Algebraic Propositional Proof Systems

Toniann Pitassi*

Departments of Mathematics and Computer Science
University of Pittsburgh

July 11, 1996

Abstract

This paper introduces algebraic proof systems for the propositional calculus. We present new results concerning the relative efficiency of these systems, and also survey what is currently known. Many open problems are presented.

1 Introduction

A fundamental problem in logic and computer science is understanding the efficiency of propositional proof systems. It has been known for a long time that $NP = coNP$ if and only if there exists an efficient propositional proof system, but despite 25 years of research, this problem is still not resolved. (See [46] for an excellent survey of this area.) The intention of the present article is to introduce a new algebraic approach to this problem. Our proof systems are simpler than classical proof systems, and purely algebraic. It is our hope that by studying proof complexity in this light, that new upper and lower bound techniques may emerge.

The use of the Nullstellensatz for propositional refutations may have been first suggested in a paper by Lovasz in 1982 ([29]). Independently, an explicit algebraic system based on Hilbert's Nullstellensatz was proposed in a later paper by Beame, Impagliazzo, Krajíček, Pitassi and Pudlák([9]) who were motivated by applications to obtaining lower bounds on the lengths of propositional proofs. The basic idea behind algebraic proof systems is as follows. Let $Q_i(\overline{x}) = 0$ be a system of algebraic equations over a fixed field F. By the basic result in algebraic geometry known as Hilbert's Nullstellensatz, the equations do not have a solution in the algebraic closure of F if and only if there exists polynomials $P_i(\overline{x})$ from $F[\overline{x}]$ such that $\sum_i P_i(\overline{x})Q_i(\overline{x}) = 1$. We can think of the polynomials P_i as a *proof* of the unsolvability of the equations Q_i.

This algebraic proof system is appealing because it is simple and nonsyntactic. Moreover, the question of how large a proof must be amounts to asking how many field operations

*Supported in part by NSF NYI grant CCR-9457782.

are required in order to generate the constant polynomial from certain initial polynomials. Moreover, this proof system is very powerful. We prove here that our algebraic system can polynomially-simulate extended Frege proofs, and if our system is polynomially bounded, then the polynomial hierarchy collapses. There are other tight connections between proofs in our algebraic system and classical proof systems.

The organization of the paper is as follows. In Section 2, we define our algebraic proof systems. In Section 3, we prove basic theorems about algebraic proofs and simulation results. In Section 4, we focus our attention on small degree algebraic proofs. Even a good understanding of the degree complexity of algebraic proofs has many applications. As shown in [9], degree lower bounds for certain unsatisfiable formulas imply lower bounds for bounded-depth Frege proofs. Also, degree lower bounds are related to separation results for NP search classes, as shown in [18]. We review these connections, lower bounds as well as lower bound techniques. In Section 5, we discuss a stronger version of small degree algebraic proofs, Gröbner proofs, which were first studied in [16]. We review their results, and discuss connections and applications to Frege lower bounds. Lastly in Section 6, we present several open problems in this area.

2 Algebraic Proof Systems

Let $C = C_1 \wedge C_2 \wedge ... \wedge C_m$ be a propositional formula over $\{x_1, ..., x_n\}$, in conjunctive normal form, where each C_i is a clause of size at most three. Each clause C_i can be converted into an equation, $\overline{C_i} = 1$ over F such that C is unsatisfiable if and only if $\{\overline{C}_1 = 0, ..., \overline{C}_m = 0\}$ has no 0/1 solution. The equations $Q = \{Q_1 = 0, ... Q_R = 0\}$ corresponding to C are: $\{\overline{C}_1 = 0, ..., \overline{C}_m = 0\}$, plus the equations $x^2 - x = 0$ for all variables x. The equations $x^2 - x = 0$ force 0-1 solutions.

We now show how to translate from the basis $\{\vee, \wedge, \neg\}$ to the basis $\{+, \times, 1\}$ over a field F. For a atomic, $t(a) = 1 - a$; $t(\neg x) = 1 - t(x)$; $t(x \vee y) = t(x)t(y)$; and lastly, $t(x \wedge y) = t(\neg(\neg x \vee \neg y) = t(x) + t(y) - t(x)t(y)$. Our translation has the property that for any truth assignment α, and any boolean formula f, f evaluates to 1 under α if and only if $t(f)$ evaluates to 0 under α. In other words, "0" represents true over the new basis.

Here is an example. Let $C = (b \vee a) \wedge (\neg a \vee c) \wedge (\neg b)$. Then $Q = \{Q_1, Q_2, ..., Q_6\}$, where $Q_1 = (1-b)(1-a) = 1 - a - b + ab$, $Q_2 = (a)(1-c) = a - ac$, $Q_3 = b$, $Q_4 = a^2 - a$, $Q_5 = b^2 - b$, $Q_6 = c^2 - c$.

Let C be a conjunctive normal form formula, and let Q be the corresponding equations. Then an *algebraic* refutation for C (over a fixed ring or field) is a set of polynomials, $P = \{P_1, .., P_R\}$ such that $\sum_{i=1}^{R} P_i(x)Q_i(x) = 1$. The degree of the refutation is defined to be the maximum degree of the P_i's. (Another definition of degree is the maximum degree of the polynomials P_iQ_i, but in this paper we will not use this definition.)

It is important to point out now that there is a big difference between the situation where the above polynomial evaluates to 1 over all 0/1 values, versus when the polynomial is identically 1. An algebraic refutation requires that the polynomial be identically 1. The following simple example illustrates this difference. Let C be an arbitrary unsatisfiable conjunctive normal form formula, and let Q the the corresponding set of equations. Then it is always possible obtain a polynomial that always evaluates to 1 by performing the logical AND of the equations in Q. For example, let $C = (x) \wedge (\neg x)$. Then $Q = \{Q_1, Q_2, Q_3\} = \{x, 1-x, x^2-x\}$. The logical AND of the Q's is $Q_1+Q_2+Q_3-Q_1Q_2-Q_2Q_3-Q_1Q_3+Q_1Q_2Q_3$. This simplifies to the polynomial $1 - x + 2x^3 - x^4$. As x ranges over $0, 1$, this polynomial always evaluates to 1, but the polynomial is not the constant 1.

Theorem 1 *The algebraic proof system over any field F is sound and complete for 3CNF formulas.*

Proof Assume that C is has an algebraic refutation, that is there exists P_i such that $\sum_i P_i Q_i = 1$, where the Q_i's are the polynomials corresponding to C. Then it follows that there is no simultaneous solution to the equations $Q_i = 0$, and thus, there is no satisfying truth assignment for C. (Note that soundness holds for any ring.) Our first proof of completeness uses the weak form of Hilbert's Nullstellensatz. Let $Q = \{Q_1, ..., Q_R\}$ be a system of algebraic equations over a field F. Then the weak form of Hilbert's Nullstellensatz states that Q does not have a solution in the algebraic closure of F if and only if 1 is in the ideal generated by $Q_1, .., Q_R$ over the algebraic closure of F. Now we want to show that in fact if there is a linear combination of the Q_i's that equals 1 in the algebraic closure of F, then there must also be a linear combination in F that equals 1. To see this, suppose that we have a degree d linear combination in the algebraic closure of F that is 1. We can set up a system of linear equations over the algebraic closure to find the coefficients of the P_i's. Because the original polynomials (the Q_i's) are all over F, the system of linear equations is also over F, and thus, we can solve for a solution over F. Thus, we have shown that the Q_i's have no solution over F if and only if 1 can be obtained as a linear combination of the Q_i's (over F).

