QUANTIFIED PROPOSITIONAL CALCULI AND FRAGMENTS OF BOUNDED ARITHMETIC

by Jan Krajíček and Pavel Pudlák in Praha (Czechoslovakia)

§ 0. Introduction

The motivation for this paper comes from a well-known and probably very difficult problem whether Bounded Arithmetic is finitely axiomatizable. Attempts to solve this problem using the machinery of mathematical logic have failed so far. It is possible that the problem can be solved by combining logic with combinatorics. This would require a transformation onto a more combinatorial problem. The finite axiomatizability of Bounded Arithmetic seems to be tightly connected with the problem whether Polynomial Hierarchy collapses to some level $\Sigma^p_1$, but no implication relating these two problems has been proved.\(^1\) Here we present a different problem of a combinatorial character and prove a relation between this problem and the problem of the finite axiomatizability of Bounded Arithmetic.

Cook [4] introduced an equational theory PV of polynomial time computable functions and showed an interesting relation between PV and propositional proof system ER (Extended Resolution). He showed that (1) PV proves soundness of ER and (2) the translation of the equalities provable in PV into propositional calculus have polynomially long proofs in ER. Buss [1] showed that $S^1_2$ (a fragment of the bounded arithmetic $S_2$) is conservative over PV; thus this relation is transferred to $S^1_2$.

The finite axiomatizability of $S_2$ is equivalent to the question whether the hierarchy $S^i_2$, $i = 1, 2, \ldots$, is increasing. We shall construct propositional proof systems $G_i$ which have similar relation to $S^{i+1}_2$ for $i \geq 1$ as ER has to $S^1_2$. Then we show that the problem about the hierarchy $S^i_2$, $i = 1, 2, \ldots$, can be reduced to a problem about the length of proofs in proof systems $G_i$, $i = 1, 2, \ldots$. The systems $G_i$ are natural extensions of a Gentzen system for the propositional logic to quantified propositional formulas with at most $i$ quantifier alternations.

The problem about $G_i$'s would require proving superpolynomial lower bounds to the length of proofs in these systems. This seems too difficult at present, as exponential lower bounds have been proved only for quite a weak system Resolution System (not extended) so far, cf. Haken [8]. However we shall show that nontrivial statements about $S_2$ and its fragments can be derived from this relation, in particular:

1) For $i > j \geq 2$ the $\forall \Sigma^p_j$-consequences if $S^i_2$ are finitely axiomatizable (Corollary 7.1),

2) for $i \geq 1$, if $S^{i+1}_2 \vdash "NP = co-NP"$, then $G_i$ proves all tautologies by proofs of polynomial length (Corollary 7.3).

(Wilkie [11] proved statement (2) for $S^1_2$ and a Frege system with the substitution rule instead of $G_0$.)

\(^1\) Added in proof: Recently Krajíček, Pudlák and Takeuti proved that $T_2^i = S^{i+1}_2$ implies $\Sigma^p_{i+2} = \Pi^p_{i+2}$ for $i \geq 1$. 
After writing the first draft of this paper (January 1988) we learned about the work of Martin Dowd [6], [7]. In [7] he gave a full proof of Cook’s theorem mentioned above and showed the same relation between the quantified propositional calculus (in our notation G) and Polynomial Space Arithmetic (PSA, an equational theory extending PV). In [6] he stated without proof a theorem which relates the fragments of $S_2$ to fragments of the quantified propositional calculus. He did not derive any corollaries of this theorem such as (1) and (2) above.

Throughout the paper we assume knowledge of Buss [1], nevertheless we recall briefly some basic definitions.

§ 1. Preliminaries

The class of quantified propositional formulas (shortly propositions) is the least class of formulas containing the atoms $p_0, p_1, \ldots$, constants 0 (falsity) and 1 (truth), closed under the connectives $\land, \lor, \neg$ and $\supset$ and with any proposition $A(p)$ containing also propositions $\exists x A(x)$ and $\forall x A(x)$, where $x$ substitutes for some occurrence of $p$ in $A(p)$. The semantical meaning of $\exists x A(x)$ is $A(0) \lor A(1)$ and of $\forall x A(x)$ is $A(0) \land A(1)$.

We shall use the usual distinction between bounded and free atoms as is the distinction between bounded and free variables in first order logic (cf. Takeuti [10]).

As usual we assume that the indices $i$ in $p_i$ and $x_i$ are written in the binary notation. Hence the lengths $|p_i|$ and $|x_i|$ of $p_i$ and $x_i$ are proportional to $\log_2(i)$.

We do not include $=\,\equiv$ among the basic connectives but we shall occasionally use $A \equiv B$ as the abbreviation for $(A \supset B) \land (B \supset A)$.

$\Sigma^i_0, \Pi^i_0 (i \geq 0)$ is a hierarchy of propositions defined similarly as is the arithmetical hierarchy:

$\Sigma^0_0, \Pi^0_0$ is the class of quantifier free propositions. Both $\Sigma^i_0$ and $\Pi^i_0$ are closed under $\land, \lor$ and the negation of a $\Sigma^i_0$-proposition is $\Pi^i_0$ and vice versa. $\Sigma^i_0$, contains both $\Sigma^i_0$ and $\Pi^i_0$ and propositions of the form $\exists x A(x)$, for $A \in \Pi^i_0$. Similarly $\Pi^i_0$ contains both $\Sigma^i_0$ and $\Pi^i_0$ and propositions of the form $\forall x A(x)$, for $A \in \Sigma^i_0$. For $A$ in $\Sigma^i_0$ respectively $B$ in $\Pi^i_0$ the propositions $\exists x A$ and $\forall x B$ are in $\Sigma^i_0$ and $\Pi^i_0$ respectively, too.

Thus a proposition in a prenex form with $i$ blocks of the like quantifiers and with the prefix beginning with the block of $\exists$'s is in $\Sigma^i_0$.

We shall consider systems of bounded arithmetic introduced by Buss [1]. Theory $S_2$ is equivalent to (more precisely conservative over) $L_0^\infty + \forall x \exists y (y = x^{\log_2(x+1)})$. The formulas in the hierarchy of formulas $\Sigma^i_0, \Pi^i_0$ define sets which are in the corresponding levels of the polynomial hierarchy $\Sigma^i_0, \Pi^i_0$. The fragments $S_2^i$ are obtained from $S_2$ by restricting the PIND-schema to $\Sigma^i_0$ formulas. The schema PIND is

$$\phi(0) \land \forall x (\phi(x/2) \supset \phi(x)) \supset \forall x \phi(x).$$

Thus the $S_2^i$ is the finite set of open formulas $\text{BASIC}$ plus $\Sigma^i_0$-PIND. The fragments $T_2^i$ are defined similarly but with the ordinary schema of induction. The system $S_2$ is the union of $S_2^i$, $i = 1, 2, \ldots$, and is equivalent to the union of $T_2^i$, $i = 1, 2, \ldots$ For the details see [1].

It is well known that formulas $\Sigma^i_0, \Pi^i_0$ are satisfied by the standard model of arithmetic.

Lemma

(i) $\text{Sat}_{0}$
(ii) $\text{Sat}_{1}$
(iii) $\text{Taut}_{1}$
(iv) $\text{Taut}_{2}$
(v) $\text{Taut}_{3}$

Lemma

(i) $\text{Sat}_{0}$
(ii) $\text{Taut}_{0}$

Definition

We shall consider such cases where $T_2^i$ instead of $P_2^i$.

The length of formulas $\phi(x, y)$ is $\log_2(x+y)$.

We shall consider such cases where $T_2^i$ instead of $P_2^i$.

Definition

The length of formulas $\phi(x, y)$ is $\log_2(x+y)$.

We shall consider such cases where $T_2^i$ instead of $P_2^i$.

Definition

Lemm
It is well known that the syntax can be easily formalized in $S_2$. In particular, formulas $\text{Sat}_i(A, \tau)$ and $\text{Taut}_i(A)$ can be constructed in $S_2$, formalizing "$\Sigma^b_i \cup \Pi^b_i$-proposition $A$ is satisfied by the truth valuation $\tau$" and "$\Sigma^b_i \cup \Pi^b_i$-proposition $A$ is satisfied by all truth valuations", respectively. As such constructions are quite standard (using recursion on notation) we shall only state the properties of such a formalization.

**Lemma 1.1.**

(i) Sat$_0$ is $A^b_1$ with respect to $S^b_2$.

(ii) Sat$_i$ is $\mathcal{B}(\Sigma^b_i)$ for $i \geq 1$.

(iii) Taut$_i$ is $\Pi^b_{i+1}$ and for $i \geq 1$ also $\forall \Sigma^b_i$.

($\mathcal{B}(X)$ denotes the class of Boolean combinations of formulas from $X$.)

