2005

TR-2005011: Logic of Proofs for Bounded Arithmetic

Evan Goris

Logic of Proofs for bounded arithmetic

Evan Goris*

October 24, 2005

Abstract

The logic of proofs is known to be complete for the semantics of proofs in PA. In this paper we present a refinement of this theorem, we will show that we can assure that all the operations on proofs can be realized by feasible, that is PTIME-computable, functions. In particular we will show that the logic of proofs is complete for the semantics of proofs in Buss' bounded arithmetic S^1_2.

1 Introduction

In [Art01] the logic LP was shown to be arithmetically complete with respect to the semantics of proofs in PA. In this paper we extend this result to S^1_2. Due to the fact that in [Art01] most calculations and constructions are done outside of the theory in question (PA there and S^1_2 in this case), such an extension might seem straightforward. However, closer inspection of the proof reveals a step which fails for sufficiently weak theories and a refinement is therefore necessary.

Another motivation for a more refined version of the arithmetical completeness theorem is as follows. Recently LP has been applied in the field of epistemic logic [AN05][Art04]. In particular, LP can express, using its proof terms, how hard it is to obtain certain knowledge. In other words, LP has a tool for tackling the logical omniscience problem. However, the length of a proof term can only be a good indication of the complexity of the evidence it represents when the operations that occur in it are feasibly computable. In this paper we show that LP is complete for exactly such a semantics. Namely, each proof term represents a PTIME-computable function.

The paper is organized as follows. In Section 2 we define the logic LP. In Section 3 we discuss S^1_2, the formalization of syntax and provability in S^1_2, the interpretation of LP in S^1_2 and show that LP is complete with respect to this interpretation. The proof is a refinement of the one given in [Art01] for the interpretation of LP in PA.

It is worthwhile of mentioning that very little actual reference is made to the theory S^1_2. Hence the phrase 'bounded arithmetic' instead of S^1_2 in the title.

*Research supported by CUNY Community College Collaborative Incentive Research Grant 91639-0001 "Mathematical Foundations of Knowledge Representation"

Basically all we require from the arithmetical theory is that we have access to a fixed point lemma, can formalize syntax and can construct a proof predicate that satisfies the Hilbert-Bernay-Löb derivability conditions. S_2^1 is (one of) the weakest theories for which this is known to be possible.

Finally the author would like to thank Professor Sergei Artëmov and Walter Dean for useful suggestions and remarks.

2 Logic of Proofs

LP has a long history and many variations. Starting in [AS93][Art94] with its most basic version up to [AN05] where LP is mixed with various modal logics. In the mean time various results are shown ranging from complexity questions [KB05][Kuz00] to the development of semantics [Mkr97][Fit05] and tableau systems [Ren04]. Here we consider the version from [Art01]. An important feature of this logic and its arithmetical completeness theorem is its ability to give a complete arithmetical semantics to S4 via its realization theorem. For an extensive overview of LP we refer the reader to [AB04].

The language of LP consist of the following. We have countably many propositional variables $p_1, p_2, \ldots$, countably many proof variables $x_1, x_2, \ldots$ and countably many axiom constants $c_1, c_2, \ldots$ The following definitions show how we can construct more complex expressions.

Definition 2.1 (LP-terms). We define LP-*terms* as follows.

- Axiom constants and proof variables are terms,
- If s and t are terms then so are $s + t$, $s \cdot t$ and $!t$.

Definition 2.2 (LP-formulas). We define LP-*formulas* as follows.

- $\bot$ and any propositional variable is a formula,
- If A and B are formulas then so is $A \to B$,
- If A is a formula and t is a term then $t{:}A$ is a formula.

Definition 2.3 (LP). As axioms we take all instances of the following schemata.

A0 "Propositional logic",

A1 $t{:}A \to A$,

A2 $s{:}(A \to B) \to t{:}A \to (s{\cdot}t){:}B$,

A3 $s{:}A \to (s + t){:}A \wedge (t + s){:}A$,

A4 $t{:}A \to !t{:}(t{:}A)$.

A5 $c{:}A$, c an axiom constant and A an instance of **A0**-**A4**.

The set of theorems of LP is obtained by closing the set of axioms under modes ponens.

Definition 2.4 (Constant specification). A constant specification in a set of pairs $\langle c, F\rangle$ where c is a proof constant and F an instance of one of $\mathbf{A0}-\mathbf{A4}$.

With $\mathsf{LP}_{\mathcal{CS}}$ we denote the fragment of LP, where $\mathbf{A5}$ is restricted to $c{:}A$ for $\langle c, A\rangle \in \mathcal{CS}$. Let us write $\bigwedge \mathcal{CS}$ for $\bigwedge\{c{:}A \mid \langle c, A\rangle \in \mathcal{CS}\}$. The following is obvious.

$$\mathsf{LP}_{\emptyset} \vdash \bigwedge \mathcal{CS} \to A \Leftrightarrow \mathsf{LP}_{\mathcal{CS}} \vdash A.$$

Let X be a finite set of formulas. Let $\mathcal{T}(X) = \{t \mid \text{ for some } A,\ t{:}A \in X\}$. We say that X is *adequate* when

- X is closed under subformulas,
- X is closed under single negation,
- If $t \in \mathrm{Sub}(\mathcal{T}(X))$ and $A \in X$ then $t{:}A \in X$.

Clearly any finite set of formulas can be extended to a finite adequate set of formulas. Also notice that when X is adequate, then $\mathrm{Sub}(\mathcal{T}(X)) = \mathcal{T}(X)$.

We say that a set of formulas Γ is *inconsistent* if for some $X_1, \ldots, X_k \in \Gamma$ we have $\mathsf{LP} \vdash \neg X_1 \vee \cdots \vee \neg X_k$. A set is *consistent* when it is not inconsistent. We say that a set Γ is *maximal consistent in* X when $\Gamma \subseteq X$, Γ is consistent and if $\Gamma \subsetneq \Gamma' \subseteq X$, then Γ' is inconsistent.

