Original In: STOC '85

The Polynomial Hierarchy and Fragments of Bounded Arithmetic

(Extended Abstract)

Samuel R. Buss
Department of Mathematics
Princeton University
February 1985

Introduction

One of the more important problems of computer science is to establish precise bounds on computational complexity and, in particular, to understand the relationship between P, NP, the polynomial hierarchy, PSPACE, EXPTIME, etc. This paper approaches these questions from the viewpoint of mathematical logic, in the hope that eventually the techniques of mathematical logic can shed light on the nature of computation.

We define below a set of formal theories of arithmetic called collectively Bounded Arithmetic. These formal theories are related to computational complexity in that, for each theory, the functions and predicates definable in a "nice" way in that theory have a certain computational complexity and conversely every function of that computation complexity is definable in that way. Thus, we present a theory S_2^1 which defines precisely the class of functions in P, another theory S_2^2 which defines precisely the class of functions which are polynomial time relative to NP, a theory U_2^1 which defines the PSPACE functions, a theory V_2^1 which defines the EXPTIME functions, and other theories corresponding to levels of the polynomial hierarchy.

We also discuss the properties of predicates which are definable in these theories. For instance, if S_2^1 proves that a predicate is in NP$\cap$co-NP , then that predicate is already in P.

The theories we discuss are first-order and second-order theories of arithmetic and are formulated in a manner analogous to Peano arithmetic. The weakest theory, S_2^1, which defines the polynomial time functions, is related to the equational system PV introduced by Cook [3]. The theories S_2^1 and PV have the same open PV-equations as theorems.

Finally we state some strong versions of the Gödel incompleteness theorems and we give a proof-theoretic principle which is equivalent to NP=co-NP.

Because of space considerations, no proofs are included in this abstract. A detailed discussion and full proofs will appear in [2].

The Polynomial Hierarchy

The Meyer-Stockmeyer polynomial hierarchy is a hierarchy of predicates with domain the natural numbers. We begin by repeating the usual definition of the polynomial hierarchy and in the next section we will state an alternative definition, which is more useful for our

purposes.

Definition: $|x|$ is the length of the binary representation of x, i.e., $|x|=\lceil \log_2(x+1)\rceil$. Note that $|0|$ is 0. If $\vec{x}$ is the vector $x_1, \ldots, x_n$, then $|\vec{x}|$ denotes the vector $|x_1|, \ldots, |x_n|$.

Definition: We say that a function f has polynomial growth rate iff there is a polynomial p such that for all $\vec{x}$, $|f(\vec{x})|\leqslant p(|\vec{x}|)$.

Definition: A predicate is a function with range $\{0,1\}$. The value 0 denotes "False" and 1 denotes "True."

Definition: Let X be a set of functions with polynomial growth rate. Then $PTC(X)$, the *polynomial-time closure* of X, is the set of functions computable by a Turing machine (i.e. a transducer) with some finite set of oracles $Q_1, \ldots, Q_k \in X$.

Definition: The polynomial time hierarchy consists of the following classes defined inductively:

$P=\Delta_1^p$ is the set of predicates on the natural numbers which are recognized by a polynomial time Turing machine.

$NP=\Sigma_1^p$ is the set of predicates on the natural numbers which are recognized by a nondeterministic Turing machine.

Σ_i^p is the set of predicates Q such that there is a $R\in\Delta_i^p$ and a polynomial q, so that for all $\vec{x}$

$$Q(\vec{x}) \iff (\exists y\leqslant 2^{q(|\vec{x}|)})R(\vec{x},y).$$

Π_i^p is the set of predicates Q such that there is a $R\in\Sigma_i^b$, so that for all $\vec{x}$

$$Q(\vec{x}) \iff \neg R(\vec{x}).$$

Δ_{i+1}^p is the set of predicates Q in $PTC(\Sigma_i^p)$.

