Propositional Proof Complexity: Past, Present, and Future

Paul Beame and Toniann Pitassi

Abstract. Proof complexity, the study of the lengths of proofs in propositional logic, is an area of study that is fundamentally connected both to major open questions of computational complexity theory and to practical properties of automated theorem provers. In the last decade, there have been a number of significant advances in proof complexity lower bounds. Moreover, new connections between proof complexity and circuit complexity have been uncovered, and the interplay between these two areas has become quite rich. In addition, attempts to extend existing lower bounds to ever stronger systems of proof have spurred the introduction of new and interesting proof systems, adding both to the practical aspects of proof complexity as well as to a rich theory. This note attempts to survey these developments and to lay out some of the open problems in the area.

1. Introduction

One of the most basic questions of logic is the following: Given a universally true statement (tautology) what is the length of the shortest proof of the statement in some standard axiomatic proof system? The propositional logic version of this question is particularly important in computer science for both theorem proving and complexity theory. Important related algorithmic questions are: Is there an efficient algorithm that will produce a proof of any tautology? Is there an efficient algorithm to produce the shortest proof of any tautology? Such questions of theorem proving and complexity inspired Cook's seminal paper on NP-completeness notably entitled "The complexity of theorem-proving procedures" [**35**] and were contemplated even earlier by Godel in his now well-known letter to von Neumann (see [**86**]).

The above questions have fundamental implications for complexity theory. As formalized by Cook and Reckhow [**37**], there exists a propositional proof system giving rise to short (polynomial-size) proofs of all tautologies if and only if NP equals co-NP. Cook and Reckhow were the first to propose a program of research aimed at attacking the NP versus co-NP problem by systematically studying and proving strong lower bounds for standard proof systems of increasing complexity. This program has several important side effects.

First, standard proof systems are interesting in their own right. Almost all theorem-proving systems implement a deterministic or randomized procedure that is based on a standard propositional proof system, and thus upper and lower bounds on these systems shed light on the inherent complexity of any theorem-proving system upon which it is based. The most striking example is Resolution on which almost all propositional theorem provers (and even first-order theorem provers) are based.

Secondly, and of equal or greater importance, lower bounds on standard proof systems additionally prove that a certain class of algorithms for the satisfiability problem will fail to run in polynomial-time.

This program has led to many beautiful results as well as to new connections with circuit complexity within the last twenty years. In this article, we will try to highlight some of the main discoveries, with emphasis on the interplay between logic, (circuit) complexity theory, and combinatorics that has arisen. We omit all proofs; see [**92**] for a quite readable survey that includes detailed proofs of many of the earlier results.

In section 2, we define various proof systems that will be discussed throughout this article. In section 3, we review some of the main lower bounds that have been proven for standard proof systems, emphasizing the combinatorial techniques and connections to circuit complexity that have been shown. Finally in section 4, we list some of the main open questions and promising directions in the area.

2. Propositional proof systems

What exactly is a propositional proof? Cook and Reckhow were possibly the first to make this and related questions precise. They saw that it is useful to separate the idea of providing a proof from that of being efficient. Since there are only finitely many truth assignments to check, why not allow the statement itself as a proof? What extra value is there in a filled out truth table or a derivation using some axiom/inference scheme? The key observation is that a proof is easy to check, unlike the statement itself. Of course we also need to know the format in which the proof will be presented in order to make this check. That is, in order to identify some character string as a proof we must see it as an instance of some general format for presenting proofs. Therefore a *propositional proof system S* is defined to be a polynomial-time computable predicate S such that for all F,

$$F \in \mathrm{TAUT} \quad \Leftrightarrow \quad \exists p.\ S(F,p). \tag{1}$$

That is, we identify a proof system with a polynomial time procedure that checks the correctness of proofs.[1] Property (1) ensures that the system S is

[1]Cook and Reckhow's definition is formally different although essentially equivalent to this one. They define a proof system as a polynomial-time computable *onto* function $f : \Sigma^* \rightarrow \mathrm{TAUT}$ which can be thought of as mapping each string viewed as potential proof

logically both sound and complete. The *complexity* $comp_S$ of a propositional proof system S is then defined to be the smallest bounding function $b : \mathbb{N} \to \mathbb{N}$ on the lengths of the proofs in S as a function of the tautologies being proved, i.e. for all F,

$$F \in \mathrm{TAUT} \Leftrightarrow \exists p. |p| \leq b(|F|).\ S(F, p).$$

Efficient proof systems correspond to those of polynomial complexity; these are called *p-bounded*.

Given these definitions, many natural questions arise: How efficient are existing proof systems? How can one compare the relative efficiencies of proof systems? Can one classify proof systems using reduction as we do languages? Is there a proof system of optimal complexity (up to a polynomial)?

The key tool for comparing proof systems is p-simulation. A proof system T *p-simulates* a proof system S iff there is a polynomial-time computable function f mapping proofs in S into proofs in T, that is for all $F \in \mathrm{TAUT}$, $S(F, p) \Leftrightarrow T(F, f(p))$. We use non-standard notation and write $S \leq_p T$ in this case. Clearly, it implies that $comp_T \leq comp_S^{O(1)}$. (One says that T *weakly p-simulates* S iff we have this latter condition but we do not know if such a reducing function f exists.) One says that S and T are *p-equivalent* iff each p-simulates the other. Obviously, two p-equivalent proof systems either are both p-bounded or neither is.

2.1. Frege and extended-Frege proofs. Cook and Reckhow did more than merely formalize the intuitive general notions of the efficiency of propositional proofs. They also identified two major classes of p-equivalent proof systems which they called Frege and extended-Frege systems in honor of Gottlob Frege who made some of the first attempts to formalize mathematics based on logic and set theory [**42, 43**] (and whose work is now best known as the unfortunate victim of Russell's famous paradox concerning the set of all sets that are not members of themselves).

A Frege system $\mathcal{F}$ is defined in terms of a finite, implicationally complete (enough to derive every true statement) set $\mathcal{A}_\mathcal{F}$ of axioms and inference rules. The general form of an inference rule is written as $\frac{A_1, \ldots, A_k}{B}$ where $A_1, \ldots, A_k$ and B are propositional formulas; the rule is an axiom if $k = 0$. A formula H follows from formulas $G_1, \ldots, G_k$ using this inference rule if there is a consistent set of substitutions σ of formulas for the variables appearing in the rule such that $G_i = A_i^\sigma$ for $i = 1, \ldots, k$ and $H = B^\sigma$.

For a Frege system $\mathcal{F}$, a typical set of axioms $\mathcal{A}_\mathcal{F}$ might include the axiom of the excluded middle $\frac{}{A \vee \neg A}$ or identity $\frac{}{A \to A}$ as well as the cut rule $\frac{A \vee C, \neg C \vee B}{A \vee B}$ or *modus ponens* $\frac{A, A \to B}{B}$. A proof of a tautology F in $\mathcal{F}$ consists of a finite sequence $F_1, \ldots, F_r$ of formulas, called *lines*, such that $F = F_r$

onto the tautology it proves. The analogous function f in our case would map (F, p) to F if $S(F, p)$ were true and would map it to a trivial tautology, $(x \vee \neg x)$, otherwise. In the converse direction one would define $S(F, p)$ to be true iff $f(p) = F$.

and each F_j either is an an instance of an axiom in $\mathcal{A}_{\mathcal{F}}$ or follows from some previous lines $F_{i_1} \ldots , F_{i_k}$ for $i_1, \ldots , i_k < j$ using some inference rule of $\mathcal{A}_{\mathcal{F}}$. An equivalent way of using a Frege system works backwards from $\neg F$ to derive a contradiction such as $p \wedge \neg p$. The size of a Frege proof is typically defined to be the total number of symbols occurring in the proof. The proof can also be tree-like or dag-like: in the tree-like case, each intermediate formula can be at used at most once in subsequent derivations; in the more general dag-like case, an intermediate formula can be used unboundedly many times. Krajíček [**60**] has shown that for Frege systems, there is not much loss in efficiency in going from a dag-like proof to a tree-like proof.

