Unified Semantics for Modality and λ-terms via Proof Polynomials *

Sergei N. Artemov †

Abstract

It is shown that the modal logic $\mathcal{S}4$, simple λ-calculus and modal λ-calculus admit a realization in a very simple propositional logical system $\mathcal{LP}$, which has an exact provability semantics. In $\mathcal{LP}$ both modality and λ-terms become objects of the same nature, namely, proof polynomials. The provability interpretation of modal λ-terms presented here may be regarded as a system-independent generalization of the Curry-Howard isomorphism of proofs and λ-terms.

1 Introduction

The Logic of Proofs ($\mathcal{LP}$, see Section 2) is a system in the propositional language with an extra basic proposition $t\!:\!F$ for "t is a proof of F". $\mathcal{LP}$ is supplied with a formal provability semantics, completeness theorems and decidability algorithms ([3], [4], [5]).

In this paper it is shown that $\mathcal{LP}$ 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.

The idea of logic of proofs can be found in Gödel's lecture [14] (see also [20]) first published in 1995, where a constructive version of the modal provability logic $\mathcal{S}4$ was sketched. This sketch does not contain formal definitions and lacks some important details, without which a realization of $\mathcal{S}4$ cannot be completed. The first presentations of $\mathcal{LP}$ (independent of [14]) took place at the author's talks at the conferences in Münster and Amsterdam in 1994.

Gabbay's Labelled Deductive Systems ([12]) 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 [6]. Intuitionistic Type Theory by Martin-Löf [17], [18] also makes use of the format $t\!:\!F$ with its informal provability reading.

* *Logic, Language and Computation'97, CSLI Publications*, Stanford University, 1998.

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

2 Logic of Proofs and Proof Polynomials

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

the usual language of propositional boolean logic
proof variables $x_0, \ldots, x_n, \ldots$, proof constants $a_0, \ldots, a_n, \ldots$
functional 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

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

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: $!, \cdot, +, :, \neg, \wedge, \vee, \to$. 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 last section of the paper.

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

A0. Axioms 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*"

Rules of inference:

$$\text{R1.}\qquad \frac{\Gamma\vdash F\to G \qquad \Gamma\vdash F}{\Gamma\vdash G} \qquad \text{"modus ponens".}$$

R2. *if* $\mathbf{A}$ *is an axiom A0 – A4, and c a proof constant, then* $\vdash c{:}\mathbf{A}$ *"necessitation"*

The definition of the intuitionistic logic of proofs $\mathcal{ILP}$ can be obtained from the definition of $\mathcal{LP}$ by replacing *A0* by the list of axiom scheme *A0I* for the propositional intuitionistic logic.

A *Constant Specification* ($\mathcal{CS}$) in $\mathcal{LP}$ ($\mathcal{ILP}$) is a finite set of formulas $c_1{:}A_1,\ldots,c_n{:}A_n$ such that c_i is a constant, and F_i an axiom *A0 – A4* (*A0I, A1 – A4* respectively). Each derivation in $\mathcal{LP}$ ($\mathcal{ILP}$) naturally generates the $\mathcal{CS}$ consisting of all formulas introduced in this derivation by the *necessitation* rule.

2.3 Comment. The system $\mathcal{LP}$ is correct and complete with respect to the provability semantics in a classical formal system, e.g. Peano Arithmetic $\mathcal{PA}$ ([3],[5], cf. also Section 7 of this paper). $\mathcal{ILP}$ is correct with respect to the provability interpretation 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.

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. [24]). 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(I), A1 – A4*. One may 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 does 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.

The deduction theorem holds in $\mathcal{LP}$ and $\mathcal{ILP}$.