Lemma 2.5. *If Γ is maximal consistent in X then for every $\neg A \in X$ we have $A \in \Gamma$ or $\neg A \in \Gamma$.*

Proof. Suppose for a contradiction that $A \notin \Gamma$ and $\neg A \notin \Gamma$. Then both $\Gamma \cup \{A\}$ and $\Gamma \cup \{\neg A\}$ are contained in X, properly extend Γ and are thus inconsistent. So there are $X_1, \ldots, X_k \in \Gamma \cup \{A\}$ and $Y_1, \ldots, Y_n \in \Gamma \cup \{\neg A\}$ such that $\mathsf{LP} \vdash \neg X_1 \vee \cdots \neg X_k$ and $\mathsf{LP} \vdash \neg Y_1 \vee \cdots \vee \neg Y_n$. By the consistency of Γ we find i, j such that $A = X_i$ and $\neg A = Y_j$. W.l.o.g. we assume $i = j = 1$. But then $\mathsf{LP} \vdash \neg X_2 \vee \cdots \vee \neg X_k \vee \neg Y_2 \vee \cdots \vee \neg Y_n$, and thus Γ is inconsistent. A contradiction. □

The following lemma is an immediate corollary to Lemma 2.5

Lemma 2.6. *Let X be adequate and let Γ be maximally consistent in X. Then*

1. *If $A, A \to B \in \Gamma$ then $B \in \Gamma$,*
2. *If $t{:}A \in \Gamma$ then $A \in \Gamma$,*
3. *If $s{:}(A \to B) \in \Gamma$, $t{:}A \in \Gamma$ and $s \cdot t \in \mathcal{T}(X)$ then $(s \cdot t){:}B \in \Gamma$*
4. *If $s{:}A \in \Gamma$ or $t{:}A \in \Gamma$ then $s + t \in \mathcal{T}(X)$ implies $(s + t){:}A \in \Gamma$*
5. *If $t{:}A \in \Gamma$ and $!t \in \mathcal{T}(X)$ then $!t{:}(t{:}A) \in \Gamma$*

Proof. As an example let us show Item 4. Suppose $s{:}A \in \Gamma$ and $s + t \in \mathcal{T}(X)$. By adequateness of X we have $(s + t){:}A \in X$. Since X is closed under single negation we have $\neg(s + t){:}A \in X$. By Lemma 2.5 we thus have $\neg(s + t){:}A \in \Gamma$ or $(s + t){:}A \in \Gamma$. In the first case Γ would be inconsistent, which is not so and thus $(s + t){:}A \in \Gamma$ as required. □

Notice that in the above two lemmas Γ is not necessarily finite. If we choose X finite and Γ maximally consistent in X, then Γ is finite as well. In [Art01] an explicit algorithm is given that, given a formula A such that $\mathsf{LP} \nvdash A$, constructs a finite Γ that satisfies Lemma 2.6 such that $\neg A \in \Gamma$.

3 Arithmetical interpretation

In this section we introduce the fragment of arithmetic S_2^1 [Bus86][Kra95][Bus98], the formalization of syntax and provability in S_2^1 and the interpretation of LP in S_2^1. The discussion on S_2^1 is basically meant to fix the notation, for a detailed treatment of the subject see [Bus86][Bus98].

3.1 Bounded arithmetic

Bounded arithmetic was introduced by Parikh in his famous paper [Par71] (see also [Bus99]), where he formulated a first-order theory PB that is now known as $I\Delta_0$. The core motivation was to build a theory that somehow captured the informal notion of feasibility and it did so in the sense that $I\Delta_0$ does not guarantee the existence of too fast growing functions like, for example, the exponential function 2^x.

As was already realized by Parikh, $I\Delta_0$ seemed to weak to intensionally formalize its own syntax. A slightly stronger theory $I\Delta_0 + \Omega_1$ was shown in [PW87] to be able to do this. Just as $I\Delta_0$, $I\Delta_0 + \Omega_1$ does not guarantee the existence of a function of exponential growth rate. (It was later shown by Solovay that a great deal, in any case enough for the Gödel incompleteness theorems, could already be done in $I\Delta_0$.)

In this paper we will use the approach of [Bus86]. In [Bus86] the theory S_2, a conservative extension of $I\Delta_0 + \Omega_1$, and its fragments S_2^i were formulated. The theories S_2^i are first-order theories in the language $\mathcal{L} = \{+, \times, \lfloor \frac{x}{2} \rfloor, |\ |, \sharp\}$. The intended meaning of $+$ and $\times$ is as usual. $\lfloor \frac{x}{2} \rfloor$ is the bitwise shift-right operation, $|x|$ is the length of the binary representation of x and the intended meaning of $x \sharp y$ is $2^{|x||y|}$.

A quantifier in a formula is *bounded* if its of the form $\forall x \leq t$ or $\exists x \leq t$, where t is a term. We say that the quantifier is *sharply bounded* if t is of the form $|t'|$. The classes of formulas Σ_i^b are defined as follows.

- A formula is Σ_1^b if all quantifiers are bounded and all universal quantifiers are sharply bounded.
- A formula is Π_1^b if its negation is Σ_1^b.

- The class of formulas Σ^b_{i+1} is the least class that contains $\Sigma^b_i \cup \Pi^b_i$ and is closed under $\wedge$, $\vee$ and bounded existential quantification and sharply bounded universal quantification.
- The class of formulas Π^b_{i+1} is the least class that contains $\Sigma^b_i \cup \Pi^b_i$ and is closed under $\wedge$, $\vee$ and bounded universal quantification and sharply bounded existential quantification.

In addition we define

- A formula is Δ^b_i if it is both Σ^b_i and Π^b_i.

As it turns out, this hierarchy has close relations to the polynomial time hierarchy. For example, the Σ^b_1 definable sets are precisely the NP sets and thus a statement like $\Sigma^b_1 \neq \Pi^b_1$ (that is NP$\neq$co-NP) is, unlike its counterpart in the arithmetical hierarchy, a difficult open questions.

All the (axiomatizations of the) theories S^i_2 contain a basic set of axioms BASIC, defining the recursive properties of the functions in the language. The theory S^i_2 is then axiomatized over BASIC by the polynomial induction scheme for Σ^b_i formulas ϕ:

$$\phi(0) \wedge \forall x \leq y(\phi(\lfloor \frac{x}{2} \rfloor) \rightarrow \phi(x)) \rightarrow \phi(y).$$

The intuition behind this scheme is that in order to verify $\phi(x)$ for some x, then we can do this in $|x|$ (that is a linear) number of steps. (The theory S_2 has no restriction on what formulas can be used in the polynomial induction scheme.) From now on we mainly focus on S^1_2.

