A Study of Proof Search Algorithms for Resolution and Polynomial Calculus

Maria Luisa Bonet* Nicola Galesi †
Universitat Politecnica de Catalunya
Department de Llenguatges i Sistemes Informatics
Jordi Girona Salgado 1-3 Barcelona Spain
e-mail{bonet,galesi}@lsi.upc.es

Abstract

This paper is concerned with the complexity of proofs and of searching for proofs in two propositional proof systems: Resolution and Polynomial Calculus (PC). For the former system we show that the recently proposed algorithm of BenSasson and Wigderson [2] for searching for proofs cannot give better than weakly exponential performance. This is a consequence of showing optimality of their general relationship reffered to in [2] as size-width trade-off. We moreover obtain the optimality of the size-width trade-off for the widely used restrictions of resolution: regular, Davis-Putnam, negative, positive and linear. As for the second system, we show that the direct translation to polynomials of a CNF formula having short resolution proofs, cannot be refuted in PC with degree less than $\Omega(\log n)$. A consequence of this is that the simulation of resolution by PC of Clegg, Edmonds and Impagliazzo [11] cannot be improved to better than quasipolynomial in the case we start with small resolution proofs. We conjecture that the simulation of [11] is optimal.

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 the main open problem in complexity theory: $P = NP$? (see [12]). But also proving superpolynomial lower bounds is 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. One the complexity of the smallest possible proof, which might be exponential in the size of the tautology, and the second the complexity of the proof search.

Respect to the first problem, Cook-Reckhow [12] proved that $NP \neq Co-NP$ is equivalent to the statement that for every possible propositional proof system, there is a class of tautologies that require superpolynomial size proofs (in the size of the tautology). This means that most probably no propositional proof system can prove all tautologies efficiently, otherwise P would be equal to NP, which we believe 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. In Bonet,Pitassi and Raz [5] a notion of automatizability is defined. We say 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 [15, 5, 6] 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 are some algorithms to find proofs in some proof systems. For instance, [2, 1] gave algorithms for resolution, and [11] for polynomial calculus. The algorithms of [2, 1] are both weakly exponential for resolution, and [11] is polynomial for the system of polynomial calculus. Therefore polynomial calculus is automatizable. In this paper we study the performance of the algorithm proposed in [2] for finding resolution refutations. We also compare this algorithm with that of [11] based on the Grobner basis algorithm for finding proofs in Polynomial Calculus.

Ben-Sasson and Wigderson in [2] introduced a new complexity measure for Resolution refutations. The *width* of a refutation is defined as the maximal number of literals in any clause of the refutation. The importance of this new measure is twofold. On one side they were able to give a general relationship between the width and the length of a refutation, reducing the problem of giving lower bounds on the length to that of giving lower bounds on the width. The

[*] Partly supported by Project CICYT, TIC 98-0410-C02-01
[†] Partly Supported by an EC grant under the TMR project

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 side they 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. In this paper we are faced with the following question (also stated in [2] as an open problem). Can the width-size trade-off be improved ? We give a negative answer to this question showing that the result of [2] is optimal. Namely for a given 3-CNF F over $O(n^2)$ variables we show that: (1) F has a polynomial size resolution refutation; and (2) Any resolution refutation of F requires a clause of size $\Omega(n)$. The main consequence of our result is that the proof search algorithm of [2] is not going to be efficient for finding resolution refutations. Another interesting open question is whether for some known restrictions of Resolution (that lie between tree-like and general resolution) it is possible to improve the size-width trade-off given for general resolution. Here we also show that it is not the case for the following restrictions of Resolution: regular, Davis-Putnam, positive, negative and linear Resolution. Finally, [2] also gave a width-size trade-off for tree-like resolution. [3] have proved that for this restriction of resolution, the trade-off is also optimal.

Clegg et al. [11] defined a new propositional proof system, that later has been called polynomial calculus (PC). This is an algebraic system for refuting CNF formulas translated to polynomial equations. In their paper they gave an efficient proof search algorithm based on the Groebner basis algorithm. Their proof complexity measure was the maximum degree of the polynomials used in the proof, and the maximum degree had to be a constant. Comparing the work of [2] and [11] it is easy to observe that basically the width complexity measure is for Resolution what the degree complexity measure is for PC. Hence, given the optimality result for the width-size trade-off, it is natural to ask whether some polynomial formulation of the same formula used to obtain the optimality result, requires degree $\Omega(n)$ PC refutations. This would mean that there is a formula easy to refute in Resolution but for which both the Grobner basis algorithm and the width based algorithm fail to find proofs quickly. We show a weaker form of the above result: namely that for a given unsatisfiable CNF with polynomial size resolution refutations, its direct translation to polynomials requires PC refutations of degree not less than $\Omega(\log n)$. Therefore proving that the simulation of resolution in [11] cannot be bether than quasipolynomial, in the case we start with a polynomial size resolution refutation.

Our lower bound proof extends the PC lower bound technique introduced by Razborov in [17] to a formula obtained as a modification of the pigeonhole principle defined by Goerdt in [13]. It is hence of independent interest since this technique was known to work only for the PHP formula. We moreover conjecture that our result can be improved, to show that the simulation of [11] is the best possible in the case the resolution proof is small. (Recall that without this last restriction the optimality of this simulation is known given the PC degree lower bounds of [17, 14].)

Observe that, as noticed by A. Wigderson, the performance of the Groebner basis algorithm can be very good if we are able to translate the initial clauses into polynomials of degree 1. For instance the Tseitin graph tautologies require exponential size resolution refutations, but, depending on the translation to polynomials, the degree of the PC proof can be $O(1)$, or has to be linear in the number of variables ([8]). This means that depending on the polynomial translation, the Grobener Basis algorithm can find PC refutations of formulas very quickly, even when these formulas require exponential size Resolution refutations.

In the final section we prove that under a fixed standard translation to polynomials the width based algorithm of [2] cannot have a better performance than the Grobner Basis proof search algorithm of [11]. This is a consequence of a lemma giving a Polynomial Calculus simulation of Resolution.

In section 2 we give some 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 that the simulation of resolution by polynomial calculus cannot be bether than quasipolynomial for small resolution proofs. In section 5 we show how to obtain width lower bounds for resolution from degree lower bounds for polynomial calculus, and some applications of this observation.

