2003

TR-2003007: On the Complexity of the Reflected Logic of Proofs

Nikolai V. Krupski

On the Complexity of the Reflected Logic of Proofs

Nikolai V. Krupski

Department of Math. Logic and the Theory of Algorithms,
Faculty of Mechanics and Mathematics,
Moscow State University, Moscow 119899, Russia.
e-mail: nvk@lpcs.math.msu.ru
Tel/Fax: +7(095)9393031

Keywords: proof theory, explicit modal logic, logic of proofs, proof polynomials, symbolic models, disjunctive property, complexity.

Abstract

Artemov's system $\mathcal{LP}$ captures all propositional invariant properties of a proof predicate "x proves y" ([1, 3]). Kuznets in [5] showed that the satisfiability problem for $\mathcal{LP}$ belongs to the class Π_2^p of the polynomial hierarchy. No nontrivial lower complexity bound for $\mathcal{LP}$ is known. We describe quite expressive syntactical fragment of $\mathcal{LP}$ which belongs to NP. It is $r\mathcal{LP}_{\wedge,\vee}$ – the set of all theorems of $\mathcal{LP}$ which are monotone boolean combinations of quasiatomic formulas (facts of sort "t proves F").

A new decision algorithm for this fragment is proposed. It is based on a new simple independent formalization for $r\mathcal{LP}$ (the reflected fragment of $\mathcal{LP}$) and involves the corresponding proof search procedure. Essentially $r\mathcal{LP}$ contains all the theorems of $\mathcal{LP}$ supplied with additional information about their proofs. We show that in many respects $r\mathcal{LP}$ is simpler than $\mathcal{LP}$ itself. This gives the complexity bound (NP) for $r\mathcal{LP}$. In addition we prove a suitable variant of the disjunctive property which extends this bound to $r\mathcal{LP}_{\wedge,\vee}$.

1 Introduction

The Propositional Logic of Proofs $\mathcal{LP}$ was introduced by S.N. Artemov in [1]. It describes the invariant (i.e. independent on the choice of the particular deductive systems) propositional properties of arithmetical proof predicate "x proves y" together with computable operations on proofs induced by admissible inference rules: "$\cdot$" – the application of the rule *modus ponens*,"!" – proof checking,"+" – the union of proofs (imitates the nondeterministic choice). Besides the standard propositional part the $\mathcal{LP}$ language contains the proof terms which are used as representations of particular proofs. They are build from proof variables and proof constants with the help of function symbols $\cdot$, !, +. The proof predicate is expressed by the supplementary constructor of quasiatomic formulas $(t : F)$ where t is a term and F is a formula.

The Logic of Proofs $\mathcal{LP}$ is sound and complete with respect to the arithmetical provability semantics. This semantics leads to the unified approach for constructing the provability interpretations for intuitionistic logic, modal logic and lambda terms (see [1],[2], [3]). A more simple, *symbolic* semantic was proposed in [4] where the decidability of $\mathcal{LP}$ was proved. The complexity analysis of this decision procedure gives the best known upper bound: $\mathcal{LP}$ belongs to $\prod_2^p$ (see [5]).

We are seeking for syntactical fragments of $\mathcal{LP}$ with lower complexity. A trivial example of such a fragment is the set of all $\mathcal{LP}$-theorems which do not contain proof terms at all. It coincides with the set of all tautologies and is *co-*NP-complete. In this paper it is shown that there exist quite expressive syntactical fragments of $\mathcal{LP}$ which belong to NP.