- A formula ϕ is Σ^b_1 in S^1_2 if for some Σ^b_1 formula σ we have $\mathsf{S}^1_2 \vdash \phi \leftrightarrow \sigma$.
- A formula ϕ is Π^b_1 in S^1_2 if for some Π^b_1 formula π we have $\mathsf{S}^1_2 \vdash \phi \leftrightarrow \pi$.
- A formula ϕ is Δ^b_1 in S^1_2 if there is a Σ^b_1 formula σ and a Π^b_1 formula ϕ such that $\mathsf{S}^1_2 \vdash \pi \leftrightarrow \sigma$ and $\mathsf{S}^1_2 \vdash \phi \leftrightarrow \sigma$.

Of course, in S^1_2 can proof the polynomial induction scheme for any formula that is Σ^b_1 in S^1_2.

An important relation of bounded arithmetic with complexity theory is as follows.

Theorem 3.1. *If $\sigma(x, y)$ is a Σ^b_1 formula such that $\mathsf{S}^1_2 \vdash \forall x \exists! y \sigma(x, y)$. Then $\sigma(x, y)$ defines the graph of a PTIME-computable function.*

For a proof of this theorem, and its reverse: every PTIME-computable function has a Σ^b_1-definition that is provably total in S^1_2, see [Bus86][Kra95].

One of the fundamental theorems of [Par71] projects to S^1_2 as follows (see [Bus98]).

Theorem 3.2 (Parikh's Theorem). *Let $\sigma(x, y)$ be a Σ^b_1 formula such that $\mathsf{S}^1_2 \vdash \forall x \exists y \sigma(x, y)$. Then there exists a term t such that $\mathsf{S}^1_2 \vdash \forall x \exists y \leq t(x) \delta(x, y)$.*

Since the exponential function majorizes any term of S^1_2, Parikh's theorem implies that exponentiation does not have a provably total Σ^b_1 definition.

One further fundamental result is as follows. Let f be a new function symbol and let $\phi(x,y)$ be a Σ^b_1 formula such that $\mathsf{S}^1_2 \vdash \forall x \exists! y \phi(x,y)$. Let $\Sigma^b_i(f)$ and $\mathsf{S}^1_2(f)$ be defined exactly as Σ^b_1 and S^1_2 but in the language $\mathcal{L} \cup \{f\}$ (in particular f may be used in bounding terms and induction schemes) and $\mathsf{S}^1_2(f)$ has an additional axiom $f(x) = y \leftrightarrow \phi(x,y)$.

Theorem 3.3. *$\mathsf{S}^i_2(f)$ is conservative over S^a_2 nd any $\Sigma^b_i(f)$ formula is $\mathsf{S}^i_2(f)$ equivalent to a Σ^b_i formula.*

In more informal terms, Σ^b_i definable functions can be freely used. A similar statement holds for Δ^b_1 definable predicates. For a proof see [Bus98].

3.2 Formalization of syntax

First some notation. Elements from $\mathbf{N}$ are printed in boldface: $\mathbf{n}$, $\mathbf{m}$ etc. A sequence $x_1, \ldots, x_n$ is usually written as $\overline{x}$. The Gödel number of some syntactic object $\int$ is denoted by $\ulcorner \int \urcorner$. For exact details on such a coding we refer the reader to [Bus86].

A simple but important definition is the canonical representation (numeral), of an element of $\mathbf{N}$. We define

$$\underline{\mathbf{0}} = 0$$
$$\underline{\mathbf{n}} = \begin{cases} S(S(0)) \cdot \underline{\mathbf{m}} & \text{if } \mathbf{n} = 2\mathbf{m} \\ S(S(S(0)) \cdot \underline{\mathbf{m}}) & \text{if } \mathbf{n} = 2\mathbf{m}+1 \end{cases}$$

We can define a Σ^b_1 function $\mathrm{num}(x)$ in S^1_2 such that for any $\mathbf{n} \in \mathbf{N}$ we have

$$\mathrm{num}(\mathbf{n}) = \ulcorner \underline{\mathbf{n}} \urcorner.$$

In addition we can define a function $\mathrm{sub}(x,y,z)$, Σ^b_1 definable in S^1_2, that satisfies the following.

$$\mathsf{S}^1_2 \vdash \mathrm{sub}(\underline{\ulcorner \phi(\overline{y}, x) \urcorner}, \underline{\ulcorner x \urcorner}, \underline{\ulcorner t \urcorner}) = \underline{\ulcorner \phi(\overline{y}, t) \urcorner}. \tag{1}$$

Using such functions one can proof a fixed point theorem [Bus98].

Lemma 3.4. *For any formula $\phi(\overline{x}, y)$ there exists a formula $\epsilon(\overline{x})$ such that*

$$\mathsf{S}^1_2 \vdash \epsilon(\overline{x}) \leftrightarrow \phi(\overline{x}, \ulcorner \epsilon(\overline{x}) \urcorner).$$

From now on we will not make distinction between numerals and numbers (that is, we systematically confuse numerals with elements of the standard model).

In what follows we let $\mathrm{isProof}(x)$ be a Δ^b_1-formula that defines the codes of proofs. It is well known that there exists a Δ^b_1 formula $\mathrm{Proof}(x,y)$ for which we have the following.

$$\mathsf{S}^1_2 \vdash \phi \Leftrightarrow \mathbf{N} \models \exists x \mathrm{Proof}(x, \ulcorner \phi \urcorner). \tag{2}$$

Moreover there exist PTIME computable functions $\oplus$, $\otimes$ and e for which the following holds.

$$\mathbf{N} \models \text{Proof}(x, \ulcorner \phi \urcorner) \wedge \text{Proof}(y, \ulcorner \phi \to \psi \urcorner) \to \text{Proof}(y \otimes x, \ulcorner \psi \urcorner), \tag{3}$$

$$\mathbf{N} \models \text{Proof}(x, \ulcorner \phi \urcorner) \vee \text{Proof}(y, \ulcorner \phi \urcorner) \to \text{Proof}(x \oplus y, \ulcorner \phi \urcorner), \tag{4}$$

$$\mathbf{N} \models \text{Proof}(x, \ulcorner \phi \urcorner) \to \text{Proof}(e(x), \ulcorner \text{Proof}(x, \ulcorner \phi \urcorner) \urcorner). \tag{5}$$

3.3 Arithmetical interpretation of LP

Any Δ_1^b formula $\text{Prf}(x, y)$ that satisfies (2) will be called a *proof predicate*. If it in addition comes with functions $\oplus$, $\otimes$ and e that satisfy the conditions (3), (4) and (5) that way say that it is a *normal proof predicate*. We say that a structure

$$\langle \text{Prf}(x, y), \oplus, \otimes, e, * \rangle$$

is an *arithmetical interpretation* (which we also denote by $*$) when Prf(x,y) is a normal proof predicate with the functions $\oplus$, $\otimes$ and e. Moreover, $*$ is a mapping from propositional variables to sentences of S_2^1 and from proof variables and proof constants to numbers.

