Some open problems in bounded arithmetic and propositional proof complexity (research proposal paper)

$Id: alanorp.tex,v 1.5 2002/12/10 04:57:36 alan Exp $ LaTeX'd on January 3, 2005

Alan Skelley

January 3, 2005

1 Introduction

The broad relevance and importance of bounded arithmetic and propositional proof complexity are well appreciated; these subjects are two of three which are interconnected in various and interesting ways, the third subject being computational complexity theory. This latter area is rife with old, well-established open problems and one good way to get at them is by studying the other two, whose problems are different yet connected, and many of which may be tractable.

The purpose of this paper is to discuss some possible avenues of research related to bounded arithmetic and propositional proof complexity. Some of these are specific technical open questions related closely to recent research and are placed in their context. Others are more vague general directions along with some discussion of what fruit may be borne by efforts so directed. Yet others are larger issues which are not expected to be resolved but which should be kept in mind and toward which some incremental progress may be possible.

Some necessary definitions and background will be presented first although this will mostly be kept to a minimum.

2 Definitions and Background

2.1 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. We now have many theories such as Cook's PV [8], the S_2 and T_2 hierarchies of Buss [4], and others. A key definition is that of the syntactic quantifier complexity of a formula:

Definition 2.1 (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.*

We will also refer to Σ_i^B (referred to by some authors as $\Sigma_i^{1,b}$) and $\Sigma_i^{\mathcal{B}}$ which are, respectively, over second- and third-order languages and which count the alternations of the appropriate kind of quantifier. It shall be important whether or not the language contains the smash function for numbers and we shall be sure to make it clear.

Now with these classes in mind, S_2^i and T_2^i are theories (with smash) consisting of BASIC (defining basic properties of the language) and additionally T_2^i has the scheme Σ_i^b-IND:

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

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

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

in each case for every $\phi \in \Sigma_i^b$. U_2^i and V_2^i are similarly defined except over a second-order language with smash and induction on the class of formulas with the appropriate number of alternations of bounded second-order quantifiers.

A crucial definition is that of definability:

Definition 2.2. *Let Φ be a class of formulas, T be a theory of bounded arithmetic and $f : \mathbb{N}^k \to \mathbb{N}$ 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 Propositional Proof Systems

The primary motivation for studying propositional proof systems is the theorem of Cook and Reckhow [10] 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.

Definition 2.3. *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 2.4. *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, $P(g(x)) = Q(x)$.*

The proof systems we shall consider are mainly based on PK, Gentzen's sequent-based system, which is p-equivalent to Frege systems. G_i and G_i^* are quantified versions restricted to i alternations of quantifiers, and the latter is additionally treelike. G has no restriction on quantifiers. We will also discuss BPLK, defined in [19] and based on Boolean Programs, defined in [7].

Definition 2.5 (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 $A_i(\overline{\phi})$. There is no free/bound distinction between variables in the language of Boolean programs.

Definition 2.6 (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 [19] and is expected since evaluating both Boolean programs and quantified Boolean formulas is PSPACE complete:

Theorem 2.7. *BPLK and G are polynomially equivalent.*

Some basic results are that $G_i =_p G^*_{i+1}$ as proof systems for TAUT_i when $i > 0$ [13, 16] and $G^*_1 =_p$ extended Frege [15]. $G_0 =_p G^*_0$ (i.e. Frege is p-equivalent to treelike Frege) [12] but this is unknown for $i > 0$.

2.3 Translations of Bounded Arithmetic into Propositional Logic

An important fact 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 [8] 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 [18] 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 [14] 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 [16] 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 [17] showed a similar relation between U_2^1 and G.

3 Open Problems

3.1 Canonical Proof System For a Complexity Class

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. Perhaps a general technique could be devised to deal with many such classes at once.

3.2 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 [7], 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.

3.3 Questions About the "Weak Fragments" of Theories and Proof Systems

3.3.1 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 [16]; 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 [9].

In the other direction, the KPT witnessing theorem implies that if the S_2 hierarchy collapses then so does the PH. Buss [5] and Zambella [20] 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.

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.

3.3.2 Collapsing weak fragments of G

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.

3.3.3 Lower bounds for G

Conversely, proving either an unconditional lower bound or one conditional on a weaker complexity assumption like P$\neq$PSPACE also would seem not to imply anything since a related complexity collapse may not necessarily be formalizable. Of course, an unconditional lower bound for G-proofs of TAUT would seem to be difficult to obtain but for example a lower bound for G_1 or G_1^* proofs of TAUT_i may be more reasonable.

3.4 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 [6] 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 $1 \leq j < i$.

