Strong ETH Holds for Regular Resolution

Chris Beck*
Princeton University
cbeck@princeton.edu

Russell Impagliazzo†
University of California, San Diego
russell@cs.ucsd.edu

ABSTRACT

We obtain asymptotically sharper lower bounds on resolution complexity for k-CNF's than was known previously. We show that for any large enough k there are k-CNF's which require resolution width $(1-\tilde{O}(k^{-1/4}))n$, regular resolution size $2^{(1-\tilde{O}(k^{-1/4}))n}$, and
general resolution size $(3/2)^{(1-\tilde{O}(k^{-1/4}))n}$.

Categories and Subject Descriptors:

Categories and Subject Descriptors

F.0 [**Theory of Computation**]: GeneralI.2.3[Artificial Intelligence]: Deduction and Theorem Proving;

General Terms: Theory

Keywords: Proof complexity, resolution, lower bounds, quantum classical separation

1. INTRODUCTION

The SAT problem is a canonical NP-complete problem. Non-trivial algorithms for SAT have ramifications both for the theory of computation and in applications such as hardware and protocol verification and planning. Despite a huge amount of attention from both theoretical and empirical perspectives, the exact difficulty of SAT remains somewhat mysterious. While quantitative improvements in SAT algorithms continue to be made, in many ways a wide variety of different algorithmic techniques have yielded similiar time bounds in a qualitative sense. The Exponential Time Hypothesis (ETH) and Strong Exponential Time Hypothesis (SETH) were introduced to give a precise meaning to the question of whether further improvements will be only quantitative, or substantially different [16]. These hypotheses have beeen shown to have other significant consequences in complexity, such as limits on the k-SUM problem from computational geometry, exponential algorithms for other NP-complete problems, limits to improving algorithms in parameterized complexity, and so on. Formally, ETH is the statement that, for $k \geq 3$, k-CNF SAT does not have algorithms running in time $2^{o(n)}$. Closely related is the strong ETH – that the savings possible for k-SAT goes to 0 as k goes to infinity, or, equivalently, that $k(n)$-SAT does not have deterministic algorithms in time $2^{(1-o(1))n}$ for any function $k(n) = \omega(1)$. That is, it is not even possible to get a constant polynomial advantage over brute force search that is indendent of k.

Both forms of the conjecture have a natural appeal, although there is admittedly little formal evidence for either. However, there are an increasing variety of interesting and non-trivial algorithms for SAT that seem to use unrelated algorithmic techniques ([19, 18, 14, 27, 15]), but all have roughly the same savings over exhaustive search : $\Theta(1/k)$ fractional savings over exhaustive search for k-SAT.

Empirically, it has also been observed that even tuned SAT solvers that solve 3-SAT formulas with millions of variables have difficulty with even small random k-SAT formulas for moderate k, such as 5 or 6 [25]. So SETH seems at least to be true for commonly used algorithms. Since we do not know how to show that problems in NP require even super-polynomial worst-case complexity, it seems that we are incredibly distant from any possible proof of ETH or SETH.

The challenge of lower bounds is to reason about arbitrary algorithms, including ones that are counter-intuitive. We might be able to confirm these hypotheses for at least some categories of algorithms that work in an intuitive fashion and are similar to those in use. Propositional proof complexity offers a general technique to do this. Many algorithmic techniques, when run on an unsatisfiable instance, implicitly define a "proof" that no solution exists, that can be formalized in a corresponding proof system[1]. Then size

*Research supported by NSF grants CCF-0832797, CCF-1117309, The Simons Foundation.
†Research supported by NSF grants DMS-0835373, CCF-0832797, and The Oswald Veblen Fund.

STOC'13, June 1âĂŞ4, 2013, Palo Alto, California, USA.

[1]While of course the computation transcript of any correct deterministic algorithm is in some sense a proof, in general this need not correspond to a propositional proof of any kind. It simply turns out that most if not all of the algorithms we care about and use can be broken down into a series of logical steps and deductions; one explanation is that generally we are taught to write and think about programs in a way so that there will be a simple proof of correctness of the algorithm at the end, and if there is then it will induce logical proofs of correctness for every input which roughly follow the transcript. However, if a turing machine correctly decides a language, but its correctness is e.g. independent

lower bounds on the minimum proofs for a tautology in the proof system provide a lower bound on the time required by any algorithm in the family on the negation of the tautology. Using this method, ETH has been established for any algorithm that can be formalized within the resolution system, which includes many of the most successful empirical SAT-solving techniques. More precisely, lower bounds of the form $2^{\Omega(n)}$ are known for many natural proof systems like Resolution and Polynomial Calculus [30, 17]. which correspond naturally to important families of SAT algorithms.(Weakly exponential lower bounds $\exp(n^{\epsilon})$ are also known for more exotic proof systems.) So we at least know that to negate ETH, we need to go beyond some of the standard algorithm design methods.

This raises the question of whether we can also get results establishing Strong ETH for similar classes of algorithms. A first result along these lines was by Pudlák and Impagliazzo [20], who showed that tree-like resolution requires size $2^{(1-\epsilon)n}$ for any ϵ, for k-CNFs of size cn where c, k are functions of ϵ. Here, we get a similar lower bound for regular resolution, a sub-system of resolution that is strictly more powerful than tree-like resolution and which formalizes algorithms using the Davis-Putnam procedure [8]. Specifically, we show that there are k-CNF formulas which require regular resolution proofs of size $2^{(1-\tilde{O}(k^{-1/4}))n}$. In particular, we get a somewhat improved and simplified version of the Pudlák-Impagliazzo lower bound. While we have not been able to show the same for general resolution, we do get a substantially improved exponential lower bound for general resolution, which approaches $(3/2)^n$ as k grows. An interesting interpretation is that this class of algorithms are now provably slower than Grover's quantum SAT algorithm [11].