Given an arithmetical interpretation $\langle \text{Prf}(x, y), \oplus, \otimes, e, * \rangle$, we can extend the map $*$ to the full language of LP as follows.

- $(s \cdot t)^* = s^* \otimes t^*$, $(s + t)^* = s^* \oplus t^*$ and $(!t)^* = e(t^*)$,
- $(A \to B)^* = A^* \to B^*$,
- $(t{:}A)^* = \text{Prf}(t^*, \ulcorner A^* \urcorner)$.

3.4 Arithmetical completeness

For now we focus on $\mathsf{LP}_\emptyset$. We show that $\mathsf{LP}_\emptyset$ is arithmetically complete. A more general version, for arbitrary constant specifications, will be proved as an corollary in Section 3.5.

Apart from a formalization of the syntax of S_2^1 in S_2^1 we simultaneous assume a disjoint formalization of the syntax of LP. So for any LP formula or term θ we have a code $\ulcorner \theta \urcorner$ and from this code we can deduce (in S_2^1) that it is indeed the code of an LP object (and not an S_2^1 object).

Theorem 3.5. *$\mathsf{LP}_\emptyset \vdash A$ iff for any arithmetical interpretation $*$ we have that $\mathsf{S}_2^1 \vdash A^*$*

The proof of this theorem is what constitutes the rest of this section. Soundness is a direct consequence of the definition of a normal proof predicate and the fact that A^* is Δ_1^b for any $*$. To show completeness assume that $\mathsf{LP}_\emptyset \nvdash A$. Let X be some adequate set such that $\neg A \in X$ and let Γ be maximal consistent in X such that $\neg A \in \Gamma$. (In particular Γ is finite.) We will construct a proof predicate $\text{Prf}(x, y)$, PTIME operations on codes of proofs $\tilde{\oplus}$, $\tilde{\otimes}$ and $\tilde{e}$, and a mapping $*$ from propositional variables to sentences of S_2^1, and from proof variables and

proof constants to $\mathbf{N}$ such that, (when $*$ is extended to the full language of $\mathsf{LP}_\emptyset$ as explained above) $\mathsf{S}^1_2 \vdash A^*$ for all $A \in \Gamma$.

Let us first decide on what objects should serve as 'proofs'. To begin with, all usual proofs are 'proofs'. That is all sequence of formulas, each of which is an axiom of S^1_2, or can be obtain by an inference rule from earlier formulas in the sequence. This way the left to right direction of (2) is easily satisfied. As an extra source of 'proofs' we will use the finite set of LP terms $\mathcal{T}(X)$. The theorems of a 'proof' $t \in \mathcal{T}(X)$ will be the set $\{A \mid t{:}A \in \Gamma\}$.

We now wish to formalize the contents of the last two paragraphs. Let us suppose that we have a formula $\mathrm{Prf}(x, y)$. We define a translation from LP formulas A and LP terms t to S^1_2 sentences $A^\dagger$ and numbers $t^\dagger$ as follows.

1. $p^\dagger \equiv \ulcorner p \urcorner = \ulcorner p \urcorner$, if $p \in \Gamma$ and $p^\dagger \equiv \ulcorner p \urcorner \neq \ulcorner p \urcorner$ otherwise,
2. $t^\dagger = \ulcorner t \urcorner$, for any proof term t,
3. $(A \to B)^\dagger \equiv A^\dagger \to B^\dagger$,
4. $(t{:}A)^\dagger \equiv \mathrm{Prf}(t^\dagger, \ulcorner A^\dagger \urcorner)$.

To carry out the proof as in [Art01], we would like to construct a function $\mathrm{tr}(p, f)$ which, given a code p of a normal proof predicate $\mathrm{Prf}(x, y)$ and a code f of an LP-formula F gives us the code of the S^1_2 sentence $F^\dagger$. There is some difficulty constructing a Σ^b_1 definition of such a function over S^1_2.

Fact 3.6. There exists a sequence $F_0, F_1, F_2, \ldots$ of LP formulas such that for any S^1_2 term $s(x)$, there exists $n \geq 0$ such that $\ulcorner F^\dagger_n \urcorner > s(\ulcorner F_n \urcorner)$.

A proof of this fact can be found in the appendix. By Parikh's theorem [Kra95], such a function cannot be shown total in S^1_2. Since we only need $\ulcorner F^\dagger \urcorner$ for only finitely many F's, (namely those $F \in X$) the following suffices. Let $\mathrm{sub}_{x,y}(p, z_1, z_2)$ be a Σ^b_1 definable and provably total function which satisfies (see (1) above)

$$\mathrm{sub}_{x,y}(\ulcorner \phi(x, y) \urcorner, \ulcorner t_1 \urcorner, \ulcorner t_2 \urcorner) = \ulcorner \phi(t_1, t_2) \urcorner.$$

As usual, we write $\dot{\to}$ for the Σ^b_1 definable and provably total function that satisfies

$$\ulcorner \phi \urcorner \dot{\to} \ulcorner \psi \urcorner = \ulcorner \phi \to \psi \urcorner.$$

For each $F \in X$ we define with induction on F a Δ^b_1-formula $\phi_F(p, x)$, defining the graph of $\mathrm{tr}(p, \ulcorner F \urcorner)$, as follows.

