comput. complex. **10** (2001), 261 – 276
1016-3328/01/040261–16

computational complexity

OPTIMALITY OF SIZE-WIDTH TRADEOFFS FOR RESOLUTION

Maria Luisa Bonet and Nicola Galesi

Abstract. This paper is concerned with the complexity of proofs and of searching for proofs in resolution. We show that the recently proposed algorithm of Ben-Sasson & Wigderson for searching for proofs in resolution cannot give better than weakly exponential performance. This is a consequence of our main result: we show the optimality of the general relationship called size-width tradeoff in Ben-Sasson & Wigderson. Moreover we obtain the optimality of the size-width tradeoff for the widely used restrictions of resolution: regular, Davis–Putnam, negative, positive.

Keywords. Resolution, complexity of proofs, automated theorem proving.

Subject classification. 03F20, 68T15, 03B05.

1. Introduction

Proof Complexity Theory is concerned with proving non-trivial lower bounds on the length of proofs of classes of tautologies in sound and complete propositional proof systems. This question is closely related to an important open problem in complexity theory. Cook & Reckhow (1979) proved that $NP \neq co\text{-}NP$ iff for every propositional proof system there is a class of tautologies that require superpolynomial size proofs. Proving superpolynomial lower bounds is also very relevant to the study of automated theorem provers. In many applications, given a possible tautology, we are faced with the problem of finding a proof of it, if one exists. Then we encounter two problems: (1) the complexity of the smallest possible proof, which might be exponential in the size of the tautology, and (2) the complexity of the proof search.

Regarding the first problem, most probably no propositional proof system can prove all tautologies efficiently, otherwise P would be equal to NP (Cook & Reckhow 1979), which is generally believed to be false. One approach to fixing the inherent inefficiency of propositional proof systems is to use the more efficient ones. Then we are faced with the second problem. How hard is it then to find proofs? It seems that the more efficient a proof system is, the

harder it is to find proofs in it. Bonet *et al.* (2000b) defined a notion of automatizability. A propositional proof system is *automatizable* if and only if there is a deterministic procedure to find proofs in that system in polynomial time with respect to the smallest proof in that system. In the sequence of papers Krajíček & Pudlák (1998) and Bonet *et al.* (1998, 2000b) it is proved that any propositional proof system that simulates bounded-depth Frege is not automatizable, unless some widely accepted cryptographic conjectures are violated. There is also evidence that resolution is not automatizable (see Alekhnovich & Razborov (2001)).

There are some algorithms to find proofs in some proof systems. For instance, Davis *et al.* (1962), Ben-Sasson & Wigderson (2001) and Beame & Pitassi (1996) gave algorithms for resolution, and Clegg *et al.* (1996) for polynomial calculus. The algorithms of Ben-Sasson & Wigderson (2001) and Beame & Pitassi (1996) are both weakly exponential for resolution, and that of Clegg *et al.* (1996) is polynomial for the system of polynomial calculus with bounded polynomial degree. Therefore this bounded-degree polynomial calculus is automatizable. In this paper we study the performance of the algorithm proposed in Ben-Sasson & Wigderson (2001) for finding resolution refutations.

Ben-Sasson & Wigderson (2001), based on previous work of Haken (1985) and Beame & Pitassi (1996), defined the *width* as a complexity measure for resolution refutations. Width of a refutation is the maximal number of literals in any clause of the refutation. The importance of considering this measure is twofold. On the one hand, they were able to give a general relationship between the width and length of a refutation, reducing the problem of giving lower bounds on the length to that of giving lower bounds on the width. The width-size relation can be stated as follows: If F, an unsatisfiable formula over n variables, has a resolution refutation of size S, then it has a resolution refutation of width $O(\sqrt{n \log S})$. Through this relationship they obtained a unified method to prove most of the previously known lower bounds for resolution. On the other hand, Ben-Sasson & Wigderson (2001) made explicit a new simple proof search algorithm based on searching for clauses of increasing size. This algorithm works in time $T(n) = n^{O(w)}$ where w is the minimal width of any refutation of F.

The tradeoff of Ben-Sasson & Wigderson (2001) shows that for k-CNF, with k a constant, if we can give a width lower bound equal to the number of variables, then we have an exponential lower bound for the size of refuting the formula. An immediate interesting question arising from the work of Ben-Sasson & Wigderson (2001) is whether we can improve the size-width tradeoff there obtained. That is, can we get a weaker width lower bound (e.g. to the

square root of the number of variables), but still obtain an exponential lower bound for the size?

Notice that an affirmative answer to this question for the case of unrestricted resolution could give exponential lower bounds for the size of resolution refutations of formulas such as the PHP_n^m, $m \geq n^2$, which at this point use a more complicated argument (see Raz 2002; Razborov 2002).