The Link to Mathematical Logic

We now work in a first order language of **N** (the natural numbers). We will use the functions symbols 0, S, $+$, $\cdot$, $\#$, $|x|$, $\lfloor \frac{1}{2}x \rfloor$ and the predicate symbol $\leqslant$, where

$$\lfloor \tfrac{1}{2}x \rfloor = \text{the greatest integer} \leqslant \frac{x}{2}$$
$$|x| = \lceil \log_2(x+1) \rceil$$
$$x\#y = 2^{|x|\cdot|y|}$$

and the other symbols have their usual meanings. The inclusion of the function $\#$ (pronounced "smash", see Nelson [8] and Hook [6]) is very important, as it has the growth rate needed to define polynomial time functions. In particular, the $\#$ function allow us to express terms $2^{q(|\vec{x}|)}$ in the language of Bounded Arithmetic, where q is any polynomial with nonnegative coefficients.

Definition: A *bounded quantifier* is a quantifier of the form $(\forall x\leqslant t)$ or $(\exists x\leqslant t)$ where t is any term. A *sharply bounded quantifier* is a bounded quantifier of the form $(\forall x\leqslant |t|)$ or $(\exists x\leqslant |t|)$. $(\forall x)$ and $(\exists x)$ are *unbounded quantifiers*.

A *bounded formula* is a formula with no unbounded quantifiers.

We define a hierarchy Σ_i^b, Π_i^b of bounded formulae by counting alternations of quantifiers, ignoring the sharply bounded quantifiers.

Definition: Σ_i^b and Π_i^b are sets of bounded formulae defined inductively by:
(1) $\Sigma_0^b = \Pi_0^b$ is the set of formulae with all quantifiers sharply bounded.
(2) If $A \in \Sigma_i^b$ then $(\forall x \leqslant t)A$ is in Π_{i+1}^b and $(\forall x \leqslant |t|)A$ and $(\exists x \leqslant t)A$ are in Σ_i^b.
(3) If $A \in \Pi_i^b$ then $(\exists x \leqslant t)A$ is in Σ_{i+1}^b and $(\exists x \leqslant |t|)A$ and $(\forall x \leqslant t)A$ are in Π_i^b.
(4) The logical connectives $\wedge$, $\vee$, $\neg$ and $\supset$ are treated in the usual manner.

The next theorem can be thought of as an alternative definition for the polynomial time hierarchy. It states that a predicate belongs to certain level of the polynomial hierarchy iff it is expressible by a formula of Bounded Arithmetic of a certain complexity.

Theorem 1: Let $i \geqslant 1$. A predicate Q is in Σ_i^p iff there is a Σ_i^b–formula ϕ such that for all $\vec{n} \geqslant 0$,

$$Q(\vec{n}) \iff \mathbf{N} \models \phi(\vec{n})$$

This theorem is due to Stockmeyer [11], Wrathall [14], and Kent-Hodgson [7].

Definition: Let R be a theory of Bounded Arithmetic and A be a formula. Then A is Δ_i^b *with respect to* R iff R proves that A is equivalent to both a Σ_i^b– and a Π_i^b–formula.

Bounded Arithmetic

We next review the usual definition of Bounded Arithmetic. The most important axioms for Bounded Arithmetic are the induction axioms for bounded formulae.

Definition: The Σ_i^b–IND axioms are of the form

$$A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \supset (\forall x)A(x)$$

where A is any Σ_i^b–formula.

We define our first hierarchy of theories of Bounded Arithmetic by restricting the use of the induction axiom to a subset of the bounded formulae.

Definition: T_2^i is the first-order theory with language 0, S, $+$, $\cdot$, $\#$, $\lfloor \frac{1}{2}x \rfloor$, $|x|$, $\leqslant$ and with the following axioms:

(1) A finite set of open axioms defining simple properties of the function and relation symbols.
(2) The Σ_i^b–IND axioms.

$T_2^{(-1)}$ is the theory with only axioms (1). T_2 is the theory $\cup T_2^i$.

The theory T_2 is equivalent to the theory called elsewhere $I\Delta_0 + \Omega_1$ (see [13]). We shall be interested in subtheories of T_2; however, the subtheories T_2^i are not suitable for our purposes. Instead of the IND axioms, we need to use a modified version of the induction axioms,

called PIND. By using the PIND axioms we will be able to axiomatize subtheories of T_2 with desirable properties.

