Narrow Proofs May Be Maximally Long (Extended Abstract)

Albert Atserias
Universitat Politècnica de Catalunya
E-mail: atserias@lsi.upc.edu

Massimo Lauria
KTH Royal Institute of Technology
E-mail: lauria@kth.se

Jakob Nordström
KTH Royal Institute of Technology
E-mail: jakobn@kth.se

***Abstract*—We prove that there are 3-CNF formulas over n variables that can be refuted in resolution in width w but require resolution proofs of size $n^{\Omega(w)}$. This shows that the simple counting argument that any formula refutable in width w must have a proof in size $n^{O(w)}$ is essentially tight. Moreover, our lower bounds can be generalized to polynomial calculus resolution (PCR) and Sherali-Adams, implying that the corresponding size upper bounds in terms of degree and rank are tight as well. Our results do not extend all the way to Lasserre, however—the formulas we study have Lasserre proofs of constant rank and size polynomial in both n and w.**

***Keywords*-proof complexity; resolution; polynomial calculus; PCR; Sherali-Adams; Lasserre; size; length; width; degree; rank**

I. Introduction

Proof complexity studies how hard it is to prove that propositional logic formulas are tautologies. While the original motivation for this line of research, as discussed in [27], was to prove superpolynomial lower bounds on proof size for increasingly stronger proof systems as a way towards establishing $\mathsf{NP} \neq \mathsf{co\text{-}NP}$ (and hence $\mathsf{P} \neq \mathsf{NP}$), it is probably fair to say that most current research in proof complexity is driven by other concerns.

One such concern is the connection to SAT solving. By a standard transformation, any propositional logic formula can be converted to another formula in conjunctive normal form (CNF) that has the same size up to constant factors and is unsatisfiable if and only if the original formula is a tautology. Any algorithm for solving SAT defines a proof system in the sense that the execution trace of the algorithm constitutes a polynomial-time verifiable witness of unsatisfiability.[1] In fact, most modern-day SAT solvers can be seen to search for proofs in systems at fairly low levels in the proof complexity hierarchy, and upper and lower bounds for these proof systems hence give information about the potential and limitations of the corresponding SAT solvers. In this work, we focus on such proof systems.

A. Background

The dominant strategy in applied SAT solving today is so-called *conflict-driven clause learning (CDCL)* [7], [47], [48], which is ultimately based on the *resolution* proof system [19]. The most studied complexity measure for resolution is *size* (also referred to as *length*), which gives lower bounds on the running time on CDCL solvers and for which (optimal) exponential lower bounds are known [25], [41], [56]. Another more recently studied measure is *space*, which corresponds to memory usage, and for which (again optimal) linear lower bounds have been proven [1], [12], [30]. For all of these results, the concept of *width*, measured as the size of a largest clause in a resolution proof, has turned out to play a key role. Width was identified as a crucial resource already in [34], and strong lower bounds on the width of resolution proofs have been shown to imply lower bounds on proof size [15] and space [3].

Interestingly, although the relationships and trade-offs between width and space in resolution are by now fairly well-understood [11], [13], as are those between size and space [8], [10], [14], very basic questions about the connections between size and width have remained open. For instance, the argument in [15] that width gives a lower bound on size works by transforming a short resolution proof into a narrow one, but this transformation causes an exponential increase in the size. It is not known whether such a blow-up is necessary, i.e., if there are trade-offs between size and width, or whether the analysis in [15] can be sharpened to show that short proofs can be made simultaneously narrow. Also, as noted in the same paper, an upper bound w on the refutation width for a formula over n variables implies a proof size of at most $n^{O(w)}$ simply by counting the number of possible distinct clauses of width w. Again, it is not clear how tight this argument is—for all standard formula families in the literature known to be refutable in small enough width w there are refutations in size $n^{O(1)}$ independent of the width complexity (in fact, even in size *linear* in the formula size). To the best of our knowledge, it has been open whether there exist formulas refutable in width $w = O(\sqrt{n})$ that require size $n^{\Omega(w)}$, i.e., with the width complexity appearing in the exponent.

From a theoretical point of view, the ubiquity of CDCL in SAT solving is somewhat puzzling since resolution is a quite weak proof system. A different approach is to translate CNF formulas to multilinear polynomials and do Gröbner basis computations, which corresponds to *polynomial calculus resolution (PCR)* as defined in [1], [26]. Intriguingly, although PCR is known to be exponentially

[1] Such a witness is often referred to as a *refutation* rather than a *proof*, and these two terms are sometimes used interchangeably.

stronger than resolution, implementations of search methods for this proof system such as PolyBoRi [21], [22] have a hard time competing with CDCL solvers.

Proof size and space in PCR is defined in analogy with resolution, and the measure corresponding to width of clauses is (total) *degree* of polynomials. It is straightforward to show that PCR can simulate resolution efficiently with respect to all of these measures, meaning that the same worst case upper bounds as in resolution apply to PCR. It was proven in [43] that strong degree lower bounds imply strong size lower bounds, which is a close parallel to the size-width relation for resolution in [15], and this size-degree relation has been employed to prove exponential lower bounds on size in a number of papers, with [2] perhaps providing the most general setting. Optimal (linear) lower bounds on space were obtained in [20] building on [1], [32], but it is worth noting that these bounds are *not* derived from degree lower bounds—it remains unknown whether an analogue of [3] holds for PCR (although [31] recently reported some progress on this and related open questions). Strong trade-offs between size and space as well as between degree and space have been shown in [10], but—again in analogy with resolution—the exact relations between size and degree remains unclear. The same blow-up as in [15] occurs in [43] when small size is converted to small degree, but it is not known whether this is necessary or just an artifact of the proof. Also, by [26] we know that a degree upper bound of d implies proof size at most $n^{\mathrm{O}(d)}$, but it has been open whether this is tight or not.

Yet another way to achieve greater expressivity than in resolution is to translate clauses into linear inequalities and manipulate them using 0-1 linear programming. Perhaps the simplest and most well-known example of this approach is the *cutting planes* proof system introduced in [28] based on ideas in [24], [35]. In this paper, however, we will be interested in somewhat related but different *semialgebraic* methods operating on linear programming relaxations of the CNF translations, such as the *Sherali-Adams*, *Lovász-Schrijver*, and *Lasserre* hierarchies used for attacking NP-hard optimization problems. We discuss this next.

The *Sherali-Adams (SA)* method [55] provides a hierarchy of linear programming relaxations of any given 0-1 integer program. The nth level of the hierarchy, where n is the number of 0-1 integer variables, wipes out the integrality gap and is thus exact, but also leads to an exponential blow-up in problem size. The main point of the method, however, is that any linear function of the variables can be optimized over the kth level of the hierarchy in time $n^{\mathrm{O}(k)}$, and in particular feasibility of the kth level relaxation can be checked in that time. In the context of proof complexity, what this means is that if the kth level relaxation of the integer programming formulation of a CNF formula in infeasible (the minimal such k is known as the *SA-rank* of the integer program), then there is an $n^{\mathrm{O}(k)}$-time algorithm that can detect this. Furthermore, since the kth level of the hierarchy is an explicitly defined linear program, its infeasibility can be certified as a positive linear combination of its defining inequalities. Such a certificate is a rank-k Sherali-Adams refutation of the corresponding CNF formula.

The *Lovász-Schrijver* approach [46] can be thought of as (and indeed it is formally equivalent to) an iterated version of the level-2 SA-relaxation. The point is again that any linear function can be optimized over the linear program after k iterations in time $n^{\mathrm{O}(k)}$. Lovász and Schrijver also introduced a method LS^{+}, which uses semidefinite programming instead of linear programming, and which is significantly stronger in some notable cases of interest in combinatorial optimization.

The *Lasserre* method [44], finally, is basically the Sherali-Adams method with semidefinite programming conditions at all levels of the hierarchy. Again it stratifies into levels and the kth level can be solved in time $n^{\mathrm{O}(k)}$. Moreover, Lasserre's method is the strongest of all three in the sense that, level by level, it provides the tightest of all three approximations of the integer linear program. We refer to [23], [45] for a more detailed discussion of Sherali-Adams, Lovász-Schrijver and Lasserre and a comparison of their relative strength.

In view of the important algorithmic applications that these methods have (see, e.g., [50] and subsequent work), it is a natural question whether the upper bounds $n^{\mathrm{O}(k)}$ for rank k are tight, just as for resolution and polynomial calculus resolution.

From the proof complexity side, some notable early papers investigating semialgebraic proof systems were published around the turn of the millennium [38], [39], [51], but then this area of research seems to have gone dormant. In the last few years, these proof systems have made an exciting reemergence in the context of hardness of approximation, revealing unexpected and intriguing connections between approximation and proof complexity. Some examples of this is the paper [54] essentially rediscovering results from [36], and more recent papers such as [6], [49]. There have also been papers such as [9] and (the very recent) [40] focusing on *semantic* versions of these proof systems, with less attention to the actual syntactic derivation rules used.

