A Conditional Lower Bound for a System of Constant-Depth Proofs with Modular Connectives

Alexis Maciel
Clarkson University
Department of Computer Science
Potsdam, NY 13699-5815, U.S.A.
alexis@clarkson.edu

Toniann Pitassi
University of Toronto
Department of Computer Science
Toronto, Ontario, Canada
toni@cs.toronto.edu

Abstract

It is known that constant-depth Frege proofs of some tautologies require exponential size. No such lower bound result is known for more general proof systems. We consider Sequent Calculus proofs in which formulas can contain modular connectives and only the cut formulas are restricted to be of constant depth. Under a plausible hardness assumption concerning small-depth Boolean circuits, we prove an exponential lower bound for such proofs. We prove this lower bound directly from the computational hardness assumption. By using the same approach, we obtain the following additional results. We provide a much simpler proof of a known (unconditional) lower bound in the case where only conjunctions and disjunctions are allowed. We establish a conditional exponential separation between the power of constant-depth proofs that use different modular connectives. Finally, under a plausible hardness assumption concerning the polynomial-time hierarchy, we show that the hierarchy G_i^* of quantified propositional proof systems does not collapse.

1. Introduction

Restricted proof systems have attracted a lot of attention, in large part due to their role in automated theorem provers. For example, Haken [4] showed that the Pigeonhole Principle, a simple, natural and ubiquitous tautology, requires exponential-size Resolution proofs. This means that any theorem prover that works by constructing a Resolution proof—and that is virtually all propositional theorem provers—will require exponential time to prove the Pigeonhole Principle, no matter how efficient it is at finding a proof.

Various extensions of Resolution have also been investigated. For example, constant-depth Frege proofs were also shown unable to prove the Pigeonhole Principle in subexponential size [1, 11, 14]. And Cutting Planes were shown to have no subexponential proof of a certain basic principle concerning colorings of undirected graphs [6, 15].

To this day, however, no lower bound result is known for any proof system more general than constant-depth Frege. For example, a natural extension of constant-depth Frege is to permit the use of modulo m connectives in the proofs, for some constant m. We call this proof system $\mathrm{ACC}^0[m]$-Frege, because of its relation to the circuit class $\mathrm{ACC}^0[m]$.

In this article, we consider a variant of $\mathrm{ACC}^0[m]$-Frege and prove a lower bound for this proof system under a plausible hardness assumption concerning $\mathrm{ACC}^0[m]$ circuits.

The proofs that we consider are tree-like Sequent Calculus proofs in which formulas contain conjunctions, disjunctions, negations and modulo m connectives of unbounded arity and the cut formulas are restricted to be of constant depth. We call this system $\mathrm{ACC}^0[m]$-PK*. This is a natural proof system that has at least one advantage over the usual definition of constant-depth Frege systems: it is complete for all tautologies, not just constant-depth formulas.

Note that the power of $\mathrm{ACC}^0[m]$-PK* is closely related to the power of $\mathrm{ACC}^0[m]$-Frege: over constant-depth tautologies, the two systems are polynomially equivalent. This means that for any constant-depth tautology, there is a polynomial relation between the size of the smallest $\mathrm{ACC}^0[m]$-PK* proof and the size of the smallest $\mathrm{ACC}^0[m]$-Frege proof.

The hardness assumption concerns $\mathrm{ACC}^0[m]$ circuits. It is known that $\mathrm{ACC}^0[p]$ circuits require exponential size to compute the modulo q function, if p and q are distinct primes. It is also known that $\mathrm{ACC}^0[p]$ circuits of subexponential size can compute this function correctly on at most a $1/2 + o(1)$ fraction of the inputs [16, 17]. In the case of AC^0 circuits, the hardness result is much stronger: AC^0 circuits of depth d and subexponential size can approximate the modulo q function on at most a $1/2 + 1/2^{n^{1/(d+1)}}$ fraction of the inputs [5]. Since modulo p gates are known not to be useful in the (exact) computation of the modulo q function [16], it is natural to conjecture that a stronger hardness results also holds for $\mathrm{ACC}^0[p]$ circuits; that is, that $\mathrm{ACC}^0[p]$ circuits of depth d and subexponential size cannot compute the modulo q function (or some other polynomial-size $\mathrm{ACC}^0[q]$ function) correctly on more than a $1/2 + 1/2^{n^{1/(d+1)}}$ fraction of the inputs. In fact, it also seems likely that if q is a prime that does not divide m, then there is an $\mathrm{ACC}^0[q]$ function that cannot be computed correctly by depth-d $\mathrm{ACC}^0[m]$ circuits of subexponential size on more than a $1/2 + 1/2^{n^{1/(d+1)}}$ fraction of the inputs.

It is under this last assumption that we show that $\mathrm{ACC}^0[m]$-PK* proofs of a variant of the Pigeonhole Principle must be of exponential size, if q is a prime that does not divide m. The tautology is defined by replacing each of the variables of the Pigeonhole Principle by an AND-OR-NOT formula expressing an $\mathrm{ACC}^0[q]$ function Hard_q that is hard to approximate by $\mathrm{ACC}^0[m]$ circuits. Each of these formulas is over a separate subset of the original propositional variables. We call this tautology $\mathrm{PHP}(\mathrm{Hard}_q)$.

As far as we know, this is the first known lower bound result for an extension of constant-depth Frege proofs under an assumption seemingly weaker than NP not closed under complementation. And note that size-s $\mathrm{ACC}^0[m]$-PK* proofs of $\mathrm{PHP}(\mathrm{Hard}_q)$ imply size-s $\mathrm{ACC}^0[m]$-Frege proofs of the Pigeonhole Principle. Therefore, our new lower bound is a necessary condition for a lower bound on the size of $\mathrm{ACC}^0[m]$-Frege of the Pigeonhole Principle.

We now say a word about the approach we use to obtain our lower bound result. The Pigeonhole Principle lower bound for AC^0-Frege was obtained by adapting the technique used to prove that AC^0 circuits require exponential size to compute the parity function [5]. Knowing that when p and q are distinct primes $\mathrm{ACC}^0[p]$ circuits require exponential size to compute the modulo q function [16], it is natural to hope that the technique behind that circuit lower bound might be useful in proving a lower bound for $\mathrm{ACC}^0[p]$-Frege proofs. Unfortunately, attempts at adapting this circuit lower bound technique to prove the corresponding proof complexity lower bound have been unsuccessful, despite considerable effort.

On the other hand, the lower bounds for the Cutting Planes proof system were obtained by using circuit lower bounds directly, not the underlying techniques. This approach relies on the fact that the Cutting Planes proof system has the interpolation property: small Cutting Planes proofs of tautologies of a certain type yield small circuits computing a function related to the tautology. A lower bound on the size of these circuits then implies a lower bound on the size of the proofs. Unfortunately, constant-depth Frege, and all of its extensions, probably do not have the interpolation property, as this would imply that Blum integers can be factored in time 2^{n^ε} for arbitrary small ε [2].

Our lower bound result for $\mathrm{ACC}^0[m]$-PK* is obtained by using the above-mentioned circuit hardness conjecture directly but not through the interpolation method. Instead, we essentially show that the cut formulas, which are $\mathrm{ACC}^0[m]$ formulas, are unable to help the proof figure out the value of the Hard_q formulas contained in the $\mathrm{PHP}(\mathrm{Hard}_q)$ tautology. In some sense, the proof then reduces to a cut-free proof of the Pigeonhole Principle and it is known that such proofs have exponential size.

