Short Proofs Are Narrow—Resolution Made Simple

ELI BEN-SASSON AND AVI WIGDERSON

Hebrew University, Jerusalem, Israel

Abstract. The *width* of a Resolution proof is defined to be the maximal number of literals in any clause of the proof. In this paper, we relate proof width to proof length (=size), in both general Resolution, and its tree-like variant. The following consequences of these relations reveal width as a crucial "resource" of Resolution proofs.

In one direction, the relations allow us to give *simple*, *unified* proofs for almost all known exponential lower bounds on size of resolution proofs, as well as several interesting new ones. They all follow from width lower bounds, and we show how these follow from natural expansion property of clauses of the input tautology.

In the other direction, the width-size relations naturally suggest a simple dynamic programming procedure for automated theorem proving—one which simply searches for small width proofs. This relation guarantees that the running time (and thus the size of the produced proof) is at most quasi-polynomial in the smallest tree-like proof. This algorithm is never much worse than any of the recursive automated provers (such as DLL) used in practice. In contrast, we present a family of tautologies on which it is exponentially faster.

Categories and Subject Descriptors: F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*proof theory*; I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving

General Terms: Algorithms, Theory

1. *Introduction*

The central task of Proof-Complexity theory is to prove nontrivial lower bounds on the length (=size) of proofs, for nontrivial propositional proof-systems. This is done for three interrelated reasons.

(1) By the famous theorem of Cook and Reckhow [1979] NP = Co-NP iff there exists a propositional proof system, which can prove every tautology τ in length polynomial in $|\tau|$. Thus, super-polynomial size lower bounds for stronger and stronger proof systems hopefully brings us somewhat closer to asserting NP $\neq$ Co-NP.

This research was supported by grant number 69/96 of the Israel Science Foundation, founded by the Israel Academy for Sciences and Humanities.

Authors' present addresses: E. Ben-Sasson, 150 Stockton Street, Princeton, NJ 08540; A. Wigderson, Institute of Computer Science, Hebrew University, Jerusalem, Israel, e-mail: avi@cs.huji.ac.il.

(2) Automated Theorem Proving is essential for various aspects of (mainly practical) computer science. It is usually implemented with simple propositional proof systems. Discovering hard tautologies sheds light on the possibilities and usefulness of various Automated Theorem Proving techniques.

(3) Simple propositional systems are nonuniform analogs of natural fragments of Peano Arithmetic, most notably the various Bounded Arithmetic systems, which capture in some sense "polynomial time reasoning." Thus, lower bounds in the former yields independence results in the latter. Stunning evidence of Razborov [1995] and Razborov and Rudich [1994] show that the reasoning Complexity theory applied so far for proving circuit lower bounds lies within these fragments. Thus, such lower bounds may clarify the limits of our (human, rather than automated) proof techniques.

Despite some very recent impressive successes in proving super-polynomial lower bounds on a variety of propositional proof systems, it is probably fair to say that we are still far from understanding the reasoning power of very simple ones.

The Resolution proof system, which is the focus of this paper, is perhaps the simplest nontrivial one. All assertions in this proof system are clauses (namely disjunction of literals). A tautology is represented by its negation—as a set of contradicting clauses. (This is always possible by the NP-completeness of SAT.) The proof (or refutation) uses a simple deduction rule to generate more clauses from these "axioms clauses" until a contradiction is reached in the form of the empty (FALSE) clause. Resolution forms the basis of many automated theorem proving procedures used in practice.

Being so simple and fundamental, Resolution was a natural system to attack. However, proving size lower bounds even for it turned out to be very challenging. The first super-polynomial lower bounds were presented by Tseitin [1968]. Only 20 years later did the first exponential lower bounds appear, in the seminal work of Haken [1985]. Other examples soon followed [Urquhart 1987; Chvátal and Szemerédi 1988], based on Haken's method of proof. All of them seemed to require a significant array of technical tools and calculations (most notably, random restrictions). Very basic questions regarding Resolution are still open, and attempts to resolve them and simplify existing proofs are part of the current line of research (e.g., Razborov et al. [1997], Buss and Pitassi [1997], and Beame et al. [2000]).

The main message of this paper is that Resolution is best studied when we focus on *width*. Like size, width is a "resource" of proofs we may want to minimize. It is defined to be the number of literals in the largest clause of the proof.

The main observation of this paper is a relation between these two fundamental resources:

—If a contradiction τ over n variables, has a tree-like refutation of *size* S_T, then it has a refutation of maximal *width* $\log_2 S_T$.

—If τ has a *general* resolution refutation of *size* S, then it has a refutation of maximal *width* $O(\sqrt{n \log S})$.

Both the notion of width and the relations above, gradually surfaced in previous papers and we merely make them explicit. Reading through the existing lower bound proofs, it is evident that wide clauses play a central role, with the

following logic: If a Resolution proof is short, then random restrictions will "kill" all wide clauses with high probability. But a separate argument shows that they still have to exist even in refutations of the restricted tautology. Thus, the proof has to be long. This notion was originally made explicit by Galil [1977] that used linear lower bounds on width for proving exponential lower bounds on the Restricted Width algorithm described in Section 8.

Moreover, identical functional relations as those we obtain for width vs. size, appear in Clegg et al. [1996] between *degree* in Polynomial Calculus proofs, and size of Resolution proofs, and we simply observe that their argument actually applies to give these stronger relations. That paper, and the subsequent [Beame and Pitassi 1996], were the main inspiration for our work.

The first major application of our explicit width-size relations is significant simplification and unification of almost all known exponential lower bounds on Resolution proof length. Naturally, this understanding leads to new lower bounds as well. The main point is that now, to prove size lower bounds, it is sufficient to prove width lower bounds. It removes the need for random restrictions, and allows us to concentrate on the original tautology rather than restricted forms of it.

We develop a general strategy for proving width lower bounds, which follows Haken's original proof technique but for the above reason is now simple and clear. It reveals that large width is implied by certain natural expansion properties of the clauses (axioms) of the tautology in question. We show that in the classical examples of the Pigeonhole Principle, Tseitin graph tautologies, and random k-CNF's, these expansion properties are quite simple to prove (indeed, they comprised in some implicit way the simple part of the existing lower bound proofs).

We further illustrate the power of this approach by proving new exponential lower bounds to two different restricted versions of the pigeon-hole principle. One restriction allows the encoding of the principle to use arbitrarily many extension variables in a structured way (completely unstructured extension variables make the proof system as strong as Extended Frege, for which no lower bounds are known). The second restriction allows every pigeon to choose a hole from some constant size set of holes.

The second major application of our relations is in automatization results for the Resolution proof system. This is the basic problem faced in the analysis of automatic provers searching for a proof; how long will they run, as a function of the shortest existing proof of the input tautology.

The relations suggest the use of the following simple (dynamic programming) algorithm: Set $i = 1$. Start with the axioms, and try to derive all clauses of width at most i. If the empty clause is derived, we are done. If not, increase i by 1 and repeat.

This algorithm has been originally suggested and investigated by Galil [1977] who proved exponential lower bounds for natural inputs, using linear lower bounds on width.

Clearly, the running time on any tautology τ over n variables is at most $n^{O(w)}$, when w is the minimal width of a proof of τ. By the relations above, this time is at most $S_T(\tau)^{O(\log n)}$ (namely, quasi-polynomial in the minimal tree-like Resolution proof length), and at most $\exp(\sqrt{n \log S(\tau)})$ (namely, subexponential in the minimal general Resolution proof length). These bounds were already obtained

by Beame and Pitassi [1996] who adopted the Clegg et al. [1996] algorithm from the Polynomial Calculus system to resolution.

Note that the relation to tree-like proofs is of particular importance, due to the fact that the most popular automated provers such as Davis et al. [1962] produce tree-like Resolution proofs. Thus, our algorithm never runs much longer than these provers on any tautology. Our final contribution, described below, is a new collection of natural tautologies, for which our algorithm is exponentially faster than such provers.

We give a construction which associates to every directed graph G on n edges a tautology $\tau(G)$ with the following properties.

(1) $\tau(G)$ has $O(n)$ variables and $O(n)$ clauses.
(2) $\tau(G)$ has general Resolution proofs of length $O(n)$.
(3) $\tau(G)$ has Resolution proofs of width $O(1)$.
(4) Every tree-like Resolution proof of $\tau(G)$ has length at least $\exp(P(G))$, where $P(G)$ is the classical pebbling number of the graph G.