Here we give a negative answer to this question for the case of unrestricted resolution. In fact we find a k-CNF formula built over $O(n^2)$ variables, MGT_n, having polynomial size unrestricted resolution refutations, but requiring $\Omega(n)$ (the square root of the number of variables) width to be refuted. Combining our result with the size-width tradeoff for tree-like resolution it turns out that MGT_n requires exponential size tree-like resolution proofs, and therefore it provides an exponential separation between tree-like and unrestricted resolution (see also Ben-Sasson & Wigderson 2001; Bonet *et al.* 2000a).

In this paper we also study whether for restrictions of resolution, different from tree-like, we can obtain a better tradeoff result than that given for unrestricted resolution. We give a negative answer to this question. For restrictions of resolution such as regular, Davis–Putnam, positive and negative, we show the optimality of the same size-width tradeoff given for unrestricted resolution. As for the case of unrestricted resolution this fact implies an exponential separation between tree-like resolution and all the restrictions considered. Similar results were obtained in Bonet *et al.* (2000a) and Ben-Sasson *et al.* (2001).

Another important, even if negative, consequence of our result is that the proof search algorithm proposed by Ben-Sasson & Wigderson (2001) is not going to be efficient for finding resolution refutations in any of the restrictions of resolution we consider.

In Section 2 we give the preliminary definitions that will be needed throughout the paper. In Section 3 we show the optimality of the width-size method. In Section 4 we prove some consequences of the results of Section 3.

2. Preliminaries

Let $\mathcal{V} = \{x_1, \ldots, x_n\}$ be a set of boolean variables. For a variable $x_i \in \mathcal{V}$ let $\bar{x}_i$ denote the negation of x_i. A literal ℓ_i can be either a variable x_i or its negation, with the convention that if $\ell_i = \bar{x}_i$, then $\bar{\ell}_i = x_i$ and that $\bar{\bar{x}}_i = x_i$. A clause is a disjunction of literals $\{\ell_{i_1}, \ldots, \ell_{i_k}\}$, possibly empty. A C(onjunctive) N(ormal) F(orm) formula is a conjunction of clauses. For a formula F, let Lit(F) be the set of literals of F.

Resolution is a refutation proof system for formulas in CNF form based on the following *resolution rule*:

$$\frac{C \vee x \quad \bar{x} \vee D}{C \vee D}$$

where if C and D have common literals, they appear only once in $C \vee D$. A resolution *refutation* of a CNF formula F is a derivation of the empty clause from the clauses defining F, using the above inference rule.

We also consider some restrictions of the resolution proof system:

- *Tree-like* resolution: Any clause in the proof is used at most once as premise in a resolution rule.
- *Regular* resolution: The proofs are restricted in such a way that any variable can be eliminated at most once in any path from an initial clause to the empty clause.
- *Davis–Putnam* resolution: The proofs are restricted in such a way that there exists an ordering of the variables such that if a variable x is eliminated before a variable y on any path from an initial clause to the empty clause, then x is before y in the ordering. Notice that Davis–Putnam resolution is necessarily regular.
- *Negative* resolution: The application of the resolution rule is restricted to the case in which one of the premises must not contain any positive literals.
- *Positive* resolution: The application of the resolution rule is restricted to the case in which one of the premises must not contain any negative literals.

Let $R \vdash F$ denote that R is a refutation of F (we use the notation $\vdash_{tl}$ to denote that the proof is tree-like). The *size* $|R|$ of a refutation R is the number of clauses used in R. The size complexity $S(\vdash F)$ of refuting a CNF formula F in resolution is defined as $\min_{R \vdash F} |R|$.

Following Ben-Sasson & Wigderson (2001) the *width* $w(F)$ of a CNF formula F is defined to be the number of literals of the largest clauses in F. The *width* $w(R)$ of a refutation R is defined as the size of the greatest clause appearing in R. The width $w(\vdash F)$ of refuting a formula F is defined as $\min_{R \vdash F} w(R)$.

The following theorem, proved in Ben-Sasson & Wigderson (2001), gives the tradeoff between size and width for the tree-like and the dag-like resolution systems:

THEOREM 2.1 (Ben-Sasson & Wigderson 2001). *Let F be any unsatisfiable CNF formula. Then:*

(i) $S(\vdash_{tl} F) \geq 2^{(w(\vdash F)-w(F))}$;

(ii) $S(\vdash F) \geq \exp\left(c \cdot \dfrac{(w(\vdash F)-w(F))^2}{|\mathrm{Lit}(F)|}\right)$,

where the constant c is an absolute constant, independent of F.