The exact complexity of SAT has taken on an even greater significance in theoretical computer science due to the recent results of Williams [31], that show that even minute savings for circuit SAT can be used to prove circuit lower bounds. While this holds for general circuit SAT rather than k-SAT, we can often relate SAT problems for different classes of circuits. For example, if the AC^0 SAT algorithm of Impagliazzo, Matthews and Paturi [15] were to be substantially improved, it can be proved using Williams' results that $NEXP \not\subseteq NC^1$.

2. TECHNIQUES

Resolution has been intensively studied at least since the work of Davis and Putnam [8] in the early sixties. Despite its apparent simplicity, no exponential lower bounds were known until Haken's result [12] in 1985. Today there remain only a small number of techniques to give lower bounds in Resolution – Random Restrictions [12, 2], the Size-Width Tradeoffs [4], and the Pseudowidth technique which originally appeared in work of Raz [21] and was further developed by Razborov [22, 23].

One of the fundamental building blocks of previous lower bounds research has been resolution width lower bounds for systems of $\mathbb{F}_2$-linear equations; whether studied in the form of Tseitin tautologies, or random k-XOR CSPs, this result as appears in [4, 5] is essential to a great deal of subsequent work, in polynomial calculus [3], Lasserre hierarchy lower bounds [26, 10], and other results. Generally speaking, expanding systems of $\mathbb{F}_2$ linear equations require resolution width $\Omega(n)$. However, the width to refute these is not $(1-\epsilon)n$ as one might hope (for proving lower bounds), but rather there is always an upper bound of $n/2 + o(n)$. Ben-Sasson and Impagliazzo [3] gave a probabilistic construction of a width $n/2$, size $2^{n/2}$ resolution refutation of any such system, based on adding (a subset of) the $\mathbb{F}_2$ equations in a random order to obtain $1 = 0$ and simulating the linear algebra proof in resolution. They showed that with high probability, no intermediate equation has more than $n/2 + o(n)$ variables, due to random cancellations in the sum, thus the simulation results in a resolution proof of the claimed parameters.

Impagliazzo and Pudlák's result is also for a family of $\mathbb{F}_2$ linear equations, so to obtain size lower bounds of $2^{(1-\epsilon)n}$ they had to do significant work to overcome the fact that the width lower bounds are only $n/2$. They analyzed proof size via a Prover Adversary game which they introduced, and their technique works by considering not just the widest clause in the proof, but also for a series of subsequent smaller clauses which occur in the proof. They are able to show that the total combined width of all such clauses encountered approaches n, and their technique exploits this (implicitly) to obtain their lower bound.

In this paper, we consider equations over $\mathbb{F}_p$ rather than $\mathbb{F}_2$. When we add random linear combinations of equations over $\mathbb{F}_p$, a variable cancels with probability only $1/p$ rather than $1/2$, so the construction which gave a width upper bound of $n/2$ for $\mathbb{F}_2$ in this case can only yield $(1-1/p)n$. Thus it is natural to guess that as p gets large the true width will approach n. One apparent drawback of this is that encoding $\mathbb{F}_p$ equations as boolean CSPs can result in significant complications, and it seems inevitable that one will have to think hard about partially constrained $\mathbb{F}_p$ linear systems and technical results from additive combinatorics. By a judicious choice of encoding scheme, we manage to avoid this and obtain an unexpectedly simple proof.

The width lower bound $(1-\epsilon)n$ which we thus obtain immediately implies the tree-like size lower bounds which we desire. To extend this to DAG-like proof systems, a natural idea is to employ a generalization of the Impagliazzo Pudlák prover adversary game [20] which was developed in [1], there with the goal of sharp time space tradeoffs in Regular Resolution. In this argument, a probabilistic adversary interacts with the proof, inducing a distribution of random paths through the proof DAG. A counting argument based on this can be used to obtain size lower bounds, in a manner similar to the bottleneck counting argument first introduced by Haken [12]. In the full result of [1], this adversary is replaced with a random restriction argument in order to obtain results for General Resolution. In this work, we succeeded in adapting the techniques there to obtain sharp size lower bounds in regular resolution, but also we managed to dramatically simplify it in this context.

Finally, we introduce new random restriction techniques which are useful to get stronger lower bounds in general resolution. Rather than adhering to the usual paradigm of us-

of ZFC or similar, we shouldn't necessarily expect that the transcripts of this turing machine correspond to intelligible proofs. This is one way that the goals of proof complexity seem less ambitious than the goals of some lower bounds areas, since it only applies to algorithms that are at least somewhat intuitive. On the other hand, proof complexity essentially permits algorithms which are nondeterministic, so there is also some added difficulty.

ing random restrictions to kill wide clauses, we examine the width lower bound more carefully to give a tighter analysis.

In the next section we define the relevant proof systems and the preliminaries which we will need. In section (3), we prove the width lower bound, which implies the tree-like size bound. In section (4), we give an overview of the techniques for the main result, and together with a lemma from section (3) deduce the size lower bound for resolution.

3. PRELIMINARIES

3.1 Basic Definitions

We consider Boolean formulas over a set of variables $\{x_1, \ldots, x_n\}$. As usual, a literal is a Boolean variable x_i or its negation $\overline{x_i}$, a clause is a disjunction of literals, and a CNF is a conjunction of clauses. We think of clauses as being specified by their sets of literals, and CNFs as specified by their sets of clauses. For a clause C, we use the notation $\mathrm{Vars}(C)$ for the set of variables appearing in C. The *width* $w(C)$ of a clause C is $|\mathrm{Vars}(C)|$ and the width of a set or sequence of clauses F, is the maximum width of clauses in F. The *size* of a CNF formula F is the total number of literal occurrences in the formula, i.e., $\sum_{C \in F} w(C)$.