Various Frege systems (which have also been called Hilbert systems or Hilbert-style deduction systems) appear frequently in logic textbooks. However, it is difficult to find two logic textbooks that define precisely the same such system. Cook and Reckhow showed that these distinctions do not matter; namely, all Frege systems are p-equivalent. Furthermore, they showed that Frege systems were also p-equivalent to another class of proof systems appearing frequently in logic textbooks called sequent calculus or Gentzen systems. These systems manipulate pairs of sequences (or sets) of formulas, written as $\Gamma \rightarrow \Delta$, where Γ and Δ are sequences of formulas with the intended interpretation being that the conjunction of the formulas in Γ implies the disjunction of the formulas in Δ. Therefore to prove a formula F in the sequent calculus, one proves the corresponding sequent $\rightarrow F$.

In any sequent calculus system, there is an underlying basis set of connectives, B. B can consist of unbounded fan-in or bounded fan-in connectives, and the only requirement is that B be a complete basis. (Typically, B is the standard basis consisting of $\wedge$, $\vee$ and $\neg$.) The only initial sequent is $A \rightarrow A$ for any formula A defined over B.

Additionally there are three types of rules for deriving new sequents from previous ones: (i) structural rules; (ii) logical rules; and (iii) the cut rule. Structural rules do not actually manipulate the underlying formulas of the sequent, but instead they allow one to operate on the sequences of formulas as sets rather than sequences. A typical structural rule is contraction which allows us to derive $\Gamma, A \rightarrow \Delta$ from $\Gamma, A, A \rightarrow \Delta$. The logical rules allow us to build larger formulas from previous ones, according to the truth-table definition of each of the connectives in B. More precisely, for each connective in B, there are two logical rules, one for introducing the connective on the left and one for introducing the connective on the right. For example, if $\wedge$ is in B, then the $\wedge$-left rule would allow us to derive $\Gamma, A \wedge B \rightarrow \Delta$ from $\Gamma, A, B \rightarrow \Delta$, and the $\wedge$-right rule would allow us to derive $\Gamma \rightarrow A \wedge B, \Delta$ from $\Gamma \rightarrow A, \Delta$ and $\Gamma \rightarrow B, \Delta$.

Of particular importance for sequent calculi is the cut rule: From $\Gamma, A \rightarrow \Delta$ and $\Gamma \rightarrow A, \Delta$, derive $\Gamma \rightarrow \Delta$. Gentzen showed that the cut rule is unnecessary but it may have a huge impact on proof length. If the cut rule is removed then one obtains a much weaker system than Frege systems. This system is known as analytic tableaux or cut-free LK. While analytic tableaux are often more efficient than truth tables, somewhat surprisingly they cannot even p-simulate truth tables because their worst-case complexity is $\Omega(n!)$ rather than $O(n2^n)$ for truth tables (see [**92**]).

The other major class of proof systems identified by Cook and Reckhow includes systems that permit one to extend Frege proofs by introducing new propositional variables to stand for arbitrary formulas appearing in the proof. All such systems, which are called extended-Frege, are p-equivalent to each other. These systems appear to be much more succinct than Frege proofs and can conveniently express many mathemtical arguments quite naturally. It has been shown by Dowd (unpublished) and also by Krajíček and Pudlák [**61**] that extended-Frege proofs are also p-equivalent to substitution-Frege (sF) proofs. (In sF, one is allowed to use each line of a Frege proof immediately as if it were an axiom; that is, new lines follow from existing ones by substituting arbitrary formulas for their propositional variables.)

2.2. CNF refutations and Resolution. Using the construction of the standard reduction from SAT to 3-SAT, one can take an arbitrary propositional formula F and convert it to a CNF or 3-CNF formula in such a way that it has only polynomially larger size and is unsatisfiable iff the original formula was a tautology. To do this one adds new variables x_A to stand for each of its subformulas A and clauses to specify that the value at each connective is computed correctly as well as one clause of the form $\neg x_F$. In this way, one can consider any sound and complete system that produces refutations for CNF formulas as a general propositional proof system.

In the 1960's several such refutation systems were developed. The most powerful of these systems is Resolution [**84**], which in its propositional form is a very specialized form of a Frege proof system that can only manipulate clauses and has only one inference rule, the resolution rule

$$\frac{A \vee x,\ \ B \vee \neg x}{A \vee B}$$

called *resolution on variable* x and is a special form of cut. The contradictory formula to be derived is simply an empty clause (which can be seen as the result of resolving on clauses p and $\neg p$).

Resolution was pre-dated by two systems known as Davis-Putnam procedures which are still the most widely used in propositional theorem proving. The general idea of these procedures is to convert a problem on n variables to problems on $n-1$ variables by eliminating all references to some variable. The former [**40**] which we call DP does this by applying all possible uses of the resolution rule on a given variable to eliminate it. The latter [**39**], which we call DLL and is the form used today, branches based on the possible truth assignments to a given variable; although at first this does not look like Resolution, it is an easy argument to show that this second form is equivalent to the special class of tree-like Resolution proofs. As a proof system, Resolution is strictly stronger than DP [**50**] which is strictly stronger than DLL [**92**]. The reasons for DLL's popularity are related to its proof search properties which we discuss below.

A more general but still restricted form of Resolution is called *regular* Resolution, which was introduced and analyzed by Tseitin [**89**]. A regular Resolution refutation is a Resolution refutation whose underlying directed acyclic graph has the property that along each path from the root (empty clause) to a leaf (initial clause), each variable is resolved upon at most once. It is not too hard to see that any minimal tree-like Resolution refutation is regular; also, the DP algorithm trivially produces a regular Resolution proof. For a period of 15 years after Tseitin's analysis, although there were improvements in the bounds derived for regular resolution [**49**], understanding general resolution seemed out of reach.

2.3. Circuit-complexity-based Proof Systems. One of the most powerful insights that has developed in the study of propositional proof complexity is that there is a parallel between circuit-based complexity classes and propositional proof systems. This insight was first made by Cook [**36**] where he established a close connection between polynomial-size extended Frege proofs and proofs using "polynomial-time" reasoning. (In more familiar terms, he showed that extended Frege proofs are the nonuniform analog of polynomial-time proofs systems such as PV or S^1_2, in the same way that polynomial-size circuits are the nonuniform analog of the complexity class P.) The same intuition (applied to other proof systems) was subsequently used to obtain other important results by Ajtai [**4, 1**] and Buss [**29**]. More generally, the parallel between circuit classes and proof systems has greatly broadened the range of proof systems that are typically considered and has led to new techniques for analyzing proof systems and circuit classes. We first briefly outline the general form of this correspondence and then we re-examine and refine some of the proof systems above in this light.

Typically, circuit-based complexity classes are defined by giving a structural characterization of a class of circuits and then placing some bound on the size of the circuits involved. For many circuit-based complexity classes C this size bound is polynomial. For any such class C we can consider a Frege-style proof system whose lines are circuits with the same structural characterization as the circuits defining C but which do not necessarily satisfy the size bound. The set of circuits of this type must be closed under substitution into any formula appearing in an axiom or inference rule of the system. Although the notation is not precise in general, we call such a proof system C-Frege. (Since we can always assume that our goal formula is in CNF, there is no problem representing it in C-Frege for virtually any non-trivial C.)

For example, the complexity class corresponding to the set of all polynomial size propositional formulas is NC^1 so NC^1-Frege would just be another name for Frege. It is also easy to observe that the extension rule of extended-Frege proofs builds circuits in terms of the original propositional variables in which the new variables give the values computed by sub-circuits. Thus extended-Frege could also be called $\mathsf{P/poly}$-Frege. Resolution is a Frege system that manipulates simple clauses (or, alternatively, terms if one views it dually as a proof system rather than a refutation system) but there isn't a convenient name for this complexity class of depth 1 formulas. Note that with this intuition it is clear that extended-Resolution, the natural generalization of Resolution that permits the introduction of new propositional variables, is p-equivalent to extended-Frege since it clearly can generate any circuit in $\mathsf{P/poly}$ with a polynomial number of extensions.

A very natural new proof system arising from this correspondence is a generalization of Resolution to arbitrary constant-depth, unbounded fan-in formulas/circuits (for constant depth there is no difference between polynomial-size formulas and circuits). This new system, AC^0-Frege, also known as constant- or bounded-depth Frege, first arose from the study of bounded first-order arithmetic (the word 'bounded' in that case derives from polynomial complexity bounds rather than from the depth). AC^0-Frege proofs derive from translations of proofs in certain systems of bounded first-order arithmetic [**70, 28**] which are restrictions/extensions of Peano arithmetic that model feasible inference. Although these motivations are important in the study of constructive logic, space considerations do not permit us to go into detail about them; we refer the interested reader to [**60**] where many of these connections are described in detail. Typically, lower bounds on the size of AC^0-Frege proofs can show that related first-order tautologies are unprovable in a given system of bounded arithmetic. These translations are analogous to those of Furst, Saxe, and Sipser [**47**], and Sipser [**87**] which convert oracle computations in the polynomial hierarchy to constant-depth unbounded fan-in circuits. In addition to AC^0-Frege, proof systems for $\mathsf{AC}^0[p]$-Frege and TC^0-Frege and their subclasses have also been studied extensively.