2 Preliminaries

RESOLUTION is a refutation proof system for formulas in CNF form with the following inference rule: $\frac{C \vee x \quad D \vee \bar{x}}{C \vee D}$. A Resolution refutation for an inital set Σ of clauses is a derivation of the empty clause from Σ using the above inference rule. Several restrictions of the resolution proof system have appeared in the literature [18]. Here we consider the following five: (1) the REGULAR resolution system in which 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; (2) the DAVIS-PUTNAM resolution system in which 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; (3) the NEGATIVE (resp. POSITIVE) resolution system, or N-resolution (resp. P-resolution) for short, where in each application of the resolution rule one of the premises must not contain any positive (resp. negative) literal; (4) the LINEAR resolution system in which the refutation is a sequence of clauses $(C_0, C_1, \ldots, C_n)$ such that C_0 is an initial clause, C_n is the empty clause and for all i, $1 \leq i \leq n$ in the resolution step $\frac{C_{i-1} \quad B_{i-1}}{C_i}$, the clause B_{i-1} is either an initial clause or such that $B_{i-1} = C_j$ for some $j < i$.

Let $R \vdash F$ (resp. $R \vdash_{tl} F$) denote that R is a general (resp. tree-like) resolution refutation of F. The *size* $|R|$ of a refutation R in any of the above systems is defined as the number of clauses used in R. The size complexity $S(\vdash F)$ (respectively $S_T(\vdash F)$) of deriving a CNF formula F in general resolution (respectively in tree-like resolution) is defined as $\min\limits_{R \vdash F} |R|$ (respectively $\min\limits_{R \vdash_{tl} F} |R|$). Following [2] the *width* $w(F)$ of a CNF formula is defined to be the size (i.e. 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 $w(\vdash F)$ (resp. $w(\vdash_{tl} F)$) of deriving a formula F in general (resp. tree-like) resolution is defined as $\min\limits_{R \vdash F} w(R)$ (resp. $\min\limits_{R \vdash_{tl} F} w(R)$). The size-width relationship obtained in [2] is given by the following theorem:

Theorem 2.1 *([2]) Let F be any unsatisfiable formula over n variables. Then: (1) $S_T(\vdash F) \geq 2^{(w(\vdash_{tl} F) - w(F))}$; (2) $S(\vdash F) \geq \exp(\Omega(\frac{(w(\vdash F) - w(F))^2}{n}))$.*

POLYNOMIAL CALCULUS (PC) is a refutation system for formulas in CNF expressed as a sequence of polynomials over a field K. A PC REFUTATION is a sequence of polynomials ending with 1 such that each line is either an initial polynomial or is inferred from two previous polynomials by the following rules: (1) SUM: $\frac{f \quad g}{\alpha f + \beta g}$ for $\alpha, \beta \in K$: (2) PRODUCT: $\frac{f}{xf}$, for any variable x.

The DEGREE of a refutation is the maximal degree of a polynomial used in the proof. To force 0-1 solutions the axioms $x^2 - x$, for all x, are always included among the initial polynomials. We define a *standard traduction* tr from formulas in CNF to polynomials in the following way: (1) $tr(x) = 1 - x$; (2) $tr(\bar{x}) = x$; (3) $tr(x \vee y) = tr(x) \cdot tr(y)$.

We consider the CNF formula GT_n expressing the negation of the property that in any directed graph closed under transitivity and with no cycles of size two there is a source node. We obtain the following formula easily expressible as a CNF:

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

where the clauses in (1) encode the transitivity closure property, those in (2) the property that there are no cycles of size two and those in (3) say that each node receives at least an edge from some other node (i.e. there is no source node). This formula was first formulated by Krishnamurthy in [16] and then Stalmark in [19] gave polynomial size resolution refutations.

We consider a modification of the pigeon hole principle defined in [13]. Let n be of the form 2^k, for some k and let $m = \log_2 n$ (all the $\log$ are in base 2). For each $j = 1, \ldots m$ let $Part(j)$ the partition of $[n]$ induced by j the following way:

$$\begin{array}{l}
Part(j) := \\
\quad \{\{i, i+1, \ldots, i + (2^j - 1)\} \mid \\
\qquad\qquad i = 1, 1 + 2^j, 1 + 2 \cdot 2^j, \ldots, 1 + (\frac{n}{2^j} - 1) \cdot 2^j\}
\end{array}$$

Definition 2.1 *We say that i and i', in $[n]$ are j-COMPATIBLE if and only if they are in different elements of $Part(j)$.*

The CNF formula defining the modified pigeon hole principle $MPHP_n$ is given by the following clauses, where $x_{i,j}$ means that the pigeon i is sitting in the hole j:

$$\begin{array}{ll}
\bigvee_{j=1}^{m} x_{i,j} & i \in [n] \\
\bar{x}_{i,j} \vee \bar{x}_{i',j} & j \in [m], i \neq i' \in [n] \text{ , not } j\text{-compatible} \\
\bar{x}_{i,j} \vee \bar{x}_{i,k} & i \in [n], j \neq k \in [m]
\end{array}$$

Observe that the clauses defining our $MPHP_n$ are a superset of the clauses defining the $MPHP_n$ of [13]. Goerdt gave in [13] polynomial size unrestricted refutations for $MPHP_n$.

3 Optimality of the width-size Method and its Consequences

In this section we show the optimality of the size-width trade-off of [2]. As formulated in [2] the question is the following: can one find an unsatisfiable k-CNF formula F over n variables such that $w(F)$ is constant, $S(\vdash F) = O(n^{O(1)})$ and $w(\vdash F) \geq \Omega(\sqrt{n})$? We show that a modification of the GT_n formula verifies the properties required to answer the above question not only for unrestricted resolution but also for various other restrictions. We start by giving a resolution refutation for GT_n that fullfills all the considered restrictions (all but N-resolution), and then we discuss how to modify GT_n to obtain our result.

We sligthly modify the proof of [19] in order to show that the upper bound also works for the following restrictions of resolution: regular, positive, Davis-Putnam and linear resolution.

