Bounded Arithmetic vs. Propositional Proof Systems vs. Complexity Classes (depth oral survey)

`$Id: alanodepth.tex,v 1.18 2002/04/01 04:54:38 alan Exp $` LaTeX'd on January 3, 2005

Alan Skelley

January 3, 2005

1 Introduction

We often argue that a particular mathematical concept is important if it is natural, which means that it surfaces in many places with different origins and definitions, and robust, such that a variety of disparate formulations of it end up being equivalent or at least closely related. Likewise, the applicability, maturity, and importance of a body of results are greater when that field is found to have a strong connection to another. Three areas of study intricately connected in such a useful way are computational complexity, the proof theory of arithmetic and propositional proof complexity.

Computational complexity is the study of computation and the resources required to perform it. A staggering number of different kinds of computation all fall into the domain of this field. It has practical aspects, directly impacting how real computations are done by real computers, and yet seemingly fundamental, easily explained problems remain unsolved despite a good deal of effort. A particularly glaring example is the famous P vs NP problem, which asks if those two classes of problems are equal. Starting from the NP-completeness results of Cook [13] the pressure mounted with no relief, leading even to detailed, formal analysis of known proof techniques and why they are all ineffectual at tackling such problems [30]. Many complexity classes are studied and conjectures about separations and hierarchies abound, yet results are elusive.

A different way of studying computational complexity is indirectly through logic, and in particular, bounded arithmetic. Many connections between the fields are known: among them, that complexity classes can be characterized as those sets or functions definable in certain theories, and that predicates or functions from certain complexity classes can be used to define new logics in various ways. Results about either area can have implications for the other.

Bounded arithmetic and propositional proof systems are related in several ways: due to Cook [14] and others, there are translations from formulas of bounded arithmetic to polynomial-sized families of propositional or quantified propositional formulas which additionally have very interesting properties relating the theories and the proof systems, and also have complexity implications. Another connection is that a theory's ability to prove different kinds of consistency of related propositional proof systems has a bearing on its power relative to other theories, and the relative complexity of proofs in the proof systems.

Finally, the full circle back to computational complexity is completed with the work of Cook and Reckhow in [10] and [16]. They show that NP=co-NP if and only if there exists a polynomially bounded proof system, and additionally introduce many of the important definitions in the area such as those of proof systems, polynomial simulations, and so on. These results, and others concerning the complexity of witnessing proofs of quantified propositional formulas, drive the study of propositional proof complexity and the search for lower bounds on propositional proof systems. Fine examples are the superpolynomial lower bounds for resolution, due to Haken [17] and bounded depth Frege systems, due to Ajtai [1]. For many seemingly stronger systems, however, no such results are known.

In this survey we will discuss the various results hinted at above. We intentionally omit from our focus some of the weaker systems of bounded arithmetic and propositional proof systems about which some lower bounds are known and concentrate instead on stronger systems and theories about which good bounds or separations are only conjectured. The survey is organized as follows: In section 2 we introduce several systems of bounded arithmetic and results concerning them and relating them to complexity theory. In section 3 we present some relevant propositional proof systems and their complexity-theoretic ramifications. Section 4 contains a discussion of results relating bounded arithmetic and propositional proof systems, and finally in section 5, we mention open problems and some possible research topics related to the results discussed.

2 Bounded Arithmetic and Complexity

The study of bounded arithmetic was initiated in 1971 by Parikh with his system $I\Delta_0$, similar to Peano Arithmetic, but with the important restriction of the induction scheme to Δ_0 formulas: those whose quantifiers are bounded, i.e. of the form $Qx(x \leq t(x) \to \phi(x))$ for some quantifier Q and term $t(x)$ in the language. An important consequence of this restriction is given by Parikh's theorem, which states that any function which can be proved total in $I\Delta_0$ can be bounded by a term in the language:

Theorem 2.1 (Parikh, 1971). *Assume that $\theta(\overline{a}, b)$ is a Δ_0 formula and that*

$$I\Delta_0 \vdash \forall \overline{x} \exists y \theta(\overline{x}, y).$$

Then there is a term $t(\overline{x})$ such that

$$I\Delta_0 \vdash \forall \overline{x} \exists y < t(\overline{x}), \theta(\overline{x}, y).$$

This theorem implies that $I\Delta_0$ cannot prove theorems requiring exponentiation, or even the existence of numbers whose length is polynomial in the length of input parameters. This rules out reasoning about any computations using more than linear time, many logical notions such as substitution and polynomial-length sequences, and so on.

The first logical theory designed to reason about all "feasible," i.e. polynomial-time concepts, and only those concepts, was Cook's PV [14]. This is an equational theory which has function symbols for every polynomial-time function, and is defined with the help of Cobham's earlier characterization of polynomial-time functions as the closure of a certain set of initial functions under composition and limited recursion on notation. A fundamental property of PV is its connection to the propositional proof system EF, which will be discussed in section 4.1. Later first-order theories IPV and CPV [12], the former intuitionistic and the latter classical, have a more expressive language allowing interesting properties of graph theory and combinatorics to be stated, yet are conservative over PV, which is to say that every statement in the language of PV which is provable in IPV or CPV is provable in PV. Krajíček, Pudlák and Takeuti [24] defined a hierarchy of theories PV_i based on PV, whose lowest member, PV_1, is also referred to as QPV.

