THE BULLETIN OF SYMBOLIC LOGIC
Volume 13, Number 4, Dec. 2007

THE COMPLEXITY OF PROPOSITIONAL PROOFS

NATHAN SEGERLIND

Abstract. Propositional proof complexity is the study of the sizes of propositional proofs, and more generally, the resources necessary to certify propositional tautologies. Questions about proof sizes have connections with computational complexity, theories of arithmetic, and satisfiability algorithms. This is article includes a broad survey of the field, and a technical exposition of some recently developed techniques for proving lower bounds on proof sizes.

CONTENTS

Received June 27, 2007.

This survey article is based upon the author's Sacks Prize winning PhD dissertation, however, the emphasis here is on context. The vast majority of results discussed are not the author's, and several results from the author's dissertation are omitted.

1079-8986/07/1304-0001/$7.50

Part 1. A tour of propositional proof complexity.

§1. Is there a way to prove every tautology with a short proof? One way to certify that a propositional formula is a tautology is to present a proof of the formula in a propositional calculus, such as the system $\mathcal{F}$ below:

DEFINITION 1.1. *The formulas of $\mathcal{F}$ are the well-formed formulas over the connectives $\land$, $\lor$, $\to$ and $\neg$. The inference rule of $\mathcal{F}$ is modus ponens (from A and $A \to B$ infer B), and its axioms are all substitution instances of*:

1. $A \to (B \to A)$	2. $A \land B \to B$
3. $(A \to B) \to (A \to B \to C) \to (A \to C)$	4. $A \land B \to A$
5. $A \to A \lor B$	6. $A \to B \to A \land B$
7. $(A \to B) \to (A \to \neg B) \to \neg A$	8. $B \to A \lor B$
9. $(A \to C) \to (B \to C) \to (A \lor B \to C)$	10. $\neg\neg A \to A$

*Let τ be a propositional formula. An $\mathcal{F}$-*proof of τ *is a sequence of formulas $F_1, \dots, F_m$ so that $F_m = \tau$, and each F_i is either an axiom, or follows from the application of modus ponens to two formulas F_j and F_k, with $j, k < i$.*

The completeness theorem for $\mathcal{F}$ guarantees that every tautology has an $\mathcal{F}$-proof. Moreover, most proofs of the completeness theorem give quantitative bounds on proof sizes: Every tautology τ on n variables has an $\mathcal{F}$-proof in which there are at most $2^{O(n)}$ formulas, each of which has size polynomial in the size of τ. Of course, for many tautologies, much smaller proofs are possible. Does *every* tautology have an $\mathcal{F}$-proof significantly smaller than the exponential length derivation? More generally, does there exist a propositional proof system in which every tautology has a small proof?

This question requires a clarification of what is meant by "propositional proof sytem". For example, any algorithm for deciding satisfiability of a Boolean formula can be viewed as a proof system, with an execution trace for a run that declares ψ to be unsatisfiable being viewed as a proof that $\neg\psi$ is a tautology. Another possibility would be to formalize the definitions of propositional formulas and tautologies in ZFC and present a proof in formal ZFC that the formula in question is a tautology. This might seem extreme but by using high-level mathematics, some proofs might be shorter than possible with a more commonplace system such as $\mathcal{F}$. These methods and the proof system $\mathcal{F}$ share three properties that seem necessary for any method of certifying tautologies: Every tautology has a proof, only tautologies have proofs, and valid proofs are computationally easy to verify.

DEFINITION 1.2 (modified from [69]). *Let F denote the set of propositional formulas over the connectives $\land$, $\lor$, $\to$ and $\neg$, with a countably infinite supply of propositional variables. An* abstract propositional proof system *is a polynomial time function $V : F \times \{0,1\}^* \to \{0,1\}$ such that for every tautology τ*

there is a proof $P \in \{0,1\}^*$ *with* $V(\tau, P) = 1$ *and for every non-tautology* τ, *for every* P, $V(\tau, P) = 0$. *The size of the proof is* $|P|$.

Defintion 1.2 equates propositional proof systems with non-deterministic algorithms for the language of tautologies. In particular, if a family of tautologies possess polynomial-size proofs in the sense of Definition 1.2, then that family of tautologies is in *NP*.[1]

Definition 1.3. *A propositional proof systems is said to be* polynomially bounded *if there exists a constant c so that for every tautology* τ, *there exists a proof P with* $|P| \leq c|\tau|^c$ *and* $V(\tau, P) = 1$.

Theorem 1.1. [69] *There exists a polynomially bounded propositional proof system if and only* $NP = coNP$.

Proof. Let *TAUT* denote the language of propositional tautologies over the connectives $\wedge$, $\vee$, $\neg$, and $\rightarrow$, and let $TAUT^c$ denote the language of non-tautologies. If $NP = coNP$ then there is a polynomial-time nondeterministic Turing machine that decides *TAUT*, call this machine A. The procedure that takes a tautology τ and a string S and checks that S is an an accepting computational history of A on input τ is a polynomially bounded proof system for *TAUT*. Now suppose that there is a polynomially-bounded propositional proof system V. Choose a constant c so that every tautology τ has a V-proof of length at most $c|\tau|^c$. The nondeterministic algorithm that on input τ simply guesses a string S of length $\leq c|\tau|^c$ and verifies that S is a V-proof of τ correctly decides *TAUT*. Because *TAUT* is *coNP*-complete under polynomial-time many-one reductions, we have that $coNP \subseteq NP$. Furthermore, this places $TAUT^c \in coNP$, and since $TAUT^c$ is *NP*-complete we have $NP \subseteq coNP$ and thus $NP = coNP$. $\dashv$

Because $P = NP \Rightarrow NP = coNP$, showing that there is no polynomially-bounded propositional proof system would also show that $P \neq NP$. So resolving the existence of a polynomially-bounded propositional proof system "in the expected direction" is probably a tough problem.

Showing that $\mathcal{F}$ is not polynomially bounded seems to be an easier problem than showing $NP \neq coNP$—it is a particular proof system with a simple syntactic structure. However, whether or not $\mathcal{F}$ is polynomially bounded has resisted decades of effort, and this problem can be viewed as the fundamental open problem in propositional proof complexity- "Are the Frege systems polynomially bounded?" *Frege systems* are the axiom-and-inference-rule based derivation systems exemplified by the system $\mathcal{F}$.

[1]It is natural to ask what happens if the proof verification procedure is a randomized or quantum algorithm. With a randomized classical verifier, families of tautologies with polynomial-size proofs fall into the complexity class of "Merlin–Arthur games" (MA), which, modulo plausible conjectures in computational complexity, is the same class as *NP* [97, 84]. For a quantum verifier, families of tautologies with polynomial-size proofs are in the class $QCMA$, and it is not known how this class relates to *NP* [4].

Definition 1.4. [69] *A* Frege system *is an axiomatic proof system that is implicationally complete. An axiomatic proof system has two parts*:

1. *A finite set of propositional tautologies, $A_1, \dots, A_k$, called the axioms.*
2. *A finite set of tuples of formulas $(A_0, \dots, A_l)$ such that for each tuple $\bigwedge_{i=1}^{l} A_i \rightarrow A_0$ is a tautology. These tuples are called* inference rules *and are not necessarily of the same arity.*

A derivation of a propositional formula τ from hypotheses $\mathcal{H}$ is a sequence of formulas $F_1, \dots, F_m$ so that $F_m = \tau$ and each F_i is either a member of $\mathcal{H}$, a substitution instance of an axiom, or, there is an inference rule of $(A_0, A_1, \dots, A_l)$ and a substitution σ so that $F_i = A_0[\sigma]$, for each $j = 1, \dots, l$, the formula $A_j[\sigma]$ is among the formulas $F_1, \dots, F_{i-1}$. A proof of τ is a derivation of τ from the empty set of hypotheses.

A propositional proof system $\mathcal{G}$ is said to be implicationally complete *if for all formulas $F_0, \dots, F_k$, whenever $F_1, \dots, F_k \models F_0$, there exists a $\mathcal{G}$-derivation of F_0 from the hypotheses $F_1, \dots, F_k$.*

The particular choice of axioms and inference rules does not affect proof sizes too much, as derivations in one Frege system can be efficiently translated into derivations in any other Frege system. Implicational completeness is used in the proof of this fact.

Theorem 1.2. [69] *There exists a polynomially-bounded Frege system if and only if all Frege systems are polynomially bounded.*

Establishing superpolynomial proof size lower bounds for the Frege systems seems beyond current techniques, so people have focused their attention on proving size lower bounds for Frege systems that use only formulas of some limited syntactic form. These results can be interpreted as partial results towards the larger goals of proving that the Frege systems are not polynomially-bounded and proving that $NP \neq coNP$. Furthermore, these special cases are interesting on their own terms: Proof size lower bounds for restricted Frege systems can establish run time lower bounds for satisfiability algorithms and independence results for first-order theories of arithmetic.

§2. Satisfiability algorithms and theories of arithmetic.

2.1. The efficiency of satisfiability algorithms. Many satisfiability algorithms heuristically construct proofs in a restricted fragment of a Frege system. By identifying tautologies that require large proofs in the proof system, we identify limitations for the satisfiability algorithms that apply no matter which heuristics are used. Knowledge of these limitations helps explain why some algorithms are faster than others on certain instances, and helps guide the development of new algorithms.

The best known connection between a proof system and satisfiability algorithms is that between resolution and satisfiability algorithms such as the Davis–Logemann–Loveland procedure, the Davis–Putnam procedure, and contemporary clause learning algorithms. This brings us a minor technical issue: Because satisfiability algorithms distinguish between satisfiable and unsatisfiable formulas (as opposed to tautological and non-tautological formulas), it is cleaner to compare satisfiability algorithms with *refutation systems*. A refutation of ϕ in a Frege system is a derivation of a contradiction from ϕ. Because the axioms are tautologies and the inference rules are sound, a refutation of ϕ certifies that ϕ is unsatisfiable. Every refutation system can be viewed as a proof system because ϕ is a tautology if and only if $\neg\phi$ is unsatisfiable

DEFINITION 2.1. Resolution *is a propositional refutation system that manipulates clauses, and has two inference rules: The* resolution rule, *"From $A \vee x$ and $B \vee \neg x$, infer $A \vee B$", and the* subsumption rule, *"From A, infer $A \vee x$". A resolution refutation of a CNF $\bigwedge_{i=1}^{m} C_i$ is a sequence of clauses $D_1, \dots, D_s$ so that $D_s = \emptyset$, and each D_i either is one of the clauses $C_1, \dots, C_m$, or follows from the preceding clauses D_j, $j < i$ by application of one of the inference rules.*

A basic satisfiability algorithm is the Davis–Logemann–Loveland (DLL) procedure [112]. Below we present pseudocode for a simple DLL-based satisfiability algorithm.[2] The input F is a CNF represented as a set of clauses and the input π is a partial assignment to the variables, represented as a set of literals. The procedure returns 0 if $F \restriction_\pi$ is unsatisfiable and 1 if $F \restriction_\pi$ is satisfiable. To decide if F is satisfiable, run $\mathrm{DLL}(F, \emptyset)$. A sample run of the DLL algorithm is presented in Figure 1.

$\mathrm{DLL}(F, \pi)$:

1. If for all $C \in F$, $C \restriction_\pi = 1$, return 1
2. If there exist a clause $C \in F$ so that $C \restriction_\pi = 0$, return 0
3. (Unit Propagation) If there exists a clause $C \in F$ so that $C \restriction_\pi = l$, then return $\mathrm{DLL}(F, \pi \cup \{l\})$
4. (Decision)
   (a) Heuristically choose a variable x that is unset by π
   (b) Heuristically choose a value $v \in \{0, 1\}$
   (c) Return $\mathrm{DLL}(F, \pi \cup \{x^v\}) \vee \mathrm{DLL}(F, \pi \cup \{x^{1-v}\})$

When an implementation of the DLL algorithm finds a CNF F to be unsatisfiable, its execution tree corresponds to a resolution refutation of F. The idea is to label each leaf by a clause of F falsified by the branch, and then proceed upwards resolving on each variable that is branched upon. Unit propagation on a variable is treated as a decision node in which one child is immediately falsified. The conversion of the DLL tree in Figure 1 into a resolution refutation is demonstrated in Figure 2. This conversion holds regardless of the heuristic choices used for branching at steps 4a and 4b.

[2]The original version of the procedure included a "Pure Literal Rule": If there exists a literal l that occurs only positively in F then we may set l to 1. Contemporary satisfiability engines usually omit this rule. The translation into resolution is easily seen to hold even when the pure literal rule is used.

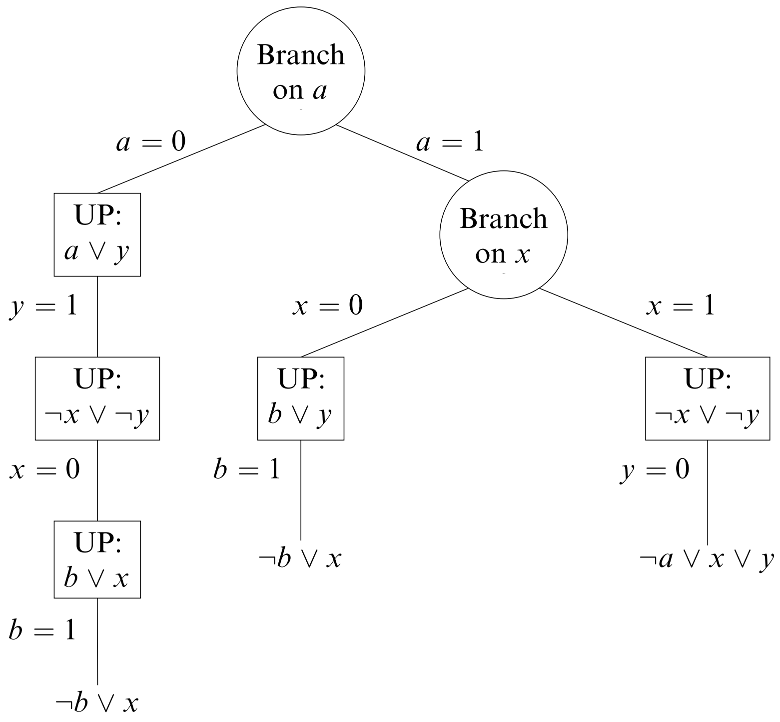

FIGURE 1. A DLL refutation of the set of clauses $\neg a \vee \neg x \vee \neg y$, $a \vee y$, $\neg b \vee x$, $b \vee x$, $\neg x \vee \neg y$. "UP" written above clause denotes a unit propagation caused by that clause. Beneath each branch is a clause which is falsified by the partial assignment of that branch.

LEMMA 2.1. *If some implementation of the DLL algorithm deems a CNF F to be unsatisfiable within s steps, then there is a resolution refutation of F of size at most s.*

The Davis–Putnam procedure is another satisfiability algorithm based upon resolution [71]. Below we present pseudocode for a simple DP-based satisfiability algorithm. Again, the input F is a CNF represented as a set of clauses. The procedure returns 0 if F is unsatisfiable and 1 if F is satisfiable.

$DP(F)$:
1. Order the variables as $x_1, \dots, x_n$.
2. For $i = 1, \dots, n$:
   (a) For each clause $C \vee x_i \in F$, and each clause $D \vee \neg x_i \in F$, add $C \vee D$ to F
   (b) Remove all clauses containing x_i from F
3. If the empty clause belongs to F then return 0, otherwise return 1

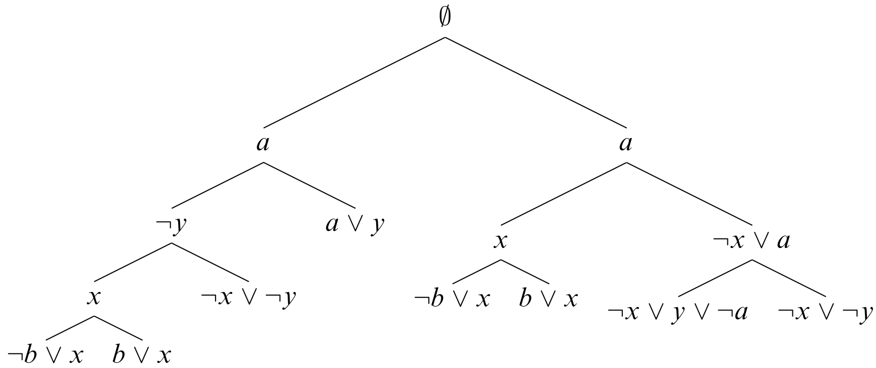

FIGURE 2. The resolution refutation of the set of clauses $\neg a \vee \neg x \vee \neg y$, $a \vee y$, $\neg b \vee x$, $b \vee x$, $\neg x \vee \neg y$ that corresponds to the DLL run of Figure 1.

The execution of the Davis–Putnam algorithm on an unsatisfiable CNF corresponds to a resolution refutation. This is demonstrated in Figure 3.

LEMMA 2.2. *If the Davis–Putnam algorithm deems a CNF F to be unsatisfiable within s steps, then there is a resolution refutation of F of size at most s.*

Notice that the conversion from the execution trace of a DLL algorithm into a resolution refutation preserves the structure of the backtracking tree. In the jargon of propositional proof complexity, the derivation of Figure 2 is said to be *tree-like* and the derivation of Figure 3 is said to be *DAG-like*. In Figure 2, the literal x is derived twice, whereas in Figure 3, it is derived once and used twice. The ability to reuse previously derived formulas, rather than repeatedly rederiving them, can make general resolution exponentially more efficient than tree-like resolution.

THEOREM 2.3 ([37] building upon [67, 159, 44, 38]). *There exists a family of unsatisfiable CNFs, $\{F_n\}_{n=1}^{\infty}$, with $|F_n| = O(n)$, so that tree-like resolution refutations of F_n are all of size $2^{\Omega(n/\log n)}$ but F_n possesses DAG-like resolution refutations of size $O(n)$.*

Theorem 2.3 shows that algorithms that generate DAG-like resolution refutations can be exponentially more efficient than any algorithm that generates tree-like resolution refutations—even those with idealized optimal branching heuristics. While the Davis–Putnam procedure creates DAG-like refutations, it is often unsatisfactory because it can derive many unnecessary clauses and has large memory requirements. However, in recent years there has been progress with other methods that generate DAG-like resolution proofs in a more efficient manner than the Davis–Putnam approach. Algorithms based on *DLL with clause learning* [157, 24, 111, 119, 83, 77] perform a DLL backtracking search augmented with the ability to create

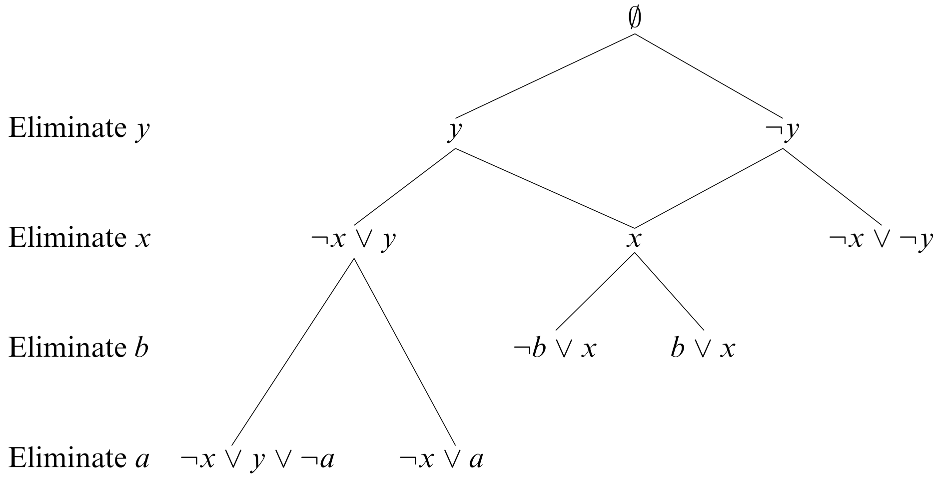

FIGURE 3. The resolution refutation of the set of clauses $\neg a \vee \neg x \vee \neg y$, $a \vee y$, $\neg b \vee x$, $b \vee x$, $\neg x \vee \neg y$ generated by the Davis–Putnam procedure with the variable order a, b, x, y.

new ("learned") clauses and remove these new clauses when unneeded. This process constructs DAG-like resolution refutations [109], and it is known that versions of these algorithms can efficiently refute the CNFs of Theorem 2.3 [30].

The satisfiability algorithms that we have discussed so far—DLL backtracking, the Davis–Putnam procedure, and DLL with clause learning, share some limitations. Each implements resolution, and therefore none can quickly refute a CNF that requires large resolution refutations. Consider the pigeonhole principle, the statement that $n + 1$ pigeons cannot be placed into n holes without a collision. This fact can be encoded as an unsatisfiable CNF as follows: For each $i = 1, \dots, n + 1$, there is a clause $\bigvee_{j=1}^{n} x_{i,j}$—"pigeon i gets some hole", and for all $1 \leq i < j \leq n + 1$, and all $1 \leq k \leq n$, $\neg x_{i,k} \vee \neg x_{j,k}$—"pigeon i and pigeon j do not share hole k". (This CNF is so important that we give it a name, PHP_n^{n+1}.) A famous result of Armin Haken shows that the pigeonhole principle requires exponentially large resolution refutations.

THEOREM 2.4. [87, 60, 29, 38] *Resolution refutations of* PHP_n^{n+1} *require size* $2^{\Omega(n)}$.

COROLLARY 2.5. *All DLL, Davis–Putnam or DLL with clause learning algorithms run for* $2^{\Omega(n)}$ *many steps when processing* PHP_n^{n+1}.

Many satisfiability algorithms have been proposed that can efficiently refute the propositional pigeonhole principle (and thereby go beyond the abilities of resolution based solvers). Techniques based on symmetry-exploitation [76, 75, 74, 16], integer programming [73], and ordered-binary decision diagrams [61, 62, 14, 120, 121, 125] have been suggested. Proof search for these systems is a developing art, and none of these algorithms has yet to consistently out-perform resolution based solvers over general instances.

2.2. Independence results for weak-theories of arithmetic. One notion of constructivity in arithmetic is to restrict the use of induction so that the definable functions have restricted growth rates. A well known example of such a system is Parikh's theory $I\Delta_0$, which formalizes strongly finitist arguments that disallow the use of exponentiation [126, 55].

DEFINITION 2.2. *The* bounded formulas *over the language* $+, \cdot, \leq, 0, 1$ *are those meeting the following recursive definition*:

1. *All quantifier free formulas are bounded.*
2. *If* $\phi(y)$ *is a bounded formula and* t *is a term, then* $\forall y < t\ \phi(y)$ *and* $\exists y < t\ \phi(y)$ *are bounded formulas.* (*Either* ϕ *or* t *or both might contain free variables different from* y.)

$I\Delta_0$ *is a first-order theory with function symbols* $+$ *and* $\cdot$, *binary relation symbol* $<$, *and constants* 0 *and* 1. *As axioms, the theory includes the universal closures of each of the following formulas*:

$a+0=a$	$(a+b)+c=a+(b+c)$	$a+b=b+a$
$a<b\rightarrow\exists x,\ a+x=b$	$0=a \vee 0<a$	$0<1$
$0<a\rightarrow 1\leq a$	$a<b\rightarrow a+c<b+c$	$a\cdot 0=0$
$a\cdot 1=a$	$(a\cdot b)\cdot c=a\cdot(b\cdot c)$	$a\cdot b=b\cdot a$
$(a<b\wedge c\neq 0)\rightarrow a\cdot c<b\cdot c$	$a\cdot(b+c)=(a\cdot b)+(a\cdot c)$	

In addition, for every bounded *formula* ϕ, *there is an axiom*:

$$\phi(0) \wedge \big(\forall x \phi(x) \rightarrow \phi(x+1)\big) \rightarrow \forall x \phi(x)$$

The functions definable by $I\Delta_0$ are rudimentary (in the language of computational complexity, they belong to the linear-time hierarchy) [126, 39, 161, 107]. Other theories of bounded arithmetic correspond to other complexity classes. For example, in Buss's theory S_2^1 the Σ_1^b definable functions are exactly the polynomial time computable functions. At present we do not know much about which arguments can be formalized in the various theories of bounded arithmetic. Learning more might shed light on the P versus NP problem—for example, if strong pseudorandom number generators exist, then superpolynomial circuit size lowerbounds for SAT are independent of the theory S_2^2 [140, 53].[3]

[3]The connections with cryptography and complexity go the other direction as well, for example, if the RSA function is secure against polynomial-size circuits, then S_2^1 cannot prove Fermat's little theorem [105].

For many classical results, it is unknown whether or not they can be proved in $I\Delta_0$. In particular, it is not known whether or not $I\Delta_0$ can prove the infinitude of the primes. It is known that if $I\Delta_0$ can prove the pigeonhole principle, then $I\Delta_0$ can prove the infinitude of the primes [160, 128]. However, the relationship between $I\Delta_0$ and the pigeonhole principle is sticky.

DEFINITION 2.3. *Let $I\Delta_0(R)$ denote $I\Delta_0$ with its language expanded to include the relation symbol R. Let $php(R)$ denote the following sentence in the language of $I\Delta_0(R)$*:

$$\forall n \neg \Big(\big(\forall x_0 < n+1\, \forall x_1 < n+1\, \forall y < n\ (x_0 = x_1) \vee \neg R(x_0, y) \vee \neg R(x_1, y) \big) \\ \wedge \big(\forall x < n+1\, \exists y < n\, R(x, y) \big) \Big)$$

THEOREM 2.6. *There is no $I\Delta_0(R)$ proof of $PHP(R)$.*

What Theorem 2.6 means for $I\Delta_0$ is that there is no "schematic" $I\Delta_0$ proof of the pigeonhole principle, one in which we take the proof of $php(R)$ in $I\Delta_0(R)$ and then substitute a bounded formula ϕ for R to obtain an $I\Delta_0$ proof of $php(\phi)$. It is still open whether or not $I\Delta_0$ can prove $php(\phi)$ for every bounded ϕ, but such proofs would have to be done on a formula-by-formula basis that makes use of the structure of ϕ.

We now sketch the proof of Theorem 2.6.

DEFINITION 2.4. [128] *Let ϕ be a bounded formula in the language $I\Delta_0(R)$ with free variables $x_1, \ldots, x_m$. For each $\vec{n} \in \mathbb{N}^m$ we define $\langle \phi \rangle_{\vec{n}}$ by induction on the structure of ϕ as follows*:

ϕ	$\langle \phi \rangle_{\vec{n}}$
$s(\vec{y}) = t(\vec{y})$	0, *if* $s(\vec{n}) \neq t(\vec{n})$, *or* 1, *if* $s(\vec{n}) = t(\vec{n})$
$s(\vec{y}) < t(\vec{y})$	0, *if* $s(\vec{n}) \geq t(\vec{n})$, *or* 1, *if* $s(\vec{n}) < t(\vec{n})$
$R(s(\vec{y}), t(\vec{y}))$	$x_{i,j}$ *where* $i = s(\vec{n})$ *and* $j = t(\vec{n})$
$\eta \vee \theta$	$\langle \eta \rangle_{\vec{n}} \vee \langle \theta \rangle_{\vec{n}}$
$\eta \wedge \theta$	$\langle \eta \rangle_{\vec{n}} \wedge \langle \theta \rangle_{\vec{n}}$
$\eta \to \theta$	$\langle \eta \rangle_{\vec{n}} \to \langle \theta \rangle_{\vec{n}}$
$\neg \eta$	$\neg \langle \eta \rangle_{\vec{n}}$
$\exists y < t(\vec{x})\, \eta(y, \vec{x})$	$\bigvee_{j=1}^{b} \langle \eta(j, \vec{x}) \rangle_{\vec{n}}$ *where* $b = t(\vec{n})$
$\forall y < t(\vec{x})\, \eta(y, \vec{x})$	$\bigwedge_{j=1}^{b} \langle \eta(j, \vec{x}) \rangle_{\vec{n}}$ *where* $b = t(\vec{n})$

Because the terms of this language are polynomials, each existential (universal) quantifier translates into a disjunction (conjuntion) with at most polynomially many disjuncts (conjuncts). An easy induction argument bounds the size and alternation depth of the propositional translations in terms of the first-order formula.