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

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

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

2.4 Proposition. (Lifting Lemma) *Given a derivation* $\mathcal{D}$ *in* $\mathcal{LP}$ *or* $\mathcal{ILP}$ *of the type*

$$\vec{s}{:}\Gamma, \Delta \vdash F,$$

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

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

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(I), A1 – 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 $G \rightarrow F$ and G. Then, by the induction hypothesis, there are proof polynomials $u(\vec{s},\vec{y})$ and $v(\vec{s},\vec{y})$ such that $u{:}(G \rightarrow F)$ and $v{:}G$ are both derivable in $\mathcal{LP}$ from $\vec{s}{:}\Gamma, \vec{y}{:}\Delta$. By *A1*, $\vec{s}{:}\Gamma, \vec{y}{:}\Delta \vdash (uv){:}F$, 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*: $c{:}A \rightarrow !c{:}c{:}A$, to get $\vec{s}{:}\Gamma, \vec{y}{:}\Delta \vdash !c{:}F$, and put $t :=!c$.
◀

It is easy to see from the proof that the lifting polynomial $t(\vec{s},\vec{y})$ is nothing but a blueprint of $\mathcal{D}$. Thus $\mathcal{LP}$ accommodates its own proofs as terms. The necessitation rule

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

is a special case of Lifting. Note, that here p is the blueprint of a proof of F implicitly mentioned in "$\vdash F$".

2.5 Comment. Operations "$\cdot$" and "!" are present for deterministic proof systems (systems where each proof proves only one theorem) as well as for non-deterministic ones (where a proof can prove several different theorems). 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 Hilbert or Gentzen style proof systems may be considered as either deterministic (by assuming that a proof derives only the end formula/sequent of a proof tree) or as non-deterministic (by assuming that a proof derives all intermediate formulas assigned to the nodes of the proof tree). The logic of strictly deterministic proof systems was studied in [1], [2], and in [15], where it meets a complete axiomatization (system $\mathcal{FLP}$).

3 Realization of modal logic in $\mathcal{LP}$

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. A straightforward induction on a derivation in $\mathcal{LP}$ demonstrates that if $\mathcal{LP} \vdash F$, then $\mathcal{S}4 \vdash F^o$. As it was shown in [3], [5] the converse also holds. Namely, $\mathcal{LP}$ suffices to realize any $\mathcal{S}4$ theorem.

Under $\mathcal{IS}4$ we mean the intuitionistic modal logic on the basis of $\mathcal{S}4$ (cf. [7], [16], [21], where $\mathcal{IS}4$ was studied under the name $\mathbf{IS4}_\Box$). Basically the same algorithm (below) provides a realization of $\mathcal{IS}4$ in $\mathcal{ILP}$.

3.1 Example. $\mathcal{IS}4 \vdash (\Box A \wedge \Box B) \to \Box(A \wedge B)$

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

1. $A, B \vdash A \wedge B$, by propositional logic
2. $x:A, y:B \vdash t(x,y):(A \wedge B)$, by Lifting
3. $\vdash x:A \wedge y:B \to t(x,y):(A \wedge B)$, from 2.

3.2 Example. $\mathcal{IS}4 \vdash (\Box A \vee \Box B) \to \Box(A \vee B)$.

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

1. $A \to A \vee B, \quad B \to A \vee B$
2. $a:(A \to A \vee B), \quad b:(B \to A \vee B)$, by *necessitation*,
3. $x:A \to (a \cdot x):(A \vee B)$, from 2 by *A2*
4. $y:B \to (b \cdot y):(A \vee B)$, from 2 by *A2*
5. $ax:(A \vee B) \to (ax+by):(A \vee B)$, $by:(A \vee B) \to (ax+by):(A \vee B)$, by *A4*
6. $(x:A \vee y:B) \to (ax+by):(A \vee B)$

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

3.3 Comment. 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

$$\text{`` } x \textit{ is a proof of } A\text{''} \to \text{`` } f(x) \textit{ is a proof of } B\text{''}.$$

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

3.4 Theorem. *If $\mathcal{IS}4 \vdash F$, then $\mathcal{ILP} \vdash F^r$ for some normal realization r.*

Proof. Consider a cut-free sequent formulation of $\mathcal{IS}4$, with sequents $\Gamma \Rightarrow F$, where Γ is a finite set 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.[24]):