We consider the reflected fragment of $\mathcal{LP}$, i.e. the set $r\mathcal{LP}$ of all formulas of the form $t : F$ which are provable in $\mathcal{LP}$. Note that $\mathcal{LP}$ is imbedded into $r\mathcal{LP}$:

$$\mathcal{LP} \vdash F \Leftrightarrow \mathcal{LP} \vdash t : F \text{ for some proof term } t \tag{1}$$

([1], Lifting lemma). At the same time $r\mathcal{LP}$ is much more simple. We prove that $r\mathcal{LP}$ is a theory of a single symbolic model. The construction of this model can be described explicitly by a simple calculus C which gives an independent formalization for $r\mathcal{LP}$. All the inference rules of C are the introduction rules for operations $\cdot$, !, + and proof constants. Every derivation of a formula $t : F$ is isomorphic to some subtree of the tree representation of the term t. The proof search in the C calculus gives the decision algorithm for $r\mathcal{LP}$ and the complexity bound: $r\mathcal{LP}$ belongs to NP. The comparison

of the complexity bounds shows that the imbedding (1) may be not quite effective – the size of the corresponding proof term t may be exponential in the length of F.

The deductive and model descriptions of $r\mathcal{LP}$ provides the means to answer some general questions about the structure of $\mathcal{LP}$ derivations. We consider the disjunctive property. It turns out to be essential in the proof of the same upper bound (NP) for another syntactical fragment – $r\mathcal{LP}_{\wedge,\vee}$, the set of all $\mathcal{LP}$-theorems which are monotone boolean combinations of quasiatomic formulas.

For a calculus L the disjunctive property is a statement of the form:

$$L \vdash F \vee G \Leftrightarrow L \vdash F \text{ or } L \vdash G.$$

It simplifies the proof search procedures in the case of a formula $F \vee G$: we may consider the $\vee$-introduction rules only. It is valid for the intuitionistic propositional logic but not for the classical propositional logic. For $\mathcal{LP}$ the general form of disjunctive property is not valid too because $\mathcal{LP}$ extends the classical propositional logic. We prove the restricted form

$$\mathcal{LP} \vdash s : F \vee t : G \Leftrightarrow \mathcal{LP} \vdash s : F \text{ or } \mathcal{LP} \vdash t : G$$

which is sufficient to extend the complexity bound from $r\mathcal{LP}$ to $r\mathcal{LP}_{\wedge,\vee}$.

Similar questions concerning the possibility of the proof search simplification arise when we search for an $\mathcal{LP}$-proof of a formula which has one of the forms $(t \cdot s) : F$, $(t + s) : F$ or $(!t) : F$. The analysis of the C calculus gives a uniform answer for all of them (see the corollary 5.2).

2 Preliminaries

The language of $\mathcal{LP}$ contains the usual language of classical propositional logic, proof variables x_i, proof constants c_i, functional symbols: monadic !, binary $+$ and $\cdot$, operator symbol ":" of the type "*term*:*formula*". Proofs are presented by proof terms which are built from the proof variables and the proof constants using the operations on proofs $(!, +, \cdot)$. Formulas are constructed from propositional letters and boolean constants in the usual way with additional rule:

if F is a formula and t is a term, then $t : F$ is a formula.

$SVar$ denotes the set of all propositional letters, Tm – the set of all terms, Fm – the set of all formulas.

The logic of proofs $\mathcal{LP}$ is defined by the following calculus:

Axioms:

A0 Axioms of the classical propositional logic in the language of $\mathcal{LP}$

A1 $t : F \to F$

A2 $t : (F \to G) \to (s : F \to (t \cdot s) : G)$

A3 $t : F \to (t + s) : F, \quad s : F \to (t + s) : F$

A4 $t : F \to !t : (t : F)$

Rules :

(Modus ponens) $\dfrac{F \to G \quad F}{G}$;

(Necessitation) $\vdash c : A$ where c is a proof constant and A is one of axioms A0-A4.

Definition 2.1 A constant specification (CS) is a finite set of formulas of the form $c : A$ where c is a proof constant and A is one of the axioms **A0**–**A4**. Let $\mathcal{LP}_{CS}$ be the fragment of $\mathcal{LP}$ with axioms **A0**–**A4**, modus ponens rule and the restricted form of Necessitation rule:

(Necessitation$_{CS}$) $\vdash c : A$ for $c : A \in CS$.