One of the simplest and most widely studied propositional proof systems is resolution, which operates with clauses and has one rule of inference, the resolution rule: $\frac{A \vee x \qquad B \vee \overline{x}}{A \vee B}$. We say that the variable x is *resolved* in this instance of the resolution rule. A *resolution refutation* of a CNF formula (a set of clauses) is a sequence of clauses ending in the empty clause $\perp$, (representing the constant truth value "false"), each of which is either one of the clauses of the formula (an "axiom") or follows from two earlier clauses via the resolution rule. (The term *resolution proof* is used more generally to refer to any inference of this sort that may not necessarily result in $\perp$.) Every resolution proof naturally corresponds to a directed acyclic graph (DAG), termed the *proof DAG*, in which every clause derived via the resolution inference rule has a directed edge between a derived clause and each of its antecedents, oriented to show dependence. (Note that, formally, a resolution proof corresponds to one of possibly many topological sorts of its proof DAG.)

The *size* or *length* of a resolution proof is the total number of clauses in the proof.

A resolution proof is *tree-like* if its proof DAG has the structure of a tree. A resolution proof is *regular* if along each path in the proof DAG, each variable is resolved at most once. The unrestricted model is often called *general* resolution for contrast with regular and tree-like resolution.

It is easy to see that a tree-like proof of minimum size is regular without loss of generality.

Clauses are permitted to appear multiple times in a resolution proof; in general resolution this is unnecessary when only proof size is a concern, but in restricted forms this can become important.

Resolution is *sound and complete* in that every CNF formula is unsatisfiable if and only if it has a (tree-like) resolution refutation.

A *restriction* is a partial assignment of truth values to variables of a formula, resulting in some simplification. Formally a restriction is a mapping $\rho : X \to \{0, 1, \star\}$. Restrictions on X can be identified with partial assignments on X by viewing unassigned inputs as being mapped to $\star$ and vice versa. For a partial assignment σ with $\sigma(x_i) = \star$ for some variable x_i, we will sometimes use the notation e.g. $\sigma \cup \{x_i = 0\}$ to denote the same partial assignment with $x_i \mapsto 0$ instead.

The restriction of a clause C by ρ, denoted by $C|_\rho$ is the clause obtained from C by setting the value of each $x \in \rho^{-1}(\{0, 1\})$ to $\rho(x)$, and leaving each $x \in \rho^{-1}(\star)$ as a variable. The restriction of a set of clauses is defined by restricting each one. The restriction of a resolution refutation of a CNF is a refutation of its restriction.

4. EXPANDING MATRICES

As alluded to in the introduction, our tautologies will be based on systems of $\mathbb{F}_p$ linear equations, for increasingy large values of p. We plan to simulate these in resolution, which has boolean variables, by associating a collection of bits to each $\mathbb{F}_p$ variable. One natural strategy to refute such systems in resolution is to simulate an algebraic argument; if such a linear system is unsatisfiable, it must be possible to add multiples of the equations together in some order to derive a contradictory equation. Such a derivation can be simulated in resolution by having a collection of clauses for each equation, whose conjunction is semantically equivalent to the equation. If we only add two equations together at a time, the simulation may follow this by resolving some combinations of their underlying clauses to produce a set of clauses semantically equivalent to the resulting equation. This is easy to see, particularly by appealing to the completeness of resolution.

A little thought shows that the cost of this simulation will be essentially dominated by the size of the support of the largest intermediate equation which we hold in memory; for an equation with k variables, we will need a number of underlying clauses which is exponential in k. Thus for a system of equations to be hard for resolution, it should be the case that any linear algebra refutation must contain an intermediate equation of very large support. The previous work we mentioned earlier, which has focused on the $\mathbb{F}_2$ case, has produced techniques to prove that this property must hold for *expanding* linear systems, that is, linear systems whose underlying matrix is an expander graph, and this is used crucially to obtain lower bounds.

In this section and the next, we will show how to construct linear systems over $\mathbb{F}_p$, which are contradictory, yet any linear algebra refutation must at some point contain an equation which contains almost all of the variables. In the next section we will also show how to translate these into tautologies that are difficult for resolution.

We give the following variation on the concept of an expanding matrix.

DEFINITION 4.1. *Say that a matrix A over a field $\mathbb{F}_p$ is an $\mathbb{F}_p$-expander with parameters (r_1, r_2, c) if, for every column vector v of support size $r_1 \leq |v| \leq r_2$, the support of Av satisfies $c \leq |Av|$.*

When A is the incidence matrix of a graph, thought of over $\mathbb{F}_2$, and v is the characteristic vector of a set of vertices, it is easy to see that Av indicates the boundary edges, so $\mathbb{F}_p$ expansion generalizes the familiar notion from graph theory. On the other hand, whereas for graphs we can only reasonably hope that balanced cuts will contain say half of the edges, from $\mathbb{F}_p$ expanders we can hope for more.

In particular for a random sparse $n \times n$ $\mathbb{F}_p$-matrix, and a small constant δ, we could reasonably expect to obtain

a $(\delta n, 3\delta n, (1-2/p)n)$-expander. As a rough heuristic in support of this, the function computed by this matrix will look like a random map, at least on input vectors of relatively large hamming weight. At the same time, there are relatively few vectors in $\mathbb{F}_p^n$ of hamming weight between δn and $3\delta n$. Instead, the typical vectors have weight $(1-1/p)n$. Thus the chance that every vector of weight between $\delta n, 3\delta n$ maps to such a vector is high.

For our purposes it is not important to get a deterministic construction of $\mathbb{F}_p$ expanders, so we prove only existence by a probabilistic argument.