In what follows we will define the graph tautology that we will use to prove the optimality of the size-width tradeoff. A binary relation R is *asymmetric* if and only if for all pairs (x,y), $(x,y)\in R \rightarrow (y,x)\notin R$, and is *linear* if for all (x,y), either $(x,y)\in R$ or $(y,x)\in R$. A *(strict) ordering* is a transitive and asymmetric relation. A *linear (strict) ordering* is a strict order that is also linear.

It is straightforward to note that any strict order over a finite set has at least one minimal element. When we consider the directed graph associated to a strict order, the previous property is equivalent to the following: *each directed acyclic graph, closed under transitivity and without loops, must have a source node.*

We define a CNF formula, GT_n, expressing the negation of the previous property. For all $i,j\in[n]$, $i\neq j$, let $x_{i,j}$ be a variable whose intended meaning is that (i,j) is a directed edge in a graph over $[n]$. GT_n is then defined by the following clauses over $n(n-1)$ variables:

$$\begin{array}{lll} (1) & \bar{x}_{i,j}\vee\bar{x}_{j,k}\vee x_{i,k}, & i,j,k\in[n],\ i\neq j,\ j\neq k,\ k\neq i,\\ (2) & \bar{x}_{i,j}\vee\bar{x}_{j,i}, & i,j\in[n],\ i\neq j,\\ (3) & \bigvee_{k=1,\,k\neq j}^{n} x_{k,j}, & j\in[n], \end{array}$$

where the clauses in (1) say that the graph is transitive, those in (2) that it is asymmetric and those in (3) that there is no source node.

Krishnamurthy (1985) was the first to consider this formula and to study its complexity for resolution refutations. He conjectured that GT_n required long proofs in resolution. Stalmark (1996) refuted Krishnamurthy's conjecture giving polynomial size unrestricted resolution refutations.

3. Optimality of the size-width tradeoff

Consider the equation giving the size-width tradeoff for the case of unrestricted resolution:

$$S(\vdash F) \geq \exp\left(c \cdot \frac{(w(\vdash F)-w(F))^2}{|\mathrm{Lit}(F)|}\right),$$

where c is an absolute constant independent of the formula F.

One way to show that this tradeoff is (almost) optimal is to find an infinite family of unsatisfiable CNF formulas F_n such that

- $w(F_n)$ is $O(1)$,
- $S(\vdash F_n)$ is $n^{O(1)}$,
- $w(\vdash F_n)$ is $\Omega(\sqrt{|\mathrm{Lit}(F_n)|})$,
- $|\mathrm{Lit}(F_n)|$ goes to infinity as n grows.

Note that when F fulfills the first three conditions, the tradeoff formula does not give us an exponential lower bound.

We will consider a modification of the formula GT_n, which we call MGT_n, and we show that MGT_n fulfills the above requirements. We will show (see Theorem 3.2) that polynomial size resolution refutations for MGT_n can be easily obtained from polynomial size resolution refutations of GT_n. Therefore we start by giving a polynomial size resolution refutation for GT_n. This was first given by Stalmark (1996). Our proof slightly differs from that of Stalmark. However, this difference allows us to show that our refutations respect the further restrictions of being Davis–Putnam (and therefore regular) and positive.

THEOREM 3.1. *There are polynomial size refutations of GT_n in the following proof systems: (i) dag-like resolution, (ii) Davis–Putnam resolution, (iii) regular resolution, (iv) positive resolution.*

PROOF. We start by giving a scheme of the proof. Consider the following abbreviations:

$$\begin{array}{rll} C_m(j) := \bigvee_{i=1, i\neq j}^{m} x_{i,j} & & \text{for all } j \in [n], \\ A(i,j,k) := \bar{x}_{i,j} \vee \bar{x}_{j,k} \vee x_{i,k} & & \text{for all } i,j,k \in [n],\ i \neq j \neq k \neq i, \\ B(i,j) := \bar{x}_{i,j} \vee \bar{x}_{j,i} & & \text{for all } i,j \in [n],\ i \neq j. \end{array}$$

The idea of the proof is to obtain clauses of the form $C_m(j)$ from $m = n$ down to $m = 2$ in the following way:

$$\begin{array}{llll} C_n(1) & C_n(2) & \ldots\ C_n(n-1) & C_n(n) \\ C_{n-1}(1) & C_{n-1}(2) & \ldots\ C_{n-1}(n-1) & \\ \vdots & & & \\ C_2(1) & C_2(2) & & \end{array}$$

For each k, $C_k(1), \ldots, C_k(k)$ are obtained in parallel, and for $j = 1, \ldots, k-1$, each $C_{k-1}(j)$ is obtained using the clauses $C_k(j)$ and $C_k(k)$ derived in the previous step, and the initial clauses $A(1,k,j), A(2,k,j), \ldots, A(k-1,k,j)$ and $B(j,k)$. At the end we easily derive the empty clause from $C_2(1)$, $C_2(2)$ and $B(2,1)$. Now we provide more details of the proof and show that it respects the various restrictions of resolution.