Definition: The Σ_i^b–$PIND$ axioms are of the form

$$A(0)\wedge(\forall x)(A(\lfloor \tfrac{1}{2}x \rfloor)\supset A(x))\supset(\forall x)A(x)$$

where A is any Σ_i^b–formula.

Definition: S_2^i is the first-order theory with language 0, S, $+$, $\cdot$, $\#$, $\lfloor \frac{1}{2}x \rfloor$, $|x|$, $\leqslant$ and with the following axioms:

(1) The finite set of open axioms of T_2^i, which define simple properties of the function and relation symbols.

(2) The Σ_i^b–$PIND$ axioms.

$S_2^{(-1)}$ is the theory with only axioms (1). S_2 is the theory $\cup S_2^i$.

It is not immediately obvious from the definitions that the theory S_2 is as strong as T_2; however, this is indeed the case. In fact we have:

Theorem 2: If $i\geqslant 1$, $S_2^i \Longrightarrow T_2^{i-1}$ and $T_2^i \Longrightarrow S_2^i$.

Corollary 3: $S_2 \equiv T_2$.

So the theories S_2^1, S_2^2, S_2^3, $\cdots$ do form a hierarchy of subtheories of T_2 and their union is all of T_2. These fragments of T_2 are the most natural and useful subtheories of Bounded Arithmetic for our purposes.

It is an open question whether the theories S_2^i are all distinct, or whether the hierarchy of theories collapses.

Other Axiomatizations of Bounded Arithmetic

There are a variety of other axiomatizations for Bounded Arithmetic. Among these are the following:

Definition: The Σ_i^b–$LIND$ axioms are:

$$A(0)\wedge(\forall x)(A(x)\supset A(Sx))\supset(\forall x)A(|x|)$$

where A is any Σ_i^b–formula. The Σ_i^b–MIN axioms are:

$$(\exists x)A(x)\supset(\exists x)(A(x)\wedge(\forall y<x)(\neg A(y)))$$

where A is any Σ_i^b–formula. The Σ_i^b–$LMIN$ axioms are:

$$(\exists x)A(x)\supset A(0)\vee(\exists x)(A(x)\wedge(\forall y\leqslant\lfloor \tfrac{1}{2}x \rfloor)(\neg A(y)))$$

where A is any Σ_i^b–formula. The Σ_i^b–replacement axioms are:

$$(\forall x\leqslant|t|)(\exists y\leqslant s)A(x,y)\leftrightarrow(\exists w\leqslant SqBd(t,s))(\forall x\leqslant|t|)(A(x,\beta(Sx,w))\wedge\beta(Sx,w)\leqslant s)$$

where A is any Σ_i^b–formula, β is the Gödel beta function and $SqBd$ is a term which depends on the precise definition of β.

We define the Π_i^b–IND, Π_i^b–$PIND$, Π_i^b–$LIND$, and Π_i^b–MIN axioms similarly.

Paris and Kirby [10] have carried out a detailed analysis of the comparative strengths of the various axiomatizations of Peano arithmetic. We present below the analogous results for Bounded Arithmetic. There are some differences between axiomatizations of Bounded Arithmetic and of Peano arithmetic. In particular, Σ_{i+1}^b–MIN is equivalent to Π_i^b–MIN in the presence of S_2^1, whereas, Paris and Kirby [10] show that Σ_i^0–MIN is equivalent to Π_i^0–MIN in the presence of P^-.

Theorem 4: In the presence of the theory S_2^1, the following implications hold: $(i \geq 0)$

(a)
$$\begin{array}{l} \Sigma_{i+1}^b\text{–}IND \iff \Pi_{i+1}^b\text{–}IND \iff \Sigma_{i+1}^b\text{–}MIN \iff \Pi_i^b\text{–}MIN \\ \Downarrow \\ \Sigma_{i+1}^b\text{–}PIND \iff \Pi_{i+1}^b\text{–}PIND \iff \Sigma_{i+1}^b\text{–}LMIN \iff \Sigma_{i+1}^p\text{–}LIND \iff \Pi_{i+1}^b\text{–}LIND \\ \Downarrow \\ \Sigma_i^b\text{–}IND \end{array}$$

(b) Σ_{i+1}^b–replacement $\Longrightarrow$ Σ_i^b–$PIND$ $\Longrightarrow$ Σ_i^b–replacement

The Main Theorem

We are now ready to state our main theorem as it applies to first order theories of Bounded Arithmetic.