The correspondence between circuit classes and proof systems has not only been fruitful in developing ideas for new proof systems. It has also been the avenue for applying circuit lower bound techniques to propositional proofs. Some of the major progress of the last decade building on the original insight due to Ajtai [**4, 1**], has been in achieving lower bounds for AC^0-Frege proof systems and their extensions. In general, the intuition for this approach is that any tautology that needs to use in its proof some concept that is not representable in complexity class C will not be efficiently provable in C-Frege.

3. The State of the Art in Proof Complexity

We give a quick tour of the state of the art in propositional proof complexity. As is always inevitable in a short survey such as ours, space considerations do not permit us to do justice to the full range of results available. Although we will not always emphasize the connections very strongly, many of these results have been inspired or derived from methods in circuit complexity, and conversely lower bounds for particular proof systems imply lower bounds for restricted families of algorithms for solving SAT. We expect this cross-fertilization to continue.

In keeping with the general program of proving that proof systems are not efficient, much work has been devoted to proving lower bounds on the sizes of proofs of specific tautologies. One can broadly characterize several classes of formulas for which these lower bounds have been shown. The first two classes consist of the propositional translations of combinatorial or counting principles. The first of these involves highly symmetric counting principles. A canonical example here is the translation of the pigeonhole principle, which prohibits 1-1 functions from m to n for $m > n$, as an unsatisfiable CNF formula $\neg PHP_n^m$ with mn atoms p_{ij} denoting whether or not i is mapped to j. Related principles include the counting principles $Count_p^n$ for $n \not\equiv 0 \pmod p$ which express the property that n cannot be perfectly partitioned into sets of size p and onto-PHP_n^m which only prohibits bijections rather than all 1-1 functions. The second class consists of much less symmetric counting principles, such as the 'odd-charged graph' principles for bounded-degree graphs, one for each graph, which express the property that the sum of the degrees in any of its sub-graphs is even; these are of particular interest when the underlying graph is a bounded-degree expander [**91**]. The third class of formulas are 'minterm-maxterm' formulas associated with any monotone function, which express the fact that any minterm and any maxterm of such a function must overlap; these are of particular interest when the function is known to require exponential-size monotone circuits. A related family of formulas is obtained by taking a language L that is in $\mathsf{NP} \cap \mathsf{coNP}$, and writing the formula expressing the fact that any instance x cannot have both a 'yes' witness and a 'no' witness. Finally, there are k-CNF formulas randomly chosen from an appropriate distribution.

We summarize known bounds for these formulas by considering the various proof systems one by one. Four basic methods can be identified for proving these lower bounds: (1) the bottleneck counting method; (2) the method of restrictions; (3) the interpolation method; (4) algebraic methods. The first three methods are quite specific whereas the last method is the youngest, probably the most powerful, and already has many facets.

Some of the state of our knowledge of proof complexity lower bounds can be summarized by the following chain

$$\text{Resolution} <_p \mathsf{AC}^0\text{-Frege} <_p \mathsf{AC}^0[p]\text{-Frege} \leq_p \mathsf{TC}^0\text{-Frege}.$$

The separations match all the major circuit complexity separations known with the notable exception of the last one. However, the prospect for proof complexity seems better than that for circuit complexity: The results of [**81**] indicate that to make further progress in circuit lower bounds will likely require very new, nonconstructive techniques. However such barriers do not currently exist in proof complexity: that is, proving superpolynomial lower bounds for Frege systems might be no harder than what we currently know how to do. The techniques used to obtain some of the most recent results in proof complexity use insights from areas of mathematics apparently unrelated to those applied to show circuit complexity bounds.

3.1. Resolution. Resolution is the most well-studied model. Exponential lower bounds are now known for all of the major classes of formulas listed above. The first superpolynomial lower bound for Resolution was obtained by Tseitin in the 1960's for the odd-charged graph tautologies in the special case of regular Resolution [**89**]. Interestingly, obtaining an improvement of this bound to an exponential one by Galil [**49**] was a driving force behind some of the early work in the development of the theory of expander graphs [**48**].

There was a 15+ year gap before the first superpolynomial lower bound for proofs in general Resolution was obtained by Haken [**52**] who showed exponential lower bounds for the pigeonhole principle. Subsequently, exponential bounds have also been shown for the odd-charged graph formulas [**91**], random k-CNF formulas with various clause/variable ratios [**32, 46, 10, 9**], and minterm-maxterm formulas [**75**]. The proofs of all of the strongest forms of these bounds for Resolution, other than those for the minterm-maxterm formulas, involve a technique known as *bottleneck counting* due to Haken.

In this method, one views the proof as a directed acyclic graph of clauses and views the truth assignments as flowing from the root of the directed acyclic graph to a leaf, where an assignment flows through a clause C if and only if: (i) it flows through the parent clause of C and (ii) the assignment falsifies C. Each assignment can be seen to flow through a unique path in any Resolution refutation. The idea is to show that for the formula in question, there must exist a large set of truth assignments with the property that each must pass through a large clause. Since a large clause cannot falsify too many assignments, this implies that there must exist many large clauses and hence the proof must be large.

An essential lemma in any bottleneck counting argument is to show that any Resolution refutation of F must involve a large clause. Recently, it has been shown [**17**], using ideas from [**33**], that for a suitable choice of parameters this lemma is also sufficient, namely any Resolution refutation of small size can be converted into a refutation with only small clauses.

Another method used to obtain exponential Resolution lower bounds, used for example for the minterm-maxterm formulas, is the method of interpolation, which will be discussed in section 3.3.

In addition to lower bounds for general resolution, there is also practical interest in understanding the behavior of the special cases of DP and DLL algorithms. (See [**68, 34**].) Random k-CNF formulas have been of particular interest in this regard and there is a variety of results giving more precise bounds on their properties both as proof systems and as satisfiability algorithms at various clause/variable ratios [**32, 46, 10, 9**].

3.2. AC^0-Frege systems and their extensions. While Haken's bound for resolution was a major breakthrough, it is the paper by Ajtai [**4**] giving super-polynomial lower bounds for proofs of the pigeonhole principle in AC^0-Frege systems that has formed the basis of much of the research in proof complexity over the last decade. As mentioned above, this gave the first connection between the techniques of circuit complexity and those of proof complexity. Ajtai's result has been improved to exponential lower bounds and these apply to all the symmetric counting principles in the first class above, e.g. [**14, 74, 63**]. Despite this success, no lower bounds are known for AC^0-Frege proofs of formulas in the other classes above although there are certain other tautologies for which we know a superpolynomial separation as a function of the depth [**59**].

The *restriction method*, by which the lower bounds above were shown comes from Ajtai's paper [**4**]. The essence of this idea is to apply restrictions to try to simplify each of the formulas in the proof yet leave the input tautology still highly non-trivial. Thus the basic method is very similar to the random-restriction method, used to show that AC^0 cannot compute parity. However, it is necessarily more complex: Since the circuits appearing in a sound proof always compute the constant function 1, the usual simplification induced by restrictions applied to circuits must be replaced by one that includes a form of approximation as well.[2] Using this method, one shows that if the proof is too short, then there exists a restriction such that after applying the restriction to the short proof, what results is a very trivial proof of a formula of the same basic form, but on a reduced number of variables. Then a contradiction can be reached by showing that such a trivial proof cannot exist.

Once it is known that the pigeonhole principle is not provable in AC^0-Frege one can immediately obtain a stronger system by adding PHP_n^{n+1} as an axiom schema for arbitrary n; i.e., one is permitted to derive lines in the proof by substituting arbitrary formulas for the variables of some PHP_n^{n+1}. Ajtai [**5**] showed that even in this stronger system $Count_2^n$ does not have polynomial-size proofs. There is now some quite interesting structure known about the relative proof strength of these augmented AC^0-Frege systems in which some axiom scheme is added to the basic system.