Theorem 3.1 *There are polynomial size refutations of GT_n in the following proof systems: (i) general resolution, (ii) Davis-Putnam resolution, (iii) regular resolution, (iv) positive resolution, (v) linear resolution.*

Proof. We start by giving the general resolution refutation, then we discuss why this proof falls in any of the restricted versions of resolutions We adopt the following abbreviations. Let:

$$\begin{array}{ll}
A(i,j,k) := x_{i,j} \wedge x_{j,k} \rightarrow x_{i,k} & i \neq j \neq k \in [n] \\
B(i,j) := (x_{i,j} \rightarrow \bar{x}_{j,i}) & i \neq j \in [n] \\
C_m(j) := \bigvee_{i=1, i\neq j}^{m} x_{i,j} & \left\{ \begin{array}{l} j \in [n] \\ m \in [n] \end{array} \right. \\
C_m := \bigwedge_{j=1}^{n} C_m(j) & m \in [m] \\
D_{k-1}^{j}(i) := C_{k-1}(j) \vee \bar{x}_{i,k} & \left\{ \begin{array}{l} k \in [n]/\{1\} \\ i \in [k-1] \\ j \in [n] \end{array} \right. \\
E_{k-1}^{j}(i) := (C_{k-1}(j) \vee \bigvee_{\ell=i}^{n} x_{\ell,k}) & \left\{ \begin{array}{l} k \in [n]/\{1\} \\ i \in [k-1] \\ j \in [n] \end{array} \right.
\end{array}$$

The proof proceeds by steps downward from n to 2. At the k-th step, for each $j = 1, \ldots, n$, we prove $C_{k-1}(j)$ using the inital clauses $A(1,k,j)$, $B(k,j)$ and the clauses $C_k(j)$ and $C_k(k)$ obtained at the previous step. At the end we have proved C_2 from which a contradiction is obtained in 2 steps using $B(1,2)$. Now we give a description of how to perform the k-th step obtaining in parallel the clauses $C_{k-1}(1), C_{k-1}(2), \ldots, C_{k-1}(n)$. For a generic value $j \in [n]$ we obtain $C_{k-1}(j)$ by the following steps:
(a): Perform in parallel the following resolutions steps, each one resolving the variable $x_{k,j}$:

$$\begin{array}{ll}
(1)\ \frac{C_k(j) \quad A(1,k,j)}{D_{k-1}^{j}(1)} & (2)\ \frac{C_k(j) \quad A(2,k,j)}{D_{k-1}^{j}(2)} \\
\vdots & \\
(j-1)\ \frac{C_k(j) \quad A(j-1,k,j)}{D_{k-1}^{j}(j-1)} & (j)\ \frac{C_k(j) \quad B(j,k)}{D_{k-1}^{j}(j)} \\
(j+1)\ \frac{C_k(j) \quad A(j+1,k,j)}{D_{k-1}^{j}(j+1)} & \ldots (n)\ \frac{C_k(j) \quad A(n,k,j)}{D_{k-1}^{j}(n)}
\end{array}$$

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

$$\begin{array}{ll}
(1)\ \frac{C_k(k) \quad D_{k-1}^{j}(1)}{E_{k-1}^{j}(1)} & (2)\ \frac{E_{k-1}^{j}(1) \quad D_{k-1}^{j}(2)}{E_{k-1}^{j}(2)} \\
\ldots\ldots\ldots & (k-1)\ \frac{E_{k-1}^{j}(n) \quad D_{k-1}^{j}(k-1)}{C_{k-1}(j)}
\end{array}$$

It is easy to see that such a refutation is a Positive resolution, indeed at each resolution step one of the involved clauses is always made by positive literals.
It is also easy to see that the following order of elimination of the variables is respected:

$x_{n,1}, x_{n,2}, \ldots, x_{n,n-1}$
$x_{1,n}, x_{2,n}, \ldots, x_{n-1,n}$
$x_{n-1,1}, x_{n-1,2}, \ldots, x_{n-1,n}$
$x_{1,n-1}, x_{2,n-1}, \ldots, x_{n-2,n-1}$
$\vdots$
$x_{2,1}$
$x_{1,2}$

Therefore the refutation is Davis-Putnam as well as Regular.
To see that the refutation is Linear observe that the following sequence of clauses defines the order of the linear elimination:

$$\begin{array}{l}
C_n(n), \\
\quad C_n(1), B(n,1), A(2,n,1), \ldots, A(n-1,n,1) \\
\qquad D_n^1(1), \ldots, D_n^1(n), E_n^1(1), \ldots, E_n^1(n), \\
\quad C_n(2), A(1,n,2), B(n,2), \ldots, A(n-1,n,2), \\
\qquad D_n^2(1), \ldots, D_n^2(n), E_n^2(1), \ldots, E_n^2(n), \\
\quad \vdots \\
\quad C_n(n-1), A(1,n,n-1), \ldots, A(n-2,n,n-1), B(n,n-1) \\
\qquad D_n^{n-1}(1), \ldots, D_n^{n-1}(n-1), E_n^{n-1}(1), \ldots, E_n^{n-1}(n-1), \\
C_{n-1}(n-1), \\
\quad C_{n-1}(1), B(n-1,1), A(2,n-1,1), \ldots, A(n-2,n-1,1), \\
\qquad D_{n-1}^1(1), \ldots, D_{n-1}^1(n-1), E_{n-1}^1(1), \ldots, E_{n-1}^1(n-1), \\
\quad C_{n-1}(2), A(1,n-1,2), B(n-1,2), \ldots, A(n-2,n-1,2), \\
\qquad D_{n-1}^2(1), \ldots, D_{n-1}^2(n-1), E_{n-1}^2(1), \ldots, E_{n-1}^2(n-1) \\
\quad \vdots \\
\quad C_{n-1}(n-2), \ldots, A(n-3,n-1,n-2), B(n-1,n-2) \\
\qquad D_{n-1}^{n-2}(1), \ldots, D_{n-1}^{n-2}(n-1), E_{n-1}^{n-2}(1), \ldots, E_{n-1}^{n-2}(n-1), \\
C_{n-2}(n-2), \\
\vdots \\
\{\}\ \square
\end{array}$$

It is easy to observe that in the above refutation there are clauses of size $O(n)$. We show below that in fact any refutation of GT_n must have clauses of such width. But unfortunately the initial clauses of GT_n are also of linear size in n so that we cannot obtain optimality of the size-width trade-off. We consider a modification of the formula GT_n, MGT_n, with approximately the same number of variables as GT_n, such that: (1) $w(MGT_n) = 3$; (2) from a refutation of GT_n we can easily find a refutation of MGT_n, and (3) $w(\vdash MGT_n) \geq \Omega(n)$. To define MGT_n, we introduce for each $j \in [n]$, n new variables $y_{0,j}, \ldots y_{j-1,j}, y_{j+1,j}, \ldots y_{n,j}$ and substitute the clauses in (3) by the following clauses:

$$(3') \quad \bar{y}_{0,j} \wedge \bigwedge_{i=1, i\neq j}^{n} (y_{i-1,j} \vee x_{i,j} \vee \bar{y}_{i,j}) \wedge y_{n,j}$$

Theorem 3.2 *There are polynomial size refutations for the formula MTG_n in any of the following proof systems: (i) general resolution, (ii) positive resolution, (iii) Davis-Putnam resolution, (iv) regular resolution, (v) linear resolution.*

