Electronic Colloquium on Computational Complexity, Report No. 126 (2007)

On the relative efficiency of resolution-like proofs and ordered binary decision diagram proofs

Nathan Segerlind

November 4, 2007

Abstract

We show that tree-like OBDD proofs of unsatisfiability require an exponential increase ($s \mapsto 2^{s^{\Omega(1)}}$) in proof size to simulate unrestricted resolution, and that unrestricted OBDD proofs of unsatisfiability require an almost-exponential increase ($s \mapsto 2^{2^{(\log s)^{\Omega(1)}}}$) in proof size to simulate $\mathrm{Res}(O(\log n))$. The "OBDD proof system" that we consider has lines that are ordered binary decision diagrams in the same variables as the input formula, and is allowed to combine two previously derived OBDDs by any sound inference rule. In particular, this system abstracts satisfiability algorithms based upon explicit construction of OBDDs and satisfiability algorithms based upon symbolic quantifier elimination.

1 Introduction

A great amount of effort is put into developing algorithms for the Boolean satisfiability problem. This is understandable; Boolean satisfiability is a flexible and expressive framework for formulating computational problems from domains as diverse as optimization, planning and formal verification. The Boolean satisfiability problem is NP-complete, so it is doubted that there is an algorithm for efficiently solving all satisfiability instances. However, even if we cannot automatically solve *all* instances of Boolean satisfiability efficiently, the more instances that can be solved automatically and efficiently, the better.

Presently, the state-of-the-art satisfiability algorithms are variants DPLL search with clause learning [32, 34, 18, 17]. All of these are implementations of propositional resolution [42], so we call them *resolution based solvers*. Resolution based solvers have several inherent limitations, the most famous being their need for exponential time to perform simple counting arguments such as "$n+1$ objects cannot be placed injectively into n holes" [21, 46, 14, 7, 39, 5]. Motivated by the weaknessses of resolution-based solvers, a number of algorithms for solving Boolean satisfiability using *ordered binary decision diagrams* have been proposed [10, 45, 19, 12, 13, 1, 36, 35, 3, 15, 38, 22, 4, 24]. These techniques show promising behavior on many benchmarks. Several of these algorithms efficiently generate proofs of unsatisfiability for CNFs known to require exponential running times for resolution based methods. The question we address in this paper is "For which satisfiability instances are resolution-like techniques more efficent than OBDD-based approaches?"

At first thought, there should be *no* instances for which resolution does better than OBDD-based methods: An OBDD-based technique known as symbolic quantifier elimination can efficiently simulate resolution [4]. However, comparisons between software implementations of symbolic quantifier elimination algorithms and resolution-based algorithms have revealed incomparable behavior: There are satisfiability instances for which OBDD-based methods perform much better than

ISSN 1433-8092

resolution-based methods, but also instances for which resolution-based methods perform much better than OBDD-based methods [38, 22]. The first main result of the paper suggests that the inference structure of the OBDD derivations generated by these algorithms is partly to blame for this failure to realize theoretical potential. DPLL with clause learning algorithms generate "DAG-like" derivations whereas existing OBDD-based solvers using explicit construction and symbolic quantifier elimination methods [19, 1, 38, 22, 44] construct "tree-like" derivations. We show that tree-like OBDD derivations cannot efficiently simulate DAG-like resolution derivations.

Theorem 1 *For infinitely many values of n, there exists a CNF ϕ_n in n variables such that ϕ_n has a resolution refutation of size $n^{O(1)}$ but all tree-like OBDD refutations of ϕ_n have size at least $2^{n^{\Omega(1)}}$.*

The second question that we address in this paper is "Which propositional arguments can be efficiently simulated by DAG-like OBDD methods?". Not only can unrestricted OBDD proofs of unsatisfiability efficiently simulate resolution proofs of unsatisfiability [4], they can simulate several other interesting proof systems, such as Gaussian refutations over a finite field, and cutting planes refutations with unary coefficients [4].

There are parallels between the efficiency of a propositional proof system, and the computational power of the circuits which may be used as lines of the proof system. It is known that there are functions computable by $(2+\lceil \log n \rceil)$-DNFs that cannot be computed by any OBDD of subexponential size, cf. [33]. In light of the inability of OBDDs to efficiently simulate $(2+\lceil \log n \rceil)$-DNFs, one might suspect that OBDD refutations cannot p-simulate propositional proofs that manipulate $(2+\lceil \log n \rceil)$-DNFs, ie. Res(k) (k-DNF resolution) with $k = 2+\lceil \log n \rceil$ [28]. The second main result of the paper shows:

Theorem 2 *For infinitely many values of n, there exists a CNF ϕ_n in n variables such that ϕ_n has a Res($O(\log n)$) refutation of size $n^{O(\log n)}$ but all OBDD refutations of ϕ_n have size at least $2^{n^{\Omega(1)}}$.*

1.1 Using OBDDs to solve Boolean satisfiability

An OBDD is a read-once branching program in which the variables appear according to a fixed order along every path (ie. the nodes are arranged in levels, all nodes at a level query the same variable, and each variable corresponds to at most one level) [10, 11, 33]. The ordering restriction enforces canonicity: For each fixed ordering, the OBDD computing a Boolean function is unique up to a linear-time computable normal form, cf. [33]. Because of this canonicity property, the equality test for two Boolean functions represented as OBDDs is simply a check that their OBDDs are identical. However, the choice of variable ordering can affect the size of the OBDD by an exponential factor and choosing a suitable variable ordering for a task is of utmost importance.

Theorems 1 and 2 apply to any OBDD-based derivation system that manipulates OBDDs in the same variables as the input formulas and applies sound inference rules. In particular, Theorem 1 and Theorem 2 apply to the OBDD-derivations generated to two classes of OBDD-based satisfiability algorithms, explicit construction and symbolic quantifier elimination. These results do not clearly apply to a third class of OBDD-based satisfiability algorithms, "compressed resolution" or "compressed search" techniques [12, 13, 35, 36, 37], as those techniques construct OBDDs in variables different from those of the input CNF.