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

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

If $\mathcal{IS}4 \vdash F$, 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 $\mathcal{ILP} \vdash \bigwedge \Gamma^r \to B^r$ for any sequent $\Gamma \Rightarrow B$ in $\mathcal{T}$. We will also speak about a sequent $\Gamma \Rightarrow B$ being derivable in $\mathcal{ILP}$ meaning $\mathcal{ILP} \vdash \bigwedge \Gamma \to B$. 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 $\mathcal{ILP}$-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 $\mathcal{ILP}$. The axioms $S \Rightarrow S$ and $\bot \Rightarrow$ are derivable in $\mathcal{ILP}$. 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 $\mathcal{ILP}$ given that the premises are. The induction steps corresponding to these moves are straightforward.

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 $\mathcal{ILP}$, 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 $\mathcal{ILP}$. Since

$$\mathcal{ILP} \vdash t{:}Y \to (u_1 + \ldots + u_{i-1} + t + u_{i+1} + \ldots + u_{n_f}){:}Y,$$

we have

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

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 $\mathcal{ILP}$ remain derivable in $\mathcal{ILP}$, 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 $\mathcal{ILP}$. Note that the realization of $\Box$'s built by this procedure is normal. Moreover, the formula F^r may be regarded as the result of a skolemization procedure with respect to quantifiers on proofs (Comment 3.2) with the corresponding instantiation of Skolem functions by proof polynomials.

◄

3.5 Comment. It follows from 3.3 that $\mathcal{IS}4$ is nothing but a lazy version of $\mathcal{ILP}$ when we don't keep track on the proof polynomials assigned to the occurrences of $\Box$. Each theorem of $\mathcal{IS}4$ admits a decoding via $\mathcal{ILP}$ as a statement about specific proofs. The language of $\mathcal{ILP}$ is more rich than the one of $\mathcal{IS}4$. In particular, $\mathcal{IS}4$ theorems admit essentially different realizations in $\mathcal{ILP}$. For example, consider two theorems of $\mathcal{ILP}$ 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.

4 Gentzen formulation of $\mathcal{ILP}$

The Gentzen style system $\mathcal{ILPG}$ for $\mathcal{ILP}$ can be defined as follows (cf. the system **G2i** from [24]). 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

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

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

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

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

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

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

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

where in (Rc)-rule $\mathbf{A}$ is an axiom *A0I* – *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 *A0I* (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

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}'\\ F,\Gamma \Rightarrow F\end{array}}{t{:}F,\Gamma \Rightarrow F}}{\Gamma \Rightarrow t{:}F \to F}\,,$$

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

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

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

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

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

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

4.2 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$\rightarrow$).

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 $(L \rightarrow)$

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

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 $(R\cdot)$

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

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 $\mathcal{D}''$.

4.3 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 $\bot, \Delta \Rightarrow A \rightarrow B$, then put $\mathcal{D}'$ to be $\bot, A, \Delta \Rightarrow B$. 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

$$\dfrac{\begin{array}{c}\mathcal{D}_1 \\ A, \Gamma \Rightarrow B\end{array}}{\Gamma \Rightarrow A \rightarrow B}\;.$$

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

◀

4.4 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

$$\dfrac{\begin{matrix}\mathcal{D}_1\\ F, \Gamma \Rightarrow F\end{matrix}}{t{:}F, \Gamma \Rightarrow F}\ (\text{L:}),$$

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

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

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

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

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

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

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

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 3.3 and combine the new derivation $\mathcal{D}_3$

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

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 4.1. 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 [24] when convenient.

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

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

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 $\perp,\Gamma \Rightarrow F$;
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 ([24]).

Case 2. We permute the cut upward in a standard way (cf.[24]) 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.[24]). There is one additional concern here compared to [24]: 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

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

This is transformed into

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

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

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

This is transformed into a derivation with a lower ranked cut

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

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 (4.4) 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 4.3 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 4.3 and 4.4 all new cuts have lower rank.