Proof. The proof proceeds the following way. From the clauses in $(3')$ obtain the clauses in (3) eliminating the y variables. Then we apply the polynomial size proof for GT_n to these new clauses. Observe that the first part of the proof is in fact a tree-like proof of size quadratic in n and, since the y variables are different for different $j \in [n]$, 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:

$y_{0,1}, \dots, y_{n,1}$,
$y_{0,2}, \dots, y_{n,2}$,
$\vdots$
$y_{0,n}, \dots, y_{n,n}$,

Moreover if for each $j \in [n]$ we start by eliminating the $y_{j,n}$ variable it is easy to see that the new first part is also a positive resolution. Finally, to prove that this proof is a linear resolution proof, consider for $j = 1, \dots, n$ the following definition:

$$G_j(i) = \begin{cases} y_{n,j} & \text{if } i = n \\ (x_{n,j} \vee x_{n-1,j} \dots x_{i,j} \vee y_{i-1,j}) & i = 1, \dots, n-1 \\ C_n(j) & \text{if } i = 0 \end{cases}$$

Then the order of the clauses in the linear resolution of MTG_n is obtained from the order of the linear resolution for GT_n by putting for each $j = 1, \dots, n$ the sequence of clauses $G_j(n), \dots G_j(1)$ just before the clause $C_n(j)$. □

Theorem 3.3 *There is a $3-$CNF formula F on $O(n^2)$ variables verifying the following two properties: (1) F has polynomial size resolution refutations; (2) any resolution refutation of F contains a clause having at least $\Omega(n)$ variables.*

This theorem is an immediate consequence of Theorem 3.2 and the following theorem.

Theorem 3.4 *Any resolution proof of MTG_n must have a clause of size $\Omega(n)$.*

First we introduce the notion of critical truth assignment for the formula GT_n. A critical truth assignment is a *linear* directed acyclic graph over n distinct nodes and closed under transitivity. The idea is that if the variable $x_{i,j}$ corresponds to whether there is a directed edge (i, j) in the graph, then such a linear graph falsifies only one among the initial clauses in (3). This is because the graph is closed under transitivity, there are no cycles, and every node except for the first one in the line has a predecessor. A critical truth assignment can be also defined by the adjacency matrix of such linear graphs (observe that the diagonal elements are not present). The assignments can be obtained by the following algorithm: chose an index $j_1 \in [n]$ and put 0s in all the positions of column j_1 and all 1s in the empty positions of row j_1 of the matrix. ¿From the remaining indexes of $[n]$ choose another index j_2 and put all 0s in the empty positions of the column j_2 and all 1s in the empty positions of the row j_2. Repeat this process until the matrix is full. We call such an assignment a j_1-critical assignment.

Let B_j be the formula in $(3')$. A j-critical assignment for MTG_n is defined the following way: first we give a j-critical assignment for the x's variables and then we assign values to the y's in such a way as to make false only the formula B_j and true all the other B_k's for $k \neq j$.

Let A_j be the conjunction of the clauses $\bar{x}_{i,j} \vee \bar{x}_{j,i}$ for all $i \in [n], i \neq j$. Consider the formula C_j defined as the conjunction of $A_j \wedge B_j$ and let $Vars(j)$ be the set of variables of C_j, that is, in $Vars(j)$ we have all the variables that mention the node j.

Proof of Theorem 3.4
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 MTG_n, $\mu(C)$ is the size of the minimal $I \subseteq [n]$ such that all critical truth assignments satisfying C_I also satisify C. $\mu(C_i) \leq 1$, $\mu(\{\}) = n$, and μ is obviously subadditive w.r.t. the resolution rule, therefore in any resolution proof of MTG_n there is a clause, say C such that $\frac{n}{3} \leq \mu(C) \leq \frac{2n}{3}$. We show that this clause will contain $\geq \frac{n}{6}$ literals. Assume for the sake of contradiction that $|C| < \frac{n}{6}$. First of all notice that since $\mu(C) \geq \frac{n}{3}$ the following claim holds:

Claim 3.1 *There exists at least an $l \in I$ such that no variable from $Vars(l)$ belongs to C.*

Proof of the Claim
Each variable $x_{i,j}$ belongs to two differents sets $Vars(i)$ and $Vars(j)$. In the worst case all the variables in C mention different nodes so that we capture at most $\frac{2n}{6}$ different sets $Vars(\cdot)$. Since $I > \frac{n}{3}$, then there is at least an index in I verifying the claim. □
Consider any critical assignment α such that $\alpha(C_l) = 0, \alpha(C) = 0$ and for all $j \in I/\{l\}$ $\alpha(C_j) = 1$. This assignment must exist by the minimality of I and moreover it satisfies all the clauses C_i for $i \in [n]/\{l\}$. Define $J = [n]/I$. We have that $|J| > \frac{n}{3}$ (since $|I| \leq \frac{2n}{3}$). Therefore by the same argument used in Claim 3.1 for the set I we deduce that there is at least a $j \in J$ such that no variable from $Vars(j)$ appears in C. We build an assignment β from α such that $\beta(C_i) = 1$ for all $i \in I$ and $\beta(C) = 0$, and this is a contradiction. β is built the following way: change all $x_{i,j}$ such that $\alpha(x_{i,j}) = 1$ to 0. Change all the symmetric values $x_{j,i}$ such that $\alpha(x_{j,i}) = 0$ to 1. This first change does not affect the value of C since no variable from $Vars(j)$ appears in C. Observe that after this change the variable $x_{j,l}$ will have the value 1. Therefore it remains to change consistently the values of the variables $y_{i,l}$ in such a way to satisfy C_l (i.e. such that $\beta(C_l) = 1$). This last change will not affect the value of C since no variable from $Vars(l)$

appears in C. Also notice that the variables $y_{i,s}$ for $s \neq l$ don't need to change value in β. The existence of this β leads to a contradiction. □

Our result has several consequences. First of all the width-size relationship of [2] for tree-like resolution together with Theorem 3.4 give a lower bound of $2^{\Omega(n)}$ for tree-like resolution proofs of MTG_n. Theorem 3.2 gives another exponential separation between unrestricted resolution and tree-like resolution as in [4, 2].