Explicit construction. In the literature, this is sometimes called the "OBDD apply" method. In this method, a variable ordering is selected, the OBDD for the CNF with respect to that ordering

is constructed, and it is checked whether this OBDD is the constant false [10]. There are two opportunities for cleverness - the variable ordering used to construct the OBDDs, and the order in which the clauses are joined together, cf. [45, 1, 22]. Empirical studies [45, 15] and a mathematical analysis of the restricted algorithm in which the clauses are conjoined in the same order as the input presentation [20] have suggested that this method is incomparable with resolution.

Symbolic quantifier elimination. This method extends the explicit construction method by strategically eliminating variables via the application of existential quantifiers [19, 1, 38, 22, 44]. To determine if a CNF $\bigwedge_{i=1}^{m} C_i(\vec{x})$ is satisfiable, rather than build an OBDD for $\bigwedge_{i=1}^{m} C_i(\vec{x})$, it suffices to build one for $\exists \vec{x} \bigwedge_{i=1}^{m} C_i(\vec{x})$. This is can be more efficient because it is often the case that the OBDD for $\exists \vec{x} F(\vec{x}, \vec{y})$ are significantly smaller than the OBDD for $F(\vec{x}, \vec{y})$. One example of this approach is to first heuristically partition the variables into sets $X_1, \ldots X_k$ and the clauses into sets $A_1, \ldots A_k$ so that for each $i = 1, \ldots k$, the variables of X_i do not appear in the clauses belonging to sets $A_{i+1}, \ldots A_k$, then construct the OBDD for the quantified Boolean formula:

$$\exists X_k \left(\ldots \left(\exists X_2 \left(\exists X_1 \bigwedge_{C \in A_1} C(X_1, \ldots X_k) \right) \wedge \bigwedge_{C \in A_2} C(X_2, \ldots X_k) \right) \ldots \right) \wedge \bigwedge_{C \in A_k} C(X_k)$$

1.2 Inference structure: Tree-like versus DAG-like derivations

The proofs of unsatisfiability that we consider in this article, be they OBDD derivations, resolution derivations, or Res($O(\log n)$) derivations, proceed by starting with the clauses of the input CNF, and repeatedly applying inference rules to derive new constraints, until a contradiction is obtained. The expressions derived in this manner can be arranged in an acyclic graph, placing clauses from the input CNF on the sources, and making the antecedents of the inference used to derive an expression the parents of that expression in the DAG. Arbitrary derivations are said to be *DAG-like*, and a derivation is said to be *tree-like* if the DAG is a directed tree. This is a natural distinction to make, and some satisfiability algorithms generate DAG-like derivations whereas others generate tree-like derivations.

Sometimes DAG-like derivations can be exponentially more efficient than tree-like derivations (this is the case for resolution [8, 6]), but sometimes tree-like derivations can simulate DAG-like derivations with only a polynomial increase in derivation size (this is the case for Frege derivations [25, 26]). One consequence of Theorem 1 is that DAG-like OBDD derivations can be exponentially more succinct than tree-like OBDD derivations.

1.3 Comparison with previous work

In [20], Groote and Zantema prove that certain limited OBDD derivations cannot polynomially simulate resolution refutations. Their results apply to the a system that builds an OBDD for the input CNF by conjoining the clauses of the CNF in the order of the input listing (ie. to process $C_1 \wedge (C_2 \wedge C_3)$, an OBDD for $C_2 \wedge C_3$ is built and then one for $C_1 \wedge (C_2 \wedge C_3)$ is built). In fact, in that paper they give a size lower bound for refutations of a formula of the form $\neg x \wedge (x \wedge \psi)$, which is trivial to refute if the formula is processed as $(\neg x \wedge x) \wedge \psi$. Theorem 1 strengthens their result qualititatively by applying to methods that perform a preprocessing step to choose a more efficient order for processing the clauses, and indeed, it applies to any tree-like method of constructing OBDDs, even those using symbolic quantifier elimination.

Jan Krajíček's exponential lower bound for general OBDD refutations, as stated in [29], shows that general OBDD refutations require an exponential increase in size to simulate Frege systems.

However, the techniques of [29] can be applied to a slightly different CNF to show that OBDD refutations require an almost-exponential increase in size to simulate a constant-depth Frege system whose lines are depth-three formulas of the form "OR of ANDs of small ORs". Theorem 2 qualitatively strengthens the result of [29] by showing that OBDD refuations require an almost-exponential increase in size to simulate a constant-depth Frege system whose lines are depth-two formulas of the form "ORs of small ANDs". Similarly, Theorem 1, qualitatively strengthens the result of [43] demonstrating exponential lower bounds for tree-like OBDD refutations, as the result stated in [43] shows that tree-like OBDD systems cannot efficiently simulate Frege systems.

In [4], Atserias, Kolaitis, and Vardi formalized the OBDD-based propositional proof system incorporating symbolic quantifier elimination, and proved that for each fixed variable ordering, there is a CNF of size N that requires size $2^{N^{\Omega(1)}}$ to refute in the OBDD proof system using that particular variable ordering. It turns out that the techniques of [4] can be used to show that for each fixed variable ordering, there is a CNF with quasipolynomial size $\text{Res}(O(\log n))$ refutation that requires exponential size to refute with an OBDD refutation *using that particular variable ordering*. Theorem 2 qualitatively strengthens their result by demonstrating a fixed CNF with quasipolynomial size $\text{Res}(O(\log n))$ refutations that requires exponential size to refute with OBDD refutation *using any possible variable ordering*.

1.4 Outline of the paper and the proof techniques

We begin with a discussion of the basic OBDD, resolution and $\text{Res}(k)$ refutation systems in Section 2. Both Theorem 1 and Theorem 2 are proved by an indirect application of feasible interpolation, which we recap in Section 3.