LEMMA 4.2. *Let d be a sufficiently large integer and p a sufficiently large prime, at most $O(\sqrt{d}/\log d)$. For any large enough n, there is an $n+1 \times n$ matrix over $\mathbb{F}_p$ such that*

- *Each row is supported on exactly d entries.*
- *The matrix is an (r_1, r_2, c)-$\mathbb{F}_p$-expander, where*
  - $r_1 = n/\sqrt{d}$,
  - $r_2 = 3n/\sqrt{d}$,
  - $c = (1 - O(1/p))(1 - O(1/\sqrt{d}))n)$.
- *No nontrivial linear combination of fewer than $3n/\sqrt{d}$ rows is the zero vector.*

PROOF. The proof is in two steps. First, we see that the support of the random matrix is a classical expander, as is well known; we will actually need this to hold for two different ranges of the parameters. Then we show that conditioned on the support being expanding, over the remaining randomness we almost surely have an $\mathbb{F}_p$ expander as needed.

Let B denote a random $\{0,1\}$-valued matrix of dimensions $n+1 \times n$, in which rows are independently chosen from the uniform distribution on vectors of support d, and let A denote the random $\mathbb{F}_p$ matrix in which nonzero $\mathbb{F}_p$ values are substituted for the ones of B independently.

First we show that with high probability over B, any set S of exactly $n/\sqrt{d}$ rows has ones in at least $(1-O(1/\sqrt{d}))n$ columns. For any column, the chance that it is missed is at most $(1-d/n)^{|S|} \leq \exp(-d|S|/n)$. The chance that any set of δn columns are missed is therefore at most $\exp(-d\delta|S| + H(\delta)n)$, where H is the binary entropy function, so for $\delta = O(\frac{1}{\sqrt{d}})$, the chance that any set S does not expand so much is $\ll 2^{-n}$. We conclude that with high probability every set of at least $n/\sqrt{d}$ rows has ones in at least $(1-O(1/\sqrt{d}))n$ columns.

Assuming this holds for B, we now show that with high probability over A, A is $(n/\sqrt{d}, 3n/\sqrt{d}, (1-O(1/p))(1-O(1/\sqrt{d}))n)$-$\mathbb{F}_p$ expanding. Fix any column vector v of support size $n/\sqrt{d} \leq |v| \leq 3n/\sqrt{d}$. The indices of Av are distributed independently over the coins of A, some distributed uniformly over $\mathbb{F}_p$ and some distributed as the constant zero. By the assumption for B, the number which are distributed uniformly is at least $(1-O(1/\sqrt{d}))n$. We expect only a fraction $1/p$ of these to be zero, and by standard tail bounds on the binomial distribution, we conclude that for any constant c_0, there exists a constant c_1 so that the probability to get fewer than $(1-c_1/p)(1-O(1/\sqrt{d}))n$ nonzero entries is at most $\exp(-c_0(1-O(1/\sqrt{d}))n/p)$.

Now we simply take a union bound over the number of vectors v to consider. This number is at most

$$\binom{n}{3n/\sqrt{d}} \cdot (p-1)^{3n/\sqrt{d}},$$

the log of which is asymptotically at most $n/\sqrt{d} \cdot (\log d + \log p)$. Therefore so long as $1/p = \Omega(1/\sqrt{d} \cdot (\log d + \log p))$, we can take c_0 large enough so that the tail bound dominates the union bound and A is $\mathbb{F}_p$-expanding asymptotically almost surely. Taking $p = O(\sqrt{d}/\log d)$ suffices.

Finally, by a standard calculation it can be shown that sets of rows of size at most $\leq n/\sqrt{d}$ expand by a factor $\Omega(\sqrt{d})$ at least in B, and that in this case, the probability over A that any vector supported on such a small set does not have image equal to zero is very small, using arguments similar to the above. If vectors supported on fewer than $n/\sqrt{d}$ positions don't have image zero, and $\mathbb{F}_p$ expansion holds between that value and $3n/\sqrt{d}$, then no vector of support $\leq 3n/\sqrt{d}$ has image zero, so no subset of at most $3n/\sqrt{d}$ rows has the zero vector as a nontrivial linear combination. □

5. WIDTH BOUND

Now we will be defining unsatisfiable linear systems using $\mathbb{F}_p$ expanders.

To obtain width lower bounds, we use a technique based on the *semantic measure* of proof lines, which was standardized by Ben-Sasson and Wigderson.

DEFINITION 5.1. *The* semantic measure *of a proposition P with respect to a set of propositions $\mathcal{A} = \{A_1, \ldots, A_m\}$ is*

$$\mu_{\mathcal{A}}(P) := \min_{S \subseteq [m]: \bigwedge_{i \in S} A_i \models P} |S| \ ,$$

that is, the minimal number of propositions from $\mathcal{A}$ which semantically implies P.

OBSERVATION 5.2 ([4]). *Let ϕ denote a minimally unsatisfiable CNF consisting of m clauses.*

- *For any clause C of ϕ, $\mu_\phi(C) = 1$.*
- *For any inference $C_1, C_2 \vdash C_3$, $\mu(C_3) \leq \mu(C_1) + \mu(C_2)$.*
- *$\mu(\bot) = m$.*

As an immediate corollary of these observations, in any refutation, in any proof system, of any minimally unsatisfiable set of m constraints, there must exist a proof line C such that $\mu(C) \in [m/3, 2m/3]$.

It is easy to see that for any matrix A as in the lemma, we can choose a vector $\vec{b}$ such that $A\vec{y} = \vec{b}$ is not satisfiable. Such a system will be *close* to minimally unsatisfiable, since if any subset S of the equations is contradictory, there must be a linear combination of them which produces $0 = 1$, and we argued before that this does not happen when $|S| \leq 3n/\sqrt{d}$.