Consider the following abbreviations:

$$D^j_{k-1}(i) := C_{k-1}(j) \vee \bar{x}_{i,k}, \qquad k \in [n] \setminus \{1\},\ i \in [k-1],\ j \in [n],$$

$$E^j_{k-1}(i) := C_{k-1}(j) \vee \bigvee_{\ell=i}^{k-1} x_{\ell,k}, \quad k \in [n] \setminus \{1\},\ i \in [k-1],\ j \in [n].$$

The proof proceeds by steps going from $m = n$ to $m = 2$. All the clauses $C_n(j)$, for all $j \in [n]$, are initial clauses and therefore derivable. At the $n-k+1$-th step, for each $j = 1, \ldots, k-1$, we prove in parallel each of the $C_{k-1}(j)$ in the following way:
(a) Perform in parallel the following resolution steps, each one resolving the variable $x_{k,j}$:

$$\begin{array}{ll}
(1) & \dfrac{C_k(j) \quad A(1,k,j)}{D^j_{k-1}(1)} \\[2ex]
(2) & \dfrac{C_k(j) \quad A(2,k,j)}{D^j_{k-1}(2)} \\[2ex]
\vdots & \\
(j-1) & \dfrac{C_k(j) \quad A(j-1,k,j)}{D^j_{k-1}(j-1)} \\[2ex]
(j) & \dfrac{C_k(j) \quad B(j,k)}{D^j_{k-1}(j)} \\[2ex]
(j+1) & \dfrac{C_k(j) \quad A(j+1,k,j)}{D^j_{k-1}(j+1)} \\[2ex]
\vdots & \\
(k-1) & \dfrac{C_k(j) \quad A(k-1,k,j)}{D^j_{k-1}(k-1)}
\end{array}$$

(b) $C_{k-1}(j)$ is then obtained by the following (tree-like) refutation in which we are resolving on the variables $x_{1,k}, x_{2,k}, \ldots, x_{k-1,k}$:

$$
\begin{array}{ll}
(1) & \dfrac{C_k(k) \quad D^j_{k-1}(1)}{E^j_{k-1}(1)} \\
(2) & \dfrac{E^j_{k-1}(1) \quad D^j_{k-1}(2)}{E^j_{k-1}(2)} \\
\vdots & \\
(k-1) & \dfrac{E^j_{k-1}(k-1) \quad D^j_{k-1}(k-1)}{C_{k-1}(j)}
\end{array}
$$

Such a refutation respects the positive restriction, since at each resolution step one of the premises contains only positive literals. It is also easy to see that the following order of elimination of the variables is respected:

$$
\begin{array}{l}
x_{n,1}, x_{n,2}, \ldots, x_{n,n-1} \\
x_{1,n}, x_{2,n}, \ldots, x_{n-1,n} \\
x_{n-1,1}, x_{n-1,2}, \ldots, x_{n-1,n} \\
x_{1,n-1}, x_{2,n-1}, \ldots, x_{n-2,n-1} \\
\vdots \\
x_{2,1} \\
x_{1,2}
\end{array}
$$

Therefore the refutation is Davis–Putnam as well as regular. □

In the above refutation there are clauses of size $\Omega(n)$. We show below that in fact any refutation of GT_n must have clauses of that width. Note however that in GT_n there are initial clauses of size $n-1$ and therefore GT_n does not fulfill the first requirement needed to show the optimality of the size-width tradeoff. We consider a modified version of GT_n, MGT_n, such that: (1) $w(MGT_n) \leq 3$; (2) $|\mathrm{Lit}(MGT_n)| = 2n^2 - n$; (3) from a refutation of GT_n we can easily find a refutation of MGT_n, and (4) $w(\vdash MGT_n) \geq \Omega(n)$.

Consider the clauses with large width in the definition of GT_n:

$$\bigvee_{k=1,\, k\neq j}^{n} x_{k,j} \qquad \text{for } j \in [n].$$

For each $j \in [n]$ let $y_{0,j}, \ldots, y_{j-1,j}, y_{j+1,j}, \ldots, y_{n,j}$ be n new *extension* variables. MGT_n is defined by replacing the large clauses in the definition of GT_n with the following set of clauses of constant width:

$$\overline{y}_{0,j} \wedge \bigwedge_{i=1,\, i\neq j}^{n} (y_{i-1,j} \vee x_{i,j} \vee \overline{y}_{i,j}) \wedge y_{n,j} \qquad \text{for all } j \in [n].$$

The intended meaning of the y variables is:

$$y_{ij} = \bigvee_{k=1,\, k\neq j}^{i} x_{kj}.$$

MGT_n has then constant initial width, and it is defined over $2n^2 - n$ literals. It remains to prove that: (1) there are polynomial size resolution refutations of MGT_n and (2) the minimal width for refuting MGT_n is $\Omega(n)$ (observe that the number of variables in MGT_n is $O(n^2)$). We start by proving that the upper bound holds, even for the restrictions of resolution.