The high-level proof strategy is similar to that used in [29]: The core technique is a translation that takes a CNF F and creates a new CNF "$perm(F^m)$"[1] such that for any ordering of the variables of $perm(F^m)$, $\prec$, and any ordering of the variables of F, $\prec^*$, the existence of a small OBDD refutation of $perm(F^m)$ with respect to $\prec$ implies the existence of a small OBDD refutation of F with respect to the order $\prec^*$ (Lemma 13). We then take standard CNFs F that are known to require exponentially large tree-like OBDD refutations (general OBDD refutations) with a special fixed variable ordering, and then apply Lemma 13 to show that $perm(F^m)$ requires exponentially large refutations with respect to *any* variable ordering.

Some care is needed to show that when there are small resolution ($\text{Res}(O(\log n))$) refutations of F, there are also small resolution ($\text{Res}(O(\log n))$) refutations of $perm(F^m)$. This is the difference between our our translation method and that used in [29]. Our translation is more "depth efficient" in the sense that when F has a polynomial-size resolution refutation (of a certain form), $perm(F^m)$ will have a polynomial-size resolution refutation, and when F has a quasipolynomial size $\text{Res}(O(\log n))$ refutation (of a certain form), $perm(F^m)$ will have quasipolynomial size $\text{Res}(O(\log n))$ refutation. These properties do not hold for the translation of [29], as that translation increases refutation depth.

The translation from F to $perm(F^m)$ is presented in a sequence of steps in Section 4. Each variable of F is replaced by a disjunction of m new variables, the new formula is converted to a CNF by applying deMorgan's law and the distributive rule, and the resulting CNF is permuted according to small almost-universal family of permutations. With each step of the translation, we

[1]The notation $perm(F^m)$ is chosen because each variable of F is replaced by a disjunction of m new variables, and then the CNF is permuted according to small almost-universal family of permutations, see Section 4. For the purposes of this outline, the reader can think of it simply as "the translation of F".

show that if the initial CNF F has a resolution ($\mathrm{Res}(O(\log n))$) refutation that satisfies certain properties, $perm(F^m)$ has a resolution ($\mathrm{Res}(O(\log n))$) refutation that is not much larger.

The key "reordering lemma", Lemma 13, is presented in Section 5. The intuition is as follows: Suppose you have mn balls, with m in each of n distinct color classes, and they are arbitrarily distributed in n bins, with each bin containing m balls. You wish to select some permutation of the mn balls so that after renaming according to the permutation, each of the n bins now contains a ball from each the n color classes. An average permutation would place $\frac{m}{n}$ balls from a given color class in a given bin. By taking m sufficiently large with respect to n, and applying a second-moment calculation, you can guarantee that every bin contains at least one ball from every color class, and you need only draw the permutation from a small almost-universal family of permutations. This suffices to prove the reordering lemma because the color classes are simply the new variables $\{y_{i,j} \mid j \in [m]\}$ that correspond to each variable x_i of F, and the bins are simply the "blocks" of y variables in positions 1 through m, $m+1$ through $2m$, etc, according to refutation order $\prec$ used to refute $perm(F^m)$. When x_i occurs in position k according to $\prec^*$, we just apply the guaranteed permutation and pick out some $y_{i,j}$ from the k'th block of $y_{i,j}$ variables.

We prove Theorem 1 in Section 6, and we prove Theorem 2 in Section 7. At this point, we apply Lemma 14 to allow the application of previously established interpolation bounds for particular CNFs, and then carefully inspect the known refutations of those CNFs to demonstrate that the translation does not increase refutation size too much.

2 Background and notation

A *literal* is a variable or its negation. For a propositional variable x, we will write x^0 to denote $\neg x$ and x^1 to denote x. A *term* is a constant 0 or 1 or a conjunction of literals. Our convention is that a term is specified as a set of literals, with 1 corresponding to the empty set and 0 to any literal and its negation. We say that a term T contains a literal l if $l \in T$, and that a term T contains a variable x if either $x \in T$ or $\neg x \in T$. We will often identify literals with terms of size one, and will write l instead of $\{l\}$. A *DNF* is a disjunction of terms, specified as a set of terms; we sometimes write $F = \bigvee_i T_i$ and sometimes write $T_i \in F$. A *k-DNF* is a DNF whose terms are each of size at most k. A DNF is said to be *positive* if all of its terms contain only positive literals. A *clause* is a 1-DNF, i.e. a disjunction of literals. The *width* of a clause C, written $w(C)$, is the number of literals appearing in C. The width of a set of clauses is the maximum width of any clause in the set. The *negative width* of a clause C, written $nw(C)$, is the number of negative literals appearing in C. The negative width of a set of clauses is the maximum negative width of any clause in the set. A *CNF* is a conjunction of clauses, specified as a set of clauses. A *k-CNF* is a CNF whose clauses are each of width at most k. A *Boolean circuit* is a directed acyclic graph with a unique sink whose source nodes are labeled with literals and whose internal nodes are partitioned into "AND gates" and "OR gates", and for a given setting to the variables, the output of the circuit is computed in the usual manner. A *Boolean formula* is a circuit whose underlying DAG is a directed tree. A Boolean circuit is said to be *monotone* if its source nodes are labeled only with positive literals.

A *restriction* ρ is a map from a set of variables to $\{0, 1, *\}$. For a formula F, *the restriction of F by ρ*, $F \restriction_\rho$ is defined as usual: We subtitute 0 for x with $\rho(x) = 0$, 1 for x with $\rho(x) = 1$, and simplify only when a sub-expression has become explicitly constant. For any restriction ρ, let $\mathrm{dom}(\rho)$ denote the set of variables to which ρ assigns the value 0 or 1.