LEMMA 2.7. [128] *Let ϕ be a bounded formula in the language of $I\Delta_0(R)$ with free variables $x_1, \ldots, x_m$. There exist constants $c, d \in \mathbb{N}$ so that for all $\vec{n} \in \mathbb{N}^m$ with $N = \max_i n_i$, $|\langle\phi\rangle_{\vec{n}}| \leq N^c$ and $dp(\langle\phi\rangle_{\vec{n}}) \leq d$.*

The translation preserves the structure of $I\Delta_0(R)$ proofs (up to small number of "clean-up" steps).

THEOREM 2.8. [128] *Let ϕ be a bounded formula in the language of $I\Delta_0(R)$. Let $x_1, \ldots, x_m$ be the free variables of ϕ. If $I\Delta_0(R)$ proves $\forall\vec{x}\phi(\vec{x})$ then for each $\vec{n} \in \mathbb{N}^m$, the propositional formula $\langle\phi\rangle_{\vec{n}}$ has a Frege proof of alternation-depth $O(d)$ and size $(\max_i n_i)^{O(1)}$.*

A break-through result of Miklós Ajtai showed that there are no polynomial-size, constant-depth Frege proofs of the $n+1$ to n pigeonhole principle [6].

THEOREM 2.9. [6, 106, 130] *All depth d Frege proofs of PHP_n^{n+1} require size $\Omega\left(2^{n^{1/6^d}}\right)$.*

By Theorem 2.8, if $I\Delta_0(R)$ could prove $php_n^{n+1}(R)$, then that proof would translate into a family of polynomial-size, constant alternation-depth Frege proofs for PHP_n^{n+1}, contradicting Theorem 2.9. Thus we obtain Theorem 2.6.

§3. A menagerie of Frege-like proof systems.

In this section, we describe and compare many propositional proof systems that come from satisfiability algorithms and translations from theories of bounded arithmetic. We focus on propositional systems that can be viewed as Frege systems whose formulas are restricted to a particular syntactic form.

The notion used to compare all of these different propositional proof systems is *p-simulation*. We consider a proof system A to be at least as efficient as a system B if every B-proof can be efficiently translated into an A proof.

DEFINITION 3.1. *Let V_1 and V_2 be abstract propositional proof systems. We say that V_1 p-*simulates *V_2 if there is a polynomial time computable function f so that whenever τ is a tautology and $V_1(\tau, P) = 1$, $V_2(\tau, f(P)) = 1$. Let $g : \mathbb{N} \to \mathbb{N}$. We say that V_1 is g-separated from V_2 if there exists a infinite family of tautologies $\{\tau_n \mid n = 1, \ldots \infty\}$ so that for all n, $s_{V_2}(\tau_n) \geq g(s_{V_1}(\tau_n))$. These definitions are adapted in the obvious manner for refutation systems.*

Theorem 1.2 is usually stated in its stronger form: "All Frege systems p-simulate one another" [69].

3.1. Some Frege systems with restricted formulas.

Resolution: The resolution system and its connections with satisfiability algorithms were discussed at length in Subsection 2.1. Resolution also arises from translations of very weak theories of arithmetic into propositional logic, for example, the fragment of $I\Delta_0(R)$ that allows induction only on Σ_1^b formulas, cf. [100, 102]. Theorem 2.4 shows that resolution is not polynomially bounded.

Res (k): The Res (k) systems generalize resolution by using formulas that are k*-DNFs* instead of only clauses [102, 20]. The inference rules for Res (k) are the same as those for resolution, but with the addition of rules for *and-introduction* $\frac{x_1 \vee C \cdots x_k \vee C}{(\bigwedge_{i=1}^k x_i) \vee C}$ and *and-elimination* $\frac{(\bigwedge_{i=1}^k x_i) \vee C}{x_i \vee C}$. The Res (k) systems correspond to translations from certain weak theories of bounded arithmetic, for example, the fragment of $I\Delta_0(R)$ that allows induction only on Σ_2^b formulas, cf. [102]. The Res (k) systems also play a significant role in understanding the proof complexity of the *weak pigeonhole principles*, variants of the pigeonhole principle in which there are many more pigeons than holes [102, 110, 20, 152]. Because Res (k) systems are special kinds of Frege systems with constant alternation depth, Theorem 2.9 shows that the Res (k) systems are not polynomially bounded.

Constant-depth Frege systems: A *depth d Frege system* (or, d*-Frege*) is a Frege system in which the formulas are restricted to have alternation depth at most d. For a function $s : \mathbb{N} \to \mathbb{N}$, it is said that a family of tautologies $\{\tau_n \mid n = 1, \dots \infty\}$ possesses size $s(n)$ *constant-depth* Frege proofs if there exist a constant d so that each τ_n possesses a depth d Frege proof of size at most $s(n)$.

Constant-depth Frege systems generalize the resolution and Res (k) systems, which are depth one and depth two systems, respectively. Extensions to resolution based satisfiability algorithms, such as caching previously refuted subformulas, can be formalized as constant-depth Frege systems [28]. As shown in Subection 2.2, constant-depth proofs arise naturally from translations of proofs from the first-order theory $I\Delta_0(R)$ [128]. Theorem 2.9 shows that constant depth Frege systems are not polynomially bounded.

The exact formulation of the inference rules and axioms is not relevant —a variant of Theorem 1.2 shows that proofs can be translated between any two constant-depth Frege systems with at most a polynomial increase in size and a linear increase in depth.

Constant-depth Frege with counting axioms modulo m: "You cannot partition a set of odd cardinality into sets of size two." Facts like this are the beginnings of the connections between combinatorics and algebra, and they entail many other results (for example, the onto pigeonhole principle, which states that there is no onto, injective relation from $n+1$ pigeons to n holes, cf. [5]). These "counting principles" can be formulated as propositional formulas as follows: For a modulus $m > 1$ and finite set V of size indivisible by m, the formula $Count_m^V$ has a variable x_e for each $e \in \binom{V}{m}$, and:

$$\mathrm{Count}_m^V = \bigvee_{v \in V} \Big(\bigwedge_{\substack{e \in [V]^m \\ e \ni v}} \neg x_e \Big) \vee \bigvee_{\substack{e,f \in [V]^m \\ e \perp f}} \big(x_e \wedge x_f \big) .$$

Augment a depth d Frege system with substitution instances of the Count_m^V formulas, and we have a "d-Frege + CA_m" system. These systems are capable of efficiently formalizing arguments based on the unsatisfiability of linear equations modulo m, and more generally, arguments based on Hilbert's Nullstellensatz over $\mathbb{Z}_m$ [95].

It is known that for every m, constant-depth Frege systems with counting axioms modulo m are not polynomially bounded [5, 7, 27, 57]. Furthermore, when p and q are coprime, there is no sub-exponential size derivation of the counting principles modulo q from the counting principles modulo p [7, 27, 57].

Constant-depth Frege with counting gates: A natural extension to bounded arithmetic is the introduction of a bounded modular counting quantifier $Q_m x < t\ \psi(x)$, meaning that the number of $x < t$ with $\psi(x)$ satisfied is zero modulo m [127]. Consider the system that extends $I\Delta_0(R)$ with counting quantifiers modulo m. The analog of Theorem 2.8 for this system is that its proofs translate into propositional proofs in a *constant-depth Frege system with counting gates*. The lines of these systems are formulas that, in addition to $\wedge$, $\vee$ and $\neg$ gates, have arbitrary fan-in $MOD_{m,a}$ gates (which takes the value 1 when the sum of its inputs is a mod m and 0 otherwise). Alternation depth is calculated in a similar way, and the following axioms are added for reasoning about the $MOD_{m,a}$ gates:

1. $MOD_{m,0}(\emptyset)$,
2. $\neg MOD_{m,a}(\emptyset)$ for $a = 1, \ldots, m-1$,
3. $MOD_{m,a}(\phi_1, \ldots, \phi_k, \phi_{k+1}) \equiv \big(MOD_{m,a}(\phi_1, \ldots, \phi_k) \wedge (\neg\phi_{k+1})\big) \vee \big(MOD_{m,a-1}(\phi_1, \ldots, \phi_k) \wedge \phi_{k+1}\big)$ for all $a = 0, \ldots, m$ and $k \geq 0$.

We abbreviate the name of these systems to "d-Frege + CG_m". It is widely conjectured that constant-depth Frege systems with counting gates are not polynomially bounded, however, no unconditional proof of this is known. Interestingly, superpolynomial size lower bounds are known constant alternation depth formulas built from $\wedge$, $\vee$, $\neg$, and modular counting connectives [138, 155, 45], but it not known how to extend the techniques from formulas to proof systems.

Polynomial calculus: Clauses correspond naturally to polynomials over a field, for example the clause $x \vee \neg y \vee z$ can be viewed as the polynomial

$(1-x)y(1-z) = y - zy - xy + xyz$. The satisfying assignments of the clause are exactly the zero-one roots of the polynomial. In light of this, one way to solve the CNF satisfiability problem is to translate the given CNF into a system of polynomials over a field, and then use Groebner's basis algorithm to decide if the system of polynomials has a common zero-one root [66].

The steps of the Groebner basis algorithm over a field $\mathbb{F}$ can be simulated by the following refutation system: Treat as axioms the clauses of the input CNF (translated into polynomials), as well as $x^2 - x$ for each variable x (this enforces that all roots are zero-one). As inference rules, we may derive gf where f has been previously derived and g is an arbitrary polynomial, and we may derive $\alpha f + \beta g$, where $\alpha, \beta \in \mathbb{F}$ and both f and g have been previously derived. When 1 has been derived, we know that the initial set of clauses is unsatisfiable. Completeness for the polynomial calculus follows from Hilbert's Nullstellensatz [66, 129]. The size of a polynomial calculus derivation is the number of monomials that it contains, and it is known that over any field, the polynomial calculus is not polynomially bounded [66, 141, 93].

The translation of clauses into polynomials results is not size efficient. For example, $x_1 \vee \cdots \vee x_n$ translates into a polynomial with 2^n many monomials. The extension *polynomial calculus with resolution* (PCR) adds to the polynomial calculus an extension variable y_i for each original variable x_i along with an equation $y_i = 1 - x_i$. This system behaves much like the polynomial calculus, but it p-simulates resolution.

Nullstellensatz refutations: The Nullstellensatz refutation system is a restricted form of the polynomial calculus. Rather than iteratively derive new polynomials in the ideal generated by the polynomials of the CNF until a contradiction is found, a Nullstellensatz refutation lists an explicit combination that yields the polynomial "1". Each clause C_j is translated into a polynomial p_j. A Nullstellensatz refutation of $\bigwedge_{j=1}^m C_j$ is a list of polynomials $f_1, \ldots, f_m, g_1, \ldots, g_n$ so that $1 = \sum_{j=1}^m f_j p_j + \sum_{i=1}^n g_i(x_i^2 - x_i)$. The completeness of the system follows from Hilbert's Nullstellensatz. The size of a Nullstellensatz refutation is the number of monomials in the list $f_1, \ldots, f_m, g_1, \ldots, g_n$.

The Nullstellensatz refutation system over $\mathbb{Z}_p$ is closely related to constant-depth Frege proofs with counting axioms modulo q: Known lower bound proofs for constant-depth Frege systems with counting axioms modulo q build upon lower bounds on Nullstellensatz refutations [27, 57, 33]. Furthermore, constant-depth Frege systems with mod q counting axioms p-simulate Nullstellensatz refutations [95], size lower bounds for Nullstellensatz refutations are necessary for size lower bounds for constant-depth Frege systems with counting axioms.

Cutting planes: Clauses can be identified with inequalities over zero-one valued variables, for example, $x \vee \neg y \vee z$ translates into $x + (1 - y) + z \geq 1$, so that the satisfying assignments of the clause are exactly the zero-one solutions of the inequality. This allows us to bring powerful techniques from integer optimization to the Boolean satisfiability problem. One such method is the cutting planes technique for converting integer programming problems into linear programming problems by repeatedly applying the following "cutting planes inference rule": From $\sum_{i=1}^{n} ca_i x_i \geq a$, where $c \in \mathbb{N}$, $c > 0$, and each $a_i \in \mathbb{Z}$, infer $\sum_{i=1}^{n} a_i x_i \geq \lceil \frac{a}{c} \rceil$ [85, 63].

Cutting planes derivations can be viewed as a Frege-like refutation system that manipulates linear inequalities: There are axioms $0 \leq x$ and $x \leq 1$ for each variable x, and in addition to the cutting planes inference rule, we may add inequalities (from $f \geq a$ and $g \geq b$ infer $f + g \geq a + b$), and perform positive multiplication (from $f \geq a$ infer $\beta f \geq \beta a$ for any $\beta \geq 0$). The orginal CNF is unsatisfiable if and only if there is a derivation of $1 \geq 0$.

The cutting planes refutation system p-simulates resolution, and provides polynomial size refutations of PHP_n^{n+1}. Satisfiability algorithms based on so-called *pseudoboolean methods* construct cutting planes refutations when run on unsatisfiable CNFs [72, 17, 73].

Lovász–Schrijver refutations: The Lovász–Schrijver lift-and-project method is a way to convert zero-one programming problems into linear programming problems [108]. The first observation is that if one knows that a linear inequality $f(\vec{x}) \geq t$ holds and that all variables x_i take values in $[0, 1]$, then for any variable x_i, $x_i f(\vec{x}) \geq x_i t$ and $(1-x_i) f(x) \geq (1-x_i)t$. Of course, this derives quadratic inequalities that hold for all $\vec{x} \in [0, 1]^n$. However, by incorporating the fact that for Boolean solutions, $x_i^2 = x_i$ for all $i \in [n]$, one can derive new linear constraints that hold for all zero-one solutions to the problem. If one repeats this procedure n times, the resulting polytope will be the convex hull of the zero-one solutions to the problem. For problems of propositional logic, we can convert a CNF into inequality form and use the procedure to determine whether the set of solutions is empty.

There are many formulations of the Lovász–Schrijver systems, but we discuss only the LS_+ system, which is one of the most powerful variants commonly considered. The lines of an LS_+ refutation are quadratic inequalities over the rationals. There are axioms $x \geq 0$, $-x \geq -1$, and $x^2 - x = 0$ for every variable x, and $f^2 \geq 0$ for every affine function f. From a linear inequality $f \geq t$ we may infer $xf \geq xt$ and $(1 - x)f \geq (1 - x)t$ for any variable x. From $f \geq a$ and $g \geq b$ we may infer that $f + g \geq a + b$, and from $f \geq a$ we may infer $\beta f \geq \beta a$ for any $\beta \geq 0$. The orginal CNF is unsatisfiable if and only if there is a derivation of $1 \geq 0$.

Presently, it is not known if the LS_+ refutation system is polynomially bounded. However, two special cases, the LS_0 system (in which multiplication is noncommutative and $-xy$ does not cancel yx) [70], and the tree-like LS_+ system [96], are known to not be polynomially bounded.

Ordered-binary decision diagrams: The Boolean satisfiability problem would be trivial if the CNFs considered could be efficiently reduced to a canonical form—to decide if a CNF is unsatisfiable, we would need only check that its canonical form is the constant false. Ordered binary decision diagrams (OBDDs) are data structures for canonically representing Boolean functions[4] [46, 47, 117]. The catch is that the canonical OBDD can sometimes be exponentially large. However, OBDDs often have reasonable sizes for Boolean functions encountered in engineering practice, and they are widely used in circuit synthesis and model checking, cf. [46, 47, 114, 65].

Presently, there are two kinds of satisfiability algorithms based upon OBDDs in the satisfiability literature. The first kind builds the OBDD for the given CNF and tests if it is the constant false [46, 86, 3, 125, 91, 154]. This approach can be extended to eliminate variables using existential quantification (a technique called *symbolic quantifier elimination* [86, 125, 91]). The second kind of approach uses the OBDDs to succinctly represent an exponentially large resolution or breadth-first search [61, 62, 120, 121, 122]. Such techniques are called *compressed search* or *compressed resolution*.

Algorithms that explicitly construct OBDDs and symbolic quantifier elimination algorithms can be simulated by the OBDD-based propositional proof system formalized by Atserias, Kolaitis and Vardi [22]. In this system, a variable ordering for constructing OBDDs is fixed, the clauses of the CNF are each transformed into an OBDD, and new OBDDs are constructed according to the following inference rules: From an OBDD A, then we may infer any OBDD B such that $A \Rightarrow B$, (in particular, from an OBDD $A(x, \vec{y})$ we may infer $\exists x A(x, \vec{y})$), and from two OBDDs A and B we may infer $A \wedge B$. The given CNF is unsatisfiable if and only if this system can derive the constantly-false OBDD.

Recently announced results show that OBDD refutations are not polynomially bounded [103, 151]. No nontrivial bounds are known for proof systems corresponding to the compressed search or compressed resolution algorithms.

In contrast with the other proof systems discussed in this section, it is not known whether or not Frege systems p-simulate OBDD refutations. This is because we do not know how to convert OBDDs into Boolean formulas without an exponential increase in size.

[4] More precisely, an OBDD is a read-once branching program in which the variables appear according to a fixed order along every path. It is the fixed ordering that guarantees canonicity.

System	p-simulates	Cannot p-simulate
resolution		Res (2) [19, 152], cutting planes [6], Nullstellensatz
Res (k)	resolution, Res $(k-1)$	Cutting planes, Res $(k+1)$ [152, 150]
d-Frege	Res (k), $(d-1)$-Frege	Cutting planes [6], $(d+1)$-Frege [99]
d-Frege $+ CA_p$	$\mathbb{Z}_p$-Nullstellensatz [95], d-Frege	polynomial calculus mod p, constant-depth Frege $+ CG_p$ [94]
d-Frege $+ CG_p$	d-Frege $+ CA_p$	
$\mathbb{F}$-Nullstellensatz		resolution [48]
$\mathbb{F}$-polynomial calculus	$\mathbb{F}$-Nullstellensatz	Res $\left(\Theta(\log^2 n)\right)$ [141, 110]
$\mathbb{F}$-PCR	$\mathbb{F}$ polynomial calculus, resolution	Res $\left(\Theta(\log^2 n)\right)$ [141, 110]
Cutting planes	resolution	Frege systems [132]
Lovász–Schrijver	resolution	
OBDD refutations	resolution, Gaussian elimination, cutting planes with unary coefficients [22]	Frege systems [103]

FIGURE 4. Some known p-simulations and non-p-simulations between propositional proof systems.

Known simulation and non-simulations for these propostional proof systems are presented Figure 4.

3.2. Tree-like versus DAG-like proofs. For many propositional proof systems, proof sizes depend dramatically on the inference structure. In Subsection 2.1, we saw this for resolution: Theorem 2.3 shows that DAG-like resolution is exponentially separated from tree-like resolution. The notions of being tree-like or DAG-like apply to any Frege-like system that derives new formulas from axioms and hypotheses by the application of inference rules.

DEFINITION 3.2. *Let $C_1, \ldots, C_m$ be a derivation in some Frege-like system. The derivation is said to be* tree-like *if every formula is used as an antecedent to an inference rule at most once. Arbitrary derivations are said to be* DAG-like.

resolution	exponential separation [38, 40]
Res (k)	exponential separation [78]
constant-depth Frege	polynomial simulation [99]
C.D. Frege with counting axioms	”
C. D. Frege with counting gates	”
Frege systems	”
polynomial calculus	exponential separation [48]
cutting planes	exponential separation [44]
Lovász–Schrijver	unknown
OBDD refutations	unknown

FIGURE 5. Comparisons between the DAG-like and tree-like forms of some proof systems.

Tree-like systems arise from proof search algorithms based on backtracking search, and from translations of first-order proofs.[5] However, they can sometimes be less efficient than their DAG-like counterparts. For some propositional proof systems, the DAG-like system has an exponential speed-up over the tree-like system, but for others, the tree-like system p-simulates the DAG-like system. The most general result on this is Krajíček's Lemma which shows that for many proof systems, the tree-like system p-simulates the DAG-like system. Here we state it only for constant-depth Frege systems.

LEMMA 3.1 (Krajíček's Lemma, [99, 100]). *If τ has a size s, depth d DAG-like Frege proof, then τ has a size $O(s^2)$, depth $d + 1$ tree-like Frege proof.*

Currently known relationships between the tree-like and DAG-like versions of various propositional proof systems are summarized in Figure 5.

§4. Reverse mathematics of propositional principles. In addition to asking questions focused on propositional proof systems—“Is this system polynomially bounded? Does this system p-simulate that system?”—we can also ask questions tha focus on particular tautologies—“Which proof systems can

[5]Converting a fixed first-order proof into tree-like form incurs an exponential increase in the size of that first-order proof but this affects the sizes of the propositional translations by only a constant factor.

System	PHP_n^{n+1}	PHP_n^{2n}	$PHP_n^{n^2}$
Resolution	$2^{\Omega(n)}$ / $2^{O(n)}$	$2^{\Omega(n)}$ / $2^{O(n)}$	$2^{n^{\Omega(1)}}$ / $2^{O(n)}$
Res $\left(O\left(\frac{\log n}{\log\log n}\right)\right)$	$2^{n^{\Omega(1)}}$ / $2^{O(n)}$	$2^{n^{\Omega(1)}}$ / $2^{O(n)}$	$n^{O(1)}$ / $2^{O(n)}$
Res $\left(\log^{O(1)} n\right)$	$2^{n^{O(1)}}$ / $2^{O(n)}$	$n^{O(1)}$/ $n^{O(\log^{O(1)} n)}$	$n^{O(1)}$ / $n^{O(\log^{O(1)} n)}$
d-Frege	$2^{n^{\Omega(1)}}$ / $2^{O(n)}$	$n^{O(1)}$ / $n^{(\log n)^{O(1/d)}}$	$n^{O(1)}$ / $n^{O(\log^{(d)} n)}$
d-Frege+ CG_m	$2^{n^{\Omega(1)}}$ / $2^{O(n)}$	$n^{O(1)}$ / $n^{(\log n)^{O(1/d)}}$	$n^{O(1)}$ / $n^{O(\log^{(d)} n)}$
Frege	$n^{O(1)}$ / $n^{O(1)}$	$n^{O(1)}$ / $n^{O(1)}$	$n^{O(1)}$ / $n^{O(1)}$
Polynomial Calculus	$2^{\Omega(n)}$ / $2^{O(n)}$	$2^{\Omega(n)}$ / $2^{O(n)}$	$2^{\Omega(n)}$ / $2^{O(n)}$
PCR	$2^{\Omega(n)}$ / $2^{O(n)}$	$2^{\Omega(n)}$ / $2^{O(n)}$	$n^{O(1)}$ / $2^{O(n)}$

FIGURE 6. Known lower bounds / upper bounds for refutation sizes of pigeonhole principles. For all $m > n$, the cutting planes, Lovász–Schrijver, and OBDD systems each refute PHP_n^m with polynomial size refutations. References: Resolution lower bounds [87, 29, 38], resolution upper bound is folklore, Res(k) lower bounds [20, 152], Res(k) upper bounds [110], d-Frege lower bounds [6, 130, 106], d-Frege upper bounds [18, 128], Frege upper bound [51], and the polynomial calculus and PCR lower bounds [141, 93].

efficiently prove this tautology?". This can be thought of as reverse mathematics for propositional principles. Reverse mathematics studies the axioms that are necessary to prove theorems of mathematics (cf. [153]). In contrast, the propositional systems we consider are complete, so the focus is not on provability but on efficiency. Two families of principles that have received much attention are the weak pigeonhole principles and random 3-CNFs.

4.1. Weak pigeonhole principles. The *weak pigeonhole principle* states that for integers $m > n$, m pigeons cannot be injectively associated with n holes. Encoded as as the unsatisfiable CNF PHP_n^m, there are mn variables $x_{i,j}$, with interpretation "pigeon i goes to hole j", and for each $i \in [m]$, there is a clause $\bigvee_{j\in[n]} x_{i,j}$, and for each $i, i' \in [m]$ with $i \neq i'$, there is a clause $\neg x_{i,j} \vee \neg x_{i',j}$. When $m \gg n$, this CNF is called the *weak pigeonhole principle* because it is "more contradictory" than the $n + 1$ to n pigeonhole principle. Current understanding of the proof complexity of various weak pigeonhole principles is summarized in Figure 6.

The weak pigeonhole principle naturally arises in many contexts. In industrial satisfiability applications, it can arise when analyzing systems in which

System	Lower Bound
Resolution	$2^{\frac{n}{\Delta^{4/k-2+\epsilon}}}$ [64, 29, 38]
Res $\left(O(\sqrt{\log n/\log\log n})\right)$	$2^{n/2^{O(k^2)}}$ [20, 152, 8]
Constant Depth Frege	$\Omega(n)$
Polynomial calculus	$2^{\Omega(n)}$ [93, 36, 13]
Cutting planes	$\Omega(n)$
Lovász–Schrijver	$\Omega(n)$
OBDD Refutations	$\Omega(n)$

FIGURE 7. Best known lower bounds for refuting random 3-CNFs on Δn clauses. A lower bound S means that with probabilty $1 - o(1)$ as $n \to \infty$, a 3-CNF on Δn clauses requires size S to be refuted in that system.

many agents are competing for exclusive access to resources from a small pool, such as locks or channels [15]. Size lower bounds for refutations of the weak pigeonhole principles can be useful starting points for proving other results. By showing that a CNF F has a small derivation from PHP^m_n, we show that the smallest refutation of F is no smaller than the smallest refutation of PHP^m_n. Some striking results obtained through such techniques show that resolution-based methods cannot prove superpolynomial circuit size lower bounds for NP [137, 144].

In the study of bounded arithmetic, it is known that $I\Delta_0$ can prove the infinitude of primes from the $2n$ to n weak pigeonhole principle [128]. By Theorem 2.8, a necessary condition for $I\Delta_0(R)$ to be able to prove $php^{2n}_n(R)$ is that there exist polynomial size, constant depth Frege refutations of PHP^{2n}_n. It seems plausible that there are small constant-depth refutations of PHP^{2n}_n. The known upper bounds for PHP^{2n}_n and $PHP^{n^2}_n$ in constant-depth Frege are barely-superpolynomial. Furthermore, there are polynomial-size, constant-depth formulas that distinguish betweeen the cases when $< 1/3$ of the input bits are set to 1 and these case when $> 2/3$ of the input bits are set to 1 [136]. However, it is not known how to use these formulas in a refutation of PHP^{2n}_n.