B. Our results

The main contribution of this paper is showing that the upper bounds on proof size in terms of width for resolution, degree for PCR, and rank for Sherali-Adams are essentially tight (up to constant factors in the exponent). Moreover, an interesting feature of our result is that we can actually use the *same formula family* to prove tightness simultaneously for all the proof systems. What this means is that we obtain upper bounds on size in resolution that tightly match lower bounds in the much stronger systems PCR and Sherali-Adams (which are in turn tight for these systems since resolution width is an upper bound on both PCR degree and Sherali-Adams rank). The formal statement of this result is as follows.

Theorem 1. *Let $w = w(n)$ be such that $w = O(n^c)$ for some positive constant $c < 1/4$. Then there are 3-CNF formulas $F_{n,w}$ with $\mathrm{O}(wn)$ clauses over $\mathrm{O}(n)$ variables such that the following holds.*

1) *$F_{n,w}$ has a resolution refutation in simultaneous size $n^{\mathrm{O}(w)}$, width $\mathrm{O}(w)$ and space $\mathrm{O}(w)$.*
2) *Any refutation of $F_{n,w}$ in resolution, PCR, or Sherali-Adams must have size $n^{\Omega(w)}$.*

For resolution this actually shows something slightly stronger than that the counting upper bound on size in terms of width is tight. Namely, since the formulas in Theorem 1 have the same asymptotic upper bound on space as on width, it follows that even for formulas of space complexity $\mathrm{O}(w)$—which is a more stringent requirement than width complexity $\mathrm{O}(w)$—it is still impossible to obtain any size upper bound better than $n^{\mathrm{O}(w)}$ in general.

Theorem 1 has an interesting consequence for the analysis of CDCL solver performance, which we state as a formal corollary. By way of background, it was shown in [4] that if a CNF formula F over n variables has a resolution refutation in width w, then with high probability any CDCL solver[2] will only need time $n^{\mathrm{O}(w)}$ to decide that F is indeed unsatisfiable.[3] An obvious question is whether this result is tight. Theorem 1 shows that the answer is "yes," since no CDCL solver can run faster than the shortest resolution proof it can possibly find.[4]

Corollary 2. *There are formulas F over n variables refutable in resolution in width w for which any resolution-based CDCL solver cannot run faster than $n^{\Omega(w)}$, and hence the result in [4] is optimal up to constants in the exponent.*

Another interesting aspect of our lower bound for resolution is in the context of Berkholz's EXPTIME-completeness result for deciding resolution width [16]. What Berkholz showed is that given a formula F over n variables and a parameter w, it cannot be decided in time less than $n^{(w-3)/12}$ whether F has a resolution refutation in width w or not. Optimizing the constants in Theorem 1, we can show that there are 4-CNF formulas refutable in width w for which no resolution refutation can be shorter than $n^{w/2-\mathrm{o}(1)}$. It is worth noting that this bound is stronger than that in [16], although it of course applies only for the more restricted setting where the algorithm has to output a width-w resolution refutation rather than for the general decision problem. Still, we believe this sheds interesting light on Berkholz's result.

C. Discussion of proof techniques

We conclude the overview by outlining the proof of the lower bound in Theorem 1 for resolution and how it differs from previously used methods. At a high level, our proof is a standard restriction argument, but it turns out to have some twists which we believe might be of interest and could be useful elsewhere.[5]

Before going into the details of our new restriction argument, let us revisit previous lower bounds on size in terms of width and see how they fall short of proving what we are after. On the one hand, the result in [15] states that if a 3-CNF formula on n variables requires width w to refute in resolution, then it also requires size $2^{\Omega(w^2/n)}$. This lower bound is vacuous for w smaller than $\sqrt{n}$ and, in any case, can never be larger than $2^{\Omega(w)}$ since w is bounded by n. On the other hand, for formulas refutable in width w smaller than $\sqrt{n}$, a direct random restriction argument can sometimes still be applied to get meaningful lower bounds. The idea is that setting a random literal to true will kill off a $\frac{w}{2n}$-fraction of the wide clauses on average. After r rounds of such restrictions, the expected number of surviving wide clauses is at most $\left(1-\frac{w}{2n}\right)^r S$, where S is the size of the refutation, and choosing $r = (2n/w)\log S$ brings the number of wide clauses down to zero. A contradiction is then derived by showing that the residual formula still requires width w to refute. Note, however, that we cannot apply the restriction for more than n rounds (or else there will be no residual formula to argue about), and so the best size lower bound this method can achieve is again $2^{\Omega(w)}$, which is smaller than the $n^{\Omega(w)}$ bound that we are after.

In some sense, the problem is that using restrictions in the style of Håstad's switching lemma [42] does not work in our setting. Instead, it turns out that a seemingly weaker argument inspired by Furst-Saxe-Sipser [33] is just what we need. Let us now describe this modified restriction argument and how it overcomes the problems discussed above.

We start with a carefully chosen family of formulas $F_{n,w}$ and an associated distribution over random restrictions ρ_n. Then we assume that we have a resolution refutation π of $F_{n,w}$ in size $n^{\mathrm{o}(w)}$ and analyze how a randomly chosen restriction ρ_n affects π. We get two cases:

[2]This result holds for a fairly general mathematical model of what a CDCL solver is, which agrees reasonably well with how state-of-the-art solvers are actually implemented in practice.

[3]Perhaps this might not seem so impressive at first sight—after all, exhaustive search in bounded width runs within this time bound deterministically—but the point is that a CDCL solver is very far from doing exhaustive width search and does not care at all about the existence or non-existence of narrow refutations.

[4]This is of course assuming that the solver does not implement features such as, e.g., cardinality reasoning or extended resolution, since these fall outside of the standard CDCL framework and go beyond resolution-based reasoning.

[5]In fact, in a sense this has already happened in that our paper heavily draws on ideas from [5], which used a similar approach in a very different context.

1) For clauses C in the refutation π that are noticeably wide, ρ_n is very likely to satisfy a literal in C and so the clause disappears.
2) Clauses that are not so wide will not be satisfied by ρ_n, but since they are reasonably small they are very likely to be shortened by ρ to width strictly less than w.

Admittedly, the first case looks no different from the standard restriction argument, and the second case seems quite weak. But the point is that by considering also the second case, we can afford a noticeably bigger bound for "wide" than in the standard argument, thus getting a bigger probability of success. This is the key to the argument. The rest is now standard: $F_{n,w}$ and ρ_n are chosen so that $F_{n,w}$ restricted by ρ_n is a bounded-width version of a pigeonhole principle (PHP) formula with w pigeons that are supposed to fit into $w-1$ holes. Since π is short enough, by a counting argument there is some restriction ρ_n that eliminates all wide clauses to give a resolution refutation of the PHP formula in width less than w. It is a straightforward separate argument that such a narrow resolution refutation cannot exist, and the lower bound on resolution refutation size follows.

The lower bounds for PCR and Sherali-Adams are quite similar. The restriction part of the argument is basically the same, but one has to work a bit harder to prove the final punchline that the restricted refutations have impossibly low degree and rank, respectively.

It should perhaps be stressed that while the final argument is quite straightforward and natural (at least for resolution), a crucial component in the proof is to find the right formulas $F_{n,w}$ and associated restrictions ρ_n to plug into the argument, and to make a case analysis of the action of ρ_n as above. Both of these aspects use the techniques developed in [5] in an essential way.

D. Outline of this paper

The rest of this paper is organized as follows. After having given the necessary preliminaries in Section II, we state the main theorem for resolution and give a full proof in Section III. In Section IV, we discuss how the theorem can be strengthened to polynomial calculus resolution and Sherali-Adams and why our approach does not work for Lasserre. We omit most of the details due to space constraints, however, and refer the reader to the upcoming full-length version for full proofs. We conclude in Section V with some final remarks and a discussion of open problems.

II. Preliminaries

A *literal* over a Boolean variable x is either the variable x itself (a *positive literal*) or its negation $\overline{x}$ (a *negative literal*). A *clause* $C = a_1 \vee \cdots \vee a_k$ is a disjunction of literals. A *k-clause* is a clause that contains at most k literals. A *CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of clauses. A *k-CNF formula* is a CNF formula consisting of k-clauses. We think of clauses and CNF formulas as sets: the order of elements is irrelevant and there are no repetitions. We denote the logical true value as $\top$ and the logical false value as $\bot$. The empty clause (containing no literals) is also denoted $\bot$, since it is always false. For integers m and n, $m < n$, we use the standard notation $[n] = \{1, 2, \ldots, n\}$ and $[m, n] = \{m, m+1, \ldots, n\}$.

A *resolution derivation* of a clause C from a CNF formula F is a sequence of clauses $(C_1, \ldots, C_\tau)$ such that $C_\tau = C$ and for $1 \leq t \leq \tau$ the clause C_t is obtained by one of the following derivation rules:

- **Axiom:** C_t is a clause in F (an *axiom clause*);
- **Inference:** $C_t = A \vee B$, where $C_i = A \vee x$ and $C_j = B \vee \overline{x}$ for $1 \leq i, j < t$;
- **Weakening:** $C_t \supseteq C_i$ for some $1 \leq i < t$.