2.1 Ordered binary decision diagrams and satisfiability algorithms

Definition 2.1 *(cf. [11, 33]) A* binary decision diagram *(also known as a* branching program*) is a rooted, directed acyclic graph in which every nonterminal node u labeled by a variable x_u and has two out-arcs, one called t_u and the other called f_u. Sinks are labeled by Boolean values. The function represented by a branching program is calculated by starting at the root and following a path to the sink as follows: If the current node u is labeled by the variable x_u, and x_u is assigned the value true, then follow the arc t_u, otherwise follow the arc labeled f_u. The value that the function takes is the value labeled on the sink that is reached at the end of this process. The* size *of a binary decision diagram is its number of nodes as a DAG. An* ordered binary decision diagram (OBDD) *is a binary decision diagram in which: Along every path from the source to a sink, every variable is queried at most once, and, there is fixed ordering so that along all paths from the source to a sink, the variables are queried consistently with that order. When F is an OBDD, and π is a fixed ordering of the variables used by F, F is said to be a* π-OBDD *if the variable ordering used by F is consistent with π.*

Definition 2.2 *Let $\mathcal{C}$ be a set of clauses in variables from a set V. An* OBDD derivation from $\mathcal{C}$ with respect to a variable ordering $\preceq$ on V *is a sequence of OBDDs $F_1, \ldots, F_m$ so that each OBDD is built from the variables of V with respect to the order $\preceq$, and each F_i either is a clause in $\mathcal{C}$, or follows from the preceding $F_1, \ldots F_{i-1}$ by an application of the following inference rule: If A and B are previously derived OBDDs, we may infer any OBDD C such that $A \wedge B \Rightarrow C$. For a set of clauses $\mathcal{C}$, an* OBDD refutation of $\mathcal{C}$ *is a derivation from $\mathcal{C}$ whose final line is the OBDD "false". The size of an OBDD refutation is the sum of the sizes of its OBDDs.*

This inference rule is very powerful. However, because whether or not $A \wedge B \Rightarrow C$ can be decided in time polynomial in the sizes of OBDDs A, B and C (cf. [33]), the OBDD refutation system is a refutation system in the sense of Cook and Reckhow [16].

OBDD based satisfiability algorithms based upon explicit construction and symbolic quantifier elimination generate proofs of unsatisfiability by using far more restricted inference rules. Let A and B be OBDDs in the variables V with ordering $\preceq$, where $\vec{x}$, $\vec{y}$, $\vec{z}$ are tuples of variables from V:

$$\text{Conjunction: } \frac{A(\vec{x},\vec{y}) \quad B(\vec{y},\vec{z})}{A(\vec{x},\vec{y}) \wedge B(\vec{y},\vec{z})} \qquad \text{Projection: } \frac{A(x,\vec{y})}{\exists x A(x,\vec{y})}$$

The explicit construction method for solving satisfiability constructs OBDD derivations using only the conjunction rule, and symbolic quantifier elimination methods use only the conjunction and projection rules. Notice that each of these inference rules is a special case of the the OBDD system's one inference rule.

2.2 Resolution and Res(k)

Resolution is a refutation system for propositional logic. The input to a resolution refutation is a set of clauses $\mathcal{C}$; a resolution refutation consists of a derivation of the empty clause from the clauses in $\mathcal{C}$ using only the resolution inference: $\frac{A \vee x \quad \neg x \vee B}{A \vee B}$. Notice that every line in a resolution refutation is a clause. The Res(k) refutation system is a generalization of resolution that can reason using k-DNFs.

Definition 2.3 *[28] Res(k) is the refutation system whose lines are k-DNFs and whose inference rules are given below (the variable set is $\{x_1, \ldots x_n\}$, A, A^*, and B are k-DNF's, with $A \subseteq A^*$, $1 \leq j \leq k$, l is a literal, $I_1, \ldots I_j \subseteq [n]$ with $I = I_1 \cup \cdots \cup I_j$, and $|I| \leq k$, and $J \subseteq I$):*

Subsumption: $\dfrac{A}{A^*}$ *AND-introduction:* $\dfrac{A \vee \bigwedge_{i\in I_1} l_i \;\cdots\; A \vee \bigwedge_{i\in I_j} l_i}{A \vee \bigwedge_{i\in I} l_i}$

Cut: $\dfrac{A \vee \bigwedge_{i\in I} l_i \quad B \vee \bigvee_{i\in I} \neg l_i}{A \vee B}$ *AND-elimination:* $\dfrac{A \vee \bigwedge_{i\in I} l_i}{A \vee \bigwedge_{i\in J} l_i}$

Let $\mathcal{C}$ be a set of k-DNFs. A Res(k) derivation from $\mathcal{C}$ *is a sequence of k-DNFs $F_1, \ldots, F_m$ so that each F_i either belongs to $\mathcal{C}$ or follows from the preceding lines by an application of one of the inference rules. For a set of k-DNFs $\mathcal{C}$, a* Res(k) refutation of $\mathcal{C}$ *is a derivation from $\mathcal{C}$ whose final line is the empty clause. The size of a Res(k) refutation is the number of lines it contains.*

3 Feasible interpolation

The lower bounds on OBDD refutation sizes that we prove are based upon feasible interpolation results, which are in turn based upon communication complexity. This is a well-developed method for analyzing the sizes of propositional proofs, cf. [23, 41, 9, 27].

Definition 3.1 *(cf. [30]) Let $m, n \in \mathbb{N}$ be given and let R be a set. Let $f : \{0,1\}^m \times \{0,1\}^n \to R$ be a function of two Boolean inputs. A* two-player communication protocol for f *is a process in which Player I has private access to $x \in \{0,1\}^m$, Player II has private access to $y \in \{0,1\}^n$, the players proceed in turns sending binary strings to one another, and at the end, both players know the value of $f(x,y)$. The* cost of a protocol *is the maximum of the number of bits communicated between the two players, taken over all possible settings of the inputs. The* communication complexity of f *is the minimum cost of a two-player protocol that computes f.*

Definition 3.2 *Let $n, s, t \in \mathbb{N}$ be given. Let $A \subseteq \{0,1\}^n \times \{0,1\}^s \times \{0,1\}^t$ be given. The* monotone communication complexity of A, $MCC(A)$, *is the minimum communication between the two players necessary to compute any of the following tasks, when Player I has access only to $(u, y^u) \in \{0,1\}^n \times \{0,1\}^s$ and Player II has access only to $(v, z^v) \in \{0,1\}^n \times \{0,1\}^t$:*

1. *Decide whether $(u, y^u, z^v) \in A$.*
2. *Decide whether $(v, y^u, z^v) \in A$.*
3. *If $(u, y^u, z^v) \in A$ and $(v, y^u, z^v) \notin A$, either find $i \leq n$ such that $u_i = 1$ and $v_i = 0$, or, agree that there exists $u' \geq u$ such that $(u', y^u, z^v) \notin A$.*

Lemma 3 *[29] Let $n, s, t \in \mathbb{N}$ be given, and let V be a set of $n + s + t$ many variables paritioned so that the first n variables are the x's, the middle s variables are the y's, and the last t variables are the z's. Let π be an ordering on the variables that places all y's before all x's and all x's before all z's. Let P be a π-OBDD of size S, and let $A = P^{-1}(1)$. $MCC(A) = O((\log S)(\log n))$*

We use the following monotone interpolation theorem.