Following Cook's work on PV, to remedy the deficiency in $I\Delta_0$ Paris and Wilkie tried adding function symbols with faster growth rates such as $\omega_1(x) := x^{|x|}$, along with appropriate defining axioms, to the language. The addition of ω_1 in particular results in a very interesting theory $I\Delta_0 + \Omega_1$ with many implications for and connections to complexity theory and propositional proof systems, and it is mainly with subtheories of this theory that we shall be concerned.

2.1 Definitions

Buss [4] introduced the theories S_2^i and T_2^i, which we shall describe shortly. First, consider the following hierarchy of classes of formulas, the definition of which is a slight adaptation of the definition in [20]:

Definition 2.2 (Buss, 1986). *Σ_i^b and Π_i^b are the smallest classes of formulas satisfying the following:*

1. *$\Sigma_0^b = \Pi_0^b$ are the sharply bounded formulas, whose quantifiers are all of the form $(Qx < |t|), Q \in \{\forall, \exists\}$ for some term t.*
2. *If ϕ is Σ_i^b or Π_i^b then it is also Σ_j^b and Π_j^b for all $j > i$.*
3. *If $\phi(x)$ is Σ_i^b then $\forall x < t(x)\phi(x)$ is Π_{i+1}^b.*
4. *If $\phi(x)$ is Π_i^b then $\exists x < t(x)\phi(x)$ is Σ_{i+1}^b.*
5. *If ϕ is Σ_i^b (Π_i^b) then $\neg\phi$ is Π_i^b (Σ_i^b respectively).*
6. *Σ_i^b and Π_i^b are closed under $\vee$ and $\wedge$.*
7. *Σ_i^b (Π_i^b) is closed under existential (universal) quantification and sharply bounded quantification.*

Now with these classes in mind, S_2^i and T_2^i are theories over the language L_2 consisting of the language of PA with the addition of $\{\lfloor \frac{x}{2} \rfloor, |x|, x\#y\}$, where it is intended that $x\#y = 2^{|x||y|}$. Both theories contain BASIC, a set of 32 open axioms expressing properties of the symbols in the language, and in addition T_2^i has the scheme Σ_i^b-IND:

$$\phi(0) \wedge \forall x(\phi(x) \rightarrow \phi(x+1)) \rightarrow \forall x\phi(x)$$

while S_2^i has instead the scheme Σ_i^b-PIND:

$$\phi(0) \wedge \forall x(\phi(\lfloor x \rfloor) \rightarrow \phi(x)) \rightarrow \forall x\phi(x)$$

in each case for every $\phi \in \Sigma_i^b$. Clearly $S_2^i \subseteq T_2^i$, and it can be shown that $T_2^i \subseteq S_2^{i+1}$.

The “2” subscript refers to the presence of the $x\#y$, or smash function, in the language. Other possibilities for this subscript include “1”, meaning that no smash function is present, or $i > 2$, meaning that the function $x\#_i y := 2^{|x|\#_{i-1}|y|}$ is present.

Buss also defined second-order theories U_2^1 and V_2^1. He first gives a definition analogous to 2.2 for bounded second-order formulas, where $\Sigma_i^{1,b}$ formulas are classified by counting the alternations of bounded second-order quantifiers and ignoring bounded first-order quantifiers. U_2^1 (respectively V_2^1) is the theory composed of the BASIC axioms, $\Sigma_0^{1,b}$-Comprehension, which postulates the existence of second-order objects equivalent to given $\Sigma_0^{1,b}$ predicates of one variable, and $\Sigma_1^{1,b}$-PIND (respectively, $\Sigma_1^{1,b}$-IND).

A crucial definition is that of definability:

Definition 2.3. *Let Φ be a class of formulas, T be a theory of bounded arithmetic and $f : \mathbb{N}^k \rightarrow \mathbb{N}$ a function. Then f is Φ-definable in T iff there exists a formula $D_f(\overline{x}, y) \in \Phi$ such that*

$$T \vdash \forall \overline{x} \exists y D_f(\overline{x}, y),$$

and $D_f(\overline{x}, f(\overline{x}))$ is true in the standard model.

A function is strongly definable if the theory additionally proves that the y satisfying $D_f(\overline{x}, y)$ is unique.

2.2 Definability and Witnessing Theorems

In this section we list some of the important results connecting theories of bounded arithmetic and complexity classes through definability of functions.