A *resolution refutation* of F is a derivation of the empty clause $\bot$ from F.

Every resolution derivation $\pi = (C_1, \ldots, C_\tau)$ can be associated with a directed acyclic graph G_π with vertices labelled by clauses C_t in π and edges (C_i, C_j) if C_j is obtained by an inference or a weakening step and C_i is used as a premise in that step. The derivation π is said to be *tree-like* if G_π is a tree. The *(clause) space* of π at time t is the number of clauses derived before or at time t that will be used after or at time t, i.e., all clauses C_i, $i \leq t$, in G_π having an outgoing edge to clauses C_j, $j \geq t$ (plus the clause C_t itself). The space of π is the maximum space at any time t in the derivation. The *width* of π is the maximum number of literals in any clause C_t in π, and the *size* (or *length*) of $\pi = (C_1, \ldots, C_\tau)$ is τ.

In *polynomial calculus resolution (PCR)* one instead refutes an unsatisfiable formula F over variables $x_1, \ldots, x_n$ by reasoning in terms of polynomials in the ring $\mathbb{F}[x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n]$, where $\mathbb{F}$ is some fixed field and x_i, $\overline{x}_i$ are formally independent variables. It is natural to think of polynomials as being satisfied by an assignment when they evaluate to 0, so in PCR the truth values $\top$ and $\bot$ are represented by 0 and 1, respectively, and a clause $\bigvee_{i \in \mathcal{I}} x_i \vee \bigvee_{i \in \mathcal{J}} \overline{x}_i$ is translated into the one-term polynomial $\prod_{i \in \mathcal{I}} x_i \cdot \prod_{i \in \mathcal{J}} \overline{x}_i$. A *PCR derivation* of a polynomial R from a set of polynomials $\mathcal{S} = \{Q_1, \ldots, Q_m\}$ is a sequence $(P_1, \ldots, P_\tau)$ such that $P_\tau = R$ and for $1 \leq t \leq \tau$ the polynomial P_t is obtained by one of the following derivation rules:

- **Boolean axiom:** P_t is $x^2 - x$ for some variable x (or $\overline{x}$);
- **Complementarity axiom:** P_t is $1 - x - \overline{x}$ for some variable x;
- **Initial axiom:** P_t is one of the polynomials $Q_j \in \mathcal{S}$;
- **Linear combination:** $P_t = \alpha P_i + \beta P_j$ for $1 \leq i, j < t$ and some $\alpha, \beta \in \mathbb{F}$;
- **Multiplication:** $P_t = x P_i$ for $1 \leq i < t$ and some variable x.

A *PCR refutation* of F is a PCR derivation of 1 from the set of polynomials representing the clauses of F as explained above. Note that the Boolean axioms make sure that variables can only take values $\top = 0$ and $\bot = 1$, and the complementarity axioms enforce that x and $\overline{x}$ take opposite values.

The *degree* of a PCR derivation π is the maximum of the (total) degrees of the polynomials in π. The *size* of π is the total number of terms[6] in π counted with repetitions, where we always write the polynomials expanded out as sums of terms. The *space* measure can also be generalized from resolution, counting terms instead of clauses, but we will not really need it in this paper.

Let us next discuss *semialgebraic* proof systems. All such proof systems encode a CNF formula as a set of polynomial inequalities over the reals. A clause $\bigvee_{i\in\mathcal{I}} x_i \vee \bigvee_{i\in\mathcal{J}} \overline{x}_i$ is represented by the inequality $\sum_{i\in\mathcal{I}} x_i + \sum_{i\in\mathcal{J}}(1-x_i) - 1 \geq 0$, where we identify $\top = 1$ and $\bot = 0$—note that this is the opposite of the convention for PCR. A CNF formula F is represented by the inequalities corresponding to its clauses. A *Sherali-Adams (SA) derivation* of an inequality $R \geq 0$ from a set of polynomial inequalities $\{Q_1 \geq 0, \ldots, Q_m \geq 0\}$ is an equation of the form

$$\textstyle\sum_{t=1}^{\tau} \alpha_t \cdot \prod_{i\in\mathcal{I}_t} x_i \cdot \prod_{i\in\mathcal{J}_t}(1-x_i) \cdot P_t = R \ , \qquad (1)$$

where $\alpha_t \in \mathbb{R}^+$ and P_t is one of the Q_j, or an *axiom* of the form $x_i^2 - x_i$ or $x_i - x_i^2$, or the constant 1. A *Lasserre derivation* of $R \geq 0$ is an equation of the form (1) where in addition P_t can be a square Q^2 for any arbitrary polynomial Q. Note that Sherali-Adams and Lasserre are *static* proof systems in that they have "one-shot" derivations, in contrast to resolution and PCR that construct derivations dynamically step by step.

We can augment Sherali-Adams by twin variables $\overline{x}_i$ whose intended meaning is the negation of x_i, i.e., $1 - x_i$.[7] We define a *Sherali-Adams resolution (SAR) derivation* to be an SA-derivation as in (1) except that the set of variables is $\{x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n\}$ and that P_t can also be a complementarity axiom $1 - x_i - \overline{x}_i$ or $-1 + x_i + \overline{x}_i$.

A *Sherali-Adams (SA), SAR, or Lasserre refutation* of F is a derivation in the respective system of the inequality $-1 \geq 0$ from the inequalities $Q_1 \geq 0, \ldots, Q_m \geq 0$ that encode the clauses of F. The *rank* of the derivation is the maximum of the degrees among the polynomials $\prod_{i\in\mathcal{I}_t} x_i \cdot \prod_{i\in\mathcal{J}_t}(1-x_i) \cdot P_t$ in (1), and the *size* of the derivation is the sum of the sizes of the polynomials in the sum, measured as the number of terms when each polynomial is expanded out as a sum of terms and before cancellation with terms from other polynomials. (The reason that we have to be a bit careful with this definition is that we have complete cancellation in a refutation except for the constant term -1, but of course this should not mean that every refutation has constant size.)

A *restriction* (or *partial assignment*) ρ is a partial mapping from variables to $\{\bot, \top\}$. We identify ρ with the set of literals it sets to true. The *domain* of ρ is denoted $\mathrm{dom}(\rho)$ and the size of ρ is $|\rho| = |\mathrm{dom}(\rho)|$. The restriction $C\restriction_\rho$ of a clause C by ρ is the trivial clause $\top$ if ρ sets some literal of C to true—such a clause can just be removed from any formula or derivation—and otherwise it is the clause resulting from deleting all literals in C set to false by ρ. The restriction $F\restriction_\rho$ of a CNF formula F is the conjunction of its restricted clauses, and a restricted resolution derivation $\pi\restriction_\rho$ is the sequence of the restrictions of the clauses in π. It is a basic fact that if π is a refutation of F, then $\pi\restriction_\rho$ is a refutation of $F\restriction_\rho$.

For PCR derivations and the polynomials therein, restrictions are defined similarly: a restricted term vanishes if one of its variables is set to $\top = 0$ and is otherwise obtained by deleting all variables set to $\bot = 1$, and a restricted polynomial is the sum of its restricted terms. Again, restrictions preserve PCR refutations. For SA and SAR, the definition is analogous except the roles of 0 and 1 are reversed.

III. Upper and Lower Bounds in Resolution

In this section, we state the special case of our main result for the resolution proof system and give a full proof. The idea is to convey the main ideas of the argument while avoiding the additional technical details that are needed for the stronger proof systems. Let us start by presenting the slightly more detailed version of Theorem 1, but restricted to resolution, which is what we will prove.

Theorem 3. *Let $k = k(n)$ be any integer-valued function such that $k(n) \leq n/4\log n$. Then there is a family of 3-CNF formulas $\{F_{n,k}\}_{n\geq 1}$, where $F_{n,k}$ has $\mathrm{O}(n^2)$ variables and $\mathrm{O}(kn^2)$ clauses, such that:*

1) *$F_{n,k}$ has a tree-like resolution refutation in size $\mathrm{O}(k^k n^k)$, width $2k+1$, and space $2k+3$;*
2) *any resolution refutation of $F_{n,k}$ has size $\Omega(n^{k-1}/(3k^2\log n)^k)$.*

Straightforward calculations show that if $k(n) = \mathrm{O}(n^c)$ for $c < 1/2$, then the upper bound is $n^{\mathrm{O}(k)}$ and the lower bound is $n^{\Omega(k)}$.