The construction itself is motivated by a special case of it for the Pyramid graph of Bonet et al. [2000], which was in turn motivated by Raz and McKenzie [2000]. However, the simple adaptation of the pebbling bound into a tree-like Resolution lower bound for the Pyramid does not directly extend to arbitrary graphs, and we believe that the general connection in (4) is of independent interest. The proofs of the main claims are presented in Ben-Sasson et al. [2000].

At any rate, this construction allows us to use much harder graphs to pebble than Pyramids. Specifically, Celoni et al. [1977] explicitly construct for every n a graph G_n of size $O(n)$ satisfying $P(G_n) = \Omega(n/\log n)$. For these graphs, (4) implies exponential runtime for any tree-like automated prover, while (3) guarantees polynomial runtime for our dynamic programming algorithm, giving the desired exponential separation of the two automated proof search methods.

The paper is organized as follows: Section 3 states and proves the Size-Width relations. Section 4 summarizes the lower bounds on width for Tseitin formulas, random k-CNF's and the Pigeonhole Principle and its variants. Section 5 presents the general strategy for proving width lower bounds, and Section 6 applies the strategy to obtain the main results presented in Section 4. Section 7 discusses the tightness of the trade-off. Finally, Section 8 discusses the efficiency of the Automated Theorem Proving algorithm based on the Size-Width trade-off.

2. *Definitions*

2.1. GENERAL. x will denote a Boolean variable, ranging over $\{0, 1\}$. Throughout this paper we shall identify 1 with *True* and 0 with *False*. A *literal* over x is either x (denoted also as x^1) or $\bar{x}$ (denoted also as x^0). A *clause* is a disjunction of literals. We say that a variable x *appears* in C (denoted $x \in C$) if a literal over x appears in C. A CNF formula is a conjunction of clauses. Let $\mathcal{F} = \{C_1, C_2 \ldots C_m\}$ be a CNF formula over n variables, a Resolution *derivation* π of a clause A from $\mathcal{F}$ is a sequence of clauses $\pi = \{D_1, D_2 \ldots D_S\}$ such that the last clause is A and each line D_i is either some initial clause $C_j \in \mathcal{F}$ or is derived from the previous lines using one of the following derivation rules:

(1) The Resolution Rule:

$$\frac{E \vee x \; F \vee \bar{x}}{E \vee F}$$

(2) The Weakening Rule:

$$\frac{E}{E \vee F}$$

where $x \in \{x_1, x_2, \ldots, x_n\}$ and E, F are arbitrary clauses. The Weakening Rule is not essential, as even without it the Resolution proof system is complete with respect to refutations, but we add it for the sake of simplicity. A resolution *refutation* is a resolution derivation of the empty clause 0. The graph G_π of a derivation π is a DAG with the clauses of the derivation as nodes, and for derivation step edges are added from the assumption clauses to the consequence clause. A derivation π is called *tree-like* if G_π is a tree; we may make copies of the original clauses in $\mathcal{F}$ in order to make a proof tree-like. The *size* of the derivation π is the number of lines (clauses) in it, denoted S_π. $S(\mathcal{F})$ ($S_T(\mathcal{F})$) is the *minimal size* of a (Tree-like) refutation of $\mathcal{F}$.

2.2. RESTRICTIONS. For C a clause, x a variable and $a \in \{0, 1\}$, the *restriction* of x on a is:

$$C|_{x=a} \stackrel{\text{def}}{=} \begin{cases} C & \text{if } \; x \text{ does not appear in } C \\ 1 & \text{if } \; \text{the } \textit{literal } x^a \text{ appears in } C \\ C \backslash \{x^{1-a}\} & \text{otherwise.} \end{cases}$$

Similarly, $\mathcal{F}|_{x=a} \stackrel{\text{def}}{=} \{C|_{x=a} : C \in \mathcal{F}\}$. For $\pi = \{C_1, \ldots C_s\}$ a derivation of C_s from $\mathcal{F}$ and $a \in \{0, 1\}$, let $\pi|_{x=a} = \{C'_1, \ldots C'_s\}$ be the restriction of π on $x = a$, defined inductively by:

$$C'_i = \begin{cases} C_i|_{x=a} & C_i \in \mathcal{F} \\ C'_{j_1} \vee C'_{j_2} & C_i \text{ was derived from } C_{j_1} \vee y \text{ and} \\ & C_{j_2} \vee \bar{y} \text{ via a resolution step,} \\ & \text{for } j_1 < j_2 < i \\ C'_j \vee A|_{x=a} & C_i = C_j \vee A \text{ via the weakening rule,} \\ & \text{for } j < i \end{cases}$$

The consequence of resolving a clause B with 1 is defined to be B. We shall assume without loss of generality that $\pi|_{x=a}$ does not contain the clause 1, by removing all such clauses from $\pi|_{x=a}$.

2.3. WIDTH. The *width* of a clause C, denoted $w(C)$, is defined to be the number of literals appearing in it. The *width* of a set of clauses is the *maximal width* of a clause in the set, that is, $w(\mathcal{F}) = \max_{C \in \mathcal{F}} \{w(C)\}$. In most cases, input tautologies $\mathcal{F}$ will have $w(\mathcal{F}) = O(1)$.

The width of *deriving* a clause A from the formula $\mathcal{F}$, denoted $w(\mathcal{F} \vdash A)$ is defined by $\min_\pi\{w(\pi)\}$ where the minimum is taken over all derivations π of A from $\mathcal{F}$. We also use the notation $\mathcal{F} \vdash_w A$ to mean that A can be derived from $\mathcal{F}$

in width w. We will be mainly interested in the width of refutations, namely in $w(\mathcal{F} \vdash 0)$.

3. *The Size-Width Relations*

The following lemmata and theorems are a direct translation of Clegg et al. [1996] to Resolution derivations. Stated informally, they say that if $\mathcal{F}$ has a *short* resolution refutation then it has a refutation with *small width*.

LEMMA 3.1. *For $a \in \{0, 1\}$, if $\mathcal{F}|_{x=a} \vdash_w A$ then $\mathcal{F} \vdash_{w+1} A \vee x^{1-a}$.*

PROOF. $\mathcal{F}|_{x=a}$ is created from $\mathcal{F}$ by disposing of all initial clauses that include the literal x^a and removing the literal x^{1-a} from all other initial clauses where it appears. Let $\mathcal{F}'$ be the set of initial clauses containing the literal x^{1-a}, and let π be a width w derivation of A from $\mathcal{F}_{x=a}$. Add the literal x^{1-a} to all clauses in π and call the new derivation π'. We claim that π' is a legal resolution derivation. If $C \in \mathcal{F}'$ then $C \vee x^{1-a}$ is an initial clause of $\mathcal{F}$. If $C \in \mathcal{F} \backslash \mathcal{F}'$ then $C \vee x^{1-a}$ can be derived from C by a single weakening step. Finally, if C was derived from A, B via a resolution step, then $C \vee x^{1-a}$ is the resolution consequence of $A \vee x^{1-a}$, $B \vee x^{1-a}$. It is easy to see that the width of each clause in π' is larger by 1 than the matching clause in π. □

LEMMA 3.2. *For $a \in \{0, 1\}$, define $\mathcal{F}_{x^a}$ as the set of all clauses in $\mathcal{F}$ containing the literal x^a. If $\mathcal{F}|_{x=a} \vdash_{k-1} 0$ and $\mathcal{F}|_{x=1-a} \vdash_k 0$, then $w(\mathcal{F} \vdash 0) \leq max\{k, w(\mathcal{F}_{x^a})\}$.*

PROOF. According to Lemma 3.1, if $\mathcal{F}|_{x=a} \vdash_{k-1} 0$, then $\mathcal{F} \vdash_k x^{1-a}$. We now resolve the clause x^{1-a} with all clauses in $\mathcal{F}_{x^a}$ and derive $\mathcal{F}|_{x=1-a}$. This part will have width $w(\mathcal{F}_{x^a})$. Finally, by the assumption, we can refute $\mathcal{F}|_{x=1-a}$ with width k. □

THEOREM 3.3. $w(\mathcal{F} \vdash 0) \leq w(\mathcal{F}) + log\, S_T(\mathcal{F})$.