**Lemma 1.2.** For $i \geq 0$, $S^b_i$ proves that for all propositions $A, B$ of appropriate complexity and for all $k$ it holds that

(i) $\text{Sat}_i (A \circ B, \tau) \equiv \text{Sat}_i (A, \tau) \circ \text{Sat}_i (B, \tau)$, for $\circ = \land, \lor, \Rightarrow$ and $\text{Sat}_i (\neg A, \tau) \equiv \neg \text{Sat}_i (A, \tau)$;

(ii) $\text{Sat}_i (\exists x A(x), \tau) \equiv \exists x \text{Sat}_i (A(0) \lor A(1), \tau) \equiv (\exists \varepsilon \leq 1) \text{Sat}_i (A(p), \tau' \langle p, \varepsilon \rangle)$

and analogically for $\forall$, where $\tau' \langle p, \varepsilon \rangle$ is the truth valuation $\tau'$ extending $\tau$ by putting $\tau'(p) = \varepsilon$, and $p$ does not occur in $\exists x A(x)$;

(iii) $\text{Sat}_{i+1} (A, \tau) \equiv \text{Sat}_i (A, \tau)$, for $A \in \Sigma^b_i \cup \Pi^b_i$;

(iv) $\text{Sat}_i (\exists x_1 \ldots \exists x_k A(x_1, \ldots, x_k), \tau) \\
\quad \equiv \exists \tau' (\tau' = \langle \langle p_1, \varepsilon_1 \rangle, \ldots, \langle p_k, \varepsilon_k \rangle \rangle) \land \varepsilon_j \leq 1 \land \text{Sat}_i (A(p_1, \ldots, p_k), \tau' \langle p, \varepsilon \rangle)$

where $p_1, \ldots, p_k$ do not occur in $\exists x_1 \ldots \exists x_k A(x_1, \ldots, x_k)$, and analogically for $\forall$;

(v) $\text{Sat}_i (\lor_{(\varepsilon_1, \ldots, \varepsilon_k) \in S} A(p_j | \varepsilon_j), \tau) \equiv (\exists \varepsilon_1, \ldots, \varepsilon_k) \in S) \text{Sat}_i (A(p_j | \varepsilon_j), \tau)$

and analogically for $\land$, where $S$ is a subset of $\{0, 1\}^k$.

**Definition.** A polynomial time computable binary relation $P(x, y)$ is a quantified propositional proof system (shortly: proof system) iff $\exists d P(d, A)$ implies $A \in \bigcup_{i \geq 0} \text{Taut}_i$, where $\text{Taut}_i$ is the set of tautological $\Sigma^b_i$-propositions. We shall write $d : P \vdash A$ instead of $P(d, A)$ and we shall call $d$ a $P$-proof of $A$.

The length of a formula or a proof will be denoted by $|A|, |d|$, respectively. We think of formulas and proofs as 0-1 sequences, thus we can use the same symbol as it is used for $\log_2(x + 1)$ in [1].

We shall often use statements about proof systems in fragments of arithmetic. In such cases we shall tacitly assume that we have a fixed arithmetical definition of $P$, which is $A^b_1$ in $S^b_2$.

**Definition.** For $P$ a proof system and $i \geq 0$, $i$-$\text{RFN}(P)$ is the formula

$(\forall d, A, \tau) \ (d : P \vdash A \land A \in \Sigma^b_i \Rightarrow \text{Sat}_i (A, \tau))$.

**Lemma 1.3.** For $i \geq 1$, $i$-$\text{RFN}(P)$ is an $\forall \Sigma^b_i$-sentence, and 0-$\text{RFN}(P)$ is an $\forall \Pi^b_i$-sentence.
Definition. For \( P, Q \) proof systems and \( i \geq 0 \), \( P \) \( i \)-polynomially simulates \( Q \) iff there is a polynomial time computable function \( f(x, y) \) such that for any \( \Sigma^i \)-proposition \( A \), if \( d : Q \vdash A \) then \( f(d, A) : P \vdash A \). \( P \geq^i Q \) will denote "\( P \) \( i \)-polynomially simulates \( Q \)" and \( P \sim^i Q \) will denote the conjunction of \( P \geq^i Q \) and \( Q \geq^i P \).

This notion generalizes the notion of polynomial simulation introduced by Cook-Reckhow [5].

Finally let us recall some standard proof systems: Frege system \( F \), extended Frege system \( EF \), Frege system with substitution \( SF \) and extended resolution \( ER \) (cf. [5]).

§ 2. Quantified propositional calculi

Proof systems for quantified propositional calculus have been considered several times; for the history see Church [3]. We shall define a system \( G \) and its fragments \( G_i \), for \( i \geq 0 \). Our system is similar to that considered by Dowd [6, 7], which in turn is based on some earlier ones.

The calculus \( G \) is defined in a sequential manner analogically to the definition of \( LK \) in Takeuti [10]. The important difference is that a sequent may be a premise of more than one inferences. Thus proof figures of \( G \)-proofs are not trees but directed graphs.

The calculus \( G \) works with sequents of propositions. The rules of the calculus \( G \) are

(a) the rule of initial sequent,
(b) structural rules,
(c) cut rule,
(d) propositional rules,
(e) quantifier rules.

Now we shall describe the rules explicitly.

(a) The initial sequents are the sequents of the form \( p \rightarrow p, 0 \rightarrow, \rightarrow 1 \), for \( p \) a free atom.

The rules (b), (c), (d) are identical with those of Takeuti [10].

(e) Quantifier rules are

\[
(\forall \text{ left}) \quad \frac{A(B), \Gamma \rightarrow \Delta}{\forall x A(x), \Gamma \rightarrow \Delta}, \quad (\forall \text{ right}) \quad \frac{\Gamma \rightarrow \Delta, A(p)}{\Gamma \rightarrow \Delta, \forall x A(x)},
\]

\[
(3 \text{ left}) \quad \frac{A(p), \Gamma \rightarrow \Delta}{\exists x A(x), \Gamma \rightarrow \Delta}, \quad (3 \text{ right}) \quad \frac{\Gamma \rightarrow \Delta, A(B)}{\Gamma \rightarrow \Delta, \exists x A(x)},
\]

with the proviso that \( p \) does not occur in the lower sequents of \( (\forall \text{ right}) \) and \( (3 \text{ left}) \).

The \( G \)-proofs are sequences of sequents satisfying obvious conditions.

For \( i \geq 0 \) define \( G_i \) by \( d : G_i \vdash \Gamma \rightarrow \Delta \) iff \( d : G \vdash \Gamma \rightarrow \Delta \) and all propositions occurring in \( d \) are in \( \Sigma^i \cup \Pi^i \). In particular, \( G_i \vdash A \) (i.e. \( G_i \vdash \rightarrow A \)) implies \( A \in \Sigma^i \cup \Pi^i \). This completes the definition of the calculi that we shall need.
Now we shall show that the substitution rule can be polynomially simulated in $G$ and each fragment $G_i$, for $i \geq 1$. We assume that only quantifier free propositions may be substituted. (This is needed for the proof of Lemma 2.1.) Clearly it is sufficient to consider only the following special case of the substitution rule

\[
\frac{\Gamma(p) \rightarrow \Delta(p)}{\Gamma(A) \rightarrow \Delta(A)}
\]

where $A$ is a quantifier free proposition which does not contain $p$ and is substituted for all occurrences of $p$ in $\Gamma(p) \rightarrow \Delta(p)$.

**Lemma 2.1.** Let SG and $SG_i$ be the systems $G$ and $G_i$ augmented with the substitution rule. Then for $i \geq 0$, $G \sim^i SG$, and for $i \geq 1$, $G_i \sim^i SG_i$. Moreover, these facts are provable in $S^i_1$.

**Proof.** Clearly we need only to show the simulation of the substitution rule in $G_i$. Consider a substitution of the form ($\ast$). Thus we have a proof of

\[Z_1: \Gamma(p) \rightarrow \Delta(p)\]

in $G_i$ and we want to derive

\[Z_2: \Gamma(A) \rightarrow \Delta(A)\]

in $G_i$. Using the induction on the length of $\Gamma$ and $\Delta$ one can show that

\[Z_3: p \equiv A, \Delta(p), \Gamma(A) \rightarrow \Delta(A), \]

\[Z_4: p \equiv A, \Gamma(A) \rightarrow \Delta(A), \Gamma(p), \]

\[Z_5: \exists x(x \equiv A)\]

are derivable in $G_i$ by proofs whose size is polynomial in the length of $\Gamma, \Delta, A$. Applying the cut-rule to $Z_1, Z_4$ we obtain

\[Z_6: p \equiv A, \Gamma(A) \rightarrow \Delta(A), \Delta(p), \]

and applying it again to $Z_3$ and $Z_6$ we obtain

\[Z_7: p \equiv A, \Gamma(A) \rightarrow \Delta(A)\].