The main results of Buss [4] are as follows: Firstly, that the strongly Σ_i^b-definable functions of S_2^i are exactly those computable in polynomial time with an oracle for a Σ_{i-1}^p predicate, i.e. functions from a functional version of the well known polynomial-time hierarchy. Furthermore, if S_2^1 proves that a predicate is in NP $\bigcup$ co-NP, then it is in fact in P. He also shows how to relativise S_2^i by adding a free second-order variable, and that an analogous definability result connects these theories to computations with an oracle. Secondly, he shows that $S_2^1(\mathrm{PV})$, which is S_2^1 extended by the language of PV and axioms defining all its function symbols, is conservative over PV. Finally, Buss shows that that the strongly $\Sigma_1^{1,b}$-definable functions of U_2^1 and V_2^1 are those computable in polynomial space, and those computable in exponential time, respectively. These latter results for second order theories are extended by Buss, Krajíček and Takeuti [6] to U_2^i and V_2^i, and the many analogues between first- second-order theories are seen to be part of a pattern formalized in the RSUV isomorphism of Takeuti [32] and Razborov [29].

Later results have added to what is known about definability in these theories. Of particular interest is the fact from [5] that the definable functions of T_2^1 are exactly those expressible as the composition of a PLS problem and a projection, where PLS is Papadimitriou's NP search class of polynomial local search problems. Chiari and Krajíček have extended this result to characterize the Σ_2^b and Σ_3^b definable multifunctions in T_2^2 as oracle PLS problems and suggest that a more complete understanding of these and related definabilites will be useful for proving non-conservation results. Another important example, which shall figure prominently in the next subsection, is the Krajíček-Pudlák-Takeuti (KPT) witnessing theorem [24]:

Theorem 2.4 (Krajíček, Pudlák and Takeuti, 1991). *Let $i \geq 1$ and assume that $\phi(a, x, y)$ is an $\exists\Pi_i^b$-formula. Suppose*

$$T_2^i \vdash \exists x \forall y, \phi(a, x, y)$$

Then there are $\square_{i+1}^p$-functions $f_1(a), f_2(a, b_1), ..., f_k(a, b_1, ..., b_{k-1})$ with all free variables shown such that T_2^i proves

$$\phi(a, f_1(a), b_1) \vee \phi(a, f_2(a, b_1), b_2) \vee ... \vee \phi(a, f_k(a, b_1, ..., b_{k-1}), b_k)$$

This is also true for PV_{i+1} in place of T_2^i and for PV_1 if $i = 0$.

2.3 Relating the Collapse of Theories with the Collapse of Complexity Classes

Since results are known characterizing fairly precisely the definable functions of many theories, it is reasonable to expect some relation between questions of theories coinciding versus questions of complexity classes coinciding. This is certainly the case of the $S_2 = T_2$ hierarchy under discussion, which will serve as a good example. Something to note at the start is a nice feature of the theories S_2^i and T_2^i, namely that each of them is finitely axiomatizable [23]; therefore, the S_2 hierarchy collapses iff S_2 itself is finitely axiomatizable.

Now, if it were the case not only that the polynomial hierarchy collapsed, but also that this collapse was uniform enough that S_2 could prove it, then the S_2 hierarchy would also collapse. This is so intuitively because some sufficiently high level of S_2 would be strong enough to prove all the induction axioms of S_2, by proving them equivalent due to the PH collapsing to induction axioms of lower quantifier complexity. There is still however the possibility that the PH could collapse but that the proof of that fact might not be formalizable in S_2, in which case the S_2 hierarchy might still be strict. This type of relationship seems to be typical of theories and the complexity classes of functions definable in them; for another example see Cook [15].

In the other direction, the KPT witnessing theorem stated above implies that if the S_2 hierarchy collapses then so does the PH. Buss [7] and Zambella [33] independently strengthen this result by showing that the collapse of the PH would in fact be provable in S_2.

A general pattern is that the collapse of complexity classes seems to be related most closely to the collapse of particular fragments of related theories. In many cases, the status of other fragments of the theories may have different or unknown implications. For example, the collapse of the universal fragments of the theories S_2^i does not obviously imply the collapse of the entire theories (and thus of the PH). Another example is that although we know that $S_2^1(\mathrm{PV})$ is conservative over PV, as is QPV, the KPT witnessing theorem just discussed tells us that if S_2^1 is conservative over QPV, then the PH collapses. Finally, it is not known how the potential equality of PSPACE and PH may be related to the question of conservativity of U_2^1 over S_2, although it is plausible that some relation may hold. Certainly there are many unsolved problems of this kind which are meritorious of further attention.

2.4 Candidates for Separation

The standard candidate for separating a theory from one containing it would be the consistency of the smaller theory. However, Paris and Wilkie [27] show that even S_2 augmented with an axiom stating the totality of exponentiation does not prove the consistency of the induction-free Robinson's Arithmetic Q. Not even $\mathrm{BdCon}(S_2^1)$, a restricted consistency statement asserting only that the bounded fragment of S_2^1 is consistent, can be proved in S_2 [28]. More natural candidates, then, would be theorems of mathematics whose proofs require reasoning about concepts which are not in the corresponding complexity class of definable functions of the weaker theory; however, actually finding these seems to be difficult. The most natural candidates appear to be statements of consistency of related propositional proof systems, which will be discussed in section 4.2.