This ends the proof of lemma 4.5.

◀

4.6 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 4.5 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 4.1 use Lemma 4.6 to eliminate every cut in a given derivation beginning with the uppermost ones.

◀

5 Natural deduction system and λ-terms for $\mathcal{ILP}$

5.1 Definition. The *natural deduction system $\mathcal{LPN}$ for $\mathcal{LP}$* is obtained from a usual natural deduction system for propositional logic (cf. [11], [24]) in the language of $\mathcal{LP}$ extended by the following rules

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

$$\frac{t\!:\!A}{(t+s)\!:\!A}\,(+I) \qquad\qquad \frac{t\!:\!A}{(s+t)\!:\!A}\,(+I) \qquad\qquad \frac{\begin{matrix}\mathcal{D}\\ \mathbf{A}\end{matrix}}{c\!:\!\mathbf{A}}\,\textit{(cI)},$$

where $\mathbf{A}$ is an axiom of the Hilbert version of $\mathcal{LP}$ (Definition 2.2), 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 $\mathbf{A}$ is an axiom *A0*, then $\mathcal{D}$ is the straightforward normal derivation of $\mathbf{A}$ in the natural deduction system for $\mathcal{I}nt$. For other axioms *A1* – *A4* the standard derivations are

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

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

Note that a standard derivation has no undischarged premises.

5.2 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

$$\dfrac{\begin{matrix}[\neg A]\\ \mathcal{D}\\ \bot\end{matrix}}{A}\;.$$

Under $\mathcal{LPN} \vdash \Gamma \Rightarrow A$ or $\mathcal{ILPN} \vdash \Gamma \Rightarrow A$ we mean "*A is derivable from assumptions* Γ" in $\mathcal{LPN}$ or in $\mathcal{ILPN}$ respectively.

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

1. $\Gamma \vdash_{\mathcal{ILP}} A$
2. $\mathcal{ILPG} \vdash \Gamma \Rightarrow A$
3. $\mathcal{ILPN} \vdash \Gamma \Rightarrow A$.

This fact is established by the standard mutual simulations of derivations in all three systems (cf. section 3.3 in [24]). 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 $\mathcal{ILPN}$.

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

Proof. A straightforward induction of the depth of a derivation. The number of steps in the algorithm constructing $\mathcal{D}'$ is bounded by a polynomial of the length of $\mathcal{D}$.
◄

5.4 Definition. *Contractions* for $\mathcal{ILPN}$ include all usual contractions for propositional logic ($\wedge-, \vee-, \to$*-contractions, permutation contractions*) (cf. section 6.1.3 in [24]), 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 derivation $\mathcal{D}$ is normal if no contraction is possible anywhere in $\mathcal{D}$.

5.5 Theorem. (From Gentzen to normal deductions in $\mathcal{ILP}$)

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

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 [24].
◀

5.6 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 5.5, 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 4.1 one could establish a strong normalization property of $\mathcal{ILPG}$ and $\mathcal{ILPN}$.
◀

Extending the term calculus for the intuitionistic logic ([24]) 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).

5.7 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 $\mathcal{I}nt$ ([24], 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) $\qquad$ $$\frac{t\!:\!\perp}{E^{\perp}_{A}(t)\!:\!A}\ (\perp Et)$$

$$\frac{s\!:\!A \qquad t\!:\!B}{\mathbf{p}(s,t)\!:\!(A\wedge B)}\ (\wedge It) \qquad\qquad \frac{t\!:\!(A_0\wedge A_1)}{\mathbf{p_j}(t)\!:\!A_j}\ j\in\{0,1\},\ (\wedge Et)$$

$$\frac{t : A_j}{\mathbf{k}_j(t) : (A_0 \vee A_1)}\ j \in \{0,1\},\ (\vee It) \qquad \frac{t : (A \vee B) \quad \begin{matrix}[u : A] \\ \vdots \\ s : C\end{matrix} \quad \begin{matrix}[v : B] \\ \vdots \\ s' : C\end{matrix}}{E^{\vee}_{u,v}(t,s,s') : C}\ (\vee Et)$$

$$\frac{\begin{matrix}[u : A] \\ \vdots \\ t : B\end{matrix}}{\lambda u.t : (A \to B)}\ (\to It) \qquad \frac{s : (A \to B) \qquad t : A}{App(s,t) : B}\ (\to Et)$$