We will also provide two constructive proofs of completeness. The first proof is the analogue of a truth table proof, and the second is a type of tableau proof. A truth table proof can be viewed as a decision tree where each internal node in the tree is labelled with a variable, and the 2 outedges are labelled with that variable and its negation. The tree is a truth table proof if and only if for every path, the partial truth assignment specified by the edges along that path force the CNF formula to false. A tableaux proof can also be viewed as a decision tree, but now where each internal node in the tree is labelled with one of the clauses C_i in the CNF formula, and the outedges are labelled with $\neg C_i$ and l_j for all literals l_j in C_i. The decision tree is a tableau proof if and only if for every path, the partial truth assignment specified by the edges along that path force the CNF formula to false. The size of a decision tree proof (in either case) is the total number of nodes in the tree. In other words, a truth table proof is obtained by trying all possible assignments with possible shortcuts, and a tableau proof is obtained by "multiplying" the CNF formula to DNF form, with possible shortcuts. Let $C = C_1 \wedge C_2 \wedge ... \wedge C_m$ be an unsatisfiable 3CNF formula over $\{x_1, ..., x_n\}$, and let $Q_1, ..., Q_m$ be the corresponding polynomials.

For the truth table proof, we first partition the truth assignments into m classes K_i, $i = 1, ..., m$ such that if $\alpha \in K_i$, then α falsifies clause C_i. For a truth assignment α, let the term corresponding to α be $\Pi_{i=1}^{n}(\gamma(x_i))$, where $\gamma(x_i) = x_i$ if $\alpha(x_i) = 0$, and $\gamma(x_i) = 1 - x_i$ if $\alpha(x_i) = 1$. (The term has value 1 under the assignment α, and otherwise has value 0.) Let $P_i = \sum_{\alpha \in K_i} T_i(\alpha)$, where $T_i(\alpha)$ is the term corresponding to α, where the terms that also appear in Q_i are deleted. Then $\sum P_i Q_i$ is the sum of the 2^n terms, corresponding to all 2^n truth assignments, and this sum is identically 1. Note that the P_i's are each multilinear polynomials, and thus the proof has degree at most n.

For example, let $C = (x_1 \vee \neg x_2)(x_2 \vee x_3)(\neg x_3)(\neg x_1)$; then $Q_1 = (1 - x_1)(x_2)$, $Q_2 = (1 - x_2)(1 - x_3)$, $Q_3 = x_3$, and $Q_4 = x_1$. We will divide the truth assignments into 4 classes as follows: $K_1 = \{010\}$, $K_2 = \{000\}$, $K_3 = \{001, 011, 101, 111\}$, and $K_4 = \{100, 110\}$. Thus, we obtain $\sum_{i=1}^{4}(1 - x_3)[(1 - x_1)(x_2)] + (1 - x_1)[(1 - x_2)(1 - x_3)] + ((1 - x_1)(1 - x_2) + (1 - x_1)(x_2) + (x_1)(1 - x_2) + (x_1)(x_2))[x_3] + ((1 - x_2)(1 - x_3) + x_2(1 - x_3))[x_1]$, which is equal to the polynomial 1.

Another canonical proof is obtained by expanding the original $3CNF$ formula to obtain an equivalent DNF formula. Again let $C = C_1 \wedge C_2 \wedge ... \wedge C_m$ be the original 3CNF formula. We want to obtain the polynomial $t(C_1 \wedge ... \wedge C_m)$ as a linear combination of $Q_1, ..., Q_m$. This polynomial evaluates to 1 on all 0/1 assignments. However, this polynomial is not identically one, because it contains nonlinear terms. Below we show how to add the appropriate linear combination of initial polynomials $(x_i^2 - x_i)$ in order to obtain the 1 polynomial. Note that unlike the previous proof, this one requires the use of the initial polynomials $x^2 - x = 0$. The following lemma shows that it is always possible to add the appropriate linear combination of initial polynomials $(x_i^2 - x_i)$ to get the 1 polynomial.

Lemma 2 *Let Q be a polynomial in $x_1, ..., x_n$ such that Q is equal to 1 over all $0, 1$ assignments to $x_1, ..., x_n$. Then there exists a polynomial S, which is a linear combination of the initial polynomials $(x_i^2 - x_i)$, such that $Q + S = 1$.*

Proof Because Q evaluates to 1 over all $0, 1$ assignments, $Q' = 1 - Q$ evaluates to 0 over all $0, 1$ assignments. We will prove by induction on the number of variables n, that any polynomial (such as Q') evaluating to 0 over all $0, 1$ assignments can be written as a linear combination of the initial polynomials. Then setting $S = 1 - Q$, it follows that $Q + S = 1$, where S is of the desired form. The base case is when there are no variables, and in this case Q' must be identically 0. Now assume that Q' is a polynomial in $x_1, ..., x_n$. By writing Q' as a polynomial in x_n, it is not hard to see that we can write Q' as $Q_0 + x_n Q_1 + (x_n^2 - x_n)Q_2$, where Q_0 and Q_1 are polynomials in $x_1, ..., x_{n-1}$. Now Q_0 must be zero on all $0, 1$ assignments, and likewise Q_1 must be zero on all $0, 1$ assignments. Thus by applying the induction hypothesis to both Q_0 and to Q_1, it follows that Q' can be written as a linear combination of the initial polynomials. It is interesting to note that S will be of exponential size in the worst case.

This completes the proof of the lemma, and also completes our second constructive completeness proof.

Note that although we are restricting our attention to 3CNF formulas, more general formulas can be handled by introducing new, intermediate formulas. To simplify our exposition, we will assume in this paper that all formulas are in 3CNF form.

3 Algebraic Size Complexity

Perhaps the most general and natural definition of proof size is the algebraic size complexity of the proof.

Definition An algebraic circuit over a field F is a directed, acyclic graph of fanin 2, and fanout 2, and where each intermediate vertex is labelled with a field operation, $+$, $-$, or $*$, and where the leaf vertices are labelled with variables $x_1, ..., x_n$ and with elements from the field F. An algebraic circuit, C over $x_1, .., x_n$ computes a rational function in the obvious way. The *size* of the circuit is the number of intermediate vertices in the circuit. The *depth* of the circuit is the maximum height of the underlying directed acyclic graph. An algebraic circuit is an algebraic formula if each vertex has fanout 1. An *unbounded fan-in* algebraic circuit over F is an algebraic circuit where each intermediate vertex can have arbitrary fanin. The *size* of an unbounded fan-in algebraic circuit is the total number of edges in the circuit, and the *depth* is the maximum height of the underlying directed acyclic graph.

Definition The algebraic proof system over F is *polynomially-bounded* if for every unsatisfiable 3CNF formula, f, there exists polynomials $Q_1, ..., Q_m$, and $P_1, ..., P_m$ such that: (1) The polynomials associated with f are $Q = \{Q_1, .., Q_m\}$; (2) $\sum_{i=1}^{m} P_i Q_i = 1$; (3) Each P_i can be computed by an algebraic circuit over F in polynomial size in $|f|$.

Definition Let $L \subseteq \Sigma^*$, where Σ is a finite alphabet, and Σ^* denotes all finite strings over Σ. (Typically, L encodes either the set of all tautological formulas, or the set of all unsatisfiable formulas.) Then a Cook-Reckhow proof system for L is a function $f : \Sigma^* \rightarrow L$, where f is an onto, polynomial-time computable function. A Cook-Reckhow proof system, f, is polynomially bounded if there is a polynomial $p(n)$ such that for all $y \in L$, there is an $x \in \Sigma^*$ such that $y = f(x)$ and $|x|$ (the length of x) is at most $p(|y|)$.

A key property of a Cook-Reckhow proof system is that, given an alleged proof, there is an efficient method for checking whether or not it really is a proof. For most standard, axiomatic proof systems (Extended Frege, Frege, even ZFC), there is actually a very efficient method for checking whether or not it is really a proof. This property leads to the following theorem.