3 Propositional Proof Systems and Complexity

In this section we discuss some of the many connections between propositional proof systems, which we first formally define, and complexity. The first connection is visible even as the definitions are presented; namely, that when formulated in a Gentzen sequent style, many known propositional proof systems can be seen to be very similar, with the only difference between them being the computational power of what can be written at each line of the proof (or alternatively, what is allowed in the **cut** rule). Examples are Boolean formulas in Frege systems, single literals in resolution, Boolean circuits in extended Frege systems. Another example is the system G, which is a sequent-based system where formulas in the sequents are quantified boolean formulas (QBFs). These formulas have propositional variables and also propositional quantifiers. In this case, then, since evaluating QBFs is PSPACE-complete, the computational power which can be harnessed in sequents is PSPACE. We can restrict G to G_i by restricting the number of alternations of quantifiers allowed in the formulas, and the reasoning power is then that of Σ_i^p predicates.

3.1 Preliminaries

3.1.1 Propositional Proof Systems

Definition 3.1. *A* proof system *P for a set S is a surjective polynomial-time computable function $P : \Sigma^* \to S$ for some alphabet Σ.*

We are interested in proof systems both for TAUT, the set of (quantifier-free) propositional tautologies, and for TAUT_i, the set of quantified propositional tautologies from $\Sigma_i^q \bigcup \Pi_i^q$, to be defined below. A P-proof of a tautology τ is a string π such that $P(\pi) = \tau$. We denote by $|\pi|$ the number of symbols in π. We have the following important notion which allows us to compare the power of proof systems:

Definition 3.2. *If P and Q are proof systems, we say that P* polynomially simulates *(p-simulates) Q and write $P \leq_p Q$ if there is a polynomial-time computable function g such that for every string x, $P(g(x)) = Q(x)$.*

Though proof systems need not be of this form, proofs in any of the systems commonly studied are sequences of lines, where each line is a valid statement in some language. Such systems then have a treelike subsystem, wherein each line may be used only once as a hypothesis.

3.1.2 LK and Quantified Propositional Logic

A popular proof system is Gentzen's sequent system LK. LK is actually a proof system for predicate logic but we shall consider only the propositional fragment. Each line of an LK-proof is a sequent, a string of the form $\Gamma \longrightarrow \Delta$, where Γ and Δ are possibly empty finite sequences of propositional formulas. A sequent is satisfied if and only if either one of the formulas on the left (the *antecedent*) is falsified, or one of the formulas on the right (the *succedent*) is satisfied. Each sequent in a proof is either an initial sequent of the form $0 \longrightarrow$, $\longrightarrow 1$ or $a \longrightarrow a$ for an atom a, or it is derived from previous ones (its hypotheses) via one of the following inference rules (this set is the same as in [9], which is a slight modification of the ones in [20]):

weakening:

$$\textbf{left}\quad \frac{\Gamma \longrightarrow \Delta}{A, \Gamma \longrightarrow \Delta} \qquad \text{and} \qquad \textbf{right}\quad \frac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, A}$$

exchange:

$$\textbf{left}\quad \frac{\Gamma_1, A, B, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, B, A, \Gamma_2 \longrightarrow \Delta} \qquad \text{and} \qquad \textbf{right}\quad \frac{\Gamma \longrightarrow \Delta_1, A, B, \Delta_2}{\Gamma \longrightarrow \Delta_1, B, A, \Delta_2}$$

contraction:

$$\textbf{left}\quad \frac{\Gamma_1, A, A, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, A, \Gamma_2 \longrightarrow \Delta} \qquad \text{and} \qquad \textbf{right}\quad \frac{\Gamma \longrightarrow \Delta_1, A, A, \Delta_2}{\Gamma \longrightarrow \Delta_1, A, \Delta_2}$$

$\neg$: introduction:

$$\textbf{left}\quad \frac{\Gamma \longrightarrow \Delta, A}{\neg A, \Gamma \longrightarrow \Delta} \qquad \text{and} \qquad \textbf{right}\quad \frac{A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \neg A}$$

$\wedge$: introduction:

$$\textbf{left}\quad \frac{A, B, \Gamma \longrightarrow \Delta}{A \wedge B, \Gamma \longrightarrow \Delta} \qquad \text{and} \qquad \textbf{right}\quad \frac{\Gamma \longrightarrow \Delta, A \quad \Gamma \longrightarrow \Delta, B}{\Gamma \longrightarrow \Delta, A \wedge B}$$

$\vee$: introduction:

$$\textbf{left}\quad \frac{A, \Gamma \longrightarrow \Delta \quad B, \Gamma \longrightarrow \Delta}{A \vee B, \Gamma, \longrightarrow \Delta} \qquad \text{and} \qquad \textbf{right}\quad \frac{\Gamma \longrightarrow \Delta, A, B}{\Gamma \longrightarrow \Delta, A \vee B}$$

cut:

$$\frac{\Gamma \longrightarrow \Delta, A \quad A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta}$$

This system is p-equivalent to Frege systems, defined in [16]. When we add the additional rule that for a previously unused variable r and any formula ϕ, the sequents $r \longrightarrow \phi$ and $\phi \longrightarrow r$ may be introduced, and further stipulate that these extension atoms may not appear in the endsequent of a proof, we obtain a system equivalent to extended Frege systems, from the same paper.