Using ($\exists$: left) we get

\[Z_8: \exists x(x \equiv A), \Gamma(A) \rightarrow \Delta(A)\].

Thus $Z_2$ follows from $Z_3$ and $Z_8$ by cut. In this way the proof is increased only by an additive factor which is polynomial in the length of $\Gamma, \Delta, A$. Hence it is a polynomial simulation. Since all the transformations are elementary, they can be performed in $S^i_1$. \(\Box\)

For $G_0$ and $SG_0$ it is an open problem whether $G_0$ polynomially simulates $SG_0$. We know only the following relations:

**Lemma 2.2.** $S^i_2$ proves

(i) $G_0 \sim^0 F$, (ii) $SG_0 \sim^0 SF \sim^0 ER \sim^0 EF$.

**Proof.** $G_0 \sim^0 F$, $SF \geq^0 EF$ have been shown in [5]. $SG_0 \sim^0 SF$ and $ER \sim^0 EF$ are easy. $EF \geq^0 SF$ has been shown in [6], [9]. \(\Box\)

**Corollary 2.3.** $S^i_2$ proves $G_i \geq^0 ER$. \(\Box\)
§ 3. Translation of bounded formulas into propositions

We define a translation of bounded formulas into propositions. The translation we use is a generalization of the translation used in Cook [3], Dowd [6], and Krajiček-Pudlák [9].

For \( k \geq 0 \) define \( k(i) = 0 \) or \( 1 \), the \( i \)-th digit of \( k \), by \( k = \sum_{i \geq 0} k(i) \cdot 2^i \). Observe that for \( i > |k| \), \( k(i) = 0 \). Sometimes we shall use the following abbreviations: For a proposition \( B \) with free atoms \( p_0, p_1, \ldots \) and \( k \geq 0 \), we abbreviate \( B(p_0/k(0), p_1/k(1), \ldots) \) by \( B(p/k) \) or simply \( B(k) \).

Take a bounded formula \( A(a_1, \ldots, a_k) \). As all functions in the language of \( S_2 \) are polynomial time computable, there exists a polynomial \( p_A(x) \) such that for any \( n_1, \ldots, n_k \) with \( |n_1|, \ldots, |n_k| \leq m \) one needs to compute only numbers with the length \( \leq p_A(m) \) in order to decide the truth value of \( A(n_1, \ldots, n_k) \). This is proved by induction on the complexity of the terms occurring in \( A \) and the complexity of \( A \).

Any polynomial \( q(x) \) satisfying \( \forall x(q(x) \geq p_A(x)) \) will be called a bounding polynomial of \( A \).

For any bounded polynomial \( q(x) \) of \( A \) we shall construct a sequence of propositions \([A]_{m,m}, m \geq 0\), with the following property (we shall occasionally omit the indices \( m, q(m) \), if there is no danger of confusion): If \( a_1, \ldots, a_k \) are all free variables of \( A \) then the only free atoms of \([A]_{m,m} \) are \( p_0(m), \ldots, p_1(m), \ldots, p_2(m), \ldots, p_k(m) \) and for any \( n_1, \ldots, n_k \) with \( |n_1|, \ldots, |n_k| \leq m \) it holds:

\[ A(a_i/n_i) \text{ is true if and only if } [A]_{m,m}(p_i/n_i) \text{ is true.} \]

Moreover, we want the following properties of \([A] \) which we state as a lemma.

Lemma 3.1. For \( A \in \Sigma^b_1 \), \( i \geq 0 \), we have:

1. \([A] \in \Delta^b_i \) with respect to \( G_i \) for \( i = 0 \), and \([A] \in \Sigma^b_i \) for \( i \geq 1 \);
2. \([A]_{m,m} \] \( \leq r(m) \), for some polynomial \( r(x) \) depending only on \( A \) and \( q(x) \);
3. \([A \circ B] \) is \([A] \circ [B] \) for \( \circ = \land, \lor, \neg \), \( \iff \); \( \neg [A] \) is \( \neg [A] \);
4. \((\exists x \leq |t|) A(x) \) is \( \exists y \in S \) \([a \leq |t| \land A(a)](p_i/e_i) \),
5. \((\forall x \leq t) A(x) \) is \( \forall y \in S \) \([a \leq t \land A(a)](p_i/e_i) \),
6. \([A \circ B] \) is \([A] \circ [B] \) for \( \circ = \land, \lor, \neg \), \( \iff \); \( \neg [A] \) is \( \neg [A] \);
7. \([A \circ B] \) is \([A] \circ [B] \) for \( \circ = \land, \lor, \neg \), \( \iff \); \( \neg [A] \) is \( \neg [A] \);
8. for \( A(a) \in \Sigma^b_1 \), \( t \) a term, \( a \) a free variable, \( q(x) \) a bounding polynomial of \( A(t) \),

\[ S^1_{\alpha} \vdash \forall y \langle G_1 \vdash [t = a \land A(a)]_{m,m} \rightarrow [A(t)]_{m,m} \rangle. \]
It suffices to construct \([A]_{q(m)}^n\) for \(A\) atomic, since conditions (3)–(7) determine the construction for other bounded formulas. The translation of atomic formulas
\[
l(a_1, \ldots, a_k) = s(a_1, \ldots, a_k)
\]
and
\[
l(a_1, \ldots, a_k) \leq s(a_1, \ldots, a_k)
\]
is defined as follows: Associate with any variable \(a_i\) free atoms \(p_{a_i}^0, \ldots, p_{a_i}^{q(m)}\). As any function \(f\) in the language of \(S_2\) is polynomial time computable, there are Boolean circuits \(C_i^m\) of the size polynomial in \(m, q(m)\) computing \(f\) on inputs of the length \(\leq m\). Combining these circuits one can construct circuits \(C_i^m\) computing any term \(t\) and having again size polynomial in \(m, q(m)\). Circuit \(C_i^m\) has some dummy input nodes \(p_{a_i}^j\), for \(j > m\), and may have also some dummy output nodes \(q_j\) if \(q(m)\) is larger than the length of the output. We assume that these nodes are labelled e.g. by \(0\). Boolean circuit \(C_i^m\) can be easily turned to a \(\Sigma_i^p\)-proposition \(B_i^m(p_{a_1}, \ldots, p_{a_k}, q)\). So for \(n_1, \ldots, n_k\) with the length \(\leq m\) we have:
\[
l(n_1, \ldots, n_k) = n\quad \text{iff} \quad B_i^m(n_1, \ldots, n_k, n)\quad \text{is true.}
\]
We shall occasionally say that the atoms \(q_j\)'s are associated with the term \(t\).

Define the translation of atomic formulas:

(a) \([t(a_1, \ldots, a_k) = s(a_1, \ldots, a_k)]_{q(m)}^n\) is
\[
\forall x_0 \ldots \exists x_{q(m)} (B_i^m(p_{a_1}, \ldots, q_j/x_j) \land B_i^m(p_{a_1}, \ldots, q_j/x_j)).
\]
This can be also written in a \(\Pi_i^0\)-form
\[
\forall x \forall y (B_i^m(p_{a_1}, \ldots, x) \land B_i^m(p_{a_1}, \ldots, y) \Rightarrow \bigwedge_{i=0}^{q(m)} x_i \equiv y_i).
\]
(b) \([t(a_1, \ldots, a_k) \leq s(a_1, \ldots, a_k)]_{q(m)}^n\) is
\[
\exists x \exists y (B_i^m(p_{a_1}, \ldots, x) \land B_i^m(p_{a_1}, \ldots, y) \land \bigwedge_{i=0}^{q(m)} \bigwedge_{j=i+1}^{q(m)} x_j \equiv y_j \Rightarrow (x_i \supset y_i)).
\]
Again this has a \(\Pi_i^0\)-form, too,
\[
\forall x \forall y (B_i^m(p_{a_1}, \ldots, x) \land B_i^m(p_{a_1}, \ldots, y) \Rightarrow \bigwedge_{i=0}^{q(m)} \bigwedge_{j=i+1}^{q(m)} x_j \equiv y_j \Rightarrow (x_i \supset y_i)).
\]
Now, having \(A \in \Sigma_i^b\) for \(i \geq 1\) choose such a form \((\Sigma_i^p \text{ or } \Pi_i^0)\) of the translations of the atomic subformulas of \(A\) so that \([A] \in \Sigma_i^p\).