PROOF. We prove by induction on b and n, the number of variables, that if $S_T(\mathcal{F}) \leq 2^b$ then $w(\mathcal{F} \vdash 0) \leq w(\mathcal{F}) + b$. If $b = 0$, then $0 \in \mathcal{F}$, and we're done. Otherwise, the last derivation is

$$\frac{x \quad \bar{x}}{0},$$

where x, $\bar{x}$ were derived by tree-like derivations T_x, $T_{\bar{x}}$ of sizes S_x, $S_{\bar{x}}$, respectively, and $S_T = S_x + S_{\bar{x}} + 1$. $T_x|_{x=0}$ ($T_x|_{x=1}$) is a tree-like refutation of $\mathcal{F}|_{x=0}$ ($\mathcal{F}|_{x=1}$) of size at most S_x ($S_{\bar{x}}$). Assume without loss of generality, $S_x \leq 2^{b-1}$. By induction on b, $w(\mathcal{F}|_{x=0} \vdash 0) \leq w(\mathcal{F}) + b - 1$ and by induction on n, $w(\mathcal{F}|_{x=1} \vdash 0) \leq w(\mathcal{F}) + b$. Applying Lemma 3.2 completes the proof. □

COROLLARY 3.4. $S_T(\mathcal{F}) \geq 2^{(w(\mathcal{F} \vdash 0) - w(\mathcal{F}))}$

THEOREM 3.5. $w(\mathcal{F} \vdash 0) \leq w(\mathcal{F}) + O(\sqrt{n\, ln\, S(\mathcal{F})})$.

PROOF. Let π be a minimal size refutation of $\mathcal{F}$, of size S. Let $k = w(\mathcal{F})$. If $S = 1$, then $0 \in \mathcal{F}$, and we're done. Otherwise, set $d \stackrel{\text{def}}{=} \lceil \sqrt{2n \ln S(\mathcal{F})} \rceil$, and $a \stackrel{\text{def}}{=} (1 - (d/2n))^{-1}$. Let π^* be the set of *fat* clauses in π, having width greater than d. We prove by induction on b and n, that if $|\pi^*| < a^b$ then $w(\mathcal{F} \vdash 0) \leq$

$d + k + b$. The base case ($b = 0$) is trivially true. For the induction step, there are $2n$ literals, so by the Pigeonhole Principle there is some literal (without loss of generality) x appearing in $\geq d/2n \cdot |\pi^*|$ fat clauses. Restricting $x = 1$ removes all the clauses in which x appears, and leaves us with a refutation of $\mathcal{F}|_{x=1}$ with at most $(1 - (d/2n)) \cdot |\pi^*| < a^{b-1}$ fat clauses. $\mathcal{F}|_{x=1} \vdash_{d+k+b-1} 0$, by induction on b, $\mathcal{F}|_{x=0} \vdash_{d+k+b} 0$, by induction on n. Applying Lemma 3.2 completes the proof. □

COROLLARY 3.6.

$$S(\mathcal{F}) = exp(\Omega(w(\mathcal{F} \vdash 0) - w(\mathcal{F}))^2 n$$

If $w(\mathcal{F}) \sim \#Variables$, Corollaries 3.4 and 3.6 are useless for obtaining lower bounds. This will not be a problem in practice, since all the formulas we shall consider, either have constant initial width (i.e., Tseitin formulas, random k-CNF's), or can be reduced to such formulas (i.e., the Pigeonhole Principle).

4. *Results*

Over the past 30 years, several exponential lower bounds on size have been obtained, for several different contradictions. We wish to provide new and simple lower bounds for most of these tautologies, and we organize this section as follows. We start each subsection with a definition of the contradiction, and discuss its initial size and width. This is followed by a lower bound on the width of the refutation. Exponential lower bounds on size of refutations, via Corollaries 3.4 and 3.6, conclude the discussion.

4.1. TSEITIN FORMULAS

4.1.1. *Definition*. A Tseitin contradiction is an unsatisfiable CNF capturing the basic combinatorial principle that for every graph, the sum of degrees of all vertices is even.

Definition 4.1 (*Tseitin Formulas*). Fix G a finite connected graph, with $|V(G)| = n$. $f: V(G) \to \{0, 1\}$ is said to have *odd-weight* if $\Sigma_{v \in V(G)} f(v) \equiv 1 \pmod 2$. Denote by $d_G(v)$ the degree of v in G (i.e., the number of edges incident with v). Fix f an odd-weight function. Assign a distinct variable x_e to each edge $e \in E(G)$. For $v \in V(G)$, define $PARITY_v \stackrel{\text{def}}{=} (\oplus_{v \in e} x_e \equiv f(v) \pmod 2)$. The Tseitin Contradiction of G and f is:

$$\tau(G, f) = \bigwedge_{v \in V(G)} PARITY_v.$$

If the maximal degree of G is constant, then the initial size and width of $\tau(G, f)$ is small as well:

LEMMA 4.2. *If d is the maximal degree of G, then $\tau(G, f)$ is a d-CNF with at most $n \cdot 2^{d-1}$ clauses, and $nd/2$ variables.*

4.1.2. *Width Lower Bound*. The width of refuting $\tau(G, f)$ is bounded from below by the *expansion* of the graph G, hereby defined.

Definition 4.3 (*Expansion*). For G a finite connected graph, the *Expansion* of G is

$$e(G) \stackrel{\text{def}}{=} \min\{|E(V', V\backslash V')|: V' \subseteq V,$$
$$|V|/3 \leq |V'| \leq 2|V|/3\}$$

The main claim of this section is (for a proof, see Section 6.1):

THEOREM 4.4. *For G a connected graph and f an odd-weight function on $V(G)$, $w(\tau(G,f) \vdash 0) \geq e(G)$.*

COROLLARY 4.5 [URQUHART 1987]. *For G a 3-regular connected Expander (i.e., $e(G) = \Omega(|V|)$), and f an odd-weight function on $V(G)$, $S(\tau(G,f)) = 2^{\Omega(|\tau(G,f)|)}$.*

4.2. THE PIGEONHOLE PRINCIPLE

4.2.1. *Definition*. The Pigeonhole Principle with m pigeons and n pigeonholes, states that there is no 1-1 map from m to n, as long as $m > n$. This can be stated by a formula on $n \cdot m$ variables x_{ij}, $1 \leq i \leq m$, $1 \leq j \leq n$, where $x_{ij} = 1$ means that i is mapped to j.

Definition 4.6 (PHP_n^m). PHP_n^m is the conjunction of the following set of clauses:

$$P_i \stackrel{\text{def}}{=} \bigvee_{1 \leq j \leq n} x_{ij} \text{ for } 1 \leq i \leq m.$$

$$H_{i,i'}^j \stackrel{\text{def}}{=} \bar{x}_{ij} \vee \bar{x}_{i'j} \text{ for } 1 \leq i < i' \leq m, 1 \leq j \leq n.$$

Whenever $m > n$, PHP_n^m is an unsatisfiable CNF with $m \cdot n > n^2$ variables, $O(m^2)$ Clauses and initial width n.

The large initial width of PHP_n^m and its large number of variables does not allow the derivation of size lower bounds directly from width lower bounds. In the next two subsections, we generalize these pigeonhole tautologies in two different ways—the first reducing the width using extension variables, and the second reducing the number of variables but fixing most of them. The lower bounds for both are new and of independent interest.

4.2.2. *Width Lower Bound for Tree-Like Resolution*. It is easy to check that $w(PHP_n^m \vdash 0) \leq n$. This means that using the standard formulation of PHP_n^m one cannot achieve a lower bound on size via the Size-Width trade-off. Therefore, we shall need to reduce PHP_n^m to a *constant width* analog, and prove $\Omega(n)$ lower bounds on width for this formulation.

Definition 4.7 ($EPHP_n^m$). For $f(\vec{x})$ a Boolean function, a *Nondeterministic Extension* of f is a function $g(\vec{x}, \vec{y})$ such that $f(\vec{x}) = 1$ iff $\exists \vec{y} g(\vec{x}, \vec{y}) = 1$. The $\vec{x}$ variables are called *Original* variables, and the $\vec{y}$ are called *Extension* variables.

A *Row-Extension* of PHP_n^m, denoted $EPHP_n^m$ is derived by replacing every row axiom P_i with some nondeterministic extension CNF formula EP_i, using distinct extension variables $\vec{y}_i$ for distinct rows.