4.2. Random 3-CNFs. It may be that for some propositional proof system $\mathcal{P}$, there are tautologies that require superpolynomially large proofs in $\mathcal{P}$, yet such tautologies are rare. We address this possibility by studying refutation sizes needed for random 3-CNFs.

Consider the experiment that generates a random 3-CNF on n variables by choosing Δn many 3-clauses uniformly, independently and with replacement. This distribution is called $F_3^{\Delta,n}$. The parameter Δ is called the *clause density*.

Empirical study of satisfiability algorithms suggests that there is a threshold value for Δ (it seems to be approximately 4.2), above which a random 3-CNF is almost surely unsatisfiable and below which a random 3-CNF is almost surely satisfiable [118]. Rigorously, it is known for each value of n that there is *some* threshold but its value has not been been rigorously determined [81]. This value is called the *satisfiability threshold.* Empirical studies also suggest that for values of Δ far below or far above the satisfiability threshold, it is computationally easy to solve satisfiability for CNFs of that clause density. However, when the clause density is close to the satisfiability threshold, random 3-CNFs seem to require exponential run times to be refuted by known satisfiability algorithms. Propositional proof complexity rigorously explains this behavior for several satisfiability algorithms and proof systems. Figure 7 summarizes currently known lower bounds for refuting random 3-CNFs.[6]

It seems plausible that random 3-CNFs of appropriate clause densities might require superpolynomial proofs of unsatisfiability in any propositional proof system. There is little to suggest that this is actually the case, but there is even less contradicting it. A surprising connection between this question and the computational complexity of approximating combinatorial optimization problems was discovered by Uri Feige: If refuting random 3-CNFs of arbitrarily large constant clause density requires superpolynomial size refutations in all abstract proof systems, then several approximation problems (that resist analysis via current PCP-based techniques) cannot be solved in polynomial time [80].

§5. Feasible interpolation.

Consider the propositional form of Craig's interpolation theorem:

THEOREM 5.1. *Let $\phi(\vec{x}, \vec{y})$ and $\psi(\vec{x}, \vec{z})$ be propositional formulas. If $\phi(\vec{x}, \vec{y}) \to \psi(\vec{x}, \vec{z})$ is a tautology, then exists a propositional formulas $\theta(\vec{x})$ so that $\phi(\vec{x}, \vec{y}) \to \theta(\vec{x})$ and $\theta(\vec{x}) \to \psi(\vec{x}, \vec{z})$ are both tautologies. The formula θ is called an* interpolant.

The standard proof of Theorem 5.1 guarantees the existence of an interpolant whose size is at most exponentially large in the number of variables. In general, the exponential blow-up is probably necessary: If the size of θ were bounded by a polynomial in the sizes of ϕ and ψ, then $NP \cap coNP$ would have polynomial size circuits [123]. However, for many propositional proof

[6]Recently Galesi and Lauria announced an exponential lower bound for refuting random 3-CNFs of constant clause density in the "polynomial calculus plus Res (k)" over finite fields of characteristic $\neq 2$ [82]. This system is the strongest (in terms of p-simulations) for which we have lower bounds for refuting random 3-CNFs.

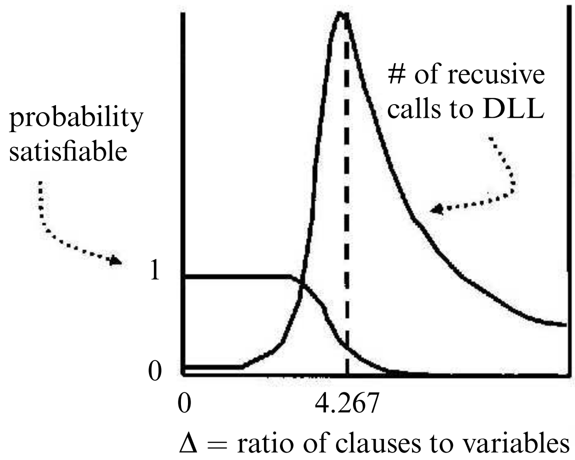

FIGURE 8. Overlay of graphs depicting the probability of satisfiability for a random 3-CNF with $n = 50$ many variables on Δn many clauses being satisfiable, and the the number of recursive calls made by DLL on randomly generated 3-CNFs with $n = 50$ variables and Δn many clauses. In the region near $\Delta = 4.267$, the probability of a random 3-CNF being satisfiable switches from 1 to 0. The number of recursive calls made by the DLL algorithm sharply spikes near this satisfiability threshold. Data from [118].

systems, we can bound the size of the interpolant by a polynomial in the size of the *proof* of $\phi(\vec{x}, \vec{y}) \rightarrow \psi(\vec{x}, \vec{z})$. This phenomenon is called *feasible interpolation*. Feasible interpolation has been used to prove size lower bounds for propositional proof systems [92, 41, 132, 101, 22, 151, 103], and it has found applications in formal verification and theorem proving [115, 116].

Systems known to have feasible interpolation include resolution [132], cutting planes [132], Lovász–Schrijver refutations [133], and the polynomial calculus [135]. To date, the absence of feasible interpolation has been guaranteed for non-trivial proof systems only under cryptographic assumptions. Among these results are: "If one-way functions exist, then Frege systems do not have feasible interpolation" [104], and "if factoring Blum integers is hard, then constant-depth Frege systems do not have feasible interpolation" [42, 43]. It is not known, even under cryptographic assumptions, whether or not Res (k) has feasible interpolation for any $k \geq 2$.

§6. Further connections with satisfiability algorithms.

6.1. Space complexity of refutations. Satisfiability algorithms based on clause learning and the Davis–Putnam procedure maintain a set of clauses called the *clause database*. These are previously derived consequences of the input CNF, saved for future re-use. The size of the clause database is a major bottleneck on the performance of such algorithms, so it is natural to ask "How large must the clause database be to refute a given CNF?" This leads to the notion of space complexity for resolution refutations.

DEFINITION 6.1 ([79] and [10]). *Let F be a CNF. A resolution refutation presented in* configuration form *is a sequence of sets of clauses $S_1, \ldots, S_m$ satisfying the following properties*:

1. $S_1 = \emptyset$
2. *The empty clause belongs to S_m.*
3. *Each S_{i+1} follows from S_i by either (a) Removing a clause from S_i (b) Resolving two clauses from S_i and adding the resolvent to S_{i+1} or (c) Adding a clause from F to S_i*

The clause space of $S_1, \ldots, S_m$ *is* $\max_{i \in [m]} |S_i|$. *The* variable space of $S_1, \ldots, S_m$ *is* $\max_{i \in [m]} \sum_{C \in S_i} |C|$. *For a resolution refutation Γ, let $sp(\Gamma)$ be the minimum space needed to present Γ as a sequence of configurations, and let $vsp(\Gamma)$ be the minimum variable space needed to present Γ as a sequence of configurations. Let $sp(F)$ denote the minimum clause space of a resolution refutation of F and let $vsp(F)$ denote the minimum variable space of a resolution refutation of F.*

Clearly, it the goal of any resolution based satisfiability engine worth its salt is to find a derivation that simultaneously has small size and small space. How are these two parameters related?

The space needed to refute a CNF is in general not the same as the size needed to refute the CNF. For example, the implication chain x_0, $\neg x_n$, and $\neg x_i \vee x_{i+1}$, for $i = 0, \ldots, n-1$, has a linear size, constant space refutation. However, there are many connections between the space needed to refute a CNF and its other requirements.

THEOREM 6.1. [79] *Let F be a CNF in n variables. Let $size(F)$ denote the least size of a resolution refutation of F, let $size_T(F)$ denote the least size of a tree resolution refutation of F, and let $height(F)$ denote the least height of a resolution refutation of F. We have that*: $sp(F) \leq n+1$, $size(F) \leq \binom{sp(F)+height(F)}{sp(F)}$, *and* $2^{sp(F)} - 1 \leq size_T(F)$.

It is known that for some unsatisfiable CNFs, it is impossible to simultaneously obtain optimal size and optimal space in tree-like resolution. The case for general resolution remains open.

THEOREM 6.2. [35] *There exists a faily of CNFs $\{T_n\}_{n=1}^{\infty}$ so that each T_n has a tree-like resolution refutation of size $O(n)$, but any resolution refutation Γ of T_n has $vsp(\Gamma) \cdot \log|\Gamma| = \Omega(n/\log n)$.*

A recently announced result of Hertel and Pitassi gives a very strong trade-off between optimal resolution size and optimal resolution variable space.

THEOREM 6.3. [89] *There is a family of CNFs $\{F_n\}$ and clauses $\{C_n\}$ such that $F_n \models C_n$, and all resolution derivations of C_n from F_n that use variable space $vsp(F_n)$ have size $2^{\Omega(n)}$, but, there exists a size $O(n)$, variable space $vsp(F_n) + 3$ derivation of C_n from F_n.*

Resolution width provides a lower bound for space, but the lower bound is not tight.

THEOREM 6.4. [21] *Let F be an unsatisfiable CNF, let $iw(F)$ denote the maximum width of a clause of F and let $w(F)$ denote the minimum width of a resolution refutation of F. We have that $sp(F) \geq w(F) - iw(F) + 1$.*

THEOREM 6.5. [124] *For all $k \geq 4$ there is a family of k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $O(n)$ so that $w(F_n) = O(1)$ but $sp(F_n) = \Theta(\log n)$.*

6.2. Automatizability. A big difference between propositional proof complexity and the study of satisfiability algorithms is that just because a tautology has a short proof, there is not necessarily a good way to automatically find it.

DEFINITION 6.2. *Let $f : \mathbb{N} \to \mathbb{N}$ be given. A propositional proof system $\mathcal{P}$ is said to be f-*automatizable *if there is an algorithm A so that for every tautology τ, whenever there is an $\mathcal{P}$ proof of size S, the algorithm A terminates within $f(S)$ steps and outputs some $\mathcal{P}$ proof of τ.*

There are some positive results for automatizability: Tree-like resolution is $n^{O(\log n)}$ automatizable [38, 29], as is the treelike polynomial calculus over any field [66]. Negative results depend upon conjectures in computational complexity and cryptography. It is known that neither resolution no tree-resolution is polynomial-time automatizable unless the $W[P]$ hierarchy in parameterized complexity collapses [12]. Moreover, there is no automatizability for Frege systems if one-way functions exist [104], and under the assumption that "factoring Blum integers is hard", there is no automatizability for any system that can polynomially simulate constant-depth Frege systems [43, 42].

For many purposes, it would suffice if the existence of a small $\mathcal{P}$ proof guaranteed that we could quickly find a proof in some other system $\mathcal{Q}$. This leads to the related notion is of *weak automatizability* [19]. It turns out the resolution is weakly automatizable if and only if Res (2) has feasible interpolation [19].

6.3. Lower bounds for satisfiability algorithms on satisfiable formulas. Propositional proof complexity can tell us why a satisfiability algorithms take a long time to run on some unsatisfiable CNF, but what can be said about the running times of satisfiability algorithms on satisfiable CNFs?

When analyzing how a DLL-style backtracking algorithm performs on a satisfiable CNF, you must take into account the method that chooses the branching variable and which setting ($x = 0$ or $x = 1$) to explore first. This is because a completely unrestricted, exponential-time heuristic could find a satisfying assignment, and then guide the DLL search to that assignment within n decision steps.

The family of *myopic branching heuristics* has been successfully analyzed on satisfiable CNFs. When choosing the branching variable and which branch to explore first, a myopic heuristic can make use of the partial assignment at that point of the recursion tree, inspect at most $n^{1-\epsilon}$ many clauses of the input CNF, make full use of the formula with all negation signs removed, and make full use of a variable frequency-analysis from the full CNF. A somewhat orthogonal class of variable-centric heuristics has also been studied. In variable-centric heuristics, the variable to branch upon is selected using an arbitrary method, but the decision whether to first explore the branch with $x = 0$ or the branch with $x = 1$ is made randomly.

THEOREM 6.6. [11] *For every myopic DLL algorithm $\mathcal{A}$ that reads at most $K(n)$ clauses per step, for each n there is a satisfiable formula Φ_n so that with probability* $1 - 2^{\Omega((n/(K \log^{O(1)})))}$, *$\mathcal{A}$ requires time* $2^{\Omega(n/\log^{O(1)} n)}$ *on input Φ_n.*

For each $k \geq 3$ there is $c > 0$ and a family of satisfiable $(k+1)$-CNF formulas G_n so that for every DLL algorithm $\mathcal{A}$ with a variable-centric branching heuristic, the probability that $\mathcal{A}$ finds a satisfying assignment on input G_n with fewer than 2^{cn} steps is at most 2^{-n}.

A particularly interesting class of satisfiable CNFs are random 3-CNFs with clause densities just below the satisfiability threshold. These seem to be hard, however, unconditional results are only known for very weak branching heuristics that use some fixed order for branching upon variables along every branch of the search tree. (DLL with such a heuristic is called *ordered DLL*.) For ordered DLL, it is known for a range of clause densities just below the satisfiability threshold, a constant-fraction of the random k-CNFs require exponential run times to refute.

THEOREM 6.7. [2, 1] *With uniformly positive probability, ordered-DLL requires time $2^{\Omega(n)}$ on random k-CNFs of clause density c, where $k = 4$ and $c > 7.5$, or $k \geq 5$ and $c > (11/k)2^{k-2}$. Moreover, a random k-CNF of clause density c is almost-surely satisfiable if $k = 4$ and $c < 7.91$, or $k \geq 5$ and $c < 2^k(\ln 2) - (k+4)/2$.*

§7. Beyond the Frege systems.

7.1. Some powerful propositional proof systems. These are some of the propositional proof systems conjectured to be more superpolynomially more efficient than the Frege systems. No superpolynomial proof size lower bounds are known for any of these systems, and the only p-simulations known are the obvious ones.

Extended Frege: Extended Frege systems extend Frege systems with the ability to introduce definitions: At step $i+1$ of a derivation, the formula A_{i+1} may follow from $A_1, \ldots, A_i$ either by the usual inference rules of the

Frege system, or, A_{i+1} can be of the form $x \leftrightarrow B$ where x is a variable not appearing in $A_1, \dots, A_i$, and B is a Boolean formula. The variable x is called an *extension variable*.

Extended Frege systems can be also be defined as Frege systems that manipulate circuits instead of formulas. For this reason, the distinction between Frege systems and extended Frege systems can be viewed as analogous to the distinction between Boolean formulas and Boolean circuits in circuit complexity.

Quantified Frege systems: Quantified Boolean formulas extend Boolean formulas by allowing the introduction of quantifiers, $\exists x F(x, \vec{y})$ or $\forall x F(x, \vec{y})$, where the semantics is that $\exists x F(x, \vec{y})$ is satisfied if and only if $F(0, \vec{y}) \vee F(1, \vec{y})$ is satisfied and $\forall x F(x, \vec{y})$ is satisfied if and only if $F(0, \vec{y}) \wedge F(1, \vec{y})$ is satisfied. Quantified Frege systems are analagous to standard first-order proof systems, except that the "terms" are propositional formulas. Quantified Boolean formula are conjectured to have exponentially more succinct representations for some Boolean functions than is possible with Boolean formulas, but this has not been proved. It is easily seen that quantified Frege systems p-simulate extended Frege systems, cf. [100].

Propositional ZFC: A proof for a tautology need not be written in a classical propositional calculus, indeed, it might be more intuitive and succinct to bring to bear some higher mathematic formalized in *ZFC* (or Peano's arithmetic, or whatever theory you prefer). The proof would be formalized in some standard way, and the verification procedure would check that each line of the proof is an instance of an axiom or follows from the preceding by application of the inference rules. All of the other propositional proof systems discussed in this survey can be p-simulated by such a system, as can any proof system whose correctness is provable in ZFC.

What tautologies might require superpolynomially large proofs in powerful systems such as these? As discussed in Section 4.2, it seems plausible that random 3-CNFs of certain clause densities almost surely require superpolynomially-large proofs in any proof system, but other than that, there are no candidates.

Possible separations between these strong systems such as these are more difficult to identify. It is quite a challenge to even propose natural propositional tautologies that give superpolynomial separations between such systems. If we do not mind unnatural tautologies, then it suffices to cosider partial consistency statements- propositional encodings of statements such as "If P is a $\mathcal{P}$ proof of τ then τ is a tautology". It turns out that, for proof systems $\mathcal{P}$ that can p-simulate Frege systems, if $\mathcal{P}$ does not p-simulate a proof system $\mathcal{Q}$, then $\mathcal{P}$ requires superpolynomial size to prove the partial consistency statements for $\mathcal{Q}$.

THEOREM 7.1. [68, 104, 52] *Let $\mathcal{P}$ be a propositional proof system that p-simulates Frege systems, and let $\mathcal{Q}$ be any propositional proof system. $\mathcal{P} + Con_{\mathcal{Q}}$ p-simulates $\mathcal{Q}$.*

Separations based on partial consistency would be great to have—but they reveal little about the kinds of arguments that can be efficiently performed in one proof system but not in another. For the problem of separating extended Frege systems from Frege systems, there are natural combinatorial tautologies that are have polynomial size extended Frege proofs and are conjectured to require superpolynomial size Frege proofs. The partial consistency of extended Frege systems can be shown equivalent to a combinatorial statement about the non-existence of sinks in certain directed graphs [23]. Another candidate is a propositional encoding of the principle $AB = I \Rightarrow BA = I$ [156]. The latter seeks to make use of the conjecture that the inverse of a matrix cannot be computed by a polynomial-size Boolean formula.

7.2. Optimal proof systems. It may well be that there is some "universal" propositional proof system that p-simulates all other propositional proof systems. In the literature, such a proof system is called p-optimal. Whether or not a p-optimal proof system exists is a major open question, and there is little evidence either way. The existence of p-optimal proof systems is guaranteed by implausible computational complexity hypotheses—for example, "if $EXPEXP = NEXPEXP$ then there is a p-optimal proof system [104, 98]. On the other hand, if p-optimal proof systems exist, then there also exist complete sets for semantic classes such as UP [139, 148, 98]—a consequence that is unexpected, but not particularly controversial.[7]

The most natural candidate for a p-optimal proof system is propositional ZFC, but this is possibly an artifact of the fact that we develop propositional proof systems and prove their consistency inside ZFC. It may be that bringing in assumptions from beyond ZFC could enable more succinct proofs of propositional tautologies.

This leaves us with three possibilities:

1. Propositional ZFC (and perhaps something weaker) is p-optimal. This would be a remarkable conservation result: For the purposes of certifying propositional tautologies, there would no benefit to adding further axioms.
2. Propostional ZFC is not p-optimal, but some other system is. In this case, identifying a p-optimal system and its properties would be of utmost importance.
3. There is no p-optimal propositional system. If this is the situation, then independence raises its head in one of the most basic tasks of logic: No matter what (polynomial-time decidable) axioms of mathematics you accept, the correctness of some method for certifying propositional tautologies is independent of those axioms.

[7] A statement in computational complexity equivalent to the existence of a p-optimal proof system is given in [104].

Part 2. Some lower bounds on refutation sizes.

Much of the appeal of propositional proof complexity lies in the fact that we can prove limitations for non-trival proof systems. In this section, we present size lower bounds for refutations of random 3-CNFs and weak pigeonhole principles. These have all been proved in recent years, and use a family of related techniques that build upon and extend the size-width trade-off for resolution.

Space limitations prevent us from discussing all known techniques for establishing proof size lower bounds. Many interesting results and techniques have been omitted, among them: The use of feasible interpolation to establish size lower bounds for cutting planes [92, 41, 132], Lovász–Schrijver [32], and OBDD refutations [151, 103], "rank" lower bounds for cutting planes and Lovász–Schrijver refutations [49, 9], degree lower bounds for the Nullstellensatz system [59, 57, 54] and the polynomial calculus [66, 141, 93, 56, 13], using extensions of Håstad's switching lemma to establish exponential size lower bounds for constant-depth refutations of the $n + 1$ to n pigeonhole principle [6, 130, 106], and lower bounds for constant-depth Frege systems with counting axioms via a combination of the Håstad switching lemma and Nullstellensatz degree lower bounds [5, 7, 27, 147, 57, 33, 94].

Background from probabilistic combinatorics. Our presentation is not self-contained: We omit proofs of standard lemmas from discrete probability and probabilistic combinatorics.

A common framework in proof complexity is to use expansion in the clauses of the CNF (or some higher-level constraints) to guarantee that the CNF requires large width to refute in resolution. For a thorough introduction to expansion and its applications in discrete mathematics and computer science, see [90]. The following definition is more often phrased in the language of bipartite graphs, but matrix notation better suits our perspective.

DEFINITION 7.1. *Let A be a Boolean matrix with m rows and n columns. For a set of rows, $I \subseteq [m]$, we define the* boundary of I in A, *$\partial_A(I)$ as $\partial_A(I) = \{j \in [n] : |\{i \in I \mid A_{i,j} = 1\}| = 1\}$.*

We say that A an (r, η)-boundary expander if for every $I \subseteq [m]$ with $|I| \leq r$ we have that $|\partial_A(I)| \geq \eta|I|$. We say that an (r, η)-boundary expander is a (d, r, η)-boundary expander if every column of A contains at most d ones.

LEMMA 7.2. *Let $\Delta > 0$ be a constant, and let $m = \Delta n$. Let A be a random matrix from $\{0, 1\}^{m \times n}$ so that A is chosen uniformly among matrices with exactly three ones in each row. For all constants $\Delta > 0$, $\eta < 1$, there exists some constant δ so that with probability $1 - o(1)$, $A_{n,\Delta}$ is a $(\delta n, \eta)$-boundary expander.*

In some of the lower bound arguments, we make use of the following form of the Chernoff–Hoeffding bounds:

LEMMA 7.3 (Chernoff–Hoeffding bounds, cf. [113]). *Let $X_1, \dots, X_n$ be independent random indicator variables. Let $\mu = E\left[\sum_{i=1}^n X_i\right]$. For every $\epsilon > 0$: $Pr\left[\sum_{i=1}^n X_i < (1-\epsilon)\mu\right] \leq e^{-\epsilon^2\mu/2}$ and $Pr\left[\sum_{i=1}^n X_i > (1+\epsilon)\mu\right] \leq e^{-\frac{\epsilon^2\mu}{2(1+\epsilon/3)}}$.*

COROLLARY 7.4. *Let $X_1, \dots, X_n$ be independent random indicator variables. Let $\mu = E\left[\sum_{i=1}^n X_i\right]$. $Pr\left[\sum_{i=1}^n X_i < \frac{\mu}{2}\right] \leq e^{-\mu/8}$ and $Pr\left[\sum_{i=1}^n X_i > 2\mu\right] \leq e^{-3\mu/8}$. Furthermore, for any B be with $B \geq \mu$, $Pr\left[\sum_{i=1}^n X_i > 2B\right] \leq e^{-3B/8}$.*

PROOF. The first two inequalities specialize Lemma 7.3 with $\epsilon = 1/2$ and $\epsilon = 1$, respectively. For the third claim, choose a family of independent, random indicator variables $X_1^*, \dots, X_n^*$ with $X_i \leq X_i^*$ for each $i = 1, \dots, n$, and $\sum_{i=1}^n EX_i^* = B$. The probability that $\sum_{i=1}^n X_i$ exceeds $2B$ is less than the probability that $\sum_{i=1}^n X_i^*$ exceeds $2B$, which by the preceding claim is at most $e^{-3B/8}$. ⊣

§8. The size-width trade-off for resolution. The task of proving lower bounds on the sizes of resolution refutations has been simplified in recent years by the discovery of the *size-width trade-off*: If every resolution refutation of a CNF F contains a clause with many variables, then every resolution refutation of F is large.

DEFINITION 8.1. *The* width *of a clause is the number of variables appearing in the clause; the* width *of a resolution derivation is the maximum width of a clause in the derivation. For a set of clauses F, $w(F)$ denotes the minimum width of a resolution refutation of F, $S(F)$ denotes the minimum size of a DAG-like resolution refutation of F, and $S_T(F)$ denotes the minimum size of a tree-like resolution refutation of F. The* initial width of F, written $iw(F)$, *is the maximum width of a clause in F.*

THEOREM 8.1. [38] *Let F be an unsatisfiable set of clauses in n variables. We have that $w(F) - iw(F) \leq \log S_T(F)$ and that $w(F) - iw(F) \leq 1 + 3\sqrt{n \ln S(F)}$.*

COROLLARY 8.2. *Let F be an unsatisfiable set of clauses on n variables. We have that $S_T(F) \geq 2^{(w(F)-iw(F)}$ and that $S(F) \geq 2^{\Omega\left((w(F)-iw(F))^2/n\right)}$.*

While the size-width trade-off is sufficient for establishing resolution size lower bounds, it is not necessary. In particular, the quality of the lower bound falls off rapidly with the number of variables in the CNF, and it gives only trivial bounds when minimum width of a refutation is at most the square-root of the the number of variables. This can be a wild underestimation of minimum refutation size, as there are unsatisfiable CNFs on n variables that require resolution refutations of size $2^{n^{\Omega(1)}}$ but which posses refutations of width at most $o(\sqrt{n})$. This limitation to the applicability of the size-width trade-off can be overcome with a sparsification trick (cf. Subsection 8.1), or it can require completely new techniques (cf. Section 11).

The size-width trade-off is not known to apply to stronger proof systems (in particular, nothing like it is known to hold for the Res (k) systems), but ideas developed here will be useful when analyzing those stronger systems in Sections 9 and 10.

The presentation here closely follows [66] and [38]. As in those works, the proof of Theorem 8.1 builds upon a sequence of simple lemmas.

LEMMA 8.3. *For $v \in \{0,1\}$, if $F \restriction_{x=v}$ can be refuted in width $\leq w$, then there is a width $\leq w+1$ derivation of x^{1-v} from F.*

PROOF. Let Γ be the width w refutation of $F \restriction_{x=v}$. Without loss of generality, no clause of Γ contains the variable x. Obtain a derivation Γ' as follows: By using subsumption inferences, infer $C \vee x^{1-v}$ for every $C \in F$. Follow this by a derivation that follows the structure of Γ, but in which every clause C has been replaced by $C \vee x^{1-v}$. The sequence of clauses Γ' clearly as width at most $w+1$. Moreover, it is a valid resolution derivation from F: If $C \in F \restriction_{x=v}$, then either $C \vee x^{1-v} \in F$ or $C \in F$; in the either case, $C \vee x^{1-v}$ follows from a clause of F by subsumption. Clearly all subsumption inferences in Γ become valid subsumption inferences in Γ'. Consider the case when C follows from a resolution step applied to $C \vee y$ and $C \vee \neg y$. Because the variable x appears in no clause of Γ, $y \neq x$, and thus $C \vee x^{1-v}$ follows from a resolution step applied to $C \vee x^{1-v} \vee y$ and $C \vee x^{1-v} \vee \neg y$. ⊣

LEMMA 8.4. *For all CNFs F, all literals x, all $k \in \mathbb{N}$, and all values $v \in \{0,1\}$, if $w(F \restriction_{x=v}) \leq k-1$ and $w(F \restriction_{x=1-v}) \leq k$ then $w(F) \leq \max\{k, iw(F)\}$.*

PROOF. By Lemma 8.3, there is a resolution derivation of x^{1-v} from F of width at most k. Take this derivation, and then resolve x^{1-v} with every clause of F that contains x^v to derive $F \restriction_{x=1-v}$. This step requires width at most $iw(F)$. Now refute $F \restriction_{x=1-v}$; by hypothesis, this can be done with width at most k. ⊣

LEMMA 8.5. *For any set of clauses F, $w(F) \leq iw(F) + \log S_T(F)$.*

PROOF. We will show that for every set of clauses F and every tree-like refutation of F, Γ, $w(F) \leq iw(F) + \log|\Gamma|$. This proves the claim by taking a refutation of minimum size.