(8) is proved easily by induction on the length of \(t\) and \(A\). \(\Box\)

Lemma 3.2. For \(A(a) \in \Sigma_i^b, i \geq 1, A(a)\) with one free variable \(a\), and \(q(x)\) a bounding polynomial of \(A\),
\[
S_2 \vdash \forall y (\text{Taut}_i([A]_{q(y)}^{|y|}) \Rightarrow \forall x (|x| \leq |y| \Rightarrow A(x))).
\]

Proof. We shall prove a stronger statement by induction on the length of \(A(a_1, \ldots, a_k) \in \Sigma_i^b\):

(*) \(S_2 \vdash \forall y \forall x_1 \ldots x_k (|x_1| \leq |y| \land \ldots \land |x_k| \leq |y| \land \tau(x_1, \ldots, x_k) \equiv A(x_1, \ldots, x_k))\),
where \(\tau(x_1, \ldots, x_k)\) is the substitution which substitutes \(x_j\) for the propositional variables corresponding to \(a_j, j = 1, \ldots, k\). For \(A\) atomic one can use \(\Sigma_i^b\)-PIND to prove the formula in \(S_2^i\), since \(\text{Sat}_i(A, \tau) \equiv \text{Sat}_0(A, \tau)\) by Lemma 1.2 and \(\text{Sat}_0\) is
Now, let \( A \) be a formula of \( \Sigma^b \) which are "i st
thus in
From (1)
As the above
We have shown to be able to extend
the arithmetization procedure

Lemma 3.4. (i) For \( A(a) \in \Sigma^b \), \( q(x) \) a bounding polynomial of \( A \),
\[ S^b_1 \vdash (G_i \vdash [A(a)]^{|y|}) \vdash \forall x(|x| \leq |y| \Rightarrow A(x)). \]

The same holds for \( i = 0 \) with \( G_1 \) instead of \( G_0 \).

Proof. Part (i) is simple: Choose the witnesses of the \( \exists \)-quantifiers of \( A(a) \) and using their digits compute the truth value of \([A(a)]^{|y|} \).

(ii) We shall prove the statement for \( i = 0 \). The case \( i \geq 1 \) is essentially the same.
\( \text{Taut}_0(A) \) is defined as
\[ \forall \tau(|\tau| \leq |A| \Rightarrow \text{Sat}_0(A, \tau)), \]
where we have to take \( \text{Sat}_0 \) in \( \Pi^b_1 \)-form. \( \text{Sat}_0(A, \tau) \) is defined by
\[ \forall w ("w is a computation of the value of \( A \) on \( \tau \)" \Rightarrow "the last bit of \( w \) is 1"). \]
Thus the translation of \( \text{Taut}_0(A) \) in the propositional calculus has the following form
\[ \forall p \forall q \text{Comp}_A(p, q) \Rightarrow q, \]
where \( p \) is a vector of atoms associated with \( \tau \), \( q \) is associated with \( w \), \( q \) is the last element of \( q \), and \( \text{Comp}_A \) is the translation of "\( w \) is a computation of the value of \( A \) on \( \tau \)". We shall assume that \( p \) are just the atoms of \( A \). In \( q \) certain atoms code the truth value of subformulas of \( A \) computed on \( p \). If the variables in \( q \) are suitably ordered, it is possible to prove (using \( \text{PIND} \) of \( S_1^b \)) that
\[ G_i \vdash \text{Comp}_A(p, q) \Rightarrow (q_i \equiv A_i), \]
where \( q_i \) corresponds to a subformula \( A_i \) of \( A \). In particular, we have
\[ G_i \vdash \text{Comp}_A(p, q) \Rightarrow (q_i \equiv A_i). \]
Now, let $\text{Comp}_A^i(p, q_1, \ldots, q_i)$ be subformulas of $\text{Comp}_A(p, q)$ which express that there are \textquotedblleft $i$ steps of the computation\textquotedblright. Again by PIND on $i$ one can show

$$G_1 \vdash \exists q_1, \ldots, \exists q_i \text{Comp}_A^i(p, q_1, \ldots, q_i),$$

thus in particular

(2) \hspace{1cm} G_1 \vdash \exists q \text{Comp}_A(p, q).

From (1) and (2) we obtain easily

$$G_1 \vdash \forall \rho \forall q (\text{Comp}_A(p, q) \supset q_r) \supset A.$$

As the above proof can be done in $S^2_1$, we have proved (ii). $\square$

We have not quite specified the translation $[A]$ for atomic formulas. If we want to be able to prove relation between weak fragments of arithmetic and weak propositional proof systems, we have to choose \textquoteleft natural\textquoteright Boolean circuits computing the arithmetical functions in the atomic formulas. Again, we state our last condition as a lemma.

**Lemma 3.5.** For $A$ any axiom of BASIC and $q(x)$ any bounding polynomial of $A$,

$$S^2_1 \vdash \forall y (G_1 \vdash [A]^{[y]}_{(p(y))}).$$

**Proof.** We shall use the construction of Cook [4]. He introduced an equational theory PV which has a function symbol for each polynomial time computable function. He defined translations of equations of PV into the propositional calculus such that the translations of equalities provable in PV have proofs of polynomial length in ER. Dowd [7] proved this simulation using EF instead of ER. The simulation can be extended to the theory PV 1 which is an extension of PV to open formulas. The proof actually gives an explicit polynomial time algorithm which, for given $m$, constructs an EF proof of $[A]^m$ and, moreover, this can be formalized in $S^2_1$. Buss [1] has shown a close relation of PV and PV 1 to $S^2_1$; in particular, if we translate formulas containing $\leq$ using Cook\textquoteleft s function LESS, all open theorems of $S^2_1$ become provable in PV 1. Thus we define our translation into quantified propositional calculus by taking Cook\textquoteleft s one for equations in the language of $S_2$ and by adding quantifiers to it as described above. Now the translation of atomic formulas will be different from the one described above, since we shall use equations $\text{LESS}(t, s) = 0$ instead of $t \leq s$. But one can show in $S^2_1$ that they are equivalent (and, moreover, it is irrelevant for this paper). Thus we obtain the condition of Lemma 3.5. $\square$

§ 4. Relations between propositional proof systems and theories

This section develops a general connection between propositional proof systems and theories. We tacitly assume that the languages of theories discussed contain the language of $S^2_1$.

We shall write $\forall \Sigma_i^b(T)$ for the set of all $\forall \Sigma_i^b$-consequences of $T$.

**Definition.** For $i \geq 0$, $P$ a proof system and $T$ a theory, $P$ simulates $\forall \Sigma_i^b(T)$ iff for any $\forall x A(x) \in \forall \Sigma_i^b(T)$ there is a bounding polynomial $p(x)$ of $A$ such that

$$S^2_1 \vdash \forall y (P \vdash [A]^{[y]}_{(p(y))}).$$
Definition. For \( i \geq 0 \) a proof system \( P \) is \( i \)-regular iff \( S_1^i \) proves

(i) \( P \models \Sigma^b_1 G_1 \),
(ii) \( P \vdash A \supset B \land P \vdash A \supset P \vdash B \),
(iii) for \( A \in \Sigma^b_3 \), \( |y| \geq |A| \)
\[
P \vdash [\text{Taut}_i(A)]^{|y|} \supset P \vdash A,
\]
where \( g(x) \) is a bounding polynomial of \( \text{Taut}_i \). Observe that an \( i \)-regular proof system satisfies Lemmas 3.2, 3.4 and 3.5. This is the motivation for their definition.

Theorem 4.1. Let \( T \supseteq S_1^i \) and \( P \) be an \( i \)-regular proof system.

(i) Suppose \( i \geq 2 \), \( P \) simulates \( \forall \Sigma^b_1(T) \) and \( T \models i\cdot\text{RFN}(P) \). Then
\[
\forall \Sigma^b_1(T) \equiv (S_1^i + i\cdot\text{RFN}(P)),
\]
thus \( \forall \Sigma^b_1(T) \) is finitely axiomatizable.

(ii) Suppose \( i \geq 0 \), \( P \) simulates \( \forall \Sigma^b_1(T) \) and \( T \models i\cdot\text{RFN}(Q) \) for some propositional proof system \( Q \). Then
\[
S_1^i \vdash P \supseteq Q.
\]
(iii) Suppose \( i \geq 0 \), \( P \) simulates \( \forall \Sigma^b_1(T) \) and \( T \vdash \text{NP} = \text{coNP} \). Then there exists a polynomial \( p(x) \) such that \( T \) proves
\[
(\forall A \in \text{TAUT}_i) \exists d: P \vdash A \land |d| \leq p(|A|).
\]

Statement (ii) generalizes a construction of Cook [3] using which he showed (ii) for \( P = \text{ER}, T = \text{PV} \) and \( j = 0 \). Statement (iii) could be used to generalize a result of Wilkie [11] who proved (iii) for \( T = S_1^i \) and \( P = \text{SF} \).