Theorem 5: Let $i \geq 1$ and A be a Σ_i^b–formula. Suppose

$$S_2^i \vdash (\forall \vec{x})(\exists y)A(\vec{x}, y)$$

Then there is a function $f \in PTC(\Sigma_{i-1}^p)$, a formula $B \in \Sigma_i^b$ and a term t so that

(1) $S_2^i \vdash (\forall \vec{x})(\forall y)(B(\vec{x}, y) \supset A(\vec{x}, y))$
(2) $S_2^i \vdash (\forall \vec{x})(\exists y \leq t)B(\vec{x}, y)$
(3) $S_2^i \vdash (\forall \vec{x})(\forall y)(\forall z)(B(\vec{x}, y) \wedge B(\vec{x}, z) \supset y = z)$
(4) For all $\vec{n}$, $\mathbf{N} \models B(\vec{n}, f(\vec{n}))$

Proof: (Outline.) The proof of Theorem 5 consists logically of two parts.

First, assume we have an S_2^i–proof P of $(\forall \vec{x})(\exists y)A(\vec{x}, y)$. Then by Gentzen's cut elimination theorem there is a term t and an S_2^i–proof P^* of $(\exists y \leq t)A(\vec{x}, y)$ which has no free cuts (see Takeuti [12] for a discussion of Gentzen's cut elimination). In particular, P^* contains only Σ_i^b– and Π_i^b–formulae. This proof P^* is obtained from P by a constructive procedure; however, the size of P^* is bounded only by a non-elementary (superexponential) function of the size of P.

Second, once we have the proof P^* we can obtain a $PTC(\Sigma_{i-1}^p)$–algorithm for computing f. In fact, P^* is a direct description of an algorithm to compute f; that is to say, P^* embodies a $PTC(\Sigma_{i-1}^p)$–algorithm which computes f.

A complete proof of Theorem 5 will appear in [2].

Note that when $i=1$, the function f is in P, i.e., f is a polynomial time function.

Definition: Let R be a theory of Bounded Arithmetic. The function f is Σ_i^b—definable by R iff there is a Σ_i^b—formula B and a term t so that

(1) $R\vdash(\forall\vec{x})(\exists y\leq t)B(\vec{x},y)$
(2) $R\vdash(\forall\vec{x})(\exists!y)B(\vec{x},y)$
(3) For all $\vec{n}$, $\mathbf{N}\models B(\vec{x},f(\vec{x}))$

For all the theories of Bounded Arithmetic discussed in this paper (indeed, for any natural theory of Bounded Arithmetic) the condition (1) in the above definition of Σ_i^b—definable is superfluous. In fact, for these theories, condition (2) implies that there exists a term t such that condition (1) holds. See Parikh [9] for a proof of similar results.

We have the following converse to Theorem 5:

Theorem 6: If $f\in PTC(\Sigma_{i-1}^b)$, then f is Σ_i^b—definable by S_2^i.

Hence the functions Σ_i^b—definable by S_2^i are precisely the functions in $PTC(\Sigma_{i-1}^b)$. In particular, the polynomial time functions are exactly those functions which can be Σ_1^b—defined by S_2^1.

We can restate the above theorems using predicates instead of functions:

Theorem 7: $(i\geq 1)$. Suppose $A\in\Sigma_i^b$, $B\in\Pi_i^b$, and $S_2^i\vdash A\leftrightarrow B$. Then there is a predicate $Q\in\Delta_i^p$ so that for all nonnegative $\vec{n}$,

$$Q(\vec{n}) \iff \mathbf{N}\models A(\vec{n}) \iff \mathbf{N}\models B(\vec{n})$$

Conversely, if $Q\in\Delta_i^p$, then there are A and B so that the above holds.