One standard extension is:

Example 4.8. Replace every P_i with the following 3-CNF over $n + 2$ clauses and $2n + 1$ variables:

$$\bar{y}_{i0} \wedge \bigwedge_{j=1}^{n}(y_{ij-1} \vee x_{ij} \vee \bar{y}_{ij}) \wedge y_{in}.$$

The main claim of this section is (for a proof of the following theorem and Corollary 4.10, see Section 6.2):

THEOREM 4.9. *For $m > n$, $w(EPHP_n^m \vdash 0) \geq n/3$.*

COROLLARY 4.10. *For all $m > n$ and any Row Extension of PHP_n^m, $S_T(EPHP_n^m) = 2^{\Omega(n)}$.*

As PHP_n^m is a legitimate Row Extension of PHP_n^m, we get:

COROLLARY 4.11 [BUSS AND PITASSI 1997]. *For all $m > n$, $S_T(PHP_n^m) = 2^{\Omega(n)}$*

4.2.3. *Width Lower Bound for General Resolution*. The number of variables of PHP_n^m is quadratic in the width of its refutation. Hence, in order to apply Corollary 3.6 and derive exponential lower bounds on size for *general* resolution, we shall need the following generalization, which restricts PHP_n^m by limiting the number of holes into which a pigeon may enter. This will decrease the number of underlying variables, while leaving the width lower bound in place.

Definition 4.12 ($G - PHP$). Let $G = ((V \cup U), E)$ be a bipartite graph, $|V| = m$, $|U| = n$. Assign each edge a distinct variable x_e. $G - PHP$ is the conjunction of the following clauses:

$$P_v \quad \stackrel{\text{def}}{=} \quad \bigvee_{v \in e} x_e \text{ for } v \in V.$$

$$H^u_{v,v'} \quad \stackrel{\text{def}}{=} \quad \bar{x}_e \vee \bar{x}_{e'} \text{ for } e = (v, u), e' = (v', u), v, v' \in V, \; v \neq v', u \in U.$$

$G - PHP$ is a natural generalization of the Pigeonhole Principle, because $PHP_n^m = K_{m,n} - PHP$. This generalization will prove useful by the following observation, presented here without a proof:

LEMMA 4.13. *For any two bipartite graphs G, G' over the same vertex set, if $E(G') \subseteq E(G)$, then $S(G' - PHP) \leq S(G - PHP)$.*

The width of refuting $G - PHP$ is bounded from below by the expansion of G, for the following bipartite version of expansion (we define it already with the parameters that will yield linear width, but this can be generalized):

Definition 4.14 (*Bipartite Expansion*). For a vertex $u \in U$, let $N(u)$ be its set of neighbors. For a subset $V' \subset V$ let its *boundary* be

$$\partial V' \stackrel{\text{def}}{=} \{u \in U : |N(u) \cap V'| = 1\}.$$

In words, the boundary of V' is the set of vertices in U that have exactly one neighbor in V'.

We call a bipartite graph G an (m, n, d, r, e)-expander if $|V| = m$, $|U| = n$, the degree of vertices in V is at most d, and for all $V' \subset V$, $|V'| \leq r \; |\partial V'| \geq e|V'|$ (in words: every small set has large boundary).

The main theorem of this section is (for a proof, see Section 6.2):

THEOREM 4.15. *For every bipartite graph G that is an (m, n, d, r, e)-expander, $w(G - PHP \vdash 0) \geq (r \cdot e)/2$.*

For $m = n + 1$, a simple union bound calculation shows that there exist bipartite graphs which are $(m, n, 5, n/c, 1)$-expanders, for some constant $c > 1$, yielding:

COROLLARY 4.16 [HAKEN 1985]. $S(PHP_n^{n+1}) = 2^{\Omega(n)}$

The best-known lower bound for the Weak Pigeonhole Principle (i.e., when m is much larger than n) is $S(PHP_n^m) = 2^{\Omega(n^2/m)}$ [Buss and Turán 1988]. When $m \gg n$, the same simple calculation shows that there exists a $(m, n, \log m, \Omega(n/\log m), \frac{3}{4}\log m)$-expander, yielding a slightly less than optimal lower bound:

COROLLARY 4.17 [BUSS AND TURAN 1988]. $S(PHP_n^m) = 2^{\Omega(n^2/m \log m)}$

4.3. RANDOM k-CNF'S

4.3.1. *Definition*. We start with a formal definition of a random k-CNF:

Definition 4.18 (*Random k-CNF's*). Let $\mathcal{F} \sim \mathcal{F}_k^{n,\Delta}$ denote that $\mathcal{F}$ is a random k-CNF formula on n variables and $m = \Delta \cdot n$ clauses, chosen at random by picking $\Delta \cdot n$ clauses i.i.d from the set of all $\binom{n}{k} \cdot 2^k$ clauses, with repetitions. Δ is called the *clause density*.

In their seminal paper, Chvátal and Szemerédi [1988] showed that for $k \geq 3$ whp a random k-CNF formula with *constant* clause density $\Delta \geq \ln 2 \cdot 2^k$, is unsatisfiable and requires an exponential length refutation.

4.3.2. *Width Lower Bounds*. For the sake of simplicity, we shall work only with $k = 3$, and these bounds can be easily extended to any fixed k (see, e.g., Beame et al. [2000]). The following is proven by use of a straightforward Union bound (for a proof, see Section 6.3):

THEOREM 4.19. *For any $0 \leq \epsilon \leq 1/2$, if $\Delta = n^{(1/2)-\epsilon}$ and $\mathcal{F} \sim \mathcal{F}_3^{n,\Delta}$, then whp*

$$w(\mathcal{F} \vdash 0) = \Omega(\Delta^{-2/(1-\epsilon)} \cdot n).$$

Using this *width* lower bound, we easily obtain the best currently known lower bounds on *size*:

COROLLARY 4.20 [BEAME ET AL. 2000]. *For any $0 \leq \epsilon \leq 1/2$, if $\Delta = n^{(1/2)-\epsilon}$ and $\mathcal{F} \sim \mathcal{F}_3^{n,\Delta}$, then whp*

$$S(\mathcal{F}) = exp(\Omega(\Delta^{-4/(1-\epsilon)} \cdot n)).$$

5. *Proof Strategy*

All lower bounds on width follow the same strategy:

(1) Define a complexity measure μ: $\{Clauses\} \to N$ such that $\mu(Axiom) \leq 1$.
(2) Prove $\mu(0)$ is *large*.
(3) Infer that in *any* refutation there is some clause C with *medium* size $\mu(C)$.
(4) Prove that if $\mu(C)$ is *medium*, then $w(C)$ is *large*.

We shall now formalize and explain this strategy. First, we will define a measure that will satisfy conditions (1)–(3).

Definition 5.1. ($\mu_{\mathcal{A}}$) for f a Boolean function, let $Vars(f)$ denote the set of variables appearing in f. Let $\alpha \in \{0,1\}^{Vars(f)}$ be an assignment to f. We say that α *satisfies* f, if $f(\alpha) = 1$. For C a clause and Γ a set of Boolean functions, let $V = Vars(\Gamma) \cup Vars(C)$. We say that Γ *implies* C, denoted $\Gamma \models C$, if every assignment satisfying every function $\gamma \in \Gamma$ satisfies C as well.

Let $\mathcal{A}$ be an unsatisfiable set of Boolean functions, that is, $\mathcal{A} \models 0$, and let C be a clause.

$$\mu_{\mathcal{A}}(C) \stackrel{\text{def}}{=} \min\{|\mathcal{A}'|: \mathcal{A}' \subseteq \mathcal{A}, \mathcal{A} \models C\}.$$

$\mu_{\mathcal{A}}$ is a subadditive complexity measure with respect to resolution steps:

LEMMA 5.2. *Suppose D was inferred from B, C by a single resolution step. Then for any set of Boolean functions $\mathcal{A}$:*

$$\mu_{\mathcal{A}}(D) \leq \mu_{\mathcal{A}}(B) + \mu_{\mathcal{A}}(C).$$

In order to assure Condition 1 of the strategy, we want $\mu(Axiom)$ to be small:

Definition 5.3 (*Compatibility*). For $\mathcal{F}$ a nonsatisfiable CNF we say that $\mathcal{A}$ *is compatible* with $\mathcal{F}$ if $\mathcal{A} \models 0$ and $\forall C \in \mathcal{F}\ \mu(C) \leq 1$.