Induct on the number of variables in F, denoted by n, and $\lceil \log|\Gamma| \rceil$, denoted by b. If $b=0$, then Γ is a length 1 refutation, and thus $\emptyset \in F$. Therefore, the minimum width of a refutation of F is $0 \leq w(F)+b$. Note that if $n=0$, we necessarily have that $b=0$.

For the induction step, let $n, b \geq 1$, and assume that for all sets of clauses F' in fewer than n variables and all tree refutations Γ' of F', $w(F') \leq iw(F') + \log|\Gamma'|$, and that for all sets of clauses F' on n variables such that $\lceil \log|\Gamma'| \rceil \leq b-1$, $w(F') \leq iw(F') + \log|\Gamma'|$. Let a set of clauses F, and a tree-like resolution refutation of F, Γ, be be given so that $b = \lceil \log|\Gamma| \rceil$. The final clause of Γ is $\emptyset$, so the final inference is the resolution of x and $\neg x$

for some variable x. Let Γ_x and $\Gamma_{\neg x}$ be the sub-derivations of Γ that lead to x and $\neg x$, respectively. Note that $|\Gamma| = 1 + |\Gamma_x| + |\Gamma_{\neg x}|$. Without loss of generality, $|\Gamma_x| \leq 2^{b-1}$. Notice that $\Gamma_x \restriction_{x=0}$ is a refutation of $F \restriction_{x=0}$ in $n-1$ variables and of size at most 2^{b-1}; apply the induction hypothesis to conclude that it has resolution refutation of width at most $b-1$. Similarly, $\Gamma_{\neg x} \restriction_{x=1}$ is a refutation of $F \restriction_{x=1}$ in $n-1$ variables and of size at most 2^b; apply the induction hypothesis to conclude that it has resolution refutation of width at most b. By Lemma 8.4, $w(F) \leq b + iw(F)$. ⊣

Lemma 8.6. *For any set of clauses F, $w(F) \leq iw(F) + 1 + 3\sqrt{n \ln S(F)}$.*

Proof. Let Γ be a minimum size refutation of F, and let $S = |\Gamma|$. Set $d = \sqrt{2n \ln S(F)}$, and $a = (1 - d/2n)^{-1}$. Let W be the set of clauses from F of width $\geq d$. Call such clauses "wide". We show by induction on n and b that if $|W| < a^b$ then $w(F) \leq iw(F) + d + b$. Observe that the claim trivially holds when $d \geq n$, because every refutation that uses at most n variables has width at most n, so we may assume that $d < n$. In the base case, $b = 0$ and there are no clauses in Γ of width more than d, so $w(F) \leq d \leq iw(F) + d$. In the induction step, suppose that $|\Gamma| < a^b$. Because there are $2n$ literals making at least $d|W|$ appearances in the wide clauses, there is a literal x that appears in at least $\frac{d}{2n}|W|$ of the wide clauses. Setting $x = 1$, $\Gamma \restriction_{x=1}$ is a refutation of $F \restriction_{x=1}$ with at most $\left(1 - \frac{d}{2n}\right)|W| < a^{b-1}$ many wide clauses. By the induction hypothesis, $w(F \restriction_{x=1}) \leq d + iw(F) + b - 1$. On the other hand, $\Gamma \restriction_{x=0}$ is a refutation with at most $|W| < a^b$ many large clauses and in $n-1$ many variables. By induction on the number of variables, $w(F \restriction_{x=0}) \leq d + iw(F) + b$. Therefore by Lemma 8.4, $w(F) \leq d + iw(F) + b$. This concludes the proof by induction.

Now, for any size S refutation of Γ, we have that $|W| < a^{\lfloor \log_a(|W|) \rfloor + 1}$ and that $|W| \leq S$. Applying the inequality demonstrated in the previous paragraph (with the same definitions for a and d), we have $w(F) \leq iw(F) + \lfloor \log_a(|W|) \rfloor + 1 + d \leq iw(F) + \log_a(S) + 1 + d$ so that:

$$\begin{aligned} w(F) - iw(F) &\leq 1 + d + \log_a(S) = 1 + d + \log_{\left(\frac{2n}{2n-d}\right)}(S) \\ &= 1 + d + \log_{\left(1+\frac{d}{2n-d}\right)} S = 1 + d + (\ln S) \log_{\left(1+\frac{d}{2n-d}\right)}(e) \\ &= 1 + d + (\ln S)\left(\ln\left(1 + (d/(2n-d))\right)\right)^{-1}. \end{aligned}$$

Because $0 \leq d < n$, we have that $0 \leq d/(2n-d) < 1$, so we may apply the inequality $\ln(1+x) \geq x - x^2/2 \geq x/2$ with $x = d/(2n-d)$. Therefore:

$$\begin{aligned} w(F) - iw(F) &\leq 1 + d + (\ln S)\left(d/2(2n-d)\right)^{-1} \\ &\leq 1 + d + (\ln S)(2 \cdot 2n/d) \\ &= 1 + \sqrt{2n \ln S} + 2 \cdot 2n(\ln S)/(\sqrt{2n \ln S}) \\ &= 1 + 3\sqrt{2n \ln S}. \end{aligned}$$ ⊣

8.1. Exponential lower bounds for the $2n$ to n weak pigeonhole principle. We cannot directly apply the size-width trade-off of Corollary 8.2 to the pigeonhole principle: There are width n refutations of PHP^m_n, and the number of variables is $mn \geq n^2$, therefore a direct application of Corollary 8.2 yields a size lower bound that is constant. One way to get around this is to prove the lower bound for an even weaker pigeonhole principle—one in which each pigeon finds only a small number of holes acceptable.

Definition 8.2. *Let $G = (U \cup V, E)$ be a bipartite graph. The* pigeonhole principle of G, $PHP(G)$, *is the set of clauses For each $u \in U$, there is $\bigvee_{\substack{v \in V \\ \{u,v\} \in E}} x_{u,v}$. For each $u, u' \in [m]$, with $u \neq u'$, and each $v \in V$ with $\{u, v\} \in E$ and $\{u', v\} \in E$, there is $\neg x_{u,v} \vee \neg x_{u',v}$. The* maximum degree of G, $\Delta(G)$, *is defined to be* $\max_{v \in V} \deg(v)$.

Notice that $iw(PHP(G))$ is the larger of two and the maximum degree of a left vertex of G.

Definition 8.3. *Let G be a bipartite graph with m left nodes and n right nodes. We say that G is an (m, n, d, r, η)-boundary expander if the adjacency matrix $A \in \{0, 1\}^{m \times n}$ (with $A_{i,j} = 1$ iff i is adjacent to j in G) is an (d, r, η)-boundary expander in the sense of Definition* 7.1.

Lemma 8.7. [38] *Let G be a bipartite graph that is an (m, n, d, r, η)-boundary expander. $w(PHP(G)) \geq \frac{r\eta}{2}$.*

Proof. For each i, let P_i denote the clause $\bigvee_{j \sim_G i} x_{i,j}$. Let H denote the set of CNF $\bigwedge_{i,i',j} \left(\neg x_{i,j} \vee \neg x_{i',j}\right)$. For each clause C in Γ, let $\mu(C) = \min\{|I| \colon H \wedge \bigwedge_{i \in I} P_i \models C\}$. Observe that $\mu \colon \Gamma \to \{0, \dots, m\}$ maps each axiom to 0 or 1. Moreover, $\mu(\emptyset) \geq r$ because G is an (m, n, d, r, η)-expander, and thus Hall's matching condition guarantees that every $I \subseteq [m]$ with $|I| < r$ has a matching into $[n]$. Finally, μ is subadditive with respect to the resolution rule: $\mu(A \vee B) \leq \mu(A \vee x) + \mu(B \vee \neg x)$. This allows us to choose a clause C in Γ with $r/2 \leq \mu(C) < r$.

Choose $I_0 \subseteq [m]$ so that $|I_0| = \mu(C)$ and $H \wedge \bigwedge_{i \in I_0} P_i \models C$. Let $j_0 \in \delta(I_0)$ be given. Suppose for the sake of contradiction that C contains no variable of the form x_{i,j_0} with $i \in [m]$. Choose $i_0 \in I_0$ so that $i_0 \sim_G j_0$, and choose an assignment α satisfying $H \wedge \bigwedge_{i \in I_0 \setminus \{i_0\}} P_i$ and falsifying C. Because C contains no variable of the form x_{i,j_0} and $j_0 \not\sim_G i$ for all $i \in I_0 \setminus \{i_0\}$, we may assume that $\alpha(x_{i,j_0}) = 0$ for all $i \in [m]$.

Define the assignment α' to agree with α off x_{i_0,j_0} and to set $x_{i_0,j}$ to 1. Because C does not contain the variable $x_{i_0,j}$, $\alpha' \not\models C$. However, $\alpha' \models H \wedge \bigwedge_{i \in I_0} P_i$-contradiction. Therefore, for every $j_0 \in \delta(I_0)$ there is some variable x_{i,j_0} present in C, so the width of C is at least $|\delta(I_0)| \geq \frac{\eta r}{2}$. ⊣

Observe that when G has maximum left-degree d, there are dm variables in $PHP(G)$, therefore by Corollary 8.2:

COROLLARY 8.8. [38] *Whenever G is a bipartite (m, n, d, r, η)-expander,* $S(PHP(G)) \geq 2^{\frac{r^2\eta^2}{4dm}}$.

THEOREM 8.9. [38] *For all integers $m > n > 0$, $S(PHP_n^{n+1}) \geq 2^{\Omega(n)}$ and* $S(PHP_n^m) \geq 2^{\frac{n^2}{m \log m}}$.

PROOF. Let $G_{n+1,n}$ be a bipartite $(n+1, n, 5, n/c, 1)$-expander (such an expander exists by a simple probabilistic calculation with c a constant greater than 1, cf. [90]). Let σ be the partial assignment on $Vars(PHP_n^{n+1})$ so that $\sigma(x_{i,j}) = x_{i,j}$, if $(i,j) \in E(G_n)$, and $\sigma(x_{i,j}) = 0$ otherwise. Let $\Gamma = C_1, \ldots, C_m$ be a resolution refutation of PHP_n^{n+1}. Clearly, $\Gamma \restriction_\sigma$ is resolution refutation of $PHP_n^{n+1} \restriction_\sigma = PHP(G_{n+1,n})$. By Corollary 8.8, the size of $\Gamma \restriction_\sigma$ is at least $2^{n/4c^2}$. For $m = \Theta(n)$, we use a similar argument with a $(m, n, \log m, \Omega(\frac{n}{\log m}), \frac{3}{4}\log m)$ expander. ⊣

8.2. Exponential lower bounds for refutations of random k-CNFs. It is possible to prove that random 3-CNFs of constant clause density require resolution refutations of linear width directly using the boundary expansion technique of Lemma 8.7. However, a slight modification gives quantitatively better bounds.

DEFINITION 8.4. [64] *Let F be a set of clauses over the variable set V. The* boundary of F, $\partial(F)$, *is defined as*:

$$\partial(F) = \{v \in V \mid v \text{ appears in exactly one clause of } F\}.$$

Let $s(F)$ be the minimum size of an unsatisfiable subset of F. Define the expansion of F *as* $e(F) = \min\{|\delta(F_0)| : F_0 \subseteq F,\ s(F)/2 \leq |F_0| < s(F)\}$.

THEOREM 8.10. [64] *For any set of clauses F, $w(F) \geq e(F)$.*

PROOF. We define a notion of clause complexity as follows: For any clause C, $\mu(C)$ is equal to the minimum size of $F_0 \subseteq F$ so that $F_0 \models C$.

Let Γ be a resolution refutation of F. Because μ is subadditive with respect to resolution, each $A \in F$ has $\mu(A) = 1$, and, by the definition of $s(F)$, $\mu(\emptyset) \geq s(F)$, there exists a clause C in Γ so that $s(F)/2 \leq \mu(C) < s(F)$. Let $F_0 \subseteq F$ be so that $|F_0| = \mu(C)$ and $F_0 \models C$.

We now show that for each variable in $\partial(F_0)$ also appears in C. Let D be the unique clause in F_0 with $x \in D$. Because $F_0 - D \not\models C$, we may and choose an assignment α so that α satisfies every clause of $F_0 \setminus D$ but not C. Let α^* be α with its value on x flipped. Because $x \in D$, $\alpha^* \models D$, and because x does not appear in any other clause of F_0, $\alpha^* \models F_0$. Since $F_0 \models C$, we also have that $\alpha^* \models C$. Because $\alpha^* \models C$ and $\alpha \not\models C$, we must have that $x \in C$. Because the size of $\partial(F_0)$ is at least $e(F)$, the lemma is proved. ⊣

Plugging the excellent expansion parameters of random k-CNFs into the width inequality of Theorem 8.10, and then applying the size-width trade-off of Corollary 8.2 yields size lower bounds for refutations of random k-CNFs.

LEMMA 8.11 (See [29, 26] for proofs.). *If F is distributed according to $F_k^{\Delta,n}$ then with probability $1 - o(1)$ as $n \to \infty$: $s(F) = \Omega\left(n/\Delta^{1/(k-2)}\right)$ and $e(F) = \Omega\left(n/\Delta^{2/(k-2)}\right)$.*

THEOREM 8.12. *For F distributed as $F_k^{\Delta,n}$, with probability $1 - o(1)$ as $n \to \infty$: Every treelike resolution refutation of F has size at least $2^{n/\Delta^{2/(k-2)+\epsilon}}$, and every resolution refutation of F has size at least $2^{n/\Delta^{4/(k-2)+\epsilon}}$.*

PROOF. Combining Lemma 8.11 and Theorem 8.10, we have that with probability $1 - o(1)$, $w(F) \geq \Omega\left(n/\Delta^{2/(k-2)}\right)$. An application of Corollary 8.2 shows that in this event: $S_T(F) \geq 2^{\Omega\left(\left(n/\Delta^{2/(k-2)}\right)-k\right)}$ and $S(F) \geq 2^{\Omega\left(\left(n/\Delta^{2/(k-2)}-k\right)^2/n\right)} = 2^{\Omega\left(\left(n/(\Delta^{4/(k-2)}-2k\Delta^{2/(k-2)}+k^2)\right)\right)}$. ⊣

§9. **The small restriction switching lemma.** There is no known analog of the size-width trade-off that holds for Res (k) for any $k \geq 2$. However, we can reduce size lower bounds for Res (k) refutations to width bounds for resolution refutations using a technique called the *small restriction switching lemma*. A switching lemma is a guarantee that after the application of a randomly chosen partial assignment, a disjunction of small ANDs can be represented by a conjunction of small ORs, thus "switching" an OR into an AND. This turns the k-DNFs of a Res (k) refutation into narrow clauses, so that the Res (k) refutation becomes a narrow resolution refutation (after some clean-up of the inference steps).

In this section, we prove the small restriction switching lemma and its connection with resolution width, and use these to prove that Res (k) refutation of PHP_n^{2n} require size $2^{n^{\Omega(1)}}$ (this presentation closely follows [149] and [152]). In Section 10, we combine the small restriction switching lemma with expansion clean-up techniques to prove that almost all random 3-CNFs of constant clause density requre size $2^{n^{\Omega(1)}}$ to be refuted in Res (k).

Another variety of switching lemma, the Håstad-style switching lemmas, have been used to establish exponential size lower bounds for constant-depth Frege proofs of PHP_n^{n+1} [34, 130, 106] and the modular counting principles [27, 57, 33, 94]. Such techniques are powerful—they can be iterated to prove proof size lower bounds for constant depth systems- but they seem too crude to analyze refutation sizes for PHP_n^{2n} or for random 3-CNFs. This is because switching lemmas of this form must set an overwhelming majority of the variables to 0 or 1 in order to collapse a k-DNF into a CNF of narrow clauses. Consider the standard formulation for distributions that set bits independently:

THEOREM 9.1 ("Håstad's switching lemma" [88], cf. [45, 25]). *Let positive integers k and w be given. Setting $\phi = (1 + \sqrt{5})/2$ and $\gamma = 2/\ln\phi$ (note that*

$\gamma > 4$), *we have that for any k-DNF F, if we construct an assignment ρ by independently setting each bit to* 0 *with probability $p/2$, to* 1 *with probability $p/2$, and leave it unset with probability* $1 - p$:

$$Pr_\rho[F \restriction_\rho \text{ cannot be computed by a } w\text{-CNF}] \leq (\gamma(1-p)k)^w.$$

To collapse a k-DNF to a w-CNF using Theorem 9.1, it is necessary for the probability of a variable being set (p in the notation of Theorem 9.1) to be strictly more than $1 - \frac{1}{\gamma k} \geq 1 - (1/4k) \geq 3/4$. Futhermore, when k is a superconstant function of n, almost all of the bits must be set. On the other hand, if a partial matching matches a majority of the pigeons in the $2n$ to n pigeonhole principle, the original CNF becomes trivially false. The small restriction switching lemma of Theorem 9.2 can apply to k-DNFs even when the probability of setting a variable is vanishingly small. This enables the small restriction switching lemma to be applied in many contexts when Håstad's switching lemma cannot.

9.1. The small restriction switching lemma.

DEFINITION 9.1. *A* decision tree *is a rooted binary tree in which every internal node is labeled with a variable, the edges leaving a node correspond to whether the variable is set to* 0 *or* 1, *and the leaves are labeled with either* 0 *or* 1. *Every path from the root to a leaf may be viewed as a partial assignment. For a decision tree T and $v \in \{0,1\}$, we write the set of paths (partial assignments) that lead from the root to a leaf labeled v as $Br_v(T)$. We say that a decision tree T* strongly represents *a DNF F if for every $\pi \in Br_0(T)$, for all $t \in F$, $t \restriction_\pi = 0$ and for every $\pi \in Br_1(T)$, there exists $t \in F$, $t \restriction_\pi = 1$. The representation height of F, $h(F)$, is the minimum height of a decision tree strongly representing F.*

Notice that the function computed by a decision tree of height h can also be computed by both an h-CNF and an h-DNF.

The switching lemma exploits a trade-off based on the minimum size of a set of variables that meets each term of a k-DNF.

DEFINITION 9.2. *Let F be a DNF, and let S be a set of variables. If every term of F contains a variable from S, then we say that S* is a cover of F. *The* covering number of F, *$c(F)$, is the minimum cardinality of a cover of F.*

For example, the 3-DNF $xyz \vee \neg x \vee yw$ has covering number two.

We now give a general condition on the distributions of partial assignments for which our switching lemma holds: That the distribution almost always satisfies any k-DNF with a large cover number.

THEOREM 9.2. [152] *Let $k \geq 1$, let $s_0, \dots, s_{k-1}$ and $p_1, \dots, p_k$ be sequences of positive numbers, and let $\mathcal{D}$ be a distribution on partial assignments so that for every $i \leq k$ and every i-DNF G, if $c(G) > s_{i-1}$, then $Pr_{\rho \in \mathcal{D}}\left[G \restriction_\rho \neq 1\right] \leq p_i$.*

Then for every k-DNF F:

$$Pr_{\rho\in\mathcal{D}}\left[h(F\restriction_\rho) > \sum_{i=0}^{k-1} s_i\right] \le \sum_{i=1}^{k} 2^{\left(\sum_{j=i}^{k-1} s_j\right)} p_i.$$

Proof. We proceed by induction on k. First consider $k = 1$. If $c(F) \le s_0$, then at most s_0 variables appear in F. We can construct a height $\le s_0$ decision tree that strongly represents $F\restriction_\rho$ by querying all of the variables of $F\restriction_\rho$. On the other hand, if $c(F) > s_0$ then $\Pr_{\rho\in\mathcal{D}}\left[F\restriction_\rho \neq 1\right] \le p_1$. Therefore, $h(F\restriction_\rho)$ is non-zero with probability at most $p_1 = p_1 2^{\sum_{j=1}^{k-1} s_j}$.

For the induction step, assume that the theorem holds for all k-DNFs, let F be a $(k+1)$-DNF, and let $s_0,\dots,s_k$ and $p_1,\dots,p_{k+1}$ be sequences of positive numbers satisfying the hypotheses of the theorem. If $c(F) > s_k$, then by the conditions of the lemma, $\Pr_{\rho\in\mathcal{D}}\left[F\restriction_\rho \neq 1\right] \le p_{k+1}$. Because $p_{k+1} \le \sum_{i=1}^{k+1} 2^{\sum_{j=i}^{k} s_j} p_i$, we have that $h(F\restriction_\rho)$ is non-zero with probability at most $\sum_{i=1}^{k+1} 2^{\sum_{j=i}^{k} s_j} p_i$.

Consider the case when $c(F) \le s_k$. Let S be a cover of F of size at most s_k. Let π be any assignment to the variables in S. Because each term of F contains at least one variable from S, $F\restriction_\pi$ is a k-DNF. By combining the induction hypothesis with the union bound, we have that

$$\begin{aligned}\Pr_{\rho\in\mathcal{D}}\left[\exists\pi\in\{0,1\}^S\ h((F\restriction_\rho)\restriction_\pi) > \sum_{i=0}^{k-1} s_i\right] &\le 2^{s_k}\left(\sum_{i=1}^{k} 2^{\left(\sum_{j=i}^{k-1} s_j\right)} p_i\right)\\ &< \sum_{i=1}^{k+1} 2^{\left(\sum_{j=i}^{k} s_j\right)} p_i.\end{aligned}$$

In the event that $\forall\pi\in\{0,1\}^S$, $h((F\restriction_\rho)\restriction_\pi) \le \sum_{i=0}^{k-1} s_i$, we construct a decision tree for $F\restriction_\rho$ as follows. First, query all variables in S unset by ρ, and then underneath each branch, β, simulate a decision tree of minimum height strongly representing $(F\restriction_\rho)\restriction_\beta$. For each such β, let $\hat\beta$ be the part of the assignment $\rho\cup\beta$ restricted to the variables of S, and note that $\hat\beta$ is a total assignment to the variables of S with $(F\restriction_\rho)\restriction_\beta = (F\restriction_\rho)\restriction_{\hat\beta}$. Therefore the height of the resulting decision tree is at most $s_k + \max_{\pi\in\{0,1\}^S} h((F\restriction_\rho)\restriction_\pi) \le \sum_{i=0}^{k} s_i$.

Now we show that the decision tree constructed above strongly represents $F\restriction_\rho$. Let π be a branch of the tree. Notice that $\pi = \beta\cup\sigma$, where β is an assignment to the variables in $S\setminus\mathrm{dom}(\rho)$ and σ is a branch of a tree that strongly represents $(F\restriction_\rho)\restriction_\beta$. Consider the case that π leads to a leaf labeled 1. In this case, σ satisfies a term t' of $(F\restriction_\rho)\restriction_\beta$. We may choose a term t of F so that $t' = (t\restriction_{\rho\cup\beta})$, and $\pi = \beta\cup\sigma$ satisfies the term $t\restriction_\rho$ of $F\restriction_\rho$. Now consider the case that π leads to a leaf labeled 0. There are two cases, $(F\restriction_\rho)\restriction_\beta = 0$ and $(F\restriction_\rho)\restriction_\beta \neq 0$. If $(F\restriction_\rho)\restriction_\beta = 0$, then for every term t of $F\restriction_\rho$, t is inconsistent with β and hence with π. If $(F\restriction_\rho)\restriction_\beta \neq 0$ then because the sub-tree underneath β strongly represents $(F\restriction_\rho)\restriction_\beta$, for every term t of $(F\restriction_\rho)\restriction_\beta$, t is inconsistent with σ. Therefore, every term of $F\restriction_\rho$ is inconsistent with either β or σ, and thus with $\pi = \beta \cup \sigma$. ⊣

COROLLARY 9.3. *Let $k \geq 1$, $d > 0, 1 \geq \delta > 0, 1 \geq \gamma > 0, s$, and let $\mathcal{D}$ be a distribution on partial assignments so that for every k-DNF G, $Pr_{\rho\in\mathcal{D}}\left[G\restriction_\rho \neq 1\right] \leq d2^{-\delta(c(G))^\gamma}$. Then for every k-DNF F, $Pr_{\rho\in\mathcal{D}}\left[h(F\restriction_\rho) > 2s\right] \leq dk2^{-\delta' s^{\gamma'}}$, where $\delta' = 2(\delta/4)^k$ and $\gamma' = \gamma^k$.*

PROOF. Let $s_i = (\delta/4)^i(s^{\gamma^i})$, and $p_i = d2^{-4s_i}$. Note that $s_{i-1}/4 \geq (\delta/4)s_{i-1} = (\delta/4)(\delta/4)^{i-1}s^{\gamma^{i-1}} \geq (\delta/4)^i s^{\gamma^i} = s_i$. It follows that $\sum_{j=i}^k s_j \leq \sum_{j\geq i} s_i/4^{j-i} \leq 2s_i$. Also, for any i-DNF G, with $c(G) \geq s_{i-1}$, $\Pr_{\rho\in\mathcal{D}}\left[G\restriction_\rho \neq 1\right] \leq d2^{-\delta(c(G))^\gamma} \leq d2^{-\delta s_{i-1}^\gamma} = 2^{-\delta(\delta/4)^{i-1}(s^{\gamma^{i-1}})^\gamma} = d2^{-4s_i}$. Thus, we can apply theorem 9.2 with parameters $p_1,\dots,p_k,s_0,\dots,s_{k-1}$. For every k-DNF F:

$$\begin{aligned}
\Pr_{\rho\in\mathcal{D}}\left[h(F\restriction_\rho) > 2s\right] &\leq \Pr_{\rho\in\mathcal{D}}\left[h(F\restriction_\rho) > \sum_{i=0}^{k-1} s_i\right]\\
&\leq \sum_{i=1}^{k} 2^{\left(\sum_{j=i}^{k-1} s_j\right)} p_i\\
&\leq \sum_{i=1}^{k} 2^{2s_i}(d2^{-4s_i})\\
&\leq dk2^{-2s_k} = dk2^{-\delta' s^{\gamma'}}.
\end{aligned}$$

⊣

9.2. Converting Res (k) refutations into resolution refutations. Applications of the small-restriction switching lemma use the fact that when the lines of a Res (k) refutation are strongly represented by short decision trees, the refutation can be converted into a narrow resolution refutation. This does not depend the particular, definition of the $Res(k)$ system, but only upon a property called *strong soundness*: If F is inferred from $F_1,\dots,F_j$, and $t_1,\dots,t_j$ are mutually consistent terms of $F_1,\dots,F_j$ respectively, then there is a term t of F implied by $\bigwedge_{i=1}^j t_i$. In other words, any reason why $F_1,\dots,F_k$ are true implies a reason why F is true. This is stronger than mere soundness.