In the system above in which PHP_n^{n+1} was added as an axiom schema, all of the $Count_p^n$ principles in fact are now known to require exponential size proofs [**15, 83**]; conversely, given any $Count_p^n$ axiom schema, any bounded-depth Frege proofs of PHP_n^{n+1} or onto-$PHP_n^{n+p^{\epsilon \log n}}$ requires exponential size but onto-PHP_n^{n+1} is trivial [**11**]. Thus PHP_n^{n+1} is exponentially stronger than onto-PHP_n^{n+1}. Quite precise conditions are now known

---

[2] [**4**] used the language of forcing to describe this approximation; the cleanest way of expressing this approximation is in terms of so-called k-evaluations which are described in [**13, 92, 11**] and are a modification of the definitions in [**63**].

under which exponential separations exist between the various $Count_p^n$ principles in this context [**23, 13, 11**].

The proofs for these results begin with the same restriction method strategy described above. However, with the additional axiom schema there is the further requirement that each line in the proof where the axiom schema is applied be simplified just as the rest of the proof is. The need to prove this latter requirement motivated the introduction of Nullstellensatz proofs [**13**] (see below) and the exponential separations are all derived from lower bounds on the degrees of such proofs.

The method involving Nullstellensatz degree lower bounds is not the only one that has been used for obtaining such separations. In proving the first super-polynomial separations between the various $Count_p^n$, Ajtai [**2**] used certain properties of concisely represented symmetric systems of linear equations [**3**] which he proved using structural results in the theory of representations of the symmetric group [**55**] over $GF(p)$ to show that the axiom schemas are appropriately simplified. This technique has recently been employed to give super-polynomial lower bounds for very powerful algebraic proof systems (see sections 3.4 and 3.5).

3.3. Cutting Planes and Interpolation. The method of using cutting planes for inference in the study of polytopes in integer programming was first described by Gomory [**51**], modified and shown to be complete by Chvátal [**30**], and first analyzed for its efficiency as a proof system in [**38**]. It is one of two classes of proof systems developed by representing unsatisfiably problems as integer or 0-1 programming problems, the other being a collection of systems due to Lovasz and Schrijver [**64**] which are described in detail in the open problems section.

Cutting Planes proofs manipulate integer linear inequalities: One can add inequalities or multiply them by positive constants but the truly powerful rule is the rounded division rule:

$$ca_1x_1 + ca_2x_2 + \dots ca_kx_k \geq b \quad \Rightarrow \quad a_1x_1 + a_2x_2 + \dots a_kx_k \geq \lceil b/c \rceil.$$

A refutation of a set of integer linear inequalities is a sequence of inequalities, where each inequality is either one of the original ones or follows from previous inequalities by applying one of the above rules, and where the final inequality is $1 \geq 0$. To refute a CNF formula, one first converts each clause into an equivalent integer linear inequality. Cutting planes proofs can simulate Resolution efficiently and easily prove all of the symmetric counting tautologies mentioned above.

Exponential lower bounds have been shown for cutting planes proofs of the minterm-maxterm formulas using a method called *interpolation*. In this method one begins with an unsatisfiable formula of the form $F = A(x,z) \wedge B(y,z)$ where we view x and y each as a vector of 'private' variables, and z as a vector of 'shared' variables. After any assignment τ to the common z variables is made, in the remaining formula $A(x,\tau) \wedge B(y,\tau)$, it must be the case that either A is unsatisfiable or B is unsatisfiable. The associated interpolation problem for F takes as input an assignment τ to the common variables, and outputs A only when $A(x,\tau)$ is unsatisfiable, and outputs B only when $B(y,\tau)$ is unsatisfiable. Of course, sometimes both A and B may be acceptable answers. (This problem is called the interpolation problem since it is equivalent to Craig Interpolation. In the typical formulation, G is a tautological formula in the form $A'(x,z) \rightarrow B'(y,z)$. In our formulation F is $\neg G$, A is A', and B is $\neg B'$.)

For arbitrary F, it may be very difficult to solve the interpolation problem associated with F: if L is a decision problem in $\mathsf{NP} \cap \mathsf{co\text{-}NP}$, then if we define $A(x,z)$ to be a formula stating that x is a 'yes' witness for the instance z, and let $B(y,z)$ state that y is a 'no' witness for z, then the existence of a polynomial time algorithm for the interpolation problem would have surprising consequences for complexity theory! (See [**69**]). However it might still be that, whenever F (of the above form) has a short refutation in some proof system S, the interpolation problem associated with F has a polynomial-time solution. This possibility was first suggested by Krajíček. If this situation exists for proof system S, then we say that S has the *feasible interpolation property.* There is also a monotone version of feasible interpolation. Namely, $F = A(x,z) \wedge B(y,z)$ is monotone if z occurs only positively in A, and in this case the interpolation problem is monotone. S has the monotone feasible interpolation property if whenever F is monotone and has a short S-proof, then the associated interpolation problem has polynomial-size (uniform) monotone circuits.

Razborov [**80**] and independently Bonet, Pitassi and Raz [**20**] were the first to use the above idea to obtain exponential lower bounds for certain proof systems. In [**80**], a formula (formalizing that SAT does not have polynomial-size circuits) is constructed with the property that the associated interpolant problem has no polynomial-time circuits, under cryptographic assumptions. [**20**] constructs a monotone formula with the property that the associated interpolant problem has no monotone polynomial-time circuits (under no complexity assumptions). On the other hand they show that small-weight Cutting Planes has monotone feasible interpolation, thus implying exponential lower bounds. Pudlák [**75**] significantly extended the above ideas by showing that unrestricted Cutting Planes also has a form of monotone feasible interpolation. This combined with new exponential lower bounds for monotone real circuits [**75, 53**] gives unconditional lower bounds for Cutting Planes.

Feasible interpolation can thus give very good lower bounds for many proof systems (sometimes only under cryptographic assumptions). In addition to the unconditional lower bounds mentioned above, conditional lower bounds have been shown for all of the following systems: Resolution, Cutting Planes, Nullstellensatz [**33**], Polynomial Calculus [**76**], as well as any proof system where the underlying formulas in the proof have small probabilistic communication complexity [**20**]. Unfortunately there are strong negative results, showing that the interpolation method *cannot* be applied to give lower bounds for the following proof systems, again under various cryptographic assumptions: Extended Frege [**62**], Frege, TC^0-Frege [**21**], and even AC^0-Frege [**65, 19**].

A disadvantage of the interpolation method for obtaining lower bounds is that it applies only to formulas of a very special form. Thus, for example, nothing is known about the length of the shortest Cutting Planes proofs for either the odd-charged graph formulas or for random formulas.

3.4. Algebraic proof systems. The Nullstellensatz [**13**] and Polynomial Calculus [**33**] proof systems are based on a special case of Hilbert's famous Nullstellensatz which relates the question of the non-existence of simultaneous zeros of a family of multivariable polynomials in certain fields to the question of the existence of coefficient polynomials witnessing that 1 is in the ideal generated by these polynomials. (In fact they use a generalization of this special case to rings as well as fields.)

To use this relation one first expresses an unsatisfiable Boolean formula as a system of constant-degree polynomial equations in some polynomial ring. For the propositional versions of the symmetric counting principles, these translations are quite natural, for example PHP^m_n when translated has polynomials $\sum_{j\leq n} x_{ij} - 1 = 0$ for each $i \leq m$, as well as $x_{ij}x_{i'j} = 0$ for each $i \neq i' \leq m$ and $j \leq n$. More generally, there are natural low-degree translations of arbitrary CNF formulas that are similar to those used in probabilistically checkable proofs (PCP) [**8, 41**]. To use these mechanisms to detect 0-1 solutions only, we add the equations $x^2 - x = 0$ for each variable x. Hilbert's Nullstellensatz implies that such a system $\{\vec{Q}(\vec{x}) = 0\}$ does not have a solution if and only if there exists a family of multi-variate polynomials $\vec{P}$ such that $\sum_i P_i(\vec{x})Q_i(\vec{x}) \equiv 1$. It is easy to see that the $x^2 - x = 0$ equations guarantee that degree at most n is sufficient.

The complexity in the Nullstellensatz proof system is simply the size of the dense representation of the coefficient polynomials and thus of the form $n^{O(d)}$ where d is the largest degree required. In the Polynomial Calculus proof system one does not need to explicitly write down these polynomials all at once but rather one can give a derivation that demonstrates their existence involving polynomials of low degree along the way. The size of each of these polynomials is also based on this dense representation.