We will always pick a compatible $\mathcal{A}$ and use it to define $\mu = \mu_{\mathcal{A}}$. Note that part 2 of the strategy puts another requirement on $\mathcal{A}$, namely that no "small" subset of it is contradictory. However, this would be intuitively easy to achieve with "hard" tautologies.

We now claim that part 3 is deduced from the definitions:

LEMMA 5.4. *If $\mathcal{A}$ is compatible with $\mathcal{F}$, then in every refutation of $\mathcal{F}$ there must be a clause C with*

$$\frac{\mu(0)}{3} \leq \mu(C) \leq \frac{2\mu(0)}{3}.$$

The rest of the section is devoted to proving the connection between Condition 4 and the expansion properties of a set of *sensitive* functions which is *compatible* with the input formula:

Definition 5.5 (*Sensitivity*). A Boolean function f is called *Sensitive* if any two distinct falsifying assignments $\alpha, \beta \in f^{-1}(0)$, have Hamming distance greater than 1. Examples of Sensitive functions are *PARITY* and *OR*.

For $\mathcal{A}$ a set of Boolean functions, and $f \in \mathcal{A}$, a *Critical Assignment* for f is an assignment $\alpha \in \{0,1\}^{Vars(\mathcal{A})}$ such that

$$g(\alpha) = \begin{cases} 0 & g = f \\ 1 & g \neq f, g \in \mathcal{A} \end{cases}$$

For $\alpha, \beta \in \{0,1\}^{Vars(\mathcal{A})}$, we say that β is the result of *flipping* α on the variable x, if

$$\beta(y) = \begin{cases} 1 - \alpha(y) & y = x \\ \alpha(y) & \text{otherwise} \end{cases}$$

We shall define the *expansion* of a CNF formula in terms of its minimal boundary:

Definition 5.6 (*Boundary*). For f a Boolean function and x a variable, we say that f is *dependent* on x if there is some assignment α such that $f(\alpha) = 0$, but flipping α on x *satisfies* f.

For $\mathcal{A}$ a set of Boolean functions, the *Boundary* of $\mathcal{A}$, denoted $\partial\mathcal{A}$, is the set of variables x such that there is a *unique* function $f \in \mathcal{A}$ that is dependent on x.

A *critical assignment* to a *sensitive* function can be easily changed to a satisfying assignment, by *flipping* a *boundary* variable. Formally:

LEMMA 5.7. *If $f \in \mathcal{A}$ is Sensitive, α is a Critical Assignment for f, and $x \in Vars(f) \cap \partial\mathcal{A}$, then flipping α on x yields an assignment β that satisfies $\mathcal{A}$.*

We define the expansion of $\mathcal{F}$ to be the minimal boundary of a medium size subformula of $\mathcal{F}$:

Definition 5.8 (*Expansion*). For $\mathcal{A} \models 0$, let $k = \mu_{\mathcal{A}}(0)$. Define the *Expansion* of $\mathcal{A}$ to be:

$$e(\mathcal{A}) \stackrel{\text{def}}{=} \min\left\{ \left| \partial\mathcal{A}' \right| : \mathcal{A}' \subset \mathcal{A}\ \frac{1}{3} \cdot k \leq \left| \mathcal{A}' \right| \leq \frac{2}{3} \cdot k \right\}.$$

The main tool, used in proving most lower bounds on width, presents the connection between *width* and *expansion*:

THEOREM 5.9. *For $\mathcal{F}$ an unsatisfiable CNF,*

$$w(\mathcal{F} \vdash 0) \geq \max e(\mathcal{A}),$$

where the maximum is taken over all sets $\mathcal{A}$ of sensitive functions, compatible with $\mathcal{F}$.

PROOF. Fix some $\mathcal{A}$ that is compatible with $\mathcal{F}$, and let $\mu_{\mathcal{A}}(0) = k$. By Lemma 5.4, there must exist some clause C such that $k/3 \leq \mu_{\mathcal{A}}(C) \leq 2k/3$. Let $\mathcal{A}' \subset \mathcal{A}$ be a minimal set such that $\mathcal{A}' \models C$. We claim that any variable $x \in \partial\mathcal{A}'$ must appear in C. To see this, notice that, for every $f \in \mathcal{A}'$, there is some assignment α_f such that $\alpha_f(C) = \alpha_f(f) = 0$ and $\alpha_f(g) = 1$ for all $g \in \mathcal{A}'$ $g \neq f$. This follows from the minimality of $\mathcal{A}'$, for otherwise $\mathcal{A}' \backslash f \models C$. Suppose, for the sake of contradiction, that $x \in \partial\mathcal{A} \cap Vars(f)$ but $x \notin C$. By Lemma 5.7, flipping α on x satisfies $\mathcal{A}'$, but the new assignment agrees with α on $Vars(C)$. Hence, $\mathcal{A}' \not\models C$, contradiction. □

6. *Proof of Main Results*

6.1. TSEITIN FORMULAS. We start with a proof of Theorem 4.4, to illustrate the simplicity of the strategy. We shall need the following lemma from Urquhart [1995]:

LEMMA 6.1 [URQUHART 1995]. *If G is connected, then $\tau(G, f)$ is contradictory iff f is an odd weight function.*

PROOF (THEOREM 4.4). We use the notation of Section 5. Set $\mathcal{A}_V = \{PARITY_v : v \in V(G)\}$ and denote $\mu(C) = \mu_{\mathcal{A}_V}(C)$. Every axiom C is one of the defining axioms of $PARITY_v$. Clearly, for this very same v $PARITY_v \models C$. Hence for any axiom C, $\mu(C) = 1$. So far we have shown that $\mathcal{A}_V$ is compatible for $\tau(G, f)$. Next we claim that $\mu(0) = |V(G)|$, because for any $|V'| < |V(G)|$ $\mathcal{A}_{V'}$ is satisfiable. This latter claim is seen by the following reasoning: Let v be some vertex in $V \backslash V'$. Look at the formula $\tau(G, f')$ for

$$f'(u) = \begin{cases} 1 - f(u) & u = v \\ f(u) & \text{otherwise} \end{cases}$$

By Lemma 6.1, $\tau(G, f')$ is satisfiable. $\mathcal{A}_{V'}$ is a subformula of $\tau(G, f')$, and hence satisfiable as well. $\mathcal{A}_{V(G)}$ is a collection of *PARITY* functions, which are *Sensitive*. Finally, for $V' \subseteq V$, $\partial\mathcal{A}_{V'} = \{x_e : e \in E(V', V \backslash V')\}$. This is true because, if $e = (v, u)$, $v \in V'$, $u \in V \backslash V'$, then $PARITY_v$ is the only function of $\mathcal{A}_{V'}$ dependent on x_e. Hence, $e(\mathcal{A}_V) \geq e(G)$ and we apply Theorem 5.9 to complete the proof. □

6.2. THE PIGEONHOLE PRINCIPLE. In this section, we give the proofs of the main results regarding the Pigeonhole Principle, that is, Theorem 4.9, Corollary 4.10, and Theorem 4.15.

PROOF (THEOREM 4.9). The proof follows the strategy presented in Section 5, and we shall use the same notation, but the notion of *Boundary* has to be altered slightly. Define $\mathcal{A} = \{A_i : 1 \leq i \leq m\}$ where A_i is the conjunction of EP_i and all hole axioms $H_j^{i,i'}$. Let us denote $A_I = \wedge_{i \in I} A_i$. Set $\mu(C) = \mu_{\mathcal{A}}(C)$.

Clearly, $\mu(Axiom) \leq 1$, $\mu(0) = n + 1$, and μ is subadditive. Hence, in every refutation π there must be a clause C with $n/3 \leq \mu(C) < 2n/3$. Fix such a C and fix a minimal $I \subset [m]$ such that $A_I \models C$. Let $R(C)$ the set of rows who have a literal in C.