3 Symbolic models for $\mathcal{LP}$

Definition 3.1 A function $* : Tm \to 2^{Fm}$ that assigns to every $\mathcal{LP}$–term a set of $\mathcal{LP}$–formulas is called a (proof-theorem) *assignment table* if it satisfies the following conditions:

1. If $F \to G \in *(t)$ and $F \in *(s)$ then $G \in *(t \cdot s)$.
2. $*(t) \cup *(s) \subseteq *(t + s)$.
3. If $F \in *(t)$ then $t : F \in *(!t)$.

Definition 3.2 We define a partial ordering on the set of all assignment tables: $* \leq *'$ if $*(t) \subseteq *'(t)$ holds for all proof terms t.

Definition 3.3 A *model* M is a triple $(v, *, \models)$, where v is a truth-assignment, i.e. a mapping $v : SVar \to \{True, False\}$, $*$ is an assignment table and $\models$ is a truth relation which is defined by the first two components in the following way:

1. For propositional letters $M \models S \Leftrightarrow v(S) = True;\ M \not\models \bot$
2. $M \models F \to G \Leftrightarrow \not\models F$ or $\models G$
3. $M \models t : F \Leftrightarrow F \in *(t)$

Definition 3.4 A *pre-model* P is a triple $(v, *, \models_p)$, where v is a truth-assignment, $*$ is an assignment table and the definition of a truth relation $\models_p$ is similar to $\models$ (see Definition 3.3) except for the case

$$P \models_p t : F \Leftrightarrow F \in *(t) \text{ and } P \models_p F.$$

Definition 3.5 For two models $M = (v, *, \models)$ and $M' = (v', *', \models')$ we shall write $M \leq M'$ if $* \leq *'$ and $v \leq v'$ (the latter means that for every $S \in SVar$ holds $v(S) = True \Rightarrow v'(S) = True$).

Definition 3.6 The model $M = (v, *, \models)$ is called *reflexive* if $F \in *(t)$ implies $M \models F$.

Definition 3.7 Let the constant specification CS be given. A model M (a pre-model P) is called a CS-model (a CS-pre-model) if $M \models CS$ ($P \models_p CS$). An assignment table $*$ is called a CS-table if $F \in *(c)$ holds for all formulas $c : F \in CS$.

Theorem 3.8 *([4], [5])* $\mathcal{LP}_{CS} \vdash F \Leftrightarrow F$ *is valid in all reflexive* CS*-models.*

Theorem 3.9 *([4], [5])* $\mathcal{LP}_{CS} \vdash F \Leftrightarrow F$ *is valid in all* CS*-pre-models.*

For formulas of the form $t : F$ we will prove a stronger variant of the theorem 3.9 (see the theorem 4.1).

Lemma 3.10 *For every constant specification* CS *there exists a* CS*-table* $*$ *for which*

$$F \in *(t) \Rightarrow \mathcal{LP}_{CS} \vdash F$$

holds for every term t *and every formula* F*.*

Proof. Let some constant specification be fixed: $CS = c_1 : A_1, \ldots, c_n : A_n$. We define $*$ as the least CS-table satisfying the condition:

$$*(c_i) = \{F_j | c_i = c_j, 1 \leq j \leq n\}, \quad i = 1, \ldots, i = n.$$

Let us describe the construction of $*$. We fix a sequence $t_1, t_2, \ldots$ containing all the proof terms of the language where every term is met infinitely many times. The procedure consists of ω steps.

Step 0.

$$*(t) := \begin{cases} \{F_j | c_i = c_j, 1 \leq j \leq n\}, & \text{where } t = c_i,\ i = 1, \ldots, n; \\ \emptyset, & \text{otherwise.} \end{cases}$$

Step $k > 0$. Update the value of $*(t_k)$:

Case $t_k = hs$.

$$*(t_k) := *(t_k) \cup \{G \mid F \to G \in *(h) \text{ and } F \in *(s) \text{ for some formula } F\}.$$

Case $t_k = h + s$.

$$*(t_k) := *(t_k) \cup *(h) \cup *(s).$$

Case $t_k = !s$.

$$*(t_k) := *(t_k) \cup \{s : F \mid F \in *(s)\}.$$

The result of this procedure is the required assignment table $*$.