Theorem 4 *(cf. [29]) Let $n, s, t \in \mathbb{N}$ be given, and let V be a set of $n + s + t$ many variables paritioned so that the first n variables are the x's, the middle s variables are the y's, and the last t variables are the z's. Let π be a linear ordering on $vars(F)$ that is consistent with $y < x < z$. Let*

$A_1, \ldots A_m \subseteq \{0,1\}^{n+s}$ *and* $B_1, \ldots B_l \subseteq \{0,1\}^{n+t}$. *Furthermore, suppose that the sets* $A_1, \ldots A_m$ *satisfy the following monotonicity condition:*

$$\left((u, y^u) \in \bigcap_{j=1}^{m} A_j \right) \wedge (u \leq u') \rightarrow (u', y^u) \in \bigcap_{j=1}^{m} A_j$$

For each $i \in [m]$ *let* $\tilde{A}_i = \{(x,y,z) \mid (x,y) \in A_i\}$ *and for each* $j \in [m]$ *let* $\tilde{B}_j = \{(x,y,z) \mid (x,z) \in B_j\}$. *If there is an OBDD refutation of the sets* $\tilde{A}_1, \ldots \tilde{A}_m, \tilde{B}_1, \ldots \tilde{B}_l$ *of size* S *then the two sets* $U = \{u \in \{0,1\}^n \mid \exists y^u \in \{0,1\}^s,\ (u, y^u) \in \bigcap_{j=1}^{m} A_j\}$ *and* $V = \{v \in \{0,1\}^n \mid \exists z^v \in \{0,1\}^t,\ (v, z^v) \in \bigcap_{j=1}^{l} B_j\}$ *can be separated by a monotone circuit of size at most* $S^{O(\log n)}$. *Furthermore, if there is a tree-like OBDD refutation of the sets* $\tilde{A}_1, \ldots \tilde{A}_m, \tilde{B}_1, \ldots \tilde{B}_l$ *of size* S *then the two sets* $U = \{u \in \{0,1\}^n \mid \exists y^u \in \{0,1\}^s,\ (u, y^u) \in \bigcap_{j=1}^{m} A_j\}$ *and* $V = \{v \in \{0,1\}^n \mid \exists z^v \in \{0,1\}^t,\ (v, z^v) \in \bigcap_{j=1}^{l} B_j\}$ *can be separated by a monotone formula of size at most* $S^{O(\log n)}$.

4 The translation

4.1 Blowing up a CNF

For the first stage of the translation, we substitute a disjunction of new variables for old variables, and make repeated use of the deMorgan's law and the distribution of ANDs across ORs: $X \vee \neg(Y \vee Z)$ is equivalent to $(X \vee \neg Y) \wedge (X \vee \neg Z)$.

Definition 4.1 *Let* F *be a CNF in variables* $x_1, \ldots x_n$, *and let* $m \geq 1$ *be an integer. The* m-th disjunctive blow-up of F, F^m, *is the CNF on the new variables* $y_{i,j}$, *with* $i \in [n]$ *and* $j \in [m]$, *constructed by substituting the disjunction* $\bigvee_{j=1}^{m} y_{i,j}$ *for each variable* x_i, *and then applying deMorgan's law and the distributive law to convert the new formula into CNF form.*

To make our arguments more uniform, we generalize the blowing-up construction.

Definition 4.2 *Let* F *be a* k_0*-DNF in variables* $\{x_1, \ldots x_n\}$. *For each* $i = 1, \ldots n$, *let* D_i *be a* k_1*-DNF in variables* $\{y_1, \ldots y_N\}$. *Let* $\Sigma_D(F)$ *denote the set of* $(k_0 k_1)$*-DNFs obtained by substituting* $D_i(\vec{y})$ *for each occurrence of* x_i, *applying deMorgan's law, and the distributive law.*

In this article, we perform this transformation only when the formula F has a particular form.

Definition 4.3 *Let* F *be a* k*-DNF. We say that* F contains no negations in compound terms *if all terms of* $t \in F$ *with* $|t| \geq 2$ *are conjunctions of positive literals. We say that* F has negative width at most w *if the number of terms containing negative literals is at most* w.

Lemma 5 *Let* F *be a* k_0*-DNF in the variables* $\{x_1, \ldots x_n\}$, *and, for each* $i = 1, \ldots n$, *let* D_i *be a* k_1*-DNF in variables* $\{y_1, \ldots y_N\}$. *If no compound term of* F *contains a negative literal, then, setting* F_0 *to be the part of* F *containing all purely positive terms, and setting* G_0 *to be the single* $(k_0 k_1)$*-DNF in* $\Sigma_D(F_0)$, *we have that:*

$$\Sigma_D(F) = \{G_0 \vee \bigvee_{\neg x_i \in F} \bigvee_{l \in T_i} \neg l \mid T \in \prod_{\neg x_i \in F_1} D_i\}$$

Corollary 6 *Let F be a k_0-DNF in $\{x_1, \ldots x_n\}$, and, for each $i = 1, \ldots n$, let D_i be a positive k_1-DNF in variables $\{y_1, \ldots y_N\}$. Suppose that F contains no negated literals in compound terms, that each D_i contains at most m many terms, and that F contains at most w many negated literals.*

1. *$\Sigma_D(F)$ contains at most m^w many (k_0k_1)-DNFs.*
2. *For every $G \in \Sigma_D(F)$, G contains at most wk_1 many negated literals.*
3. *For every $G \in \Sigma_D(F)$, G contains no negated literals in compound terms.*

Corollary 7 *If F is a CNF in n variables with S many clauses, each of which has negative-width at most w, then F^m contains at most $m^w S$ many clauses.*

Definition 4.4 *We say that a Res(k) refutation Γ is in* no-compound-negation form *if every line of Γ contains no negations in compound terms. We say that Γ has negative width at most w if every line of Γ has negative width at most w.*

The following lemma follows from a standard "apply the translation and then clean-up the inference steps" argument that is placed in the Appendix.