A variety of Nullstellensatz lower bounds are known for the symmetric counting principles but no Nullstellensatz lower bounds are known for the other formulas above. (For example, the odd-charged graph tautologies have easy proofs over $\mathbb{Z}_2$.)

One drawback of the Nullstellensatz system (although not Polynomial Calculus) is that a simple chain of inference of length n using modus ponens requires non-constant degree $\Theta(\log n)$ [**26**]. It is even possible that certain principles may be proved using degree 2 using Polynomial Calculus but require degree $\Omega(\sqrt{n})$ Nullstellensatz proofs [**33**].

Although it is trivial to prove $Count_r^n$ in constant degree in $\mathbb{Z}_r$, degree lower bounds of $n^{\Omega(1)}$ have been shown in $\mathbb{Z}_r$ for PHP_n^m [**12**], $Count_s^n$ for most $s \neq r$ [**23**] and onto-$PHP_n^{n+r^{\omega(1)}}$ [**11**]. So far, all of the Nullstellensatz bounds mentioned were shown using the notion of a dual *design* [**12, 25**]. This is typically a combinatorial construction that guarantees that one can derive a solution to a set of dual equations that express the coefficient of the constant term in $\sum_i P_i \cdot Q_i$ as a linear combination of higher degree coefficients in the indeterminate coefficients of the P_i. Since $\sum_i P_i \cdot Q_i$ is supposed to represent the polynomial 1 this would be impossible. [**23**] introduced a nice technique (used in [**11**]) that makes such designs easier to construct.

Using a more general class of designs the degree bounds for PHP_n^m were were improved to degree $n/2+1$ by Razborov [**77**]. Remarkably, Razborov's result also applies to the Polynomial Calculus proof system and was not only the first non-trivial lower bound shown for that system but also is one of strongest known for the Nullstellensatz system. It involves an explicit computation of the Groebner basis of the ideal generated by the PHP_n^m equations. Razborov also extends this lower bound to obtain stronger, nearly linear, degree lower bounds in Polynomial Calculus for related tautologies as a function of the number of variables.

Recently, Krajicek [**58**] has shown non-constant degree lower bounds for $Count_s^n$ in the Polynomial Calculus proof system over $\mathbb{Z}_r$ by extending the results of Ajtai regarding symmetric linear equations and the structure of representations of the symmetric group [**3**] mentioned earlier.

3.5. $\mathsf{AC}^0[r]$-Frege systems. We have already seen how one can extend AC^0-Frege proofs by adding axioms for counting modulo r. A far more general, and in some sense, more natural way to add the power of modular counting to a proof system is to include it fundamentally in the structure of the objects about which one reasons; that is, to introduce modular counting connectives into the lines of the proofs themselves and new inference rules for manipulating these formulas. $\mathsf{AC}^0[r]$-Frege is precisely such a system. Even the Polynomial Calculus proof system modulo r may be viewed as a subsystem of an $\mathsf{AC}^0[r]$-Frege proof system in which all the modular connectives are at the top [**72**].

At present there are no lower bounds known for $\mathsf{AC}^0[r]$-Frege, even when r is a prime. One program for obtaining such bounds was laid out in [**23**]. where it is shown how to convert an $\mathsf{AC}^0[p]$-Frege proof of F to a Polynomial Calculus proof of a system that involves the polynomials for F plus certain low degree extension polynomials (which mimic the low degree approximations used by [**78, 88**] for $\mathsf{AC}^0[p]$ lower bounds).

3.6. Frege systems and TC^0-Frege systems. Just as $\mathsf{AC}^0[r]$-Frege proofs include counting modulo r as a first-class concept, so TC^0-Frege proofs include counting up to a threshold as a first-class concept. Loosely speaking, TC^0-Frege proofs are proofs where the underlying class of formulas are small-weight, constant-depth threshold formulas. Thus, for example, Cutting Planes proofs can be viewed as a special case of restricted TC^0-Frege proofs.

We only have partial results on TC^0-Frege proofs. Maciel and Pitassi [**66**] have shown proof-theoretic analogues of circuit complexity constructions [**7**] to relate TC^0- and $\mathsf{AC}^0[p]$-Frege proofs. In particular they show how one can convert polynomial-size $\mathsf{AC}^0[2]$-Frege proofs into restricted quasipolynomial-size depth 3 TC^0-Frege proofs.

The potential list of candidate hard problems for TC^0-Frege is quite short, in part because there are efficient TC^0-Frege proofs for so many formulas for which lower bounds are known in other systems and because the basic techniques for dealing with these formulas fundamentally break down. There are polynomial-size TC^0-Frege proofs of all of the symmetric counting principles [**29**] as well as the odd-charged graph principles [**91**]. And, as mentioned above, the interpolation method cannot apply to TC^0-Frege assuming that factoring Blum integers is hard [**21**]. Since TC^0-Frege proofs are special cases of Frege proofs, the same problems apply to Frege proofs as well. Some candidates for hard tautologies for these systems have been suggested in [**18**].

3.7. Optimal proof systems. Research in proof complexity was originally motivated in part as a way of proving $\mathsf{NP} \neq \mathsf{co\text{-}NP}$, by proving superpolynomial lower bounds for increasingly powerful proof systems. An important question is whether such a chain of results will ever actually lead us to a proof of $\mathsf{NP} \neq \mathsf{co\text{-}NP}$. In other words, is there an optimal proof system? This question is quite important and is still open. However, some partial results have been obtained [**67, 61, 16**] relating this existence to the equivalence of certain complexity classes.

3.8. Proof Search. While lengths of proofs are important, it is also important to be able to *find* proofs quickly. Clearly, if we know that a proof system S is not polynomially-bounded, then no efficient deterministic procedure can exist that will produce short proofs of all tautologies. But is it possible to find short proofs of all tautologies that have short proofs? To this end, [**21**] defines a proof system S to be *automatizable* if there exists a deterministic algorithm that takes as input a tautology F, and outputs an S-proof of F in time polynomial in the size of the shortest S-proof of F.

Automatizability is very important for automated theorem proving, and is very similar to an older concept, k-provability. The k-provability problem for a proof system S is as follows. The input is a formula F, and a number k, and the input should be accepted if and only if F has an S-proof of size at most k. Clearly a proof system S is not automatizable if the k-provability problem cannot be approximated to within any polynomial factor.

Which proof systems are automatizable, and for which proof systems is k-provability hard? It has been shown [**6, 22, 54**] that the k-provability problem is NP-hard for essentially every standard propositional proof system, and furthermore using the PCP theorem, that the k-provability problem cannot be approximated to within any constant factor, unless $\mathsf{P} = \mathsf{NP}$.

The above hardness results show that finding good estimates of the proof length is hard, even for very simple proof systems such as Resolution. But it may still be that most proof systems are automatizable. However, under stronger assumptions, one can show that many proof systems are not automatizable. These results are shown by exploiting a connection between interpolation and automatizability. In particular, it can be shown that if a proof system S does not have feasible interpolation, then this implies that S is not automatizable. Thus, feasible interpolation gives us a formal tradeoff between the complexity/strength of S and the ability to *find* short proofs quickly. Using this connection, it has been shown that AC^0-Frege proofs as well as any proof system that can p-simulate AC^0-Frege, is not automatizable, under cryptographic assumptions. (See [**62, 21, 65, 19**].)

Are there any proof systems that are automatizable? Both the Nullstellensatz and Polynomial Calculus proof systems as well as DLL are actually search procedures as well as nondeterministic algorithms. But, unlike DLL, the Nullstellensatz and Polynomial Calculus algorithms are guaranteed to find short proofs if they exist [**13, 33**]; that is, they are automatizable. (Any proof of degree d in n variables may be found using linear algebra in time $n^{O(d)}$.) Nullstellensatz proofs may be exponentially smaller than bounded-depth Frege proofs, but they also may be exponentially larger than Resolution proofs [**33**]. Polynomial Calculus proofs are at worst quasi-polynomially larger than the best proofs under any DLL algorithm and from this one can derive a method of searching for short DLL proofs that is guaranteed to succeed [**33**]. In [**10**], this algorithm is converted to a direct search procedure for such proofs. It appears fruitful to investigate Polynomial Calculus proofs as a theorem-proving tool to see if they can be refined to compete with DLL algorithms. Preliminary results of this form appear in [**33**].