Corollary 8: If $A(\vec{x})$ is a formula such that S_2^1 proves that A is equivalent to both a Σ_1^b— and a Π_1^b—formula (i.e., S_2^1 proves that $A\in$NP$\cap$co-NP) then $A(\vec{x})$ represents a predicate in P.

Relationship Between S_2^1 and PV

Cook [3] introduced a formal system called PV in an attempt to capture the strength of polynomial time computation in a formal theory. PV is an equational theory (i.e. no quantifiers allowed) which has a function symbol for each polynomial time function and an induction scheme, called "induction on notation," which is analogous to Δ_1^b—$PIND$. It turns out that there is a close relationship between the theories PV and S_2^1.

The first thing to notice is that by Theorem 6, S_2^1 can introduce function symbols for every polynomial time function. In fact, when the proof of Theorem 6 is examined, it is seen that S_2^1 can introduce all of the function symbols of PV in such a way that the introduced symbols provably satisfy all the axioms of PV. This extension of S_2^1 by definitions is called $S_2^1(PV)$.

It is immediately obvious that $S_2^1(PV)$ is an extension of PV. In fact more than that is true:

Theorem 9: Let $t=u$ be an equation of PV. Then $S_2^1(PV)\vdash t=u$ iff $PV\vdash t=u$.

(Cook independently conjectured that Theorem 9 was true.) We can strengthen Theorem 9 as follows. If A is Π_2^b–formula and $S_2^1(PV)\vdash A$ then there is an open equation A^* of PV such that $PV\vdash A^*$ and such that A^* implies A in a natural way. We shall omit the precise statement of this result for lack of space; but as a general idea of what is involved, suppose A is $(\forall x)(\exists y\leqslant t)B(x,y)$ where B is an equation of PV. Then A^* would be of the form $B(x,f(x))\wedge f(x)\leqslant t$ where f is some PV function symbol. We can summarize by saying that, after making allowances for their different languages, S_2^1 and PV have the same Π_2^b–formulae as theorems.

Second Order Bounded Arithmetic

We next will work with second order theories of Bounded Arithmetic by using second order variables which range over predicates. We could also use second order variables for function symbols with polynomial growth rate; however, this adds nothing essentially new, so for simplicity, we shall only use predicate variables. See [2] for a complete discussion of the definition of second order Bounded Arithmetic.

For our second order theories, we modify the definition of *bounded formula* to allow second order quantifiers. A bounded formula is now any formula which has all first order quantifiers bounded and may include arbitrary second order quantifiers. We define a new hierarchy of bounded formulae by counting alternations of second order quantifiers and ignoring first order (bounded) quantifiers.

Definition:

$\Sigma_0^{1,b}=\Pi_0^{1,b}$ is the set of bounded formulae with no second order quantifiers.

$\Sigma_{i+1}^{1,b}$ is defined inductively by :

(1) $\Sigma_{i+1}^{1,b}\supseteq\Pi_i^{1,b}$.

(2) If $A\in\Sigma_{i+1}^{1,b}$ then $(\forall x\leqslant t)A$ and $(\exists x\leqslant t)A$ are in $\Sigma_{i+1}^{1,b}$ (x is a first order variable).

(3) If $A\in\Sigma_{i+1}^{1,b}$ then $(\exists\phi)A$ is in $\Sigma_{i+1}^{1,b}$ (ϕ is a second order variable).

(4) $\wedge$, $\vee$, $\neg$, $\supset$ are treated in the usual fashion.

$\Pi_{i+1}^{1,b}$ is defined dually.

Definition: The $\Sigma_i^{1,b}$–CA comprehension axioms are:

$$(\forall\vec{z})(\forall\vec{\phi})(\exists\psi)(\forall\vec{y})(\psi(\vec{y})\leftrightarrow A(\vec{y},\vec{z},\vec{\phi}))$$

where A is any $\Sigma_i^{1,b}$–formula.

Definition: U_2^i is the second order theory of Bounded Arithmetic with the nonlogical symbols of T_2 and the following axioms:

(1) The open axioms of T_2,

(2) The $\Sigma_0^{1,b}$–CA axioms, and

(3) The $\Sigma_i^{1,b}$–$PIND$ axioms.