Lemma 8 *(proof in the Appendix) Let $\mathcal{F}$ be a set of DNFs in the variables $\{x_1, \ldots x_n\}$, and, for each $i = 1, \ldots n$, let D_i be a positive k_1-DNF in variables $\{y_1, \ldots y_N\}$, such that each D_i contains at most m many terms. If $\mathcal{F}$ has a Res(k_0) refutation of size S in non-compound-negation form, and of negative width at most w, then $\Sigma_D(\mathcal{F})$ has a Res(k_0k_1) refutation of size at most $2m^{w+k_0}S$.*

Corollary 9 *If F is a CNF in n variables with s many clauses, each of which has negative-width at most w, and F has a Res(k) refutation of size S and negative width at most w, then F^m has a Res(k) refutation of size at most $2m^{w+1}S$.*

4.2 Permuting the variables

A standard trick for confounding OBDDs is to apply a permutation to the input variables. For our purposes, it suffices to use permutations of one particular kind, essentially the almost-universal hash functions of Carter and Wegman [47].

Definition 4.5 *[47] Let $t \in \mathbb{N}$ be given. Let $\mathbb{F} = GF(2^t)$. Define the set Π_t to be the set of all mappings given by $x \mapsto ax + b$ with $a, b \in \mathbb{F}$, $a \neq 0$*

Lemma 10 *[47] Let $t \in \mathbb{N}$ be given. $|\Pi_t| = 2^t(2^t - 1)$. Every mapping in Π_t is a permutation of $[2^t]$. Furthermore, for each $x_1, x_2, y_1, y_2 \in [2^t]$ such that $x_1 \neq x_2$ and $y_1 \neq y_2$, $Pr_{f \in \Pi_t}[f(x_1) = y_1,\ f(x_2) = y_2] = \frac{1}{2^t(2^t-1)}$.*

Definition 4.6 *Let F be a CNF in the variables $x_1, \ldots x_n$. Set $t = \lceil \log_2 n \rceil$. Let $x_{n+1}, \ldots x_{2^t}$ be completely new variables. Use $l = 2t$ many new variables, $z_1, \ldots z_l$, to encode the permutations of Π in some surjective fashion. The CNF perm(F) is the CNF obtained as follows: For each assignment to the z's, $\alpha \in \{0,1\}^l$, let π denote the permutation from Π encoded α. For every π from Π and every clause $\bigvee_{i \in I} x_i^{\epsilon_i}$ from F, there is a clause $\bigvee_{i \in [l]} z_i^{1-\alpha_i} \vee \bigvee_{i \in I} x_{\pi(i)}^{\epsilon_i}$ in perm(F). For each x_i and $\pi \in \Pi$, define $\pi(x_i)$ to be $x_{\pi(i)}$.*

Lemma 11 *Let F be a CNF in n variables. The number of variables in $perm(F)$ is $\leq 2n + 2\log n$. If F contains S many clauses, then $perm(F)$ contains at most $4n^2 S$ many clauses.*

Lemma 12 *If there is a size S resolution refutation of F then there is a resolution refutation of $perm(F)$ of size $< 6n^2 S$.*

Proof: For each value of $\vec{z}$, there is a size S refutation of $perm(F) \restriction_{\vec{z}}$. Therefore there is a size $2^l S$ derivation of the set of clauses $\{\bigvee_i z_i^{v_i} \mid \vec{v} \in \{0,1\}^l\}$ from $perm(F)$. The resulting set of clauses is the complete-tree contradictions on l many variables, and it is well-known that these CNFs have resolution refutations of size at most $2^{l+1} - 1$. Therefore the size of the refutation is at most $2^l S + 2^{l+1} - 1 \leq 2n^2 S + 4n^2 - 1 < 6n^2 S$.

■

Lemma 13 *Let F be a CNF on the variables $\{x_1, \ldots x_n\}$. Set $t = \lceil \log_2 n \rceil$, set $N = 2^t$. Let Γ be an OBDD refutation of $perm(F)$, whose variable ordering on $\{x_1, \ldots x_N\}$ is $v_1, \ldots v_N$. For every $\pi \in \Pi_t$, there is an OBDD refutation of F, in the variables $\{x_1, \ldots x_n\}$, with size at most $|\Gamma|$, and that uses a variable ordering consistent with $\pi(v_1), \ldots \pi(v_N)$.*

Proof: Let α be an assignment to $\vec{z}$ that selects the permutation π^{-1}. We apply the restriction α to Γ, and we see that the clauses of $perm(F) \restriction_\alpha$ that survive are exactly the clauses of the form $\bigvee_{i \in I} x_{\pi^{-1}(i)}^{\epsilon_i}$ where $\bigvee_{i \in I} x_i^{\epsilon_i}$ is a clause of F. We now rename the variables according to π: Within each OBDD of the refutation, replace each query to x_i by a query to $x_{\pi(i)}$. Every OBDD is now constructed according to the order $\pi(v_1), \ldots \pi(v_N)$. For each $\bigvee_{i \in I} x_i^{\epsilon_i}$ in F, the OBDD for $\bigvee_{i \in I} x_{\pi^{-1}(i)}^{\epsilon_i}$ becomes the OBDD for $\bigvee_{i \in I} x_i^{\epsilon_i}$. Proof structure is preserved because the relation $A \wedge B \Rightarrow C$ between OBDDs is preserved under substitutions to the variables.