If $|C| \geq n/3$, we are done. Otherwise, there must be some $i \in I \backslash R(C)$. Take any assignment α such that $A_{I \backslash i}(\alpha) = 1$, $A_i(\alpha) = C(\alpha) = 0$, which must exist by the minimality of I. Without loss of generality, α sets all variables outside $R(C) \cup I \backslash i$ to 0. By the definition of the A_k's the 1's of original variables in α must be a partial matching. But as $|C| < n/3$ and $|I| \leq 2n/3$, there must be a column j in which no original variable is set to 1. Flip the assignment α to set x_{ij} to 1, and extend the nondeterministic variables y_i in any way to set EP_i to 1. Call this new assignment β. It is easy to verify that $A_I(\beta) = 1$, $C(\beta) = 0$, contradiction. □

PROOF (COROLLARY 4.10). If $w(EPHP_n^m) = 3$, we apply Corollary 3.4. If this is not the case, one can replace every clause with a nondeterministic 3-CNF extension, in the manner described in Example 4.8. Call this new formulation

$E^2PHP_n^m$. It is easy to verify that $E^2PHP_n^m$ is still a legal Row Extension of the Pigeonhole Principle. It is also easy to verify that if τ is the nondeterministic extension of a clause C, derived as described in 4.8, one can derive C from τ in $w(C)$ tree like resolution steps. Hence, the exponential lower bound for treelike refutations of $E^2PHP_n^m$ carries over to $EPHP_n^m$. □

PROOF (THEOREM 4.15). Similar to the previous proof of Theorem 4.9. Define $\mathcal{A} = \{A_v : v \in V$ with A_v as the conjunction of P_v and all hole axioms $H_u^{v,v'}$. Let us denote $A_{V'} = \wedge_{v \in V'} A_v$. Set $\mu(C) = \mu_{\mathcal{A}}(C)$.

Again, $\mu(Axiom) \leq 1$, $\mu(0) \geq r$ (because every V' of size $|V'| \leq r$ has a matching into U), and μ is subadditive. Hence, in every refutation π there must be a clause C with $r/2 \leq \mu(C) < r$. Fix such a C and fix a minimal $V' \subset V$ such that $A_{V'} \models C$.

We claim that for each $u \in \partial V'$, there must appear in C some variable $x_{(\hat{v},u)}$, for some $\hat{v} \in V$ (but not necessarily in V'). Indeed, for such a boundary u, let v be its only neighbor in V'. Assume for the sake of contradiction that C has no variable $x_{(\hat{v},u)}$. Let α be the assignment satisfying $A_{V' \setminus \{v\}}$ and falsifying A_v and C. Clearly α assigns zero to $x_{v,u}$. We assume without loss of generality that α sets to zero all variables $x_{(\hat{v},u)}$, because this cannot falsify any axiom $P_{v'}$ for $v' \in V'$ (recall that $x_{(v,u)}$ is a boundary variable). Thus, we may flip α on $x_{(v,u)}$, and get an assignment satisfying $A_{V'}$, without changing the value of C (zero), contradiction. Hence, the width of C is at least the size of its boundary, and the theorem is proven. □

6.3. RANDOM k-CNF's. In this section, we prove Theorem 4.19. In order to prove lower bounds on width of refuting a random formula $\mathcal{F}$, it is enough to look at the expansion properties of the following hypergraph. The vertex set is the set of variables, and each clause defines an edge. Formally:

Definition 6.2. For $\mathcal{F}$ a 3-CNF formula on n variables and m clauses, let $H_{\mathcal{F}}$ denote the 3-uniform hypergraph on n vertices and $m = \Delta \cdot n$ edges defined by

$$V(H_{\mathcal{F}}) = \{1, 2, \ldots, n\}$$

$$E(H_{\mathcal{F}}) = \{(i_1, i_2, i_3) : \exists C \in \mathcal{F} \text{ such that } x_{i_1}, x_{i_2}, x_{i_3} \in C\}.$$

For any subset V' of vertices, let $E(V')$ denote the set of edges within V' (i.e., for any $e \in E(V')$, all 3 vertices are in V'), and similarly for any subset E' of edges, let $V(E')$ denote the set of vertices covered by E'.

We shall need a generalized definition of expansion, suited for hypergraphs. We do not attempt to optimize constants.

Definition 6.3 (*Hypergraph Expansion*). For H a 3-uniform hypergraph, with Δn edges, where $\Delta = n^{1/2-\epsilon}$, $0 \leq \epsilon \leq 1/2$, the *expansion* of H is

$$e(H) \stackrel{\text{def}}{=} \min\{2|V(E')| - 3|E'| : E' \subset E, n \cdot (80\Delta)^{-2/(1-\epsilon)} \leq |E'| \leq 2n \cdot (80\Delta)^{-2/(1-\epsilon)}\}.$$

Positive expansion by itself is not enough to ensure high refutation width, and we need to be sure that $\mathcal{F}$ does not include a "small" unsatisfiable subformula. For this, we need the following definition.

Definition 6.4 (*Partial Matchability*). H, a 3-uniform Hypergraph, with Δn edges, where $\Delta = n^{1/2-\epsilon}$, $0 \leq \epsilon \leq 1/2$, is called *Partially matchable* if $\forall E' \subset E$ such that $|E'| \leq 2n \cdot (80\Delta)^{-2/(1-\epsilon)}$ we have $|V(E')| \geq |E'|$.

The main theorem of this section states that if $H_{\mathcal{F}}$ is partially matchable, then the expansion bounds the width from below:

THEOREM 6.5. *For an unsatisfiable* 3*-CNF*, *such that* $H_{\mathcal{F}}$ *is partially matchable*: $w(\mathcal{F} \vdash 0) \geq e(H_{\mathcal{F}})$.

PROOF. We use the notation of Section 5. Set $\mathcal{A}$ to be $\mathcal{F}$ and let $\mu = \mu_{\mathcal{A}}$. Clearly $\mu(Axiom) = 1$. Thus, $\mathcal{F}$ is a set of sensitive functions, compatible with $\mathcal{F}$. By the matchability of $H_{\mathcal{F}}$, $\mu(0) \geq 2n \cdot (80\Delta)^{-2/(1-\epsilon)}$, since we can find a matching from every small size subformula $\mathcal{F}' \subset \mathcal{F}$, $|\mathcal{F}'| \leq 2n \cdot (80\Delta)^{-2/(1-\epsilon)}$ into $Vars(\mathcal{F}')$, and use this matching to find an assignment satisfying $\mathcal{F}'$. Finally, $|\partial\mathcal{F}'| \geq 2|V(E')| - 3|E'|$, where $E' = E(H_{\mathcal{F}'})$, because every variable (vertex) *outside* the boundary of $\mathcal{F}'$ (E') must be covered by at least 2 clauses (hyperedges), so $|V(E')| \leq |\partial\mathcal{F}'| + \frac{1}{2}(3|E'| - |\partial\mathcal{F}'|)$. Applying Theorem 5.9 completes the proof. □

Thus, we have reduced Theorem 4.19 to proving high expansion and partial matchability of the underlying hypergraph. For the proper clause density, a random $\mathcal{F}$ conforms to the conditions of the previous theorem, and the following lemma, which states this explicitly, completes the proof of Theorem 4.19. This lemma was originally proved by Beame et al. [2000]. The proof uses a simple union bound calculation, and is presented in the appendix, for the sake of completeness.

LEMMA 6.6 [BEAME ET AL. 2000]. *For every* $0 \leq \epsilon \leq 1/2$, *if* $\Delta = n^{(1/2)-\epsilon}$ *and* H *is a random* 3*-uniform hypergraph on* n *vertices and* Δn *edges*, *then whp*:

(1) $e(H) \geq \epsilon n(80\Delta)^{-2/(1-\epsilon)}$.
(2) H *is partially matchable*.

7. *Tightness of the Bounds*

7.1. TIGHTNESS FOR TREE-LIKE RESOLUTION. How large can be the gap between $S_T(\tau)$ and $w(\tau \vdash 0)$?

Our answer is that the gap can be very large. Specifically, we shall show a family of tautologies that can be refuted in constant width, but for which the minimal tree-like refutation has size that is exponential in the input size. This family of contradictions, which is a generalization of Raz and McKenzie [1999] and Bonet et al. [1998], is presented in the following definition. For these contradictions, we prove a connection between pebbling and tree-like Resolution size, and they provide us with an exponential gap between general Resolution and tree-like Resolution (Corollary 7.4).

Definition 7.1. A circuit G is a DAG with a single target, in which each vertex has fan-in 2 or 0. A vertex with fan-in 0 is called a *source* and a vertex with fan-out 0 is called a *target*. Associate a *pair* of Boolean variables $x(v)_0, x(v)_1$ with every vertex $v \in V(G)$. Peb_G, the *Pebbling Contradiction* of G is the conjunction of:

Source Axioms: $x(s)_0 \vee x(s)_1$ for s a Source vertex.

Target Axioms: $\bar{x}(t)_0 \wedge \bar{x}(t)_1$ for t a Target (Two singleton clauses per target vertex).