Recall the definition of $w(\mathcal{C})$ from Definition 8.1.

THEOREM 9.4. *Let $\mathcal{C}$ be a set of clauses of width $\leq h$. If $\mathcal{C}$ has a Res (k) refutation so that for each line F of the refutation, $h(F) \leq h$, then $w(\mathcal{C}) \leq kh$.*

Proof. We will use the short decision trees to construct a narrow refutation of $\mathcal{C}$ in resolution augmented with subsumption inferences: Whenever $A \subseteq B$, $\frac{A}{B}$. These new inferences simplify our proof, but they may be removed from the resolution refutation without increasing the size or the width.

For each initial clause $C \in \mathcal{C}$, we let T_C be the decision tree that queries the (at most h) variables in C, stopping with a 1 if the clause becomes satisfied and stopping with a 0 if the clause becomes falsified. For the other lines, F, let T_F be a shortest decision tree that strongly represents F.

For any partial assignment π let C_π be the clause of width $\leq h$ that contains the negation of every literal in π, i.e., the clause that says that branch π was not taken.

We construct a resolution proof of width $\leq kh$ by deriving C_π for each line F of the refutation and each $\pi \in \mathrm{Br}_0(T_F)$.

Notice that for $\pi \in \mathrm{Br}_0(T_\emptyset)$, $C_\pi = \emptyset$, and for each $C \in \mathcal{C}$, for the unique $\pi \in \mathrm{Br}_0(T_C)$, $C_\pi = C$.

Let F be a line of the refutation that is inferred from the previously derived formulas $F_1, \dots, F_j$, $j \leq k$. Assume we have derived all $C_\pi \in \mathrm{Br}_0(T_{F_i})$ for $1 \leq i \leq j$.

To guide the derivation of $\{C_\pi \mid \pi \in \mathrm{Br}_0(T_F)\}$, we construct a decision tree that represents the the conjunction of $F_1, \dots, F_j$. The tree (call it T) begins by simulating, T_{F_1} and outputting 0 on any 0-branch of T_{F_1}. On any 1-branch, it then simulates T_{F_2}, etc. If all j branches are 1, T outputs 1; otherwise T outputs 0. The height of T is at most $jh \leq kh$, so the width of any such C_π, with $\pi \in \mathrm{Br}(T)$ is at most kh.

Every $\sigma \in \mathrm{Br}_0(T)$ contains some $\pi \in \bigcup_{i=1}^{j} \mathrm{Br}_0(T_{F_i})$. Therefore, $\{C_\sigma \mid \sigma \in \mathrm{Br}_0(T)\}$ can be derived from the previously derived clauses by subsumption inferences.

On the other hand, if $\sigma \in \mathrm{Br}_1(T)$, there exists $\pi_1 \in \mathrm{Br}_1(T_{F_1}), \dots, \pi_j \in \mathrm{Br}_1(T_{F_j})$ so that $\pi_1 \cup \dots \cup \pi_j = \sigma$. Because the decision trees $T_{F_1}, \dots, T_{F_j}$ strongly represent the k-DNFs $F_1, \dots, F_j$, there exist terms $t_1 \in F_1, \dots, t_j \in F_j$ so that $\bigwedge_{i=1}^{j} t_i$ is satisfied by σ. By strong soundness of $\mathrm{Res}(k)$, there exists $t \in F$ so that σ satisfies t.

Let $\sigma \in \mathrm{Br}_0(T_F)$ be given. Because T_F strongly represents F, σ sets all terms of F to 0. So by the preceding paragraph, for all $\pi \in \mathrm{Br}(T)$, if π is consistent with σ, then $\pi \in \mathrm{Br}_0(T)$.

We now begin the derivation of $\mathrm{Br}_0(T_F)$. Let $\sigma \in \mathrm{Br}_0(T_F)$ be given. For each node v in T, let π_v be the path (viewed as a partial assignment) from the root to v. Bottom-up from leaves to root, we inductively derive $C_{\pi_v} \vee C_\sigma$, for each v so that π_v is consistent with σ. When we reach the root, we will have derived C_σ.

If v is a leaf, then $\pi_v \in \mathrm{Br}_0(T)$ (because it is consistent with σ), and it has already been derived.

If v is labeled with a variable that appears in σ, call it x, then there is a child u of v with $\pi_u = \pi_v \cup \{x\}$. Therefore, $C_{\pi_v} \vee C_\sigma = C_{\pi_u} \vee C_\sigma$. By induction, the clause $C_{\pi_u} \vee C_\sigma$ has already been derived.

If v is labeled with a variable x that does not appear in σ, then for both of the children of v, call them v_1, v_2, the paths π_{v_1} and π_{v_2} are consistent with σ. Moreover, $C_{\pi_{v_1}} \vee C_\sigma = x \vee C_{\pi_v} \vee C_\sigma$ and $C_{\pi_{v_2}} \vee C_\sigma = \neg x \vee C_{\pi_v} \vee C_\sigma$. Resolving these two previously derived clauses gives us $C_{\pi_v} \vee C_\sigma$. $\dashv$

COROLLARY 9.5. *Let $\mathcal{C}$ be a set of clauses of width $\leq h$, let Γ be a Res (k) refutation of $\mathcal{C}$, and let ρ be a partial assignment so that for every line F of Γ, $h(F \restriction_\rho) \leq h$. Then $w(\mathcal{C} \restriction_\rho) \leq kh$.*

9.3. Lower bounds the $2n$ to n weak pigeonhole principle. Here we prove:

THEOREM 9.6. *For every $c > 1$, there exists $\epsilon > 0$ so that for all n sufficiently large, if $k \leq \sqrt{\log n / \log \log n}$, then every Res$(k)$ refutation of PHP_n^{cn} has size at least 2^{n^ϵ}.*

We contrast this with the known upper bounds for PHP_n^{2n}: Maciel, Pitassi and Woods [110] demonstrate quasipolynomial size refutations of PHP_n^{2n} in Res(polylog(n)). Our results show that super-constant sized conjunctions are necessary for sub-exponential size proofs of the weak pigeonhole principle.

Alexander Razborov has announced an improvement of Theorem 9.6:

THEOREM 9.7. [145] *For every $c > 1$, there exists $\epsilon, \delta > 0$ so that for all n sufficiently large, if $k \leq \epsilon \log n / \log \log n$, then every Res$(k)$ refutation of PHP_n^{cn} has size at least 2^{n^δ}.*

His proof uses a switching lemma that is less general (in particular, it does not clearly apply to random 3-CNFs as we need in Section 10). For this reason we present the version based upon the more general switching lemma.

As in Subsection 8, we perform the analysis on $PHP(G)$ where G is a suitable bipartite graph. (See Definition 8.2 for the definition of $PHP(G)$.)

First, all Res (k) refutations are put into a normal form in which no term of any DNF asks that two pigeons be mapped to the same hole.

DEFINITION 9.3. [20] *Let $G = (U \cup V, E)$ be a bipartite graph. A term is said to be in* pigeon-normal-form *if it does not contain two literals $x_{u,v}$ and $x_{u',v}$ with $u \neq u'$. A DNF is said to be in pigeon-normal-form if all of its terms are in pigeon-normal-form and a Res (k) refutation is said to be in pigeon normal form if every line is in pigeon-normal-form.*

Every Res (k) refutation of $PHP(G)$ can be transformed into a refutation in pigeon normal form which at must doubles the number of lines in the proof. When there is an AND-introduction inference that creates a line not in pigeon normal form, say

$$\frac{(A \vee x_{u,v})\ (A \vee x_{u',v}) \dots (A \vee l_j)}{A \vee \left(x_{u,v} \wedge x_{u',v} \wedge \bigwedge_{i=3}^{j} l_i\right)}.$$

Replace the inference by a derivation that resolves $A \vee x_{u',v}$ with $\neg x_{u,v} \vee \neg x_{u',v}$ to obtain $A \vee \neg x_{u,v}$. Resolve this with $A \vee x_{u,v}$ to obtain A. We may proceed through the rest of the proof with A because it subsumes $A \vee x_{u,v} \wedge x_{u',v} \wedge \bigwedge_{i=3}^{j} l_i$.

Now we define our family of random restrictions.

Definition 9.4. *For a bipartite graph $G = (U \cup V, E)$ and a real number $p \in [0, 1]$, let $\mathcal{M}_p(G)$ denote the distribution on partial assignments ρ given by the following experiment*:

Independently, for each $v \in V$, with probability $1 - p$ choose to match v and with probability p leave v unmatched. If v is matched, uniformly select a neighbor u of v, set $x_{u,v}$ to 1, and for every $w \neq u$ that is a neighbor of v, set $x_{w,v}$ to 0. Moreover, for each $v' \neq v$, set $x_{u,v'} = 0$.

Let V_ρ be the set of vertices of V matched by ρ, let U_ρ be the set of vertices of U matched by ρ, and let $S_\rho = U_\rho \cup V_\rho$.

It is easy to check that for any $\rho \in \mathcal{M}_p(G)$, we have that $PHP(G) \restriction_\rho = PHP(G - S_\rho)$.

Lemma 9.8. *Let $p \in [0, 1]$, $i \in [k]$ be given. Let $G = (U \cup V, E)$ be a bipartite graph with $\Delta = \Delta(G)$. Let F be an i-DNF in pigeon-normal-form*: $Pr_{\rho \in \mathcal{M}_p(G)} \left[F \restriction_\rho \neq 1\right] \leq 2^{-\frac{(\log e)(1-p)^i c(F)}{i\Delta^{i+1}}}$.

Proof. For a term T, define the *holes of* T as $\mathrm{Holes}(T) = \{v \mid x_{u,v} \in T \text{ or } \neg x_{u,v} \in T\}$. We say that two terms T and T' are *hole-disjoint* if $\mathrm{Holes}(T) \cap \mathrm{Holes}(T') = \emptyset$.

Because F contains at least $c(F)/i$ many variable-disjoint terms, and each hole $v \in V$ appears in at most Δ many variables, F must contain at least $c(F)/i\Delta$ many hole-disjoint terms.

The events of satisfying hole-disjoint terms are independent, and for a given term, T, the probability that $T \restriction_\rho = 1$ is at least $(1 - p)^i/\Delta^i$. This is because with probability $(1 - p)^i$, every hole of T is matched, and with probability at least $1/\Delta^i$ the holes are matched in a way that satisfies T (here we use that F is in pigeon-normal-form). Therefore, we have that:

$$\begin{aligned} \Pr_\rho \left[F \restriction_\rho \neq 1\right] &\leq \left(1 - (1 - p)^i/\Delta^i\right)^{\frac{c(F)}{i\Delta}} \\ &\leq \left(e^{-(1-p)^i/\Delta^i}\right)^{\frac{c(F)}{i\Delta}} = 2^{-\frac{(\log e)(1-p)^i c(F)}{i\Delta^{i+1}}}. \end{aligned}$$ ⊣

For the proof to work, we need that after the application of a random restriction ρ, with high probability, $G - S_\rho$ contains a good boundary expander as a subgraph (and therefore $PHP(G) \restriction_\rho$ requires large width to refute). We call such graphs *robust*.

DEFINITION 9.5. *A bipartite graph G with m left vertices, n right vertices, and maximum right degree d is said to be (p, r, η)-*robust, *if when ρ is selected from $\mathcal{M}_p(G)$, with probability at least $\frac{1}{2}$, $G - S_\rho$ contains an $(m - (1-p)n, pn, d, r, \eta)$-boundary expander as a subgraph.*

All we need for the size lower bound is the following lemma, which is proven in [152]. The proof is a straightforward probabilistic construction: A random subgraph of a random graph is itself a random graph, random graphs are good expanders.

LEMMA 9.9. [152] *For all $c > 1$, there exists $d, c_1, c_2 > 0$ so that for n sufficiently large, there exists a bipartite graph G on vertex sets $[cn]$ and $[n]$ that is $(3/4, c_1(n/\ln n), c_2 \ln n)$-robust and has $\Delta(G) \leq d \log n$.*

LEMMA 9.10. *For any $c > 1$ and $d, c_1, c_2 > 0$, there exists $\epsilon > 0$ so that for all n sufficiently large, if $k \leq \sqrt{\log n / \log\log n}$ and G is a $(3/4, c_1(n/\ln n), c_2 \ln n)$-robust bipartite graph with vertex sets of sizes cn and n and $\Delta(G) \leq d \log n$, then $S_k(PHP(G)) \geq 2^{n^\epsilon}$.*

PROOF. By Lemma 9.8, for each $i \in [k]$ and every i-DNF F,

$$\Pr_{\rho \in \mathcal{M}_{3/4}(G)}\left[F \restriction_\rho \neq 1\right] \leq 2^{-\frac{(\log e)(1-3/4)^i c(F)}{i(d \log n)^{i+1}}} = 2^{-\frac{(\log e) c(F)}{i \cdot 4^i (d \log n)^{i+1}}}.$$

In the interest of obtaining a better bound, we will not appeal to Corollary 9.3, but directly apply Theorem 9.2. We define sequences $s_0, \dots, s_k$ and $p_1, \dots, p_k$ for use in the switching lemma. Set $s_0 = \frac{3}{4k}(c_1 c_2 n/2 - 1)$. For each $i \in [k]$, set $s_i = s_{i-1} \cdot \left(\frac{\log e}{2i4^i (d \log n)^{i+1}}\right)$. For each $i \in [k]$ set $p_i = 2^{-2s_i}$. For every i-DNF F so that $c(F) > s_{i-1}$, we have the following inequality:

$$\Pr_{\rho \in \mathcal{M}_{3/4}(G)}\left[F \restriction_\rho \neq 1\right] < 2^{-\frac{(\log e) s_{i-1}}{i \cdot 4^i (d \log n)^{i+1}}} = 2^{-2\frac{(\log e) s_{i-1}}{2i4^i (d \log n)^{i+1}}} = 2^{-2s_i} = p_i.$$

An easy calculation (presented below in Lemma 9.12) shows that there exists $\epsilon > 0$ so that for sufficiently large n, $s_k \geq n^\epsilon$. Suppose that Γ is a $\mathrm{Res}(k)$ refutation of $PHP(G)$ of size less than 2^{n^ϵ}. By an application of Theorem 9.2 and the union bound, we have:

$$\begin{aligned}\Pr_{\rho \in \mathcal{M}_{3/4}(G)}\left[\exists F \in \Gamma,\ h(F \restriction_\rho) > \sum_{i=0}^{k-1} s_i\right] &\leq 2^{n^\epsilon} \sum_{i=1}^{k} p_i 2^{\sum_{j=i}^{k-1} s_j} \\ &\leq 2^{s_k} \sum_{i=1}^{k} p_i 2^{\sum_{j=i}^{k-1} s_j} = \sum_{i=1}^{k} p_i 2^{\sum_{j=i}^{k} s_j}.\end{aligned}$$

We now bound $p_i 2^{\sum_{j=i}^{k} s_j}$ for each $i > 0$. For each i, $s_{i+1} < \frac{1}{4} s_i$ so $\sum_{j=i}^{k-1} s_j \leq \frac{4}{3} s_i$. This gives us the following inequality:

$$p_i 2^{\sum_{j=i}^{k-1} s_j} = 2^{\sum_{j=i}^{k-1} s_j - 2s_i} \leq 2^{(4/3-2)s_i} = 2^{-(2/3)s_i} \leq 2^{-(2/3)s_k} \leq 2^{-(2/3)n^{\epsilon}}.$$

Therefore:

$$\begin{aligned}
&\Pr_{\rho \in \mathcal{M}_{3/4}(G)} \left[\exists F \in \Gamma,\ h(F \restriction_\rho) > (c_1 c_2 n/2 - 1)/k\right] \\
&\qquad \leq \Pr_{\rho \in \mathcal{M}_{3/4}(G)} \left[\exists F \in \Gamma,\ h(F \restriction_\rho) > \sum_{i=0}^{k-1} s_i\right] \\
&\qquad \leq \sum_{i=1}^{k} p_i 2^{\sum_{j=i}^{k-1} s_j} \leq \sum_{i=1}^{k} 2^{-(2/3)n^{\epsilon}} \leq k 2^{-(2/3)n^{\epsilon}} = 2^{\log k - (2/3)n^{\epsilon}}.
\end{aligned}$$

For n sufficiently large, this probability is strictly less than $1/2$. Because G is a $(3/4, c_1(n/\ln n), c_2 \ln n)$-robust for $\rho \in \mathcal{M}_{3/4}(G)$, with probability at least $1/2$, $G - S_\rho$ contains a $((c - 1/4)n, (3/4)n, d, c_1(n/\ln n), c_2 \ln n)$-boundary expander. Let β be the assignment that zeroes out the edges not in the expanding subgraph, and by Lemma 8.7, $w(PHP(G) \restriction_\rho) \geq w(PHP(G) \restriction_{\rho\cup\beta}) \geq \frac{c_1(n/\ln n)c_2 \ln n}{2} = \frac{c_1 c_2 n}{2}$. However, $\forall F \in \Gamma$, $h(F \restriction_\rho) \leq \frac{1}{k}(c_1 c_2 n/2 - 1)$, so by Corollary 9.5, there is a resolution refutation of $PHP(G) \restriction_\rho$ of width $\leq c_1 c_2 n/2 - 1$. Contradiction. ⊣

Theorem 9.11. [152] *For each $c > 1$, there exists $\epsilon > 0$ so that for all n sufficiently large, if $k \leq \sqrt{\log n/\log\log n}$, then every $Res(k)$ refutation of PHP_n^{cn} has size at least $2^{n^{\epsilon}}$.*

Proof. Apply Lemma 9.9 and choose d so that for sufficiently large n, there exists a $(3/4, c_1(n/\ln n), c_2 \ln n)$-robust graph G on vertex sets cn and n, with $\Delta(G) \leq d \log n$. By Lemma 9.10, there exists $\epsilon > 0$ so that for $k \leq \sqrt{\log n/\log\log n}$, $S_k(PHP(G)) \geq 2^{n^{\epsilon}}$. Because $PHP(G)$ can be obtained by setting some of the variables of PHP_n^{cn} to 0, every Res (k) refutation of PHP_n^{cn} can be converted into a Res (k) refutation of $PHP(G)$ of the same or lesser size. Therefore, all Res (k) refutations of PHP_n^{cn} must have size at least $2^{n^{\epsilon}}$. ⊣

Lemma 9.12. *There exists $\epsilon > 0$, so that all n sufficiently large, with $k \leq \sqrt{\log n/\log\log n}$ and $s_0, \dots, s_k$ defined as in the proof of Lemma 9.10, $s_k \geq n^{\epsilon}$.*

Proof. The recursive definition of the s_i's gives:

$$s_k = \frac{1}{2^k}(\log e)^k \frac{1}{k!}\left(\frac{1}{4}\right)^{\sum_{j=1}^{k} j} \left(\frac{1}{d \log n}\right)^{\sum_{j=2}^{k+1} j} \frac{3}{4k}(n/24 - 1).$$

Because $k \le \sqrt{\log n / \log \log n}$, we have that $\frac{1}{2^k}(\log e)^k \frac{1}{k!}\left(\frac{1}{4}\right)^{\sum_{j=1}^{k} j} \frac{3}{4k} = n^{-o(1)}$. Therefore:

$$\begin{aligned} s_k &= n^{-o(1)}(1/d \log n)^{(k+2)(k+1)/2}(n/24 - 1) \\ &= n^{-o(1)} 2^{-(\log(d \log n))(k^2+3k+2)/2}(n/24 - 1). \end{aligned}$$

Because $k \le \sqrt{\log n / \log \log n}$ and d is a constant, for n sufficiently large, $(\log(d \log n))(k^2 + 3k + 2)/2 = (\log n)(1 + o(1))/2$. Therefore,

$$s_k = n^{-o(1)} 2^{-(\log n)(1+o(1))/2}(n/24 - 1)$$

and there exists $\epsilon > 0$ so that for all n sufficiently large, $s_k \ge n^{\epsilon}$. $\dashv$

§10. Expansion clean-up and random 3-CNFs. In this section we study the sizes of refutations needed to refute random 3-CNFs (as given by the distribution $F_3^{\Delta,n}$ described in Subsection 4.2). In particular, we give the proof (due to Misha Alekhnovich) that that random 3-CNFs of constant clause density almost surely require exponentially large Res (k) refutations, for $k \le \sqrt{\log n / \log \log n}$. The Res (k) systems are among the most powerful propostional proof systems for which non-trivial lower bounds are known for the refutation of random 3-CNFs.

THEOREM 10.1. [8] *Let Δ be a constant. For n sufficiently large with respect to Δ, with probability $1 - o(1)$ over 3-CNFs F chosen according to $F_3^{\Delta,n}$, every Res $\left(\sqrt{n/\log \log n}\right)$ refutation of F has size at least $2^{n^{1-o(1)}}$.*

The proof of Theoerem 10.1 uses the the small restriction switching lemma (Theorem 9.2), but with a twist. As in other applications of Theorem 9.2, a random restriction is used to transform a small Res (k) refutation into a narrow resolution refutation. In order to get a contradiction, it is shown that the surviving system of equations is still expanding and therefore requires high-width to refute. This is ensured via an *expansion clean-up procedure* that is applied after the random restriction. Expansion clean-up techniques have proved useful for other bounds in proof complexity and the zero-one optimization [13, 11, 9].

As in [8], we prove the stronger result that systems of linear equations over GF_2, $Ax = b$, require exponentially large Res (k) refutations when A is a $(\Delta n, n, \Theta(1), \Theta(n), \Theta(1))$ boundary expander. This simplifies the analysis of the random restrictions, cf. Lemma 10.11.

10.1. From 3-CNFs to systems of linear equations.

DEFINITION 10.1. *Let F be a 3-CNF in variables $x_1, \dots, x_n$. The system $A^F x = b^F$ over GF_2 is defined as follows: Translate each clause $x_{j_1}^{\epsilon_1} \vee x_{j_2}^{\epsilon_2} \vee x_{j_3}^{\epsilon_3}$ into the equation $x_{j_1} + x_{j_2} + x_{j_3} = \epsilon_1 + \epsilon_2 + \epsilon_3$ over GF_2.*

For a system of equations over GF_2, $Ax = b$, we create an equivalent CNF, $\mathcal{C}_{A,b}$, as follows: Each equation $x_i + x_j + x_k = b$ is encoded as four clauses *of width* 3: *Let* $B = \{(\epsilon_1, \epsilon_2, \epsilon_3) \in GF_2^3 \mid \epsilon_1 + \epsilon_2 + \epsilon_3 \neq b\}$, *and identify* $x_i + x_j + x_k = b$ *with* $\bigwedge_{\vec{\epsilon}\in B}(x_i^{1-\epsilon_1} \vee x_j^{1-\epsilon_2} \vee x_k^{1-\epsilon_3})$. *Let* $\mathcal{C}_{A,b}$ *denote the set of clauses obtained by applying this transformation to all equations of* $Ax = b$.

We state some easy observations without proof:

LEMMA 10.2. *Let* F *be a* 3*-CNF in variables* $x_1, \ldots, x_n$. *If the system* $A^F x = b^F$ *is satisfied, then* F *is also satisfied, but not necessarily vice-versa. For every system of equations* $Ax = b$, *the CNF* $\mathcal{C}_{A,b}$ *is satisfied if and only if the system of equations* $Ax = b$ *is satisfied. For any* 3*-CNF* F, $F \subseteq \mathcal{C}_{A^F,b^F}$. *If there is a size* S $Res(k)$ *refutation of* F, *then there is a size* S $Res(k)$ *refutation of* $\mathcal{C}_{A^F,b^F}$.

10.2. Expansion and expansion clean-up.

LEMMA 10.3. *Let* $Ax = b$ *be a system of equations so that* A *is an* (r, η)*-boundary expander with* $\eta > 0$. *For every* $I \subseteq [m]$ *with* $|I| \leq r$, $A_I x = b_I$ *is satisfiable.*

PROOF. Otherwise, by linear algebra, there is $I' \subseteq I$ with $\sum_{i\in I'} A_i x - b_i = 1$. Notice that $I' \neq \emptyset$ and $\partial_A(I') = \emptyset$. However, by the expansion of A, $|\partial_A I'| > \eta|I'| > 0$; contradiction. ⊣

DEFINITION 10.2. *Let* $A \in \{0, 1\}^{m\times n}$ *be an* (r, η)*-boundary expander, and let* $J \subseteq [n]$ *be given. Define the relation* $\vdash^e_J$ *on subsets of* $[m]$ *as*:

$$I_1 \vdash^e_J I_2 \iff |I_2| \leq (r/2) \wedge \left|\partial_A(I_2) \setminus \left(\bigcup_{i\in I} A_i \cup J\right)\right| < (\eta/2)|I_2|. \tag{1}$$

Define the expansion closure of J, $ecl_A(J)$, *via the following iterative procedure*: *Initially let* $I = \emptyset$. *So long as there exists* I_1 *so that* $I \vdash^e_J I_1$, *let* I_1 *be the lexicographically first such set, replace* I *by* $I \cup I_1$ *and remove all rows in* I_1 *from the matrix* A. *Set* $ecl_A(J)$ *to be the value of* I *after this process stops. The matrix* A *is often clear from the context, and we accordingly drop the subscript. Let* the clean up of A after removing J, $\mathrm{CL}_J(A)$, *be the matrix that results by removing all rows of* $ecl(J)$ *and all columns of* $\bigcup_{i\in ecl_A(J)} A_i$ *from* A.

LEMMA 10.4. *Let* $A \in \{0, 1\}^{m\times n}$ *and* $J \subseteq [n]$ *be given. If* $CL_J(A)$ *is non-empty, then* $CL_J(A)$ *is an* $(r/2, \eta/2)$*-boundary expander.*

PROOF. Suppose that I_1 is a set of $\leq r/2$ many rows of $\mathrm{CL}_J(A)$ such that $|\partial_{\mathrm{CL}_J(A)}(I_1)| < (\eta/2)|I_1|$. Consider a column $j \in \partial_A(I_1)$. There is exactly one $i \in I_1$ with $A_{i,j} = 1$, so clearly there is at most one $i \in I_1$ with $(\mathrm{CL}_J(A))_{i,j} = 1$. Moreover, if $j \notin J \cup \bigcup_{i\in ecl(J)} A_i$, then j is incident with exactly one row $i \in I_1$ in $\mathrm{CL}_J(A)$, so $j \in \partial_{\mathrm{CL}_J(A)}(I_1)$. Therefore: $\partial_A(I_1) \subseteq \partial_{\mathrm{CL}_J(A)}(I_1) \cup \bigcup_{i\in ecl(J)} A_i \cup J$. Therefore:

$$|\partial_A(I_1) \setminus \bigcup_{i\in ecl(J)} A_i \cup J| \leq |\partial_{\mathrm{CL}_J(A)(I_1)} \setminus \bigcup_{i\in ecl(J)} A_i \cup J| < (\eta/2)|I_1|.$$

So $ecl(J) \vdash_J^e I_1$, contrary to the definition of $ecl(J)$. $\dashv$

LEMMA 10.5. *Let* $A \in \{0,1\}^{m\times n}$ *be an* (r,η)*-boundary expander, and let* $J \subseteq [n]$ *be given. If* $|J| < \eta r/4$ *then* $|\,ecl_A(J)| < (2/\eta)|J|$.