THEOREM 3.2. *There are polynomial size refutations for the formula MGT_n in any of the following proof systems: (i) resolution, (ii) positive resolution, (iii) Davis–Putnam resolution, (iv) regular resolution.*

PROOF. The proof follows a standard argument. From the initial clauses of MGT_n,

$$\bar{y}_{0,j} \wedge \bigwedge_{i=1,\, i\neq j}^{n} (y_{i-1,j} \vee x_{i,j} \vee \bar{y}_{i,j}) \wedge y_{n,j} \qquad \text{for all } j \in [n],$$

we derive the initial clauses of GT_n,

$$\bigvee_{k=1,\, k\neq j}^{n} x_{k,j} \qquad \text{for all } j \in [n],$$

resolving on the y variables one at a time and in a tree-like fashion. Then we apply the polynomial size proof of GT_n to these new clauses using the other initial clauses. The part of the proof eliminating the y variables is in fact a tree-like proof of size quadratic in n. Since the y variables get resolved once only, the regularity of the proof is preserved. It is also easy to see that the new first part of the proof is a Davis–Putnam resolution since the following order of elimination of the y variables is respected:

$$\begin{aligned} &y_{0,1}, \ldots, y_{n,1}, \\ &y_{0,2}, \ldots, y_{n,2}, \\ &\vdots \\ &y_{0,n}, \ldots, y_{n-1,n}. \end{aligned}$$

To obtain a positive resolution refutation we only have to be careful to eliminate the $y_{k,j}$ variables starting from $y_{n,j}$ for all $j \in [n]$. □

In order to prove the lower bound for the width of refuting MGT_n, we need to introduce the notion of a critical truth assignment for the formula MGT_n. We start by giving the definition of critical assignments for GT_n and then extend it to the case of MGT_n. A *linear directed graph* is the graph associated to a strict linear order, i.e. a directed acyclic graph, closed under transitivity, without loops, and in which any two nodes are linked by an edge (see Figure 3.1).

Figure 3.1: Two linear directed graphs giving a 2-cta and a 4-cta for GT_6

Definition 3.3. A *critical truth assignment* α for GT_n is a linear directed graph.

The idea is that if the variable $x_{i,j}$ corresponds to the existence of a directed edge (i, j) in the graph, then such a linear graph falsifies only one among the initial clauses of GT_n. This is because the graph is closed under transitivity, there are no cycles, and every node except the first one in the line has a predecessor. We call a critical assignment a *j-cta* if j is the first element in the linear graph. A j-cta falsifies only the initial clause

$$\bigvee_{i\in[n],\, i\neq j} x_{i,j}.$$

Switching from a j-cta to a k-cta is very easy in terms of the linear graph. We put the node k in the first position in the line, and move all nodes before k one position forward. In terms of the adjacency matrix of the graph this means changing the 1's to 0's in column k and the 0's to 1's in row k. The following matrices show how to obtain a 4-cta from a 2-cta for GT_6 (see also Figure 3.1).

2-cta for GT_6					
$*$	0	0	**1**	1	1
1	$*$	1	**1**	1	1
1	0	$*$	**1**	1	1
0	**0**	**0**	$*$	**0**	1
0	0	0	**1**	$*$	1
0	0	0	0	0	$*$

4-cta for GT_6					
$*$	0	0	**0**	1	1
1	$*$	1	**0**	1	1
1	0	$*$	**0**	1	1
1	**1**	**1**	$*$	**1**	1
0	0	0	**0**	$*$	1
0	0	0	0	0	$*$

Denote by C_j the formula $\bar{y}_{0,j} \wedge \bigwedge_{i=1,\, i\neq j}^{n}(y_{i-1,j} \vee x_{i,j} \vee \bar{y}_{i,j}) \wedge y_{n,j}$ in the definition of MGT_n.

DEFINITION 3.4. A *j-cta* α *for* MGT_n is obtained from a j-cta for GT_n by assigning boolean values to the extension variables y in such a way that $\alpha \not\models C_j$ and for all $k \neq j$, $\alpha \models C_k$.

The next lemma shows that the previous definition is sound, that is, we can always extend a j-cta for GT_n to a j-cta for MGT_n.