Quantified propositional logic is what results when we add propositional quantifiers to our language. The semantics of $\forall x \phi(x, \overline{p})$ is that this formula is satisfied by a particular assignment if and only if $\phi(0, \overline{p}) \wedge \phi(1, \overline{p})$ is. Likewise the truth value of $\exists x \phi(x, \overline{p})$ is the same as that of $\phi(0, \overline{p}) \vee \phi(1, \overline{p})$. We can define a hierarchy of quantified Boolean semiformulas. The following is completely analogous to Definition 2.2:

Definition 3.3. *The classes Π_i^q and Σ_i^q are defined as follows:*

1. *$\Sigma_0^q = \Pi_0^q$ are the quantifier-free propositional semiformulas.*
2. *If ϕ is Σ_i^q or Π_i^q then it is also Σ_j^q and Π_j^q for all $j > i$.*
3. *If $\phi(x)$ is Σ_i^q then $\forall x \phi(x)$ is Π_{i+1}^q.*
4. *If $\phi(x)$ is Π_i^q then $\exists x \phi(x)$ is Σ_{i+1}^q.*
5. *If ϕ is Σ_i^q (Π_i^q) then $\neg\phi$ is Π_i^q (Σ_i^q respectively).*
6. *Σ_i^q and Π_i^q are closed under $\vee$ and $\wedge$.*
7. *Σ_i^q (Π_i^q) is closed under existential (universal) quantification.*

Now, the proof system G is obtained by augmenting the set of inference rules of LK with the following:

$\forall$: introduction:

$$\textbf{left} \quad \frac{A(B), \Gamma \longrightarrow \Delta}{\forall x A(x), \Gamma \longrightarrow \Delta} \qquad \text{and} \qquad \textbf{right} \quad \frac{\Gamma \longrightarrow \Delta, A(p)}{\Gamma \longrightarrow \Delta, \forall x A(x)}$$

$\exists$: introduction:

$$\textbf{left} \quad \frac{A(p), \Gamma \longrightarrow \Delta}{\exists x A(x), \Gamma \longrightarrow \Delta} \qquad \text{and} \qquad \textbf{right} \quad \frac{\Gamma \longrightarrow \Delta, A(B)}{\Gamma \longrightarrow \Delta, \exists x A(x)}$$

where B is any formula and the atom p replaced does not occur in the conclusion of the corresponding inference. G_i is G with the restriction that all formulas appearing in a proof must be Σ_i^q or Π_i^q. We shall consider G not only as a proof system for TAUT, but also for TAUT$_i$.

3.1.3 Boolean Programs

Boolean programs were introduced in [11] and are a way of specifying Boolean functions. Boolean programs are something like a generalization of the technique of using new atoms to replace part of a Boolean formula, which idea is the basis of extended Frege systems. As is the case with that system, and more so with the quantified propositional calculus, it appears that the use of boolean programs allows formulas to be abbreviated significantly. The following definition is from that paper:

Definition 3.4 (Cook-Soltys). *A* Boolean Program *P is specified by a finite sequence $\{f_1, ..., f_m\}$ of function symbols, where each symbol f_i has an associated arity k_i, and an associated defining equation*

$$f_i(\overline{p_i}) := A_i$$

where $\overline{p_i}$ is a list $p_1, ..., p_{k_i}$ of variables and A_i is a formula all of whose variables are among $\overline{p_i}$ and all of whose function symbols are among $f_1, ..., f_{i-1}$. In this context the definition of a formula *is:*

1. *0,1, and p are formulas, for any variable p.*

2. *If f is a k-ary function symbol in P and $B_1, ..., B_k$ are formulas, then $f(B_1, ..., B_k)$ is a formula.*

3. *If A and B are formulas, then $(A \wedge B), (A \vee B)$ and $\neg A$ are formulas.*

The semantics are as for propositional formulas, except that when evaluating an application $f_i(\overline{\phi})$ of a function symbol, the value is defined, using the defining equation, to be $A_i(\overline{\phi})$.