V_2^i is defined like U_2^i except that V_2^i has the $\Sigma_i^{1,b}$–IND axioms instead of the $\Sigma_i^{1,b}$–$PIND$ axioms.

Our main theorem for second order Bounded Arithmetic is:

Theorem 10:

(a) The $\Sigma_1^{1,b}$–definable functions of U_2^1 are precisely the PSPACE functions (i.e. the functions computable by a polynomial space bounded Turing machine) which have polynomial growth rate.

(a) The $\Sigma_1^{1,b}$–definable functions of V_2^1 are precisely the EXPTIME functions (i.e. the functions computable by a exponential time bounded Turing machine) which have polynomial growth rate.

Corollary 11: If $U_2^1 \equiv V_2^1$ then PSPACE=EXPTIME.

Of course, it is an open question whether U_2^1 and V_2^1 are equivalent.

Gödel Incompleteness Theorems

One of the most important open questions about Bounded Arithmetic is whether or not the hierarchy of theories

$$S_2^1 \subseteq T_2^1 \subseteq S_2^2 \subseteq T_2^2 \subseteq \cdots$$

is proper. In many ways this is analogous to the open question of whether the polynomial time hierarchy collapses. Alex Wilkie has asked whether or not S_2 is finitely axiomatizable. This is related to our question since S_2 is finitely axiomatizable iff the hierarchy of theories collapses, that is, iff $S_2 \equiv S_2^i$ for some i. Other possibilities include $S_2^i \equiv T_2^i$ for all i, or $S_2^{i+1} \equiv T_2^i$ for all $i \geq 1$. We conjecture that the theories S_2^i and T_2^j are all distinct.

Analogous problems arose in a classical setting when fragments of Peano arithmetic were defined by restricting induction to subclasses of the arithmetic hierarchy. These problems were solved by, on one hand, using Gödel incompleteness arguments to show that each theory can not prove its consistency and, on the other hand, showing that each theory can prove the consistency of the weaker ones. Unfortunately, we have not been able to make these arguments work in the setting of Bounded Arithmetic. However, since the negative results are somewhat interesting in their own right, we present them below.

In S_2^1 we can define Gödel codings for metamathematical concepts such as "term," "formula," "proof," etc. Furthermore, these metamathematical functions can be Σ_1^b–defined and the metamathematical predicates can be Δ_1^b–defined. (A predicate is Δ_1^b–defined iff it is provably equivalent to both a Σ_1^b– and a Π_1^b–formula.) This reflects the fact that all these metamathematical functions and predicates are polynomial time. Also, the metamathematical definitions in S_2^1 are intensionally correct (in the sense of Feferman [5]).

In particular, S_2^1 can define the following formulae:

$Prf^i(u,v) \iff$ "v is the Gödel number of a formula and u is the Gödel number of an S_2^i–proof of v"

$PrfBD^i(u,v) \iff Prf^i(u,v)$ and "the proof u contains no unbounded quantifiers"

$PrfFCF^i(u,v) \iff Prf^i(u,v)$ and "the proof u is free-cut free"

$Con(S_2^i) \iff \neg(\exists u) Prf^i(u, \ulcorner 0=1 \urcorner)$

$BDCon(S_2^i) \iff \neg(\exists u) PrfBD^i(u, \ulcorner 0=1 \urcorner)$

$$FCFCon(S_2^i) \iff \neg(\exists u)(\exists w)(\exists \ulcorner A \urcorner)(Prf FCF^i(u, \ulcorner A \urcorner) \wedge Prf FCF^i(w, \ulcorner \neg A \urcorner))$$

We use $\ulcorner$ and $\urcorner$ as quotation marks meaning "the Gödel number of". The last three formulae express the "consistency", the "bounded consistency" and the "free-cut free consistency" of S_2^i. See Takeuti [12] for the definition of free cut. If R is any axiomatizable theory, then we define the formulae $Con(R)$, $BDCon(R)$, and $FCFCon(R)$ to express the various consistency properties for R.