LEMMA 3.5. *Any j-cta for GT_n can be extended to a j-cta for MGT_n.*

PROOF. Let α be a j-cta for GT_n; β will be a j-cta for MGT_n. Since α falsifies the initial clause $\bigvee_{k=1,\,k\neq j}^{n} x_{k,j}$, it also falsifies the formula C_j for any possible assignment to the variables $y_{i,j}$ for $i = 0, \ldots, n$. To define β we have to specify how to assign values to the $y_{i,k}$ variables for $i = 0, \ldots, n$ and for $k \neq j$, in such a way as to satisfy C_k. We can do this by considering $y_{i,k} = \bigvee_{l=1,\,l\neq k}^{i} x_{j,k}$, and assigning the same value to $y_{i,k}$ as α assigns to the disjunction. □

Notice that there is not a unique way to extend a j-cta for GT_n to a j-cta for MGT_n. The variables $y_{i,k}$ do not have to take the intended meaning for the j-cta to satisfy C_k.

Recall that C_j is the formula $\bar{y}_{0,j} \wedge \bigwedge_{i=1,\,i\neq j}^{n}(y_{i-1,j} \vee x_{i,j} \vee \bar{y}_{i,j}) \wedge y_{n,j}$.

Let $\mathrm{Vars}(j)$ be the set of variables in C_j together with $x_{j,i}$ for all $i \neq j$; i.e. the variables $x_{i,j}, x_{j,i}$ for all $i \in [n]$, $i \neq j$, and the variables $y_{i,j}$ for $i \in \{0\} \cup [n]$, $i \neq j$.

LEMMA 3.6. *Let K be a clause, and l the number of literals appearing in K. Then at most $2 \cdot l$ sets* $\mathrm{Vars}(j)$ *have variables appearing in K.*

PROOF. Every variable $x_{i,j}$ belongs to two different sets, $\mathrm{Vars}(i)$ and $\mathrm{Vars}(j)$. Each variable $y_{s,i}$ belongs to one set, $\mathrm{Vars}(i)$. So each variable of K can belong to at most two sets $\mathrm{Vars}(j)$. □

THEOREM 3.7. *Any resolution proof of MGT_n must have a clause of width $\Omega(n)$.*

PROOF. For each $I \subseteq [n]$, let C_I be defined as $\bigwedge_{i\in I} C_i$. For any clause C in a resolution proof of MGT_n, let I_C be a minimal $I \subseteq [n]$ such that all critical truth assignments satisfying C_I also satisfy C. For any clause C we define a *measure* $\mu(C)$ as the cardinality of I_C. Observe that for any $i \in [n]$, $\mu(C_i) \leq 1$. Moreover, $\mu(\{\}) = n$ since C_I is satisfiable when $I \neq [n]$, and μ is obviously subadditive, that is, for any step in the resolution proof, the measure of the conclusion is less than or equal to the sum of the measures of its premises.

Therefore in any resolution proof of MGT_n there is a clause, say K, such that $n/3 \leq \mu(K) < 2n/3$. We show that this clause will contain $\geq n/6$ literals. Assume for the sake of contradiction that $|K| < n/6$. Consider the set I_K and notice that $|I_K| = \mu(K) \geq n/3$. By Lemma 3.6 there is an $l \in I_K$ such that no variable from Vars(l) belongs to K.

Consider any critical assignment α such that $\alpha(C_l) = 0$, $\alpha(K) = 0$ and for all $j \in I_K \setminus \{l\}$, $\alpha(C_j) = 1$. This assignment exists by the minimality of I_K and moreover it satisfies all the clauses C_i for $i \in [n] \setminus \{l\}$. Define $J = [n] \setminus I_K$. Since $|J| \geq n/3$ (because $|I_K| < 2n/3$), by Lemma 3.6 we conclude that there is at least one $j \in J$ such that no variable from Vars(j) appears in K. We build a j-critical truth assignment β from α such that $\beta(C_i) = 1$ for all $i \in I_K$ and $\beta(K) = 0$, and this leads to a contradiction by the definition of I_K.

β is built in the following three steps. At the first step we change the value of the x variables in MGT_n, switching the l-cta α to a j-cta. This first step does not affect the value of K since we only change the variables $x_{i,j}$ and the variables $x_{j,i}$. These are in Vars(j) and no variable from Vars(j) appears in K.

For β to be a j-cta it has to satisfy the formulas C_k for $k \neq j$. Consider the case $k = l$. After the first step $\beta(x_{j,l}) = 0$, hence the second step in the definition of β will be to change the values of the variables $y_{i,l}$ for $i = 0, \ldots, n$ as in Lemma 3.5. This last change will not affect the value of K since for $i = 0, \ldots, n$ the variables $y_{i,l}$ belong to Vars(l) and no variable from Vars(l) appears in K.