3.5 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 [6] 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 [11] and in more detail in [1]. It would be interesting either to find propositional proof systems whose witnessing problems were exactly projections of these other local search problem classes, or else to understand what is special about PLS.

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.

3.6 Theories for PSPACE and Above

In the attachment, we present some recent work on a third-order theory W_1^1 for PSPACE. This theory has the advantage of being smash-free and is associated with PSPACE in the same way that V^1 is associated with P, namely with respect to the definable functions of strings. This approach has the benefit of making things much simpler than in the case of Buss-style theories and this is evident in several aspects of the exposition on W_1^1.

In the same attachement we also give a translation of theorems of W_1^1 into families of proofs in BPLK which is much simpler than the analogous translation of U_2^1 into G. The BPLK proofs constructed are much more natural than the case for G since, for example, iterating a function is just a matter of composing function symbols while for quantified Boolean formulas, quantifier tricks must be used. Furthermore, the statement of the translation lemma 8.4 itself is much more natural than the case for G: The lemma states that BPLK can prove that a sequent in the W_1^1 proof can be witnessed by some PSPACE function in the right form. The problem is that the third-order quantifiers (second-order in the case of U_1^1) cannot be expressed in the propositional language. In the case of G this is handled by the cumbersome method of asserting that for any formula (to be substituted in place of the inexpressible quantifier on the left), G can prove that some PSPACE function computes a witness for the inexpressible quantifier on the right. In the case of BPLK, the inexpressible quantifier is simply replaced by a "free" function symbol representing an arbitrary Boolean function.

Some particular open questions arising from these results are:

First, prove the replacement theorem of W_1^1 with **strict** $\Sigma_1^{\mathcal{B}}$ induction. We have been unable to carry this through without slightly stronger induction. Chris Pollett, an expert on such matters, says that although the problem is open, it should be possible.

Second, the idea of a free function symbol suggests possibly adding quantifiers to function symbols. If done correctly, this may lead to proof systems associated with the levels of the nondeterministic exponential time hierarchy analogous to G and the PH. Such proof systems would allow a more direct translation of the theorems of W_1^1, U_2^i, and indeed W_2^i once suitably defined, since now the higher-order quantifiers would be expressible in the propositional language. The translations would be more akin to those of S_2^i and T_2^i into G_i^* and G_i. It would be instructive to see how various bounds on these systems would imply or be implied by bounds on the corresponding subsystems of G.

3.7 Relativized Propositional Proof Systems

There has recently been some local interest in relativized propositional proof systems. One reference is [2] and another is a working paper by Cook. These systems reason about propositional formulas with an oracle, the oracle being indexed by sequences of formulas whose truth values spell out a binary string.

In the formulation of [2], the proof system is relative to a particular oracle, so that theorems such as the following hold:

Theorem 3.1. *For every oracle $\mathcal{A}$, there exists an $\mathcal{A}$-relativized super propositional proof system iff $NP^{\mathcal{A}}$ =co-$NP^{\mathcal{A}}$.*

The authors of that paper give some conditions on complexity classes and show that for some complexity classes, a particular collapse implies the existence of an optimal (ordinary) proof system, and that for other kinds of classes there is an oracle relative to which the collapse occurs and yet there is no optimal (relative to that oracle) propositional proof system. An open problem to investigate in this context would be whether known oracle separations of complexity classes imply lower bounds for particular oracle-relativized proof systems, perhaps along the lines of recent work by Maciel and Pitassi on exploiting computational hardness to prove lower bounds. This problem may be hard, however.

In the formulation of Cook, the oracle is an implicit free variable of exponential size. The satisfiability problem is still in NP and the validity problem is thus in co-NP. Cook also defines a quantified propositional calculus QPC(R) (with only propositional quantifiers) and in this case the satisfiability problem for such formulas is NEXP-complete. There seems to be some connection between polynomial-size proofs in G and exponential sized proofs in QPC(R) which bears further investigation. Other open questions are to translate subtheories of $S_2(R)$ into proofs in the appropriate relativized subsystems of G (i.e. QPC(R)) and to try and use oracle separations of complexity classes to separate subsystems of QPC(R).

3.8 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].

3.9 Consistency Strength

Using the idea of Cook [8], [15], [17] 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. Perhaps relativized versions of these are candidates for separating relativized proof systems as mentioned above.

3.10 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] 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.

[2] Ben-David and Gringauze. On the existence of propositional proof systems and oracle-relativized propositional logic. In *ECCCTR: Electronic Colloquium on Computational Complexity, technical reports*, 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 R. Buss. Relating the bounded arithmetic and polynomial time hierarchies. *Annals of Pure and Applied Logic*, 75(1–2):67–77, 12 September 1995.

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

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

[8] 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.

[9] 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.

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

[11] 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.

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

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

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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