Theorem 3 ([19]) *There exists a polynomially-bounded Cook-Reckhow propositional proof system if and only of $NP = coNP$.*

The above theorem does not appear to hold for algebraic proofs because there is no known deterministic polynomial time algorithm to check whether or not a polynomial is identically

1, even in the case of finite fields. (In other words, there is no efficient procedure to check that it is a proof.) Nonetheless, the probabilistic polynomial-time algorithm due to Schwartz allows us to prove that if algebraic proofs are polynomially-bounded, then the polynomial hierarchy collapses.

Theorem 4 *For any prime p, if the algebraic proof system over Z_p is polynomially-bounded, then $PH = \Sigma_2^p$.*

Proof We will first show that there is a *coRP* algorithm for testing whether or not an algebraic proof is valid. The algorithm is essentially due to Schwartz [43]. Let P, Q be the input; that is, P and Q are sequences of polynomials (represented by algebraic circuits) over $x_1, ..., x_n$, and we want to test whether or not $R = \sum P_i Q_i - 1$ is identically zero over Z_p. The idea will be to randomly choose elements from a sufficiently large extension field of Z_p. If R is not zero, then the number of zeroes of R will be small relative to the size of the extension field, so with high probability, we will find one by selecting at random. More precisely, let $r(n)$ denote the algebraic circuit size of R, and let $d = p^{r(n)}$. Note that d is an upper bound on the total degree of R. Let $s = \log_p(4d)$, and let $Z_p(\theta)$ be the extension field of Z_p, where we add θ, the s^{th} root of unity. This field has p^s many elements, each of which can be represented as a degree $s-1$ polynomial in θ. The size of our extension field is therefore greater than $4d$. Now if R is not identically zero, then the fraction of elements of $Z_p(\theta)$ which are zeroes of R is at most $d/|Z_p(\theta)|$, which is at most $1/4$. (See, for example [43], Corollary 1.) Our algorithm will be to select elements of $Z_p(\theta)$ at random for $x_1, ..., x_n$ and output 0 if and only if R evaluates to 1. (Of course, one has to be somewhat careful in implementing this algorithm, to be sure to represent the elements of $Z_p(\theta)$ as degree $s-1$ polynomials, and then to carry out the evaluation of R on a randomly chosen value in polynomial time.) Clearly if R is identically zero, the algorithm will output 1 with probability 1. On the other hand, if R is not zero, then with probability at least $3/4$, the algorithm will output 0.

Now suppose that the algebraic proof system over Z_p is polynomially-bounded. We will now describe a randomized NP-type algorithm that solves a $coNP$ hard problem. More specifically, the algorithm will take as input a $3CNF$ formula over $x_1, ..., x_n$, and in polynomial time, output 1 if there exists some NP guess such that all probabilistic paths consistent with that NP guess output 0, and the algorithm outputs 1 if for every NP guess, almost all probabilistic paths consistent with that NP guess output 1. We will call such an algorithm an NP-coRP algorithm. The algorithm is simple: given a $3CNF$ formula, the algorithm first guesses a polynomial size algebraic proof over Z_p, and then runs the above *coRP* algorithm to test whether it really is a valid proof. If the algebraic proof system is polynomial bounded, the algorithm is correct.

We will now show that the existence of this NP-coRP algorithm for the unsatisfiability problem implies that $\Pi_2^b = \Sigma_2^p$, and thus the polynomial-time hierarchy collapses to Σ_2^p. Consider a Σ_2^p decision problem $B(x) = \exists v \forall w B'(x, v, w)$, where B' is polynomial-time computable. By the previous algorithm, it follows that this Σ_2^p problem can also be computed by an NP-coRP algorithm. Thus, there exists a polynomial-time predicate, $A(x, q, r)$, where

q indicates the existential variables, and r indicates the random variables and such that the following holds: If $B(a) = 1$, then there exists an assignment α to q such that for all assignments β to r, $A(a, \alpha, \beta) = 0$ and if $B(a) = 0$ then for all assignments α to q, for all but a 1/4-fraction of assignments, β, to r, $A(a, \alpha, \beta) = 1$. We will now show that our problem, $B(x)$, computable by an NP-coRP algorithm, can also be computed by a Π_2^p algorithm; the Π_2^p algorithm will be obtained by moving the probabilistically quantified r to the other side of the existentially quantified q. That is, our Π_2^p algorithm has x is the input variables, q as the (same) sequence of existential variables, but now r' is a sequence of $O(\log n)$ many values of random variables r. The Π_2^p algorithm outputs 1 on $x = a$ if and only if for all assignments, β', to r', there exists an assignment α to q such that $A(a, \alpha, \beta_i) = 0$ for all $\beta_i \in \beta$. Because A is polynomial-time computable, the above algorithm is in Π_2^p. To see that it is correct, assume first that $B(a) = 1$. Then by the NP-coRP algorithm, there exists an assignment α to q such that all assignments β to r give $A(a, \alpha, \beta) = 0$, and thus, the above Π_2^p algorithm also outputs 1. On the other hand, if $B(a) = 0$, then for every assignment α to q, almost all assignments β to r are such that $A(a, \alpha, \beta) = 1$. Thus by an averaging argument, there exists a sequence of $O(\log n)$ many assignments, $\beta_1, ..., \beta_{O(\log n)}$, such that for any α, at least one of the β_i's gives $A(a, \alpha, \beta_i) = 1$. Therefore, we have shown that if our algebraic proof system over Z_p is polynomially bounded, then any Π_2^p decision problem $B(x)$ can be computed by a Σ_2^p algorithm, and thus the polynomial hierarchy collapses.

It should also be possible to extend the above theorem to hold for any finite field, and also for the field $\mathcal{Q}$, of rational numbers. We can also show that algebraic proof systems are at least as powerful as Extended Frege, as is evidenced by the following theorem.

Theorem 5 *For any commutative ring R, Frege proofs (and Extended Frege proofs) can be polynomially simulated by algebraic proofs with polynomial size.*

Proof

We will prove the first part of the above theorem by formulating a Frege proof in the Sequent Calculus, which we describe below. A *cedent* is any sequence $F_1, \ldots, F_n$ of formulas separated by commas. Cedents are sometimes designated by $\Gamma, \Delta, \ldots$ (capital Greek letters). A *sequent* is given by $\Gamma \rightarrow \Delta$, where Γ, Δ are arbitrary cedents. The size [resp. depth] of a cedent $F_1, \ldots, F_n$ is $\sum_{1 \leq i \leq n} size(F_i)$ [resp. $\max_{1 \leq i \leq n}(depth(F_i))$]. The size [resp. depth] of a sequent $\Gamma \vdash \Delta$ is $size(\Gamma) + size(\Delta)$ [resp. $\max(depth(\Gamma), depth(\Delta))$]. The intended interpretation of the sequent $\Gamma \rightarrow \Delta$ is that the conjunction of the formulas in Γ implies the disjunction of the formulas in Δ. An *initial sequent* is of the following form: $F \rightarrow F$ where F is any formula over the connectives: unbounded fanin $\wedge$, unbounded fanin $\vee$, and $\neg$.