PROOF. Suppose for the sake of contradiction that $|\,ecl(J)| \geq (2/\eta)|J|$. Let $I_1,\dots,I_t$ be the sequence of subsets of $[m]$ that are taken in cleaning procedure, with each $|I_i| \leq r/2$.

First we inductively show that for each $s \leq t$, $|\partial_A\left(\bigcup_{i=1}^s I_i\right) \setminus J| \leq (\eta/2)|\bigcup_{i=1}^s I_i|$. For the base case, Equation 1 yields $|\partial_A(I_1)\setminus J| \leq (\eta/2)|I_1|$. For the induction step, assume that $|\partial_A\left(\bigcup_{i=1}^s I_i\right) \setminus J| \leq (\eta/2)|\bigcup_{i=1}^s I_i|$ for an arbitrary $s < t$. By Equation 1, $|\partial_A(I_{s+1}) \setminus \left(J \cup \bigcup_{i\in\bigcup_{i=1}^s I_i} A_i\right)| \leq (\eta/2)|I_{s+1}|$. Because rows added to $ecl(J)$ are removed from the matrix after each stage of cleaning, the sets $I_1,\dots,I_t$ are pairwise disjoint, thus:

$$\begin{aligned}\Big|\partial_A\Big(\bigcup_{i=1}^{s+1} I_i\Big) \setminus J\Big| &\leq \Big|\partial_A\Big(\bigcup_{i=1}^{s} I_i\Big) \setminus J\Big| + \Big|\partial_A(I_{s+1}) \setminus \Big(J \cup \bigcup_{i\in\bigcup_{i=1}^s I_i} A_i\Big)\Big| \\ &\leq (\eta/2)\Big|\bigcup_{i=1}^{s} I_i\Big| + (\eta/2)\Big|I_{s+1}\Big| = (\eta/2)\Big|\bigcup_{i=1}^{s+1} I_i\Big|\end{aligned}$$

Now, let i_0 be the first index with $|\bigcup_{i=1}^{i_0} I_i| > (2/\eta)|J|$. Note that $|\bigcup_{i=1}^{i_0} I_i| \leq |\bigcup_{i=1}^{i_0-1} I_i| + |I_{i_0}| \leq (2/\eta)|J| + r/2 \leq (2/\eta)(\eta r/4) + r/2 = r$. Therefore by expansion, $|\partial_A\left(\bigcup_{i=1}^{i_0} I_i\right)| > \eta|\bigcup_{i=1}^{i_0} I_i|$. Therefore:

$$\begin{aligned}\Big|\partial_A\Big(\bigcup_{i=1}^{i_0} I_i\Big) \setminus J\Big| &\geq \eta\Big|\bigcup_{i=1}^{i_0} I_i\Big| - |J| > \eta\Big|\bigcup_{i=1}^{i_0} I_i\Big| - (\eta/2)\Big|\bigcup_{i=1}^{i_0} I_i\Big| \\ &= (\eta/2)\Big|\bigcup_{i=1}^{i_0} I_i\Big|.\end{aligned}$$

This contradicts the previously established fact that $|\partial_A\left(\bigcup_{i=1}^{i_0} I_i\right) \setminus J| \leq (\eta/2)|\bigcup_{i=1}^{i_0} I_i|$. $\dashv$

LEMMA 10.6. *Let* $A \in \{0,1\}^{m\times n}$ *be an* (r,η)*-boundary expander, and let* $J \subseteq [n]$ *be given. For all* $I_0 \subseteq [m]$*, if* $\partial_A(I_0) \subseteq J$ *then* $I_0 \subseteq ecl_A(J)$.

PROOF. We show that for every $I \subseteq [m]$, $I \vdash_J^e (I_0 \setminus I)$. The claim follows by induction, as eventually every row of I_0 will be added to $ecl(J)$. Let A^* be the submatrix of A with the rows of I deleted. Let $j \in \partial_{A^*}(I_0 \setminus I)$ be given. If $j \in \partial_A(I_0)$, then by the hypothesis $\partial_A(I_0) \subseteq J$, $j \in J$. Otherwise, there are $i_1, i_2 \in I_0$ with $A_{i_1,j} = A_{i_2,j} = 1$, but i_2 is not a row of A^*, that is, $i_2 \in I$. Therefore, $j \in \bigcup_{i\in I} A_i$. Thus we have that $\partial_{A^*}(I_0 \setminus I) \subseteq J \cup \bigcup_{i\in I} A_i$ so that $I \vdash_J^e (I_0 \setminus I)$. $\dashv$

10.3. Extracting an expanding matrix with bounded column degree.

LEMMA 10.7. *For all constants* $\Delta > 0$, *there are constants* $c, d > 0$ *so that with probability* $1 - o(1)$ *over* F *chosen by the distribution* $F_3^{\Delta,n}$, *there exists a partial assignment* ρ *so that* $C_{A^F, b^F} \restriction_\rho$ *is a sub-CNF of* $C_{\hat{A},\hat{b}}$ *where* $\hat{A}$ *is an* $(m', n, d, cn, 2/5)$*-boundary expander with* $m' \geq m/2$.

PROOF. By Lemma 7.2, there is a constant $c > 0$ so that with probability $1 - o(1)$, in the system $A^F x = b^F$, the matrix A^F is a $(cn, 0.8)$-boundary expander. Set $r = cn$. Let J be $r/5$ many columns of largest hamming weight in A. Let $\hat{A} = CL_J(A)$. By Lemma 10.4, $\hat{A}$ is an $(r/2, 2/5)$-boundary expander. Let b be the maximum number of ones in a column of A that does not belong to J. Because there are $3\Delta n$ many ones in the matrix A, $(r/5)b = |J|b \leq 3\Delta n$. Therefore, $b \leq \frac{3\Delta n}{r/5} = \frac{15\Delta n}{cn} = \frac{15\Delta}{c}$. Set $d = \frac{15\Delta}{c}$. The matrix $\hat{A}$ contains at least $m/2$ rows because by Lemma 10.5, $|ecl(J)| \leq (2/c)|J| \leq (2/(4/5))(r/5) = r/2$, and thus the maximum number of rows deleted is $r/2 < m/2$.

Because $|ecl(J)| \leq r/2 < r$, by Lemma 10.3, there exists a partial assignment ρ to the variable of $\bigcup_{i \in ecl(J)} A_i$ that satisfies every equation $A_i x = b_i$ with $i \in ecl(J)$. Consider the system of equations $(Ax = b) \restriction_\rho$. If an equation $A_i x = b_i$ is not satisfied by ρ, then $i \notin ecl(J)$, and the restriction of $A_i x = b_i$ by ρ is $\hat{A}_i x = \hat{b}_i$ for some $\hat{b}_i \in \{0, 1\}$ (possibly $b_i \neq \hat{b}_i$). Therefore, $(Ax = b) \restriction_\rho$ is a subsystem of $\hat{A}x = \hat{b}$, and thus $\mathcal{C}_{A,b} \restriction_\rho$ is sub-CNF of $\mathcal{C}_{\hat{A},\hat{b}}$. ⊣

10.4. Local consistency and a normal form.

DEFINITION 10.3. *Let* t *be a term. We define* $ecl(t)$ *to be* $ecl(Vars(t))$. *We say that* t *is* locally consistent *if for the formula* $t \wedge [A_{ecl(t)} x = b_{ecl(t)}]$ *is satisfiable.*

LEMMA 10.8. *Let* t *be a locally consistent term. For every* $I \subseteq [m]$ *with* $|I| < r/2$, *the formula* $t \wedge [A_I x = b_I]$ *is satisfiable.*

PROOF. Suppose that $t \wedge [A_I x = b_I]$ is unsatisfiable. By linear algebra, there are $I' \subseteq I$ and $t' \subseteq t$ so that:

$$\sum_{i \in I'} (A_i - b_i) + \sum_{x_j^{\epsilon_j} \in t'} (x_j - \epsilon_j) = 1.$$

This forces $\partial_A(I') \subseteq Vars(t')$, so that by Lemma 10.6, $I' \subseteq ecl(t)$. This contradicts the hypothesis that t is locally consistent. ⊣

DEFINITION 10.4. *A DNF* F *is said to be in* normal form *if every term* $t \in F$ *is locally consistent.*

LEMMA 10.9. *Let* A *be an* (m, n, d, r, η) *boundary expander. Let* Γ *be a* $Res(k)$ *refutation of* $\mathcal{C}_{A,b}$. *There is a refutation* Γ' *of* $\mathcal{C}_{A,b}$ *so that the set of* k*-DNFs appearing in* Γ' *can be partitioned into two sets,* Γ'_0 *and* Γ'_1 *satisfying*:

1. *Every formula in* Γ'_0 *is a clause of width* $\leq \frac{6k}{\eta}$.
2. $|\Gamma'_1| \leq k|\Gamma|$ *and every DNF in* Γ'_1 *is locally consistent*.

Proof. First, we show that for every term $t = \bigwedge_{j=1}^{l} x_{i_j}^{\epsilon_j}$ that is locally inconsistent with respect to $Ax = b$, there is a width $\leq (6l/\eta)$ resolution derivation of $\bigvee_{j=1}^{l} x_{i_j}^{1-\epsilon_j}$ from $\mathcal{C}_{A,b}$. By Lemma 10.5, $|\mathit{ecl}(t)| \leq 2l/\eta$, so $A_{\mathit{ecl}(t)}x = b_{\mathit{ecl}(t)}$ is a system of at most $2l/\eta$ many equations, with each equation contains at most 3 variables. Therefore, the set of clauses encoding $A_{\mathit{ecl}(t)}x = b_{\mathit{ecl}(t)}$ is a set of width 3 clauses in at most $6l/c$ many variables. Because $\left(A_{\mathit{ecl}(t)}x = b_{\mathit{ecl}(t)}\right) \models \bigvee_{j=1}^{l} x_{i_j}^{1-\epsilon_j}$, by the implicational completeness of resolution with subsumption, there is a width $\leq 6l/eta$ derivation of $\bigvee_{j=1}^{l} x_{i_j}^{1-\epsilon_j}$.

The refutation Γ' is constructed as follows:

1. For every locally inconsistent term $t = \bigwedge_{j=1}^{l} x_{i_j}^{\epsilon_j}$ that appears in Γ, derive $\bigvee_{j=1}^{l} x_{i_j}^{\epsilon_j}$ using a derivation of width at most $6k/\eta$. These are the clauses of Γ'_0.
2. Let ρ be the partial assignment that falsifies every locally inconsistent literal l, that is, $\rho(l) = 0$ if l is locally inconsistent and all other variables are unset. For every locally inconsistent literal l, resolve $\neg l$ against all clauses of $\mathcal{C}_{A,b}$ that contain l, thus deriving $\mathcal{C}_{A,b} \restriction_\rho$. These clauses are locally consistent, and are placed into Γ'_1.
3. Now follow the proof structure of $\Gamma \restriction_\rho$, but do not construct any locally inconsistent terms of size ≥ 2: Inferences of the form $\frac{x_{i_1}^{\epsilon_1} \vee G \cdots x_{i_l}^{\epsilon_l} \vee G}{t \vee G}$ are replaced by resolution inferences against $x_{i_1}^{1-\epsilon_1} \vee \cdots \vee x_{i_l}^{1-\epsilon_l}$ to derive G. These clauses are placed in Γ'_1.

$\dashv$

10.5. Random restrictions and the switching lemma.

Definition 10.5. *Let* $A \in \{0,1\}^{m \times n}$ *be an* (r, η)*-boundary expander and let* $b \in \{0,1\}^m$ *be given. Let* $\rho_{A,b}$ *be partial assignment to the variables* $x_1, \ldots, x_n$ *generated by the following experiment: Uniformly select a subset* $X_1 \subseteq \{x_1, \ldots, x_n\}$ *of size* $\frac{\eta r}{4}$. *Let* $\hat{I} = \mathit{ecl}(X_1)$ *and let* $\hat{x} = X_1 \cup \{x_j \mid \exists i \in \hat{I},\ A_{i,j} = 1\}$. *The restriction* $\rho_{A,b}$ *is a uniformly selected assignment to* $\hat{x}$ *satisfying* $A_{\hat{I}}\hat{x} = b_{\hat{I}}$.

In the above definition, take note that $|X_1| \leq \eta r/4$, so that by Lemma 10.5, $|\mathit{ecl}(X_1)| < (2/\eta)|X_1| = (2/\eta)(\eta r/4) = r/2 < r$. Therefore, by Lemma 10.3, the system of equations $A_{\hat{I}}x = b_{\hat{I}}$ is satisfiable.

Definition 10.6. *Let* A *be a system of equation in variables* V. *Let* G_A *be the bipartite graph whose left vertices are* V *and whose right vertices are the equations of* A. *The* distance between two variables u and v, $d_A(u, v)$, *is their distance in the graph* G_A. *The distance between two terms* t_1 *and* t_2, $d_A(t_1, t_2)$, *is the minimum distance between variables u and v with u appearing in t_1 and v appearing in t_2.*

LEMMA 10.10. *Let A be an (r, η) boundary expander. Let I be a set of rows with $|I| < r/2$ and let t be a term so that the formula $t \wedge [A_I x = b_I]$ is satisfiable. The for any satisfiable term t_1 with $|t_1| \leq k$ and $d_A(ecl(t), t_1) > 4k/\eta$, the formula $t_1 \wedge t \wedge [A_I x = b_I]$ is also satisfiable.*

PROOF. Suppose that $t \wedge t_1 \wedge [A_I x = b_I]$ is unsatisfiable. By linear algebra, there is $t' \subseteq t$, $t_1' \subseteq t_1$ and $I' \subseteq I$ so that

$$\sum_{i \in I'} (Ax_i - b_i) + \sum_{x_j^{\epsilon_j} \in t'} (x_j - \epsilon_j) + \sum_{x_k^{\epsilon_k} \in t_1'} (x_k - \epsilon_k) = 1.$$

We immediately have that $\partial_A(I') \subseteq Vars(t') \cup Vars(t_1')$. Furthermore, because $(A_I x = b_I) \wedge t$ and t_1 are both satisfiable, there is a path connecting t_1 to t in G_{A_I}.

Case 1. $|I' \setminus ecl(t)| > 2k/\eta$. In this case,

$$\left|\partial_A(I') \setminus \Big(\bigcup_{i \in ecl(t)} A_i \cup Vars(t)\Big)\right| \leq |t_1| \leq k$$
$$= (\eta/2)(2k/\eta) < (\eta/2)|I' \setminus ecl(t)|.$$

In light of this and the fact that $|I'| \leq |I| < r/2$, $ecl(t) \vdash_t^e I' \setminus ecl(t)$. This contradicts the property that $ecl(t)$ is closed.

Case 2. $|I' \setminus ecl(t)| \leq 2k/\eta$. The minimum length path joining t_1 to $ecl(t)$ passes through at most $|I' \setminus ecl(t)|$ many variables not in $ecl(t)$ before reaching in $ecl(t)$, and thus it has length at most $\leq 2(2k/\eta) = 4k/\eta$. This contradicts the hypothesis $d_A(ecl(t), t_1) > 4k/\eta$. ⊣

LEMMA 10.11. *Let $Y \subseteq X$ be a set of variables. Assume that b is a partial assignment to Y that is distributed uniformly over some affine subspace of $\{0, 1\}^X$. For any term t in l many Y variables, either $Pr_b[t \restriction_b = 1] = 0$ or $Pr_b[t \restriction_b = 1] \geq 2^{-l}$.*

PROOF. Let $a + \mathcal{L}$ be the affine subspace of $\{0, 1\}^X$ over which b is distributed. Write $t = \bigwedge_{i=1}^{l} x_i^{\epsilon_i}$. Choose a basis extending the independent variables of t, ie. choose $I \subseteq [l]$ and vectors $\{e_i \mid i \in I\} \subseteq \{0, 1\}^X$ that are linearly independent modulo $\mathcal{L}$, and so that for $i \in [l] \setminus I$, b_i is equal to an affine combination of $\{b_j \mid j < i\}$. We immediately have that the probability that the term t is satisfied is either 0 or $2^{-|I|} \geq 2^{-l}$. ⊣

LEMMA 10.12. *Let A be an (m, n, d, r, η)-boundary expander such that $d \geq 2$. Let $b \in \{0, 1\}^m$ be arbitrary. There exists $a > 0$ (dependent upon only η and the ratio r/n, and increasing in both quantities) such that for any k-DNF F so that F is in normal form:*

$$Pr_{\rho_{A,b}}[F \restriction_{\rho_{A,b}} \neq 1] < 2^{-c(F)/d^{ak}}.$$

PROOF. Let F be a k-DNF in normal form with covering number $c(F)$. Let $\rho_{A,b}$, X_1 and $\hat{I}$ be generated as in Definition 10.5. The DNF F contains at least $c(F)/k$ many variable disjoint terms, and each of these has its variables contained in X_1 with independent probability $(\eta r/4n)^k$. Therefore, there expected number of variable disjiont terms from F whose variables are contained in X_1 is at least $c(F)/k(\eta r/4n)^k = \frac{c(F)}{k(\eta r/4n)^k}$. Let B_1 denote the event that there are strictly fewer than $\frac{c(F)}{2k(\eta r/4n)^k}$ many terms of F whose variables are contained in X_1. By the Chernoff bounds, Corollary 7.4, the probability of B_1 is at most $e^{-\frac{c(F)}{16k(\eta r/4n)^k}}$.

Consider the event that B_1 fails. Denote the set of variable disjoint terms from F whose variables are contained in X_1 as F_0. Define $M = \lfloor\frac{\eta\cdot c(F)}{4k^2 d^{\lceil 4k/\eta\rceil}(\eta r/4n)^k}\rfloor$. Let t_1 be the first term in F_0. Because t_1 is locally consistent, by Lemma 10.8, $t_1 \wedge [A_{\hat{I}}x = b_{\hat{I}}]$ is satisfiable, and thus by Lemma 10.11, t_1 is satisfied by $\rho_{A,b}$ with probability at least 2^{-k}. If t_1 is satisfied, terminate the process. Otherwise, we repeat as follows: Suppose that we have considered terms $t_1, \ldots, t_l$ from F_0. Let $t^{(l)}$ be the term corresponding to the values given to $Vars(t_1) \cup \cdots \cup Vars(t_l)$ by $\rho_{A,b}$. In the following paragraph it is shown that so long as $l \leq M$, there is a term $t \in F_0$ with $d_A(t, ecl(t^{(l)})) > 4k/\eta$; let t_{l+1} be such a term. By Lemma 10.10, t_{l+1} is consistent with $A_{\hat{I}}x = b_{\hat{I}} \wedge t^{(l)}$, and thus by Lemma 10.11, t_{l+1} is satisfied by $\rho_{A,b}$ with probability at least 2^{-k}. Let B_2 denote the event that none of the terms $t_1, \ldots, t_M$ are satisfied by ρ: Multiplying out the conditional probabilities shows that the probability of B_2 is at most $(1 - 2^{-k})^M \leq e^{-M/2^k}$.

Now we show that for any term t with $|t| < Mk$, there exists a term $t' \in F_0$ so that $d(ecl(t), t') > 4k/\eta$. Let V^* be the set of all variables at distance $\leq 4k/\eta$ from $ecl(t)$. Because $|t| < Mk \leq \lfloor\frac{\eta\cdot c(F)}{4k^2 d^{\lceil 4k/\eta\rceil}(\eta r/4n)^k}\rfloor \cdot k \leq \eta r/4$, by Lemma 10.5, $|ecl(t)| \leq 2|t|/\eta < 2Mk/\eta$. Because $|V^*| \leq d^{\lceil 4k/\eta\rceil}|ecl(t)| < d^{\lceil 4k/\eta\rceil}2Mk/\eta < d^{\lceil 4k/\eta\rceil}2\lfloor\frac{\eta\cdot c(F)}{4k^2 d^{\lceil 4k/\eta\rceil}(\eta r/4n)^k}\rfloor k/\eta \leq \frac{c(F)}{2k(\eta r/4n)^k} \leq |F_0|$, and F_0 contains only variable disjoint terms, there exists a term $t' \in F_0$ with $Vars(t) \cap V^* = \emptyset$. In other words, $d(ecl(t), t') > 4k/\eta$.

The event that $F \restriction_\rho \neq 1$ is contained within $B_1 \cup B_2$. Therefore, the probability that $F \restriction_\rho \neq 1$ is at most

$$e^{-\frac{c(F)}{16k(\eta r/4n)^k}} + e^{-\lfloor\frac{\eta\cdot c(F)}{4k^2 d^{\lceil 4k/\eta\rceil}(\eta r/4n)^k}\rfloor/2^k}$$

Taking a sufficiently small with respect to η and r/n completes the proof. $\dashv$

10.6. Width bound for expanding systems of linear equations.

LEMMA 10.13. *If A is an (m, n, d, r, η)-boundary expander, then* $w(\mathcal{C}_{A,b}) \geq \frac{r\eta}{2}$.

Proof. For each $i \in [m]$, let E_i denote the conjunction of clauses equivalent to $A_i x = b_i$. Define the measure of a clause C as $\mu(C) = \min\{|I|: \bigwedge_{i \in I} E_i \models C\}$. Observe that $\mu\colon \Gamma \to \{0, \ldots, m\}$ maps each clause of $\mathcal{C}_{A,b}$ to 1. Furthermore, $\mu(\emptyset) \geq r$ by Lemma 10.3. Finally, μ is subadditive with respect to the resolution rule: $\mu(A \vee B) \leq \mu(A \vee x) + \mu(B \vee \neg x)$.

Choose a clause C in Γ with $r/2 \leq \mu(C) < r$. Choose $I_0 \subseteq [m]$ so that $|I_0| = \mu(C)$ and $\bigwedge_{i \in I_0} E_i \models C$. Let $j_0 \in \delta(I_0)$ be given and let $i_0 \in I_0$ be the unique neighbor of j_0 in I_0. Suppose for the sake of contradiction that no variable of C contains the variable x_{j_0}. Choose an assignment α satisfying $\bigwedge_{i \in I_0 \setminus \{i_0\}} E_i$ and falsifying C. Define the assignment α' to agree with α off x_{j_0} and to set x_{j_0} to 1. Because C does not contain the variable x_{j_0}, $\alpha' \not\models C$. However, $\alpha' \models \bigwedge_{i \in I_0} E_i$. Thus we contradict the defining property of I_0, so for every $i_0 \in \delta(I_0)$ there is some variable X_{i_0, j_0} present in C and thus the width of C is at least $|\delta(I_0)| \geq \frac{\eta r}{2}$. ⊣

10.7. Proving Theorem 10.1.

Proof (of Theorem 10.1). By Lemma 10.7, there are constants $c, d > 0$ so that with probability $1 - o(1)$ over F selected by the distribution $F_3^{\Delta, n}$, there exists a partial assignment κ so that $C_{A^F, b^F} \restriction_\kappa$ is a sub-CNF of $C_{\hat{A}, \hat{b}}$ for some $(m', n, d, cn, 0.4)$-boundary expander $\hat{A}$ with $m' \geq m/2$. Consider n sufficiently large so that $15k \leq (1/k)((cn/40) - 1)$. We show that in this event, the minimum size of any Res(k) refutation of C_{A^F, b^F} is at least $S = (d^{ak}/k^2) 2^{\frac{(1/k)((cn/40)-1)}{2d^{ak^2}}}$. Suppose for the sake of contradiction that Γ is Res(k) refutation of C_{A^F, b^F} of size strictly less than S. By application of the partial assignment κ, $\Gamma \restriction_\kappa$ is a refutation of $C_{\hat{A}, \hat{b}}$.

By Lemma 10.9, there is a refutation Γ' of $C_{\hat{A}, \hat{b}}$ so that the DNFs of Γ' can be partitioned into sets Γ'_0 and Γ'_1 so that every formula in Γ'_0 is a clause of width at most $\frac{6k}{0.4} = 15k$, all DNFs in Γ'_1 are locally consistent, and $|\Gamma'_1| \leq k|\Gamma| < kS$.

Apply a random restriction $\rho = \rho_{\hat{A}, \hat{b}}$ to Γ'. By Lemma 10.11, there is a constant $a > 0$ so that every locally consistent k-DNF F has that $Pr_\rho[F \restriction_\rho \neq 1] < 2^{-c(F)/d^{ak}}$. Thus, by Corollary 9.3, for every k-DNF F, $Pr_\rho[h(F \restriction_\rho) > (1/k)((cn/40) - 1)] < \frac{k}{d^{ak}} 2^{-\frac{(1/k)((cn/40)-1)}{2d^{ak^2}}} = 1/(kS)$. By the union bound, there exists ρ so that every $F \in \Gamma'_1$ is strongly represented by a decision tree of height at most $(1/k)((cn/40) - 1)$. Moreover, every clause in Γ'_0 is strongly represented by a decision tree of height at most $(1/k)((cn/40) - 1)$ because each such clause has width $\leq 15k \leq (1/k)((cn/40) - 1)$. Therefore, by Theorem 9.4, there is a resolution refutation of $C_{\hat{A}, \hat{b}} \restriction_\rho$ of width at most $(cn/40) - 1$.

On the other hand, $C_{\hat{A}, \hat{b}} \restriction_\rho$ is a sub-CNF of C_{A^*, b^*} where A^* is an $(r/4, 0.2)$-boundary expander. By Lemma 10.13, all resolution refutations of C_{A^*, b^*} must have width $\geq cn/40$. Contradiction. ⊣

§11. Resolution pseudowidth and very weak pigeonhole principles. We do not obtain meaningful bounds for resolution refutations of PHP^m_n by using the techniques of Subsection 8.1 when $m \geq n^2$. Restricting to $PHP(G)$ where is G is a suitably expanding m to n bipartite graph, does not work because each pigeon must be allowed at least one hole and that forces the number of variables to be at least n^2, so that Corollary 8.2 yields only that $s(PHP^m_n) \geq s(PHP(G)) \geq 2^{\Theta((n-iw(PHP(G)))^2/n^2)} = \Theta(1)$ (where $s(F)$ denotes the minimum resolution refutation size of F). Similar difficulties are encountered when one tries to extend the bottleneck counting approach of [87, 60].

The first superpolynomial size lower bound for resolution refutations of PHP^m_n, with $m \geq n^2$, was shown by Ran Raz [137] (building upon similar bounds for regular resolution [131]). Subsequently, Alexander Razborov found a short proof based on the analysis of a parameter that he dubbed the *pseudowidth* [143]. Here we present the simplest version of this argument; stronger versions appear in [143, 142, 146].