- If $F \equiv \bot$ then
$$\phi_F(p, x) \equiv x = \ulcorner 0 \neq 0 \urcorner$$
.
- If $F \equiv p \in \Gamma$ then
$$\phi_F(p, x) \equiv x = \ulcorner \ulcorner p \urcorner = \ulcorner p \urcorner \urcorner$$
.

- If $F \equiv p \notin \Gamma$ then
$$\phi_F(p,x) \equiv x = \ulcorner \ulcorner p \urcorner \neq \ulcorner p \urcorner \urcorner$$
.

- If $F \equiv F_0 \to F_1$ then
$$\phi_F(p,x) \equiv \exists x_0 x_1 \leq x(\phi_{F_0}(p,x_0) \wedge \phi_{F_1}(p,x_1) \wedge x = x_0 \dot{\to} x_1).$$

- If $F \equiv t{:}F'$ then
$$\phi_F(p,x) \equiv \exists x' \leq x(\phi_{F'}(p,x') \wedge x = \mathrm{sub}_{x,y}(p, \ulcorner t \urcorner, \mathrm{num}(x'))).$$

Let Prf(x,y) satisfy the following fixed point equation (see Lemma 3.4, also recall that Γ is finite).

$$\mathsf{S}_2^1 \vdash \mathrm{Prf}(x,y) \leftrightarrow \mathrm{Proof}(x,y) \vee \bigvee_{t:F \in \Gamma} (\phi_F(\ulcorner \mathrm{Prf}(x,y) \urcorner, y) \wedge x = \ulcorner t \urcorner) \tag{6}$$

For briefity we put

$$T(x) \equiv \bigvee \{x = \ulcorner t \urcorner \mid t \in \mathcal{T}(X)\}$$
$$\phi_F(u) \equiv \phi_F(\ulcorner \mathrm{Prf}(x,y) \urcorner, u)$$

With induction on F one easily shows that for all $F, G \in \Gamma$ we have

$$\mathsf{S}_2^1 \vdash \phi_F(y) \leftrightarrow y = \ulcorner F^\dagger \urcorner \tag{7}$$

and

$$F \not\equiv G \Rightarrow F^\dagger \not\equiv G^\dagger. \tag{8}$$

Lemma 3.7. *Prf(x,y) is Δ_1^b in S_2^1*

Proof. From (7) one easily derives that for each $F \in \Gamma$, ϕ_F is Δ_1^b in S_2^1. □

Lemma 3.8. *For all $F \in X$ we have $F \in \Gamma$ implies $\mathsf{S}_2^1 \vdash F^\dagger$ and $F \notin \Gamma$ implies $\mathsf{S}_2^1 \vdash \neg F^\dagger$.*

Proof. Induction on F.

Case $F \equiv p$. In this case $F^\dagger$ is a true Δ_1^b-sentence when $F \in \Gamma$, and thus F is provable. When $F \notin \Gamma$, then it is a false Δ_1^b-sentence and thus $\neg F$ is provable.

Case $F \equiv t{:}F'$. We have that $\phi_{F'}(\ulcorner F'^\dagger \urcorner)$ is provable. If $F \in \Gamma$ then $\mathrm{Prf}(\ulcorner t \urcorner, \ulcorner F'^\dagger \urcorner)(= (t{:}F')^\dagger)$ is provable. If $F \notin \Gamma$ then by (8) we have that $\phi_{G'}(\ulcorner F' \dagger \urcorner)$ is provably false for any $t{:}G' \in \Gamma$. Since $\ulcorner t \urcorner$ is never the code of a 'real' proof in S_2^1 we also have that $\mathrm{Proof}(\ulcorner t \urcorner, \ulcorner F'^\dagger \urcorner)$ is provably false and thus $\mathrm{Prf}(\ulcorner t \urcorner, \ulcorner F'^\dagger \urcorner)(= (t{:}F')^\dagger)$ is provably false.

Case $F \equiv F_0 \to F_1$. If $F \in \Gamma$, then $F_0 \notin \Gamma$ or $F_1 \in \Gamma$. In either case we obtain from (IH) that $F^\dagger$ is provable. If $F \notin \Gamma$ then both $F_0 \in \Gamma$ and $F_1 \notin \Gamma$. Thus, again, from (IH) we obtain that $\neg F^\dagger$ is provable. □

Thus we have that, for each $F \in \Gamma$, $F^\dagger$ is S^1_2 provable. Since Γ is finite, we can find one single S^1_2 proof that proves them all. So let $\mathbf{g}$ be some number such that for all $A \in \Gamma$ we have

$$\mathbf{N} \models \mathrm{Proof}(\mathbf{g}, \ulcorner A^\dagger \urcorner). \tag{9}$$

Lemma 3.9. $\mathsf{S}^1_2 \vdash \exists x \mathit{Prf}(x, y) \leftrightarrow \exists x \mathit{Proof}(x, y)$

Proof. The right to left direction is clear by (6). For the other direction reason in S^1_2 and assume that for some x we have $\mathrm{Prf}(x, y)$. In the case $\mathrm{Proof}(x, y)$ we are done at once so assume that this is not so. Then by (6) we have

$$\bigvee_{t:F \in \Gamma} (\phi_F(y) \wedge x = \ulcorner t \urcorner).$$

For each $t{:}F \in \Gamma$ we have $\phi_F(y) \to y = \ulcorner F^\dagger \urcorner$ and thus

$$\bigvee_{t:F \in \Gamma} y = \ulcorner F^\dagger \urcorner.$$

Since we have

$$\bigwedge_{F \in \Gamma} \mathrm{Proof}(\mathbf{g}, \ulcorner F^\dagger \urcorner)$$

we conclude that $\mathrm{Proof}(\mathbf{g}, y)$. □

Let $\pi(x)$ be a function such that for all $\mathbf{n}$ and all formulas ϕ for which $\mathbf{N} \models \mathrm{Proof}(\mathbf{n}, \ulcorner \phi \urcorner)$ we have

$$\mathbf{N} \models \mathrm{Proof}(\pi(\mathbf{n}), \mathrm{Proof}(\mathbf{n}, \ulcorner \phi \urcorner) \to \mathrm{Prf}(\mathbf{n}, \ulcorner \phi \urcorner)) \tag{10}$$

Let $\underline{+}$, $\underline{\times}$ and $\underline{!}$ stand for Σ^b_1 definable functions that takes codes $\ulcorner t_0 \urcorner$, $\ulcorner t_1 \urcorner$ of LP terms t_0 and t_1 to codes $\ulcorner t_0 + t_1 \urcorner$, $\ulcorner t_0 \cdot t_1 \urcorner$ and $\ulcorner !t_0 \urcorner$ of the LP terms $t_0 + t_1$, $t_0 \cdot t_1$ and $!t_0$ resp.