The rules of inference are as follows.

structural rules

weakening

weak left:

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

weak right:

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

contraction

contract left:

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

contract right:

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

permutation

permute left:

$$\frac{\Gamma, A, B, \Delta \longrightarrow \Gamma'}{\Gamma, B, A, \Delta \longrightarrow \Gamma'}$$

permute right:

$$\frac{\Gamma \longrightarrow \Gamma', A, B, \Delta'}{\Gamma \longrightarrow \Gamma', B, A, \Delta'}$$

cut rule

$$\frac{\Gamma, A \longrightarrow \Delta \qquad \Gamma' \longrightarrow A, \Delta'}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}$$

logical rules

$\neg$-left:

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

$\neg$-right:

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

$\vee$-left:

$$\frac{A, \Gamma \longrightarrow \Delta \qquad B, \Gamma \longrightarrow \Delta}{A \vee B, \Gamma \longrightarrow \Delta}$$

$\vee$-right:

$$\frac{\Gamma \longrightarrow A, \Delta}{\Gamma \longrightarrow A \vee B, \Delta}$$

$\wedge$-left:

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

$\wedge$-right:

$$\frac{\Gamma \longrightarrow A, \Delta \qquad \Gamma \longrightarrow B, \Delta}{\Gamma \longrightarrow A \wedge B, \Delta}$$

A refutation of a CNF formula $C = C_1 \wedge \ldots \wedge C_m$ in the sequent calculus is a sequence of sequents, $S_1, \ldots, S_m$, such that each sequent is either an instance of the axiom, or follows from previous sequents by an inference rule, and the final sequent is $C_1, \ldots, C_m \rightarrow$. An extended refutation of C is a refutation of $\Gamma, C_1, \ldots, C_m \rightarrow$, where the formulas in Γ are extension formulas. The size of a proof/refutation is the total number of connectives in the proof; the length is the total number of sequents in the proof. It is a known theorem of Cook and Reckhow that there is a fixed polynomial $p(n)$ such that any Frege refutation of C of size s can be converted into a sequent calculus proof of C size at most $p(s)$, and any extended Frege proof of size s can be converted into an extended sequent calculus proof of C of size at most $p(s)$.

The theorem is proven formally by the length of the original derivation. We prove that for any line $A_1, \ldots, A_k \rightarrow B_1, .., B_l$ in the Frege proof, that there exists a linear combination $\sum_{i=1}^{k} P_{A_i} Q_{A_i} + \sum_{i=1}^{l} P_{\neg B_i}(1 - Q_{B_i}) = 1$. Furthermore, if this line has a Frege proof of size s, then the polynomials in the algebraic proof have formula size at most $p(s)$. If the proof consists of only one line, then it must be an instance of the axiom $A \rightarrow A$. The corresponding linear combination is $Q_A + Q_{\neg A} = Q_A + 1 - Q_A$, which is equal to 1. Now assume that the theorem holds for proofs of length l, and now let us consider the last line in a length $l+1$ proof. There are several cases depending on the inference rule used to derive the last line. Weakening is trivial since the same linear combination will work. For the other cases, we will need the fact that for any algebraic circuit or formula Q, it is possible to derive $Q^2 - Q$ as a linear combination of the initial polynomials $x_i^2 - x_i$. This is straightforward to prove by induction on the complexity of Q, and the size of the linear combination will be polynomially related to the size of Q. The other cases are handled below.

Case 1. ($\wedge$-Left) We want to show that if $P_A Q_A + \sum_i P_{C_i} Q_{C_i} = 1$, then we can derive $P'(Q_A + Q_B - Q_A Q_B) + \sum_i P'_{C_i} Q_{C_i} = 1$. Let $P' = P_A Q_A$ and let $P'_{C_i} = P_{C_i}$. Then we have $P'(Q_A + Q_B - Q_A Q_B) + \sum_i P'_{C_i} Q_{C_i} + (P_A Q_B - P_A)[Q_A^2 - Q_A] = P_A Q_A^2 + P_A Q_A Q_B - P_A Q_A^2 Q_B + \sum_i P_{C_i} Q_{C_i} + P_A Q_A^2 Q_B - P_A Q_A Q_B - P_A Q_A^2 + P_A Q_A = P_A Q_A + \sum_i P_{C_i} Q_{C_i} = 1$. Note that the degree increases by the degree of Q_A, and the circuit size increases by an additive constant.

Case 2. ($\vee$-Left) We want to show that if $P_A Q_A + \sum_i P_{C_i} Q_{C_i} = 1$ and $P_B Q_B + \sum_i P'_{C_i} Q_{C_i} = 1$, then we can derive $P'(Q_A Q_B) + \sum_i P''_{C_i} Q_{C_i} = 1$. Let $P' = P_A P_B$ and let $P''_{C_i} = P'_{C_i} + P_{C_i} Q_B P_B$. Then we have $P'(Q_A Q_B) + \sum_i P''_{C_i} Q_{C_i} = P_A P_B Q_A Q_B + \sum_i (P'_{C_i} + P_{C_i} Q_B P_B) Q_{C_i} = P_A P_B Q_A Q_B + \sum_i P_{C_i} Q_B P_B Q_{C_i} + 1 - P_B Q_B = P_B Q_B (P_A Q_A + \sum_i P_{C_i} Q_{C_i} - 1) + 1 = 1$. Again, the circuit size increases by an additive constant.

Case 3. (Cut) We want to show that from $P_A Q_A + \sum_i P_{C_i} Q_{C_i} = 1$ and $P_{\neg A}(1 - Q_A) + \sum_i P'_{C_i} Q_{C_i} = 1$, that we can derive $\sum_i P''_{C_i} Q_{C_i} = 1$. Let $P''_{C_i} = P_{\neg A}(1 - Q_A) P_{C_i} + P'_{C_i}$. Then we have $\sum_i P''_{C_i} Q_{C_i} = P_{\neg A}(1 - Q_A) \sum_i P_{C_i} Q_{C_i} + \sum_i P'_{C_i} Q_{C_i} = P_{\neg A}(1 - Q_A)[1 - P_A Q_A] + 1 - P_{\neg A}(1 - Q_A) = P_{\neg A} - P_A P_{\neg A} Q_A - P_{\neg A} Q_A + P_{\neg A} P_A Q_A^2 - P_{\neg A} + P_{\neg A} Q_A + 1 = 1$. Again, the circuit size increases by an additive constant.

Case 4. ($\vee$-Right) This case is the same as case 1, since we want to show that from $P_{\neg A}(1 - Q_A) + \sum_i P_{C_i} Q_{C_i} = 1$ we can derive $P'((1 - Q_A) + (1 - Q_B) + (1 - Q_A)(1 - Q_B)) + \sum_i P'_{C_i} Q_{C_i} = 1$.

Case 5. ($\wedge$-Right) This is the same as case 2, since want to show that from $P_{\neg A}(1 - Q_A) + \sum_i P_{C_i} Q_{C_i} = 1$ and $P_{\neg B}(1 - Q_B) + \sum_i P'_{C_i} Q_{C_i} = 1$, we can derive $P'((1 - Q_A)(1 - Q_B) + \sum_i P''_{C_i} Q_{C_i} = 1$.

In all cases, the algebraic size of our algebraic proof is polynomial in the number of lines in the Sequent calculus proof. The simulation of an extended Frege proof by an algebraic proof can be shown as follows. Statman ([45]) has proven that the size of an extended Frege proof is polynomially related to the minimum number of lines in a Frege proof. Because our simulation above constructs an algebraic proof which is polynomially bounded by the number of lines (steps) in the Frege proof, it follows that Extended Frege proofs can also be polynomially simulated by algebraic proofs.

It is also interesting to examine the complexity of the witnessing polynomials P_i in the algebraic proof. If we start with a sequent calculus proof that doesn't involve the cut rule, then the witnessing polynomials in the simulating algebraic proof will just be simple combinations of the initial polynomials (the Q_i's). On the other hand, each application of the cut rule gives rise to a new witnessing polynomial.

4 Degree Complexity

Definition Let $C = C_1 \wedge .. \wedge C_m$ be a 3CNF formula, and let Q be the corresponding set of equations. Then $\mathcal{P} = \{P_1, .., P_k\}$ is a *degree d* algebraic refutation of C if and only if: (1) $\sum_i P_i Q_i = 1$; and (2) Each P_i has degree at most d.