4. Open Problems

Find hard tautologies for TC^0-Frege and Frege. No examples of tautologies are known for which Frege proofs even require a super-linear number of distinct subformulas. One difficult problem that is faced when trying to prove lower bounds for Frege or Extended Frege systems is that there is a surprising lack of hard candidate tautologies. Most of the lower bounds proven thus far have been for various counting principles, all of which have polynomial-size TC^0-Frege or Frege proofs. (Even the minterm-maxterm formulas can be viewed as an application of a counting principle.)

Some candidate hard examples have been suggested in [**18**] including random k-CNF formulas. Another family of examples that is of particular interest for complexity theory, are at least as hard to prove as many of the classes of formulas discussed in this paper, but are only believed to be tautological are particular formulas given by Razborov [**80**] stating that NP is not contained in P/poly.

There is an even larger gap in tautologies that seem to separate extended Frege from Frege systems. In fact, we know of no convincing combinatorial tautologies that might have polynomial-size extended Frege proofs, but require exponential-size Frege proofs. In [**18**], several tautologies based on linear algebra are suggested to give a quasipolynomial separation between extended Frege and Frege systems. A very simple such example, suggested by Cook and Rackoff, is the propositional form of the Boolean matrix product implication $AB = I \Rightarrow BA = I$.

How hard are random k-CNF formulas? The only lower bounds known for unsatisfiability proofs of random formulas are for forms of Resolution. What about other proof systems such as: bounded-depth Frege proofs? cutting planes proofs? Nullstellensatz or polynomial calculus proofs? The absence of random unsatisfiable formulas in the list of lower bounds for systems other than Resolution is quite noteworthy, especially given the lack of good upper bounds for proofs of these formulas in any system, even extended-Frege.

Many NP-complete graph problems are easy on the average for the natural random graph probability distributions. Random k-CNF formulas under the analogous probability distributions seem surprisingly hard in the region of probabilities for which the formulas are likely unsatisfiable [**32, 9**]. Is k-UNSAT hard on the average in this sense?

For example, the best upper bound for any search algorithm for unsatisfiability proofs of random m-clause n-variable 3-CNF formulas is $2^{O(n^2/m)}$ with probability $1 - o(1)$ in n and this is tight for a class of DLL algorithms [**9**]. For Resolution the lower bound for this problem is nearly $2^{\Omega(n^5/m^4)}$ with probability $1 - O(1)$ in n [**9**]. Can these be improved?

Superpolynomial lower bounds for $\mathsf{AC}^0[r]$ Frege?. Studying $\mathsf{AC}^0[r]$-Frege is a natural next step in proving lower bounds for proof systems, in particular when r is a prime. We already mentioned the program for obtaining such lower bounds in [**23**]. For this system we do have a natural candidate for a hard tautology, namely $Count_p^n$ for prime $p \neq r$. Such a lower bound would further the circuit/proof system correspondence by extending proof complexity lower bounds to the natural analogue of the Razborov-Smolensky circuit lower bounds [**78, 88**].

Polynomial Calculus in a theorem prover? Better Resolution proof search? Designing efficient theorem provers for the propositional calculus is an important practical question. To date, DLL algorithms are the champion theorem provers although they are theoretically quite weak as proof systems. A recent challenger seems promising: A variant of the Groebner basis algorithm has been used to find Polynomial Calculus proofs [**33**] and build a fairly efficient theorem prover. Can this be tuned to compete with DLL algorithms?

Clegg *et al.* [**33**] also give simulations of DLL and Resolution by the Polynomial Calculus. One method of improving the competitiveness of Polynomial Calculus as a theorem prover would be to improve these simulations. This is also related to the question of improving the more direct methods for proof search for Resolution and DLL [**10, 17**] that were inspired by these Polynomial Calculus simulations.

New proof systems from NP-complete problems. An appealing direction in proof complexity is the possibility of using natural domains that contain NP-hard problems to seek out new and interesting proof systems which reason about objects from radically different domains from Boolean formulas. Cutting planes come from integer programming, Nullstellensatz and Polynomial Calculus systems come from systems of polynomial equations. Pitassi and Urquhart [**73**] considered the Hajos calculus for non-3-colorability which they found, surprisingly, to be equivalent to extended-Frege proofs. It is likely that there is more to be mined in this search.

Weak pigeonhole principle and the limits of Resolution. It is known that for $n < m < n^2/\log n$, PHP_n^m requires superpolynomial-size Resolution proofs [**52, 27**]. Originally it was conjectured that for any $m > n$, any Resolution refutation of PHP_n^m would require size exponential in n. However, this conjecture was shown to be false for large enough m [**24**]. Moreover, it appears that the standard method, the bottleneck counting technique, cannot be applied to obtain lower bounds for $m > n^2$. An interesting open problem is to prove lower bounds for PHP_n^m for $m > n^2$. This would likely give rise to a new lower bound method for Resolution. Additionally the complexity of the weak pigeonhole principle in various proof systems is interesting in its own right: it is known [**71**] that one can prove the existence of infinitely many primes in systems as weak as polynomial-size, bounded-depth Frege, assuming the weak-pigeonhole principle as an axiom schema. More generally, the weak pigeonhole principle can be used to carry out most combinatorial counting arguments, and is closely connected to approximate counting.

Partial results for the weak pigeonhole principle have recently been obtained [**82**]. There is a very tight connection between regular Resolution refutations and read-once branching programs which generalizes the equivalence between tree resolution and DLL mentioned earlier. Let F be an unsatisfiable formula in conjunctive normal form. The search problem associated with F takes as input a truth assignment τ to the underlying variables of F, and outputs a clause in F that is set to false by τ. Krajíček [**60**] has shown that for any F, the minimal size Resolution refutation of F is essentially equivalent to the minimal size read-once branching program to solve the related search problem. This idea was exploited in [**82**] to obtain some restricted lower bounds for Resolution proofs of weak versions of the pigeonhole principle.

Another, even older, principle for which no superpolynomial Resolution lower bounds are known is the domino-tiling principle. More details about this old problem can be found in [**90**].

Lovasz-Schrijver proof systems. A variety of inference systems for 01-programming are described by Lovasz and Schrijver [**64**] in a paper that is primarily concerned with their implications for linear programming. Like cutting planes these proof systems represent statements using systems of linear inequalities, but unlike cutting planes, they replace rounding by an ability to take linear combinations of derived degree 2 terms in intermediate steps obtained by multiplying certain inequalities and including the equations $x^2 = x$ provided all degree 2 terms cancel. (There are several versions depending on whether one can (1) add the squares of arbitrary linear terms to the appropriate side of the inequalities or (2) multiply inequalities by x or $1 - x$ or, more generally, (3) multiply by any linear term previously shown to be positive.) Lovasz and Schrijver prove a number of properties of their systems that allow one to precisely determine the depth of the proofs involved but they do not consider the issue of proof size directly.

It turns out that, despite the absence of the rounded division of cutting planes, one can easily simulate Resolution and prove the pigeonhole principle in polynomial size in their weakest system. (To see an example of the system in action consider that if $1 - a - b \geq 0$, $1 - a - c \geq 0$, $1 - b - c \geq 0$ holds for $0 \leq a \leq 1$, $0 \leq b \leq 1$, $0 \leq c \leq 1$ then at most one of a, b, c is 1 and so $a+b+c \leq 1$. This is obtained by computing $0 \leq a(1-a-b) = a - a^2 - ab = -ab$, $0 \leq a(1 - a - c) = a - a^2 - ac = -ac$, and $0 \leq (1 - a)(1 - b - c) = 1 - a - b - c + ab + ac$. Adding these inequalities leads to the desired result.)

However other problems are less clear. In particular, Lovasz (private communication) suggested the problem of deriving the maximum independent set size of graphs that can be expressed as the line graphs of odd cliques. One can easily show that this problem is completely equivalent to the parity principle and requires linear depth in the number of vertices, but this is far from either an algorithm or a lower bound. This question is intimately tied to the question of whether these systems can simulate cutting planes efficiently. Assuming that they cannot, it may be even more interesting to consider what the combination of these new systems with cutting planes is capable of.

Odd-charged graphs hard for AC^0-Frege? Urquhart has suggested that a study of Tseitin's odd-charged graph tautologies [**89**] for appropriate graphs [**91**] in bounded-depth Frege systems might also lead to lower bounds for random formulas since it appears very difficult to apply Ajtai's program to them. No results are know for these even on depth 2 Frege systems.