A. Definition of the formula

The formula we use to establish Theorem 3 formalizes a *relativized* version of the pigeonhole principle claiming that there are (partial) functions $p\colon [k] \to [n]$ and $q\colon [n] \to [k-1]$ such that p is one-to-one and defined on $[k]$, and q is one-to-one and defined on the range of p. First we describe a straightforward CNF encoding with wide clauses that we denote $RPHP^{k,n}_{k-1}$. Once the general idea is clear, we

[6] Just to make terminology precise, in this paper a *monomial* is a product of variables and a *term* is a monomial multiplied by a non-zero coefficient from the field $\mathbb{F}$.

[7] In fact, this is how PCR was extended in [1] from the original definition of polynomial calculus (PC) in [26].

transform this into a slightly more involved 3-CNF formula which is the formula we will work with.

The formula $RPHP^{k,n}_{k-1}$ is over variables $p_{u,v}$ that encode the function p, $q_{v,w}$ that encode the function q, and r_v that encode a superset of the range of p. It consists of the following collection of clauses, where u, u' range over $[k]$, v, v' range over $[n]$, and w ranges over $[k-1]$:

$$p_{u,1} \lor p_{u,2} \lor \cdots \lor p_{u,n} \qquad \text{for all } u, \tag{2a}$$
$$\overline{p}_{u,v} \lor \overline{p}_{u',v} \qquad \text{for all } u \neq u' \text{ and } v, \tag{2b}$$
$$\overline{p}_{u,v} \lor r_v \qquad \text{for all } u \text{ and } v, \tag{2c}$$
$$\overline{r}_v \lor q_{v,1} \lor \cdots \lor q_{v,k-1} \qquad \text{for all } v, \tag{2d}$$
$$\overline{r}_v \lor \overline{r}_{v'} \lor \overline{q}_{v,w} \lor \overline{q}_{v',w} \qquad \text{for all } v \neq v' \text{ and } w. \tag{2e}$$

The clauses in (2a)–(2b) say that p maps $[k]$ injectively into $[n]$; clauses (2c) say that r contains the range of p; and clauses (2d)–(2e) force q to be defined and injective on this range.

Next, we transform $RPHP^{k,n}_{k-1}$ into an extended 3-CNF version, which we denote $ERPHP^{k,n}_{k-1}$. This is done in the standard way by using extension variables to break up the wide clauses in (2a) and (2d) and the 4-clauses in (2e). For (2a) we obtain the clauses

$$p_{u,1} \lor p_{u,2} \lor y_{u,2} \;, \tag{3a}$$
$$\overline{y}_{u,v} \lor p_{u,v+1} \lor y_{u,v+1} \qquad \text{for all } v \in [2, n-3], \tag{3b}$$
$$\overline{y}_{u,n-2} \lor p_{u,n-1} \lor p_{u,n} \;, \tag{3c}$$

for all $u \in [k]$, splitting up (2d) yields

$$\overline{r}_v \lor q_{v,1} \lor z_{v,1} \;, \tag{3d}$$
$$\overline{z}_{v,w} \lor q_{v,w+1} \lor z_{v,w+1} \qquad \text{for all } w \in [k-4], \tag{3e}$$
$$\overline{z}_{v,k-3} \lor q_{v,k-2} \lor q_{v,k-1} \;, \tag{3f}$$

for all $v \in [n]$, and the rest of the clauses in $ERPHP^{k,n}_{k-1}$ are

$$\overline{p}_{u,v} \lor \overline{p}_{u',v} \qquad \text{for all } u \neq u' \text{ and } v, \tag{3g}$$
$$\overline{p}_{u,v} \lor r_v \qquad \text{for all } u \text{ and } v, \tag{3h}$$
$$\overline{r}_v \lor \overline{r}_{v'} \lor r_{v,v'} \qquad \text{for all } v \neq v', \tag{3i}$$
$$\overline{r}_{v,v'} \lor \overline{q}_{v,w} \lor \overline{q}_{v',w} \qquad \text{for all } v \neq v' \text{ and all } w, \tag{3j}$$

where as before u, u' range over $[k]$, v, v' range over $[n]$, and w ranges over $[k-1]$.

B. Proof of the upper bound

Let us now describe how we can refute the 3-CNF formula $ERPHP^{k,n}_{k-1}$ consisting of the clauses in (3a)–(3j) in resolution. In order to do so, we consider all sequences of the form $(v_1, v_2, \ldots, v_k, w_1, w_2, \ldots, w_k)$, where $v_u \in [n]$ and $w_u \in [k-1]$, and the corresponding clauses

$$\bigvee_{u\in[k]} \overline{p}_{u,v_u} \lor \bigvee_{u\in[k]} \overline{q}_{v_u,w_u}. \tag{4}$$

We derive all such clauses from the axiom clauses of $ERPHP^{k,n}_{k-1}$, and from these clauses it is then straightforward to derive contradiction. All of these (sub)derivations are efficient, so the size of the whole refutation is dominated by the number of clauses in (4).

For each clause in (4) we are in one of two cases: either $v_u = v_{u'}$ holds for some $u \neq u'$, or there must exist a pair $v_u \neq v_{u'}$ with $w_u = w_{u'}$ by the pigeonhole principle. In the former case, the clause (4) is just a weakening of the axiom (3g), namely $\overline{p}_{u,v} \lor \overline{p}_{u',v}$ with $v = v_u = v_{u'}$. In the latter case, we combine axioms $\overline{p}_{u,v_u} \lor r_{v_u}$ and $\overline{p}_{u',v_{u'}} \lor r_{v_{u'}}$ from (3h), $\overline{r}_{v_u} \lor \overline{r}_{v_{u'}} \lor r_{v_u,v_{u'}}$ from (3i), and $\overline{r}_{v_u,v_{u'}} \lor \overline{q}_{v_u,w} \lor \overline{q}_{v_{u'},w}$ from (3j), where $w = w_u = w_{u'}$, to obtain the clause $\overline{p}_{u,v_u} \lor \overline{p}_{u',v_{u'}} \lor \overline{q}_{v_u,w} \lor \overline{q}_{v_{u'},w}$, from which (4) can be derived by weakening. Since a constant number of clauses is involved in this derivation it requires only constant space, and it is straightforward to verify that it can in fact be carried out by a tree-like derivation in space 3 (i.e., keeping one clause in memory and resolving it with a sequence of axioms).

The rest of the refutation consists of derivations of all prefixes of clauses of the form (4) by backward induction. Assuming we are able to derive any prefix of length t in clause space $(2k-t)+3$ we show how to derive all prefixes of length $t-1$ in clause space $(2k-t+1)+3$. The refutation ends when we get the prefix of length 0 (i.e., the empty clause) in clause space $2k+3$.

To this end, suppose we can derive each clause of the form $A \lor \overline{q}_{v_u,w}$, $w \in [k-1]$, in clause space s. We want to derive A from all such clauses in space $s+1$. Notice that the literal $\overline{p}_{u,v_u}$ appears in A. We resolve the axiom $\overline{p}_{u,v_u} \lor r_{v_u}$, first with the axiom $\overline{r}_{v_u} \lor q_{v_u,1} \lor z_{v_u,1}$ and then with the clause $A \lor \overline{q}_{v_u,1}$ to get $A \lor z_{v_u,1}$ in space s. Once we have clauses $A \lor z_{v_u,w-1}$ in memory for some $w > 1$, we resolve them with the axiom $\overline{z}_{v_u,w-1} \lor q_{v_u,w} \lor z_{v_u,w}$ to obtain $A \lor z_{v_u,w} \lor q_{v_u,w}$. Keeping the latter clause in memory we derive $A \lor \overline{q}_{v_u,w}$ within space $s+1$ in total, and then we resolve to get $A \lor z_{v_u,w}$. We finally derive clause A by resolving the axiom $\overline{z}_{v_u,k-3} \lor q_{v_u,k-2} \lor q_{v_u,k-1}$ with $A \lor z_{v_u,k-3}$, $A \lor \overline{q}_{v_u,k-2}$ and $A \lor \overline{q}_{v_u,k-1}$. Again, the clause space of this derivation is $s+1$.

After k steps of this backward induction we get to clauses of the form $\overline{p}_{1,v_1} \lor \overline{p}_{2,v_2} \lor \ldots \lor \overline{p}_{k,v_k}$. To derive the empty clause we do k more steps of backward induction, repeating the procedure in the previous paragraph. At step u, suppose that we have all clauses of the form $A \lor \overline{p}_{u,1}$, $A \lor \overline{p}_{u,2}$, ..., $A \lor \overline{p}_{u,n}$ and want to derive A. To do so, we first resolve the axiom $p_{u,1} \lor p_{u,2} \lor y_{u,2}$ with $A \lor \overline{p}_{u,1}$ and then with $A \lor \overline{p}_{u,2}$ to get $A \lor y_{u,2}$. In order to obtain $A \lor y_{u,i+1}$ we resolve $\overline{y}_{u,i} \lor p_{u,i+1} \lor y_{u,i+1}$ with $A \lor y_{u,i}$ and then with $A \lor \overline{p}_{u,2}$. We iterate up to $A \lor y_{u,n-2}$ and finally resolve the axiom $\overline{y}_{u,n-2} \lor p_{u,n-1} \lor p_{u,n}$ with the clauses $A \lor y_{u,n-2}$, $A \lor \overline{p}_{u,n-1}$ and $A \lor \overline{p}_{u,n}$. After k steps of this backward induction we reach the empty clause and the refutation is complete.