CLAIM 5.3. *Let p be a large enough prime. There are unsatisfiable linear systems $A\vec{y} = \vec{b}$ consisting of $n+1$ $\mathbb{F}_p$-equations on n variables, each containing at most p^2 variables, in which A is an $(\gamma n, 3\gamma n, (1-c_2\gamma)n)$-$\mathbb{F}_p$ expander, and no subset of $\leq 3\gamma n$ equations is contradictory, where c_2 is a constant and $\gamma := \tilde{O}(1/p)$. (The $\tilde{O}$ notation hides* log *factors.)*

To express an $\mathbb{F}_p$ expanding system as a CNF, we encode variables as follows. For each $\mathbb{F}_p$ variable y_i we will have $\gamma^{-1}p$ boolean variables x_{ij}, thought of as taking values $\{0,1\}$, with the intended meaning that $y_i = \sum_j x_{ij} \mod p$. Thus, y_i does not determine the x_{ij}, and if a partial assignment π to the x_{ij} assigns only $\gamma^{-1}p - p$ variables, y_i is still completely unconstrained.

DEFINITION 5.4. *For $A\vec{y} = \vec{b}$ an $\mathbb{F}_p$ linear system over variables $y_1, \ldots, y_n$, let CNF ϕ denote the following conjunction in variables $x_{ij}, 1 \leq i \leq n, 1 \leq j \leq \gamma^{-1}p$:*

$$\bigwedge_{k=1}^{m} \left\{ \sum_i A_{k,i} \sum_j x_{ij} = b_k \ (mod\ p) \right\} .$$

Naturally we replace each equation above with its trivial CNF representation, which has only $\ell p \gamma^{-1}$ variables, if each equation in the linear system has only ℓ variables.

THEOREM 5.5. *Let ϕ be a CNF corresponding to a linear system as in Claim 5.3 via Definition 5.4, which is on $N = n \cdot \tilde{O}(p^2)$ variables. Then any resolution refutation of ϕ requires width $(1 - \tilde{O}(1/p))N$, and ϕ is an $\tilde{O}(p^4)$-CNF.*

PROOF. Let C be any clause containing fewer than $(1 - (c_2 + 1)\gamma)$ of the boolean variables. We show that if C has semantic complexity between $3\gamma n/2$ and $3\gamma n$, we contradict $\mathbb{F}_p$-expansion. By Observation 5.2 and the remarks immediately following it, this completes the proof.

By a Markov argument, there are at least a fraction $c_2\gamma$ of the y_i variables such that at least a fraction γ of their $x_{i,j}$ are unassigned. Let ρ denote the restriction corresponding to $\neg C$.

Say that a y_i variable is free if at least p of its $x_{i,j}$ variables are unassigned by ρ, and let ρ^* denote any extension of ρ whose domain is the domain of ρ, plus all $x_{i,j}$ variables corresponding to non-free y_i's.

Then in terms of the y_i variables, ρ^* corresponds semantically to a restriction which assigns all non-free variables, and leaves the free variables unset.

Thus, a subset $\{A_i\}_{i \in S}$ of the equations semantically implies C if and only if for every such ρ^*, $\{A_i|_{\rho^*}\}_{i \in S}$ implies a contradiction, and it minimally implies C if and only if every A_i is needed for some ρ^*.

An $\mathbb{F}$-linear system is unsatisfiable if and only if there is an $\mathbb{F}$-linear combination in the equations which gives $1 = 0$. If for some equation $E|_{\rho^*} = (1 = 0)$, then E only contains the variables assigned by ρ^*. That is, for each ρ^*, there exists a linear combination of the axioms $\{A_i\}_{i \in S}$ supported only on the non-free variables of ρ, and for every $i \in S$, by minimality, some ρ^*'s combination has a nonzero coefficient for A_i.

Therefore, take a random linear combination of the equations corresponding to each ρ^*. Then the resulting equation is supported again on only the non-free variables of ρ. We show that this equation is a nontrivial linear combination of many of the A_i, contradicting $\mathbb{F}_p$-expansion. Each A_i occurs in some equation, thus, in the resulting random linear combination, its coefficient is distributed uniformly over $\mathbb{F}_p$. By averaging, there exists such a combination which results in at least $(1 - 1/p)$ of the A_i having nonzero coefficients. Thus there is a linear combination supported on between $(1 - 1/p)(3/2)\gamma n \geq \gamma n$ and $3\gamma n$ equations of A with fewer than $(1 - c_2\gamma)$ of the y_i variables, contradicting $\mathbb{F}_p$ expansion as desired.

□

Since width is at most the base two log of Tree-like Size [4], this immediately implies $2^{(1-o(1))n}$ lower bounds for tree-like resolution, as mentioned previously.

6. REGULAR RESOLUTION

To obtain size lower bounds for Regular Resolution, we adapt and simplify the probabilistic adversary technique introduced in [1]. At a high level, this is a variation on the bottleneck counting argument introduced by Haken [12]. In this argument, a rule is given which maps assignments to particular clauses in the proof, at which significant "work" is done thinking about this assignment. The task is to show that we can map a large number of assignments in such a way that only a small number map to any particular clause, which implies that there are many clauses. In Haken's work this map is described explicitly – in the more modern form of the argument, due to Beame and Pitassi, a random restriction argument is used to hide these details. The bottleneck counting argument is of fundamental importance in computational complexity theory; besides underlying much of modern proof complexity, the bottleneck counting approach was also employed by Cook and Haken for monotone circuit lower bounds [13], and a report of Simon and Tsai [28] illustrated how closely related it is with the method of approximations used in other contexts. The high level plan is executed in part using ideas from [20].

THEOREM 6.1. *Any regular resolution refutation of the CNF ϕ of Theorem 5.5 has size at least $2^{(1-\tilde{O}(1/p))N}$ where N is the total number of variables.*

PROOF. We define a probabilistic process, which one may think of as an adversary in the sense of [20], which interacts with the proof, and which we think of as taking place at a particular clause in the proof at every step. The process begins at the final clause, $\perp$, and in each step moves to one of the two parents of the current clause, until at some point it stops, at some clause somewhere in the middle of the proof. The path which is followed by the process depends on what the current and parent clauses look like, what the history of the process is, and some random coins. We will show that over the random coins of the process, the probability that it stops at any particular clause of the proof is extremely small – from this we will deduce that there are many clauses.