An interesting property of Boolean programs which demonstrates their comparability to quantified Boolean formulas is the following theorem from [11]:

Theorem 3.5 (Cook-Soltys). *A Language L is in PSPACE iff L is computed by some uniform polynomial-size family of Boolean programs.*

3.1.4 BPLK

Definition 3.6 (BPLK). *The system BPLK is like the propositional system LK, but with the following changes:*

1. *In addition to sequents, a proof also includes a Boolean program which defines functions. Whenever we refer to a BPLK-proof, we shall always explicitly write it as the pair $< \pi, P >$ of the proof (sequents) and the Boolean program defining the function symbols occurring in the sequents.*

2. *Formulas in sequents are formulas in the context of Boolean programs, as defined earlier.*

3. *If the Boolean program contains a definition of the form*
$$f(\overline{p}) := A(\overline{p}),$$
the new LK rules f : **left**
$$\frac{A(\overline{\phi}), \Gamma \longrightarrow \Delta}{f(\overline{\phi}), \Gamma \longrightarrow \Delta}$$
and f : **right**
$$\frac{\Gamma \longrightarrow \Delta, A(\overline{\phi})}{\Gamma \longrightarrow \Delta, f(\overline{\phi})}$$
may be used, where $\overline{\phi}$ are precisely as many formulas as $\overline{p}$ are variables.

4. *(***Substitution Rule***) The new inference rule* **subst**
$$\frac{\Delta(q, \overline{p}) \longrightarrow \Gamma(q, \overline{p})}{\Delta(\phi, \overline{p}) \longrightarrow \Gamma(\phi, \overline{p})}$$
may be used, where all occurrences of q have been substituted for.

The following is the main result of [31]:

Theorem 3.7. *BPLK and G are polynomially equivalent.*

3.2 Complexity-Related Results

The primary motivation for studying propositional proof systems is the theorem of Cook and Reckhow [16] that NP=co-NP iff there exists a polynomially bounded proof system for propositional tautologies. There are, fortunately, many questions about these systems with less severe complexity-theoretic consequences than this one. One such question is how exactly the expressive power of a line of the proof relates to the relative efficiency of the system, which will be discussed in section 4.1. In this subsection we will discuss how other modifications to a proof system, such as the restriction to treelike proofs or the addition of a substitution rule, affects its efficiency. We will also talk about the witnessing problem for proofs of quantified tautologies.

3.2.1 Known Simulation Results

At the bottom of the G hierarchy of proof systems, which is already well above where the known lower bound results apply, we have G_0 which is polynomially equivalent to LK and Frege systems. It is also p-equivalent to its treelike subsystem G_0^* (since Frege and treelike Frege are p-equivalent [19]), something which is not known for G_i, $i > 0$. The next step up are Extended Frege and Substitution Frege systems, which are p-equivalent due to Krajíček and Pudlák [22]. These are also both p-equivalent to G_1^*. For $i > 0$, G_i p-simulates G_{i+1}^* for proofs of TAUT$_i$ [20]. The converse simulation can also be shown, either directly or with the help of results such as those in section 4.1 and the conservativity of S_2^{i+1} over T_2^1. Another way of stating this last result is that substitution-G_i is p-equivalent to G_{i+1}^* for proofs of TAUT$_i$ for all i (including $i = 0$), and for $i > 0$, substitution is a derived rule in G_i [23].

3.2.2 Witnessing Problem for Quantified Propositional Proofs

The witnessing problem for quantified propositional proofs is the following: Given a proof of a quantified propositional tautology in Σ_i^q, and values for the free variables in the endsequent, find values for the outermost existentially quantified variables of the endsequent satisfying it. For G_1^* proofs, this problem is in P, and for G_1 proofs (of Σ_1^q tautologies), it is complete for PLS. It follows from [8] and the results in the next section that the witnessing problem for G_i is complete for an oracle version of PLS with a Σ_{i-1}^p oracle, defined in that paper, for each $i > 0$. For $i = 2$, the authors find an equivalent search problem they call GLS, for generalized local search. It is open to find more natural search problems for the rest of the cases, and it is also open to find any characterization of the witnessing problems for G_i proofs of Σ_j^q tautologies for $1 \leq j < i$. Another open problem is to find propositional proof systems whose witnessing problem corresponds to one of the other well-studied NP search classes. This can be done unnaturally by adding axioms asserting the totality of these search problems to EF.

4 Bounded Arithmetic and Propositional Proof Systems

In this section we discuss some connections between systems of bounded arithmetic and propositional proof systems.

4.1 Translations into Propositional Logic

The most important such connection is that some classes of theorems of some bounded arithmetic theories can be translated into families of propositional or quantified propositional tautologies. Depending on what the theory is and what class of formulas is translated, we can draw conclusions about the lengths of proofs of these families of tautologies in various propositional proof systems. Furthermore, by adding reflection principles, axioms stating the consistency of a propositional proof system, to a weaker theory, we can axiomatize a stronger theory corresponding to that proof system.