By using the same approach, we obtain several additional results. First, we apply the lower bound technique AC^0-PK*. This leads to a much simpler proof of the known (unconditional) result that AC^0-PK* proofs of $\mathrm{PHP}(\mathrm{MOD}_2)$ must have exponential size.

Second, we establish a conditional exponential separation between the power of constant-depth proofs that use different modular connectives. In particular, we show that under the assumption that some $\mathrm{ACC}^0[q]$ function cannot be approximated by $\mathrm{ACC}^0[m]$ circuits, there exists a tautology that has quasipolynomial-size $\mathrm{ACC}^0[q]$-PK* proofs, but that requires exponential-size $\mathrm{ACC}^0[m]$-PK* proofs.

Finally, we apply our approach to Sequent Calculus style proofs systems for quantified Boolean formulas. The system G introduced by Krajíček and Pudlák [9] and given in its present form by Cook and Morioka [3], is a proof system for reasoning about quantified Boolean formulas. The system G_i^* is a subsystem of G obtained by restricting the cut rule to formulas with at most i alternations of quantifiers. For each i, the G_i^* proof system is essentially a nonuniform version of Buss's well-studied bounded arithmetic system S_2^i. We show that the G_i^* hierarchy does not collapse, under a hardness assumption about the polynomial-time hierarchy.

The rest of the article is organized as follows. In Section 2, we provide definitions and background, including a precise definition of the proof systems. In Section 3, we state our main result, the conditional lower bound for $\mathrm{ACC}^0[m]$-PK* proofs. In Sections 4 and 5, we prove the lower bound for the simpler but representative case of AC^0-PK*. The lower bound proof proceeds in two parts. First, in Section 4, we reduce the proof complexity lower bound to a lower bound on the search complexity of a certain problem in a decision tree model. Then, in Section 5, we present the proof of the decision tree lower bound. In Section 6, we prove the separation results for $\mathrm{ACC}^0[m]$-PK* proofs. In Section 7, we present our results concerning the hierarchy G_i^* of quantified propositional proof systems. We conclude, in Section 7, with a discussion of further work.

2. Definitions and Background

2.1. The Propositional Sequent Calculus

First we define the Sequent Calculus propositional proof system for AND, OR, NOT and modular connectives.

The formulas of the system are defined as usual but with connectives of unbounded arity. For example, $\vee(A_1, \ldots, A_n)$ denotes the logical OR of the multiset consisting of $A_1, \ldots, A_n$. Similarly for the AND connective. Thus commutativity of the connectives is implicit. The modular connectives $\oplus_{m,b}$, for $0 \le b < m$, are interpreted to be true if the sum of their arguments is congruent to b modulo m. In what follows, we will omit the m subscript and simply write $\oplus_b$ when there is no confusion possible.

The proof system operates on *sequents*, which are sets of formulas of the form $A_1, \ldots, A_r \to B_1, \ldots, B_t$. The intended meaning of the sequent $\Gamma \to \Delta$ is that the conjunction of the formulas in Γ implies the disjunction of the formulas in Δ.

A proof of a sequent S is normally a tree of sequents such that the root of the tree is S, the leaves of the tree are initial sequents and every non-leaf sequent in the tree follows from its children by one of the rules of inference. A sequent calculus proof can also be a directed acyclic graph (dag) with similar properties.

The *initial sequents* are of the following form: $x \to x$ where x is a propositional variable; $\to \wedge()$; $\vee() \to$; $\to \oplus_0()$; and $\oplus_b() \to$, for $1 \le b < m$.

The rules of inference are as follows. First we have simple structural rules such as weakening (formulas can always be added to the left or to the right of a sequent), contraction (two copies of the same formula on the same side of a sequent can be replaced by one), and permutation (formulas on one side of a sequent can be reordered).

The remaining rules are the cut rule and logical rules that allow us to introduce each connective on both the left side and the right side of sequents. With the cut rule, we can derive $\Gamma \to \Delta$ from $\Gamma, A \to \Delta$ and $\Gamma \to A, \Delta$. The formula A is called the *cut formula*. The logical rules are:

- (Negation-left) From $\Gamma \to A, \Delta$, derive $\neg A, \Gamma \to \Delta$.
- (Negation-right) From $A, \Gamma \to \Delta$, derive $\Gamma \to \neg A, \Delta$.
- (And-left) From $A_1, \wedge(A_2, \ldots, A_n), \Gamma \to \Delta$, derive $\wedge(A_1, \ldots, A_n), \Gamma \to \Delta$.
- (And-right) From $\Gamma \to A_1, \Delta$ and $\Gamma \to \wedge(A_2, \ldots, A_n), \Delta$, derive $\Gamma \to \wedge(A_1, \ldots, A_n), \Delta$.
- (Or-left) From $A_1, \Gamma \to \Delta$ and $\vee(A_2, \ldots, A_n), \Gamma \to \Delta$, derive $\vee(A_1, \ldots, A_n), \Gamma \to \Delta$.
- (Or-right) From $\Gamma \to A_1, \vee(A_2, \ldots, A_n), \Delta$, derive $\Gamma \to \vee(A_1, \ldots, A_n), \Delta$.
- (Mod-left) From $A_1, \oplus_{b-1}(A_2, \ldots, A_n), \Gamma \to \Delta$ and $\oplus_b(A_2, \ldots, A_n), \Gamma \to A_1, \Delta$, derive $\oplus_b(A_1, \ldots, A_n), \Gamma \to \Delta$.
- (Mod-right) From $A_1, \Gamma \to \oplus_{b-1}(A_2, \ldots, A_n), \Delta$ and $\Gamma \to A_1, \oplus_b(A_2, \ldots, A_n), \Delta$, derive $\Gamma \to \oplus_b(A_1, \ldots, A_n), \Delta$.

The size of a proof is the total size of all the sequents it contains.

Definition 1 *Let $F = \{(\Gamma_n \to \Delta_n) : n \in \mathbf{N}\}$ be a family of sequents. Then $P = \{P_n : n \in \mathbf{N}\}$ is a family of* $\mathrm{ACC}^0[m]$-PK* *proofs for F if there is a constant d such that, for every n:*

1. *P_n is a valid (tree-like) proof of $(\Gamma_n \to \Delta_n)$.*
2. *The cut formulas in P_n have depth at most d.*

We say that P is a family of AC^0-PK* *proofs for F if no modular connectives occur in P.*

Constant-depth Frege systems can also be defined in terms of the sequent calculus. An $\mathrm{ACC}^0[m]$-Frege proof is simply a dag-like $\mathrm{ACC}^0[m]$-PK* proof in which every formula has constant-depth. Similarly for AC^0-Frege.

In this article, we will only consider tautologies consisting of AND, OR and NOT connectives. The connectives will have unbounded arity. Because the sequent calculus is cut-free complete, the $\mathrm{ACC}^0[m]$-PK* proof system is complete for all tautologies while $\mathrm{ACC}^0[m]$-Frege is complete for constant-depth tautologies.

The size of a sequent calculus proof is measured in terms of the size of its conclusion. For example, if F is a family of sequents of size t_n, then a polynomial-size $\mathrm{ACC}^0[m]$-PK* proof of F would have size $t_n^{O(1)}$.