Theorem 3.5 *Any tree like resolution proof of MTG_n must have size $\Omega(2^n)$.*

We also obtain other consequences with respect to other restrictions of resolution. As we have seen in Section 2, the width-size trade-off is more powerful in the tree-like case than in the unrestricted one. This fact lead us to think that (possibly) restricting some way the resolution system it is possible to give better trade-off results than in the unrestricted case. We show that this is not the case for regular, positive, negative, Davis-Putnam and linear resolution. As we have seen MTG_n has also polynomial size refutations in all the considered restrictions. By Theorem 3.4 any resolution refutation of MTG_n (in particular in any of the considered restrictions) must have a clause of size $\Omega(n)$. This immediately implies that the width-size trade-off for general resolution cannot be improved for regular, positive, Davis-Putnam and linear resolution. In the case of negative resolution, we consider the unsatisfiable formula $\overline{MTG_n}$ in which the $x_{i,j}$ variables are replaced by $\bar{z}_{i,j}$ whose intended meaning is opposite to that of the x variables. It is easy to see that the positive resolution proof for MTG_n is in fact a negative resolution proof for $\overline{MTG_n}$ and that the lower bound technique can also be applied. Therefore also in the case of negative resolution we cannot improve the width-size trade-off obtained for unrestricted resolution by [2].

4 Lower Bounds for the Polynomial Calculus

In this section we show that any polynomial calculus refutation of the $MPHP_n$ requires degree $\Omega(\log n)$. We will use the same technique as [17, 14]. Recall from Section 2 that $m = \log n$, and the definition of j-compatible pigeons. Given $Q_i := 1 - \sum_{j\in[m]} x_{i,j}$ we adopt the following polynomial formulation of the $MPHP_n$:

$$\begin{array}{lll} (1) & Q_i = 0 & i \in [n] \\ (2) & x_{i,j}x_{i,k} = 0 & i \in [n],\ j,k \in [m] \\ (3) & x_{i,j}x_{k,j} = 0 & j \in [m],\ i,k \in [n] \text{ not } j\text{-compatible} \\ (4) & x_{i,j}^2 - x_{i,j} = 0 & i \in [n],\ j \in [m] \end{array}$$

For a polynomial x which is a product of $x_{i,j}$, let $Pigeons(x, j)$ be the set of i's such that $x_{i,j}$ is a factor in x.

Definition 4.1 *T is the set of the polynomials $x = x_{i_1,j_1}\dots x_{i_l,j_l}$ such that all i_k are distinct and for all $j_k \in [m]$ and for all i and i' in $Pigeons(x, j_k)$ i and i' are j_k-compatible.*

Using the identities (2), (3) and (4) any polynomial can be represented as a linear combination of polynomials in T. Therefore any polynomial calculus refutation carried on modulo the ideal I generated from the polynomials (2), (3) and (4), is in the vector space $Span(T)$ generated from T. From now on we assume that all the computations are modulo the ideal I.

We want to build a basis B_d for the vector space $Span(T)$ such that the elements of B_d are products of the form $\prod_{i,j} x_{i,j} \prod_i Q_i$. As in [14] (and [17]) the definition of B_d is obtained from a process that maps partial assignments into partial assignments: the pigeon dance. We consider a dummy hole 0, and we represent elements of B_d as partial assignments according to the following definition

Definition 4.2 *A is the set of the partial mappings a from $[n]$ to $[m] \cup \{0\}$ such that for all $i, i' \in [n]$, $i \neq i'$, if $a(i) = a(i') = j \neq 0$ then i and i' are j-compatible.*

Let $A_d := \{a \in A : |a| \leq d\}$. For $a \in A$ with $a = \{(i_1, j_1), \dots (i_k, j_k), (i'_1, 0), \dots, (i'_l, 0)\}$, $\hat{a}$ denote the restriction $\{(i_1, j_1), \dots (i_k, j_k)\}$ of a. Any element a of A defines a polynomial x_a the following way: $x_a = \prod_{a(i)=j, j\neq 0} x_{i,j} \prod_{a(i)=0} Q_i$. Therefore by definition of T any polynomial $x_{\hat{a}}$ associated to $\hat{a} \in A_d$ is in T_d.

Our pigeon dance differs from that of [17, 14] since sometimes a pigeon can be sent to an occupied hole. Consider the following definition:

Definition 4.3 *Given $a \in A$, we say that a hole j* is Good for the pigeon *i* in *a and we write $j \in Good(i, a)$ if $j > a(i)$ and the following condition hold: (1) either there is no $i' \in [n]$ such that $a(i') = j$ (i.e. the hole is unoccupied), or (2) for all $i' \in a^{-1}(j)$, i and i' are j-compatible.*

Our pigeon dance acts the following way: given an $a \in A$ and starting from the first pigeon in $dom(a)$ we try to move all the pigeons i in $dom(a)$ into a hole j different and strictly greater than $a(i)$ which is good for i in a.

Definition 4.4 (Dance) *Let $a \in A$ and consider $dom(a)$. A pigeon dance on a is a sequence of mappings $a_0, a_1, \dots a_n$ in A with the same domain as a, defined the following way: $a_0 = a$ and for all $0 < t \leq n$, if $a(t)$ is undefined, then $a_t = a_{t-1}$, otherwise*

$$\begin{cases} a_t(j) = a_{t-1}(j) & j \neq t \\ a_t(t) \in Good(t, a_{t-1}) & \end{cases}$$

Definition 4.5 (Minimal Dance) *Let $a \in A$ be given and let t be a pigeon index in $[n]$. By $D_t(a)$ we denote a mapping $b \in A$ such that $dom(b) = dom(a)$, and defined as follows:*

$$\begin{array}{ll} b(i) = a(i) & i \in dom(a), i \neq t \\ b(t) = min_{j \in [m]}[j \in Good(t, a)] & \end{array}$$

If $min_{j \in [m]}[j \in Good(t, a)]$ does not exists, then $b(t)$ is undefined. The MINIMAL PIGEON DANCE *on a is: $D_n(D_{n-1}(\cdots(D_1(a))\cdots)$*

The minimal dance has two main properties. It can be always defined whenever a dance is defined, and it defines a one-to-one mapping from partial assignments to partial assignments. We show these properties in the following lemmas.