One further important function which is Σ_1^b–definable in S_2^1 is the unary function $n \longmapsto \ulcorner I_n \urcorner$ where I_n is a term with value equal to n and the length of I_n is proportional to the length $|n|$ of n.

Definition: To improve readability, we use

$$S_2^i \overset{BD}{\vdash} A \quad \text{and} \quad S_2^i \overset{FCF}{\vdash} A$$

to denote the formulae $(\exists u) Prf BD^i(u, \ulcorner A \urcorner)$ and $(\exists u) Prf FCF^i(u, \ulcorner A \urcorner)$, respectively.

Lemma 12: If A is a Σ_1^b–formula, then $S_2^1 \vdash [A(x) \supset (S_2^{(-1)} \overset{FCF}{\vdash} A(I_x))]$.

Theorem 13: Let $i \geq 1$. Then $S_2^i \nvdash FCFCon(S_2^i)$. Hence, $S_2^i \nvdash BDCon(S_2^i)$ and $S_2^i \nvdash Con(S_2^i)$.

The proof of Theorem 13 follows the usual proof of the Gödel incompleteness theorems.

Now that we have seen that S_2^i does not prove its own free-cut free consistency or its own bounded consistency, a natural question is whether S_2^{i+1} can prove the free-cut free or the bounded consistency of S_2^i. If this were the case then S_2^i and S_2^{i+1} would not be equivalent. Unfortunately, the only results we have been able to obtain have been negative.

Lemma 14: Let A be any bounded formula. Suppose $S_2 \vdash (\forall x) A(x)$. Then,

$$S_2^1 \vdash (\forall x)(S_2^{(-1)} \overset{BD}{\vdash} A(I_x)).$$

Theorem 15: $S_2 \nvdash BDCon(S_2^1 + BDCon(S_2^{(-1)}))$.

Proof: (Outline.) Use Gödel diagonalization to obtain a formula $\phi = (\forall x)\phi_M(x)$ such that

$$S_2^1 \vdash [\phi \leftrightarrow (\neg S_2^1 \overset{BD}{\vdash} (\forall x)(S_2^{(-1)} \overset{BD}{\vdash} \phi_M(I_x)))]$$

and use Lemma 14.

Corollary 16: If $S_2^i \vdash BDCon(S_2^{(-1)})$ then $S_2 \nvdash BDCon(S_2^i)$.

Corollary 17: $S_2^{i+1} \vdash BDCon(S_2^i)$ can hold for **at most one** value of i.

Thus it is hopeless to try to show that S_2^{i+1} and S_2^i are different theories by showing that S_2^{i+1} proves the bounded consistency of S_2^i. However, it is an open problem whether S_2^{i+1} proves the free-cut free consistency of S_2^i. We conjecture that this is not the case.

A Proof-Theoretic Statement Equivalent to NP=co-NP

Let R be a theory with a recursively enumerable set of axioms. Then there is a polynomial time function whose range is the set of axioms of R. Hence in S_2^1 we can Δ_1^b–define the predicates $Prf_R(u,v)$ and $PrfBD_R(u,v)$, which assert that u is the Gödel number of a (bounded) proof in the theory R of the formula with Gödel number v. As before, we use $R \overset{BD}{\vdash} A$ as an abbreviation for $(\exists u)PrfBD_R(u,\ulcorner A \urcorner)$.

Definition: Let R be a theory and suppose that the language of R includes the language of Bounded Arithmetic. R is a *bounded* theory iff all axioms of R are bounded. R is *of polynomial growth rate* iff whenever A is bounded and $R \vdash (\forall \vec{x})(\exists y)A(\vec{x},y)$ then there is a term t of the language of Bounded Arithmetic such that $R \vdash (\forall \vec{x})(\exists y \leq t)A(\vec{x},y)$.

It is not difficult to see that if R is an extension of $S_2^{(-1)}$ and R is bounded, then R is of polynomial growth rate.