The first result of this form is due to Cook [14] who defines a translation from equations of PV to families of propositional formulas with polynomial-size EF proofs. Furthermore, any propositional proof system whose consistency PV can prove can be p-simulated by EF. Independently, Paris and Wilkie [26] gave a translation from bounded first-order formulas with a relation symbol R to families of propositional tautologies, and proved that if $I\Delta_0 \vdash \forall x \theta(x)$ then the translations of $\theta(x)$ have polynomial-size Frege proofs. Krajíček [21] extends this translation to handle second-order formulas and shows a similar relation between V_1^1 and polynomial-sized EF proofs, and between U_1^1 and quasipolynomial-sized Frege proofs.

Krajíček and Pudlák [23] extended Cook's result to show that whenever $A(a) \in \Sigma_i^b$ and $S_2^i \vdash A(a)$ (respectively, $T_2^i \vdash A(a)$), then the translations of $A(a)$ have polynomial-size G_i^* (respectively, G_i) proofs. Krajíček and Takeuti [25] showed a similar relation between U_2^1 and G, and such a result probably holds for BPLK as well.

It is interesting to note that in some cases, the propositional proof system corresponding to a complexity class has as lines in its proofs objects which are of exactly that complexity class (for example, G, EF) yet in other cases, the objects are of seemingly greater computational power (G_1, G_1^*). An interesting open problem is to find, for some of the latter type of examples, a canonical propositional proof system whose lines are exactly the appropriate complexity class.

4.2 Consistency Strength

Using the idea of Cook [14], [22], [25] and others define reflection principles $i - RFN(P)$ for each i and propositional proof system P, which states that P is sound for proofs of $\Sigma_i^q \bigcup \Pi_i^q$ tautologies. We have that for every i, $S_2^i \vdash i{-}RFN(G_i^*)$, $T_2^i \vdash i{-}RFN(G_i)$ and $U_2^1 \vdash i{-}RFN(G)$. Furthermore, for any proof system P such that one of the above theories, for example S_2^i, proves the reflection principle $j - RFN(P)$ for some j, the corresponding proof system, in this case G_i^*, p-simulates P for proofs of TAUT_j. In fact, every $\forall\Sigma_j^b$ consequence of S_2^i (T_2^i, U_2^1) follows from $S_2^1{+}j - RFN(G_i^*)$ (G_i, G). For this reason, these reflection principles would be candidates for separating the theories.

5 Conclusions and Open Problems

In this section we summarize some open problems related to the results discussed above.

5.1 Universal Fragments of Theories

As discussed above, it is plausible that the universal fragments of, for example, U_2^1 and S_2 might be the same without causing any complexity collapse. It would be instructive either to collapse these fragments or to find convincing reasons why it might be impossible. A related issue is that of provability of quantifier-free tautologies in the various subsystems of G. There does not seem to be any drastic consequence to complexity theory of showing, for example, that G_1 p-simulates G for such proofs.

5.2 Witnessing and Search Problems

Several lines of research are suggested: First, it would be interesting to characterize the hardness of the witnessing problems for the other subsystems of G, and indeed different kinds of definability in the subsystems of T_2 and S_2. Part of this work has recently been done by Chiari and Krajíček in [8] for Σ_2^b and Σ_3^b definability in T_2^2 but nothing general is known yet. Secondly, there are other local search problems than PLS, some of which are discussed in [18] and in more detail in [2]. It would be interesting to find propositional proof systems whose witnessing problems were exactly projections of these other local search problem classes.

5.3 Subsystems of BPLK

Another set of questions which are particularly interesting concerns the possibility of finding natural subsystems of BPLK, akin to the structure of G. In their paper [11], the authors find a natural restriction of Boolean programs, essentially amounting to extension axioms, for witnessing proofs in G_1^*. It would be instructive to find restrictions of Boolean programs which would naturally witness proofs in other subsystems of G. It would also be interesting to find some kind of a hierarchy within BPLK which may or may not correspond to the hierarchy in G.

5.4 Hard Tautologies for F or EF

Finding hard tautologies for Frege or Extended Frege systems would certainly be of interest but many authors are pessimistic about the prospects for this. See for example [3].

5.5 Canonical Proof Systems for Classes

As discussed earlier, some of the well known proof systems seem to be overpowered in terms of what can be expressed on each line. Finding an equivalent canonical proof system for these examples is an interesting problem.

5.6 Theories and Proof Systems for Other Complexity Classes

There are many complexity classes for which no corresponding theory or proof system is known. Examples include some NP search classes, but are by no means limited to these. Finding a corresponding theory and proof system and positioning it correctly with respect to already known examples could potentially prove very instructive.

References

[1] M. Ajtai. The complexity of the pigeonhole principle. In *29th Annual Symposium on Foundations of Computer Science*, pages 346–355, White Plains, New York, 24–26 October 1988. IEEE.

[2] Paul Beame, Stephen Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. *Journal of Computer and System Sciences*, 57(1):3–19, August 1998.

[3] Maria Luisa Bonet, Samuel R. Buss, and Toniann Pitassi. Are there hard examples for frege systems? In *P. Clote, J. Remmel (eds.): Feasible Mathematics II*, pages 30–56. Birkhäuser, Boston, 1995.