Recall that $T(x)$ defines the codes of the terms in $\mathcal{T}(X)$ and that $\mathrm{isProof}(x)$ defines the codes of 'real' proofs in S^1_2. We define the function $\tilde{\oplus}$ as follows.

$$x \tilde{\oplus} y = \begin{cases} x \oplus y & \mathrm{isProof}(x) \wedge \mathrm{isProof}(y) \\ \mathbf{g} \oplus y & T(x) \wedge \mathrm{isProof}(y) \\ x \oplus \mathbf{g} & \mathrm{isProof}(x) \wedge T(y) \\ x \underline{+} y & T(x) \wedge T(y) \wedge T(x \underline{+} y) \\ \mathbf{g} & T(x) \wedge T(y) \wedge \neg T(x \underline{+} y) \end{cases}$$

Similarly we define the function $\tilde{\otimes}$ as follows.

$$x \tilde{\otimes} y = \begin{cases} x \otimes y & \mathrm{isProof}(x) \wedge \mathrm{isProof}(y) \\ \mathbf{g} \otimes y & T(x) \wedge \mathrm{isProof}(y) \\ x \otimes \mathbf{g} & \mathrm{isProof}(x) \wedge T(y) \\ x \underline{\times} y & T(x) \wedge T(y) \wedge T(x \underline{\times} y) \\ \mathbf{g} & T(x) \wedge T(y) \wedge \neg T(x \underline{\times} y) \end{cases}$$

And finally we define the function $\tilde{e}$ as follows.

$$\tilde{e}(x) = \begin{cases} \underline{!}x & T(x) \wedge T(\underline{!}x) \\ \mathbf{g} & T(x) \wedge \neg T(\underline{!}x) \\ \pi(x) \otimes e(x) & \text{isProof}(x) \end{cases}$$

And to finish we define the mapping $*$.

1. $p^* \equiv \ulcorner p \urcorner = \ulcorner p \urcorner$ if $p \in \Gamma$,
2. $p^* \equiv \ulcorner p \urcorner \neq \ulcorner p \urcorner$ if $S \notin \Gamma$,
3. $x^* \equiv \ulcorner x \urcorner$, $a^* = \ulcorner a \urcorner$,

Lemma 3.10. *$\tilde{\oplus}$, $\tilde{\otimes}$ and $\tilde{e}$ are PTIME computable.*

Proof. All functions and predicates occurring in their definitions are PTIME computable. □

Now we have finished the definition of our translation of LP into S_2^1. The relation with our preliminary translation is as follows.

Lemma 3.11. *If $t \in \mathcal{T}(X)$ then $t^\dagger = t^*$. If $F \in X$ then $F^\dagger = F^*$.*

Corollary 3.12. For all $A \in \Gamma$ we have $\mathsf{S}_2^1 \vdash A^*$.

Lemma 3.13. *Prf(x, y) is a normal proof predicate, that is:*

1. $\mathbf{N} \models$ *Prf*$(x, \ulcorner \phi \urcorner) \wedge$ *Prf*$(y, \ulcorner \phi \to \psi \urcorner) \to$ *Prf*$(y \tilde{\otimes} x, \ulcorner \psi \urcorner)$,
2. $\mathbf{N} \models$ *Prf*$(x, \ulcorner \phi \urcorner) \vee$ *Prf*$(y, \ulcorner \phi \urcorner) \to$ *Prf*$(x \tilde{\oplus} y, \ulcorner \phi \urcorner)$
3. $\mathbf{N} \models$ *Prf*$(x, \ulcorner \phi \urcorner) \to$ *Prf*$(\tilde{e}(x), \ulcorner$ *Prf*$(x, \ulcorner \phi \urcorner) \urcorner)$.

Proof. Item 1. Suppose Prf$(x, \ulcorner \phi \urcorner)$ and Prf$(y, \ulcorner \phi \to \psi \urcorner)$ and let $z = y \tilde{\otimes} x$. There are five cases to consider, one for each disjunct in the definition of $\tilde{\otimes}$. *Case 1:* If both x and y code 'real' proofs then so does z and indeed z codes a proof of ψ. *Case 2:* x is the code of some $t \in \mathcal{T}(X)$ and y codes a 'real' proof. In this case Prf$(x, \ulcorner \phi \urcorner)$ implies by (6), (7) and (8) that $\phi = F^\dagger$ for some F with $t{:}F \in \Gamma$. By Lemma 2.6 we have $F \in \Gamma$. So by choice of $\mathbf{g}$ we have Proof$(\mathbf{g}, \ulcorner \phi \urcorner)$ and thus Proof$(y \otimes \mathbf{g}, \ulcorner \psi \urcorner)$. *Case 3:* The case y is the code of a term $s \in \mathcal{T}(X)$ and x codes a 'real' proof is similar to Case 2. *Case 4:* x codes a terms $t \in \mathcal{T}(X)$, y codes a term $s \in \mathcal{T}(X)$ and $s \cdot t \in \mathcal{T}(X)$. We find formulas A and B such that $A^\dagger = \phi$, $B^\dagger = \psi$, $t{:}A \in \Gamma$ and $s{:}(A \to B) \in \Gamma$. Since $s \cdot t \in \mathcal{T}(X)$ we get $(s \cdot t){:}B \in \Gamma$. Thus since $\ulcorner s \cdot t \urcorner = y \tilde{\otimes} x$ we obtain Prf$(z, \ulcorner \psi \urcorner)$. *Case 5:* x codes a term $t \in \mathcal{T}(X)$ and y codes a term $s \in \mathcal{T}(X)$ but $s \cdot t \notin \mathcal{T}(X)$. Again we find formulas A and B as in Case 4 but now we reason as follows. By Lemma 2.6, $t{:}A \in \Gamma$ implies $A \in \Gamma$ and $s{:}(A \to B) \in \Gamma$ implies $A \to B \in \Gamma$. Thus, again using Lemma 2.6 we obtain $B \in \Gamma$. By our choice of $\mathbf{g}$ we have and since $z = \mathbf{g}$ and $\psi \equiv B^\dagger$ we get Proof$(z, \ulcorner \psi \urcorner)$.