More general lower bound technique for Cutting Planes? The interpolation method has been successfully applied to obtain unconditional lower bounds for Cutting Planes proofs. However, we are quite far from understanding more generally what types of tautologies are hard for Cutting

Planes. In particular, are random formulas hard? Other families of tautologies possibly hard for Cutting Planes are the odd-charged graph tautologies.
Exponential bounds for $Count_p^n$/onto-PHP in Polynomial Calculus The lower bounds of [**58**] are barely super-polynomial and it seems unlikely that the methods used can produce much larger lower bounds. The lower bounds of Razborov [**77**] involve a direct computation of the low degree elements of the Groebner basis of the PHP_n^m polynomials. This computation relies heavily on the independence of the degree one PHP_n^m polynomials. All the other counting principles above do not have this property and it does not appear that the same method can be applied to them. Is there a more general methodology that does not require direct computation of the basis?
Probabilistically checkable algebraic proofs. The Nullstellensatz and Polynomial Calculus proof systems use the dense representation of multivariate polynomials. What happens if one modifies these proof systems to manipulate polynomials as straight-line programs that compute them rather than writing them out explicitly? One difficulty is that testing the equality of polynomials represented this way is not known to be efficiently computable deterministically. However, there are efficient probabilistic algorithms to do this check [**85**]. Such a proof system would lie outside our proof system definition since the verification predicate S would only be probabilistically checkable. See [**72**] for more details.
Unsatisfiability threshold for random k-SAT. Although this is not a proof complexity question *per se* it is of interest in understanding the proof complexity of random k-CNF formulas. There have been a number of papers analyzing the satisfiability properties of these formulas as a function of their clause-variable ratios. Recently, it has been shown that there is a sharp threshold behavior for such formulas [**44**] but it is not known precisely where such a threshold lies or even if it approaches some fixed limit. In general it is known that it lies between $2^k/k$ and $2^k \ln 2$ [**32, 31**] and for $k = 3$ it is known to lie between 3.003 and 4.598 [**45, 57**] and is conjectured to be around 4.2 [**56**]. A related question is whether or not the satisfiability problem is easy right up to the threshold.
Natural Proofs in Proof Complexity? In circuit complexity, Razborov and Rudich [**81**] suggest that, subject to some plausible cryptographic conjectures, current techniques will be inadequate for obtaining super-polynomial lower bounds for TC^0-circuits. To this point, proof complexity has made steady progress at matching the superpolynomial lower bounds currently known in the circuit world (albeit using different techniques), and the major remaining analogous result (a lower bound for $\mathsf{AC}^0[p]$-Frege proofs) also may be within reach. Unlike the circuit world, however, there is no analogue of Shannon's counting argument for size lower bounds for random functions and there does not seem any inherent reason for TC^0-Frege to be beyond current techniques. While it is true that one can show the failure of TC^0-Frege interpolation (also depending on cryptographic conjectures), this

applies to AC^0-Frege as well for which we do have lower bounds. Is there any analogue of natural proofs in proof complexity? (Razborov [**79**] has looked at the quite different question of examining particular proof systems and showing that any efficient proofs of lower bounds for circuits using such systems automatically naturalize.)

[1] M. Ajtai. The complexity of the pigeonhole principle. *Combinatorica*, 14(4):417–433, 1994.

[2] M. Ajtai. The independence of the modulo p counting principles. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 402–411, Montréal, Québec, Canada, May 1994.

[3] M. Ajtai. Symmetric systems of linear equations modulo p. Technical Report TR94-015, Electronic Colloquium in Computation Complexity, `http://www.eccc.uni-trier.de/eccc/`, 1994.

[4] Miklós Ajtai. The complexity of the pigeonhole principle. In *29th Annual Symposium on Foundations of Computer Science*, pages 346–355, White Plains, NY, October 1988. IEEE.

[5] Miklós Ajtai. Parity and the pigeonhole principle. In Samuel R. Buss and P. J. Scott, editors, *Feasible Mathematics*, pages 1–24, A Mathematical Sciences Institute Workshop, Ithaca, NY, 1990. Birkhäuser.

[6] M. Alekhnovich, S. Buss, S. Moran, and T. Pitassi. Minimum Propositional proof length is NP-hard to linearly approximate. Manuscript. 1998.

[7] E. Allender and U. Hertrampf. Depth reduction for circuits of unbounded fan-in. *Information and Computation*, 112(2):217–238, August 1994.

[8] S. Arora, C. Lund, Rajeev Motwani, M. Sudan, and Mário Szegedy. Proof verification and hardness of approximation problems. In *Proceedings 33rd Annual Symposium on Foundations of Computer Science*, Pittsburgh, PA, October 1992. IEEE.

[9] P. Beame, R. Karp, T. Pitassi, and M. Saks. On the complexity of unsatisfiability of random k-CNF formulas. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, Dallas, TX, May 1998.

[10] P. Beame and T. Pitassi. Simplified and improved resolution lower bounds. In *37th Annual Symposium on Foundations of Computer Science*, pages 274–282, Burlington, VT, October 1996. IEEE.

[11] P. Beame and S. Riis. More on the relative strength of counting principles. In P. Beame and S. Buss, editors, *Proof Complexity and Feasible Arithmetics*, DIMACS. American Mathematical Society, 1998. To appear.

[12] Paul W. Beame, Stephen A. Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 303–314, Las Vegas, NV, May 1995.

[13] Paul W. Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proc. London Math. Soc.*, 73(3):1–26, 1996.

[14] Paul W. Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, Pavel Pudlák, and Alan Woods. Exponential lower bounds for the pigeonhole principle. In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, pages 200–220, Victoria, B.C., Canada, May 1992.

[15] Paul W. Beame and Toniann Pitassi. An exponential separation between the matching principle and the pigeonhole principle. In *8th Annual IEEE Symposium on Logic in Computer Science*, pages 308–319, Montreal, Quebec, June 1993.

[16] S. Ben-David and A. Gringauze. On the existence of propositional proof systems and oracle-relativized propositional logic. Technical Report TR98-021, Electronic Colloquium in Computation Complexity, `http://www.eccc.uni-trier.de/eccc/`, 1998.

[17] E. Ben Sasson and A. Wigderson. Private communication.

[18] M. Bonet, S. Buss, and T. Pitassi. Are there hard examples for Frege systems? In P. Clote and J. Remmel, editors, *Feasible Mathematics II*, pages 30–56. Birkhauser, 1995.

[19] M. Bonet, C. Domingo, and R. Gavalda. No feasible interpolation or automatization for AC^0-Frege proof systems. Manuscript. 1998.

[20] M. Bonet, T. Pitassi, and R. Raz. Lower bounds for cutting planes proofs with small coefficients. *Journal of Symbolic Logic*, 62(3):708–728, September 1997.

[21] M. Bonet, T. Pitassi, and R. Raz. No feasible interpolation for TC^0 frege proofs. In *38th Annual Symposium on Foundations of Computer Science*. IEEE, October 1997.

[22] S. Buss. On Godel's theorems on lengths of proofs II: Lower bounds for recognizing k symbol provability. In P. Clote and J. Remmel, editors, *Feasible mathematics II*, pages 57–90. Birkhauser-Boston, 1995.

[23] S. Buss, R. Impagliazzo, J. Krajíček, P. Pudlák, A. A. Razborov, and J. Sgall. Proof complexity in algebraic systems and bounded depth Frege systems with modular counting. *Computation Complexity*, 6(3):256–298, 1997.

[24] S. Buss and T. Pitassi. Resolution and the weak pigeonhole principle. In *Proceedings of Computer Science Logic*, Lecture Notes in Computer Science. Springer-Verlag, 1997.

[25] S. R. Buss. Lower bounds on Nullstellensatz proofs via designs. In P. W. Beame and S. R. Buss, editors, *Proof Complexity and Feasible Arithmetics*, DIMACS, pages 59–71. American Math. Soc, 1997.

[26] S. R. Buss and T. Pitassi. Good degree bounds on Nullstellensatz refutations of the induction principle. In *Proceedings of the Eleventh Annual Conference on Computational Complexity (formerly: Structure in Complexity Theory)*, pages 233–242, Philadelphia, PA, May 1996. IEEE.

[27] S. R. Buss and G. Turan. Resolution proofs of generalized pigeonhole principles. *Theoretical Computer Science*, 62(3):311–317, December 1988.

[28] Samuel R. Buss. *Bounded Arithmetic*. Bibliopolis, Napoli, 1986. Volume 3 of Studies in Proof Theory.

[29] Samuel R. Buss. Polynomial size proofs of the pigeonhole principle. *Journal of Symbolic Logic*, 57:916–927, 1987.