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

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

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

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

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

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

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

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

1. $\mathbf{p}_j(\mathbf{p}(t_0, t_1)) \quad cont \quad t_j \quad (j \in \{0,1\})$,

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

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

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

Proof. Translate the proof of theorem 5.6 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 5.8.
◀

6 Abstraction in Logic of Proofs

In this section we show that $\mathcal{ILP}$ provides a standard provability semantics for the operator of λ-abstraction. This matches our earlier observation (Section 3), that $\mathcal{IS}4$-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 $\mathcal{ILP}$ both modality and modal λ-terms receive a uniform provability semantics.

The defined abstraction operator $\lambda^* x$ on proof polynomials below is a natural extension of the defined λ-abstraction operator $\lambda^* x$ in combinatory logic (cf. [24]).

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

6.2 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

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

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 $(a\cdot b\cdot c)\!:\!(A\to A)$ where a, b, c are proof constants specified by the conditions (cf. [24], section 1.3.6.)

$a\!:\!([A\to((A\to A)\to A)]\to[(A\to(A\to A))\to(A\to A)])$
$b\!:\![A\to((A\to A)\to A)]$
$c\!:\![A\to(A\to A)]$.

Let $\lambda^* x.x = (a\cdot b\cdot c)$. In fact this case coincides with the presentation of $\lambda^* x^A.x$ as $\mathbf{s}^{A,A\to A,A}\mathbf{k}^{A,A\to A}\mathbf{k}^{A,A}$ in combinatory logic (cf. [24]).

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

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

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 ground $(!I)$ 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

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

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

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

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

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

◀

6.3 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 6.2 of λ-abstraction by proof polynomials is similar the realization of $\mathcal{IS}4$-modality which is decomposed in 3.3 into a set of proof polynomials depending on a context (an $\mathcal{IS}4$-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 (below) for λ^*-abstraction, though it rules out $(!I)$. Note, that the rule $(!I)$ does not admit abstraction anyway. Indeed, in $\mathcal{ILPN}$ 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 $\mathcal{ILP}$.

6.4 Comment. 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 $\mathcal{ILPN} \vdash \vec{y}\!:\!\Gamma \Rightarrow p(\vec{y})\!:\!B$. 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.

6.5 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 $\mathcal{ILPN}$*
$y\!:\!F$	$y\!:\!F$
$\dfrac{s\!:\!A \quad t\!:\!B}{\mathbf{p}(s,t)\!:\!(A\wedge B)}$	$\dfrac{\dfrac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c\!:\!(A\to(B\to A\wedge B))\end{array} \qquad s\!:\!A}{(c\cdot s)\!:\!(B\to A\wedge B) \qquad t\!:\!B}}{(c\cdot s\cdot t)\!:\!(A\wedge B)}$
$\dfrac{t\!:\!(A_0\wedge A_1)}{\mathbf{p_j}(t)\!:\!A_j}\ j\in\{0,1\}$	$\dfrac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c\!:\!(A_0\wedge A_1\to A_j)\end{array} \qquad t\!:\!(A_0\wedge A_1)}{(c\cdot t)\!:\!A_j}$
$\dfrac{t\!:\!A_j}{\mathbf{k}_j(t)\!:\!(A_0\vee A_1)}\ j\in\{0,1\}$	$\dfrac{\begin{array}{c}\widetilde{\mathcal{D}}\\ c\!:\!(A_j\to A_0\vee A_1)\end{array} \qquad t\!:\!A_j}{(c\cdot t)\!:\!(A_0\vee A_1)}$