The power of $\mathrm{ACC}^0[m]$-PK* is closely related to the power of $\mathrm{ACC}^0[m]$-Frege when we consider only tautologies of constant depth:

Theorem 2 *The size of the smallest* $\mathrm{ACC}^0[m]$-PK* *proof of a constant-depth tautology is polynomially-related to the size of its smallest* $\mathrm{ACC}^0[m]$*-Frege proof. Similarly for* AC^0-PK* *and* AC^0*-Frege.*

Proof All the formulas in an $\mathrm{ACC}^0[m]$-PK* proof must be either subformulas of the conclusion or formulas that will be the target of a cut. Therefore, in an $\mathrm{ACC}^0[m]$-PK* proof of a constant-depth tautology, all the formulas must have constant depth, which implies that the $\mathrm{ACC}^0[m]$-PK* proof is actually an $\mathrm{ACC}^0[m]$-Frege proof.

On the other hand, any $\mathrm{ACC}^0[m]$-Frege proof of size s and depth d can be transformed into a tree-like $\mathrm{ACC}^0[m]$-Frege proof of size $s^{O(1)}$ and depth $d+1$ [7]. Such a proof is a special case of an $\mathrm{ACC}^0[m]$-PK* proof. □

In general, the Pigeonhole Principle says that if $m > n$ and m pigeons are placed into n holes, then one hole must receive more than one pigeon. This tautology can be expressed as the following sequent, which we denote by PHP_n^m:

$$\bigvee_{j=1}^{n} p_{1j}, \ \ldots, \ \bigvee_{j=1}^{n} p_{(n+1)j} \\ \quad \rightarrow \quad p_{11} \wedge p_{21}, \, p_{11} \wedge p_{31}, \, \ldots, \, p_{nn} \wedge p_{(n+1)n},$$

When m is much larger than n, typically when $m \geq 2n$, we refer to this tautology as the Weak Pigeonhole Principle. The case $m = n + 1$ is usually what is meant simply by the Pigeonhole Principle. We will use PHP to denote the corresponding tautology PHP_n^{n+1}.

One consequence of the above relationship between the $\mathrm{ACC}^0[m]$-PK* and $\mathrm{ACC}^0[m]$-Frege proof systems is an exponential lower bound on the size of $\mathrm{ACC}^0[m]$-PK* proofs of the Pigeonhole Principle. This follows from the corresponding lower bound for $\mathrm{ACC}^0[m]$-Frege proofs [1, 11, 14].

In this article, we consider variants of PHP in which the propositional variables are replaced by polynomial-size AND-OR-NOT formulas expressing a particular function f. Each of these formulas is over a separate subset of the original propositional variables. These subsets will always be of equal size. We denote such a tautology by $\mathrm{PHP}(f)$.

Typically, the formula expressing the function f will not be of constant depth. However, assuming f is not a constant function, then the $\mathrm{ACC}^0[m]$-PK* complexity of $\mathrm{PHP}(f)$ is related to the $\mathrm{ACC}^0[m]$-Frege complexity of PHP in the following way: if $\mathrm{PHP}(f)$ has an $\mathrm{ACC}^0[m]$-PK* proof of size s, then PHP has an $\mathrm{ACC}^0[m]$-PK* proof of size s and such a proof is also an $\mathrm{ACC}^0[m]$-Frege proof. Therefore, a lower bound on the size of $\mathrm{ACC}^0[m]$-PK* proofs of $\mathrm{PHP}(f)$ is a necessary condition for a lower bound on the size of $\mathrm{ACC}^0[m]$-Frege proofs of the Pigeonhole Principle.

In the following sections, we will use the symbols 0 and 1 in formulas. These symbols will simply stand for the formulas $x \wedge \neg x$ and $x \vee \neg x$, respectively. In particular, the sequents $0 \rightarrow$ and $\rightarrow 1$ have tree-like, cut-free proofs of constant size.

2.2. Constant-Depth Boolean Circuits

In this article, we will consider the standard Boolean circuit classes AC^0 and $\mathrm{ACC}^0[m]$, for constant m. All these circuits are of constant depth and consist of gates of unbounded fan-in. AC^0 circuits allow only AND, OR and NOT gates. $\mathrm{ACC}^0[m]$ also permit MOD_m gates that output 1 when the sum of their inputs is divisible by m.

It is known that AC^0 and $\mathrm{ACC}^0[p]$ circuits of subexponential size cannot compute the MOD_q function if p and q are distinct primes [5, 16]. It is also known that these circuits cannot approximate this function very well:

Theorem 3 ([5]) *Let C be a depth-d AC^0 circuit of size $2^{n^{1/(d+1)}}$. Then C cannot compute MOD_q correctly on more than a $1/2 + 1/2^{n^{1/(d+1)}}$ fraction of the inputs, for sufficiently large n.*

Theorem 4 ([16, 17]) *Let C be a depth-d $\mathrm{ACC}^0[p^k]$ circuit of size $2^{o(n^{1/2d})}$. Suppose that p and q are distinct primes. Then C cannot compute MOD_q correctly on more than a $1/2 + o(1)$ fraction of the inputs, for sufficiently large n.*

As mentioned in the introduction, it is natural to conjecture that a stronger hardness result also holds for $\mathrm{ACC}^0[p^k]$ and even $\mathrm{ACC}^0[m]$ circuits:

Conjecture 5 *There exists a Hard_q with the following properties. First, Hard_q can be computed by $\mathrm{ACC}^0[q]$ circuits of polynomial size. Second, let C be a depth-d $\mathrm{ACC}^0[m]$ circuit of size $2^{n^{1/(d+1)}}$ and suppose that q is a prime number that does not divide m. Then C cannot compute Hard_p correctly on more than a $1/2+1/2^{n^{1/(d+1)}}$ fraction of the inputs, for sufficiently large n.*

3. Main Result and Overview of Proof

Our main result is that the hardness assumption for $\mathrm{ACC}^0[m]$ circuits implies an exponential lower bound on the size of $\mathrm{ACC}^0[m]$-PK* proofs.

Theorem 6 *Suppose that q is a prime that does not divide m and that Conjecture 5 is true. Consider the tautology $\mathrm{PHP}(\mathrm{Hard}_q)$ with a block size $B = (6n^2)^{d+2}$. Then any depth-d $\mathrm{ACC}^0[m]$-PK* proof of $\mathrm{PHP}(\mathrm{Hard}_q)$ has size at least $2^{\frac{1}{3}N^{1/(2d+6)}}$, for N large enough, where N is the total number of variables in the tautology.*

Note that since the block size B depends on d, we have a different tautology for each depth. We can prove a lower bound with a single tautology for every depth but the lower bound is slightly smaller:

Theorem 7 *Suppose that q is a prime that does not divide m and that Conjecture 5 is true. Let $f(n)$ be unbounded and consider the tautology $\mathrm{PHP}(\mathrm{Hard}_q)$ with a block size $B = n^{f(n)}$. Then any $\mathrm{ACC}^0[m]$-PK* proof of $\mathrm{PHP}(\mathrm{Hard}_q)$ has size at least $2^{N^{1/(f(N)+3)}}$, for N large enough, where N is the total number of variables in the tautology.*

This is no longer an exponential lower bound, but it is still very large and certainly much larger that quasipolynomial. For example, with $f(n) = \log\log n$, we get a lower bound of $2^{N^{1/(\log\log N+3)}}$.