Finally we need β to satisfy the formulas C_s for all $s \neq l$ and $s \neq j$. In the third step we define $\beta(y_{i,s}) = \alpha(y_{i,s})$ for $s \neq l$ and $s \neq j$. Notice that no x variable is changed from 1 to 0, except for those belonging to the j-th column of the assignment (see e.g. previous example). Therefore, since $\alpha(C_s) = 1$ for all $s \neq l$ and $s \neq j$, we have $\beta(C_s) = 1$ for all $s \neq l$ and $s \neq j$.

Under the hypothesis that $|K| < n/6$, we have defined an assignment β that falsifies K, and satisfies all C_i for $i \in I_K$. This is a contradiction by the definition of I_K. Therefore, $|K| \geq n/6$.

□

The following is an immediate corollary of the previous two theorems.

THEOREM 3.8. *There is a family of constant-width CNF formulas $\{F_n\}$ on $O(n^2)$ variables with the following two properties: (1) $\{F_n\}$ has polynomial size resolution refutations; (2) any resolution refutation of $\{F_n\}$ contains a clause having width at least $\Omega(n)$.*

4. Consequences of the optimality result

Our result has several consequences. First of all the size-width tradeoff of Ben-Sasson & Wigderson (2001) for tree-like resolution together with Theorem 3.7 gives a lower bound of $2^{\Omega(n)}$ for the size of tree-like resolution proofs of MGT_n.

THEOREM 4.1. *Any tree-like resolution proof of MGT_n must have size $\Omega(2^n)$.*

This theorem allows us to prove that tree-like resolution is exponentially separated from the other restricted systems of resolution we consider. These separations were only recently obtained in Bonet *et al.* (2000a) and Ben-Sasson *et al.* (2001).

THEOREM 4.2. *Tree-like resolution is exponentially separated from unrestricted, positive, negative, regular and Davis–Putnam.*

PROOF. For the case of regular, Davis–Putnam, positive and unrestricted resolution, the result is immediate since MGT_n has polynomial size refutation in these systems, but, by the previous theorem, requires exponential size tree-like refutations. For the case of negative resolution, we consider the unsatisfiable formula $\overline{MGT}_n$ in which the $x_{i,j}$ variables are replaced by $z_{i,j}$ whose intended meaning is opposite to that of the x variables. It is easy to see that the positive resolution proof for MGT_n is in fact a negative resolution proof for $\overline{MGT}_n$. Moreover the technique to obtain the lower bound for the width can also be applied. We leave the details to the reader. This means that the shortest tree-like resolution refutations of $\overline{MGT}_n$ are exponentially long in n. And therefore tree-like resolution is also exponentially separated from negative resolution. □

As we have seen in Section 2, the size-width tradeoff for tree-like resolution is much stronger than that of dag-like resolution. This fact, combined with our optimality result for the size-width tradeoff for dag-like resolution, allows us to obtain an exponential separation between tree-like and dag-like resolution (Theorem 4.2). Therefore an interesting question is to know whether we can also obtain exponential separations between resolution and other restrictions of resolution (e.g. regular), showing better tradeoffs for the latter. Our next theorem implies that this is not the case for regular, positive, negative and Davis–Putnam resolution. In fact we prove the optimallity of the size-width tradeoff for all these restrictions.

THEOREM 4.3. *The tradeoff*

$$S(\vdash F) \geq \exp\left(c \cdot \frac{(w(\vdash F) - w(F))^2}{|\mathrm{Lit}(F)|}\right)$$

is optimal for the systems of regular, positive, negative and Davis–Putnam resolution. The constant c is again an absolute constant.

PROOF. The polynomial size refutations for MGT_n provided in Theorem 3.2 are also in all the restrictions but negative resolution. By Theorem 3.7 any resolution refutation of MGT_n (in particular in any of the considered restrictions) must have a clause of size $\Omega(n)$. The result is hence immediate for these restrictions. In the case of negative resolution the result follows by an argument similar to that applied in the previous theorem. □

A refutation system is *automatizable* if there is an algorithm $\mathcal{A}$ such that, for any unsatisfiable formula F, $\mathcal{A}$ finds a refutation of F in that system in time polynomial in the size of the shortest proof of F in that system.

Ben-Sasson & Wigderson (2001) considered a simple algorithm for finding resolution proofs based on the idea of seeking for refutations of minimal width. Let F be an unsatisfiable CNF formula. Consider the following algorithm:

C := Clauses of F
$w := 0$
While $\Box \notin C$ Do
    $w := w + 1$
    apply resolution rule to clauses in C to derive
    all possible clauses of width $\leq w$.
    Add the clauses obtained to C
End

By definition of width it is immediate to see that the possible number of clauses that the above algorithm can produce is bounded by $n^{O(w(\vdash F))}$. Therefore the running time of the algorithm on input F is bounded by $n^{O(w(\vdash F))}$. Our main result (Theorem 3.8) implies that it cannot be used to obtain the automatizability of any of the resolution systems considered. This is because we have a formula, MGT_n, that has polynomial size resolution refutations, but, since the minimal width for refuting MGT_n is $\Omega(n)$, the algorithm will require an exponential number of steps to find a refutation of MGT_n in any of the considered restrictions.