We will think of the process as building up a truth assignment by assigning one variable at a time – in the step corresponding to a clause C, if C is deduced by resolving on variable x, the process will either assign $x = 0$ and move to the clause containing the literal x, or assign $x = 1$ and move to the caluse containing the literal $\overline{x}$. Thus if π denotes the partial assignment corresponding to all previous assignments made by the process, it always maintains the invariant that π falsifies the current clause. Crucially, by regularity, the variable x is always unassigned by π.

The rule by which we will assign variable x at each step is as follows:

- If variable x corresponds to a *free* $\mathbb{F}_p$ variable of ϕ (at least $p+1$ of its boolean variables are unset by π), then x is assigned randomly
- Otherwise, we choose x so as to maximize the semantic complexity of the clause for the next round.

The crucial claim regarding this process is that in each step, the semantic complexity of the occupied clause cannot decrease by more than a factor of two. Let us prove this. In the second case above, this is easy to see, because it is a standard application of subadditivity of the semantic measure. In the first case, we claim that the semantic measure cannot decrease at all – this is because if x corresponds to a free variable, then the two parent clauses each semantically entail one of $C \vee x, C \vee \overline{x}$, but these clauses are semantically implied by a set S of equations if and only if the corresponding restrictions of S are unsatisfiable, and since x is a free variable, the corresponding restriction in terms of the $\mathbb{F}_p$ variables is the same for $C, C \vee x$, and $C \vee \overline{x}$. Thus all three of these clauses are of the same semantic complexity, and the two parent clauses are each of at least this complexity.

Since at the beginning, the contradiction clause $\perp$ at which the process begins has semantic complexity $\geq 3\gamma n$, at the end of any path, any axiom has semantic complexity 1, and in any step the measure at most halves, at some point in the process we must walk to a clause of semantic complexity between $2\gamma n$ and γn. The first time that this happens, the process is defined to stop at this clause.

What is the probability that the process stops at any particular clause C? Since π must falsify C by the time we walk to C, the process can only walk to C if it assigns all the variables which appear in C consistently with $\neg C$. By $\mathbb{F}_p$ expansion and the width argument from before, C can only have semantic complexity between γn and $2\gamma n$ if at least $(1 - c_2\gamma)n$ of the $\mathbb{F}_p$ variables are non-free under the restriction $\neg C$ – thus C assigns at least $(1 - \gamma)$ of the bits corresponding to these variables to particular values. All of these bits must be queried along any path which reaches C, and for each $\mathbb{F}_p$ variable, for the first $(1-\gamma)$ fraction of these bits, the process will respond randomly. In the event that it assigns some bit incorrectly, it can never reach C, and for each such queried bit, this happens with probability $1/2$ at least.

DEFINITION 6.2. *For any path from the root σ, and any clause C, let freedoms(σ, i, C) denote the number*

$$\max\left(|\mathit{Vars}(C) \setminus \mathit{dom}\ \sigma| - p, 0\right) .$$

This is a lower bound on the number of bits of C which would be responded to randomly if for example the next variables encountered by the adversary are all elements of $\mathit{Vars}(C) \setminus \mathit{dom}\ \sigma$. Let freedoms$(\sigma, C)$ denote

$$\sum_i \mathit{freedoms}(\sigma, i, C) .$$

CLAIM 6.3. *The probability that the adversary, having followed σ, reaches C is at most*

$$2^{-\mathit{freedoms}(\sigma, C)} .$$

PROOF. By induction on σ. The base case is that σ is large and has zero freedoms with respect to C, in which case the probability bound is trivial. Let σ be any path in the proof, leading to a clause which is deduced by resolving on variable x. In case that assigning x does not reduce the number of freedoms, by inductive hypothesis applied to the paths $\sigma \cup \{x = 0\}$ and $\sigma \cup \{x = 1\}$ the probability bound we want holds at both of these, and by averaging it holds for σ. In case that assigning x does reduce the number of freedoms, the bound obtained inductively is only a factor two worse than what we claim at σ. Since assigning x does reduce the number of freedoms, its $\mathbb{F}_p$ variable is still free so x is assigned randomly by the adversary. If the adversary assigns x to satisfy C, then he can never reach C, so for one of $\sigma \cup \{x = 0\}, \sigma \cup \{x = 1\}$ the probability to reach C is zero. We conclude that the probability that the adversary reaches C from σ is two to the minus the total number of freedoms as claimed. □

Now apply the claim in case that σ is the empty path at $\perp$, and with C any possible stopping point. Since the total number of freedoms initially is at least $(1 - c_2\gamma)(1 - \gamma) = (1 - O(\gamma))$, there is at most a $2^{-(1-O(\gamma))np\gamma^{-1}}$ probability that the adversary stops at any particular clause C. Since the process always stops at some clause, this implies that there are at least $2^{(1-O(\gamma))np\gamma^{-1}}$ clauses in any refutation of the tautology ϕ, which is on $np\gamma^{-1}$ variables.

7. GENERAL RESOLUTION

In this section, we give size lower bounds in General Resolution, using the analysis of the width lower bound in the first section, together with a quantitatively improved random restriction argument.

We can abstract the argument as follows.

DEFINITION 7.1. *For $\mathcal{C}$ a set of constraints in n boolean variables $x_1, \ldots, x_n$, and $f : \{0,1\}^m \to \{0,1\}^n$ a boolean function $f : \vec{y} \mapsto \vec{x}$, we define the f-substituted set of constraints $\mathcal{C}[f]$ in variables $y_1, \ldots, y_m$ as the set $\{C(f(\vec{y})) : C \in \mathcal{C}\}$.*

When f is n parallel copies of the parity function on ℓ bits, we denote this more succinctly by $\mathcal{C}[\oplus^\ell]$.