Lemma 4.1 *If there exists a dance on a, then there always exists a minimal dance on a.*

Proof. We prove by induction on $t = 1, \ldots, n$ that there is dance $b = b_0, b_1, \ldots, b_n$ where $b_0 = a$ such that its first t steps correpond to the first t steps of the minimal dance on a. The lemma hence follows for $t = n$. Assume to have proved the claim for $t - 1$, and let $b = b_0, b_1, \ldots, b_n$ the correct dance having the first $t - 1$ steps as in the minimal dance. We show how to build a new correct dance $c = c_0, c_1, \ldots, c_n$ having its first t steps as in the minimal dance. Let $j_{min} = \min_{j \in [m]}[j \in Good(t, b_{t-1})]$ and suppose $j = b_t(t)$. Observe that that since b is a correct dance, then j_{min} always exists and moreover $j_{min} \leq j$. Now if $j = j_{min}$, then b is making the right choice at the t-th step. In this case we define $c = b$. Otherwise $j_{min} < j$. In this case we define c the following way: for all i, $i = 1, \ldots, t-1$, $c_i = b_i$; for all $i \geq t$ we define first $c_i(i)$ the following way:

$$c_i(i) = \begin{cases} j_{min} & i = t \\ j & i > t \text{ s.t } b_i(i) = j_{min} \wedge \\ & i, t \text{ are not } j_{min}\text{-compatible} \\ b_i(i) & \text{otherwise} \end{cases}$$

We complete the definition of the c_i for $i \geq t$ as follows $c_i(j) = c_{i-1}(j)$ for $j \neq i$.

We have to prove that c is a correct dance, since the minimality is given by the definition of j_{min}. To prove that c is a correct dance we claim that:

Claim 4.1 *(1) there is no $i < t$ such that $b_t(i) = j_{min}$ and i and t are not j_{min}-compatible; (2) there could be only one $i > t$ such that $b_i(i) = j_{min}$ and i and t are not j_{min}-compatible; (3) If there is such an $i > t$ (as described in (2)), then j is in $Good(i, c_{i-1})$.*

The correctness of the dance c then follows by its definition.
Proof. (of Claim4.1)

The first point holds since otherwise $j_{min} \notin Good(b_{t-1}, t)$ (recall that $b_{t-1}(i) = b_i(i)$ for all $i \leq t-1$). The second point is also easy. Indeed if there exist two different pigeons i and i' both not j_{min}-compatible with t then, they are all three in the same elements of the partition of $[n]$ induced by j_{min}. But this is not possible since b is a correct dance and therefore i and i' must be j_{min}-compatibles. For the third point, we have to show that $j \in Good(i, c_{i-1})$. If $c_{i-1}^{-1}(j) = \emptyset$, then the result is immediate. Otherwise assume that $c_{i-1}^{-1}(j) \neq \emptyset$. We show that for any $i' \in c_{i-1}^{-1}(j)$ (i.e $c_{i'}(i') = j$), i and i' are j-compatible, from which the claim follows. Assume for sake of contradiction that i and i' are not j-compatible, we show the contradiction that b was not a correct dance. Since i and i' are not j-compatible, then they are in the same group B of $Part(j)$. By the point (2) i is the only node (except for t) for which we will modify $b(i)$. Therefore i' was already sent to j in b, i.e. $b_{i'}(i') = j$. We show that $t \in B$ from which follows the contradiction since in b we would have $b_t(t) = j$ and $b_{i'}(i') = j$ for two element in B i.e. not j-compatible. Finally to see that $t \in B$ only observe that t and i are j_{min}-compatible and $j_{min} < j$. Therefore t and i must be in the same group of $Part(j)$. Since $i \in B \in Part(j)$ then $t \in B$. □

Lemma 4.2 *The minimal dance is a one-to-one mapping.*

Proof. We show that for all $t = 1, \ldots, n$, $D_t(\cdot)$ is a 1-1 mapping. The result then follows since the minimal dance is a composition of the D_t mappings. We show that if $D_t(a) = D_t(a')$ then $a = a'$. Suppose $D_t(a) = D_t(a')$. Then $dom(a) = dom(a')$ and in particular $a(i) = a'(i)$ for all $i \in dom(a), i \neq t$. It remains to show that $a(t) = a'(t)$. We show that neither $a(t) < a'(t)$ nor $a'(t) < a(t)$. Suppose the former. We show the following contradiction:

$$D_t(a)(t) \leq a'(t) < D_t(a')(t) = D_t(a)(t)$$

To justify the first equality observe that $a'(t) \in Good(a, t)$ since $Good(a', t) = Good(a, t)$ (this is since $a(i) = a'(i)$ for all $i \neq t$) and $D_t(a)(t) = \min_{j \in [m]}[j \in Good(a, t)]$. The second inequality holds by definition of minimal dance. The other case $a'(t) < a(t)$ is completely symmetric. □

A property of any pigeon dance which ends succesfully on an $a \in A$ is that the polynomial associated to the dance is in T (this is because we are moving to strictly greater holes and therefore at the end the dummy hole 0 has disappeared).

Lemma 4.3 *If $d \leq \frac{\log n}{3}$ and $a \in A_d$, then there exists a dance on a if and only if there exists a dance on $\hat{a}$.*

Proof. If there is a dance for a then obviously there is a dance for $\hat{a}$, so that one implication is easy. For the other implication assume that the number of Q factors in x_a is different from 0 since otherwise there is nothing to prove.

Now, the worst case for the dance on $\hat{a}$ is when all the holes referred to in $\hat{a}$ are different and the dance is assigning always a new hole to each pigeon in $dom(\hat{a})$. Since there are $m = \log n$ holes and since $d \leq \frac{\log n}{3}$, then the dance on $\hat{a}$ leaves at least $\frac{\log n}{3}$ holes unused. These holes are the nodes we will use to define a dance on the whole a. That is, if the pigeon i is in $dom(a)$, then $a(i) = \hat{a}(i)$. If the pigeon $i \in dom(a) - dom(\hat{a})$, then we assign one of the unused holes to $a(i)$. Since these are new holes and since $|dom(a)| - |dom(\hat{a})| \leq d \leq \frac{\log n}{3}$, then the dance on a is well defined. □

We can now proceed to the definition of the basis B_d.

Definition 4.6

$$B_d = \{x_a : a \in A_d \textit{ there is a dance on } \hat{a}\}$$

It is easy to prove that the following monotonicity properties hold for B_d: (1) $B_{d-1} \subseteq B_d$; (2) $x_a \in B_{d-1}$ if and only if for all $i \notin dom(a)$, $x_aQ_i \in B_d$. In order to show that B_d is a basis for $Span(T_d)$ we need to define an order $\prec$ on polynomials in T_d. We will do it as in [14].