[30] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4, 1973.

[31] V. Chvátal and B. Reed. Mick gets some (the odds are on his side). In *33nd Annual Symposium on Foundations of Computer Science*, pages 620–627, Pittsburgh, PA, October 1992. IEEE.

[32] V. Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, 1988.

[33] M. Clegg, J. Edmonds, and R. Impagliazzo. Using the Gröbner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pages 174–183, Philadelphia, PA, May 1996.

[34] S. Cook and D. Mitchell. Finding hard instances of the satisfiability problem: A survey. In *DIMACS Series in Theoretical Computer Science*, 1997.

[35] Stephen A. Cook. The complexity of theorem proving procedures. In *Conference Record of Third Annual ACM Symposium on Theory of Computing*, pages 151–158, Shaker Heights, OH, May 1971.

[36] Stephen A. Cook. Feasibly constructive proofs and the propositional calculus. In *Proceedings of the Seventh Annual ACM Symposium on Theory of Computing*, pages 83–97, Albuquerque, NM, May 1975.

[37] Stephen A. Cook and Robert A. Reckhow. Time bounded random access machines. *Journal of Computer and System Sciences*, 7(4):354–375, 1973.

[38] W. Cook, C. R. Coullard, and G. Turan. On the complexity of cutting plane proofs. *Discrete Applied Mathematics*, 18:25–38, 1987.

[39] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Communications of the ACM*, 5:394–397, 1962.

[40] M. Davis and H. Putnam. A computing procedure for quantification theory. *Communications of the ACM*, 7:201–215, 1960.

[41] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost *NP*-complete. In *Proceedings 32nd Annual Symposium on Foundations of Computer Science*, pages 2–12, San Juan, Puerto Rico, October 1991. IEEE.

[42] G. Frege. *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinben Denkens*. Nebert, Halle, 1879.

[43] G. Frege. *Grundgesetze der Arithmetik*. Nebert, Halle, 1893,1901.

[44] E. Friedgut. Necessary and sufficient conditions for sharp thresholds of graph properties, and the k-sat problem. Preprint, May 1997.

[45] A. Frieze and S. Suen. Analysis of two simple heuristics on a random instance of k-SAT. *Journal of Algorithms*, 20(2):312–355, 1996.

[46] Xudong Fu. *On the complexity of proof systems*. PhD thesis, University of Toronto, 1995.

[47] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, April 1984.

[48] O. Gabber and Z. Galil. Explicit constructions of linear size superconcentrators. In *20th Annual Symposium on Foundations of Computer Science*, pages 364–370, New York, NY, 1979. IEEE.

[49] Z. Galil. On the complexity of regular resolution and the Davis-Putnam procedure. *Theoretical Computer Science*, 4:23–46, 1977.

[50] A. Goerdt. Regular resolution versus unrestricted resolution. *SIAM Journal on Computing*, 22(4):661–683, 1993.

[51] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958.

[52] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–305, 1985.

[53] A. Haken and S. A. Cook. An exponential lower bound for the size of monotone real circuits. Preprint, 1995.

[54] K. Iwama. Complexity of finding short resolution proofs. In I. Privara and P. Ruzicka, editors, *Lecture Notes in Computer Science 1295*, pages 309–318. Springer-Verlag, 1997.

[55] G. D. James. *The Representation Theory of the Symmetric Groups*. Number 682 in Lecture Notes in Mathematics. Springer-Verlag, 1978.

[56] S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random formulas. *Science*, 264:1297–1301, May 1994.

[57] L. M. Kirousis, E. Kranakis, and D. Krizanc. Approximating the unsatisfiability threshold of random formulas. In *Proceedings of the Fourth Annual European Symposium on Algorithms*, pages 27–38, Barcelona, Spain, September 1996.

[58] J. Krajíček. On the degree of ideal membership proofs from uniform families of polynomials over finite fields. Manuscript, 1997.

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

[60] J. Krajíček. *Bounded Arithmetic, Propositional Logic and Complexity Theory*. Cambridge University Press, 1996.

[61] J. Krajicek and P. Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *J. Symbolic Logic*, 54(3):1063–1079, 1989.

[62] K. Krajicek and P. Pudlák. Some consequences of cryptographic conjectures for S_2^1 and EF. In D. Leivant, editor, *Logic and Computational Complexity*, pages 210–220. Springer-Verlag, 1995.

[63] J. Krajíček, P. Pudlák, and A Woods. Exponential lower bounds to the size of bounded depth Frege proofs of the pigeonhole principle. *Random Structures and Algorithms*, 7(1), 1995.

[64] L. Lovasz and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optimization*, 1(2):166–190, 1991.

[65] A. Maciel and T. Pitassi. Non-automatizability of bounded-depth Frege proofs. Manuscript. 1998.

[66] A. Maciel and T. Pitassi. On $ACC^0[p^k]$ frege proofs. In *Proceedings of the Twenty Ninth Annual ACM Symposium on Theory of Computing*, pages 720–729, May 1997.

[67] J. Meßner and J. Toran. Optimal proof systems for propositional logic and complete sets. In *15th Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, Paris, France, February 1998. Springer-Verlag.

[68] D. Mitchell. Hard problems for csp algorithms. In *Proceedings from AAAI '98*, 1998.

[69] D. Mundici. A lower bound for the complexity of Craig's interpolants in sentential logic. *Archiv fur Mathematische Logik und Grundlagenforschung*, 23(1-2):27–36, 1983.

[70] J. Paris and A. Wilkie. Counting problems in bounded arithmetic. In *Methods in Mathematical Logic: Proceedings of the 6th Latin American Symposium on Mathematical Logic 1983*, volume 1130 of *Lecture notes in Mathematics*, pages 317–340, Berlin, 1985. Springer-Verlag.

[71] 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.

[72] T. Pitassi. Algebraic propositional proof systems. In *DIMACS Series in Discrete Mathematics*, volume 31, pages 215–243. American Math. Soc, 1997.

[73] T. Pitassi and A. Urquhart. The complexity of the Hajós calculus. In *33nd Annual Symposium on Foundations of Computer Science*, pages 187–196, Pittsburgh, PA, October 1992. IEEE.

[74] Toniann Pitassi, Paul W. Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3(2):97–140, 1993.

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

[76] P. Pudlák and J. Sgall. Algebraic models of computation and interpolation for algebraic proof systems. In P. W. Beame and S. R. Buss, editors, *Proof Complexity and Feasible Arithmetics*, DIMACS, pages 279–295. American Math. Soc, 1997.

[77] A. A. Razborov. Lower bounds for the polynomial calculus. November 1996.

[78] A. A. Razborov. Lower bounds for the size of circuits with bounded depth with basis $\{\wedge, \oplus\}$. *Mat. Zametki*, 1987.

[79] A. A. Razborov. Bounded arithmetic and lower bounds in Boolean complexity. In P. Clote and J. Remmel, editors, *Feasible Mathematics II*, pages 344–386. Birkhauser, 1995.

[80] A. A. Razborov. Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic. *Izvestiiya of the RAN*, 59:201–224, 1995.

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

[82] A. A. Razborov, A. Wigderson, and A. C. Yao. Read-once branching programs, rectangular proofs of the pigeonhole principle and the transversal calculus. In *Proceedings of the Twenty Ninth Annual ACM Symposium on Theory of Computing*, pages 739–748, El Paso, TX, May 1997.

[83] Søren Riis. *Independence in Bounded Arithmetic*. PhD thesis, Oxford University, 1993.

[84] J. A. Robinson. A machine oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[85] J. T. Schwartz. Probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, pages 701–717, 1980.

[86] M. Sipser. The history and status of the P versus NP question. In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, pages 603–618, Victoria, B.C., Canada, May 1992.

[87] Michael Sipser. A complexity theoretic approach to randomness. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, Boston, MA, April 1983.

[88] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 77–82, New York, NY, May 1987.

[89] G. S. Tseitin. On the complexity of derivation in the propositional calculus. In A. O. Slisenko, editor, *Studies in Constructive Mathematics and Mathematical Logic, Part II*. 1968.

[90] A. Urquhart. Manuscript, 1998.

[91] A. Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, 1987.

[92] A. Urquhart. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 1(4):425–467, December 1995.

Computer Science and Engineering, University of Washington, Seattle, WA 98195-2350

E-mail address: beame@cs.washington.edu

Computer Science, University of Arizona Tucson, AZ 85721

E-mail address: toni@cs.arizona.edu