals of Pure and Applied Logic*, v. 67 (1994), pp. 29-59.

[4] S. Artemov, "Operational Modal Logic," *Tech. Rep. MSI 95-29*, Cornell University, December 1995.

[5] S. Artemov, "Proof realizations of typed λ-calculi," *Tech. Rep. MSI 97-2*, Cornell University, May 1997.

[6] H. Barendregt, "Lambda calculi with types". In *Handbook of Logic in Computer Science, vol.2*, S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, eds., Oxford University Press, pp. 118-309, 1992

[7] J. van Benthem. "Reflections on epistemic logic", *Logique & Analyse*, 133-134, pp. 5-14, 1991

[8] G. Bierman and V. de Paiva, "Intuitionistic necessity revisited", *Proceedings of the Logic at Work Conference,* Amsterdam (December 1992), Second revision, June 1996 (http://theory.doc.ic.ac.uk/tfm/papers.html).

[9] G. Boolos, *The Unprovability of Consistency: An Essay in Modal Logic*, Cambridge University Press, 1979

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

[11] V. A. J. Borghuis, *Coming to Terms with Modal Logic: On the interpretation of modalities in typed λ-calculus*, Ph.D. Thesis, Technische Universiteit Eindhoven, 1994

[12] S. Buss, "The Modal Logic of Pure Provability", *Notre Dame Journal of Formal Logic*, v. 31, No. 2, 1990

[13] A. Chagrov and M. Zakharyaschev, *Modal Logic*, Oxford Science Publications, 1997.

[14] D. van Dalen, *Logic and Structure*, Springer-Verlag, 1994.

[15] D.M. Gabbay, *Labelled Deductive Systems*, Oxford University Press, 1994.

[16] J.-Y. Girard, Y. Lafont, P. Taylor, *Proofs and Types*, Cambridge University Press, 1989.

[17] K. Gödel, "Eine Interpretation des intuitionistischen Aussagenkalkuls", *Ergebnisse Math. Colloq.*, Bd. 4 (1933), S. 39-40.

[18] K. Gödel, "Vortrag bei Zilsel" (1938), in S. Feferman, ed. *Kurt Gödel Collected Works. Volume III*, Oxford University Press, 1995

[19] R. Goldblatt, "Arithmetical necessity, provability and intuitionistic logic", *Theoria*, 44, pp. 38-46, 1978.

[20] A.Heyting, *Matematische Grundlagenforschung. Intuitionismus. Beweistheorie*, Springer, Berlin, 1934

[21] G. Japaridze and D. de Jongh, "The Logic of Provability", in *S.R. Buss, ed., Handbook of Proof Theory*, Elsevier Science BV, pp. 475-546, 1998

[22] S. Kleene. "On the interpretation of intuitionistic number theory", *Journal of Symbolic Logic*, v. 10, pp. 109-124, 1945

[23] A. Kolmogoroff, "Zur Deutung der intuitionistischen Logik", *Math. Ztschr.*, Bd. 35 (1932), S.58-65.

[24] D. Kozen and J. Tiuryn, "Logic of Programs", in *Handbook of Theoretical Computer Science. Volume B, Formal Models and Semantics*, The MIT Press/Elsevier, pp. 789-840, 1990

[25] G. Kreisel, "Foundations of intuitionistic logic", in *E.Nagel, P.Suppes and A.Tarski, eds., Logic, Methodology and Philosophy of Science. Proceedings of the 1960 International Congress*, Stanford University Press, Stanford, pp. 198-210, 1962.

[26] S. Kripke, "Semantical considerations on modal logic", *Acta Philosophica Fennica*, 16, pp. 83-94, 1963.

[27] V.N. Krupski, "Operational Logic of Proofs with Functionality Condition on Proof Predicate", Lecture Notes in Computer Science, v. 1234, *Logical Foundations of Computer Science' 97, Yaroslavl'*, pp. 167-177, 1997

[28] A.V. Kuznetsov and A.Yu. Muravitsky, "The logic of provability", Abstracts of the *4-th All-Union Conference on Mathematical Logic*, p. 73, (Russian), 1976.