Pebbling Axioms: $(x(u_1)_a \wedge x(u_2)_b) \rightarrow (x(v)_0 \vee x(v)_1)$ for u_1, u_2 the sons of v, and all $a, b \in \{0, 1\}$.

Notice that $Peb(G)$ is a nonsatisfiable 4-CNF over $2|V|$ variables, with $O(|V|)$ clauses.

For G a circuit, let $P(G)$ be the pebbling measure of G, which is the minimal space needed to carry out the calculation of the circuit G, assuming the output of each node requires one memory cell. For a thorough survey of pebbling see Pippenger Pebbling [1980].

All proofs of the following claims appear in Ben-Sasson et al. [2000].

LEMMA 7.2. *For G a circuit, $S(Peb_G) = O(|V|)$ and $w(Peb_G \vdash 0) \leq 6$.*

THEOREM 7.3. $S_D(Peb_G) = 2^{\Omega(P(G))}$.

Celoni et al. [1977] present for every n explicit constructions of G_n with $|V(G)| = O(n)$ and $P(G) = \Omega(n/\log n)$. The Pebbling contradictions of these graphs provide the best-known separation between treelike and general Resolution, improving the recent $\exp(\sqrt{n})$-separation of Bonet et al. [1998].

COROLLARY 7.4. *For all large enough n, there exist formulas $\mathcal{F}_n$ of size $\mathcal{F}_n = n$, such that $S(\mathcal{F}_n) = O(n)$, and $w(\mathcal{F}_n \vdash 0) = O(1)$, but $S_T(\mathcal{F}_n) = 2^{\Omega(n/\log n)}$.*

7.2. TIGHTNESS FOR GENERAL RESOLUTION. Recently, it has been shown by Bonet and Galesi [1999] that the trade-off of Theorem 3.5 is as tight as one can hope for. Specifically, Bonet and Galesi [1999] present a natural family of graph-based k-CNF contradictions (k is constant), τ_n, for which $S(\tau_n) = n^{O(1)}$, but $w(\tau_n \vdash 0) = \Omega(\sqrt{n})$.

8. *Automated Theorem Proving*

One of the most extensively used and investigated methods for proving unsatisfiability of CNF formulas, are commonly called Davis–Putnam procedures. Actually, these procedures are derived from a system devised by Davis, Logemann and Loveland [1962], and hence we will refer to them as *DLL Procedures*. A DLL procedure relies on choosing a variable x, and trying to refute $\mathcal{F}|_{x=T}$ and $\mathcal{F}|_{x=F}$ recursively. If $\mathcal{F}$ is unsatisfiable, $DLL(\mathcal{F})$ terminates providing a tree-like resolution refutation of $\mathcal{F}$.

An immediate consequence of the Size-Width trade-off is a different procedure for refuting unsatisfiable formulas. This Algorithm is a known heuristic in the AI community. Galil [1977] investigated this algorithm, and used linear width lower bounds for proving its inefficiency for Tseitin formulas. Recently, this algorithm was suggested in Beame and Pitassi [1996], based on the Groebner Basis (GB) algorithm [Clegg et al. 1996]. The essence of this procedure is seeking a *minimal width* refutation, and can be described schematically by the following algorithm:

```
A (F)
Fix w = 0
Repeat {
        f 0 ∈ F end
        End {
              Increase w
              Derive all resolution consequences of width ≤ w
              }
        }
```

Algorithm A has running time bounded by $n^{O(w(\mathcal{F}\vdash 0))}$, as this is the maximal number of different clauses that will be encountered. For example, if $\mathcal{F}$ has a polysize treelike refutation, then $RunTime(A(\mathcal{F})) = |\mathcal{F}|^{O(\log|\mathcal{F}|)}$. Moreover, the previous section (Corollary 7.4) provides concrete examples for tautologies where Algorithm A exponentially outperforms DLL procedures:

THEOREM 8.1. *There exist unsatisfiable formulas such that* $Time(DLL(\mathcal{F}))$ *is exponentially larger than* $Time(B(\mathcal{F}))$.

PROOF. We use the notation of Section 7.1. Take $\mathcal{F} = Peb(G)$ for G with $P(G) = |V|/\log|V|$. By Lemma 7.2, $RunTime(A(\mathcal{F})) = |V|^{O(1)} = |\mathcal{F}|^{O(1)}$, while by Theorem 7.3 any tree-like refutation of $\mathcal{F}$, for example, a DLL procedure, must have $RunTime = 2^{\Omega(|\mathcal{F}|/\log|\mathcal{F}|)}$. □

9. *Open Problems*

9.1. IS RESOLUTION AUTOMATIZABLE? A proof system P is called *Automatizable* if there exists an algorithm A_P, which, when presented with a tautology (contradiction) $\mathcal{F}$, produces a proof (refutation) π of $\mathcal{F}$ in the system P and the running time of A_P is polynomial in the size of the minimal proof of A in P, that is, $Time(A_P(\mathcal{F})) \leq (S_P(\mathcal{F}))^{O(1)}$. Similarly, P is called *Quasi-Automatizable* if the running time of the above mentioned A_P is Quasi-Polynomial in the size of the minimal proof, that is, $Time(A_P(\mathcal{F})) \leq S_P(\mathcal{F})^{O(\log(S_P(\mathcal{F})))^{O(1)}}$.

The algorithm of the previous section shows finds a Resolution proof in time quasi-polynomial in the size of the smallest Tree-like Resolution proof. The following questions remain open:

Is General Resolution Automatizable? Quasi-Automatizable? Is Tree-like Resolution Automatizable? Quasi-Automatizable?

9.2. IMPROVING THE LOWER BOUND FOR RANDOM k-CNF'S. The present lower bounds for random k-CNF's in Beame et al. [2000] follow from a lower bound on the Boundary size for random k-Uniform Hypergraphs. This is obtained via the Union bound. One possible method for improving the lower bound on size of refutations of random formulas would be to replace this Union bound with a finer analysis of the probability of a random hypergraph having a small boundary.

Appendix A

PROOF [LEMMA 6.6]. The proof of the lemma, originally presented by Beame et al. [2000], is done by calculating a simple Union bound. We start by proving

part (1). For a subset of edges $E' \subseteq E$, let the *expansion constant* of E' be

$$e_{E'} \stackrel{\text{def}}{=} \frac{2|V(E')| - 3|E'|}{|E'|}.$$

In order to prove that whp $e(H) \geq \epsilon n(80\Delta)^{-2/(1-\epsilon)}$, it is enough to show that whp

$$\min\{e_{E'} : E' \subset E, n(80\Delta)^{-2/(1-\epsilon)} \leq |E'| \leq 2n(80\Delta)^{-2/(1-\epsilon)}\} \geq \epsilon.$$

Let A_i denote the event that a set of edges of size i has an expansion constant of less than ϵ, that is,

$$|V(E')| < \frac{3+\epsilon}{2} \cdot |E'|.$$

Denote.

$$\lambda \stackrel{\text{def}}{=} \frac{3+\epsilon}{2}.$$

and

$$k \stackrel{\text{def}}{=} n(80\Delta)^{-2/(1-\epsilon)} = n^{\epsilon/(1-\epsilon)} \cdot 80^{-2/(1-\epsilon)}.$$

We wish to bound the probability that $|V(E')| < \lambda|E'|$, for any edge set of size i, where $i = k \cdots 2k$. This is done by summing over all possible edge sets—there are $\binom{\Delta n}{i}$ such sets, and all possible small vertex sets—there are $\binom{n}{\lambda i}$ such sets, the probability that all edges fall in this small vertex set. The probability for a single edge to fall within the small vertex set is

$$\binom{\lambda i}{3} \Big/ \binom{n}{3} \leq \left(\frac{\lambda i}{n}\right)^3$$

(because $\lambda i < n$), and this must happen i times independently. Thus, we get

$$\Pr[A_i] \leq \binom{\Delta n}{i} \cdot \binom{n}{\lambda i} \cdot \left(\frac{\lambda i}{n}\right)^{3i}. \tag{1}$$

Using the standard estimation

$$\binom{a}{b} \geq \left(\frac{e \cdot a}{b}\right)^b$$

we get

$$\Pr[A_i] \leq \left(\frac{e\Delta n}{i}\right)^i \left(\frac{en}{\lambda i}\right)^{\lambda i} \left(\frac{\lambda i}{n}\right)^{3i} \tag{2}$$

$$\leq \left[e^{1+\lambda} \cdot \lambda^{3-\lambda} \cdot \Delta \cdot \left(\frac{i}{n} \right)^{2-\lambda} \right]^i. \tag{3}$$