Proof. (i) \( S_1^i \) is \( \forall \Sigma^b_1 \) and so \( S_1^i \subseteq \forall \Sigma^b_1(T) \) for \( i \geq 2 \). By Lemma 1.3, \( i\cdot\text{RFN}(P) \in \forall \Sigma^b_1(T) \). On the other hand, assume \( \forall x A(x) \in \forall \Sigma^b_1(T) \). Then \( S_1^i \vdash \forall y (P \vdash [A]^{|y|}) \), for some bounding polynomial. By Lemma 3.3 then
\[
S_1^i + i\cdot\text{RFN}(P) \vdash \forall y \forall x (|x| \leq |y| \supset A(x)),
\]
i.e. \( S_1^i + i\cdot\text{RFN}(P) \vdash \forall x A(x) \).

(ii) Assume \( T \models i\cdot\text{RFN}(Q) \), so
\[
(1) S_1^i \vdash (P \vdash [d: Q \vdash A \land A \in \Sigma^b_1 \supset \text{Taut}_i(A)])^{d+|A|}.
\]
By Lemma 3.4 (i), as \( d: Q \vdash A \) and \( A \in \Sigma^b_1 \) are \( \Sigma^b_1 \)-formulas and since \( P \) is \( i \)-regular we have
\[
(2) S_1^i \vdash d: Q \vdash A \land A \in \Sigma^b_1 \supset P \vdash [\text{Taut}_i(A)]^{d+|A|}.
\]
Since \( P \) is \( i \)-regular we can use Lemma 3.4 (ii) to deduce
\[
(3) S_1^i \vdash d: Q \vdash A \land A \in \Sigma^b_1 \supset P \vdash A.
\]
By the main theorem of Buss [1] there is a polynomial time function \( f \) such that
\[
S_1^i \vdash d: Q \vdash A \land A \in \Sigma^b_1 \supset f(d, A): P \vdash A.
\]
(iii) Assume \( T \vdash \text{NP} = \text{coNP} \). Then every bounded formula is equivalent to a \( \Sigma^b_1 \) formula, thus
\[
(1) T \vdash \text{Taut}_i(A) \equiv (\exists x \leq t(A)) B(x, a),
\]

From (1)

(2) \( T \vdash \text{Taut}_i(A) \)

Using this

(3) \( S_1^i \vdash \text{Taut}_i(A) \)

Hence

Next we consider theories \( T \vdash i\cdot\text{RFN}(P) \) and \( T \vdash i\cdot\text{RFN}(Q) \) for propositional theories \( T \) and \( Q \) that we will define next.

Corollary 4.2. Let \( T \vdash i\cdot\text{RFN}(P) \) and \( T \vdash i\cdot\text{RFN}(Q) \) for some propositional theories \( T \) and \( Q \) that we will define next.

(i) \( S_1^i \vdash P \supset Q \)

(ii) \( S_1^i \vdash Q \supset P \)

(iii) \( S_1^i \vdash P = Q \)

(iv) \( S_1^i \vdash P = Q \)

Then \( T \vdash P = Q \).

Proof. By Theorem 4.1.

On the other hand

Corollary 4.3. Let \( T \vdash \text{NP} = \text{coNP} \). Then

(i) \( T \vdash \text{NP} = \text{coNP} \)

(ii) \( T \vdash \text{NP} = \text{coNP} \)

Proof. By Corollary 4.2.

In the next section we will study \( i \)-regular proof systems for \( P \) a propositional theory.

Theorem 4.4. Let \( \langle \Delta \rangle \supseteq S_1^i \).

Proof. By Theorem 4.1.

By the main theorem of Buss [1] there is a polynomial time function \( f \) such that
\[
S_1^i \vdash d: Q \vdash A \land A \in \Sigma^b_1 \supset f(d, A): P \vdash A.
\]

(iii) Assume \( T \vdash \text{NP} = \text{coNP} \). Then every bounded formula is equivalent to a \( \Sigma^b_1 \) formula, thus
\[
(1) T \vdash \text{Taut}_i(A) \equiv (\exists x \leq t(A)) B(x, a),
\]

From (1)
for some $\mathcal{A}^1$-formula $B$. Define the proof system $Q$ by
\[ d : Q \vdash A \iff d \leq t(A) \land B(d, A). \]
From (1) then
\[ (2) T \vdash \iota \text{-} RFN(Q). \]
Using the statement (ii) then
\[ (3) S^1_i \vdash P \geq i \cdot Q. \]
Hence
\[ T \vdash \text{Taut}_i(A) \equiv \exists d(|d| \leq p(|A|) \land d : P \vdash A), \]
where $p(x)$ is the polynomial given by the function $f$ of (ii). \qed

Next corollary shows that, in principle, Theorem 4.1 can be used to show that two theories are different.

Corollary 4.2. Assume that for $i \geq 0$, theories $T \equiv S^1_i$ and $S$, and proof systems $P$ and $Q$ the following holds:

(i) $P$ is $i$-regular,
(ii) $P$ simulates $V\Sigma^b_i(T)$,
(iii) $S \vdash \iota \text{-} RFN(Q)$,
(iv) not $P \geq i \cdot Q$.

Then $T \not\vdash S$, in particular $T \not\vdash \iota \text{-} RFN(Q)$.

Proof. Use Theorem 4.1 (ii). \qed

On the other hand, we have the following corollary:

Corollary 4.3. Assume $S^1_i \equiv S \equiv T$, $i \geq 1$, $P$ is $i$-regular, $P$ simulates $V\Sigma^b_i(T)$ and $T \vdash \iota \text{-} RFN(P)$. Then the following statements are equivalent:

(i) $T$ is $V\Sigma^b_i$-conservative over $S$,
(ii) $S \vdash \iota \text{-} RFN(P)$.

Proof. For (i) \Rightarrow (ii) use Lemma 1.3. The other implication is proved as Theorem 4.1 (i). \qed

In the following sections we shall apply the general theorems of this section to the proof systems $G_i$ and theories $S^1_i$ and $T^1_i$.

§ 5. Provability of reflection principles

By the definition of proof systems in § 1, any formula $\iota \text{-} RFN(P)$ is true. In this section we are interested in the question which theory suffices to prove $\iota \text{-} RFN(P)$, for $P$ a calculus of § 2.

Theorem 5.1. For $i \geq 0$, $S^{i+1}_2 \vdash \iota \text{-} RFN(G_i)$.

Proof. A sequent $I \rightarrow A$ is satisfied by a truth valuation $\tau$ iff the formula $(\land I) \supset (\lor A)$ is satisfied by $\tau$. Analogically with Lemma 1.1, there are formulas $SSat_i(Z, \tau)$ and $STaut_i(Z, \tau)$ formalizing "sequent $Z$ consisting only of $\Sigma^b_i \cup \Pi^b_i$-propositions is satisfied by truth valuation $\tau" and "sequent $Z$ consisting only of $\Sigma^b_i \cup \Pi^b_i$-
Definition. For \( i \geq 0 \) a proof system \( P \) is \( i \)-regular iff \( S_i \) proves

1. \( P \vdash G_1 \),
2. \( P \vdash A \Rightarrow B \land P \vdash A \Rightarrow P \vdash B \),
3. for \( A \in \Sigma_i^0, |y| \geq |A| \\
\quad P \vdash [\text{Taut}_i(A)]^{|d|} \Rightarrow P \vdash A \)

where \( q(x) \) is a bounding polynomial of \( \text{Taut}_i \). Observe that an \( i \)-regular proof system satisfies Lemmas 3.2, 3.4 and 3.5. This is the motivation for their definition.

Theorem 4.1. Let \( T \supseteq S_i \) and \( P \) be an \( i \)-regular proof system.

(i) Suppose \( i \geq 2 \), \( P \) simulates \( \forall \Sigma_i^b(T) \) and \( T \vdash i\text{-RFN}(P) \). Then
\[
\forall \Sigma_i^b(T) \equiv (S_i + i\text{-RFN}(P)),
\]
thus \( \forall \Sigma_i^b(T) \) is finitely axiomatizable.

(ii) Suppose \( i \geq 0 \), \( P \) simulates \( \forall \Sigma_i^b(T) \) and \( T \vdash i\text{-RFN}(Q) \) for some propositional proof system \( Q \). Then
\[
S_i + P \vdash Q.
\]

(iii) Suppose \( i \geq 0 \), \( P \) simulates \( \forall \Sigma_i^b(T) \) and \( T \vdash \text{NP} = \text{coNP} \). Then there exists a polynomial \( p(x) \) such that \( T \) proves
\[
(\forall A \in \text{TAUT}_i) \exists d: P \vdash A \land |d| \leq p(|A|).
\]

Statement (ii) generalizes a construction of Cook [3] using which he showed (ii) for \( P = ER, T = PV \) and \( j = 0 \). Statement (iii) could be used to generalize a result of Wilkie [11] who proved (iii) for \( T = S_i \) and \( P = SF \).