Small degree algebraic proofs are important for several reasons. First, they correspond to a depth 1 Frege system with mod 2 gates. In other words, a constant degree algebraic refutation is a very rudimentary proof system which has the power of mod 2 reasoning. Secondly, constant degree algebraic proofs have the nice computational property that a constant degree algebraic proof can be found in polynomial time, if one exists. (Assume that

$\sum_i P_i Q_i = 1$, and the Q_i's have degree at most d. Then the total number of monomials in the Q_i's is bounded by a polynomial in d, and therefore, we can set up a system of linear equations (one for each monomial) and solve for the coefficient values in polynomial time.

Because it is simple to verify in polynomial-time whether or not $\sum_i P_i Q_i = 1$, when the Q_i's are given in standard form, constant-degree algebraic proofs can be verified in polynomial time. Thus, there exist a family of unsatisfiable formulas requiring nonconstant degree algebraic proofs, unless NP equals $coNP$. This raises the question of how hard it is to prove degree lower bounds. In this section, we will survey what is currently known, and connections between these lower bounds and results in propositional proof theory and complexity theory.

4.1 Upper bounds

The onto version of the propositional pigeonhole principle states that there is no 1-1, onto map from $n+1$ to n. This can be expressed by the following equations, with underlying variables $P_{i,j}$, $i \leq n+1$, $j \leq n$: (1) $P_{i,1} + ... + P_{i,n} - 1 = 0$, for all $i \leq n+1$; (2) $P_{1,j} + ... + P_{n+1,j} - 1 = 0$, for all $j \leq n$; and (3) $P_{i,k} P_{j,k} = 0$, for all $i, j \leq n+1$, $k \leq n$. For each n, let the above set of equations be denoted by $\neg PHP^n_{onto}$. For each n, there is a degree 0 algebraic refutation over any field of $\neg PHP^n_{onto}$. The proof is obtained by adding together all of the above equations in (1) and subtracting all of the above equations in (2). Each variable will cancel because it occurs once positively in (1) and once negatively in (2), and we are left with $n+1-n=1$.

The more general version of the propositional pigeonhole principle states that there is no 1-1 map from $n+1$ to n. For each n, the general pigeonhole principle can be expressed by equations (1) and (3) above, and is denoted by $\neg PHP^n$. In [18], it is shown that for all sufficiently large n, $\neg PHP^n$ requires degree $\sqrt{n}$ algebraic refutations over any field. The best upper bound is a degree n algebraic refutation, and we conjecture that this is optimal.

An open problem is to show that algebraic refutations of $\neg PHP^n$ over any field requires degree $O(n)$, for sufficiently large n.

4.2 Lower bounds

The first nontrivial lower bound on the degree of algebraic proofs appeared in [9]. The mod q counting principle, Mod^q_n, states that there is no way to partition a set of size n into equivalence classes, each of size exactly q. For each n, the negation of this principle ($\neg Mod^q_n$) can be expressed by the following equations, with underlying variables X_e, $e \subseteq [1, .., m]$, $|e| = q$, $m = pn+1$:

1. $\sum_{e,\, i \in e} X_e - 1 = 0$, for all $i \leq m$;
2. $X_e X_f = 0$, for all e, f, $e \cap f \neq 0$.

Theorem 6 *Let F be the field GF_p, where p is a prime power, not q. Then for sufficiently large n, any algebraic refutation of $\neg Mod^q_n$ over F requires nonconstant degree.*

The above is proven nonconstructively using Ramsey theory. Very recently, [13] substantially improved the above theorem.

Theorem 7 ([13]) *Let F be the field GF_p, where p is a prime power, not q. Then for infinitely many n, any algebraic refutation of $\neg Mod_n^q$ over F requires degree $n^{\frac{1}{\log(p+q)}}$.*

As mentioned above, there are also very good lower bounds for algebraic refutations of the pigeonhole principle.

Theorem 8 ([18]) *For any $m > n$ and any field F, any algebraic refutation of PHP_n^m over F requires degree at least $n^{1/2}$.*

The induction principle on n Boolean variables, $x_1, ..., x_n$, states that if $x_1 = 1$ and $x_n = 0$, then there must be some point i, $1 \leq i \leq n-1$ such that $x_i = 1$ and $x_{i+1} = 0$. The negation of this principle is formalized by the following equations, IND_n, over an arbitrary field k:

1. $1 - x_1 = 0$;
2. $x_i(1 - x_{i+1}) = 0$ for all $1 \leq i \leq n-1$;
3. $x_n = 0$; and
4. $x_i^2 - x_i = 0$ for all $1 \leq i \leq n$.

Buss and Pitassi [12] show that the induction principle requires degree $O(logn)$, and this bound is tight.

Theorem 9 ([12]) *Any algebraic refutation of IND_n over any field requires degree $O(\log n)$.*

The best lower bound known is for a variant of the strong induction principle, the homesitting principle. The negation of this principle states that if there are persons $[0, 1, .., n]$ and homes $[1, .., n]$, then there is a map from persons to homes such that each person i is sitting in some home j where $j \geq i$, and for all i, if person i is at home (sitting at home i), then no other person can be sitting at home i.

Theorem 10 ([16, 11]) *Any algebraic refutation of the homesitting principle over any field requires degree $O(n)$.*

4.3 The Design Method

In this section we review the primary method that has been used to obtain the above degree lower bounds.

Let R be any commutative ring, and let $\mathcal{Q} = \{Q_1, .., Q_m\}$ be a set of unsolvable equations of degree at most 3 over $R[x_1, ..., x_n]$, where m is $n^{O(1)}$. We want to show that there is no degree d set of polynomials $P_1, ..., P_m$ such that $\sum_i P_i Q_i = 1$. Assume for sake of contradiction that degree d P_i's do exist. Write P_i as $\sum_m a^i_m X_m$, where $m \in \{0,1\}^n$, X_m is the corresponding monomial, and a^i_m is the coefficient in front of that monomial in P_i. Because the total number of monomials in the P_i's is bounded by $n^{O(d)}$, we can write a system of linear equations with the coefficients a^i_m as variables such that the system of linear equations has a solution if and only if such P_i's exist. In particular, the condition $\sum_i P_i Q_i = 1$ can be specified by a system of linear equations in the a^m_i's where for each nonempty monomial m of degree at most $d+3$, we have one equation specifying that the sum of all coefficients in front of this monomial must be 0, and for the empty monomial, we have one equation specifying that the sum of all coefficients in front of the empty monomial must be 1.

Now by weak duality, if we can find a linear combination of the equations such that the left-hand-side of the linear combination is 0, then there can be no solution. (Because the total sum of the right-hand-sides of the equations is 1.) Conversely, if R is a field, then we get the converse direction as well. The name *design* refers to the linear combination of the equations witnessing the fact that the equations can have no solution; because of the structure of the original Q_i's, the properties required of the linear combination can often be seen to be equivalent to the existence of a particular type of combinatorial property, and thus it is called a design.

Here we will give a simple example of the design method, applied to show that the Mod q principle has no solutions over Q. This example is due to Pudlák. The Mod q principle, Mod^n_q, has underlying variables X_e, $e \subseteq [1,n]$, $|e| = q$. Recall that the underlying equations for the Mod q principle are as follows:

1. For all $i \in [1,n]$, we have $Q_i = \sum_{e, i \in e} X_e - 1 = 0$.

2. For all $e, f \in [1,n]^3$, where $e \cap f \neq$, we have $Q_{e,f} = X_e X_f = 0$.

Suppose there exists degree d polynomials, $P_1, ...P_m$ over Q such that the linear combination of the P's and the Q's is 1. Then it is not too hard to show that there also exists a solution such that all coefficients of the form a^i_e where $i \in e$ are zero. A matching monomial is a monomial that defines a partial q-partition of the underlying variables. (In other words, where the hyperedges in the monomial do not intersect.) When we multiply out the P's and the Q's, each nonempty matching monomial must cancel out. On the other hand, the number of occurrences of the empty monomial must be one (since the righthand side of the equation is 1). Thus we get the following equations, one for each matching monomial.