$$\dfrac{t:(A\vee B) \qquad \begin{array}{c}[u:A]\\ \vdots \\ s:C\end{array} \qquad \begin{array}{c}[v:B]\\ \vdots \\ s':C\end{array}}{E^{\vee}_{u,v}(t,s,s'):C} \qquad\qquad \dfrac{\dfrac{\dfrac{\mathcal{D}_1 \qquad \mathcal{D}_2}{(c\cdot\lambda^*u.s):((B\to C)\to(A\vee B\to C))} \quad \mathcal{D}_3}{(c\cdot\lambda^*u.s\cdot\lambda^*v.s'):(A\vee B\to C)} \qquad t:(A\vee B)}{(c\cdot\lambda^*u.s\cdot\lambda^*v.s'\cdot t):C}$$

where $\mathcal{D}_1$ is

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

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

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

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

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

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

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

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

$$\dfrac{v{:}t{:}A}{\mathbf{B}(v){:}!t{:}t{:}A}
\qquad
\dfrac{\overset{\widetilde{\mathcal{D}}}{c{:}(t{:}A\to !t{:}t{:}A)} \qquad v{:}t{:}A}{(c\cdot v){:}!t{:}t{:}A}$$

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

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

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

6.6 Theorem. (Realization of $\mathcal{ILPN}\lambda$ into $\mathcal{ILPN}$) *Under standard translations from 6.5 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 6.5 and lemma 6.2 that each standard transformation returns a pure derivation.

◀

6.7 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 $\mathcal{ILPN}$.*

6.8 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.

We will show how $\mathcal{ILPG}$ naturally emulates the modal λ-calculus for $\mathcal{IS}4$ ([7], [16], [21], cf. also [10]) and thus supplies modal λ-terms with standard provability semantics.

6.9 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 $\mathcal{IS}4$ 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 $\mathcal{IS}4$ 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 3.3. 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 6.5. At a $(\Box I)$ node 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 3.3 we may assume, that the variables $y_1, \ldots, y_k$ do not occur in $A_1^*, \ldots, A_k^*$. By 5.3, 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,

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

By 5.3, 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 6.5 to construct $h(\vec{x})$ and a pure proof

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

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

◀

7 Standard provability interpretation of $\mathcal{LP}$ and $\mathcal{ILP}$

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. In principle $\mathcal{LP}$ models any system of proofs with a proof checker operation capable of internalizing its own proofs as terms (cf. [22]). In particular, any proof system for the first order Peano Arithmetic $\mathcal{PA}$ (cf. [8], [9], [19], [23]) provides a model for $\mathcal{LP}$ with Gödel numbers of proofs being a instrument of internalizing proofs as terms. Although the soundness ($\Rightarrow$) does not necessarily refer to the arithmetical models of $\mathcal{LP}$, $\mathcal{PA}$ is convenient for establishing the completeness ($\Leftarrow$) of $\mathcal{LP}$. Given $\mathcal{LP} \nvdash F$ one can always find a proof system for $\mathcal{PA}$ along with an evaluation of variables in F which makes F false (cf. [5]).

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.

7.1 Definition. We assume that $\mathcal{PA}$ contains terms for all primitive recursive functions (cf. [23]), 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.

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

$$\mathcal{PA} \vdash \varphi \quad \Leftrightarrow \quad \text{for some } n \in \omega \quad \mathit{Prf}(n, \ulcorner\varphi\urcorner) \text{ holds}^1.$$

[1] We have omitted bars over numerals for natural numbers $n, \ulcorner\varphi\urcorner$ in the formula *Prf*.

A proof predicate *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. In particular, this property indicates that the set of theorems proven by k is finite for every k.

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

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

7.3 Comment. Every non-deterministic normal proof predicate can be made deterministic by changing from

"p proves $F_1, \ldots, F_n$" to "(p,i) proves F_i, $i = 1, \ldots n$".

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

"p proves $F_1 \wedge \ldots \wedge F_n$" as "$p$ proves each of F_i, $i = 1, \ldots n$".