■

5 The reordering lemma

Lemma 14 *(The reordering lemma) Let F be a CNF on the variables $\{x_1, \ldots x_n\}$, let $\prec$ be a total ordering of $\{x_1, \ldots x_n\}$, and let $m \in \mathbb{N}$ be given so that $\frac{2n^3}{m} + \frac{n^2}{mn-1} < 1$. If there is an OBDD refutation of the CNF $perm(F^m)$ of size at most S, with respect to* any *ordering of the variables of $perm(F^m)$, then there is an OBDD refutation of F that uses the variable ordering $\prec$ and has size at most S.*

The proof of Lemma 14 uses the following standard consequence of Chebyshev's inequality:

Lemma 15 *(from Chebyshev's Inequality, cf. [2]) Let $X_1, \ldots X_t$ be 0/1 valued random variables, and let $Y = \sum_{i=1}^t X_i$.*

$$\begin{aligned} Pr[Y = 0] &\leq \frac{Var(Y)}{(\mathbb{E}[Y])^2} = \frac{\sum_{i=1}^t Var(X_i) + \sum_{i \neq j} Cov(X_i, X_j)}{(\mathbb{E}[Y])^2} \\ &\leq \frac{\sum_{i=1}^t E(X_i) + \sum_{i \neq j} Cov(X_i, X_j)}{(\mathbb{E}[Y])^2} = \frac{\mathbb{E}[Y] + \sum_{i \neq j} Cov(X_i, X_j)}{(\mathbb{E}[Y])^2} \end{aligned}$$

Proof:(of Lemma 14) Rename the variables so that $\prec$ places them in the order $x_1, x_2, \ldots x_n$. Let $\{y_{i,j} \mid i \in [n],\ j \in [m]\}$ denote $vars(F^m)$. Set $t = \lceil \log mn \rceil$ and set $N = 2^t$. Let $w_{mn+1}, \ldots w_N$ denote the "padding variables" added to F^m to construct $perm(F^m)$. Set $V = \{y_{i,j} \mid i \in [n],\ j \in [m]\} \cup \{w_{mn+1}, \ldots w_N\}$. Let $l = 2t$, and let $z_1, \ldots z_l$ denote the variables of $perm(F)$ that select the permutation.

Let Γ be a size S refutation of $perm(F^m)$ with respect to some ordering of its variables. Let $v_1, \ldots v_N$ be the variable ordering used by Γ as induced on V. We completely disregard the positions of the z-variables in the ordering used by Γ, since we are going to fix them to 0/1 values.

Let $r = N \bmod n$. For each $k = 1, \ldots r$, let V_k be the set of variables from V that are in positions $(k-1)\lceil \frac{N}{n} \rceil + 1$ through $k\lceil \frac{N}{n} \rceil$, according to the order $v_1, \ldots v_N$. For for $k = r+1, \ldots n$, let V_k be the set of variables from V that are in positions $(k-1)\lfloor \frac{N}{n} \rfloor + r + 1$ through $k\lfloor \frac{N}{n} \rfloor + r$, according to the order $v_1, \ldots v_N$.

Choose $\pi \in \Pi$ uniformly at random. For each $i \in [n]$, $j \in [m]$, $k \in [n]$, let $\chi^k_{i,j}$ be the indicator variable for the event that $\pi^{-1}(y_{i,j}) \in V_k$. By Lemma 10, $\chi^k_{i,j}$ has expectation $|V_k|/N$. For each $i \in [n]$, $k \in [n]$, let Y^k_i be the random variable of π defined by $Y^k_i = \sum_{j=1}^m \chi^k_{i,j} = |\{\pi^{-1}(y_{i,j}) \in V_k \mid j \in [m]\}|$. The expectation of Y^k_i is $(m/N)|V_k|$. We will use Lemma 15 to bound the probability that $Y^k_i = 0$. First we bound the covariance of the $\chi^k_{i,j}$'s contributing to Y^k_i. For $j_0, j_1 \in [m]$ with $j_0 \neq j_1$:

$$\begin{aligned}
Cov(\chi^k_{i,j_0}, \chi^k_{i,j_1}) &= \mathbb{E}[\chi^k_{i,j_0} \cdot \chi^k_{i,j_1}] - \mathbb{E}[\chi^k_{i,j_0}]\mathbb{E}[\chi^k_{i,j_1}] \\
&= \left(\sum_{\substack{u,v \in V_k \\ u \neq v}} Pr[\pi^{-1}(y_{i,j_0}) = u,\ \pi^{-1}(y_{i,j_1}) = v] \right) - (|V_k|/N)^2 \\
&= \frac{|V_k|(|V_k|-1)}{N(N-1)} - (|V_k|/N)^2 = \frac{|V_k|}{N}\left(\frac{|V_k|-1}{N-1} - \frac{|V_k|}{N}\right) \\
&< \frac{|V_k|^2}{N}\left(\frac{1}{N-1} - \frac{1}{N}\right) = \frac{|V_k|^2}{N^2(N-1)} = \frac{((m/N)|V_k|)^2}{m^2(N-1)} = \frac{\left(\mathbb{E}[Y^k_i]\right)^2}{m^2(N-1)}
\end{aligned}$$

We can now apply Lemma 15 as follows:

$$\begin{aligned}
\Pr_\pi[Y^k_i = 0] &\leq \frac{1}{\mathbb{E}[Y^k_i]} + \sum_{\substack{j_0, j_1 \in [m] \\ j_0 \neq j_1}} \frac{Cov(\chi^k_{i,j_0}, \chi^k_{i,j_1})}{\left(\mathbb{E}[Y^k_i]\right)^2} \leq \frac{N}{m|V_k|} + \frac{m(m-1)(\mathbb{E}[Y^k_i])^2}{m^2(N-1)(\mathbb{E}[Y^k_i])^2} \\
&< \frac{N}{m\lfloor N/n \rfloor} + \frac{1}{N-1} \leq \frac{2mn}{m^2} + \frac{1}{mn-1} = \frac{2n}{m} + \frac{1}{mn-1}
\end{aligned}$$