THEOREM 11.1. [143] *For all natural numbers $m > n \geq 1$, every resolution refutation of PHP^m_n has size at least $2^{\sqrt{n/(512(\log_2 m)^2)}}$. Moreover, regardless of the value of m, every resolution refutation of PHP^m_n has size at least $2^{\sqrt[4]{n/4096}}$.*

11.1. A monotone normal form.

DEFINITION 11.1. [58] *The* monotone calculus *is refutation system for refuting instances of PHP^m_n. Its lines are positive clauses in the variables $x_{i,j}$, $i \in [m]$, $j \in [n]$. It has one inference rule, the* monotone rule. *Whenever $I_0, I_1 \subseteq [m]$ with $I_0 \cap I_1 = \emptyset$, and C_0, C_1 and C are positive clauses with $C_0 \cup C_1 \subseteq C$:*

$$\frac{C_0 \vee \bigvee_{i\in I_0} x_{i,j} \quad C_1 \vee \bigvee_{i\in I_1} x_{i,j}}{C}.$$

A monotone calculus refutation of PHP^m_n *is a sequence of positive clauses such that each clause is either $\bigvee_{j=1}^n x_{i,j}$ for some $i \in [m]$, or follows from two preceding clauses by the application of the monotone rule. The* size *of a monotone calculus refutation is the number of clauses that it contains. Let $s_{MC}(PHP^m_n)$ denote the minimum size of a monotone calculus refutation of PHP^m_n.*

LEMMA 11.2. [58] *For every m and n, $s_R(PHP^m_n) \geq s_{MC}(PHP^m_n)$.*

PROOF. Let Γ be a resolution refutation of PHP^m_n with $|\Gamma| = s_R(PHP^m_n)$. Replace every clause C in Γ by the positive clause C^M defined as follows: The clause C^M contains every positive literal contained in C, and for every negative literal $\neg x_{i,j}$ that appears in C, C^M contains the disjunction $\bigvee_{k\in[n]\setminus\{j\}} x_{i,k}$. Notice that whenever $A \subseteq B$, $A^M \subseteq B^M$, and that $\emptyset^M = \emptyset$. The initial clauses $\bigvee_{j\in[n]} x_{i,j}$ remain unchanged by the transformation $C \mapsto C^M$ but the initial clauses of the form $\neg x_{i,k} \vee \neg x_{j,k}$ become

$\bigvee_{l\in[n]\setminus\{k\}} x_{i,l} \vee \bigvee_{l\in[n]\setminus\{k\}} x_{j,l}$. The latter clauses are not legal initial clauses for a monotone calculus derivation, so we throw them away. Let Γ^M denote Γ with the initial clauses $\neg x_{i,k} \vee \neg x_{j,k}$ removed, and every other clause C replaced by C^M. Notice that the number of lines in Γ^M is no more than the number of lines in Γ.

We now show that Γ^M is a valid monotone calculus refutation of PHP^m_n. Consider a clause $C = A \vee \neg x_{k,j}$ that follows by resolving $A \vee x_{i,j}$ with an initial clause $\neg x_{i,j} \vee \neg x_{k,j}$. Notice that when we combine $(A \vee x_{i,j})^M = A^M \vee x_{i,j}$ with the initial clause $\bigvee_{l\in[n]} x_{l,j} = x_{k,j} \vee \bigvee_{l\in[n]\setminus\{j\}} x_{k,l}$ using the monotone rule, we obtain $A^M \vee \bigvee_{l\in[n]\setminus\{j\}} x_{k,l} = C^M$. Finally, consider a clause $C = A \vee B$ that follows from $A \vee x_{i,j}$ and $B \vee \neg x_{i,j}$ by an application of the resolution rule and B is not an initial clause of the form $\neg x_{i,j} \vee \neg x_{k,j}$. We have that $(A \vee x_{i,j})^M = A^M \vee x_{i,j}$, that $(B \vee \neg x_{i,j})^M = B^M \vee \bigvee_{k\in[n]\setminus\{j\}} x_{i,k}$, and that $C^M \subseteq B^M \vee A^M$. Applying the monotone rule derives C^M. ⊣

11.2. Pseudowidth.

Definition 11.2. *Let C be a positive clause. For each $i \in [m]$, the* holes for pigeon i in C *is defined as $J_i(C) = \{j \in [n] \mid x_{i,j}$ occurs in $C\}$. The degree for pigeon i in C is $d_i(C) := |J_i(C)|$. Let $\vec{d} \in [n]^m$ be given; we call $\vec{d}$ a* filter. *A $\vec{d}$*-axiom *is a clause $\bigvee_{j\in J} x_{i,j}$ with $|J| \geq d_i$. Let δ be given. We say that pigeon $i \in [m]$* passes *the filter if $d_i(C) < d_i$, and that it* narrowly passes *the filter $\vec{d}$ if $0 < d_i - d_i(C) \leq \delta$. Define the set of narrowly-passing pigeons for a positive clause C with filter $\vec{d}$ and margin δ as $I_{\vec{d},\delta}(C) = \{i \in [m] \mid d_i - \delta \leq d_i(C) < d_i\}$. The* pseudo-width *of C with respect to $\vec{d}$ and δ, $w_{\vec{d},\delta}(C)$, is the number of pigeons in C that narrowly pass the filter: $w_{\vec{d},\delta} = |I_{\vec{d},\delta}(C)|$. The pseudo-width of a monotone calculus refutation is the maximum pseudowidth of its clauses.*

11.3. Reducing the pseudowidth of a small refutation.

Lemma 11.3. *Let m and n be integers, with $m > n \geq 1$, and define $\delta = \frac{n}{2\log_2 m}$. Suppose that there exists a monotone calculus refutation Γ of PHP^m_n that has size $\leq S$. There exists an integer vector $\vec{d} \in [n]^m$ so that (1) for each $i \in [m]$, $d_i > \delta$, and (2) there exists a monotone calculus refuation Γ' of a set of $\vec{d}$-axioms which also has size $\leq S$ and has $w_{\vec{d},\delta}(\Gamma') \leq 16 \ln S$.*

Proof. For each clause C of Γ, define the vector $\vec{r}(C) \in [n]^m$ as $r_i(C) = \lfloor (n - d_i(C))/\delta \rfloor + 1$. Let $W(C) = \sum_{i=1}^m 2^{-r_i(C)}$. Below we use a probabilistic construction to generate $\vec{d}$ so that for every clause C of Γ:

$$W(C) \geq 2\ln S \Rightarrow \exists i \in [m],\ d_i \leq d_i(C),$$
$$W(C) \leq 2\ln S \Rightarrow |\{i \in [m] \mid d - d_i \leq \delta\}| \leq 16 \ln S.$$

Call this property "Property A". Set $t = \lfloor \log_2 m \rfloor - 1$ and let D be the random variable that takes the value $n - \delta r$ with probability 2^{-r} (for $r = 1, \ldots, t-1$), and that takes the value $n - \delta t$ with probability 2^{1-t}. Choose the vector $\vec{d}$ using m independent trials of D. Notice that property (1) is satisfied because the smallest each d_i can be is $n - \delta t = n - \delta(\lfloor \log_2 m \rfloor - 1) = \delta + n - \delta \lfloor \log_2 m \rfloor \geq \delta + n - \delta \log_2 m = \delta + n - (n/2 \log_2 m) \log_2 m = \delta + n/2 > \delta$.

Consider each clause C with $W(C) \geq 2 \ln S$. Let $H = \{i \in [m] \mid r_i(C) \leq t\}$. Clearly, $\sum_{i \in [m] \setminus H} 2^{r_i(C)} \leq m2^{-t+1} = m2^{-\lfloor \log_2 m \rfloor + 1} \leq 2$, so that $\sum_{i \in H} 2^{-r_i(C)} \geq 2 \ln S - 2$. Now consider one of the events $d_i(C) \geq d_i$ with $i \in H$. Because $d_i(C) = n - \delta \left(\frac{n - d_i(c)}{\delta} \right) \geq n - \delta \left(\lfloor \frac{n - d_i(c)}{\delta} \rfloor + 1 \right) = n - \delta r_i(C)$, we have that $Pr[d_i(C) \geq d_i] \geq Pr[n - \delta r_i(C) \geq d_i] \geq 2^{-r_i(C)}$. Because the events $d_i(C) \geq d_i$ are independent for distinct i:

$$Pr[\forall i \in [m],\ d_i(C) < d_i] \leq Pr[\forall i \in H,\ d_i(C) < d_i] = \prod_{i \in H} (1 - 2^{-r_i(C)})$$
$$\leq e^{-\sum_{i \in H} 2^{-r_i(C)}} \leq e^{-(2 \ln S - 2)} < S^{-1}.$$

Consider each clause with $W(C) \leq 2 \ln S$. Note that for all $i \in [m]$, $\delta(r_i(C) - 1) \leq n - d_i(C)$. For each $i \in [m]$, $Pr[d_i(C) \geq d_i - \delta] = Pr[d_i \leq d_i(C) - \delta] \leq Pr[d_i \leq n - \delta r_i(C)] \leq 2^{2 - r_i(C)}$. Therefore:

$$\mathbb{E}[|\{i \in [m] \mid d_i(C) \geq d_i - \delta\}|] \leq 4 \sum_{i=1}^{m} 2^{-r_i(C)} = 4W(C) \leq 8 \ln S.$$

Because the events are independent, by Corollary 7.4 (Chernoff–Hoeffding bounds), the probability that $|\{i \in [m] \mid d_i(C) \geq d_i - \delta\}| \geq 16 \ln S$ is $\leq e^{-(3/8) 8 \ln S} \leq e^{-3 \ln S} < S^{-1}$.

Because there are at most S clauses in the refutation Γ, and Property A fails at each clause with probability $< S^{-1}$, there is a choice of $\vec{d}$ so that Property A holds for every clause of Γ. Every clause C such that $\exists i \in [m],\ d_i \leq d_i(C)$ is subsumed by some $\vec{d}$-axiom $\bigvee_{j \in J} x_{i,j}$. Replace C by one of the subsuming $\vec{d}$-axioms, the pseudowidth of the $\vec{d}$-axiom is one. Notice that replacing C by $C' \subseteq C$ preserves all applications of the monotone inference rule when C is a hypothesis. We remove any inferences in which C is a consequent because it has been replaced by a $\vec{d}$-axiom. ⊣

11.4. A lower bound on pseudowidth.

Lemma 11.4. *Let $\mathcal{A}$ be a set of $\vec{d}$-axioms and let $\delta > 0$ be given with $\delta < \min_{i \in [m]} d_i$. Every monotone refutation $\mathcal{R}$ of $\mathcal{A}$ satisfies $w_{d,\delta}(\mathcal{R}) \geq \delta^2/(8n \ln |\mathcal{A}|)$.*

Proof. Let $w_0 = \frac{\delta^2}{8n \ln |\mathcal{A}|}$. Suppose for the sake of contradiction that Γ is a monotone calculus refutation of PHP_n^m with pseudowidth $< w_0$.

For each assignment a let $J_i(a) = \{j \in [m] \mid a_{i,j} = 1\}$. Set $l = \lceil \frac{\delta}{4w_0} \rceil$. Let D be the set of partial assignments a such that $\forall i_1, i_2 \in [m],\ i_1 \neq i_2 \Rightarrow J_{i_1}(a) \cap J_{i_2}(a) = \emptyset$ and $\forall i \in [m],\ |J_i(a)| \leq l$.

Let $\models$ denote entailment with respect to the assignments of D: Let $\mathcal{S}$ be a set of positive clauses and let C be a positive clause. If for all $a \in D$, $\big(\forall B \in \mathcal{S}, B(a) = 1\big) \Rightarrow \big(C(a) = 1\big)$, then we write $\mathcal{S} \models C$.

For each $i \in [m]$ let $\mathcal{A}_i$ be the set of axioms from $\mathcal{A}$ of the form $\bigvee_{j \in J} x_{i,j}$. For $i \subseteq [m]$, let $\mathcal{A}_I = \bigcup_{i \in I} \mathcal{A}_i$. For each C let $\mathcal{A}_C = \mathcal{A}_{I_{\vec{d},\delta}(C)}$. We now show that for all $C \in \mathcal{R}$, $\mathcal{A}_C \models C$. This is a contradiction because $\emptyset \in \Gamma$, yet $\mathcal{A}_\emptyset$ is the empty set of clauses so $\mathcal{A}_\emptyset \not\models \emptyset$.

For $C \in \mathcal{A}$, we have $C \in \mathcal{A}_{I_{d,\delta}(C)} = \mathcal{A}_C$ and thus $\mathcal{A}_C \models C$. Now consider the induction step: $\mathcal{A}_{C_0} \models C_0$, $\mathcal{A}_{C_1} \models C_1$, and C follows from C_0 and C_1 via the monotone rule. By soundness of the monotone rule, $\mathcal{A}_{I_{\vec{d},\delta}(C_0)} \cup \mathcal{A}_{I_{\vec{d},\delta}(C_1)} \models C$. Let $I \subseteq I_{\vec{d},\delta}(C_0) \cup I_{\vec{d},\delta}(C_1)$ be of minimal size so that $\mathcal{A}_I \models C$. Below we show that $I \subseteq I_{\vec{d},\delta}(C)$, which guarantees that $\mathcal{A}_{I_{\vec{d},\delta}(C)} \models C$.

We now show that $I \subseteq I_{d,\delta}(C)$. Suppose not and choose $i_0 \in I \setminus I_{d,\delta}(C)$. By minimality, $\mathcal{A}_{I \setminus \{i_0\}} \not\models C$. Choose $a \in D$, so that a satisfies $\mathcal{A}_{I \setminus \{i_0\}}$ but a does not satisfy C. Because all clauses in Γ are positive, we may assume that $a_{ij} = 0$ for all $i \notin I \setminus \{i_0\}$. Set $J_0 = \bigcup_{i \in I} J_i(a) \cup J_{i_0}(C)$ and set $J_1 = [n] \setminus J_0$. Notice that for every $j \in J_1$, the assignment $a \cup a_{i_0,j} = 1$ also falsifies C, and that we have $|J_1| \geq n - (2w_0 l + d_{i_0} - \delta) \geq n - d_{i_0} + \delta/2$.

Uniformly select an set J from $\binom{J_1}{l}$. Extend a to a^J by setting $a_{i_0,j} = 1$ for all $j \in J$. Notice that for all $J \in \binom{J_1}{l}$, $a^J \in D$ because $|J| = l$, and $J \subseteq J_1 \subseteq [n] \setminus \big(\bigcup_{i \in I} J_i(a)\big)$. Consider $A \in \mathcal{A}_{i_0}$. Since $|J_{i_0}(A)| \geq d_{i_0}$ we have that $|J_{i_0}(A) \cap J_1| \geq \delta/2$, and thus $|J_{i_0}(A) \cap J_1| \geq \lceil \delta/2 \rceil$. Therefore:

$$
\begin{aligned}
Pr_J[A(a^J) \neq 1] = Pr_J[J_{i_0}(A) \cap J = \emptyset] &\leq \prod_{k=1}^{\lceil \delta/2 \rceil} \left(1 - \frac{l}{|J_1| - k}\right) \\
&< \prod_{k=1}^{\lceil \delta/2 \rceil} \left(1 - \frac{l}{n - d_{i_0} + \delta/2 - k}\right) < \prod_{k=1}^{\lceil \delta/2 \rceil} \left(1 - \frac{l}{n}\right) \\
&\leq e^{-\frac{\delta l}{2n}} = e^{-\frac{\delta \lceil \delta/4w_0 \rceil}{2n}} \leq e^{-\frac{\delta^2}{8n\left(\delta^2/(8n \ln |\mathcal{A}|)\right)}} \\
&= e^{-\ln |\mathcal{A}|} = |\mathcal{A}|^{-1}.
\end{aligned}
$$

By the union bound, the probability over choices of J that there exists $A \in \mathcal{A}_{i_0}$ that is not be satisfied is < 1. Therefore there is some $J \in \binom{J_1}{l}$ such that a^J satsfies every clause of $\mathcal{A}_{i_0}$. Moreover, because a^J extends a, a^J satisfies every clause of $\mathcal{A}_{I \setminus \{i_0\}}$. On the other hand, because $J \subseteq J_1$, a^J

falsifies C. We have demonstrated $a^J \in D$ such that a^J satisfies every clause of $\mathcal{A}_I$ but a^J falsifies C, so $\mathcal{A}_I \not\models C$, contradicting the choice of I. $\dashv$

11.5. The proof of Theorem 11.1. Let Γ be a monotone calculus refutation of PHP_n^m of size S. Apply Lemma 11.3 and choose $\vec{d}$ with $\delta = n/2\log_2 m$ so that $w_{\vec{d},\delta}(\Gamma) \leq 16 \ln S$. By Lemma 11.4, however, $w_{\vec{d},\delta}(\Gamma) \geq \delta^2/(8n \ln S) = (n/2\log_2 m)^2/8n \ln S = n/(32(\log_2 m)^2(\ln S))$. Therefore $16 \ln S \geq n^2/(32(\log_2 m)^2(\ln S))$, and thus $\ln S \geq \sqrt{\frac{n}{512(\log_2 m)^2}}$.

Because a monotone calculus refutation of size S can use at most S axioms, each mentioning at most two pigeons, we always have the relation that $m \leq 2S$. Therefore $\ln S \geq \sqrt{\frac{n}{512(\log_2 2S)^2}}$. Rearranging (and bounding some sloppy constants) reveals that $8 \ln^4 S \geq \frac{n}{512}$ so that $\ln S \geq \sqrt[4]{\frac{n}{4096}}$. By Lemma 11.2, these bounds also apply to resolution refutations of PHP_n^m.

Part 3. Open problems, further reading, acknowledgments.

There are several propositional proof systems for which we do not yet have superpolynomial size lower bounds. Of particular interest are the the Lovász–Schrijver systems and constant-depth Frege systems with modular counting gates, as superpolynomial formula size bounds are known for the formulas of these proof systems but no superpolynomial size lower bounds are known for the proof systems. Lower bounds for Frege proofs conditioned upon a complexity theoretic assumption weaker than $NP \neq coNP$ would also be very interesting.

Are there polynomial size, constant-depth Frege refutations of PHP_n^{2n}? And if so, can $I\Delta_0(R)$ prove prove $php_n^{2n}(R)$? A positive resolution to this problem would solve the long-standing open problem of whether or not $I\Delta_0$ can prove the infinitude of the primes, and its negative resolution would require new techniques that distinguish between computability by constant-depth formulas and provability by constant-depth proofs [136].

There are several propositional proof systems for which the complexity of refuting random 3-CNFs is unknown, such as cutting planes, Lovász–Schrijver refutations, OBDD refutations and constant-depth Frege systems. Results for any of these would be interesting. Moreover, it would be nice if size lower bounds for refuting random 3-CNFs by arbitrary propositional proof systems could be established under a plausible complexity theoretic conjecture.

The current notion of automatizability considers only the time complexity of finding reasonably small refutations when very small refutations are known to exist. However, as discussed in Subsection 6.1, for many satisfiability algorithms, space consumption is also a bottleneck. So what can be said about automatizability that takes to accout both time and space?

Our understanding of whether or not there exists a p-optimal propositional proof system is still somewhat hazy. It would be wonderful if the (non)-existence of a p-optimal system could be shown to follow from a plausible hypothesis or to entail an implausible consequence. Metamathematical aspects could be worth investigating as well.

Further reading. For more on theories of bounded arithmetic, consult [50, 134, 100]. A survey by Alexander Razborov [144] provides further material on the proof complexity of the propositional pigeonhole principle, and gives proofs for a connection with the provability of circuit lower bounds. A survey by Jacobo Torán [158] provides further information on connections between between resolution space, size, and width.

This is not the only survey on propositional proof complexity, and the others offer a different emphasis. Results prior to 1995 are more thoroughly covered in [159] and [100]. Parallels between circuit and proof complexity are dealt with more thoroughly in [31]. Feasible interpolation, automatizability, and lower bounds for constant-depth systems via Håstad's switching lemma are covered more thoroughly in [26] than in this article.

Acknowledgments. The author wishes to thank Paul Beame for answering some questions on [33] and [2], and also for providing Figure 8. Thanks also go to Sam Buss for a helpful conversation on $I\Delta_0$, and Jakob Nordström for providing comments on an early draft.

On August 5th, 2006, propositional proof complexity lost one of its leading young contributors when Misha Alekhnovich was killed in a kayaking accident in Russia. This survey is dedicated to his memory.

Appendix. Notation.

For a binary string s we let $|s|$ denote the length of s. For a set S and a natural number k we write $\binom{S}{k}$ to denote the set of all size k subsets of S. For a graph G, we will write $\sim_G$ to denote the adjacency relation of G.

A *literal* is a variable or its negation. For a variable x, we sometimes write x^0 to denote the literal $\neg x$ and x^1 to denote the literal x. The literal x is said to be *positive*, and the literal $\neg x$ is said to be *negative*.

A *clause* is a constant 0 or 1 or a disjunction of literals. Our convention is that a clause is specified as a set of literals, with 0 corresponding to the empty set and 1 to any literal and its negation. We say that a clause C contains a literal l if $l \in C$, and that a clause C contains a variable x if either $x \in C$ or $\neg x \in C$. Dually, a *term* is a constant 0 or 1 or a conjunction of literals. Our convention is that a term is specified as a set of literals, with 1 corresponding to the empty set and 0 to any literal and its negation. We say that a term T contains a literal l if $l \in T$, and that a term T contains a variable x if either $x \in T$ or $\neg x \in T$. We often identify literals with clauses and terms of size one, and will write l instead of $\{l\}$. A *DNF* is a disjunction of terms, specified as a set of terms. A k-*DNF* is a DNF whose terms are each of size at most k. A *clause* is a 1-DNF, i.e., a disjunction of literals. The width of a clause C, written $w(C)$, is the number of literals appearing in C. The width of a set of clauses is the maximum width of any clause in the set. A *CNF* is a conjunction of clauses, specified as a set of clauses. A k-*CNF* is a CNF whose clauses are each of width at most k. Two terms t and t' are *consistent* if there is no literal l with $l \in t$ and $\neg l \in t'$.

The notation $\bigvee_{i=1}^{m} F_i$ denotes the disjunction of formulas F_i and the notation $\bigwedge_{i=1}^{m} F_i$ denotes their conjunction; the order of parenthesization is not relevant in contexts that use this notation.

For a Boolean formula F, the *alternation depth of* F, written $dp(F)$, is the maximum number of alternations between connectives along any path from F's root connective to a literal. A literal has depth zero.

A substitution is a mapping from propositional variables to propositional formulas. When F is a formula and σ is a substitution, $F[\sigma]$ denotes the formula obtained by simultaneously replacing every variable by its image under σ. There is no further simplification of the formula.

A *restriction* is a mapping from a set of variables to $\{0, 1, *\}$. This is thought of as a substition that maps every x to either 0, 1 or x (where $\rho(x) = *$ in the event that x maps to x- "x is unset"). For a formula F and a restriction ρ, *the restriction of* F *by* ρ, $F\restriction_\rho$ is defined a defined as usual, simplifying when a sub-expression has become explicitly constant. For any restriction ρ, let $\mathrm{dom}(\rho)$ denote the set of variables to which ρ assigns the value 0 or 1. Sometimes, we represent a restriction by a set of literals π, with the interpretation that a variable x maps to 0 if $\neg x \in \pi$, to 1 if $x \in \pi$ and it is unchanged otherwise.

When F and G are Boolean formulas we write $G \models F$ to mean that whenever G is satisfied, then F is also satisfied. Similarly, if S is a set of formulas and F is a formula, $S \models F$ means that whenever every formula of S is satisfied, F is also satisfied.

Let f and g be functions from $\mathbb{N}$ to $\mathbb{N}$. We write $f = O(g)$ if there exists $c > 0$ and $n_0 \in \mathbb{N}$ so that $\forall n \geq n_0,\ f(n) \leq c \cdot g(n)$. We write $f = \Omega(g)$ if there exists $c > 0$ and $n_0 \in \mathbb{N}$ so that $\forall n \geq n_0,\ g(n) \leq c \cdot f(n)$. We write $f = \Theta(g)$ if $f = O(g)$ and $f = \Omega(g)$.

REFERENCES

[1] D. Achlioptas, P. Beame, and M. Molloy, *Exponential lower bounds for DPLL below the satisfiability threshold*, ***Proceedings of the fifteenth annual ACM-SIAM symposium on discrete algorithms***, 2004, pp. 139–140.

[2] ———, *A sharp threshold in proof complexity yields lower bounds for satisfiability search*, ***Journal of Computer and System Sciences***, vol. 68 (2004), no. 2, pp. 261–276.

[3] A. Aguirre and M. Vardi, *Random 3-SAT and BDDs: The plot thickens further*, ***Principles and practice of constraint programming***, 2001, pp. 121–136.

[4] D. Aharonov and T. Naveh, *Quantum NP- a survey*, Technical Report `arXiv:quant-ph/02-210077`, arXiv.org, 2002.

[5] M. AJTAI, *Parity and the pigeonhole principle*, ***Feasible mathematics*** (S. Buss and P. Scott, editors), Progress in Computer Science and Applied Logic, vol. 9, Birkhauser, 1990, pp. 1–24.

[6] ———, *The complexity of the pigeonhole principle*, ***Combinatorica***, vol. 14 (1994), no. 4, pp. 417–433.

[7] ———, *The independence of the modulo p counting principles*, ***Proceedings of the twenty-sixth annual ACM symposium on the theory of computing***, 23–25 May 1994, pp. 402–411.

[8] M. ALEKHNOVICH, *Lower bounds for k-DNF resolution on random 3-CNFs*, ***Proceedings of the thirty-seventh annual ACM symposium on the theory of computing***, 2005, pp. 251–256.

[9] M. ALEKHNOVICH, S. ARORA, and I. TOURLAKIS, *Towards strong nonapproximability results in the Lovász–Schrijver hierarchy*, ***Proceedings of the thirty-seventh annual ACM symposium on theory of computing***, 2005, pp. 294–303.

[10] M. ALEKHNOVICH, E. BEN-SASSON, A. RAZBOROV, and A. WIGDERSON, *Space complexity in propositional calculus*, ***SIAM Journal on Computing***, vol. 31 (2002), no. 4, pp. 1184–1211.

[11] M. ALEKHNOVICH, E. HIRSCH, and D. ITSYKSON, *Exponential lower bounds for the running times of DPLL algorithms on satisfiable formulas*, ***Journal of Automated Reasoning***, vol. 35 (2005), no. 1-3, pp. 51–72.

[12] M. ALEKHNOVICH and A. RAZBOROV, *Resolution is not automatizable unless $W[P]$ is tractable*, ***Proceedings of the forty-second annual symposium on foundations of computer science***, 2001, pp. 210–219.

[13] ———, *Lower bounds for polynomial calculus: non-binomial case*, ***Proceedings of the Steklov Institute of Mathematics***, vol. 242 (2003), pp. 18–35.

[14] F. ALOUL, M. MNEIMNEH, and K. SAKALLAH, *ZBDD-based backtrack search SAT solver*, ***Eleventh IEEE/ACM workshop on logic & synthesis***, 2002, pp. 131–136.

[15] F. ALOUL, A. RAMANI, I. MARKOV, and K. SAKALLAH, *Solving difficult instances of Boolean satisfiability in the presence of symmetry*, ***IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems***, vol. 22 (2003), no. 9, pp. 1117–1137.

[16] F. ALOUL, K. SAKALLAH, and I. MARKOV, *Efficient symmetry breaking for boolean satisfiability*, ***IEEE Transactions on Computers***, vol. 55 (2006), no. 5, pp. 541–558.

[17] F. A. ALOUL, A. RAMANI, I. L. MARKOV, and K. A. SAKALLAH, *PBS: a pseudo-boolean solver and optimizer*, ***Proceedings of the fifth international conference on theory and applications of satisfiability testing***, 2002, pp. 346–353.