Let $F \in *(t)$. Then F was added to $*(t)$ at some step N and $t = t_N$. By the induction on N we prove that $\mathcal{LP}_{CS} \vdash t : F$.

Let $N = 0$. Then $t_0 = c_i$ for some i and $t_0 : F \in CS$. So $\mathcal{LP}_{CS} \vdash t_0 : F_0$.

Let $N > 0$ and $\mathcal{LP}_{CS} \vdash t_i : G$ for all G that were added to $*(t_i)$ at steps $i < N$. Let us prove that $\mathcal{LP}_{CS} \vdash t_N : F$.

If $t_N = c_i$ for some $i = 1 \ldots n$ then $t_N : F_N \in CS$, so $\mathcal{LP}_{CS} \vdash t_N : F_N$.

Let $t_N = hs$. Then $F \in *(hs)$. So, there exists a formula G and integers $i, j < N$ such that $h = t_i$, $s = t_j$, the formula $G \to F$ was added to $*(t_i)$ and the formula G was added to $*(t_j)$ at the steps i and j respectively. So, $\mathcal{LP}_{CS} \vdash t_i : (G \to F)$ and $\mathcal{LP}_{CS} \vdash t_j : G$. By **A2** we have $\mathcal{LP}_{CS} \vdash t_i : (G \to F) \to ((t_j : G) \to (t_i t_j) : F)$. Hence, $\mathcal{LP}_{CS} \vdash t_i t_j : F$ which is $\mathcal{LP}_{CS} \vdash t_N : F$.

Let $t_N = h + s$. Then $F \in *(h + s)$. Then $h = t_i$ or $s = t_i$ for some $i < N$ and F was added to $*(t_i)$ at the step i. Let, for example, $h = t_i$. It means $\mathcal{LP}_{CS} \vdash t_i : F$. By **A4** we have: $\mathcal{LP}_{CS} \vdash t_i : F \to (t_i + s) : F$. Then $\mathcal{LP}_{CS} \vdash (t_i + s) : F$ or, the same, $\mathcal{LP}_{CS} \vdash t_N : F$.

Let $t_N = !h$. Then for some $i < N$ holds $h = t_i$, $F = t_i : G$ and G was added to $*(h)$ at the step i. Then $\mathcal{LP}_{CS} \vdash t_i : G$. By **A3** we have $\mathcal{LP}_{CS} \vdash t : F \rightarrow !t : (t : F)$. Hence $\mathcal{LP}_{CS} \vdash !t_i : (t_i : F)$ or, $\mathcal{LP}_{CS} \vdash t_N : F$. ■

Comment 3.11 The backward implication

$$\mathcal{LP}_{CS} \vdash t : F \Rightarrow F \in *(t)$$

is valid for each CS-table. Indeed, let $P = < v, *, \models_p >$ be a pre-model with CS- table $*$. Then, by the theorem 3.9,

$$\mathcal{LP}_{CS} \vdash t : F \Rightarrow P \models_p t : F \Rightarrow F \in *(t).$$

4 Minimal models and the disjunctive property

Theorem 4.1 *For every constant specification CS there exists a pre-model P such that*

$$P \models_p t : F \;\Leftrightarrow\; \mathcal{LP}_{CS} \vdash t : F$$

.

Proof. Let a constant specification $CS = c_1 : A_1, \ldots, c_n : A_n$ be fixed. Let a pre-model P be $(v, *, \models_p)$ where $*$ is a CS-table from the lemma 3.10 and $v(S) := False$ for each $S \in SVar$.

By the theorem 3.9,

$$\mathcal{LP}_{CS} \vdash t : F \;\Rightarrow\; P \models_p t : F.$$

The backward implication is valid too. Indeed, if $P \models_p t : A$ then $A \in *(t)$ and, by the lemma 3.10, $\mathcal{LP}_{CS} \vdash t : A$. ■

Definition 4.2 A model $M = (v, *, \models)$ and a pre-model $P = (v', *', \models_p)$ are equivalent if the relations $\models$ and $\models_p$ coincide, i.e. $M \models F \Leftrightarrow P \models_p F$.