Proof. (i) \( S_i \) is \( \forall \Sigma_i^b \) and so \( S_i \subseteq \forall \Sigma_i^b(T) \) for \( i \geq 2 \). By Lemma 1.3, \( i\text{-RFN}(P) \in \forall \Sigma_i^b(T) \). On the other hand, assume \( \forall x A(x) \in \forall \Sigma_i^b(T) \). Then \( S_i \vdash \forall y\forall x\forall |x| \leq |y| \Rightarrow A(x) \).

By Lemma 3.3 then
\[
S_i + i\text{-RFN}(P) \vdash \forall x A(x).
\]

(ii) Assume \( T \vdash i\text{-RFN}(Q) \), so
\[
S_i + (P \vdash [d: Q \vdash A \land A \in \Sigma_i^b \Rightarrow \text{Taut}_i(A)]^{|d|+|A|}).
\]
By Lemma 3.4 (i), as \( d: Q \vdash A \) and \( A \in \Sigma_i^b \) are \( \Sigma_i^b \)-formulas and since \( P \) is \( i \)-regular we have
\[
S_i \vdash d: Q \vdash A \land A \in \Sigma_i^b \Rightarrow P \vdash [\text{Taut}_i(A)]^{|d|+|A|}.
\]
Since \( P \) is \( i \)-regular we can use Lemma 3.4 (ii) to deduce
\[
S_i \vdash d: Q \vdash A \land A \in \Sigma_i^b \Rightarrow P \vdash A.
\]
By the main theorem of Buss [1] there is a polynomial time function \( f \) such that,
\[
S_i \vdash d: Q \vdash A \land A \in \Sigma_i^b \Rightarrow f(d, A): P \vdash A.
\]

(iii) Assume \( T \vdash \text{NP} = \text{coNP} \). Then every bounded formula is equivalent to a \( \Sigma_i^1 \) formula, thus
\[
T \vdash \text{Taut}_i(A) \equiv (\exists x \leq t(A)) B(x, a),
\]
Using this, we can prove the following:

(3) \( S_i \vdash \text{Sat}(Z) \Rightarrow \text{SSat}(Z) \).

Hence
\[
\forall \Sigma_i^b(T) \equiv (S_i + i\text{-RFN}(P)),
\]

where \( p(x) \) is a bounding polynomial of \( \text{Taut}_i \).
propositions is satisfied by any truth valuation”. Also it is evident that $SSat_t \in A_{i+1}^b$ and $STaut_t \in \Pi_{i+1}^b$.

Fix $i \geq 1$. Let $A(d)$ be the formula

$$(\forall Z \leq d) (d:G_i \vdash Z \Rightarrow STaut_t(Z)).$$

Thus $A(d)$ is a $\Pi_{i+1}^b$-formula. We shall prove $A(d)$ by induction on the number of inferences in $d$, i.e. using $\Pi_{i+1}^b$-PIND. As $A(0)$ is trivially true we need only to establish

$S_i^{i+1} \vdash A([d/2]) \Rightarrow A(d).$

This is proved by checking that any rule of $G_t$ is semantically correct, i.e. that it infers a tautological sequent from tautological premises. By Lemma 1.2 this is easily checked. (Note that it is also not hard to show the semantical correctness of the substitution rule, cf. [9].) \[\Box\]

Corollary 5.2. For $i \geq 1$, $T_i \vdash i-RFN(G_t)$.

Proof. By Lemma 1.3, $i-RFN(G_t)$ is an $\forall \Sigma_i^b$-sentence. By Buss [2], $\forall \Sigma_i^b(S_i^{i+1}) = \forall \Sigma_i^b(T_i)$. Use Theorem 5.1. \[\Box\]

§ 6. Simulation of arithmetical proofs by propositional calculi

Theorem 6.1. For $i \geq 1$, $G_t$ simulates $\forall \Sigma_i^b(T_i^i)$.\[\Box\]

Proof. Assume $d:T_i^i \vdash A(a)$, where $A \in \Sigma_i^b$. By cut-elimination for $T_i^i$ (cf. Buss [1, Chapter 4]) we may assume that all formulas in $d$ are in $\Sigma_i^b \cup \Pi_i^b$. Choose a polynomial $q(x)$ which is a bounding polynomial of all formulas occurring in $d$. The idea of the simulation of $d$ is to replace any formula $B$ in $d$ by its translation $[B]_{q(m)}$ and to fill some parts in the resulted “preproof” to obtain a $G_t$-proof of $[A]_{q(m)}$.

To show that this can be done we shall proceed by induction on the number of inferences in $d$. Consider several cases according to the type of the last inference in $d$. We shall write $[\ ]$ instead of $[\ ]_{q(m)}$ and $[\ ]$ instead of $[A_1], \ldots, [A_k]$ for a cedent $\Gamma = A_1, \ldots, A_k$.

(a) $d$ is an initial sequent, i.e. a logical axiom, an equality axiom or an instance of an axiom of BASIC. The translations of the first two cases are easily proved in $G_t$. The last case is assured by Lemma 3.5.

(b) The inference is a structural rule, cut-rule or a propositional rule: These cases are handled by the corresponding rules of $G_t$.

(c) ($\forall$-right)

\[
\frac{a \leq s, \Gamma \vdash A, B(a)}{\Gamma \vdash A, (\forall x \leq s) B(x)}.
\]

Consider two subcases: (c1) $s$ is not of the form $|t|$, (c2) otherwise.

(c1) By ($\supset$ : right) derive

$[\Gamma] \vdash [A], [a \leq s \supset B(a)]$

and using $q(m) + 1$ applications of ($\forall$ : right) to the free atoms associated with $a$ derive

$[\Gamma] \vdash [A], [(\forall x \leq s) B(x)].$
(c2) First derive
\[ Z_1: \bigvee_{i \in S} [a = \bar{e} \land \bar{e} \leq |t|] \rightarrow [a \leq |t|], \]
where \( S = \{(e_0, \ldots, e_{q(m)}) \in \{0, 1\}^{q(m)+1} \mid (\forall i > |m|) \, e_i = 0\}, \)
and
\[ Z_2: [B(a)] \rightarrow [B(a)]. \]
By successively applying \((\therefore: \text{left})\) and \((\therefore: \text{right})\) to \(Z_1, Z_2\) get
\[ Z_3: [a \leq |t| \Rightarrow B(a)] \rightarrow \bigvee_{i \in S} [a = \bar{e} \land \bar{e} \leq |t|] \Rightarrow B(a). \]
Derive
\[ Z_4: \bigvee_{i \in S} [a = \bar{e} \land \bar{e} \leq |t|] \Rightarrow \bigvee_{i \in S} [B(a)] \rightarrow \bigwedge_{i \in S} [a = \bar{e} \land \bar{e} \leq |t|] \Rightarrow B(a). \]
Applying cut-rule to \(Z_3, Z_4\) we obtain
\[ Z_5: [a \leq |t| \Rightarrow B(a)] \rightarrow \bigwedge_{i \in S} [a = \bar{e} \land \bar{e} \leq |t|] \Rightarrow B(a). \]
Now derive
\[ Z_6: \bigwedge_{i \in S} [a = \bar{e} \land \bar{e} \leq |t|] \Rightarrow B(a)] \rightarrow \bigwedge_{i \in S} [a \leq |t| \Rightarrow B(a)] (p/\bar{e}), \]
and by cut from \(Z_5, Z_6\)
\[ Z_7: [a \leq |t| \Rightarrow B(a)] (p) \rightarrow \bigwedge_{i \in S} [a \leq |t| \Rightarrow B(a)] (p/\bar{e}). \]
Now use cut-rule to \(Z_7\) and to the first sequent derived in the case (c1) to obtain
\[ [t] \rightarrow [A], \bigwedge_{i \in S} [a \leq |t| \Rightarrow B(a)] (p/\bar{e}). \]
(d) \((\forall: \text{left})\)
\[ \frac{B(t), \Gamma \rightarrow A}{t \leq s, (\forall x \leq s) B(x), \Gamma \rightarrow A}. \]
Again consider two cases: (d1) \(s\) is not of the form \(|r|\), (d2) otherwise. In both cases we first derive
\[ Z_0: [t \leq s], [(\forall x \leq s) B(x)] \rightarrow [B(t)] \]
and apply cut-rule to this sequent and to
\[ [B(t)], [t] \rightarrow [A] \]
to get the wanted sequent
\[ [t \leq s], [(\forall x \leq s) B(x)], [t] \rightarrow [A]. \]
(d1) First derive
\[ Z_1: [t \leq s] \rightarrow \exists x[a \leq s \land a = t] (p/x) \]
and
\[ Z_2: [(\forall x \leq s) B(x)], \exists x[a \leq s \land a = t] (p/x) \rightarrow \exists x[a = t \land B(a)] (p/x) \]
By cut-rule from \(Z_1, Z_2\) it follows
\[ Z_3: [t \leq s], [(\forall x \leq s) B(x)] \rightarrow \exists x[a = t \land B(a)] (p/x). \]
We shall use the following
Claim. If \(C \in \Sigma^b_t \cup \Pi^b_t\), then for an appropriate bounding polynomial
\[ G_t \vdash [t = a \land C(a)] \rightarrow [C(t)]. \]
For the proof of the Claim take the open matrix of $C(a)$ and apply Lemma 3 to it. The sequent above is easily got from this sequent in $G_1$.