Item 2 is shown completely similar to Item 1 so let us show Item 3. Suppose $\mathrm{Prf}(x, \ulcorner \phi \urcorner)$. There are three cases, corresponding to the three disjunct that make up $\tilde{e}$, to consider. *Case 1:* isProof(x). In this case

$$\tilde{e}(x) = \pi(x) \otimes e(x) \text{ and } \mathrm{Proof}(x, \ulcorner \phi \urcorner).$$

We thus have

$$\mathrm{Proof}(e(x), \ulcorner \mathrm{Proof}(x, \ulcorner \phi \urcorner) \urcorner),$$

and

$$\mathrm{Proof}(\pi(x), \ulcorner \mathrm{Proof}(x, \ulcorner \phi \urcorner) \to \mathrm{Prf}(x, \ulcorner \phi \urcorner) \urcorner).$$

Clearly now $\mathrm{Proof}(\pi(x) \otimes e(x), \ulcorner \mathrm{Prf}(x, \ulcorner \phi \urcorner) \urcorner)$, and thus

$$\mathrm{Prf}(\pi(x) \otimes e(x), \ulcorner \mathrm{Prf}(x, \ulcorner \phi \urcorner) \urcorner).$$

Case 2: $T(x)$ and $T(\underline{!}x)$. In this case

$$\tilde{e}(x) = \underline{!}x$$

And by (6) we have for some $t{:}A \in \Gamma$ that

$$x = \ulcorner t \urcorner \text{ and } \phi = A^\dagger.$$

But since $!t \in \mathcal{T}(X)$ we also have that $!t{:}t{:}A \in \Gamma$, and thus $\mathrm{Prf}(\ulcorner !t \urcorner, \ulcorner (t{:}A)^\dagger \urcorner)$ is true. That is,

$$\mathrm{Prf}(\underline{!}x, \ulcorner \mathrm{Prf}(x, \ulcorner \phi \urcorner) \urcorner)$$

is true.

Case 3: $T(x)$ but $\neg T(\underline{!}x)$. In this case

$$\tilde{e}(x) = \mathbf{g}.$$

Again we have for some $t{:}A \in \Gamma$ that

$$x = \ulcorner t \urcorner \text{ and } \phi = A^\dagger.$$

So $(t{:}A)^\dagger = \mathrm{Prf}(x, \ulcorner \phi \urcorner)$, so by choice of $\mathbf{g}$ we have $\mathrm{Proof}(\mathbf{g}, \mathrm{Prf}(x, \ulcorner \phi \urcorner))$. □

Proof of Theorem 3.4. Lemmata 3.7 and 3.9 show that Prf is a proof predicate and Lemma 3.13 shows that it is normal. Finally Lemma 3.8 shows that, if S_2^1 is consistent, $\mathsf{S}_2^1 \neg \vdash A^*$. □

3.5 Axiom Specifications

In this section we abandon the restriction $\mathcal{CS} = \emptyset$. Let $\mathcal{CS}$ be some axiom specification. We say that an arithmetical interpretation $* = \langle \mathrm{Prf}(x, y), \oplus, \otimes, e, * \rangle$ meets an axioms specification $\mathcal{CS}$ when for any $\langle c, A \rangle \in \mathcal{CS}$ we have $\mathsf{S}_2^1 \vdash \mathrm{Prf}(c^*, A^*)$.

Theorem 3.14. $\mathsf{LP}_{\mathcal{CS}} \vdash A$ *iff for any arithmetical interpretation* $*$ *that meets* $\mathcal{CS}$ *we have* $\mathsf{S}_2^1 \vdash A^*$

Proof. Again, soundness is an easy consequence of the definitions. Suppose now that $\mathsf{LP}_{\mathcal{CS}} \nvdash A^*$. Then $\mathsf{LP}_\emptyset \nvdash \bigwedge \mathcal{CS} \to A$. As we have seen above this gives us an arithmetical interpretation $*$ such that $\mathsf{S}_2^1 \vdash (\bigwedge \mathcal{CS})^* \wedge \neg A^*$. Clearly $*$ meets $\mathcal{CS}$. □

A Appendix

Here we will prove Fact 3.6. Obviously a proof of such a fact requires us to be more precise about the coding of syntax. We will adopt the coding as presented in [Kra95]. A pair $\langle x, y\rangle$ of numbers x and y is coded as follows.

$$\langle x, y\rangle = \lfloor \frac{(x+y)(x+y+1)}{2} \rfloor + x$$

Using the inequality

$$(a+b)^2(c+d)^2 \leq 4(a^2+b^2)(c^2+d^2)$$

the following lemma is easily shown.

Lemma A.1. *For all* a, b, $\frac{(a+b)^2}{2} \leq \langle a, b\rangle \leq 2(a+b)^2$

Sequences are coded as tuples $\langle x, y\rangle$, where x is simply the concatenation of codes of words and y encodes where in x a word ends and a new one starts [Kra95]. That we indeed can give an intensional coding of sequences in this way is not entirely trivial but not of direct interest for our current goal. What is important for us now (and obvious given the way sequences are coded) is that if $\int$ is a sequence then

$$\text{If } \ulcorner \int \urcorner = \langle x, y\rangle \text{ then } \text{length}(\int) \leq |y| = |x|.$$

Corollary A.2. If $\text{len}(s) \geq 1$ then $s \geq \lfloor \frac{2^{\text{len}(s)}}{2} \rfloor$

Proof. Choose a, b such that $s = \langle a, b\rangle$. We have $1 \leq |a| = |b| \leq \text{len}(s)$ and thus $a + b \geq 2^{|a|-1} + 2^{|b|-1} = 2^{|a|}$. Thus $\lfloor \frac{(a+b)^2}{2} \rfloor \geq \lfloor \frac{2^{2|a|}}{2} \rfloor$ and by Lemma A.1 it follows that $s \geq \lfloor \frac{2^{2|a|}}{2} \rfloor$. □