To measure the length of the refutation we consider the prefix tree of our sequences $(v_1, v_2, \ldots, v_k, w_1, w_2, \ldots, w_k)$. Each vertex of this tree corresponds to one of the clauses A we derived during the backward induction, with the empty clause at the root and clauses (4) at the leaves. The length of the derivation of each clause is linear in the number of children, and in addition we derived the leaves with a constant number of steps. Therefore we can charge a constant amount of steps per vertex. The size of the tree is $O(k^k n^k)$, and it follows that this is also the size of the refutation. This refutation is tree-like since no intermediate clause is used more than once. The width of the refutation is $2k+1$ and reaches this maximum at the induction step from sequences of length $2k$ to sequences of length $2k-1$.

C. Proof of the lower bound for resolution

As discussed in Section I-C, we use a random restriction argument to prove our size lower bound for resolution refutations of the formula $\mathit{ERPHP}^{k,n}_{k-1}$. We define a distribution $\mathcal{D}$ on partial assignments ρ by picking a subset $\mathcal{S} = \{v_1, v_2, \ldots, v_k\}$ of k elements from $[n]$ uniformly at random and letting ρ assign values to variables as follows:

- $r_v = \top$ for $v \in \mathcal{S}$; $r_v = \bot$ otherwise;
- $r_{v,v'} = r_v \wedge r_{v'}$ for all $v \neq v'$;
- $p_{u,v_u} = \top$ for $u \in [k]$; $p_{u,v} = \bot$ for all other $p_{u,v}$;
- $y_{u,v}$ are set arbitrarily so as to satify (3a)–(3c);
- $q_{v,w}$ and $z_{v,w}$ are left unset for $v \in \mathcal{S}$;
- $q_{v,w} = b_v$ and $z_{v,w} = b_v$ for all $v \in [n] \setminus \mathcal{S}$ and all $w \in [k-1]$, where $b_v \in \{\bot, \top\}$ is chosen uniformly at random.

We want to argue that with high probability such restrictions remove or at least significantly shrink wide clauses.

Formally, let us say that a clause (or term) *mentions* a pigeon $v \in [n]$ if it contains some variable in the set $\{q_{v,1}, \ldots, q_{v,k-1}, z_{v,1}, \ldots, z_{v,k-1}\}$ and define the *pigeon-width* to be the number of pigeons mentioned. The next lemma describes the effect of random restrictions ρ from $\mathcal{D}$ on clauses (or terms) depending on their pigeon-width. Namely, a sufficiently wide clause, i.e., mentioning a lot of pigeons, is satisfied by the random restriction with high probability, whereas a narrower clause may not be set by the restriction but will with high probability contain few pigeons afterwards.

Lemma 4. *Let k, ℓ, n be natural numbers such that $n \geq 16$ and $\ell \leq k \leq n/4\log n$. Let A be either a clause or term over the variables of $\mathit{ERPHP}^{k,n}_{k-1}$ and let ρ be a random restriction sampled from $\mathcal{D}$. Then the pigeon-width of $A\restriction_\rho$ is less than ℓ with probability at least $1 - (3k^2\log n)^k/n^\ell$.*

Proof: Let us assume that A is a clause—the proof for terms (which will be used for PCR and Sherali-Adams) is completely analogous. Let $v_1, \ldots, v_r$ be the pigeons mentioned in A, sorted in some order, and let $a_1, \ldots, a_r$ be a sequence of literals such that a_i witnesses that A mentions v_i.

If $r > 2k \log n$, then the probability that the clause is not satisfied by the restriction is at most

$$\prod_{i=1}^{r} \Pr\bigl[\rho(a_i) \neq \top \big| \rho(a_j) \neq \top \text{ for } j < i\bigr] \leq$$
$$\leq \prod_{i=1}^{r} \Pr\bigl[\rho(a_i) \neq \top \big| v_j \notin \mathcal{S} \text{ for } j < i\bigr] \leq$$
$$\leq \prod_{i=1}^{r} \left(\frac{1}{2} + \frac{k}{n-i}\right) < \left(\frac{5}{8}\right)^{2k\log n} < \frac{1}{n^k} \ . \quad (5)$$

To see this, note that the event $\rho(a_i) \neq \top$ occurs either if the pigeon v_i is not picked or if the literal a_i is set to the wrong value. Assuming that no pigeon $v_1, \ldots, v_{i-1}$ has been picked before v_i, the conditional probability of v_i being included in $\mathcal{S}$ is $k/(n-i)$, and is less otherwise. If $v_i \in \mathcal{S}$, then a_i gets the wrong value with probability $1/2$. The final inequalities hold because the ratio $k/(n - 2k\log n)$ is at most $1/(2\log n)$, and therefore it is at most $1/8$ for $n \geq 16$.

If instead $r \leq 2k \log n$, we want to bound the probability that there are at least ℓ pigeons mentioned in A that are chosen in $\mathcal{S}$ and hence survive. For a subset $\mathcal{S}'$ of $i \geq \ell$ pigeons there are $\binom{r}{i}\binom{n-r}{k-i}$ ways of choosing $\mathcal{S}$ so that $\mathcal{S}' \subseteq \mathcal{S}$, and hence this happens with probability $\binom{r}{i}\binom{n-r}{k-i}/\binom{n}{k}$. Considering all possible intersections of size at least ℓ between the set of k selected pigeons and the r pigeons mentioned in A, we obtain that the probability of ℓ surviving pigeons is

$$\sum_{i=\ell}^{k} \binom{r}{i}\binom{n-r}{k-i} \bigg/ \binom{n}{k} \leq$$
$$\leq k \binom{\lfloor 2k\log n \rfloor}{k} \binom{n}{k-\ell} \bigg/ \binom{n}{k} \leq \frac{(3k^2 \log n)^k}{n^\ell} \ . \quad (6)$$

This concludes the proof. ∎

We can use Lemma 4 to show that if we hit a sufficiently short resolution refutation of $\mathit{ERPHP}^{k,n}_{k-1}$ with a restriction ρ, then in the restricted refutation all clauses will have small pigeon-width. The reason this is useful is that $\mathcal{D}$ is constructed so that the restricted formula is just the standard pigeonhole principle formula, or rather, a 3-CNF version of it (up to renaming of variables). To spell this out explicitly, after renaming the k pigeons in $[n]$ chosen by ρ to $1, \ldots, k$, what remains is the following collection of clauses (with v, v' ranging over $[k]$ and w ranging over $[k-1]$ unless stated otherwise):

$q_{v,1} \vee z_{v,1}$	for all v,	(7a)
$\overline{z}_{v,w} \vee q_{v,w+1} \vee z_{v,w+1}$	for all v and $w \in [k-4]$,	(7b)
$\overline{z}_{v,k-3} \vee q_{v,k-2} \vee q_{v,k-1}$	for all v,	(7c)
$\overline{q}_{v,w} \vee \overline{q}_{v',w}$	for all $v \neq v'$ and w.	(7d)

But the clauses (7a)–(7d), which we will denote $EPHP^k_{k-1}$, can easily be shown to require almost maximal pigeon-width in resolution.

Lemma 5. *Every resolution refutation of $EPHP^k_{k-1}$ has pigeon-width at least $k-1$.*

Proof: We use a game argument in the style of [3], [52] adapted to the notion of pigeon-width. The game is played between a *prosecutor* and a *defendant*. At each step of the game the prosecutor queries the defendant for the value of a variable of $EPHP^k_{k-1}$ and stores the answer in his record. The prosecutor is also allowed to erase variable assignments from his record after any query, but if so the defendant can answer differently next time she is asked about an erased variable. The goal of the prosecutor is to force the defendant to falsify a clause from $EPHP^k_{k-1}$, while the goal of the defendant is to answer queries without falsifying any axiom clause in the formula.

To establish the lemma, it is sufficient to show that the prosecutor cannot win unless at some point he holds a record that mentions k pigeons. The reason for this is that if there exists a resolution refutation π of pigeon-width $\ell < k-1$, then the prosecutor can use such a refutation to construct a strategy that never mentions more than $\ell + 1$ pigeons.

To build a winning strategy from a refutation π, the prosecutor walks backwards through the associated graph G_π from the final empty clause all the way to some axiom clause. The invariant maintained is that at each step the current assignment on record is the minimal falsifying assignment for the clause currently visited in G_π. At the beginning of the game the empty record corresponds to the empty clause in the refutation. If the current clause was obtained by resolution, the prosecutor queries the resolved variable (which might temporarily increase the number of pigeons on record by 1), moves to the premise falsified by the answer, and then forgets all assignments not needed to falsify that clause. For a weakening step, the prosecutor just needs to forget variables. The prosecutor wins when the game reaches a source vertex in G_π (if not earlier), since by the invariant the corresponding axiom clause is falsified by the assignment on record at that point.