Using Claim derive

$$Z_4: \exists x[a = t \land B(a)] (p/x) \rightarrow [B(t)]$$

and by cut from $Z_3$, $Z_4$ derive $Z_0$.

(d2) First derive

$$Z_1: \left[ t \leq |r| \right] \rightarrow \bigvee_{s \in S} \left[ t = a \land a \leq |r| \right] (p/\bar{s})$$

where the set $S$ is the same as in (c2). Then derive

$$Z_2: \left[ (\forall x \leq |r|) B(x) \right], \bigvee_{s \in S} \left[ t = a \land a \leq |r| \right] (p/\bar{s}) \rightarrow \bigvee_{s \in S} \left[ t = a \land B(a) \right] (p/\bar{s})$$

Using cut-rule obtain from $Z_1$, $Z_2$

$$Z_3: \left[ t \leq |r| \right], \left[ (\forall x \leq |r|) B(x) \right] \rightarrow \bigvee_{s \in S} \left[ t = a \land B(a) \right] (p/\bar{s})$$

Using Claim deriv

$$Z_4: \bigvee_{s \in S} \left[ t = a \land B(a) \right] (p/\bar{s}) \rightarrow [B(t)]$$

and by cut from $Z_3$, $Z_4$ derive $Z_0$.

(e) The $(\exists$ rules) are dual to the $(\forall$ rules) and are handled similarly.

(f) $\Sigma^p$-IND rule:

$$B(a) \rightarrow B(a+1)$$

$$B(0) \rightarrow B(t)$$

We omit the side formulas. Assume that we have derived

$$Z: [B(a)] \rightarrow [B(a+1)].$$

We assume that atoms $p$ are associated with $a$ and atoms $q$ with $t$. We cannot replace IND by cuts as there would be exponentially many of them in $m$. We shall shorten the simulation essentially using the substitution rule which is provably simulable in $G_1$ (Lemma 2.2).

(1) We shall first derive sequents

$$W_0: [B(a)] \rightarrow [B(a+2^{a})],$$

$$W_{q(m)}: [B(a)] \rightarrow [B(a+2^{q(m)})].$$

$W_0$ is $Z$. $W_{i+1}$ is derived from $W_i$ as follows: Assume that atoms $p$ are associated to $a$ and new atoms $q$ will be associated to the new variable $b$. By substitution $p \mapsto q$ derive from $W_i$

$$W'_i: [B(a)] (p/q) \rightarrow [B(a+2^i)] (p/q).$$

Using (the translation of) equality axioms derive

$$W'_2: [a + 2^i = b] (p, q), [B(a+2^i)] (p) \rightarrow [B(a)] (p/q).$$

Apply cut to $W_i$ and $W_2'$ to get

$$W'_3: [a + 2^i = b] (p, q), [B(a)] (p) \rightarrow [B(a)] (p/q).$$
Apply cut to $W_1'$ and $W_3'$ to get

$$W_4': \left[ a + 2^l = b \right] (p, q), \left[ B(a) \right] (p) \rightarrow \left[ B(a + 2^l) \right] (p/q).$$

Using (the translation of) equality axioms derive

$$W_5': \left[ a + 2^l = b \right] (p, q), \left[ B(a + 2^l) \right] (p/q) \rightarrow \left[ B(a + 2^{l+1}) \right] (p).$$

Apply cut to $W_4'$ and $W_5'$ to get

$$W_6': \left[ a + 2^l = b \right] (p, q), \left[ B(a) \right] (p) \rightarrow \left[ B(a + 2^{l+1}) \right] (p).$$

To $W_6'$ apply $(q(m) + 1)$-times (3: left) with eigenvariables $q$ to get

$$W_7': \exists x[ a + 2^l = b ] (p, x), \left[ B(a) \right] (p) \rightarrow \left[ B(a + 2^{l+1}) \right] (p).$$

Derive

$$W_8': \rightarrow \exists x[ a + 2^l = b ] (p, x)$$

and apply cut to $W_7'$ and $W_8'$ to get $W_{i+1}$.

(2) Now we shall derive sequents

$$Z_0': \left[ 2^{2^0} \geq b \right] (q), \left[ B(a) \right] (p) \rightarrow \left[ B(a + b) \right] (p, q).$$

$$Z_{q(m)}': \left[ 2^{2^0} \geq b \right] (q), \left[ B(a) \right] (p) \rightarrow \left[ B(a + b) \right] (p, q).$$

Now $Z_0'$ simply follows from $W_0$ using

$$\left[ 2^{2^0} \geq b \right] (q) \rightarrow \left[ a = b v a + 1 = b \right] (p, q).$$

$Z_{i+1}$ is derived as follows: Take new variables $c, d$ and associate with them atoms $r, s$. By substitution $p \rightarrow s, q \rightarrow r$ derive from $Z_i$

$$Z_i': \left[ 2^{l} \geq c \right] (r), \left[ B(d) \right] (s) \rightarrow \left[ B(d + c) \right] (s, r).$$

Derive from $W_i$

$$Z_2': \left[ B(a) \right] (p), \left[ a + 2^l = d \right] (p, s) \rightarrow \left[ B(d) \right] (s).$$

Apply cut to $Z_1', Z_2'$ to get

$$Z_3': \left[ 2^{l} \geq c \right] (r), \left[ a + 2^l = d \right] (p, s), \left[ B(a) \right] (p) \rightarrow \left[ B(d + c) \right] (s, r).$$

From $Z_3'$ derive

$$Z_4': \left[ 2^{l} \geq c \right] (r), \left[ b = 2^l + c \right] (q, r), \left[ B(a) \right] (p) \rightarrow \left[ B(a + b) \right] (p, q).$$

Apply to $Z_i$ and $Z_4'$ ($\lor$: left) to get

$$Z_5': \left[ 2^{l} \geq b \lor (2^{l} \geq c \land b = 2^{l} + c) \right] (q, r), \left[ B(a) \right] (p) \rightarrow \left[ B(a + b) \right] (p, q).$$

Using (3: left) applied to eigenatoms $r$ we get

$$Z_6': \exists x[ 2^{l} \geq b \lor (2^{l} \geq c \land b = 2^{l} + c) ] (q, r/x), \left[ B(a) \right] (p) \rightarrow \left[ B(a + b) \right] (p, q).$$

Derive

$$Z_7': \left[ 2^{l+1} \geq b \right] (q) \rightarrow \exists x[ 2^{l} \geq b \lor (2^{l} \geq c \land b = 2^{l} + c) ] (q, r/x).$$

Finally apply cut to $Z_6'$ and $Z_7'$ to get $Z_{i+1}$.

(3) Now we substitute to $Z_{q(m)}$ $p \rightarrow 0, q \rightarrow p'$, where $p'$ are atoms associated to get

$$\left[ 2^{2^0} \geq \ell \right] (p'), \left[ B(0) \right] \rightarrow \left[ B(\ell) \right] (p').$$

Also is simply derived

$$\rightarrow \left[ 2^{2^0} \geq \ell \right] (p').$$
Apply cut to these two sequents to get

\[ [B(0)] \rightarrow [B(\theta)]. \]

This completes the proof. □

**Corollary 6.2.** For \( i \geq 1 \), \( G_i \) simulates \( \forall \Sigma^b_i(S^{i+1}_2) \).

**Proof.** By Buss [2], \( \forall \Sigma^b_i(T^{i}_2) = \forall \Sigma^b_i(S^{i+1}_2) \), for \( i \geq 1 \). Use Theorem 6.1. □

**Corollary 6.3.** For \( i \geq j \geq 1 \),

(i) \( G_i \) simulates \( \forall \Sigma^b_i(S^{i+1}_2) \) and \( \forall \Sigma^b_i(T^{i}_2) \),

(ii) \( G_s \) simulates \( \forall \Sigma^b_i(S_2) \). □

Consider the simulation of \( \forall \Sigma^b_0 \) statements. We can choose a translation of the atomic subformulas of a \( \Sigma^b_0 \)-formula \( A \) such that \( \llbracket A \rrbracket \) is \( \Pi^b_2 \). Denote by \( \llbracket A \rrbracket^* \) the proposition arising from \( \llbracket A \rrbracket \) after omitting all quantifiers. So \( \llbracket A \rrbracket^* \in \Sigma^b_0 \) and \( \llbracket A \rrbracket^* \) may have other free atoms than those associated to some free variable of \( A \). Then it holds:

For \( m \geq 1 \), \( ... \), \( n_m \leq m \), \( A(a_1/n_1) \) is true iff \( \llbracket A \rrbracket^*(p_1/n_1) \) is tautological.