1. $\sum_i a^i_\emptyset = 1$

2. for all matching monomials m, $1 \leq |m| \leq d$, we have $\sum_{i \in m} a^i_{m-i} - \sum_{i \notin m} a^i_m = 0$, where $m - i$ is the monomial obtained from m by removing the edge containing i.

Let Eqn_m denote the left-hand-side of the above equation for the matching monomial m. In the above, we only wrote down the linear equations corresponding to monomials that are partial partitions of the variables. In order to show that there is no solution to the above equations (and hence no degree d P_i's), we will find a linear combination of the equations such that the linear combination forces the left-hand-side to 0, while the right-hand-side is nonzero. More specifically, we define a degree d design over Q to be a function f from partial matchings m of size at most d to elements in Q, with the following properties: (1) $f(\emptyset) \neq 0$; and (2) $\sum_{m,|m| \leq d} f(m) Eqn_m = 0$. If we have a degree d design, then there is no solution to the above equations because when we take the linear combination of the left-hand-side of the equations (with coefficients $f(m)$), we get 0, but on the other hand the right-hand-side is nonzero.

There is a simple degree $d = n/q$ design over Q. Namely, for $m = \emptyset$ let $f(m) = s_0 = 1$, and for all m of size exactly $t+1$, $t \geq 0$, let $f(m) = s_t$, where $s_{t+1} = s_t / \binom{n-qt-1}{q-1}$. Note that this design does not work at all over finite fields, where more solutions to the above equations are possible due to cancellation. Still, in a very recent paper [13], n^ϵ degree lower bounds were obtained over finite fields, where ϵ depends on the field.

4.4 The Interpolation Method

In this section, we discuss a less direct method of obtaining degree lower bounds. Let S be any propositional proof system. Let $f = A(x,z) \wedge B(y,z)$ be an unsatisfiable 3CNF formula, where A involves only the variables x and z, and B involves only the variables y and z. Then a feasible interpolant for f is a polynomial-time computable function (or circuit) $Int_f(z)$ with the property that if $Int_f(a) = 1$, then $A(x,a)$ is unsatisfiable, and if $Int_f(a) = 0$, then $B(y,z)$ is unsatisfiable. Mundici [30] has shown that unless $NP \cap coNP \subseteq P/poly$, there are statements f of the above form with no feasible interpolant. We are interested in the following question: If f is of the above form, and if f has a short refutation in S, then does f have a feasible interpolant? If the answer is yes, then we say that S has an effective type (1) interpolation theorem.

S has an effective type (2) interpolation theorem if there exists a polynomial p such that for any unsatisfiable formula of the form $f = A(x) \wedge B(y)$, if f has a size s refutation in S, then either $A(x)$ or $B(y)$ has a size $p(s)$ refutation in S.

Effective interpolation theorems are very important because they give rise to conditional lower bounds for the corresponding proof system. (See the papers [25, 14, 33, 30, 17] for more details.) And if a monotone version of the interpolation theorem holds, then using known lower bound for monotone circuits, it is possible to prove an (unconditional) lower bound for the proof systems. For example, this has been the method used to obtain exponential lower bounds for Resolution and Cutting Planes proofs.

Russell Impagliazzo [23] has observed that effective interpolation theorems exist whenever the proof system can be made deterministic. That is, suppose that there exists a deterministic algorithm for finding a proof of size s, that runs in time polynomial in s. Then the interpolant function is also polynomial-time computable. Namely, output 1 if and only if $A(x, a)$ has a short proof. This algorithm works because if $A(x, a)$ is unsatisfiable and $B(y, a)$ is not, then plugging in a satisfying assignment for B into the short refutation of f yields a short refutation of $A(x, a)$. It follows from this observation that small degree Nullstellensatz refutations has an effective type (1) interpolation.

Recently, Pudlak [34] has obtained a monotone version of the above interpolation theorem. Namely, he has shown that if $f = A(x, z) \wedge B(y, z)$ is monotone and has a small degree Nullstellensatz refutation, then there exists a polynomial size monotone span program for $Int_f(z)$. Then applying recent lower bounds for monotone span programs [6], it is possible to obtain nonconstant degree lower bounds for Nullstellensatz refutations of a certain principle.

4.5 Applications

4.5.1 The Relative complexity of NP search classes

Small degree algebraic refutations have several applications. First, there is a close relationship between NP search problems and algebraic refutations of a related principle. [18] exploited this connection to derive separations between certain NP search classes, using lower bounds on the degree of algebraic refutations of PHP_n^m.

4.5.2 Lower bounds for bounded-depth Frege proofs

Secondly, good degree bounds for algebraic refutations of certain principles can give lower bounds for bounded-depth Frege proofs, when this principle is added as an axiom scheme. For example, Ajtai showed that for any p, q distinct primes, the mod q counting principle cannot be proven with polynomial-size, bounded-depth Frege proofs, even with the mod p counting principle as an axiom schema. This result was strengthened in [9] and [13] by using lower bounds on the degree of algebraic refutations of the mod principles. In general, let $P1$ and $P2$ be two principles. If there is a switching lemma that holds relative to the principle $P2$, and if we have a good degree bounds on algebraic refutations of the $P1$ principle, then it is possible to show that there are no efficient bounded-depth Frege proofs of $P2$, even if $P1$ is allowed as an axiom schema.

4.5.3 Theorem proving

A third and most straightforward application of small degree algebraic proofs is as a deterministic theorem prover. As mentioned earlier, a degree d algebraic refutation can be found deterministically in polynomial time by solving a sparse system of linear equations. We are currently in the process of implementing a theorem prover using these ideas. It is interesting to compare the power of our small-degree algebraic proof system with Resolution. It turns out that the two systems are incomparable. Since both the induction principle and the homesitting principle have polynomial-sized Resolution refutations, these lower bounds show that small-degree algebraic proofs cannot efficiently simulate Resolution. On the other hand, small-degree algebraic proofs (over GF_2) can prove the mod 2 principle, while Resolution proofs require exponential size. Thus, Resolution cannot efficiently simulate our small-degree algebraic system. It would be interesting to compare the power of the two systems on randomly generated, or "average case" unsatisfiable formulas. One big advantage of the small degree algebraic system is that it is just as efficient (within a small polynomial factor) to find a proof as it is to write down a proof; for Resolution, this does not appear to be the case, since the best known algorithm to search for a polynomial-size proof runs in time $2^{O(\sqrt{n})}$ ([16]).

5 Gröbner Proofs

The above lower bounds for small-degree algebraic proofs show that while small-degree proofs have nice computational properties, they are weak when compared to a standard proof system such as Resolution. The Gröbner proof system is a generalization of a small-degree algebraic proof, where we can *iteratively* obtain small-degree polynomial consequences of a set of polynomials.