In the next two sections, we will prove a slightly simpler exponential lower bound on the size of AC^0-PK* proofs:

Theorem 8 *Consider the tautology* $\mathrm{PHP}(\mathrm{parity})$ *with a block size* $B = (6n^2)^{d+2}$. *Then any depth-d* AC^0-PK* *proof of* $\mathrm{PHP}(\mathrm{parity})$ *has size at least* $2^{\frac{1}{3}N^{1/(2d+6)}}$, *for N large enough, where N is the total number of variables in the tautology.*

This lower bound is proved in exactly the same way as Theorem 6. It is unconditional because the required hardness result is not a conjecture but an actual result (Theorem 3).

As mentioned in the previous section, the lower bound of Theorem 8 follows from the exponential lower bound on the size of AC^0-Free proofs of the Pigeonhole Principle. However, the proof we present here is much simpler.

The lower bound will be proved in two steps. First, we convert the proof into a decision tree of the same size that solves an associated search problem. Second, we prove a lower bound on the size of that decision tree.

In general, to any tautologically valid sequent $\Gamma \to \Delta$, we can associate a search problem as follows. Given any assignment to the underlying variables of the sequent, the problem is to output either a formula in Γ that is false, or a formula in Δ that is true. In the case of the PHP, this can be viewed as producing a pigeon i that goes nowhere or a hole j that receives more than one pigeon. While this search problem is trivial in most models of computation, we will show that it is hard for a certain type of decision tree.

4. Converting Proofs to Decision Trees

Let $\Gamma \to \Delta$ be a sequent. A decision tree T solving the search problem for $\Gamma \to \Delta$ will be defined as follows. The tree T is rooted and directed and has fan-out 2 (and fan-in 1). The leaves of T are labeled with formulas $f_i \in \Gamma$ or $g_i \in \Delta$. Each non-leaf vertex v is labeled with a formula f and is said to *query* that formula. The two outgoing edges from v are labeled by $f = 0$ and $f = 1$, respectively. For every total truth assignment α to the underlying variables of the sequent, if α is consistent with a path p in T with leaf label f_p, then either f_p is in Γ and α falsifies f_p, or f_p is in Δ and α satisfies f_p. The *size* of T is sum of the sizes of all formulas labeling vertices in T.

Now let $F = \{(\Gamma_n \to \Delta_n) : n \in \mathbf{N}\}$ be a family of tautologically valid sequents, where n denotes the number of distinct variables in the sequent. Let C be a circuit class. A family of C *decision trees* for F is a sequence $\{T_n : n \in \mathbf{N}\}$ of rooted, directed trees that satisfy the following properties:

1. Each T_n is a decision tree that solves the search problem for $\Gamma_n \to \Delta_n$.
2. The queries of T_n are either expressible as C circuits or are subformulas of formulas occurring in $\Gamma_n \to \Delta_n$.
3. If p is a path in T_n, f is a subformula of a formula in $\Gamma_n \to \Delta_n$ and f' is a subformula of f, then f' can appear on path p only if f appeared earlier on that path.

We now show that any small, depth-d AC^0-PK* proof can be converted into a small AC^0_d decision tree that solves the associated search problem. AC^0_d denotes AC^0 circuits of depth d.

Theorem 9 *Let P be a size-s, depth-d* AC^0-PK* *proof of* $\Gamma \to \Delta$. *There exists an* AC^0_d *decision tree T of size at most s that solves the search problem for* $\Gamma \to \Delta$.

Proof The proof is by induction on the number of lines in P. If P has only one line, then $\Gamma \to \Delta$ is an axiom $x \to x$ and thus the decision tree contains a single vertex labeled x. When $x = 0$ the leaf is labeled by $x \in \Gamma$ and when $x = 1$ the leaf is labeled by $x \in \Delta$. For the inductive step, there are several cases.

The first case is when the final inference in P is the cut inference. That is, the final line, $C = \Gamma \to \Delta$, is derived from $A = \Gamma, f \to \Delta$ and $B = \Gamma \to \Delta, f$ by cutting on $f \in C$. By induction, there is a decision tree T_A of the correct size solving the search problem for A, and similarly a decision tree T_B for B. The tree T for C will first query f. The subtree T_A will hang from the edge labeled $f = 1$, and the subtree T_B will hang from the other edge labeled $f = 0$. Consider an assignment α. If $f(\alpha) = 1$, then T_A on assignment α must output a formula in Γ that is false, or a formula in Δ that is true. Similarly, if $f(\alpha) = 0$, then T_B on assignment α must output a formula in Γ that is false, or a formula in Δ that is true. Thus T solves the search problem for C, and is of the required size.

The second case is when the final inference in P is an OR-left. That is, the final line, $C = \Gamma, (f \vee g) \to \Delta$ is derived from $A = \Gamma, f \to \Delta$ and $B = \Gamma, g \to \Delta$. Again, let T_A and T_B be the subtrees guaranteed by the inductive assumption. Modify T_B to obtain T'_B by replacing any leaf labeled by g to the label $(g \vee f)$. The tree T for C will first query f. The tree T_A will hang from the edge labeled $f = 1$ and the tree T'_B will hang from the edge labeled $f = 0$.

Now consider an assignment α. Suppose that $f(\alpha) = 1$. Then T_A on α must output some formula in Γ that is false, or some formula in Δ that is true, because f is satisfied by α. Now suppose that $f(\alpha) = 0$. Then the tree T_B on α will either output a formula in $\Gamma \cup \{f\}$ that is false, or a formula in Δ that is false. Since T'_B replaces the label g by $(g \vee f)$, the tree T'_B on alpha will either output a formula in $\Gamma \cup \{(f \vee g)\}$ that is false, or a formula in Δ that is false. This completes this case.

The AND-right case is very similar to the above OR-left case. All remaining rules take one sequent to one sequent. For these rules, the same decision tree used for the upper sequent can be used for the derived sequent as well. $\square$

5. The Decision Tree Lower Bound

In this section, we show that small decision trees cannot solve the search problem associated with $\mathrm{PHP}(\mathrm{parity})$.

Theorem 10 *Consider the tautology* $\mathrm{PHP}(\mathrm{parity})$ *with a block size* $B = (6n^2)^{d+2}$*. Let* T *be an* AC^0_d *decision tree that solves the associated search problem. Then the size of* T *is at least* $2^{\frac{1}{3}N^{1/(2d+6)}}$*, for* N *large enough, where* N *is the total number of variables in* $\mathrm{PHP}(\mathrm{parity})$.

Recall that the size of T is the total size of the queries in T. In addition, in the case of $\mathrm{PHP}(\mathrm{parity})$, the fact that T is an AC^0_d decision tree implies that the queries of T are either

1. depth-d, AND-OR formulas,
2. P_{ij}'s, or
3. subformulas of the P_{ij}'s with the restriction that a subformula cannot be queried before the P_{ij} that contains it is queried.

Note that if a subformula of a P_{ij} also happens to be a depth-d, AND-OR formula, then it can be queried without restrictions.