Notice that since $0 < \epsilon \leq 1/2$, we get $e^{1+\lambda} \cdot \lambda^{3-\lambda} < 20$. Using the definition $\Delta = n^{(1/2)-\epsilon}$, we derive

$$\Pr[A_i] \leq \left[20 \cdot \Delta \cdot \left(\frac{i}{n} \right)^{(1-\epsilon)/2} \right]^i \tag{4}$$

$$\leq [20 \cdot n^{-\epsilon/2} \cdot i^{(1-\epsilon)/2}]^i. \tag{5}$$

Finally, summing over all possible $i \in \{k, \ldots, 2k\}$, and plugging in the definition of k, we get

$$\Pr\left[\bigvee_{i=k}^{2k} A_i \right] \leq k[20 \cdot n^{-\epsilon/2} \cdot (2k)^{(1-\epsilon)/2}]^k \tag{6}$$

$$\leq k \left[\frac{20}{40} \cdot n^{-\epsilon/2 + (\epsilon/(1-\epsilon) \cdot (1-\epsilon)/2)} \right]^k \tag{7}$$

$$\leq k \cdot 2^{-k} \tag{8}$$

Thus, whp, as n tends to infinity, $\Pr[\bigvee_{i=k}^{2k} A_i]$ tends to zero.

Next we turn to part (2) of the lemma, concerning the partial matchability property. H does not have this property iff there is a set of edges E', $|E'| \leq n \cdot (80\Delta)^{-2/(1-\epsilon)}$, such that $|V(E')| < |E'|$. Let B_i be the event that for a set E' of i edges, $|V(E')| < |E'|$.

$$\Pr[B_i] \leq \binom{\Delta n}{i} \cdot \binom{n}{i} \cdot \left(\frac{i}{n} \right)^{3i} \tag{9}$$

$$\leq \left[e^2 \Delta \frac{i}{n} \right]^i \tag{10}$$

$$\leq [e^2 n^{-((1/2)+\epsilon)} i]^i \tag{11}$$

We need to sum over all $i \leq 2k$, and we split this sum into two parts, and bound each. Let $k' = n^{\epsilon/4(1-\epsilon)}$.

$$\Pr\left[\bigvee_{i=1}^{k} B_i \right] \leq \Pr\left[\bigvee_{i=1}^{k'} B_i \right] + \Pr\left[\bigvee_{i=k'+1}^{2k} B_i \right]. \tag{12}$$

We start with the first sum.

$$\Pr\left[\bigvee_{i=1}^{k'} B_i \right] \leq \sum_{i=1}^{k'} [e^2 \cdot n^{-((1/2)+\epsilon)} \cdot n^{\epsilon/4(1-\epsilon)}] \tag{13}$$

$$\leq n^{\epsilon/4(1-\epsilon)} \cdot e^2 \cdot n^{-((1/2)+\epsilon)} \cdot n^{\epsilon/4(1-\epsilon)} \tag{14}$$

Recalling that $0 \leq \epsilon \leq 1/2$, it is easy to verify that $\epsilon/2(1-\epsilon) < \frac{1}{2} + \epsilon$, and thus, $\Pr[\bigvee_{i=1}^{k'} B_i] \leq n^{-\delta}$, for some $\delta > 0$, and this probability tends to 0 as n tends to infinity.

For the second sum, we first look at Eq. (11) and note that by the definition of k, $e^2 n^{-((1/2)+\epsilon)}\, i \leq e^2 n^{-((1/2)+\epsilon)}\, 2k < 1/2$, and hence

$$\Pr\left[\bigvee_{i=k'+1}^{2k} B_i\right] \leq 2k \cdot 2^{-k'}.$$

Once again, this sum tends to zero as n tends to infinity, and the lemma is proven. □

ACKNOWLEDGMENTS. We thank Albert Atserias Perí, Juan Luis Esteban Angeles, and Maria Luisa Bonet Carbonell for carefully reading and commenting on an earlier version of the paper.

REFERENCES

BEAME, P., KARP, R., PITASSI, T., AND SAKS, M. 2000. On the complexity of unsatisfiability proofs for random k-CNF formulas. Submitted.

BEAME, P., AND PITASSI, T. 1996. Simplified and improved resolution lower bounds. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science* (Burlington, Vt., Oct.). IEEE Computer Society Press, Los Alamitos, Calif., pp. 274–282.

BEN-SASSON, E., IMPAGLIAZZO, R., AND WIGDERSON, A. 2000. Optimal separation of tree-like and general resolution. In *Electronic Colloquium on Computational Complexity Report Series*. Tech. Rep. TR00-005.

BONET, M. L., ESTEBAN, J. L., GALESI, N., AND JOHANNSEN, J. 1998. On the relative complexity of resolution refinements and cutting planes proof systems. In *Electronic Colloquium on Computational Complexity Report Series*. Tech. Rep. TR98-035.

BONET, M. L., AND GALESI, N. 1999. A study of proof search algorithms for resolution and polynomial calculus. In *Proceedings of the 40th Symposium of Foundations of Computer Science*. IEEE Computer Society, Los Alamitos, Calif., pp. 422–431.

BUSS, S., AND PITASSI, T. 1997. Resolution and the weak pigeonhole principle. In *Proceedings of the Conference for Computer Science Logic*.

BUSS, S. R., AND TURAN, G. 1988. Resolution proof of generalized pigeonhole principles. *Theoret. Comput. Sci. 62*, 311–317.

CELONI, R., PAUL, W. J., AND TARJAN, R. E. 1977. Space bounds for a game on graphs. *Math. Syst. Theory 10*, 239–251.

CHVÁTAL, V., AND SZEMERÉDI, E. 1988. Many hard examples for resolution. *J. ACM 35*, 4 (Oct.), 759–768.

CLEGG, M., EDMONDS, J., AND IMPAGLIAZZO, R. 1996. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (Philadelphia, Pa., May 22–24). ACM, New York, pp. 174–183.

COOK, S. A., AND RECKHOW, R. 1979. The relative efficiency of propositional proof systems. *J. Symb. Logic 44*, 36–50.

DAVIS, M., LOGEMANN, G., AND LOVELAND, D. 1962. A machine program for theorem-proving. *Commun. ACM 5*, 7 (July), 394–397.

GALIL, Z. 1977. On resolution with clauses of bounded size. *SIAM J. Comput. 6*, 444–459.

HAKEN, A. 1985. The intractibility of resolution. *Theoret. Comput. Sci. 39*, 297–308.

IMPAGLIAZZO, R., PUDLÁK, P., AND SGALL, J. 1997. Lower bounds for the polynomial-calculus and the groebner basis algorithm. Tech. Rep. TR97-042. Found at Electronic Colloqium on Computational Complexity, Reports Series 1997. Available at http://www.eccc.uni-trier.de/eccc/.

PIPPENGER PEBBLING, N. 1980. IBM Reserach Report RC8258. Appeared in *Proceedings of the 5th IBM Symposium on Mathematical Foundations of Computer Science*, Japan.

RAZ, R., AND MCKENZIE, P. 1999. Separation of the monotone NC hierarchy. *Combinatorica 19*, 3, 403–435.

RAZBOROV, A. A. 1995. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izv. RAN 59*, 1, 201–222.

RAZBOROV, A. A., AND RUDICH, S. 1994. Natural proofs. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing* (Montreal, Que., Canada, May 23–25). ACM, New York, pp. 204–213.

RAZBOROV, A. A., WIGDERSON, A., AND YAO, A. 1997. Read-once branching programs, rectangular proofs of the pigeonhole principle and the transversal calculus. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing* (El Paso, Tex., May 4–6). ACM, New York, pp. 739–748.

TSEITIN, G. S. 1968. On the complexity of derivation in propositional calculus. In *Studies in Constructive Mathematics and Mathematical Logic*, Part 2. Consultants Bureau, New York–London, pp. 115–125.

URQUHART, A. 1987. Hard examples for resolution. *J. ACM 34*, 1 (Jan.), 209–219.

URQUHART, A. 1995. The complexity of propositional proofs. *Bull. Symb. Logic 1*, 4, 425–467.

RECEIVED JUNE 1999; REVISED JULY 2000; ACCEPTED JULY 2000