Theorem 4.3 *([4], [5]) For every pre-model there exists an equivalent reflexive model.*

By the theorem 4.3 the result of the theorem 4.1 can be transferred to the case of reflexive models. The corresponding model turns out to be the least element in the class of all reflexive CS-models.

Theorem 4.4 *For every constant specification CS in the class of all reflexive CS-models there exists the minimal element M. For that model holds*

$$M \models t : F \quad \Leftrightarrow \quad \mathcal{LP}_{CS} \vdash t : F. \tag{2}$$

Proof. Let $CS = c_1 : A_1, \ldots, c_N : A_N$ be a constant specification, $P = (v_0, *_0, \models_0)$ be the corresponding pre-model built by the theorem 4.1. By the theorem 4.3 there exists a reflexive model $M = (v, *, \models)$, which is equivalent to P. Then

$$\mathcal{LP}_{CS} \vdash t : F \Leftrightarrow P \models_0 t : F \Leftrightarrow M \models t : F$$

In particular, $M \models CS$. We claim that M is the least reflexive CS-model. Let $M' = (v', *', \models')$ be an arbitrary reflexive CS-model.

Then $* \leq *'$, i.e. $F \in *(t) \Rightarrow F \in *'(t)$ for every formula F. Let $F \in *(t)$. Indeed, by the definition of the model $M \models t : F$ and as M and P are equivalent, $P \models_0 t : F$. Hence, by the theorem 4.1, $\mathcal{LP}_{CS} \vdash t : F$. By the theorem 3.8 we have $M' \models' t : F$, which means $F \in *'(t)$.

By the construction of P from theorem 4.1 $P \not\models_p S$ for every $S \in SVar$. Then $M \not\models S$. So, $v(S) = False$ for all $S \in SVar$ which means $v \leq v'$. Thus, $M \leq M'$. ∎

Corollary 4.5 *(Restricted disjunctive property for $\mathcal{LP}_{CS}$ and $\mathcal{LP}$)*

1. For every constant specification CS

$$\mathcal{LP}_{CS} \vdash t_1 : F_1 \vee \ldots \vee t_n : F_n \quad \Leftrightarrow \quad \mathcal{LP}_{CS} \vdash t_1 : F_1 \ \ or \ \ldots \ or \ \mathcal{LP}_{CS} \vdash t_n : F_n.$$

2.

$$\mathcal{LP} \vdash t_1 : F_1 \vee \ldots \vee t_n : F_n \quad \Leftrightarrow \quad \mathcal{LP} \vdash t_1 : F_1 \ \ or \ \ldots \ or \ \mathcal{LP} \vdash t_n : F_n.$$

Proof.

1. The implication from right to left is trivial. Let us prove the remaining one. Suppose that $\mathcal{LP}_{CS} \vdash t_1 : F_1 \vee \ldots \vee t_n : F_n$, but $\mathcal{LP}_{CS} \nvdash t_1 : F_1$, …, $\mathcal{LP}_{CS} \nvdash t_n : F_n$. By the theorem 3.8 there exist the reflexive CS-models $K_1, \ldots, K_n$ for which $K_i \not\models t_i : F_i$. Then for the least reflexive CS-model M we have $M \not\models t_i : F_i$, $i = 1, \ldots, n$. Hence, by the theorem 3.8 $\mathcal{LP} \nvdash t_1 : F_1 \vee \ldots \vee t_n : F_n$. Contradiction.

2. Follows from 1. ∎

5 The reflected fragment of $\mathcal{LP}$

The reflected fragment of $\mathcal{LP}$ ($\mathcal{LP}_{CS}$) is the set of all formulas of the form $t : F$ which are provable in $\mathcal{LP}$ (in $\mathcal{LP}_{CS}$ respectively). $r\mathcal{LP}$ denotes the reflected fragment of $\mathcal{LP}$. The least model construction from the theorem 4.4 can be reformulated explicitly as a calculi formalizing the reflected fragments of $\mathcal{LP}_{CS}$ and $\mathcal{LP}$.