We will also derive a lower bound that applies to a single tautology and decision trees with AC^0 queries of any depth:

Theorem 11 *Let* $f(n)$ *be unbounded and consider the tautology* $\mathrm{PHP}(\mathrm{parity})$ *with a block size* $B = n^{f(n)}$*. Let* T *be an* AC^0 *decision tree that solves the associated search problem. Then the size of* T *is at least* $2^{N^{1/(f(N)+3)}}$*, for* N *large enough, where* N *is the total number of variables in* $\mathrm{PHP}(\mathrm{parity})$.

The proof of these lower bounds can be seen as an iterative process that starts at the root of the tree and moves down towards the leaves. The formal proof will be by induction. Before carrying out the complete argument, we will first sketch the initial step of this induction. It is simpler than the general case but will be sufficient to illustrate the main ideas of the proof.

Say that T is an AC^0_d decision tree. Suppose that T is small. Start at the root of T and follow all paths down the decision tree until you hit either a leaf or a node that queries some P_{ij}. This generates a subtree of T. If this subtree has at least 2^n leaves, then we are done: we have shown that T is large. Otherwise, some leaf X_1 of this subtree will be reached by more than a $1/2^n$ fraction of all possible assignments. That is, by more than $2^{(n+1)nB-n}$ assignments.

Note that the nodes on the path to X_1 can only query small, depth-d, AND-OR formulas. This means that the set of assignments leading to X_1 is the set of ones of the conjunction g of these formulas. The formula g is a small, depth-$(d+1)$, AND-OR formula.

Now either X_1 is a leaf of T or it queries some P_{ij}. We will first establish that, in fact, X_1 cannot be a leaf. For the sake of contradiction, suppose that it is. Then X_1 is labeled by either a pigeon or by a hole.

Suppose first that X_1 is labeled by pigeon i. Then all the assignments that lead to X_1, i.e., the set of assignments that satisfy g, should send pigeon i to no hole. However, consider P_{i1}. Since P_{i1} is a parity formula and g is a small, depth-$(d+1)$, AND-OR formula, we know that g approximates P_{i1} very badly. In fact, at least 1/4 of the assignments that satisfy g satisfy P_{i1} and at least 1/4 satisfy $\overline{P_{i1}}$. This implies that at least some of the assignments leading to X_1 send pigeon i to hole 1. Therefore, X_1 cannot be labeled by pigeon i.

Now suppose that X_1 is labeled by hole j. Then at least $1/4^{n+1}$ of the assignments that lead to X_1 satisfy $\overline{P_{1j}} \cap \cdots \cap \overline{P_{(n+1)j}}$. Since $2^{(n+1)nB-n}/4^{n+1} = 2^{(n+1)nB-(3n+2)} > 1$, some of the assignments that lead to X_1 send no pigeon to hole j. Therefore, X_1 cannot be labeled by hole j.

Note how we already used twice the fact that small, depth-$(d+1)$, AND-OR formulas cannot approximate the parity function. We will use this hardness result once more in what follows.

We now know that X_1 cannot be a leaf so it is labeled by some P_{ij}, say $P_{i_1 j_1}$. This implies that X_1 has two children, one for the assignments that send pigeon i_1 to hole j_1 and one for the other assignments. Among all the assignments that lead to X_1, at least $1/4^{n+1}$ send pigeon i_1 to hole j_1 and no other pigeon to hole j_1. These assignments will lead to the left child of X_1. On the other hand, at least $1/4^{n+1}$ of the assignments that lead to X_1 send no pigeon to hole j_1. These assignments lead to the right child of X_1.

Essentially, this means that querying g gives virtually no information about whether pigeon i_1 goes to hole j_1. It should also be clear that at either child of X_1, we have essentially no information about which of the remaining $n-1$ holes gets more than one pigeon or about which of the remaining pigeons goes nowhere. In particular, if i is any pigeon and $j_2 \neq j_1$, then at least 1/4 of the assignments that lead to the left child of X_1 will send pigeon i to hole j_2 and at least 1/4 will not.

So at each child of X_1, we have to solve a search problem similar to the original one but involving one less hole and, on the left side, one less pigeon. Inductively, we will show that these search problems require decision trees of size 2^{n-1}, so that the total size of T is at least 2^n.

This concludes our overview of the first step of the induction. The general inductive step is more complicated since it needs to address two main differences. These differences can be illustrated by comparing the first and second steps of the induction. First, in the initial step of the induction, the decision tree solves the search problem for the entire set of assignments. In the second step, we are dealing with a restricted set of assignments. For example, at the left child of X_1 we will consider only the assignments that satisfy g, send pigeon i_1 to hole j_1 and send no other pigeon to hole j_1.

The second difference concerns the path leading to X_1. In the first step of the induction, only depth-d, AND-OR formulas could be queried on that path. In the second step of the induction, we will identify a node X_2 that will play a role similar to that of X_1. But on the path to X_2, we will find queries to depth-d, AND-OR formulas, a query to $P_{i_1 j_1}$ and, possibly, queries to subformulas of $P_{i_1 j_1}$.

We now give the precise statement of the general step of the induction. Let A_{ij} denote the set of all assignments that send pigeon i to hole j and no other pigeon to hole j. Let A_{0j} denote the set of all assignments that send no pigeon to hole j.

Theorem 12 *Consider the tautology* $\mathrm{PHP}(\text{parity})$ *with a block size* $B \geq (6n^2)^{d+2}$*. Let* T *be an* AC_d^0 *decision tree that solves the associated search problem. Suppose that the size of* T *is at most* 2^n*. Let* $0 \leq k \leq n$ *and let* X_0 *be a node in* T*. Suppose that*

$$R = \{\text{all assignments that lead to } X_0\} \cap A_{i_1 j_1} \cap \cdots \cap A_{i_k j_k}$$

where $j_1, \ldots, j_k$ *are the holes queried along the path to* X_0 *and where each* i_r *is either* 0 *or the pigeon that was queried when hole* j_r *was queried for the first time. If* $|R| \geq 2^{(n+1)nB-5kn}$ *and* n *is large enough, then the subtree rooted at* X_0 *has size at least* 2^{n-k}*.*

It is easy to verify that when X_0 is the root of T, which implies that $k = 0$, this theorem implies both Theorem 10 and Theorem 11.

As indicated earlier, the fact that small, depth-d, AND-OR formulas cannot approximate the parity function will play a central role in the proof of Theorem 12. The key lemma that will be used is as follows:

Lemma 13 *Consider some* P_{ij} *and suppose that* $B \geq (6n^2)^{d+2}$*. Let* R *be the set of assignments that satisfy the conjunction of*

a) a depth-$(d+1)$*,* AND-OR *formula* g *of size at most* $2^{B^{1/(d+2)}}$*, and*

b) subformulas of other P_{ij}*'s.*

If $|R| \geq 2^{(n+1)nB-5n^2}$ *and* n *is large enough, then both* $R \cap P_{ij}$ *and* $R \cap \overline{P_{ij}}$ *have size at least* $|R|/4$*.*

Proof Any assignment to the underlying variables of $\mathrm{PHP}(\text{parity})$ can be written as (α, β) where α is an assignment to the variables *inside* the block of P_{ij} and β is an assignment to the variables *outside* the block of P_{ij}. For any β, let $R_\beta = \{\alpha \mid (\alpha, \beta) \in R\}$. Then $|R_\beta|$ is the number of assignments in R that agree with β, so that $|R| = \sum_\beta |R_\beta|$. Similarly, $|R \cap P_{ij}| = \sum_\beta |R_\beta \cap P_{ij}|$. Therefore, to show that $|R \cap P_{ij}| \leq (3/4)|R|$, which implies that $|R \cap \overline{P_{ij}}| \geq |R|/4$, it is enough to show that each $|R_\beta \cap P_{ij}|$ is sufficiently small.