This is the translation (of \( \Pi^b_2 \)-formulas, actually) used in Krajíček-Pudlák [9]. There it is proved, using the results of Cook [4] and Buss [1], that \( S^0 \) simulates \( \forall \Sigma^b_0(S^1_2) \) if the translation \( \llbracket \rrbracket^* \) is used.

Observe in the next section that if we used the translation \( \llbracket \rrbracket^* \), the theorems would extend to the case \( i = 0 \) too with \( S^0 \) instead of \( S_0 \).

§ 7. Consequences for fragments \( S^i_2, T^i_2 \) and for \( S^i_2 \)

Now we shall explicitly state the consequences following from the results of § 5 and § 6 for \( S^i_2 \) and \( T^i_2 \).

**Corollary 7.1.** For \( i \geq j \geq 2 \), \( \forall \Sigma^b_j(S^{i+1}_2) = \forall \Sigma^b_j(T^{i}_2) \) is finitely axiomatized by \( S^i_2 + j-RFN(G_i) \).

**Proof.** Use Theorems 4.1 (i) 5.1, 5.2 and 6.3. □

**Corollary 7.2.** For \( i \geq j \geq 0 \), \( i \geq 1 \), if \( S^{i+1}_2 + j-RFN(P) \) for some proof system \( P \), then \( S^i_2 + G_i \models P \). The same holds for \( i = 0 \) and \( S^0_2 \) instead of \( S^0_0 \).

**Proof.** Use Theorems 4.1 (ii), 6.1 for the case \( i \geq 1 \). The case \( i = 0 \) follows from the results of Cook [4] and Buss [1], cf. Krajíček-Pudlák [9]. □

**Corollary 7.3.** For \( i \geq 1 \), if \( S^{i+1}_2 \vdash NP = coNP \), then there is a polynomial \( p(x) \) such that

\[(\forall A \in TAUTo) \exists d(|d| \leq p(|A|) \land d : G_i \vdash A),\]

and \( S^{i+1}_2 \) proves \((*)\). The same holds for \( i = 0 \) with \( S^0_2 \) instead of \( S^0_0 \).

**Proof.** Use Theorems 4.1 (iii), 6.2 for the case \( i \geq 1 \). The case \( i = 0 \) was proved by Wilkie [11], however it can be proved in the same way as for \( i \geq 1 \), for details cf. Krajíček-Pudlák [9]. □

Some consequences mentioned above can be transferred to \( S^i_2 \).

**Corollary 7.4.**

(i) \( S^i_2 \) is axiomatized by \( S^i_2 + \{i-RFN(G_i) \mid i < \omega \} \).
(ii) If $S_2 \vdash \text{NP} = \text{coNP}$, then there is a polynomial $p(x)$ such that

\[(\forall A \in \text{TAUT}_0) \exists d(|d| \leq p(|A|) \land d : G \vdash A),\]

and $S_2$ proves $(*)$.

(iii) If $S_2 \vdash \text{0-RFN}(P)$, for some proof system $P$, then $S_{1}^{1} \vdash \text{G} \geq \text{P}^{0}$.

Proof. Part (i) is obvious from Corollary 7.1. Parts (ii) and (iii) are derived from Corollaries 7.2, 7.3 using a simple observation: $S_{1}^{1} \vdash \text{G} \geq \text{G}^{i}$, $i \geq 0$. \(\square\)

We shall sketch a nontrivial extension of the preceding results with interesting corollaries.

Let $A$ be a true $\forall \Sigma^b_i$-sentence, $i \geq 1$. Define $G_i^A$ to be the extension of $G_i$ where we add initial sequents of the form

\[\neg \left[ A \right]_{q(m)}^m\]

for $m = 1, 2, \ldots$ and $q$ a bounding polynomial.

Theorem 7.5. For $i \geq 1$ and $A$ a true $\forall \Sigma^b_i$-sentence

(i) $G_i^A$ is an $i$-regular proof system,

(ii) $S_{1}^{i+1} + A \vdash \text{i-RFN}(G_i^A)$,

(iii) $G_i^A$ simulates $\forall \Sigma^b_i(S_{1}^{i+1} + A)$.

Proof (sketch): (i) The only nontrivial condition of the definition of $i$-regular proof systems is the condition (iii). This is proved in the same way as Lemma 3.4 (ii).

(ii) The proof follows the proof of Theorem 5.1. We have only to check that it is provable in $S_{1}^{i+1} + A$ that initial sequents of $G_i^A$ are tautologies. This follows from Lemma 3.2.

(iii) Here we need a modification of the proof of Theorems 6.1 and 6.2. Again the only difference is in initial sequents and again we use Lemma 3.2. It is also easily seen that the equality $\forall \Sigma^b_{i+1}(T_2^i + A) = \forall \Sigma^b_{i+1}(S_{1}^{i+1} + A)$ can be obtained from the proof of Buss [2]. \(\square\)

Corollary 7.6. For $i \geq j \geq 2$ and $A$ a true $\forall \Sigma^b_i$-sentence,

\[\forall \Sigma^b_j(S_{1}^{i+1} + A) \equiv \forall \Sigma^b_j(T_2^i + A)\]

and both sets are finitely axiomatized by $S_{1}^i + j$-RFN($G_i^A$).

Corollary 7.7. Suppose propositions of $\text{TAUT}_1$ have proofs of polynomial length in $G_i$, $i > 1$. Then all propositions in $\text{TAUT}_1$ have proofs of polynomial length in $G_i$.

Proof. Assume $\text{TAUT}_1$ has polynomial proofs in $G_i$. Thus, in particular, the following formula, denoted by $A$, is true: $\text{Taut}_0(B) \equiv \exists d(|d| \leq p(|B|) \land d : G_i \vdash B)$, where $p$ is a suitable polynomial. As $S_{1}^{i+1} \vdash \text{i-RFN}(G_i)$ we have

\[S_{1}^{i+1} + A \vdash \text{Taut}_0(B) \equiv \exists d(|d| \leq p(|d|) \land d : G_i \vdash B),\]

thus $S_{1}^{i+1} + A$ proves $\text{NP} = \text{coNP}$. Hence by theorems 4.1 (iii) and 7.5 $\text{TAUT}_i$ has polynomial proofs in $G_i^A$. But the formulas $[A]_{q(m)}^m$ are in $\text{TAUT}_1$ (since $A \in \Sigma^b_i$), hence they have polynomial proofs in $G_i$. Thus $G_i$ polynomially simulates $G_i^A$ and the corollary follows. \(\square\)
§ 8. Open problems, conclusions

In previous sections we have left open several questions. In particular, we do not know whether $S^1_2 \vdash G_i \equiv G_{i+1}$, whether $S^1_2 \vdash i$-RFN($G_i$) or whether $G_{i-1}$ simulates $\forall \Sigma^b_1(T^i_2)$.

It follows from the next two theorems that these problems are important.

Theorem 8.1. For $i \geq 1$, the following statements are equivalent:

(i) $S^1_2 \vdash G_i \equiv G_{i+1}$,

(ii) $S^2_{i+1} \vdash i$-RFN($G_{i+1}$),

(iii) $G_i$ simulates $\forall \Sigma^b_1(T^i_2) = \forall \Sigma^b_1(S^2_{i+2})$,

(iv) $S^2_{i+2}$ is $\forall \Sigma^b_1$-conservative over $S^1_{i+1}$.

The same holds for $i = 0$ with $S^0_0$ instead of $G_0$.

Proof. (ii) $\Rightarrow$ (i): use Corollary 7.2. (i) $\Rightarrow$ (iii): use Theorem 6.1. (iii) $\Rightarrow$ (iv): use Theorem 5.1 and Lemma 3.3. (iv) $\Rightarrow$ (ii): use Lemma 1.3 and Theorem 5.1. □

Theorem 8.2. For $i \geq 0$, the following statements are equivalent:

(i) $S^1_{i+1} \vdash (i + 1)$-RFN($G_{i+1}$),

(ii) $S^2_{i+2}$ is $\forall \Sigma^b_{i+1}$-conservative over $S^1_{i+1}$.

Proof. (i) $\Rightarrow$ (ii): use Corollary 6.2. (ii) $\Rightarrow$ (i): use Lemma 1.3 and Theorem 5.1. □

References


Jan Krajíček and Pavel Pudlák

Mathematical Institute
Czechoslovak Academy of Sciences
Žitná 25
11567 Praha 1
CSSR

(Eingegangen am 28. November 1988)