[18] A. ATSERIAS, *Improved bounds on the weak pigeonhole principle and infinitely many primes from weaker axioms*, ***Theoretical Computer Science***, vol. 295 (2003), no. 1–3, pp. 27–39.

[19] A. ATSERIAS and M. L. BONET, *On the automatizability of resolution and related propositional proof systems*, ***Information and Computation***, vol. 189 (2004), no. 2, pp. 182–201.

[20] A. ATSERIAS, M. L. BONET, and J. L. ESTEBAN, *Lower bounds for the weak pigeonhole principle beyond resolution*, ***Information and Computation***, vol. 176 (2002), pp. 136–152.

[21] A. ATSERIAS and V. DALMAU, *A combinatorial characterization of resolution width*, ***Proceedings of the eighteenth annual IEEE conference on computational complexity***, 2003, pp. 239–247.

[22] A. ATSERIAS, P. KOLAITIS, and M. VARDI, *Constraint propagation as a proof system*, ***Tenth international conference on principles and practice of constraint programming***, 2004, pp. 77–91.

[23] J. AVIGAD, *Plausibly hard combinatorial tautologies*, ***Proof complexity and feasible arithmetics***, American Mathematical Society, 1997, pp. 1–12.

[24] R. Bayardo and R. Schrag, *Using CSP look-back techniques to solve exceptionally hard SAT instances*, ***Proceedings of the second international conference on principles and practice of constraint programming***, 1996, pp. 46–60.

[25] P. Beame, *A switching lemma primer*, Technical report, Department of Computer Science and Engineering, University of Washington, 1994.

[26] ———, *Proof complexity*, ***Computational complexity theory*** (S. Rudich and A. Wigderson, editors), IAS/Park City mathematics series, vol. 10, American Mathematical Society, 2004, pp. 199–246.

[27] P. Beame, R. Impagliazzo, J. Krejíček, T. Pitassi, and P. Pudlák, *Lower bounds on Hilbert's Nullstellensatz and propositional proofs*, ***Proceedings of the London Mathematical Society***, vol. 73 (1996), no. 3, pp. 1–26.

[28] P. Beame, R. Impagliazzo, T. Pitassi, and N. Segerlind, *Memoization and DPLL: Formula caching proof systems*, ***Proceedings of the eighteenth IEEE conference on computational complexity***, 2003, pp. 225–236.

[29] P. Beame, R. Karp, T. Pitassi, and M. Saks, *The efficiency of resolution and davis–putnam procedures*, ***SIAM Journal on Computation***, vol. 31 (2002), no. 4, pp. 1048–1075.

[30] P. Beame, H. Kautz, and A. Sabharwal, *Towards understanding and harnessing the potential of clause learning*, ***Journal of Artificial Intelligence Research***, vol. 22 (2004), pp. 319–351.

[31] P. Beame and T. Pitassi, *Propositional proof complexity: Past, present, and future*, ***Current trends in theoretical computer science: Entering the 21st century*** (G. Paul, G. Rozenberg, and A. Salomaa, editors), World Scientific Publishing, 2001, pp. 42–70.

[32] P. Beame, T. Pitassi, and N. Segerlind, *Lower bounds for Lovász–Schrijver systems and beyond follow from multiparty communication complexity*, ***Proceedings of the thirty-second international colloquium on automata, languages, and programming***, 2005, pp. 1176–1188.

[33] P. Beame and S. Riis, *More one the relative strength of counting principles*, ***Proof complexity and feasible arithmetics*** (Paul Beame and Sam Buss, editors), American Mathematical Society, 1998, pp. 13–35.

[34] S. Bellantoni, T. Pitassi, and A. Urquhart, *Approximation and small depth frege proofs*, ***Proceedings of the sixth annual IEEE conference on structure in complexity theory***, 1991, pp. 367–391.

[35] E. Ben-Sasson, *Size space trade-offs for resolution*, ***Proceedings of the thirty-fourth annual ACM symposium on theory of computing***, 2002, pp. 563–572.

[36] E. Ben-Sasson and R. Impagliazzo, *Random CNFs are hard for the polynomial calculus*, ***Fortieth annual IEEE symposium on foundations of computer science***, 1999, pp. 415–421.

[37] E. Ben-Sasson, R. Impagliazzo, and A. Wigderson, *Near optimal separation of tree-like and general resolution*, ***Combinatorica***, vol. 24 (2004), no. 4, pp. 585–603.

[38] E. Ben-Sasson and A. Wigderson, *Short proofs are narrow—resolution made simple*, ***Journal of the ACM***, vol. 48 (2001), no. 2, pp. 149–169.

[39] J. H. Bennett, ***On spectra***, Ph.D. thesis, Princeton University, 1962.

[40] M. Bonet and N. Galesi, *A study of proof search algorithms for resolution and polynomial calculus*, ***Proceedings of the fortieth annual IEEE symposium on foundations of computer science***, 1999, pp. 422–431.

[41] M. Bonet, T. Pitassi, and R. Raz, *Lower bounds for cutting planes proofs with small coefficients*, ***The Journal of Symbolic Logic***, vol. 62 (1997), no. 3, pp. 708–728.

[42] ———, *On interpolation and automatization for Frege systems*, ***SIAM Journal on Computing***, vol. 29 (2000), no. 6, pp. 1939–1967.

[43] M. L. Bonet, C. Domingo, R. Gavaldà, A. Maciel, and T. Pitassi, *Non-automatizability of bounded-depth Frege proofs*, ***Computational Complexity***, vol. 13 (2004), no. 1–2, pp. 47–68.

[44] M. L. Bonet, J. L. Esteban, N. Galesi, and J. Johannsen, *On the relative complexity of resolution refinements and cutting planes proof systems*, ***SIAM Journal on Computation***, vol. 30 (2000), no. 5, pp. 1462–1484.

[45] R. Boppana and M. Sipser, *The complexity of finite functions*, ***Handbook of theoretical computer science, volume A***, Elsevier and MIT Press, 1990.

[46] R. Bryant, *Graph-based algorithms for boolean function manipulation*, ***IEEE Transactions on Computers***, vol. C-35 (1986), no. 8, pp. 677–691.

[47] ———, *Symbolic boolean manipulation with ordered binary decision diagrams*, ***ACM Computing Surveys***, vol. 24 (1992), no. 3, pp. 293–318.

[48] J. Buresh-Oppenheim, M. Clegg, R. Impagliazzo, and T. Pitassi, *Homogenization and the polynomial calculus*, ***Computational Complexity***, vol. 11 (2002), no. 3–4, pp. 91–108.

[49] J. Buresh-Oppenheim, N. Galesi, S. Hoory, A. Magen, and T. Pitassi, *Rank bounds and integrality gaps for cutting planes procedures*, ***Theory of Computing***, vol. 2 (2006), pp. 65–90.

[50] S. Buss, ***Bounded arithmetic***, Studies in Proof Theory, vol. 3, Bibliopolis, 1986.

[51] ———, *Polynomial size proofs of the propositional pigeonhole principle*, ***The Journal of Symbolic Logic***, vol. 52 (1987), no. 4, pp. 916–927.

[52] ———, *Propositional consistency proofs*, ***Annals of Pure and Applied Logic***, vol. 52 (1991), no. 1–2, pp. 3–29.

[53] ———, *Bounded arithmetic and propositional proof complexity*, ***Logic and computation*** (H. Schwichtenberg, editor), Springer-Verlag, 1997, pp. 67–122.

[54] ———, *Lower bounds on Nullstellensatz proofs via designs*, ***Proof complexity and feasible arithmetics*** (S. Buss and P. Beame, editors), American Mathematical Society, 1998, pp. 59–71.

[55] ———, *Bounded arithmetic, proof complexity and two papers of Parikh*, ***Annals of Pure and Applied Logic***, vol. 96 (1999), no. 1–3, pp. 43–55.

[56] S. Buss, D. Grigoriev, R. Impagliazzo, and T. Pitassi, *Linear gaps between degrees for the polynomial calculus modulo distinct primes*, ***Journal of Computer and System Sciences***, vol. 62 (2001), pp. 267–289.

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

[58] S. Buss and T. Pitassi, *Resolution and the weak pigeonhole principle*, ***Proceedings of the eleventh internation workshop on computer science logic***, 1997, pp. 149–156.

[59] ———, *Good degree bounds on Nullstellensatz refutations of the induction principle*, ***Journal of Computer and System Sciences***, vol. 57 (1998), pp. 162–171.

[60] S. Buss and G. Turán, *Resolution proofs of generalized pigeonhole principles*, ***Theoretical Computer Science***, vol. 62 (1988), no. 3, pp. 211–217.

[61] P. Chatalic and L. Simon, *Multi-resolution on compressed sets of clauses*, ***Proceedings of the twelfth international conference on tools with artificial intelligence***, 2000, pp. 2–10.

[62] ———, *ZRes: The old Davis–Putnam procedures meets ZBDDs*, ***Proceedings of the seventeenth international conference on automated deduction***, 2000, pp. 449–454.

[63] V. Chvátal, *Edmonds polytopes and a hierarchy of combinatorial problems*, ***Discrete Mathematics***, vol. 306 (2006), no. 10–11, pp. 886–904, First appeared in volume 4 of same journal in 1973.

[64] V. Chvátal and E. Szemerédi, *Many hard examples for resolution*, ***Journal of the ACM***, vol. 35 (1988), no. 4, pp. 759–768.

[65] E. Clarke, O. Grumberg, and D. Peled, ***Model checking***, MIT Press, 1999.

[66] M. Clegg, J. Edmonds, and R. Impagliazzo, *Using the Groebner basis algorithm to find proofs of unsatisfiability*, ***Proceedings of the twenty-eighth annual ACM symposium on the theory of computing***, 1996, pp. 174–183.

[67] P. Clote and A. Setzer, *On PHP, st-connectivity and odd-charged graphs*, ***Proof complexity and feasible mathematics***, DIMACS Series in Discrete Mathematics and Computer Science, vol. 39, American Mathematical Society, 1998, pp. 93–118.

[68] S. Cook, *Feasibly constructive proofs and the propositional calculus*, ***Proceedings of the seventh annual ACM symposium on the theory of computing***, 1975, pp. 107–116.

[69] S. Cook and R. Reckhow, *The relative efficiency of propositional proof systems*, ***The Journal of Symbolic Logic***, vol. 44 (1979), no. 1, pp. 36 – 50.

[70] S. Dash, *An exponential lower bound on the length of some classes of branch-and-cut proofs*, ***Mathematics of Operations Research***, vol. 30 (2005), no. 3, pp. 678–700.

[71] M. Davis and H. Putnam, *A computing procedure for quantification theory*, ***Journal of the ACM***, vol. 7 (1960), no. 1, pp. 201–215.

[72] H. Dixon and M. Ginsberg, *Combining satisfiability techniques from AI and OR*, ***The Knowledge Engineering Review***, vol. 15 (2000), no. 1, pp. 31–45.

[73] ———, *Inference methods for a pseudo-boolean satisfiability solver*, ***Proceedings of the eighteenth national conference on artificial intelligence***, 2002, pp. 635–640.

[74] H. Dixon, M. Ginsberg, D. Hofer, E. Luks, and A. Parkes, *Generalizing boolean satisfiability III: Implementation*, ***Journal of Artificial Intelligence Research***, vol. 23 (2005), pp. 441–531.

[75] H. Dixon, M. Ginsberg, E. Luks, and A. Parkes, *Generalizing boolean satisfiability II: Theory*, ***Journal of Artificial Intelligence Research***, vol. 22 (2004), pp. 481–534.

[76] H. Dixon, M. Ginsberg, and A. Parkes, *Generalizing boolean satisfiability I: Background and survey of existing work*, ***Journal of Artificial Intelligence Research***, vol. 21 (2004), pp. 193–243.

[77] N. Eén and N. Sörensson, *An extensible SAT-solver*, ***Proceedings of SAT 2003***, 2003.

[78] J. L. Esteban, N. Galesi, and J. Messner, *On the complexity of resolution with bounded conjunctions*, ***Theoretical Computer Science***, vol. 321 (2004), no. 2–3, pp. 347–370.

[79] J. L. Esteban and J. Torán, *Space bounds for resolution*, ***Information and Computation***, vol. 171 (2001), no. 1, pp. 84–97.

[80] U. Feige, *Relations between average case complexity and approximation complexity*, ***Proceedings of the thiry-fourth annual ACM symposium on theory of computing***, 2002, pp. 534–543.

[81] E. Friedgut, *Sharp thresholds of graph properties and the k-SAT problem*, ***Journal of the American Mathematical Society***, vol. 12 (1999), pp. 1017–1054.

[82] N. Galesi and M. Lauria, *Extending polynomial calculus to k-DNF resolution*, Technical Report 41, Electronic Colloquium on Computational Complexity, 2007.

[83] E. Goldberg and Y. Novikov, *Berkmin: A fast and robust SAT solver*, ***Proceedings of date 2002***, 2002.

[84] O. Goldreich and D. Zuckerman, *Another proof the $BPP \subseteq PH$ (and more)*, Technical Report 45, Electronic Colloquium on Computational Complexity, 1997.

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

[86] J. F. Groote, *Hiding propositional constants in BDDs*, ***Formal Methods in System Design: an International Journal***, vol. 8 (1996), no. 1, pp. 91–96.

[87] A. Haken, *The intractability of resolution*, ***Theoretical Computer Science***, vol. 39 (1985), no. 2-3, pp. 297–308.

[88] J. Håstad, *Almost optimal lower bounds for small depth circuits*, ***Advances in computing research***, vol. 5, JAI Press, 1989, pp. 143–170.

[89] P. Hertel and T. Pitassi, *An exponential time/space speedup for resolution*, Technical Report TR07-046, Electronic Colloquium on Computational Complexity, 2007.

[90] S. Hoory, N. Linial, and A. Wigderson, *Expander graphs and their applications*, ***Bulletin of the American Mathematical Society***, vol. 43 (2006), pp. 439–561.

[91] J. Huang and A. Darwiche, *Toward good elimination ordering for symbolic SAT solving*, ***Proceedings of the sixteenth IEEE conference on tools with artificial intelligence***, 2004, pp. 566–573.

[92] R. Impagliazzo, T. Pitassi, and A. Urquhart, *Upper and lower bounds for tree-like cutting planes proofs*, ***Ninth annual symposium on logic in computer science***, 1994, pp. 220–228.

[93] R. Impagliazzo, P. Pudlák, and J. Sgall, *Lower bounds for the polynomial calculus and the Groebner basis algorithm*, ***Computational Complexity***, vol. 8 (1999), no. 2, pp. 127–144.

[94] R. Impagliazzo and N. Segerlind, *Counting axioms do not polynomially simulate counting gates (extended abstract)*, ***Proceedings of the forty-second annual IEEE symposium on foundations of computer science***, 2001, pp. 200–209.

[95] ———, *Constant-depth Frege systems with counting axioms polynomially simulate Nullstellensatz refutations*, ***ACM Transactions on Computational Logic***, vol. 7 (2006), no. 2, pp. 199–218.

[96] D. Itsykson and A. Kojevnikov, *Lower bounds on static Lovász–Schrijver calculus proofs for Tseitin tautologies*, ***Zapiski Nauchnyh Seminarov POMI***, vol. 340 (2006), pp. 10–32, In Russian. Preliminary version in English appeared in ICALP 2005.

[97] V. Kabanets, *Derandomization: A brief overview*, ***Bulletin of the European Association for Theoretical Computer Science***, (2002), no. 76, pp. 88–103.

[98] J. Köbler, J. Messner, and J. Torán, *Optimal proof systems imply complete sets for promise classes*, ***Information and Computation***, vol. 184 (2003), pp. 71–92.

[99] J. Krajíček, *Lower bounds to the size of constant-depth propositional proofs*, ***The Journal of Symbolic Logic***, vol. 59 (1994), no. 1, pp. 73–86.

[100] J. Krajíček, ***Bounded arithmetic, propositional logic and complexity theory***, Cambridge University Press, 1995.

[101] J. Krajíček, *Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic*, ***The Journal of Symbolic Logic***, vol. 62 (1997), no. 2, pp. 457–486.

[102] J. Krajíček, *On the weak pigeonhole principle*, ***Fudamenta Mathematicae***, vol. 170 (2001), pp. 123–140.

[103] J. Krajíček, *An exponential lower bound for a constraint propagation proof system based on ordered binary decision diagrams*, Technical Report 7, Electronic Colloquium on Computational Complexity, 2007.

[104] J. Krajíček and P. Pudlák, *Propositional proof systems, the consistency of first-order theories, and the complexity of computations*, ***The Journal of Symbolic Logic***, vol. 54 (1989), no. 3, pp. 1063–1079.

[105] J. Krajíček and P. Pudlák, *Some consequences of cryptographical conjectures for S^1_2 and EF*, ***Information and Computation***, vol. 140 (1998), no. 1, pp. 82–94.

[106] J. Krajíček, P. Pudlák, and A. Woods, *An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle*, ***Random Structures and Algorithms***, vol. 7 (1995), no. 1, pp. 15–40.

[107] R. Lipton, *Model theoretic aspects of computational complexity*, ***Proceedings of the nineteenth annual IEEE symposium on foundations of computer science***, 1978, pp. 193–200.

[108] L. Lovász and A. Schrijver, *Cones of matrics and set-functions and* 0-1 *optimization*, ***SIAM Journal on Optimization***, vol. 1 (1991), no. 2, pp. 166–190.

[109] I. Lynce and J. Marques Silva, *An overview of backtrack search satisfiability algorithms*, ***Annals of Mathematics and Artificial Intelligence***, vol. 37 (2003), no. 3, pp. 307–326.

[110] A. Maciel, T. Pitassi, and A. Woods, *A new proof of the weak pigeonhole principle*, ***Journal of Computer and System Sciences***, vol. 64 (2002), no. 3, pp. 843–872.

[111] J. Marques-Silva and K. Sakallah, *GRASP a new search algorithm for satisfiability*, ***Proceedings of IEEE/ACM internation conference on computer-aided design***, 1996.

[112] D. Martin, G. Logemann, and D. Loveland, *A machine program for theorem proving*, ***Communications of the ACM***, vol. 5 (1962), no. 7, pp. 394–397.

[113] C. McDiarmid, *Concentration*, ***Probabilistic methods for algorithmic discrete mathematics*** (M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, editors), Algorithms and Combinatorics, vol. 16, Springer, 1998, pp. 195–248.

[114] K. McMillan, ***Symbolic model checking***, Ph.D. thesis, Carnegie Mellon, 1992.

[115] ———, *Interpolation and SAT-based model checking*, ***Proceedings of fifteenth internation conference on computer aided verification***, 2003, pp. 1–13.

[116] ———, *Applications of Craig interpolants in model checking*, ***Proceedings of eleventh international conference on tools and algorithms for construction and analysis of systems***, 2005, pp. 1–12.

[117] C. Meinel and T. Theobald, ***Algorithms and data structures in VLSI design***, Springer-Verlag, 1998.

[118] D. Mitchell, B. Selman, and H. Levesque, *Hard and easy distributions for SAT problems*, ***Proceedings of the tenth national conference on artificial intelligence***, 1992, pp. 459–465.

[119] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik, *Chaff: Engineering an efficient SAT solver*, ***Proceedings of the thirty-eighth design automation conference***, 2001, pp. 530–535.

[120] D. Motter and I. Markov, *A compressed breadth-first search for satisfiability*, ***Fourth international workshop on algorithms engineering and experiments (ALENEX)***, 2002, pp. 29–42.

[121] ———, *Overcoming resolution-based lower bounds for SAT solvers*, ***Eleventh IEEE/ACM workshop on logic and synthesis***, 2002, pp. 373–378.

[122] D. Motter, J. Roy, and I. Markov, *Resolution cannot polynomially simulate compressed-BFS*, ***Annals of Mathematics and Artificial Intelligence***, vol. 44 (2005), no. 1–2, pp. 121–156.

[123] D. Mundici, *A lower bound for the complexity of Craig's interpolants in sentential logic*, ***Archiv fur Math. Logik***, vol. 23 (1983), pp. 27–36.

[124] J. Nordström, *Narrow proofs may be spacious: Separating space and width in resolution*, ***Proceedings of the thirty-eighth annual ACM symposium on theory of computing***, 2006, pp. 507–516.

[125] G. Pan and M. Vardi, *Search vs. symbolic techniques in satisfiability solving*, ***The seventh international conference on theory and applications of satisfiability testing***, 2004.

[126] R. Parikh, *Existence and feasibility in arithmetic*, ***The Journal of Symbolic Logic***, vol. 36 (1971), no. 3, pp. 494–508.

[127] J. Paris and A. Wilkie, *Counting problems in bounded arithmetic*, ***Methods in mathematical logic***, Lecture Notes in Mathematics, no. 1130, Springer-Verlag, 1985, pp. 317–340.

[128] J. Paris, A. Wilkie, and A. Woods, *Provability of the pigeonhole principle and the existence of infinitely many primes*, ***The Journal of Symbolic Logic***, vol. 53 (1988), no. 4, pp. 1235–1244.

[129] T. Pitassi, *Algebraic propositional proof systems*, ***Descriptive complexity and finite models*** (Neil Immerman and Phokion G. Kolaitis, editors), DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, vol. 31, American Mathematical Society, 1997.

[130] T. Pitassi, P. Beame, and R. Impagliazzo, *Exponential lower bounds for the pigeonhole principle*, ***Computational Complexity***, vol. 3 (1993), pp. 97–140.

[131] T. Pitassi and R. Raz, *Regular resolution bounds for the weak pigeonhole principle*, ***Combinatorica***, vol. 24 (2004), no. 3, pp. 503–524.

[132] P. Pudlák, *Lower bounds for resolution and cutting planes proofs and monotone computations*, ***The Journal of Symbolic Logic***, vol. 62 (1997), no. 3, pp. 981–998.

[133] ———, *On the complexity of propositional calculus*, ***Sets and proofs: Invited papers from logic colloquium '97*** (S. Barry Cooper and J. Truss, editors), London Mathematical Society Lecture Note Series, vol. 258, Cambridge University Press, 1997, pp. 197–218.

[134] P. PUDLÁK and P. HAJÉK, ***Metamathematics of first-order arithmetic***, ASL Perspectives in Logic, Springer-Verlag, 1993.

[135] P. PUDLÁK and J. SGALL, *Algebraic models of computation and interpolation for algebraic proof systems*, ***Proof complexity and feasible arithmetics*** (S. Buss and P. Beame, editors), DIMACS Series in Discrete Mathematics, vol. 39, American Mathematical Society, 1998, pp. 279–295.

[136] P. RAGDE and A. WIGDERSON, *Linear-size constant-depth polylog-threshold circuits*, ***Information Processing Letters***, vol. 39 (1991), no. 3, pp. 143–146.

[137] R. RAZ, *Resolution lower bounds for the weak pigeonhole principle*, ***Journal of the ACM***, vol. 51 (2004), no. 2, pp. 115–138.

[138] A. RAZBOROV, *On the method of approximations*, ***Proceedings of the twenty first annual ACM symposium on theory of computing***, 1989, pp. 167–176.

[139] ———, *On provably disjoint NP pairs*, Technical Report 6, Electronic Colloquium on Computational Complexity, 1994.

[140] ———, *Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic*, ***Izvestiya of the Russian Academy of Science and Mathematics***, vol. 59 (1995), no. 1, pp. 201–224.

[141] ———, *Lower bounds for the polynomial calculus*, ***Computational Complexity***, vol. 7 (1998), no. 4, pp. 291–324.

[142] ———, *Improved resolution lower bounds for the weak functional pigeonhole principle*, ***Theoretical Computer Science***, vol. 303 (2001), no. 1, pp. 233–243.

[143] ———, *Improved resolution lower bounds for the weak pigeonhole principle*, Technical Report 55, Electronic Colloquium on Computational Complexity, 2001.

[144] ———, *Proof complexity of pigeonhole principles*, ***Proceedings of the the fifth international conference on developments in language theory***, Lecture Notes in Computer Science, no. 2295, 2001, pp. 100–116.

[145] ———, *Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution*, Submitted to the Journal of Glacial Refereeing. Manuscript available at `http://genesis.mi.ras.ru/~razborov/`, 2003.

[146] ———, *Resolution lower bounds for perfect matching principles*, ***Journal of Computer and System Sciences***, vol. 69 (2004), no. 1, pp. 3–27.

[147] S. RIIS, *Count(q) does not imply Count(p)*, ***Annals of Pure and Applied Logic***, vol. 90 (1997), no. 1–3, pp. 1–56.

[148] Z. SADOWSKI, *On an optimal quantified propositional proof system and a complete language for $NP \cap coNP$*, ***Proceedings of the eleventh international symposium on fundamentals of computing theory***, 1997, pp. 423–428.

[149] N. SEGERLIND, ***New separations in propositional proof complexity***, Ph.D. thesis, University of California, San Diego, August 2003.

[150] ———, *An exponential separation between Res(k) and Res($k+1$) for $k \leq \epsilon \log n$*, ***Information Processing Letters***, vol. 93 (2005), no. 4, pp. 185–190.

[151] ———, *Nearly-exponential size lower bounds for symbolic quantifier elimination algorithms and OBDD-based proofs of unsatisfiability*, Technical Report 9, Electronic Colloquium on Computational Complexity, 2007.

[152] N. SEGERLIND, S. BUSS, and R. IMPAGLIAZZO, *A switching lemma for small restrictions and lower bounds for k-DNF resolution*, ***SIAM Journal of Computing***, vol. 33 (2004), no. 5, pp. 1171–1200.

[153] S. SIMPSON, ***Subsystems of second-order arithmetic***, Perspectives in Mathematical Logic, Springer-Verlag, 1999.

[154] C. Sinz and A. Biere, *Extended resolution proofs for conjoining BDDs*, ***First international computer science symposium in Russia***, 2006, pp. 600–611.

[155] R. Smolensky, *Algebraic methods in the theory of lower bounds for Boolean circuit complexity*, ***Proceedings of the nineteenth annual ACM symposium on theory of computing***, 1987, pp. 77–82.

[156] M. Soltys and S. Cook, *The proof complexity of linear algebra*, ***Annals of Pure and Applied Logic***, vol. 130 (2004), no. 1–3, pp. 277–323.

[157] R. M. Stallman and G. J. Sussman, *Forward reasoning and dependency-directd backtracking in a system for computer-aided circuit analysis*, ***Artificial Intelligence***, vol. 9 (1977), pp. 135–196.

[158] J. Torán, *Space and width in propositional resolution*, ***Bulletin of the European Association of Theoretical Computer Science***, vol. 83 (2004), pp. 86–104.

[159] A. Urquhart, *The complexity of propositional proofs*, this Bulletin, vol. 1 (1995), no. 4, pp. 425–467.

[160] A. Woods, ***Some problems in logic and number theory, and their connections***, Ph.D. thesis, University of Manchester, 1981.

[161] C. Wrathall, *Rudimentary predicates and relative computation*, ***SIAM Journal on Computing***, vol. 7 (1978), no. 2, pp. 149–209.

DEPARTMENT OF COMPUTER SCIENCE
POST OFFICE BOX 751
PORTLAND STATE UNIVERISTY
PORTLAND, OR 97201, USA
E-mail: nsegerli@cs.pdx.edu