The idea of $\oplus$-substitution has been used in many contexts, as a tool to make formulas more difficult for restricted models. Intuitively, it can make a formula harder because even if $\mathcal{C}$ has an efficient refutation, a refutation of $\mathcal{C}[\oplus^\ell]$ might not be able to simulate that refutation efficiently, since it can't reason directly about the bits of $\mathcal{C}$. Since parity is hard to express in CNF, one would expect that $\oplus$-substition would make this simulation particularly cumbersome for resolution. For this and other reasons, indirection based on parity is a common strategy to make hard formulas.

Our argument will make use of the following notion of projection, which takes a clause in substituted variables and extracts a new clause which captures its information about the values of the original variables. (This definition makes the most sense when the substitution function gives disjoint sets of variables to each original variable.)

DEFINITION 7.2. *Let C be a clause in the variables $y_1, \ldots, y_m$ of $\mathcal{C}[f]$. Let $C = \bigvee C_i$ where C_i is the part of C_i in the variables corresponding to x_i. Define the* projection *C'_i of C_i by*

$$C'_i := \begin{cases} x_i & C_i \models (f(\vec{y}))_i = 1 \\ \overline{x_i} & C_i \models (f(\vec{y}))_i = 0 \\ 1 & \mathit{otherwise} \end{cases} .$$

Define the projection C' of C by $\bigvee C'_i$.

We observe that the semantic complexity of a clause is unchanged by projection when the substitution function is made up of boolean functions on disjoint inputs.

OBSERVATION 7.3. *The semantic complexity of C with respect to $\mathcal{C}[\oplus^\ell]$ is the same as the semantic complexity of C' with respect to $\mathcal{C}$.*

PROOF. It is easy to see that for any assignment to $y_1, \ldots, y_m$ satisfying C, its image under f satisfies C'. By the definition of projection, it is also true that any assignment to $x_1, \ldots, x_n$ which satisfies $\mathcal{C}'$ may be lifted to an assignment satisfying C. □

Our next lemma uses this to bound the probability that the projection of a restricted clause is wide.

LEMMA 7.4. *Fix any $\ell \geq 2$, and CNF ϕ in variables $x_1, \ldots, x_n$. Let ρ denote an iid random restriction which sets the variables of $\phi[\oplus^\ell]$ to $0, 1, \star$ with equal probability, conditioned on never setting all y variables associated to any x_i to a constant. For any clause C in the y variables, the probability that the projection $C|'_\rho$ has width at least $(1-\epsilon)n$ is at most $\left(\frac{2}{3}\right)^{(1-\epsilon)n\ell - O((2/3)^\ell)n - O(H(\epsilon))n}$, where H is the binary entropy function.*

PROOF. Fix an arbitrary clause C and consider the width of the projection of the restricted clause, $(C|_\rho)'$. We first show that this is small with high probability. We can analyze this by considering each variable one at a time. Let $(C|_\rho)'_i$ denote the portion of $C|_\rho$ associated to x_i, as in the definition of projection.

$$(C|_\rho)' = \bigvee_i (C|_\rho)'_i \ .$$

Traditionally, restrictions are only thought to kill clauses by setting variables of the clause to true. Thanks to the definition of projection, we can also think of killing variables of a clause by setting its *non-variables* to $\star$, since in this case the projection of that portion of the restricted clause will be trivial.

CLAIM 7.5. *For any C,*

$$\Pr_\rho[(C|_\rho)'_i \neq 1] \leq (2/3)^\ell \cdot (1 - (2/3)^\ell)^{-1} \ .$$

PROOF. For each variable in C_i, if ρ sets it oppossite to its value in C_i, then $C_i|_\rho = 1$, and $(C|_\rho)'_i = 1$. For each variable y_j associated to x_i but not appearing in C_i, if it set to $\star$, then any satisfying assignment to $C_i|_\rho$ may be flipped on this variable, still satisfying $C_i|_\rho$ but having opposite parity, thus $C_i|_\rho \not\models \oplus y_j = 1, C_i|_\rho \not\models \oplus y_j = 0$, hence $(C|_\rho)'_i = 1$. Thus if we ignore the conditioning, the probability that $(C|_\rho)'_i \neq 1$ is at most $(2/3)^\ell$. The event which we condition away has probability at most $(2/3)^\ell$, hence doing so can only increase probabilities by a fraction $(1 - (2/3)^\ell)$. □

The width of $C|'_\rho$ is the sum of the widths of $(C|_\rho)'_i$, and ρ acts independently on each C_i, so by a union bound over all subsets of $(1-\epsilon)n$ of the variables $x_1, \ldots, x_n$,

$$\Pr_\rho[\mathrm{Vars}(C|'_\rho) \geq (1-\epsilon)n] \leq$$

$$\binom{n}{(1-\epsilon)n} \left((2/3)^\ell (1 - (2/3)^\ell)^{-1}\right)^{(1-\epsilon)n} \ .$$

Using the identity $-\ln(1-x) \leq x + x^2$ for $x < 1/2$, this is upper bounded again as

$$\leq (2/3)^{(1-\epsilon)n\ell - O((2/3)^\ell)n - O(H(\epsilon))n} \ ,$$

where H is the binary entropy function. This completes the proof of Lemma 7.4.

This can be used to obtain size lower bounds. If ϕ is such that any clause of intermediate semantic complexity is wide, then the above shows that it is very unlikely that a restriction of a clause in a proof of $\phi[\oplus^\ell]$ has intermediate semantic complexity.

COROLLARY 7.6. *For any small enough $\epsilon > 0$, there exist $\tilde{O}\left(\frac{1}{\epsilon^4}\right)$-CNF formulas on n variables which have resolution complexity at least $\left(\frac{3}{2}\right)^{(1-\epsilon)n}$.*

PROOF. First we choose ℓ to simplify the bound above. Take $\ell = O(\log \frac{1}{\epsilon})$, and using the fact that $\lim_{\epsilon \to 0} \frac{H(\epsilon)}{\epsilon \log \frac{1}{\epsilon}} = O(1)$, observe that with this choice of ℓ, for small enough ϵ, $\epsilon + \frac{1}{\ell}\left((2/3)^\ell + O(H(\epsilon))\right) = O(\epsilon)$, so the above lemma implies a probability bound of $\left(\frac{2}{3}\right)^{(1-O(\epsilon))n}$.