Suppose that R is the set of ones of $g \wedge S$ where S is the conjunction of the subformulas of other P_{ij}'s referred to in the statement of the lemma. Then

$$R_\beta = \{\alpha \mid (g(\alpha, \beta) = 1) \wedge (S(\alpha, \beta) = 1)\}.$$

Since $S(\alpha, \beta)$ depends only on β, then either R_β is empty or it is the set of ones of the function g_β defined by $g_\beta(\alpha) = g(\alpha, \beta)$, i.e., the restriction of g by β. Note that g_β is also a depth-$(d+1)$, AND-OR formula of size at most $2^{B^{1/(d+2)}}$.

If R_β is nonempty, then g_β agrees with P_{ij} on the set $(R_\beta \cap P_{ij}) \cup (\overline{R_\beta} \cap \overline{P_{ij}})$. By Theorem 3, this means that $|R_\beta \cap P_{ij}| + |\overline{R_\beta} \cap \overline{P_{ij}}| \leq (1/2 + \varepsilon)2^B$, where $\varepsilon = 1/2^{B^{1/(d+2)}}$. Since $|\overline{R_\beta} \cap \overline{P_{ij}}| = (1/2)2^B - |R_\beta \cap \overline{P_{ij}}|$, we have that $|R_\beta \cap P_{ij}| - |R_\beta \cap \overline{P_{ij}}| \leq \varepsilon 2^B$. This implies that $|R_\beta \cap P_{ij}| \leq |R_\beta|/2 + (1/2)\varepsilon 2^B$.

Adding over all β, we get that $|R \cap P_{ij}| \leq |R|/2 + 2^{(n+1)nB-B}(1/2)\varepsilon 2^B$. It is easy to verify that this is no more than $(3/4)|R|$. A similar argument shows that $|R \cap \overline{P_{ij}}| \leq (3/4)|R|$. □

Proof of Theorem 12 Let T_0 be the subtree rooted at X_0. The proof is by induction on k. The base case is when $k = n$ and it is trivial since T_0 contains at least one node.

Now suppose that $0 \leq k \leq n-1$. Start at X_0 and follow all paths down the decision tree until you hit either a leaf or a node that queries some P_{ij} for $j \notin \{j_1, \ldots, j_k\}$. That is, a P_{ij} that corresponds to a hole that is not queried on the path to X_0. This generates a subtree of T_0. If this subtree has at least 2^n leaves, then we are done. Otherwise, some leaf X of this subtree will be reached by more than a $1/2^n$ fraction of all assignments in R. Let R' be the set of assignments that reach X. The size of R' is at least $2^{(n+1)nB-5kn-n}$.

The nodes on the path from the root of T to X can only query depth-d, AND-OR formulas and subformulas of P_{ij}'s with $j \in \{j_1, \ldots, j_k\}$. This implies that R' is the set of assignments in R that satisfy a conjunction of depth-d, AND-OR formulas and subformulas of P_{ij}'s with $j \in \{j_1, \ldots, j_k\}$. But R itself is the set of assignments that satisfy such a conjunction of formulas, since each A_{ij} with $j \in \{j_1, \ldots, j_k\}$ can be defined by P_{ij}'s with $j \in \{j_1, \ldots, j_k\}$. Therefore, R' is the set of assignments that satisfy a conjunction of depth-d, AND-OR formulas and subformulas of P_{ij}'s with $j \in \{j_1, \ldots, j_k\}$. Let g be the conjunction of all these depth-d, AND-OR formulas. The formula g is a depth-$(d+1)$, AND-OR formula of size at most 2^n.

We now establish that X cannot be a leaf. Suppose first that X is labeled by pigeon i. Since $k < n$, let j be some hole not in $\{j_1, \ldots, j_k\}$. Consider P_{ij}. Since $2^n \leq 2^{B^{1/(d+2)}}$, Lemma 13 implies that $R' \cap P_{ij}$ has size at least $|R'|/4$. Since $2^{(n+1)nB-5kn-n-2} \geq 1$, this implies that at least some of the assignments leading to X send pigeon i to hole j. Therefore, X cannot be labeled by pigeon i.

Now suppose that X is labeled by hole j. Because R' is contained in $A_{i_1 j_1} \cap \cdots \cap A_{i_k j_k}$, j cannot be in $\{j_1, \ldots, j_k\}$. Then, by repeated applications of Lemma 13, at least $1/4^{n+1}$ of the assignments in R' satisfy $\overline{P_{1j}} \cap \cdots \cap \overline{P_{(n+1)j}}$. Since $2^{(n+1)nB-5kn-3n-2} \geq 1$, some of the assignments that lead to X send no pigeon to hole j. Therefore, X cannot be labeled by hole j.

So we now know that X must be labeled by some P_{ij} with $j \notin \{j_1, \ldots, j_k\}$. Among all the assignments that lead to X, at least $1/4^{n+1}$ send pigeon i to hole j and no other pigeon to hole j. This set of assignments is $R' \cap A_{ij}$ and it leads to the left child of X. Since $|R' \cap A_{ij}| \geq 2^{(n+1)nB-5kn-n-2(n+1)} \geq 2^{(n+1)nB-5(k+1)n}$, the inductive hypothesis implies that the subtree rooted at the left child of X has a total query size of at least $2^{n-(k+1)}$.

Similarly, at least $1/4^{n+1}$ of the assignments that lead to X send no pigeon to hole j. This implies that the total query size of the subtree rooted at the right child of X is also at least $2^{n-(k+1)}$. Therefore, T_0 has a total query size of at least 2^{n-k}. □

6. A Separation Result for the $\mathrm{ACC}^0[m]$ and AC^0-PK* systems

The lower bound on the size of $\mathrm{ACC}^0[m]$-PK* proofs of $\mathrm{PHP}(\mathrm{Hard}_q)$, for a prime q not dividing m, is interesting in part because it is a necessary step towards a lower bound on the size of $\mathrm{ACC}^0[m]$-Frege proofs of PHP. But by focusing on a different tautology, we can use our lower bound technique to prove a separation result for the $\mathrm{ACC}^0[m]$ and AC^0-PK* systems.

Consider the Weak Pigeonhole Principle with $m = 2n$ pigeons and n holes. We will simply use WPHP to denote the corresponding tautology PHP^{2n}_n. It is known that WPHP has AC^0-Frege proofs of quasipolynomial size [13, 12]. An immediate consequence is that WPHP also has AC^0-PK* proofs of quasipolynomial size.

Using this fact and the lower bound technique of this paper, we use the tautology $\mathrm{WPHP}(\mathrm{MOD}_q)$ to separate $\mathrm{ACC}^0[q]$-PK* from AC^0-PK*.