More precisely, let $C = C_1 \wedge ... \wedge C_m$ be an unsatisfiable 3CNF formula, and let Q be the usual corresponding polynomial equations. (As before, we have the equations $x^2 - x = 0$ for each variable x as part of the initial equations Q.) Then a degree d Gröbner proof of C over F is a sequence of degree d polynomial equalities over F such that the final line is the polynomial 1, and where all other lines are either equations from Q, or follow from two previous polynomials by one of the following rules (corresponding to addition and multiplication, respectively):

1. From from $g_1(\vec{x}) = 0$ and $g_2(\vec{x}) = 0$, derive $ag_1(\vec{x}) + bg_2(\vec{x}) = 0$ where a, b are constants from F;

2. From $g(\vec{x}) = 0$ infer $xg(\vec{x}) = 0$ for x a variable.

As an example, consider the induction equations on three variables: $P_1 = (1 - x_1) = 0$; $P_2 = x_1 - x_1x_2 = 0$; $P_3 = x_2 - x_2x_3 = 0$ and $P_4 = x_3 = 0$. First, derive $R_1 = 1 - x_2$ by $(1 - x_2)P_1 + P_2$. Secondly, derive $R_2 = 1 - x_3$ by $(1 - x_3)R_1 + P_3$. Thirdly, add $R_2 + P_4$ to get $1 = 0$. Note that the degree of this Gröbner refutation is only 1, because each intermediate line in the refutation has degree at most 1. On the other hand, it is known that small-degree algebraic proofs require degree $O(\log n)$.

[16] proves the following theorem.

Theorem 11 *For any fixed constant d, there is a polynomial time algorithm with the following property: On input $g_1, ..g_m$ and h, where $g_1, ...g_m, h$ are polynomials of degree at most d in $x_1, ..x_n$, if there exists a degree d Gröbner proof of h from $g_1, ..g_m$, the algorithm finds a degree $d + 1$ such proof.*

In other words, there is a algorithm that gives a proof of within a polynomial factor in size of the shortest proof, and runs in polynomial-time in the size of the shortest proof.

The algorithm is essentially the Gröbner basis algorithm: each stage of the algorithm computes all possible degree d consequences of the degree d polynomials generated so far. The new polynomials are always written as a linear combination over the Gröbner basis, so that at each step either the vector space spanned by the current basis increases, or if it does not increase, then the algorithm terminates. The total number of stages is at most polynomial, since the number of basis elements covering the entire degree d polynomial space is polynomially bounded.

As applications of the above upper bounds, [16] proved the surprising results that polynomial-size tree-like Resolution proofs can be found deterministically in polynomial time, and that polynomial-size Resolution proofs can be found deterministically in subexponential time! (This deterministic simulation was then used to obtain simpler and somewhat stronger Resolution lower bounds [35].)

The lower bound for the homesitting principle shows that constant degree Gröbner proofs cannot be simulated by small degree algebraic proofs.

5.1 Lower bounds

There are no known non-trivial lower bounds for the Gröbner system at present, and this is an important open problem. Since small degree Gröbner refutations can be simulated by $ACC^0[2]$-Frege proofs, in order to make progress toward Frege lower bounds, we must be able to prove Gröbner lower bounds. Furthermore, a lower bound for Gröbner proofs for one of the mod principles where a switching lemma is known would yield a corresponding lower bound for Frege proofs whose formulas are parities of AC_0 formulas. [16] has made progress toward obtaining Gröbner lower bounds by proving an effective interpolation theorem for small degree Gröbner refutations. This result was proven by first showing that there exists a quasipolynomial-time algorithm that finds a constant degree Gröbner refutation, if one exists, and then by noting that an interpolation theorem follows from a deterministic simulation. Pudlák[34] has recently obtained a monotone version of this interpolation theorem, showing that exponential lower bounds for a certain monotone class of algorithms generalizing monotone span programs give rise to nonconstant lower bounds for Gröbner proofs. Thus, unconditional lower bounds for Gröbner proofs would follow from lower bounds for this class of (monotone) algorithms.

6 Open Problems

In this section we state some of the main open problems related to this work.

6.1 Degree lower bounds

As mentioned in Section 4, the best degree lower bound for the pigeonhole principle is $\sqrt{n}$, but the best upper bound is n. We would like to close this gap. What about the situation where there are many more pigeons than holes? Let $\neg PHP_n^m$ denote the pigeonhole principle where there are m pigeons and n holes. As m gets large relative to n, we would expect that it would be easier to refute $\neg PHP_n^m$, but so far this has not been shown. In particular, the $\sqrt{n}$ degree bound holds for algebraic refutations of PHP_n^m, for *any* m greater than n. Is this optimal when m is very large?

Can one prove that most randomly generated 3CNF formulas over n variables in the hard range (with about $4.3n$ clauses) require nearly linear degree algebraic proofs, over any field? This would be analogous to a result of Chvatal and Szemeredi showing that with high probability, randomly generated 3CNF formulas from this class require exponential-sized Resolution refutations.

6.2 $ACC^0[2]$-Frege lower bounds

The original motivation for defining Nullstellensatz and Gröbner proofs was to prove lower bounds for constant depth Frege systems, with mod 2 counting gates. (This proof system is often referred to as $ACC^0[2]$-Frege, because each formula in such a polynomial-size proof is in the complexity class $ACC^0[2]$.) It is still an important problem to prove lower bounds for this class of Frege systems.

One promising approach is as follows. Recently, Maciel and Pitassi [32] have shown that any quasipolynomial-size $ACC^0[2]$-Frege proof can be simulated by a quasipolynomial-size, depth 3 Frege proof of a very special form: The output gate is a weak threshold gate, the middle layer consists of mod 2 gates, and the input layer consists of AND gates of small fanin. Put another way, each formula in the depth 3 Frege proof is a probablistic, small-degree polynomial over GF_2. Using this result, one approach toward proving (conditional) lower bounds for $ACC^0[2]$-Frege proofs is to generalize the interpolation theorems for Gröbner proofs to work in the generalized case where the Gröbner proof is defined over probabilistic (rather than deterministic) small degree polynomials.

6.3 Relationship to standard proof systems

Does Extended Frege polynomially simulate algebraic proofs? Does Frege polynomially simulate algebraic NC_1-proofs? That is, algebraic proofs, where the coefficient polynomials are represented by polynomial-size algebraic formulas.

6.4 Relationship with algebraic circuit complexity

Valiant proposed studying the algebraic circuit model, and attacking the algebraic analog of the P versus NP question here first. This problem appears to be easier to solve, although, to date, very little progress has been made in this direction.

In this model, algebraic-P (over a field F) is defined to be the class of polynomials over $F[x_1, ..., x_n]$ such that: (1) the degree of the polynomial is small (bounded by a polynomial in n), and (2) there is a polynomial-size algebraic circuit for computing the polynomial.

In the same manner, we can define a restricted type of algebraic proof to be one where not only do we require the P_i's to be computed by efficient algebraic circuits, but also require that the degree of the P_i's be bounded by a polynomial. It may be simpler to study these types of proofs. There are many open questions here. First, can Extended Frege simulate these proofs? Can these proofs simulate Extended Frege? In the simulation of Extended Frege proofs by algebraic ones, we ended up with P_i's of large degree, and we suspect that this is optimal.

6.5 Theorem proving

An important practical problem is to use the small degree algebraic system and the small degree Gröbner system as deterministic theorem provers. We are currently working on this, and it will be interesting to understand practically as well as theoretically how well these systems do on standard hard examples, compared to existing backtracking methods as well as to simulated annealing methods. (See [31] for a nice survey of propositional satisfiability testing.)

In [16], it was shown that polynomial-sized Resolution proofs can be simulated by degree $O(\sqrt{n})$ Gröbner proofs. We would like to tighten this bound. Another very interesting question is whether or not Cutting Planes can be simulated by sublinear degree Gröbner proofs.