[4] S. Buss. *Bounded Arithmetic.* Bibliopolis, Naples, 1986.

[5] Samuel Buss and Jan Krajíček. An application of Boolean complexity to separation problems in bounded arithmetic. *Proceedings of the London Mathematical Society*, 69:1–21, 1994.

[6] Samuel Buss, Jan Krajíček, and Gaisi Takeuti. On provably total functions in bounded arithmetic theories R^i_3, U^i_2 and V^i_2. In Peter Clote and Jan Krajíček, editors, *Arithmetic, proof theory and computational complexity*, pages 116–61. Oxford University Press, Oxford, 1993.

[7] Samuel R. Buss. Relating the bounded arithmetic and polynomial time hierarchies. *Annals of Pure and Applied Logic*, 75(1–2):67–77, 12 September 1995.

[8] Mario Chiari and Jan Krajíček. Witnessing functions in bounded arithmetic and search problems. *The Journal of Symbolic Logic*, 63(3):1095–1115, September 1998.

[9] S. A. Cook. CSC 2429S: Proof Complexity and Bounded Arithmetic. Course notes, URL: "http://www.cs.toronto.edu/∼sacook/csc2429_98", Spring 1998.

[10] Stephen Cook and Robert Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In *Conference Record of Sixth Annual ACM Symposium on Theory of Computing*, pages 135–148, Seattle, Washington, 30 April–2 May 1974.

[11] Stephen Cook and Michael Soltys. Boolean programs and quantified propositional proof systems. *Bulletin of the Section of Logic*, 28(3), 1999.

[12] Stephen Cook and Alasdair Urquhart. Functional interpretations of feasibly constructive arithmetic. *Annals of Pure and Applied Logic*, 63(2):103–200, 10 September 1993.

[13] Stephen A. Cook. The complexity of theorem-proving procedures. In *Conference Record of Third Annual ACM Symposium on Theory of Computing*, pages 151–158, Shaker Heights, Ohio, 3–5 1971 1971.

[14] Stephen A. Cook. Feasibly constructive proofs and the propositional calculus (preliminary version). In *Conference Record of Seventh Annual ACM Symposium on Theory of Computing*, pages 83–97, Albuquerque, New Mexico, 5–7 May 1975.

[15] Stephen A. Cook. Relating the provable collapse of P to NC^1 and the power of logical theories. *DIMACS Series in Discrete Math. and Theoretical Computer Science*, 39, 1998.

[16] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44:36–50, 1979.

[17] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2–3):297–308, August 1985.

[18] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, August 1988.

[19] Jan Krajíček. On the number of steps in proofs. *Annals of Pure and Applied Logic*, 41:153–78, 1989.

[20] Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory.* Cambridge University Press, 1995.

[21] Jan Krajíček. On Frege and Extended Frege proof systems. In *P. Clote, J. Remmel (eds.): Feasible Mathematics II*, pages 284–319. Birkhäuser, Boston, 1995.

[22] Jan Krajíček and Pavel Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic*, 54(3):1063–1079, 1989.

[23] Jan Krajíček and Pavel Pudlák. Quantified propositional calculi and fragments of bounded arithmetic. *Zeitschr. f. Mathematikal Logik u. Grundlagen d. Mathematik*, 36:29–46, 1990.

[24] Jan Krajíček, Pavel Pudlák, and Gaisi Takeuti. Bounded arithmetic and the polynomial hierarchy. *Annals of Pure and Applied Logic*, 52(1–2):143–153, 1991.

[25] Jan Krajíček and Gaisi Takeuti. On bounded Σ_1^1 polynomial induction. In S. R. Buss and P. J. Scott, editors, *FEASMATH: Feasible Mathematics: A Mathematical Sciences Institute Workshop*, pages 259–80. Birkhauser, 1990.

[26] J. Paris and A. Wilkie. Counting problems in bounded arithmetic. In *Methods in Mathematical Logic*, volume 1130 of *LNM*, pages 317–40. Springer-Verlag, 1985.

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

[28] Pavel Pudlák. A note on bounded arithmetic. *Fundamenta Mathematica*, 136:85–9, 1990.

[29] Alexander A. Razborov. An equivalence between second order bounded domain bounded arithmetic and furst order bounded arithmetic. In Peter Clote and Jan Krajíček, editors, *Arithmetic, proof theory and computational complexity*, pages 247–77. Oxford University Press, Oxford, 1993.

[30] Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, August 1997.

[31] Alan Skelley. Relating the PSPACE reasoning power of Boolean programs and quantified Boolean formulas. Master's thesis, University of Toronto, 2000. Available from ECCC in the 'theses' section.

[32] Gaisi Takeuti. RSUV isomorphism. In Peter Clote and Jan Krajíček, editors, *Arithmetic, proof theory and computational complexity*, pages 364–86. Oxford University Press, Oxford, 1993.

[33] D. Zambella. Notes on polynomially bounded arithmetic. *The Journal of Symbolic Logic*, 61(3):942–966, 1996.