Corollary A.3. For any N there exists K such that if $\text{len}(s) \geq 1$ and if for all $i < \text{len}(s)$, $(s)_i \leq N$ then $s \leq 2^{K\text{len}(s)}$

Proof. Fix N and choose some K such that for all $x \geq 1$ we have $2^{4Nx+1} \leq 2^{Kx}$. Now pick some s with the properties as stated. Let a, b such that $s = \langle a, b\rangle$. Then $|a| = |b| \leq N\text{len}(s)$. Thus $(a+b) \leq 2^{|a|} + 2|b| \leq 2^{2N\text{len}(s)}$. By choice of K we have $2^{4N\text{len}(s)+1} \leq 2^{K\text{len}(s)}$ and thus the claim follows by Lemma A.1. □

Lemma A.4. $num^{(n)}(k) \geq \lfloor \frac{2^{|k|2^n}}{2} \rfloor$

Proof. Induction on n. If $n = 1$ then the bound follows from Corollary A.2 and the fact that $\text{num}(k)$ codes a sequence of length at least $|k|$. Now suppose the

lemma holds for $n \geq 1$.

$$\begin{aligned}\mathrm{num}^{(n+1)}(k) &= \mathrm{num}^{(n)}(\mathrm{num}(k)) \\ &\geq \lfloor \frac{2^{|\mathrm{num}(k)|2^n}}{2} \rfloor \\ &\geq \lfloor \frac{2^{2|k|2^n}}{2} \rfloor \\ &= \lfloor \frac{2^{|k|2^{n+1}}}{2} \rfloor\end{aligned}$$

□

Proof of Fact 3.6: Fix some proof variable x, some propositional variable p and define $F_0 = p$ and $F_{i+1} = x{:}F_i$. We will show that

$$\ulcorner F_n^\dagger \urcorner \geq \mathrm{num}^{(n)}(\ulcorner p \urcorner). \tag{11}$$

For $n = 0$ this is clear so assume it to be true for $n \geq 0$. We have

$$\begin{aligned}\ulcorner F_{n+1}^\dagger \urcorner &= \mathrm{sub}(\ulcorner \mathrm{Prf}(\ulcorner x \urcorner, y) \urcorner, \ulcorner y \urcorner, \mathrm{num}(\ulcorner F_n^\dagger \urcorner)) \\ &\geq \mathrm{num}(\ulcorner F_n^\dagger \urcorner) \\ &\geq \mathrm{num}^{(n+1)}(\ulcorner p \urcorner)\end{aligned}$$

By Lemma A.4 and (11) we obtain $\ulcorner F_n^\dagger \urcorner \geq 2^{|\ulcorner p \urcorner|2^n - 1}$. Next we show that for some K, that does not depend on n, we have

$$\ulcorner F_n \urcorner \leq 2^{Kn}. \tag{12}$$

$\ulcorner F_n \urcorner$ is nothing more that the code of a sequence of length linear in n. We only use finitely many symbols. So let N be the maximal code of the symbols used in the formulas F_n. Corollary A.3 gives us the desired K. Combining the above we get

$$\ulcorner F_n^\dagger \urcorner \geq 2^{|\ulcorner p \urcorner| \sqrt[K]{\ulcorner F_n \urcorner} - 1}$$

□

References

[AB04] S. N. Artëmov and L. D. Beklemishev. Provability logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 13, pages 229–403. Kluwer, 2nd edition, 2004.

[AN05] S. N. Artëmov and E. Nogina. On epistemic logic with justification. In R. van der Meyden, editor, *Theoretical Aspects of Rationality and Knowledge. Proceedings of the Tenth Conference (TARK 2005), June 10-12, 2005, Singapore.*, pages 279–294. National University of Singapore, 2005.

[Art94] S. N. Artëmov. Logic of proofs. *Annals of Pure and Applied Logic*, 67:29–59, 1994.

[Art01] S. N. Artëmov. Explicit provability and constructive semantics. *Bullitin of Symbolic Logic*, 7(1):1–36, 2001.

[Art04] S. N. Artëmov. Evidence-based common knowledge. Technical Report TR-2004018, CUNY Ph.D. Program in Computer Science, 2004.

[AS93] S. N. Artëmov and T. Straßen. The basic logic of proofs. In *Lecture Notes in Computer Science*, volume 702, pages 14–28. Springer-Verlag, 1993.

[Bus86] S. R. Buss. *Bounded arithmetic*. Bibliopolis, Napels, 1986. Revision of PhD. thesis.

[Bus98] S. R. Buss. First-Order Theory of Arithmetic. In S. R. Buss, editor, *Handbook of Proof Theory.* Studies in Logic and the Foundations of Mathematics, Vol.137., pages 475–546. Elsevier, Amsterdam, 1998.

[Bus99] S. R. Buss. Bounded arithmetic, proof complexity and two papers of Parikh. *Annals of Pure and Applied Logic*, 96:43–55, 1999.

[Fit05] M. Fitting. The logic of proofs, semantically. *Annals of Pure and Applied Logic*, 132:1–25, 2005.

[KB05] R. Kuznets and V. Brezhnev. Making knowledge explicit: How hard it is. Technical Report TR–2005003, CUNY Ph.D. Program in Computer Science, 2005.

[Kra95] J. Krajíček. *Bounded arithmetic, propositional logic, and complexity theory.* Cambridge University Press, New York, NY, USA, 1995.

[Kuz00] R. Kuznets. On the complexity of explicit modal logic. In *Proceedings of the 14th Internetional Conference of Computer Science Logic*, volume 1862 of *Lecture Notes in Computer Science*, pages 371–383, 2000.

[Mkr97] A. Mkrtychev. Models for the logic of proofs. In *Logical foundations of computer science*, volume 1234 of *Lecture Notes in Computer Science*, pages 266–275, 1997.

[Par71] J. B. Paris. Existence and feasability in arithmetic. *Journal of Symbolic Logic*, 36:494–508, 1971.

[PW87] J. B. Paris and A. J. Wilkie. On the scheme of induction for bounded arithmetic formulas. *Annals of Pure and Applied Logic*, 35:261–302, 1987.

[Ren04] B. Renne. Tableaux for the logic of proofs. Technical Report TR-2004001, CUNY Ph.D. Program in Computer Science, 2004.