Up until now, there has been no need to define precisely the AND-OR-NOT formula MOD_q. But now, for the upper bound part of the separation result, we need a precise definition. Let MOD_q be the formula defined recursively as follows:

$$\sum_{i=1}^{n} x_i \equiv 0 \text{ iff } \bigvee_{b=0}^{q-1} \left(\sum_{i=1}^{n/2} x_i \equiv b \wedge \sum_{i=n/2+1}^{n} x_i \equiv q - b \right)$$

Theorem 14 *Consider the tautology* $\mathrm{WPHP}(\mathrm{MOD}_q)$ *with a block size* $B = (6n^2)^{d+2}$.

1. $\mathrm{WPHP}(\mathrm{MOD}_q)$ *has* $\mathrm{ACC}^0[q]$-PK* *proofs of quasipolynomial size.*
2. *Any* AC^0_d-PK* *proof of* $\mathrm{WPHP}(\mathrm{MOD}_q)$ *has size at least* $2^{\frac{1}{3}N^{1/(2d+6)}}$, *for N large enough, where N is the total number of variables in the tautology.*

Proof (Sketch) The lower bound follows simply from the fact that the argument used to prove the lower bound for $\mathrm{PHP}(\mathrm{MOD}_q)$ (Theorem 8) depends only on the number of holes not the number of pigeons.

A quasipolynomial-size $\mathrm{ACC}^0[q]$-PK* proof of $\mathrm{WPHP}(\mathrm{MOD}_q)$ can be constructed in two stages. First, prove that the AND-OR-NEG formula MOD_q is equivalent to a formula consisting of a single $\oplus_{q,0}$ connective. Second, prove $\mathrm{WPHP}(\oplus_{q,0})$ by using essentially the AC^0-PK* proof of WPHP [12]. Finally, prove $\mathrm{WPHP}(\mathrm{MOD}_q)$ from $\mathrm{WPHP}(\oplus_{q,0})$ using the fact that MOD_q is equivalent to $\oplus_{q,0}$. □

We can also prove a conditional separation of $\mathrm{ACC}^0[q]$-PK* from $\mathrm{ACC}^0[m]$-PK* when q is a prime that does not divide m. According to Conjecture 5, let Hard_q be an $\mathrm{ACC}^0[q]$ function that is hard for $\mathrm{ACC}^0[m]$. Then Hard_q can be expressed as an $\mathrm{ACC}^0[q]$ formula F_q. In this formula, replace each of the $\oplus_{q,0}$ connectives by an equivalent AND-OR-NOT MOD_q formula. This produces an AND-OR-NOT formula that expresses the $\mathrm{ACC}^0[q]$ function Hard_q. We will again abuse notation and use Hard_q to denote this formula.

Theorem 15 *Suppose that q is a prime that does not divide* m *and that Conjecture 5 is true. Consider the tautology* $\mathrm{WPHP}(\mathrm{Hard}_q)$ *with a block size* $B = (6n^2)^{d+2}$.

1. $\mathrm{WPHP}(\mathrm{Hard}_q)$ *has* $\mathrm{ACC}^0[q]$-PK* *proofs of quasipolynomial size.*
2. *Any* $\mathrm{ACC}^0_d[m]$-PK* *proof of* $\mathrm{WPHP}(\mathrm{Hard}_q)$ *has size at least* $2^{\frac{1}{3}N^{1/(2d+6)}}$, *for N large enough, where N is the total number of variables in the tautology.*

Proof (Sketch) The lower bound again follows simply from the fact that the argument used to prove the lower bound for $\mathrm{PHP}(\mathrm{Hard}_q)$ (Theorem 6) depends only on the number of holes not the number of pigeons.

A quasipolynomial-size $\mathrm{ACC}^0[q]$-PK* proof of $\mathrm{WPHP}(\mathrm{Hard}_q)$ can be constructed in two stages. First, prove that the AND-OR-NEG formula Hard_q is equivalent to the $\mathrm{ACC}^0[q]$ formula F_q that expresses the function Hard_q. Second, prove $\mathrm{WPHP}(F_q)$ by using essentially the AC^0-PK* proof of WPHP [12]. Finally, prove $\mathrm{WPHP}(\mathrm{Hard}_q)$ from $\mathrm{WPHP}(F_q)$ by using the fact that Hard_q is equivalent to F_q. □

7. Lower Bounds for QBF Proof Systems

Krajíček and Pudlák [9] extended the propositional Sequent Calculus to obtain a proof system for proving validity of quantified Boolean formulas. The version we present here is due to Cook and Morioka [3]. The language of QBF formulas are: Boolean connectives $\vee, \wedge, \neg$, quantifier connectives $\exists, \forall$, parenthesis (and), and variables $p_i, x_i, i \geq 0$. The p-variables denote free variables, and the x-variables are bound variables. A QBF formula is defined inductively as follows: (1) The atomic formulas are (T), (F), (p_i) and (x_i) for every $i \geq 0$; (2) if ϕ and ψ are formulas, then so are $(\phi \vee \psi)$, $(\phi \wedge \psi)$, $(\neg\phi)$, $(\exists x_i \phi)$, $(\forall x_i \phi)$. Both Σ_0^q and Π_0^q denote the set of propositional formulas. For $i \geq 1$, Σ_i^q are the set of QBF formulas that can be written in prenex form with at most $i-1$ quantifier alternations beginning with $\exists$. Similarly, Π_i^q are the QBF formulas with at most $i-1$ quantifier alternations beginning with $\forall$. It is well known that the evaluation problem for Σ_i^q sentences (and Π_i^q sentences) is complete Σ_i^p (Π_i^p), the i^{th} level of the polynomial-time hierarchy. Further it is not hard to see that there exists a proof system in which every valid QBF formula has a polynomial-size proof if and only if $\mathrm{NP} = \mathrm{PSPACE}$. Similarly, for every $i \geq 0$, there exists a proof system in which every valid Σ_i^q formula has a polynomial-size proof if and only if $\mathrm{NP} = \Pi_{i+1}^p$.

The system G for QBF formulas is a Sequent Calculus system obtained by augmenting the propositional version with the following four quantifier-introduction rules, with the additional restriction that the target of every $\forall$-left and $\exists$-right step be quantifier free.

- ($\exists$-left) From $A(p), \Gamma \rightarrow \Delta$, derive $\exists x A(x), \Gamma \rightarrow \Delta$.
- ($\exists$-right) From $\Gamma \rightarrow \Delta, A(B)$, derive $\Gamma \rightarrow \Delta, \exists x A(x)$.
- ($\forall$-left) From $A(B), \Gamma \rightarrow \Delta$, derive $\forall x A(x) \Gamma \rightarrow \Delta$.
- ($\forall$-right) From $\Gamma \rightarrow \Delta, A(p)$, derive $\Gamma \rightarrow \Delta, \forall x A(x)$.

For $i \geq 0$, G_i is G with cuts restricted to $\Sigma_i^q \cup \Pi_i^q$ formulas. G_i^* is the tree-like version of G_i. There is a close connection between the G_i systems and systems of bounded arithmetic; in particular, for each i, G_i^* is the nonuniform analog of S_2^i and G_i is the nonuniform analog of T_2^i, for bounded formulas [8]. It is an important open problem whether the S_i hierarchies or the G_i hierarchies collapse. Krajíček, Pudlák and Takeuti [10] proved that the S_2^i (and T_2^i) hierarchy does not collapse, assuming that the polynomial time hierarchy does not collapse. However, the analogous result for the G_i hierarchy has not been resolved. As an application of our method, we will prove that under a plausible hardness assumption, the G_i hierarchy also does not collapse.