7.4 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:*

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

$\mathit{Prf}(k, \ulcorner\varphi\urcorner) \to \mathit{Prf}(a(k,n), \ulcorner\varphi\urcorner)$, $\quad \mathit{Prf}(n, \ulcorner\varphi\urcorner) \to \mathit{Prf}(a(k,n), \ulcorner\varphi\urcorner)$

$\mathit{Prf}(k, \ulcorner\varphi\urcorner) \to \mathit{Prf}(c(k), \ulcorner\mathit{Prf}(k, \ulcorner\varphi\urcorner)\urcorner)$.

Proof. The following function can be taken as m:

> *Given k, n put $m(k,n) = \mu z$"$\mathit{Prf}(z, \ulcorner\psi\urcorner)$ 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.

7.5 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 7.4
- an evaluation of propositional letters by sentences of arithmetic, and
- an evaluation of proof letters 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. [19]). 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 $*$.

7.6 Proposition. (Arithmetical soundness of $\mathcal{LP}$)

1. *If $\mathcal{LP} \vdash F$ with a constant specification $\mathcal{CS}$, then F is $\mathcal{CS}$-valid.*
2. *If $\mathcal{LP} \vdash F$ with a constant specification $\mathcal{CS}$, then $\mathcal{PA} \vdash F^*$ for any $\mathcal{CS}$-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 $*$ $(t{:}F \to F)^* \equiv \mathit{Prf}(t^*, \ulcorner F^* \urcorner) \to F^*$. Consider two possibilities. Either $\mathit{Prf}(t^*, \ulcorner F^* \urcorner)$ 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 $\mathit{Prf}(t^*, \ulcorner F^* \urcorner)$ is false, in which case being a false Δ_1 formula it is refutable in $\mathcal{PA}$, i.e. $\mathcal{PA} \vdash \neg \mathit{Prf}(t^*, \ulcorner F^* \urcorner)$ and again $\mathcal{PA} \vdash (t{:}F \to F)^*$.
◀

In fact $\mathcal{LP}$ also enjoys the following completeness property.

7.7 Proposition. ([3], [5])

1. *$\mathcal{LP} \vdash F$ with a constant specification $\mathcal{CS}$ iff F is $\mathcal{CS}$-valid.*
2. *$\mathcal{LP} \vdash F$ with a constant specification $\mathcal{CS}$ iff $\mathcal{PA} \vdash F^*$ for any $\mathcal{CS}$-interpretation $*$.*

Definition similar to 7.5 provides an arithmetical model for $\mathcal{ILP}$ in Heyting Arithmetic $\mathcal{HA}$.

7.8 Theorem. (Arithmetical soundness of $\mathcal{ILP}$)

If $\mathcal{ILP} \vdash F$ with a constant specification $\mathcal{CS}$, then $\mathcal{HA} \vdash F^*$ for any $\mathcal{CS}$-interpretation $*$.

8 Discussion

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 $\mathcal{LP}$ and $\mathcal{ILP}$ (definition 2.2) 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 $\mathcal{ILP}$ 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, nondeterministic λ-calculi, etc., could gain from their semantics as proof polynomials delivered by $\mathcal{LP}$.

9 Acknowledgements

This work has benefited from many interactions over the past several years with a number of mathematicians, logicians and computer scientists: H. Barendregt, L. Beklemishev, J. van Benthem, R. Constable, J.M. Dunn, V. Krupski, G. Mints, A. Nerode, E. Nogina, V. Pratt, J. Remmel, A. Scedrov, R. Shore, T. Sidon.

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 and by ARO under the MURI program "Integrated Approach to Intelligent Systems", grant DAAH04-96-1-0341.

References

[1] 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.

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

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

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

[5] S. Artemov, "Logic of Proofs: a Unified Semantics for Modality and λ-terms," *Tech. Rep. CFIS 98-06*, Cornell University, March 1998.

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

[7] 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).

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

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

[10] 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

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

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

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

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

[15] 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

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

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

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

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

[20] 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.

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

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

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

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