Let some constant specification be given

$$CS = c_1 : A_1, \ldots, c_N : A_N.$$

We define the calculus C_{CS}:

Axioms:

$$c_1 : A_1, \ldots, c_N : A_N$$

Rules:

C1 $$\frac{t : F \to G \quad s : F}{ts : G}$$

C2 $$\frac{t : F}{(t + s) : F} \qquad \frac{s : F}{(t + s) : F}$$

C3 $$\frac{t : F}{!t : t : F}$$

Let C be the calculus with the rules **C1**–**C3** and the Necessitation rule (in the same form as for $\mathcal{LP}$).

Theorem 5.1

$$\mathcal{LP}_{CS} \vdash t : F \Leftrightarrow C_{CS} \vdash t : F$$
$$t : F \in r\mathcal{LP} \Leftrightarrow C \vdash t : F$$

Proof. The second statement of the theorem is the consequence of the first one because $r\mathcal{LP}$ is the union of reflected fragments of all $\mathcal{LP}_{CS}$.

Let us prove the first one. It is easy to see that $C_{CS} \vdash t : F$ iff $F \in *(t)$, where $*$ is the assignment table from the lemma 3.10. So,

$$C_{CS} \vdash t : F \Leftrightarrow F \in *(t) \Leftrightarrow \mathcal{LP}_{CS} \vdash t : F$$

by the lemma 3.10. ■

Corollary 5.2

$$\begin{array}{lcl} \mathcal{LP} \vdash ts : F & \Leftrightarrow & \mathcal{LP} \vdash t : G \to F \textit{ and } \mathcal{LP} \vdash s : G \textit{ for some } G, \\ \mathcal{LP} \vdash (t+s) : F & \Leftrightarrow & \mathcal{LP} \vdash t : F \textit{ or } \mathcal{LP} \vdash s : F, \\ \mathcal{LP} \vdash !t : F & \Leftrightarrow & F = t : G \textit{ and } \mathcal{LP} \vdash t : G \textit{ for some } G. \end{array}$$

The best known complexity bound for $\mathcal{LP}$ was proved in [5]: $\mathcal{LP}$ belongs to the class $\prod_2^p$ of the polynomial hierarchy. The decision algorithm from [5] (which is the same as in [4]) can be applied in the case of $r\mathcal{LP}$ too and gives the same upper bound for the restricted case. We improve this upper bound for $r\mathcal{LP}$ using a different algorithm (the proof search in C calculus):

Theorem 5.3 *$r\mathcal{LP}$ belongs to NP.*

Proof. It is sufficient to prove that the deducibility problem for C belongs to NP. Consider the derivation tree of a formula $t : F$ in C. Note that the number of nodes in it is bound by the number of subterms in the term t. Every application of a rule in the derivation tree corresponds to an occurrence of some subterm s in the initial term t and has the form

$$\frac{\cdots}{s : G}$$

where different nodes correspond to the different occurrences of subterms in the term t. Let $t_n : F_n$ denote the formula in the node $n \in Node$ where $Node$ is the set of all nodes of the derivation tree. With every node $n \in Node$ we associate an equation on the syntactical variables F_k, $k \in Node$ which express the relation between the premises and the conclusion of the inference rule involved:

$$F_{n_1} = F_{n_2} \to F_n \quad \text{or} \quad F_n = F_{n_1} \quad \text{or} \quad F_n = t_{n_1} : F_{n_1}$$

for rules **C1**, **C2** and **C3** respectively (where $n_i \in Node$ mean the direct predecessors of the node n). For the *Necessitation* rule the equitation has the form

$$F_n = A$$

where A is the scheme of the corresponding axiom **A0**–**A4**; the metavariables of the scheme we also include into the set of syntactical variables. Let S be the set of all these equations extended by the equation

$$F_{n_0} = t : F$$

where n_0 is the root of the tree. It is easy to see that S can be recovered uniquely from the formula $t : F$ and from the tree labeled only by the terms t_k and by the schemes of the corresponding inference rules (in the case of *Necessitation* rule we add the scheme of the corresponding axiom **A0**–**A4** too).