Let $k = \tilde{O}\left(1/\epsilon^4\right)$, and apply the lemma above when ϕ is any k-CNF such that clauses of intermediate semantic complexity have width at least $(1 - \tilde{O}(k^{-1/4}))n = (1-\epsilon)n$, as we obtained from Theorem 5.5. Then $\phi[\oplus^\ell]$ is a $k\ell$-CNF, which we will show has resolution complexity $\left(\frac{3}{2}\right)^{(1-\epsilon)n}$ Consider any hypothetical resolution refutation of size less than this, and apply random restriction ρ. By a union bound, for some such ρ every clause of the restricted proof has a projection of width less than $(1-\epsilon)n$, which implies none of the projections have intermediate semantic complexity.

Since ρ is conditioned never to set all variables to constants, it is always true that $\phi[\oplus^\ell]|_\rho$ is a substitution of ϕ. By (the proof of) Observation 7.3, this implies the restricted proof of $\phi[\oplus^\ell]|_\rho$ thus obtained has no clause of semantic complexity intermediate between γn and $2\gamma n$, contradicting subadditivity of the semantic measure. Since $k\ell = \tilde{O}\left(\frac{1}{\epsilon^4}\right)$ this shows $\phi[\oplus^\ell]$ satisfies the claim.

□

8. CONCLUDING REMARKS

We have demonstrated that there exist tautologies on n variables which require resolution width $(1-\epsilon)n$ and regular resolution proofs of size $2^{(1-\epsilon)n}$, for any $\epsilon > 0$. Moreover, these tautologies may be taken to be k-CNF's for $k = \tilde{O}(\frac{1}{\epsilon})^4$. In general resolution we obtain lower bounds of $(3/2)^{(1-\epsilon)n}$.

A good question is how closely this can be made to match the performance of k-SAT algorithms like PPSZ. Can we get results for general resolution matching those we obtained in regular resolution? Can we find k-CNF's which require resolution width $(1 - O(1/k))n$?

9. REFERENCES

[1] P. Beame, C. Beck, and R. Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pages 213–232, May 2012.

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

[3] E. Ben-Sasson and R. Impagliazzo. Random CNF's are hard for the polynomial calculus. In *Proceedings 40th Annual Symposium on Foundations of Computer Science*, pages 415–421, New York,NY, Oct. 1999. IEEE.

[4] E. Ben-Sasson and A. Wigderson. Short proofs are narrow – resolution made simple. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 517–526, Atlanta, GA, May 1999.

[5] S. Buss, D. Grigoriev, R. Impagliazzo, and T. Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 547–556, Atlanta, GA, May 1999.

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

[7] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, July 1962.

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

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

[10] D. Grigoriev. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259:613–622, 2001.

[11] L. K. Grover. A fast quantum mechanical algorithm for database search. In G. L. Miller, editor, *STOC*, pages 212–219. ACM, 1996.

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

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

[14] T. Hertli. 3-sat faster and simpler - unique-sat bounds for ppsz hold in general. In R. Ostrovsky, editor, *FOCS*, pages 277–284. IEEE, 2011.

[15] R. Impagliazzo, W. Matthews, and R. Paturi. A satisfiability algorithm for ac^0. In Y. Rabani, editor, *SODA*, pages 961–972. SIAM, 2012.

[16] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 67:367–375, 2001.

[17] R. Impagliazzo, P. Pudlák, and J. Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

[18] R. Paturi, P. Pudlák, M. E. Saks, and F. Zane. An improved exponential-time algorithm for k-sat. *J. ACM*, 52(3):337–364, 2005.

[19] R. Paturi, P. Pudlák, and F. Zane. Satisfiability coding lemma. In *FOCS*, pages 566–574. IEEE Computer Society, 1997.

[20] P. Pudlák and R. Impagliazzo. A lower bound for dll algorithms for k-sat (preliminary version). In D. B. Shmoys, editor, *SODA*, pages 128–136. ACM/SIAM, 2000.

[21] R. Raz. Resolution lower bounds for the weak pigeonhole principle. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 553–562, Montreal, Quebec, Canada, May 2002.

[22] A. A. Razborov. Improved resolution lower bounds for the weak pigeonhole principle. Technical Report TR01-055, Electronic Colloquium in Computation Complexity, `http://www.eccc.uni-trier.de/eccc/`, 2001.

[23] A. A. Razborov. Resolution lower bounds for the weak functional pigeonhole principle. Technical Report TR01-075, Electronic Colloquium in Computation Complexity, `http://www.eccc.uni-trier.de/eccc/`, 2001.

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

[25] The international SAT Competitions. `http://www.satcompetition.org`.

[26] G. Schoenebeck. Linear level lasserre lower bounds for certain k-csps. In *FOCS*, pages 593–602. IEEE Computer Society, 2008.

[27] U. Schöning. A probabilistic algorithm for k-SAT and constraint satisfaction problems. In *Proceedings 40th Annual Symposium on Foundations of Computer Science*, pages 410–414, New York,NY, Oct. 1999. IEEE.

[28] J. Simon and S.-C. Tsai. On the bottleneck counting argument. *Theor. Comput. Sci.*, 237(1-2):429–437, 2000.

[29] G. Tseitin. On the complexity of derivation in propositional calculus. In J. Siekmann and G. Wrightson, editors, *Automation of Reasoning*, pages 466–483. Springer, Berlin, 1983.

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

[31] R. Williams. Non-uniform acc circuit lower bounds. In *IEEE Conference on Computational Complexity*, pages 115–125. IEEE Computer Society, 2011.