Acknowledgements

I would like to take this opportunity to thank the following people for the stimulating discussions and results that this paper are based upon: Paul Beame, Sam Buss, Steve Cook, Jeff Edmonds, Russell Impagliazzo, Jan Krajíček, Pavel Pudlák. I would also like to thank Maria Bonet, Ran Raz and Avi Wigderson for helpful comments. Lastly, I thank Steve Cook for the simplified proof of Lemma 2, and Sam Buss for many improvements to this paper, including the tighter proof of Theorem 4 presented here.

References

[1] Ajtai, M. (1988) The complexity of the pigeonhole principle, in: *Proc. IEEE 29th Annual Symp. on Foundation of Computer Science*, pp. 346-355.

[2] ——(1990) Parity and the pigeonhole principle, in: *Feasible Mathematics*, Eds. S.R.Buss and P.J.Scott, pp.1-24. Birkhaüser.

[3] ——(1993) Symmetric systems of linear equations modulo p, preprint.

[4] ——(1994) The independence of the modulo p counting principles, in: *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pp.402-417. ACM Press.

[5] Beame, P. (1993) A switching lemma primer, preprint.

[6] L. Babai, A. Gal, J. Kollar, L. Ronyai, T. Szabo, and A. Wigderson, "Extremal bipartite graphs and superpolynomial lower bounds for monotone span programs," In *Proceedings of the ACM Symposium on Theory of Computing*, 1996.

[7] Beame, P., Impagliazzo, R., Krajíček, J., Pitassi, T., Pudlák, P., and Woods, A. (1992) Exponential lower bounds for the pigeonhole principle, in: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pp.200-221. ACM Press.

[8] Beame, P., and Pitassi, T. (1993) An exponential separation between the matching principles and the pigeonhole principle, to appear in: *Annals of Pure and Applied Logic*. Preliminary version: University of Washington Technical Report, April 1993.

[9] Beame, P., Impagliazzo, R., Krajíček, J., Pitassi, T., Pudlák, P. (1995) Lower bounds on Hilbert's Nullstellensatz and propositional proofs, in *Thirty-fifth Annual Symposium on Foundations of Computer Science*, IEEE Press, pp. 794-806. Revised version to appear in Proceedings of the London Mathematical Society.

[10] Brownawell, D. (1987) Bounds for the degrees in the Nullstellensatz, *Annals of Mathematics* (Second Series), **126**: 577-591.

[11] Buss, S., "New lower bounds for the homesitting principle," Manuscript 1996.

[12] Buss, S., and Pitassi, T. (1995) Good degree lower bounds on Nullstellensatz refutations of the induction principle. Manuscript, 1995.

[13] Buss, S., Impagliazzo, R., Krajíček, J., Pudlák, P., Razborov, S., Sgall, J., Proof complexity in algebraic systems and constant depth Frege systems with modular counting. Manuscript 1995.

[14] Bonet, M., Pitassi, T., Raz, R. Lower bounds for Cutting Planes proofs with small coefficients. To appear in *Journal of Symbolic Logic*, 1995.

[15] Caniglia, L., Galligo, A., and Heintz, J. (1988) Some new effectivity bounds in computational geometry, in: *Proceedings 6th International Conference on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, Ed. T.Mora, pp. 131-151. Lecture Notes in Computer Science 357 (Springer Verlag, 1989).

[16] Clegg, M., Edmonds, J., and Impagliazzo, R. "Using the Groebner basis algorithm to find proofs of unsatisfiability," In *Proceedings of the ACM Symposium on Theory of Computing*, 1996.

[17] Cook, S., and Haken, A., Lower bounds for Cutting Planes proofs and monotone circuit complexity, Manuscript in preparation, 1995.

[18] Beame, P., Cook, S., Edmonds, J., Impagliazzo, R., and Pitassi, T. (1995) The relative complexity of NP search problems, in *Proceedings of the 27th ACM Symposium on Theory of Computing*, pp. 303-314.

[19] Cook, S. A., and Reckhow, A. R. (1979) The relative efficiency of propositional proof systems, *J. Symbolic Logic*, **44(1)**:36-50.

[20] Edmonds, J., The Home-sitting Principle, Manuscript 1995.

[21] Haken, A. (1985) The intractability of resolution, *Theoretical Computer Science*, **39**:297-308.

[22] Håstad, J. (1987) *Computation limits of small depth circuits.* ACM dissertation award, 1986. MIT Press.

[23] Impagliazzo, R. (1995) The Ideal Generation Proof System, Manuscript, 1995.

[24] Kollár, J. (1988) Sharp effective Nullstellensatz, *J. Amer. Math. Soc.*, **1(4)**:963-975.

[25] Krajíček, J. "Interpolation theorems, lower bounds for proof systems and independence results for bounded arithmetic," To appear in the *Journal of Symbolic Logic.*

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

[27] ——(1994) Bounded arithmetic, propositional calculus and complexity theory, *Cambridge University Press*, in print.

[28] Krajíček, J.,Pudlák, P. and Woods, A. (1991) Exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle, *Random Structures and Algorithms*, to appear.

[29] Lovasz, L., "Bounding the independence number of a graph," Bonn workshop on combinatorial optimization, *Annals of Discrete Mathematics* (16), North-Holland, 1982, pp. 213-223.

[30] Mundici, D., Tautologies with a unique Craig interpolant, uniform vs. nonuniform complexity, *Annals of Pure and Applied Logic*, Vol. 27, 1984, pp.265-273.

[31] Mitchell, D., Propositional satisfiability testing, Manuscript, University of Toronto, September 1995.

[32] Maciel, A., and Pitassi, T., "New characterizations of bounded-depth Frege proofs," Manuscript, July 1996.

[33] Pudĺak, P. "Lower bounds for resolution and cutting planes proofs and monotone computation," to appear in the *Journal of Symbolic Logic.*

[34] Pudĺak, P. "Interpolation for proof systems with polynomials," Manuscript, April 1996.

[35] P. Beame and T. Pitassi, "Simplified and improved resolution lower bounds," To appear in *Proceedings from 1996 Symposium on Foundations of Computer Science.*

[36] Paris, J. B., and Wilkie, A. J. (1985) Counting problems in bounded arithmetic, in: *Methods in Mathematical Logic*, LNM 1130, pp.317-340.Springer.

[37] Pitassi, T., Beame, P., and Impagliazzo, R. (1993) Exponential lower bounds for the pigeonhole principle, in: *Computational Complexity*, **3**:97-308.

[38] Riis, S. (1993) Independence in bounded arithmetic, PhD. Thesis, Oxford University.

[39] ——(1994) Count(q) does not imply Count(p), preprint (summer 1993, revised summer 1994).

[40] Razborov, A.A., Lower bounds on the size of bounded depth networks over a complete basis with logical addition, Matematischi Zametki, 41 (1987), pp. 598-607. English translation in Mathematical Notes of the Academy of Sciences of the USSR 41(1987) 333-338.

[41] Razborov, A.A., Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic, Izvestiiya of the RAN, 59 (1995), pp. 201-224.

[42] Razborov, A., Rudich, S. Natural proofs, in *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, 1994, pp. 204-213.

[43] Schwartz, J. "Probabilistic algorithms for verification of polynomial identities" *Journal of the Association for Computing Machinery*, 27,4, October 1980, pp. 701-717.

[44] Smolensky, R. Algebraic methods in the theory of lower bounds for Boolean circuit complexity, in *Proceedings of the 19th Annual ACM Symposium on the Theory of Computing*, ACM Press, 1987, pp.77-82.

[45] Statman, R., "Complexity of derivations from quantifier-free Horn formulae, mechanical introduction of explicit definitions and refinement of completeness theorems," *Logic Colloquium 1976*, North-Holland, Edited by R. Gandy and M. Hyland, 1977.

[46] Urquhart, A., The complexity of propositional proof systems, Survey article to appear in *Journal of Symbolic Logic*.