A formula $t : F$ is derivable in the calculus C iff there exists such a labeled tree, for which the system of equations S is unifiable. The size of the labeled tree, the length of the system S and the time required for the unifiability test (see [6]) can be bound by some polynomials on the length of the formula $t : F$. This way, the deducibility problem for the calculus C belongs to NP. ■

6 $\{\wedge, \vee\}$-combinations of quasiatomic formulas.

Definition 6.1 A formula F is a $\{\wedge, \vee\}$-combination of quasiatomic formulas if it is constructed from quasiatomic formulas (i.e. formulas of the form $t : G$) using the connectives $\wedge, \vee$ only. Let $r\mathcal{LP}_{\wedge,\vee}$ be the set

$$\{F \mid \mathcal{LP} \vdash F \text{ and } F \text{ is a } \{\wedge, \vee\}\text{-combination of quasiatomic formulas}\}$$

The upper bound from the theorem 5.3 can be extended to the $r\mathcal{LP}_{\wedge,\vee}$ fragment by the following lemma.

Lemma 6.2 *Let F be a $\{\wedge, \vee\}$-combination of quasiatomic formulas, $F = A(t_1 : F_1, \ldots, t_n : F_n)$, where $A(x_1, \ldots, x_n)$ is a $\{\wedge, \vee\}$-combination of propositional variables. Then $\mathcal{LP} \vdash F$ iff there exists a subset I of the set $\{1, \ldots, n\}$ such that*

1. *$\mathcal{LP} \vdash t_i : F_i$ for every $i \in I$*
2. *$A(x_1, ..., x_n) = 1$ for the evaluation $x_i = \begin{cases} 1, & i \in I, \\ 0, & otherwise. \end{cases}$*

Proof. The part "only if" is a trivial consequence of monotonicity of A.

Let $\mathcal{LP} \vdash F$. We define $I = \{i \mid \mathcal{LP} \vdash t_i : F_i\}$. Let B be the CNF of A. Then $B = \bigwedge_{j \in J} B_j$ where B_j is a clause consisted of positive literals only. Let $G_j = B_j[t_1 : F_1/x_1, \ldots, t_n : F_n/x_n]$. Then $\mathcal{LP} \vdash G_j$ for every $j \in J$. By the

disjunctive property (corollary 4.5) for every $j \in J$ there exists some i such that the variable x_i occurs in B_j and $\mathcal{LP} \vdash t_i : F_i$. So $B_j(x_1, \ldots, x_n) = 1$ for the evaluation given above. Thus $A(x_1, \ldots, x_n) = 1$ too.

■

Theorem 6.3 *$r\mathcal{LP}_{\wedge,\vee}$ belongs to NP.*

Proof. Lemma 6.2 provides a nondeterministic polynomial time decision procedure for $r\mathcal{LP}_{\wedge,\vee}$. Given a $\{\wedge, \vee\}$-combination of quasiatomic formulas F it is sufficient to guess the corresponding set I and then to test the conditions 1, 2. The size of I is bound by a polynomial in the length of F. For checking the first condition we use the NP-algorithm from the theorem 5.3. The test of the second condition can be performed in polynomial time.

■

References

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

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

[3] S. Artemov, "Explicit provability and constructive semantic", *The Bulletin for Symbolic Logic*, 2001, v. 6, No. 1

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

[5] R. Kuznets, "On the complexity of explicit modal logic", *Lecture Notes in Computer Science*, v. 1862, Proceedings of 14th International Conference of Computer Science Logic, 2000, pp. 371-383.

[6] M. Bidoit, J. Corbin, "A Rehabilitation of Robinson's Unification Algorithm", *Information Processing*, North Holland, 1983, vol. 83, pp. 909–914