Switching to the lower-bound perspective, let us now briefly describe a defendant strategy that works against prosecutors mentioning less than k pigeons. The defendant privately keeps a partial matching of the pigeons mentioned in the current record of the prosecutor into holes, making sure that this mapping is compatible with the partial assignment in his record. If the prosecutor asks about a variable which mentions a pigeon already in the domain of the defendant's partial matching, she answers consistently with her matching. If the prosecutor erases all variables mentioning a pigeon, the defendant removes that pigeon from the partial mapping, freeing up the corresponding hole for later reuse. If the prosecutor queries a variable that mentions a new pigeon, we are in one of two cases: either there is at least one free hole, or the record mentions $k-1$ pigeons. In the first case the defendant assigns the new pigeon to some free hole and updates her partial matching accordingly. In the second case the defendant has achieved her goal—although she is now forced to falsify a clause of $EPHP^k_{k-1}$ and loses, the prosecutor was able to win only by compiling a record that mentions k pigeons. ∎

Putting all the pieces together we can now prove the lower bound in Theorem 3. Namely, let π be a resolution refutation of $ERPHP^{k,n}_{k-1}$ of size S. Hit π with a random restriction ρ distributed according to $\mathcal{D}$. Since resolution refutations are preserved under restrictions, $\pi\restriction_\rho$ is a refutation of $ERPHP^{k,n}_{k-1}\restriction_\rho$ which, as discussed above, is $EPHP^k_{k-1}$ after renaming of variables. By Lemma 5, this refutation must have pigeon-width at least $k-1$ with probability 1. On the other hand, using Lemma 4 with $\ell = k-1$ and taking a union bound over all clauses in π, the probability that this happens is at most $S \cdot (3k^2 \log n)^k / n^{k-1}$ for large enough n. We can hence conclude that $S \geq n^{k-1}/(3k^2 \log n)^k$, and the proof of Theorem 3 is complete.

IV. Algebraic and semialgebraic proof systems

In this section, we show that the size lower bound for resolution in Section III extends to polynomial calculus resolution (PCR) and Sherali-Adams resolution (SAR) but not to Lasserre. Due to space constraints, we omit full proofs of the results for SAR and Lasserre in this extended abstract.

The main idea for the size lower bounds is to first prove a lower bound on a parameter analogous to the pigeon-width in Section III, which we call *pigeon-degree* for PCR and *pigeon-rank* for SAR, and then to plug it into the random restriction argument as in the proof of Lemma 4.

A. Refutation size lower bound for PCR

Generalizing the terminology in Section III, we say that not only the variables $q_{v,w}$ and $z_{v,w}$ of $EPHP^k_{k-1}$ but also their twins $\overline{q}_{v,w}$ and $\overline{z}_{v,w}$ are said to *mention* pigeon v. A set of such variables is said to mention a pigeon if some variable in it does. The *pigeon-degree* of a monomial is the number of pigeons that are mentioned by its variables, and the pigeon-degree of a PCR refutation of $EPHP^k_{k-1}$ is the maximum pigeon-degree of any monomial in the refutation. We now have the following analogue of Lemma 5.

Lemma 6. *Every PCR refutation of $EPHP^k_{k-1}$ has pigeon-degree at least $\lceil \frac{k-1}{2} \rceil$.*

Proof: We assume we have a PCR refutation of $EPHP^k_{k-1}$ in which all monomials mention at most d pigeons and transform it into a refutation of degree $d+1$ of an alternative formulation $APHP^k_{k-1}$ of the pigeonhole principle from k pigeons to $k-1$ holes described in [53]. This formulation does not have refutations of degree $\lceil \frac{k-1}{2} \rceil$

or less [43, Theorem 3.9], from which it follows that $d \geq \lceil \frac{k-1}{2} \rceil$.

The alternative formulation $APHP^{k}_{k-1}$ is defined on variables $x_{v,w}$ for $v \in [k]$ and $w \in [k-1]$, where $x_{v,w} = 1$ means that pigeon v sits in hole w; we stress that this interpretation of the variables is the opposite of the one we use for $EPHP^{k}_{k-1}$. Also, $APHP^{k}_{k-1}$ is not a (translation of a) CNF formula but consists of the following polynomials:

$$1 - \sum_{w \in [k-1]} x_{v,w} \qquad \text{for all } v, \tag{8a}$$

$$x_{v,w} x_{v',w} \qquad \text{for all } w \text{ and all } v \neq v', \tag{8b}$$

$$x_{v,w} x_{v,w'} \qquad \text{for all } v \text{ and all } w \neq w'. \tag{8c}$$

To obtain a degree-$(d+1)$ refutation for $APHP^{k}_{k-1}$, the first step is to apply a substitution δ to the variables in the refutation of $EPHP^{k}_{k-1}$ in pigeon-degree d. For q-variables we define $\delta(q_{v,w}) = 1 - x_{v,w}$ and $\delta(\overline{q}_{v,w}) = x_{v,w}$, and for z-variables we let $\delta(z_{v,w}) = 1 - \sum_{j > w} x_{v,j}$ and $\delta(\overline{z}_{v,w}) = 1 - \sum_{j \leq w} x_{v,j}$. This substitution transforms the refutation of the initial formula $EPHP^{k}_{k-1}$ into a sequence of polynomials over the variables in $APHP^{k}_{k-1}$.

This is not yet a valid refutation, however, and in order to deal with this we need to show how to derive each substituted polynomial in the sequence. How to do so depends on how the polynomial was derived before the substitution. For the inference steps, if we derived xp from p then $\delta(xp) = \delta(x)\delta(p)$ can be derived from $\delta(p)$ with a sequence of multiplications and linear combinations, and if the polynomial was derived via a linear combination, then the same derivation step is valid for the substituted polynomials. For the logical axioms of PCR, the polynomials that result from applying δ can be derived from $APHP^{k}_{k-1}$ in constant degree. The remaining cases are the polynomials obtained applying δ to the clauses in (7a)–(7c). Consider $\delta(\overline{z}_{v,w} q_{v,w+1} z_{v,w+1})$ for $1 \leq w < k-3$; the other cases are very similar. We have

$$\delta(\overline{z}_{v,w} q_{v,w+1} z_{v,w+1}) = \Big(1 - \sum_{j \leq w} x_{v,j}\Big)\Big(1 - x_{v,w+1}\Big)\Big(1 - \sum_{j > w+1} x_{v,j}\Big) \tag{9}$$

which expands into $1 - \sum_{j \in [k-1]} x_{v,j} + r$ where r is a degree-3 polynomial on variables $x_{v,w}$, and is in the ideal generated by $x_{v,w} x_{v,w'}$ and $x_{v,w}^2 - x_{v,w}$. Thus $\delta(\overline{z}_{v,w} q_{v,w+1} z_{v,w+1})$ can be derived from $APHP^{k}_{k-1}$.

Now we have a refutation of $APHP^{k}_{k-1}$. Our substitution exchanges variables indexed by pigeon v with degree-1 polynomials which mention just v, and therefore each monomial of this refutation mentions at most d pigeons as well. We postprocess this refutation by removing all the monomials that mention the same pigeon twice or more and all the monomials that mention more than one pigeon for the same hole. This is possible using the axioms $x_{v,w} x_{v,w'}$ and $x_{v,w} x_{v',w}$, and it gives a new refutation of degree at most $d+1$. The lemma follows. ∎

The lower bound on pigeon-degree in Lemma 6 together with Lemma 4 imply the size lower bound for PCR.

Theorem 7. *Let $k = k(n)$ be any integer-valued function such that $k(n) \leq n/4 \log n$. Any PCR refutation of $ERPHP^{k,n}_{k-1}$ has size $\Omega\big(n^{\lceil (k-1)/2 \rceil}/(3k^2 \log n)^k\big)$.*

Proof: Let $\mathcal{M}$ be the set of monomials appearing in a PCR refutation of $ERPHP^{k,n}_{k-1}$. We hit the refutation with a random restriction ρ distributed according to $\mathcal{D}$. Since restrictions preserve PCR derivations we obtain a refutation of $ERPHP^{k,n}_{k-1}\restriction_{\rho}$, which as before is $EPHP^{k}_{k-1}$.