Acknowledgements

Maria Luisa Bonet was partly supported by Project CICYT, TIC 98-0410-C02-01, Promoción General del Conocimiento PB98-0937-C04-03 and TIC2000-1970-CE. Part of this work was done while Nicola Galesi was a member of

the School of Mathematics at the Institute for Advanced Study (Princeton) supported by an NSF grant CCR-9987845. The authors would like to thank the anonymous referees for many comments about the preliminary version of this paper which appeared in Bonet & Galesi (1999). These comments contributed to improving the quality of the presentation.

References

M. Alekhnovich & A. Razborov (2001). Resolution is not automatizable unless W[P] is tractable. In *Proc. IEEE Symposium on Foundations of Computer Science (FOCS)*, 190–199.

P. Beame & T. Pitassi (1996). Simplified and improved resolution lower bounds. In *Proc. IEEE Symposium on Foundations of Computer Science (FOCS)*, 274–282.

E. Ben-Sasson, R. Impagliazzo & A. Wigderson (2001). Near optimal separations of general and tree-like resolution. Submitted. Appeared in the ECCC as TR00-005.

E. Ben-Sasson & A. Wigderson (2001). Short proofs are narrow—resolution made simple. *J. Assoc. Comput. Mach.* **48**, 149–168. A preliminary version appeared in Proc. 1999 Annual ACM Symposium on Theory of Computing.

M. Bonet, C. Domingo, R. Gavaldá, A. Maciel & T. Pitassi (1998). Non-automatizability of bounded-depth Frege proofs. In *Proc. IEEE Conference on Computational Complexity*, 15–23. To appear in J. Comput. Complexity.

M. L. Bonet, J. L. Esteban, N. Galesi & J. Johannsen (2000a). On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM J. Comput.* **30**, 1462–1484. A preliminary version appeared in Proc. 1999 IEEE Symposium on Foundations of Computer Science with the title *Exponential separations between restricted resolution and cutting planes proof systems*.

M. L. Bonet & N. Galesi (1999). A study of proof search algorithms for resolution and polynomial calculus. In *Proc. IEEE Symposium on Foundations of Computer Science (FOCS)*, 422–431.

M. L. Bonet, T. Pitassi & R. Raz (2000b). On interpolation and automatization for Frege systems. *SIAM J. Computing* **29**, 1939–1967. A preliminary version appeared in Proc. 1997 IEEE Symposium on Foundations of Computer Science with the title *No feasible interpolation for TC^0-Frege proofs*.

M. Clegg, J. Edmonds & R. Impagliazzo (1996). Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proc. 28th Annual ACM Symposium on the Theory of Computing* (Philadelphia, PA), 174–183.

S. Cook & R. Reckhow (1979). The relative efficiency of propositional proof systems. *J. Symbolic Logic* **44**, 36–50.

M. Davis, G. Logemann & D. Loveland (1962). A machine program for theorem proving. *Comm. Assoc. Comput. Mach.* **5**, 394–297.

A. Haken (1985). The intractability of resolution. *Theoret. Comput. Sci.* **39**, 297–308.

J. Krajíček & P. Pudlák (1998). Some consequences of cryptographical conjectures for S_2^1 and EF. *Inform. and Comput.* **140**, 82–94. Preliminary version in D. Leivant (ed.), *LCC '94*, Lecture Notes in Comput. Sci. 960, Springer, 1995.

B. Krishnamurthy (1985). Short proofs for tricky formulas. *Acta Inform.* **22**, 253–275.

R. Raz (2002). Resolution lower bounds for the weak pigeonhole principle. In *Proc. ACM Symposium on Theory of Computing (STOC)*, to appear.

A. Razborov (2002). Improved resolution lower bounds for the weak pigeonhole functional principle. *Theoret. Comput. Sci.*, to appear.

G. Stalmark (1996). Short resolution proofs for a sequence of tricky formulas. *Acta Inform.* **33**, 277–280.

Manuscript received 31 January 2000

Maria Luisa Bonet
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Jordi Girona Salgado 1-3
08034 Barcelona, Spain
bonet@lsi.upc.es

Nicola Galesi
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Jordi Girona Salgado 1-3
008034 Barcelona, Spain
galesi@lsi.upc.es

Department of Computer Science
University of Toronto
10 King's College Road
Toronto, Ontario
Canada M5S 3G4
galesi@cs.toronto.edu