Definition 4.7 *Let x_a and x_b be two polynomials in T_d. then $x \prec y$ if and only if $deg(x_a) < deg(x_b)$, or if $deg(x_a) = deg(x_b)$, then for the largest pigeon i such that $a(i) \neq b(i)$, we have that $a(i) < b(i)$.*

Lemma 4.4 *B_d is a basis for $Span(T_d)$ for any $d \leq \frac{\log n}{3}$.*

Proof. Under the hypothesis of the Lemma, we show: (1) that $|B_d| \leq |T_d|$ and (2) that any $x_a \in T_d$ can be expressed as a linear combination of elements of B_d, from which the Lemma follows. The first property is a consequence of the fact that the set B_d is in 1-1 correspondence with the set T_d via the minimal dance. More precisely, if $x_a \in B_d$ then we have a dance on $\hat{a}$ and since $d \leq \frac{\log n}{3}$, then by Lemma 4.3, there is dance on a and therefore by Lemma 4.1 there is a minimal dance on a that by Lemma 4.2 is a 1-1 mapping. By the property discussed above of the dance that ends correctly we then obtain the first part. For the second part we work by induction on $\prec$. Assume that for all $x' \prec x_a$ $x' \in Span(B_d)$, we show that $x_a \in Span(B_d)$. If there is a dance on a then x_a is in B_d. Otherwise we show how to express x_a as a linear combination of the elements of B_d. Let P_t be the set of all possible correct first t steps of the dance on a. We prove that $x_a \in Span(B_d)$ iff $\sum_{b \in P_t} x_b \in Span(B_d)$ by induction on $t = 0, \ldots, n$. Since there is no dance on a, then $P_n = \emptyset$ and therefore the claim follows. The base of the induction $t = 0$ follows since $P_0 = a$. For the induction step observe that if $t \notin dom(a)$ then $P_t = P_{t-1}$ and so the claim follows by induction on t. Otherwise for any $b \in P_{t-1}$, x_b is of the form $x_{t,j}x_c$. We rewrite $x_{t,j}$ with respect to the relation Q_t, so that x_b can be rewritten as

$$(1) \quad x_c - x_cQ_t - \sum_{j' \neq j} x_c x_{t,j'}$$

Observe that all the terms $x_c x_{t,j'}$ such that j' is not in $Good(t, b)$ are equals 0 so that the above sum can be written as

$$x_c - x_cQ_t - \sum_{j'<j, j' \in Good(t,b)} x_c x_{t,j'} - \sum_{j'>j, j' \in Good(t,b)} x_c x_{t,j'}$$

Observe that the first three terms in the above sum are in $Span(B_b)$. The first by induction on $\prec$. The second by induction on $\prec$ and by the monotonicity property of B_d and the third by (the second case of the definition) $\prec$. The fourth term correspond exactly to all the possible correct t-th steps of b. Therefore if we sum over all x_b for $b \in P_{t-1}$ we have that $\sum_{b \in P_t} x_b \in Span(B_d)$ iff $\sum_{b \in P_{t-1}} x_b \in Span(B_d)$. This concludes the proof of the Lemma. □

Theorem 4.1 *Any polynomial calculus refutation of $MPHP_n$ has degree not less than $\frac{\log n}{3}$.*

Proof. The proof is as in [14]. That is we prove by induction on the length of the proof that each line in a refutation of $MPHP_n$ can be expressed as a polynomial in $B_d - T_d$. Therefore since $1 \in T_d$ and it has a unique representation in each basis, then we cannot derive the polynomial 1 with a proof of degree less than or equal to d.

Recall that we are considering refutations modulo I. Therefore if a line is an axiom it is Q_i for some $i \in [n]$, and the claim follows. If a line is inferred by the sum rule the result is immediate. For the case of product, say we have $\frac{x_a}{x_a x_{i,j}}$. Therefore $|a| \leq d - 1$ and $x_a \in B_d - T_d$. By induction we have that x_a is of the form x_bQ_k for some b and k, with $x_b \in B_{d-2}$. To prove that $x_a x_{i,j} = x_b x_{i,j} \prod_k Q_k$ is in $B_d - T_d$, observe that $x_b x_{i,j} \in Span(B_{d-1})$ and therefore we can rewrite it as a sum of elements of B_{d-1}. Now if we multiply each of these terms for Q_k we obtain either 0 (if k is in $dom(b)$) or, by the monotonicity property an element of B_d. Therefore the whole sum is in $Span(B_d)$. □

So far we have proved a $\Omega(\log n)$ degree lower bound for the polynomials (1)-(4) defined at the beginning of this section. The same degree lower bound can be obtained for a different set of polynomals, 3-$MPHP_n$, expressing the same principle. This new set of polynomials is obtained substituting the polynomials $1 - Q_i$ in $MPHP_n$ by the polynomials obtained from the translation tr (see section 2) applied to the set of clauses $\bar{y}_{i,0} \wedge \bigwedge_{j=1}^{m}(y_{i,j-1} \vee x_{i,j} \vee \bar{y}_{i,j}) \wedge y_{i,m}$, where $y_{i,j}$ for $i = 1, \ldots, n$, and $j = 0, \ldots, m$ where $m = \log n$ are new variables.

Theorem 4.2 *Any polynomial calculus refutation of 3-$MPHP_n$ has degree not less than $\frac{\log n}{3}$.*

Proof. We will prove that $MPHP_n$ is (1,3)-reducible to 3-$MPHP_n$ following the definitions of (d_1, d_2)-reductions from [8]. Define $y_{i,j} = 1 - \sum_{k>j}^{m} x_{i,k}$. We prove that all the initial polynomials of 3-$MPHP_n$ (with y substituted as defined above) are derivable with a 3-degree polynomial calculus refutations from initial polynomials of $MPHP_n$. Observe that $y_{i,0} = 1 - Q_i = 0$ and $y_{i,m} = 1$. So we can prove $y_{i,0} = 0$ and $1 - y_{i,m} = 0$. Now a generic initial polynomial of 3-$MPHP_n$ of the form $(1 - y_{i,j-1})(1 - x_{i,j})(y_{i,j})$ for $2 < j < n$, is equivalent to $(\sum_{k>j-1}^{m} x_{i,k})(1 - x_{i,j})(\sum_{k>j}^{m} x_{i,k})$. This can be rewritten as $(1 - \sum_{k=1}^{j-1} x_{i,k})(1 - x_{i,j})(1 - \sum_{k>j}^{m})$. By simple calculations (using the initial axioms of $MPHP_n$) this is equal to$(1 - Q_j) + \sum_{k=1,k\neq j}^{n} x_{i,k}x_{i,j} + (1 - x_{i,j})\sum_{k=1}^{j-1} x_{i,k} \sum_{k=j+1}^{m} x_{i,k}$. Using the inital axioms of $MPHP_n$ it is easy to see that each of the three terms of this polynomial is equal to 0. □

We have found a principle, 3-$MPHP_n$, that has polynomial size resolution refutations, but such that its direct polynomial translation requires $\Omega(\log n)$ degree. Observe that this result can also be obtained using the pigeonhole principle, PHP_n^m, where $m = 2^{\sqrt{n \log n}}$. It is known that PHP_n^m has polynomial size (in m) resolution refutations (see [10]), and on the other hand there is an $O(n)$ degree lower bound for polynomial calculus proofs of it (see [14, 17]).