Assume that $|\mathcal{M}| < n^{\lceil (k-1)/2 \rceil}/(3k^2 \log n)^k$. Applying Lemma 4 with $\ell = \left\lceil \frac{k-1}{2} \right\rceil$ and taking a union bound over the monomials in $\mathcal{M}$, we conclude that there must be at least one restriction ρ in the support of $\mathcal{D}$ such that the pigeon-degree of $\pi\restriction_{\rho}$ is at most $\left\lceil \frac{k-1}{2} \right\rceil - 1$ if n is large enough. This contradicts Lemma 6, and hence $|\mathcal{M}|$ must be at least $n^{\lceil (k-1)/2 \rceil}/(3k^2 \log n)^k$. ∎

B. Refutation length lower bound for SAR

The definition of pigeon-rank is the analogue of pigeon-width for resolution (see Section III-C) and pigeon-degree for PCR (see Section IV-A). The *pigeon-rank* of an SAR refutation of $EPHP^{k}_{k-1}$ is the maximum pigeon-degree among the polynomials $\prod_{i \in \mathcal{I}_t} x_i \cdot \prod_{i \in \mathcal{J}_t} (1 - x_i) \cdot P_t$ in Equation (1).

In [29], a rank lower bound was proven on SAR refutations of PHP^{k}_{k-1}. We extend this to a pigeon-rank lower bound for $EPHP^{k}_{k-1}$. The proof is omitted in this extended abstract: the intuition is that a set of inequalities does not admit a refutation of pigeon-rank r if there is a family of distributions on partial assignments as follows.

1) There is a distribution for every set of variables that mentions at most r pigeons;
2) the distributions of two sets of variables are consistent on their intersection;
3) no initial inequality or logical axiom is falsified by any assignment in the support of any distribution.

Thus the proof of the following lemma (omitted here) amounts to showing that such a family of distributions exists for $r = k-1$.

Lemma 8. *Every SAR refutation of $EPHP^{k}_{k-1}$ has pigeon-rank at least k.*

The lower bound on pigeon-rank in Lemma 8 together with Lemma 4 imply the size lower bound for SAR.

Theorem 9. *Let $k = k(n)$ be any integer-valued function such that $k(n) \leq n/4 \log n$. Any SAR refutation of $ERPHP^{k,n}_{k-1}$ has size $\Omega\big(n^{k}/(3k^2 \log n)^k\big)$.*

Proof: Let $\mathcal{M}$ be the set of monomials appearing in a SAR refutation of $ERPHP^{k,n}_{k-1}$. We hit the refutation with a random restriction ρ distributed according to $\mathcal{D}$. Since restrictions preserve soundness of SAR proofs we obtain a refutation of $ERPHP^{k,n}_{k-1}\restriction_{\rho}$, which is $EPHP^{k}_{k-1}$.

Suppose now that $|\mathcal{M}| < n^k/(3k^2 \log n)^k$. Using Lemma 4 with $\ell = k$ and a union bound argument for $\mathcal{M}$, we conclude that there exists at least one restriction ρ in the support of $\mathcal{D}$ such that the pigeon-rank of $\pi\restriction_{\rho}$ is at most $k-1$, assuming that n large enough. But this contradicts Lemma 8, and hence the theorem follows. ■

C. Upper bound for Lasserre proofs

Our lower bound does not extend all the way up to Lasserre refutations. In the full version of this paper we show that $RPHP^{k,n}_{k-1}$ and $ERPHP^{k,n}_{k-1}$ have Lasserre refutations of size polynomial in k and n. More specifically, we show that $RPHP^{k,n}_{k-1}$ and $ERPHP^{k,n}_{k-1}$ have constant-rank Lasserre refutations, for a constant that is independent of k and n, and this immediately implies the upper bound on refutation size. Let us state the concrete lemma for reference.

Lemma 10. *$RPHP^{k,n}_{k-1}$ and $ERPHP^{k,n}_{k-1}$ have Lasserre refutations of rank 9.*

The proof builds on Lemma 1.5 in [46] which can be used to construct a rank-2 Lasserre refutation of the (standard) pigeonhole principle formula PHP^{k}_{k-1}. A similar proof for PHP^{k}_{k-1} also appears in [37].

V. Concluding remarks

In this paper, we exhibit a family of 3-CNF formulas over n variables that can be refuted in resolution in width w but require refutations of size $n^{\Omega(w)}$. Furthermore, this lower bound can be extended to polynomial calculus resolution (PCR) and Sherali-Adams. This shows that the seemingly naive counting upper bounds on proof size in terms of width (for resolution), degree (for PCR) and rank (for Sherali-Adams) are actually all tight up to small constant factors in the exponent. Furthermore, it also implies that the result in [4] that CNF formulas refutable in width w can be decided by CDCL solvers in time $n^{\mathrm{O}(w)}$ is tight (again up to constant factors in the exponent), since any resolution refutation the solver finds might have to be that large in the worst case.

Let us conclude by briefly discussing some open problems related to this line of work.

Perhaps the most obvious question concerns the tightness of our result. Our formulas have roughly $N = n^2$ variables and are refutable in width roughly $w = 2k$, and our size lower bound is on the order of $n^k = N^{w/4}$. However, the direct counting argument for width w gives an upper bound of about N^w clauses. Could this gap in the exponent be closed? If so, this would have to be for a different formula family since ours has an upper bound of the type $n^k = N^{w/4}$. One point worth noting is that one can shave a factor 2 off the gap in the exponent by considering the 4-CNF formulas obtained if the 4-clauses in (2e) are *not* converted to 3-CNF. In this case, the same upper and lower bounds still hold, but the number of variables is on the order of $N = n$.

A more fundamental question is whether we can find a formula family that exhibits the same kind of hardness for Lasserre. We already know that the formulas studied in this paper will not work. For tree-like Lovász-Schrijver (LS), however, we believe that our formulas should be hard (and that the method of proof should be similar, with long *paths* in the refutation tree playing the role of long monomials). In view of the Lasserre upper bound, for tree-like LS^+ we do not know what to believe. The main problem with our formulas is that after restriction we obtain a pigeonhole principle which is hard for resolution, PCR and Sherali-Adams (in term of rank) but easy for LS^+. A way to get a similar lower bound for Lasserre might be to find a formula that is hard for Lasserre rank and that becomes hard for Lasserre size after relativization.

A natural formula for which it would be interesting to prove similar size lower bounds as in this paper is the so-called *clique formula* claiming that there is a k-clique in some fixed n-vertex graph chosen so that this claim is false. It has been conjectured (e.g., in [18]) that such formulas require resolution refutation size n^k for the right kind of graphs, and this has been proven for the restricted case of tree-like resolution [17]. If such a lower bound could be established, it would have interesting consequences for parameterized proof complexity.

Finally, while the relations between size, width, and space in resolution are now fairly well-understood, one big open question remains. Namely, it was shown in [15] that if a formula has a short resolution refutation then it can also be refuted in small width, but this narrow refutation is obtained at the price of an exponential blow-up in size. Is this inherent, or is it just an artifact of the proof? That is, can size and width be optimized simultaneously in resolution, or are there formulas for which optimizing one of the measures must always cause a stiff penalty for the other? For size vs. space and space vs. width dramatic trade-offs are known [8], [11], [14], and these results extend also to PCR [10], but it remains wide open whether there are similiar trade-offs between size and width in resolution or between size and degree in PCR.

Acknowledgments

The authors would like to thank Mladen Mikša and Marc Vinyals for interesting discussions related to the topics of this work.

Part of the work of the first author was done while visiting KTH Royal Institute of Technology. The second and third authors were funded by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no. 279611. The third author was also supported by Swedish Research Council grants 621-2010-4797 and 621-2012-5645.

References

[1] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov, and A. Wigderson, "Space complexity in propositional calculus," *SIAM Journal on Computing*, vol. 31, no. 4, pp. 1184–1211, 2002, preliminary version appeared in *STOC '00*.

[2] M. Alekhnovich and A. A. Razborov, "Lower bounds for polynomial calculus: Non-binomial case," *Proc. Steklov Institute of Mathematics*, vol. 242, pp. 18–35, 2003, available at http://people.cs.uchicago.edu/~razborov/files/misha.pdf. Preliminary version appeared in *FOCS '01*.

[3] A. Atserias and V. Dalmau, "A combinatorial characterization of resolution width," *Journal of Computer and System Sciences*, vol. 74, no. 3, pp. 323–334, May 2008, preliminary version appeared in *CCC '03*.

[4] A. Atserias, J. K. Fichte, and M. Thurley, "Clause-learning algorithms with many restarts and bounded-width resolution," *Journal of Artificial Intelligence Research*, vol. 40, pp. 353–373, Jan. 2011, preliminary version appeared in *SAT '09*.

[5] A. Atserias, M. Müller, and S. Oliva, "Lower bounds for DNF-refutations of a relativized weak pigeonhole principle," in *Proc. 28th Annual IEEE Conference on Computational Complexity (CCC '13)*, Jun. 2013, pp. 109–120.

[6] B. Barak, F. G. S. L. Brandão, A. W. Harrow, J. A. Kelner, D. Steurer, and Y. Zhou, "Hypercontractivity, sum-of-squares proofs, and their applications," in *Proc. 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, May 2012, pp. 307–326.

[7] R. J. Bayardo Jr. and R. Schrag, "Using CSP look-back techniques to solve real-world SAT instances," in *Proc. 14th National Conference on Artificial Intelligence (AAAI '97)*, Jul. 1997, pp. 203–208.