Theorem 18: The following are equivalent:

(1) There is a bounded, finitely axiomatized, consistent extension R of S_2^1 such that for every bounded formula A,

$$R \vdash (\forall x)[A(x) \supset (R \overset{BD}{\vdash} A(I_x))].$$

(2) There is an axiomatizable, consistent extension R of S_2^1 of polynomial growth rate such that for every Π_1^b–formula A,

$$R \vdash (\forall x)[A(x) \supset (R \vdash A(I_x))].$$

(3) NP = co-NP.

Theorem 18 gives us an interesting reformulation of NP=co-NP. Although this author has had no success trying to prove or disprove (1) and (2), it seems to be a reasonable approach. In particular, the relativizations of Baker-Gill-Solovay [1] do not apply to (1) and (2). To see this, let B be a predicate of [1] so that NP^B=co-NP^B. Then if B is a new predicate symbol in R it is not at all likely that $R \vdash [B(x) \supset (R \vdash B(I_x))]$ and $R \vdash [\neg B(x) \supset (R \vdash \neg B(I_x))]$ both hold.

Theorem 18 is related to a result of Cook-Reckhow [4] on proof systems.

A natural way to try to apply Theorem 18 is by trying to show self-consistency statements are not provably provable. For example, define

$$Con_R(x) \iff \neg(\exists y \leq x)Prf_R(y,\ulcorner 0=1 \urcorner)$$

Unfortunately, we have

Theorem 19: There is bounded, consistent, axiomatizable theory R extending S_2^1 such that

$$R \vdash (\forall x)[Con_R(x) \supset (R \overset{BD}{\vdash} Con_R(I_x))].$$

In fact,

$$R \vdash (\forall x)(R \overset{BD}{\vdash} Con_R(I_x)).$$

Acknowledgements

My advisor Professor S. Kochen has provided me with invaluable assistance. Professor E. Nelson obtained prior results about Gödel's incompleteness theorem and Bounded Arithmetic; I am grateful that he made his unpublished work available to me. I am thankful to Professors R. Lipton, A. Wilkie, P. Pudlak, S. Cook and M. Dowd for their encouragement and suggestions.

[1] **T. Baker, J. Gill, R. Solovay,** "Relativizations of the P=?NP question", SIAM Journal of Computing **4** (1975) 431-442.

[2] **Samuel R. Buss,** *Bounded Arithmetic*, Ph.D. dissertation, Princeton University (to appear) 1985.

[3] **Steven A. Cook,** "Feasibly constructive proofs and the propositional calculus", Seventh ACM Symp. on Theory of Computing (1975) 83-97.

[4] **Steven A. Cook, Robert Reckhow,** "On the lengths of proofs in the propositional calculus", Proc. Sixth ACM Symposium on Theory of Computing, 1974 pp 135-148.

[5] **S. Feferman,** "Arithmetization of metamathematics in a general setting", Fundamenta Mathematicae **49** (1960) 35-92.

[6] **Jay Hook,** *A many-sorted approach to predicative mathematics*, PhD dissertation, Princeton University, 1983.

[7] **Clarence F. Kent, Bernard R. Hodgson,** "An arithmetical characterization of NP", Theoretical Computer Science **21** (1982) 255-267.

[8] **Edward Nelson,** *Predicative Arithmetic*, manuscript (to appear).

[9] **Rohit J. Parikh,** "Existence and feasibility in arithmetic", Journal of Symbolic Logic **36** (1971) 494-508.

[10] **Jeff Paris, L.A.S. Kirby,** "Σ_n collection schemes in arithmetic," in *Logic Colloquium '77*, North-Holland, 1978, pp. 199-210.

[11] **Larry J. Stockmeyer,** "The polynomial-time hierarchy", Theoretical Computer Science **3** (1976) 1-22.

[12] **Gaisi Takeuti,** *Proof Theory*, North-Holland 1975.

[13] **Alex Wilkie, Jeff Paris,** "On the scheme of induction for bounded arithmetic formulas" , *Logic Colloquium '84, Proc. of an ASL Conference in Manchester, England*, North-Holland (to appear).

[14] **Celia Wrathall,** "Complete sets and the polynomial time hierarchy", Theoretical Computer Science **3** (1976) 23-33.