Conjecture 16 *Let $i, j \geq 0$, where $i > j$. Then the (i,j)-QBF hardness conjecture is as follows. There exists a Boolean function $\mathrm{Hard}_{\mathrm{QBF}}^{i,j}$ of n input variables with the following properties. First, for all n, there is a polynomial-size Σ_i^q formula computing $\mathrm{Hard}_{\mathrm{QBF}}^{i,j}$. Secondly, any Σ_j^q formula of size $2^{n^{1/(d+1)}}$ cannot compute $\mathrm{Hard}_{\mathrm{QBF}}^{i,j}$ correctly on more than a $1/2 + 1/2^{polylog(n)}$ fraction of inputs, for sufficiently large n.*

Theorem 17 *If the (i,j)-QBF conjecture is true for $i > j \geq 0$, then G_j^* cannot polynomially simulate G_i^*.*

Proof (sketch) Consider the tautology $\mathrm{WPHP}(\mathrm{Hard}_{\mathrm{QBF}}^{i,j})$ with block size $B = n^5$. First, we will show that $\mathrm{WPHP}(\mathrm{Hard}_{\mathrm{QBF}}^{i,j})$ has G_i^* proofs of quasipolynomial size. As in the proof of the separation results of the previous section, we prove $\mathrm{WPHP}(\mathrm{Hard}_{\mathrm{QBF}}^{i,j})$ by using essentially the AC^0-PK* proof of WPHP [12]. Applying this upper bound to $\mathrm{WPHP}(\mathrm{Hard}_{\mathrm{QBF}}^{i,j})$ gives rise to G_i^* proofs of quasipolynomial size.

Secondly, we need to see that $\mathrm{WPHP}(\mathrm{Hard}_{\mathrm{QBF}}^{i,j})$ requires superpolynomial-size G_j^* proofs. This follows by adapting the lower bound argument from Theorem 6. Namely, given an alleged polynomial-size G_j^* proof of $\mathrm{WPHP}(\mathrm{Hard}_{\mathrm{QBF}}^{i,j})$, we first show how this implies a polynomial-size decision tree for the search problem associated with $\mathrm{WPHP}(\mathrm{Hard}_{\mathrm{QBF}}^{i,j})$. Now the decision tree will be a Σ_j^q decision tree, meaning that the queries will be either Σ_j^q formulas, $P_{i,j}$'s or subformulas of the $P_{i,j}$'s with the restriction that a subformula cannot be queried before the $P_{i,j}$ that contains it is queried. Secondly, we show that no polynomial-size Σ_j^q decision tree can correctly solve the search problem associated with $\mathrm{WPHP}(\mathrm{Hard}_{\mathrm{QBF}}^{i,j})$, under our hardness assumption. As before, we adapt the argument from Theorem 6, using the fact that the lower bound depends only on the number of holes and not the number of pigeons.

□

8. Conclusion

An obvious open problem is to prove our lower bounds in the dag-like case. One way to do this would be to show that the tree-like systems of depth $d+1$, can polynomially simulate the depth d dag-like system. In particular, is it true that AC^0_{d+1}-PK* can polynomially simulate AC^0_d-PK? We can ask the same question for the other proof systems studied in this paper. That is, can $\mathrm{ACC}^0_{d+1}[m]$-PK* polynomially simulate $\mathrm{ACC}^0_d[m]$-PK? Can G^*_{i+1} polynomially simulate G_i? Note that all of these simulations hold if we restrict the depth (or quantifier alternations) of the formula being proved to i.

Secondly, we would like to develop a new, general-purpose method for obtaining AC^0-Frege lower bounds for CNF formulas. For example, can we obtain a top-down strategy for the liar game formulation of AC^0-Frege for the PHP? Toward this end, we would like to know whether inapproximability results are enough to prove lower bounds for CNF formulas. For example, can we reduce the AC^0-Frege lower bound for some CNF formula, to a natural hardness assumption about AC^0, such as the inapproximability of parity by AC^0 circuits? The only known proofs require structural information about AC^0, such as the fact that under a special family of restrictions, an AC^0 function reduces to a local function (a small-depth decision tree, or a function depending on only a constant number of variables).

Thirdly, in our last application we show that the G^*_i hierarchy does not collapse go G^*_1 unless SAT can be approximated by polynomial-size circuits. In contrast, it has been known that the S^i_2 hierarchy does not collapse to S^1_2 unless the polynomial hierarchy collapses. We would like to know how this assumptions compare. In particular, do polynomial size circuits approximating SAT imply the collapse of the polynomial-time hierarchy?

Acknowledgements

Alexis Maciel was supported by NSF grant CCR-9877150. Toniann Pitassi was supported by an NSERC grant. She would like to gratefully acknowledge Maria Bonet for conversations on this subject many years ago.

References

[1] M. Ajtai. The complexity of the pigeonhole principle. In *Proc. 29th Ann. IEEE Symp. on Foundations of Computer Science*, pages 346–355, 1988.

[2] M. Bonet, C. Domingo, R. Gavalda, A. Maciel, and T. Pitassi. Non-automatizability of bounded-depth Frege proofs. In *Proc. 14th Ann. IEEE Conf. on Computational Complexity*, pages 15–23, 1999.

[3] S. Cook and T. Morioka. Quantified Propositional Calculus and a Second-Order Theory for NC1. Manuscript submitted for publication, April 2004. Preliminary version presented at SML, 2003.

[4] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[5] J. Håstad. Almost optimal lower bounds for small depth circuits. In *Proc. 18th Ann. ACM Symp. on Theory of Computing*, pages 6–20, 1986.

[6] A. Haken and S. A. Cook. An exponential lower bound for the size of monotone real circuits. *Journal of Computer and System Sciences*, 58:326–335, 1999.

[7] J. Krajíček. Lower bounds to the size of constant-depth propositional proofs. *Journal of Symbolic Logic*, 59(1):73–86, 1994.

[8] J. Krajíček. *Bounded Arithmetic, Propositional Logic and Computational Complexity* Cambridge University Press, 1995.

[9] J. Krajíček and P. Pudlák. Quantified Propositional Calculi and Fragments of Bounded Arithmetic. *Zeitschrift f. Mathematkal Logik u. Grundlagen d. Mathematik*, 36:29-46, 1990.

[10] J. Krajíček, P. Pudlák and G. Takeuti. Bounded arithmetic and the polynomial hierarchy. *Annals of Pure and Applied Logic*, 52:143–153, 1991.

[11] J. Krajíček, P. Pudlák and A. Woods. Exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle. *Random Structures and Algorithms*, 7:15–39, 1995.

[12] A. Maciel, T. Pitassi and A. Woods. A New Proof of the Weak Pigeonhole Principle. *Journal of Computer Systems Sciences*, 64:843–872, 2002.

[13] J.B. Paris, A.J. Wilkie and A.R. Woods. Provability of the pigeonhole principle and the existence of infinitely many primes. *Journal of Symbolic Logic*, 53:1235–1244, 1988.

[14] T. Pitassi, P. Beame and R. Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 97–140, 1993.

[15] P. Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–998, September 1997.

[16] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. 19th Ann. ACM Symp. on Theory of Computing*, 1987.

[17] R. Smolensky. On Representations by Low-Degree Polynomials. In *Proc. 34th Ann. Symp. on Foundations of Computer Science*, 1993.