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

[9] P. Beame, T. Pitassi, and N. Segerlind, "Lower bounds for Lovász–Schrijver systems and beyond follow from multiparty communication complexity," *SIAM Journal on Computing*, vol. 37, no. 3, pp. 845–869, 2007, preliminary version appeared in *ICALP '05*.

[10] C. Beck, J. Nordström, and B. Tang, "Some trade-off results for polynomial calculus," in *Proc. 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, May 2013, pp. 813–822.

[11] E. Ben-Sasson, "Size space tradeoffs for resolution," *SIAM Journal on Computing*, vol. 38, no. 6, pp. 2511–2525, May 2009, preliminary version appeared in *STOC '02*.

[12] E. Ben-Sasson and N. Galesi, "Space complexity of random formulae in resolution," *Random Structures and Algorithms*, vol. 23, no. 1, pp. 92–109, Aug. 2003, preliminary version appeared in *CCC '01*.

[13] E. Ben-Sasson and J. Nordström, "Short proofs may be spacious: An optimal separation of space and length in resolution," in *Proc. 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, Oct. 2008, pp. 709–718.

[14] ——, "Understanding space in proof complexity: Separations and trade-offs via substitutions," in *Proc. 2nd Symposium on Innovations in Computer Science (ICS '11)*, Jan. 2011, pp. 401–416.

[15] E. Ben-Sasson and A. Wigderson, "Short proofs are narrow—resolution made simple," *Journal of the ACM*, vol. 48, no. 2, pp. 149–169, Mar. 2001, preliminary version appeared in *STOC '99*.

[16] C. Berkholz, "On the complexity of finding narrow proofs," in *Proc. 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '12)*, Oct. 2012, pp. 351–360.

[17] O. Beyersdorff, N. Galesi, and M. Lauria, "Parameterized complexity of DPLL search procedures," *ACM Transactions on Computational Logic*, vol. 14, no. 3, p. 20, Aug. 2013, preliminary version appeared in *SAT '11*.

[18] O. Beyersdorff, N. Galesi, M. Lauria, and A. A. Razborov, "Parameterized bounded-depth frege is not optimal," *ACM Transactions on Computation Theory*, vol. 4, pp. 7:1–7:16, Sep. 2012, preliminary version appeared in *ICALP '11*.

[19] A. Blake, "Canonical expressions in Boolean algebra," Ph.D. dissertation, University of Chicago, 1937.

[20] I. Bonacina and N. Galesi, "Pseudo-partitions, transversality and locality: A combinatorial characterization for the space measure in algebraic proof systems," in *Proc. 4th Innovations in Theoretical Computer Science Conference (ITCS '13)*, Jan. 2013.

[21] M. Brickenstein and A. Dreyer, "PolyBoRi: A framework for Gröbner-basis computations with Boolean polynomials," *Journal of Symbolic Computation*, vol. 44, no. 9, pp. 1326–1345, Sep. 2009.

[22] M. Brickenstein, A. Dreyer, G.-M. Greuel, M. Wedler, and O. Wienand, "New developments in the theory of Gröbner bases and applications to formal verification," *Journal of Pure and Applied Algebra*, vol. 213, no. 8, pp. 1612–1635, Aug. 2009.

[23] E. Chlamtáč and M. Tulsiani, "Convex relaxations and integrality gaps," in *Handbook on Semidefinite, Conic and Polynomial Optimization*, M. F. Anjos and J. B. Lasserre, Eds. Springer, 2012, pp. 139–169.

[24] V. Chvátal, "Edmond polytopes and a hierarchy of combinatorial problems," *Discrete Mathematics*, vol. 4, no. 1, pp. 305–337, 1973.

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

[26] M. Clegg, J. Edmonds, and R. Impagliazzo, "Using the Groebner basis algorithm to find proofs of unsatisfiability," in *Proc. 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, May 1996, pp. 174–183.

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

[28] W. Cook, C. R. Coullard, and G. Turán, "On the complexity of cutting-plane proofs," *Discrete Applied Mathematics*, vol. 18, no. 1, pp. 25–38, Nov. 1987.

[29] S. S. Dantchev, B. Martin, and M. Rhodes, "Tight rank lower bounds for the Sherali-Adams proof system," *Theoretical Computer Science*, vol. 410, no. 21-23, pp. 2054–2063, 2009.

[30] J. L. Esteban and J. Torán, "Space bounds for resolution," *Information and Computation*, vol. 171, no. 1, pp. 84–97, 2001, preliminary versions of these results appeared in *STACS '99* and *CSL '99*.

[31] Y. Filmus, M. Lauria, M. Mikša, J. Nordström, and M. Vinyals, "Towards an understanding of polynomial calculus: New separations and lower bounds (extended abstract)," in *Proc. 40th International Colloquium on Automata, Languages and Programming (ICALP '13)*, ser. Lecture Notes in Computer Science, vol. 7965. Springer, Jul. 2013, pp. 437–448.

[32] Y. Filmus, M. Lauria, J. Nordström, N. Thapen, and N. Ron-Zewi, "Space complexity in polynomial calculus," in *Proc. 27th Annual IEEE Conference on Computational Complexity (CCC '12)*, Jun. 2012, pp. 334–344.

[33] M. Furst, J. B. Saxe, and M. Sipser, "Parity, circuits, and the polynomial-time hierarchy," *Mathematical Systems Theory*, vol. 17, no. 1, pp. 13–27, 1984.

[34] Z. Galil, "On resolution with clauses of bounded size," *SIAM Journal on Computing*, vol. 6, no. 3, pp. 444–459, 1977.

[35] R. E. Gomory, "An algorithm for integer solutions of linear programs," in *Recent Advances in Mathematical Programming*, R. Graves and P. Wolfe, Eds. New York: McGraw-Hill, 1963, pp. 269–302.

[36] D. Grigoriev, "Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity," *Theoretical Computer Science*, vol. 259, no. 1–2, pp. 613–622, May 2001.

[37] D. Grigoriev, E. A. Hirsch, and D. V. Pasechnik, "Complexity of semi-algebraic proofs," in *Proc. 19th International Symposium on Theoretical Aspects of Computer Science (STACS '02)*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2285, pp. 419–430.

[38] ——, "Complexity of semialgebraic proofs," *Moscow Mathematical Journal*, vol. 2, no. 4, pp. 647–679, 2002.

[39] D. Grigoriev and N. Vorobjov, "Complexity of null- and positivstellensatz proofs," *Annals of Pure and Applied Logic*, vol. 113, no. 1–3, pp. 153–160, Dec. 2001.

[40] M. Göös and T. Pitassi, "Communication lower bounds via critical block sensitivity," in *Proc. 46th Annual ACM Symposium on Theory of Computing (STOC '14)*, May 2014, to appear.

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

[42] J. Håstad, "Computational limitations of small-depth circuits," Ph.D. dissertation, Massachussetts Institute of Technology, 1987.

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

[44] J. B. Lasserre, "An explicit exact SDP relaxation for nonlinear 0-1 programs," in *Proc. 8th International Conference on Integer Programming and Combinatorial Optimization*, ser. Lecture Notes in Computer Science, vol. 2081. Springer, Jun. 2001, pp. 293–303.

[45] M. Laurent, "A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming," *Mathematics of Operations Research*, vol. 28, pp. 470–496, 2001.

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

[47] J. P. Marques-Silva and K. A. Sakallah, "GRASP: A search algorithm for propositional satisfiability," *IEEE Transactions on Computers*, vol. 48, no. 5, pp. 506–521, May 1999, preliminary version appeared in *ICCAD '96*.

[48] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik, "Chaff: Engineering an efficient SAT solver," in *Proc. 38th Design Automation Conference (DAC '01)*, Jun. 2001, pp. 530–535.

[49] R. O'Donnell and Y. Zhou, "Approximability and proof complexity," in *Proc. 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '13)*, Jan. 2013, pp. 1537–1556.

[50] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, May 2000.

[51] P. Pudlák, "On the complexity of propositional calculus," in *Sets and Proofs*, ser. London Mathematical Society Lecture Note Series, S. B. Cooper and J. K. Truss, Eds. Cambridge University Press, 1999, vol. 258, pp. 197–218.

[52] ——, "Proofs as games," *American Mathematical Monthly*, pp. 541–550, 2000.

[53] A. A. Razborov, "Lower bounds for the polynomial calculus," *Computational Complexity*, vol. 7, no. 4, pp. 291–324, Dec. 1998.

[54] G. Schoenebeck, "Linear level Lasserre lower bounds for certain k-CSPs," in *Proc. 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, Oct. 2008, pp. 593–602.

[55] H. D. Sherali and W. P. Adams, "A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems," *SIAM Journal on Discrete Mathematics*, vol. 3, pp. 411–430, 1990.

[56] A. Urquhart, "Hard examples for resolution," *Journal of the ACM*, vol. 34, no. 1, pp. 209–219, Jan. 1987.