Our conjecture is that the simulation of [11] is optimal for small resolution proofs. We think that some polynomial version of the formula GT_n should require $\Omega(n)$ degree in PC for some field.

5 Resolution lower bounds via degree lower bounds

The following Lemma shows that degree lower bounds imply width lower bounds as long as the initial polynomias of the PC proofs are a direct translation of the inital clauses of the resolution proofs.

Lemma 5.1 *Given a set of unsatisfiable clauses F and a resolution refutation of F , there is a polynomial calculus refutation of $tr(F)$ of degree less than or equal to $w(\vdash F)$.*

Proof. For a generic clause $A = A^+ \vee A^-$ where $A^+ = (a_{i_1} \vee \ldots \vee a_{i_k})$ and $A^- = (\bar{a}_{j_1} \vee \ldots \vee \bar{a}_{j_l})$, let $poly(A^+) = \prod_{\ell=1}^{i_k} (1 - a_\ell)$ and $poly(A^-) = \prod_{\ell=1}^{j_l} a_\ell$. Then $poly(A) = poly(A^+) \cdot poly(A^-)$. Observe that given two clauses A and B, it is easy to obtain a PC derivation of $Poly(A) = 0, \vdash Poly(A)Poly(B) = 0$ with a degree less than or equal to $w(A) + w(B)$. We show that for each line A in the resolution proof we find a PC refutation of $Poly(A) = 0$. If A is a initial clause the result follows by definition of tr. Now assume that at a resolution step we are in the following situation $\frac{A \vee x \quad \bar{x} \vee B}{A \vee B}$ by induction we have derived $Poly(A)(1 - x) = 0$ and $Poly(B)x = 0$. By the previous observation we can obtain $Poly(A)Poly(B)(1 - x) = 0$ and $Poly(A)Poly(B)x = 0$. Finally by an applycation of addition we obtain $Poly(A)Poly(B) = 0$. □

The previous lemma also shows that the degree lower bound obtained for 3-$MPHP_n$ cannot be improved. In fact [13] shows how to obtain a superpolynomial size resolution refutation of 3-$MPHP_n$ of width $O(\log n)$, and by the lemma there is also a polynomial calculus refutation of the direct translation of degree $O(\log n)$.

As a consequence of the previous lemma and the width-size trade-off [2] (see theorem 2.1), a linear (in the number of variables) degree lower bound in polynomial calculus can give us an exponential lower bound in resolution size.

Finally observe that the previous lemma is better (in the sense that it gives a smaller degree PC refutations) than the correponding simulation lemma of [11] in the case we have constant width polynomial Resolution refutations of formulas having initial clauses of constant size. Moreover it implies that under the tr translation the width base algorithm of [2] cannot be better than the Grobner basis algorithm of [11].

It would be interesting to obtain the opposite direction of lemma 5.1. Buresh-Oppenheim and Pitassi [7] have a simulation of polynomial calculus by resolution when we start with binomial equations as initial polynomials. The simulation has the property that the width is twice the degree.

Acknowledgment

Part of this work was done while N. Galesi was visiting the Dept. of Computer Science of the University of Liverpool.

References

[1] P. Beame, T. Pitassi. Simplified and Improved Resolution Lower Bound. *Proceedings of FOCS 96* pp. 274-282

[2] E. Ben-Sasson, A. Wigderson. Short Proofs are Narrow - Resolution Made Simple. *STOC 1999.*

[3] A. Biswas, S. Moran, T. Pitassi. On Width versus Size of Resolution Proofs. In preparation.

[4] M.L. Bonet, J.L. Esteban, N. Galesi, J. Johannsen. Exponential Separation between Restricted Resolution and Cutting Planes Proof Systems. *Proceedings of FOCS 1998.* pp.638-647.

[5] M. Bonet, T. Pitassi, R. Raz. No Feasible Interpolation for TC^0 Frege Proofs. *Proceedings of FOCS 97*. pp. 254-263.

[6] M. Bonet, C. Domingo, R. Gavalda, A. Maciel, T. Pitassi. Non Automatizability of Bounded Depth Frege Proofs. Proceetings of the *Fourteenth annual IEEE Conference on Computational Complexity.* May 4-6, 1999. Atlanta, Georgia. pp. 15-23.

[7] Buresh-Oppenheim, Josh and Pitassi ,Toniann, Some remarks on the Polynomial Calculus and Resolution proofs. May, 1999.

[8] S. Buss, D. Grigoriev, R. Impagliazzo, T. Pitassi. Linear Gaps Between Degrees for the Polynomial Calculus Modulo Distinct Primes. *STOC 1999.*

[9] S. Buss, R. Impagliazzo, J. Kraijeck, P. Pudlak, A. Razborov, J. Sgall. Proof complexity in algebraic systems and bounded depth Frege systems with modular counting. *Comput. Complexity*, 6(3):256-298, 1996/1997.

[10] S. Buss, T. Pitassi. Resolution and the weak Piegeon hole Principle. *Selected Papers of CSL 1997* LNCS 1414. pp. 149-156, 1997.

[11] M. Clegg, J. Edmonds, R. Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proc. of the 28-th STOC*, pp 174-183 (1996).

[12] S. Cook, R. Reckhow. The relative efficiency of propositional proof systems *Journal of Symbolic Logic* 44 (1979) pp. 36-50.

[13] A. Goerdt. Unrestricted Resolution versus N-Resolution. *Theoretical Computer Science* 93 (1992) pp. 159-167.

[14] R. Impagliazzo, P. Pudlak, J. Sgall. Lower Bounds for the Polynomial Calculus and the Groebner basis Algorithm. ECCC TR97-042. To appear in *Computational Complexity*.

[15] J. Krajicek,P. Pudlak. Some consequences of cryptographical conjectures for S_2^1 and EF, *Logic and Computational Complexity*. Lecture Notes in Computer Science, Vol. 960, (1995), pp.210-220.

[16] B. Krishnamurthy. Short Proofs for Tricky Formulas. *Acta Informatica* 22, 1985 pp. 253-275

[17] Razborov. Lower bounds for the Polynomial Calculus. *Computational Complexity* vol 7 (**4**) pp. 291-324 1998.

[18] U. Schoning. *Logic for Computer Scientists*. Birkhauser, 1989.

[19] G. Stalmark. Short Resolution Proofs for a Sequence of Tricky Formulas. *Acta Informatica* 33, 1996, pp. 277-280