[29] E. Lemmon, "New Foundations for Lewis's modal systems", *Journal of Symbolic Logic*, 22, pp. 176-186, 1957.

[30] S. Martini and A. Masini,"A computational interpretation of modal proofs", in Wansing, ed., *Proof Theory of Modal Logics*, (Workshop proceedings), Kluwer, 1994.

[31] P. Martin-Löf. "Constructive mathematics and computer programming", in *Logic, Methodology and Philosophy of Science VI*, North-Holland, pp. 153-175, 1982.

[32] P. Martin-Löf. *Intuitionistic Type Theory*, Studies in Proof Theory, Bibliopolis, Naples, 1984.

[33] Yu. Medvedev, "Finite problems", *Soviet Mathematics Doklady*, v. 3. pp. 227-230, 1962.

[34] E. Mendelson, *Introduction to mathematical logic. Third edition.*, Wadsworth, 1987

[35] G. Mints. "Lewis' systems and system T (1965-1973)". In *Selected papers in proof theory*, Bibliopolis, Napoli, 1992.

[36] J. Mitchell. "Type systems for programming languages". In van Leeuwen, ed. *Handbook of Theoretical Computer Science*, Elsevier/MIT press, 1990.

[37] A. Mkrtychev, "Models for the Logic of Proofs ", Lecture Notes in Computer Science, v. 1234, *Logical Foundations of Computer Science' 97, Yaroslavl'*, pp. 266-275, 1997

[38] R. Montague. "Syntactical treatments of modality with corollaries on reflection principles and finite axiomatizability", *Acta Philosophica Fennica*, 16, pp. 153-168, 1963.

[39] J. Myhill, "Some Remarks on the Notion of Proof", *Journal of Philosophy*, 57, pp. 461-471, 1960

[40] J. Myhill, "Intensional Set Theory", In: S. Shapiro, ed., *Intensional Mathematics*, North-Holland, pp. 47-61, 1985.

[41] C. Parsons and W. Sieg. "Introductory note to *1938a". In: S. Feferman, ed. *Kurt Gödel Collected Works. Volume III*, Oxford University Press, pp. 62-85, 1995.

[42] F. Pfenning and H.C. Wong, "On a modal lambda-calculus for S4", *Electronic Notes in Computer Science* 1, 1995.

[43] S. Shapiro. "Intensional Mathematics and Constructive Mathematics". In: S. Shapiro, ed., *Intensional Mathematics*, North-Holland, pp. 1-10, 1985.

[44] S. Shapiro. "Epistemic and Intuitionistic Arithmetic". In: S. Shapiro, ed., "Intensional mathematics", North-Holland, pp. 11-46, 1985.

[45] T. Sidon, "Provability Logic with Operations on Proofs", Lecture Notes in Computer Science, v. 1234, *Logical Foundations of Computer Science' 97, Yaroslavl'*, pp. 342-353, 1997

[46] R. Smullyan, *Diagonalization and Self-Reference*, Oxford University Press, 1994

[47] G. Takeuti, *Proof Theory*, North-Holland, 1975

[48] A.S. Troelstra and D. van Dalen, *Constructivism in Mathematics. An Introduction*, v. 1, Amsterdam; North Holland, 1988.

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

[50] V.A. Uspensky, "Kolmogorov and mathematical logic", *Journal of Symbolic Logic*, 57, No.2, 1992.

[51] V.A. Uspensky and V.E. Plisko "Intuitionistic Logic", In S.M. Nikol'ski, ed. *A.N. Kolmogorov, Collected works. Matematics and Mechanics*, pp. 394-404, 1985 (in Russian).

[52] S. Weinstein, "The intended interpretation of intuitionistic logic", *Journal of Philosophical Logic*, 12, pp. 261-270 1983.