Therefore, by the union bound, the probability that there exists an $i \in [n]$ and $k \in [n]$ with $|\{\pi^{-1}(y_{i,j}) \in V_k \mid j \in [m]\}| = 0$ is at most $\frac{2n^3}{m} + \frac{n^2}{mn-1}$. By our choice of m, this is strictly less than 1, so there exists $\pi \in \Pi_t$ so that for all $i \in [n]$ and all $k \in [n]$, $|\{\pi^{-1}(y_{i,j}) \in V_k \mid j \in [m]\}| \geq 1$. For each $i \in [n]$, let j_i be the least element of $\{j \in [m] \mid \pi^{-1}(y_{i,j}) \in V_i\}$. For all $1 \leq i < i' \leq n$, because $\pi^{-1}(y_{i,j_i}) \in V_i$ and $\pi^{-1}(y_{i',j_{i'}}) \in V_{i'}$, y_{i,j_i} precedes $y_{i',j_{i'}}$ in the ordering $\pi(v_1), \ldots \pi(v_N)$.

By Lemma 13, there is a size S OBDD refutation of F^m, in the variables $\{y_{i,j} \mid i \in [n],\ j \in [m]\}$, with respect to an order consistent with $\pi(v_1), \ldots \pi(v_N)$. Call this refutation Γ'. Let σ be the substitution to $\{y_{i,j} \mid i \in [n],\ j \in [m]\}$ given by $\sigma(y_{i,j}) = x_i$, if $j = j_i$, and $\sigma(y_{i,j}) = 0$ otherwise. Let Γ^* be the result of applying σ to Γ'. Γ^* is clearly an OBDD refutation of F, and because the

order $y_{1,j_1}, y_{2,j_2}, \ldots y_{n,j_n}$ is consistent with the variable ordering of Γ', Γ^* is built according to the order $x_1, x_2, \ldots x_n$ - that is, the order $\prec$.

■

6 Tree-like OBDD refutations do not p-simulate DAG-like resolution refutations

The CNF that we use to separate unrestricted resolution from tree-like OBDD refutations is based on the "pyramidal generation function". This function was shown to require large monotone circuit depth by Raz and McKensize [40], and later it was converted into CNF form by Bonet, Esteban, Galesi and Johannsen to show that tree-like cutting planes cannot p-simulate unrestricted resolution [8].

Definition 6.1 *[40] Let $n \in \mathbb{N}$ be given. The function GEN_n is defined as follows. Its inputs are $t \in [n] \times [n] \times [n]$, and $GEN_n(t) = 1$ if and only if $\vdash n$, where for $c \in [n]$, $\vdash c$ is defined recursively via $\vdash c$ if and only if $c = 1$ or there are $a, b \in [n]$ with $\vdash a$, $\vdash b$ and $t_{a,b,c} = 1$.*

Definition 6.2 *[8] Let $d \in \mathbb{N}$ be given. The set Pyr_d is defined as $Pyr_d := \{(i,j) \mid 1 \leq j \leq i \leq d\}$. Let $n \in \mathbb{N}$ be given, and let $t \in \{0,1\}^{n^3}$ be an input to GEN_n. We say that n is generated in a depth d pyramidal fashion by t if there exsists a mapping $m : Pyr_d \to [n]$ so that:*

1. *For every $j \in [d]$, $t_{1,1,m(d,j)} = 1$.*
2. *For every $(i,j) \in Pyr_{d-1}$, $t_{m(i+1,j),m(i+1,j+1),m(i,j)} = 1$.*
3. *$t_{m(1,1),m(1,1),n} = 1$.*

Theorem 16 *[8] There exists $\epsilon > 0$ so that for all $d \in \mathbb{N}$, with $n = \binom{d+1}{2}^{15} + 2$, every monotone Boolean formula that outputs a 1 on inputs to GEN_n for which n is generated in a depth d pyramidal fashion, and outputs 0 on all inputs where GEN_n is 0, must have size at least 2^{n^ϵ}.*

Definition 6.3 *[8] Let d be given, and let $n = \binom{d+1}{2}^{15} + 2$. Let there be a variable $q_{i,j,a}$ for each $(i,j) \in Pyr_d$ and $a \in [n]$, let there be a variable $p_{a,b,c}$ for each $a, b, c \in [n]$, and let there be a variable r_a for each $a \in [n]$. The CNF $ColGen_d(\vec{p}, \vec{q}, \vec{r})$ consists of the following clauses:*

(1)	$\bigvee_{a \in [n]} q_{i,j,a}$	*for $(i,j) \in Pyr_d$*
(2)	$\neg q_{d,j,a} \vee p_{1,1,a}$	*for $1 \leq j \leq d$, $a \in [n]$*
(3)	$\neg q_{1,1,a} \vee p_{a,a,n}$	*for $a \in [n]$*
(4)	$\neg q_{i+1,j,a} \vee \neg q_{i+1,j+1,b} \vee \neg q_{i,j,c} \vee p_{a,b,c}$	*for $(i,j) \in Pyr_{d-1}$ and $a, b, c \in [n]$*
(5)	$\neg p_{1,1,a} \vee \neg r_a$	*for $a \in [n]$*
(6)	$\neg p_{a,a,n} \vee r_a$	*for $a \in [n]$*
(7)	$r_a \vee r_b \vee \neg p_{a,b,c} \vee \neg r_c$	*for $a, b, c \in [n]$*

Furthermore, let $Gen_d(\vec{p}, \vec{q})$ be the CNF given by the clauses of 1, 2, 3, and 4, and let $Col_d(\vec{p}, \vec{r})$ be the CNF given by the clauses of 5, 6, and 7.

The following facts follow immediately from Definition 6.3, and applications of Corollary 7 and Lemma 11.

Lemma 17 *Let d tend to infinity, and let $m : \mathbb{N} \to \mathbb{N}$ be function of d, eg. $m = m(d)$.*

1. *The number of variables in $ColGen_d$ is $\Theta(d^{90})$, the number of clauses in $ColGen_d$ is $O(d^{92})$, and the negative width of $ColGen_d$ is 3.*
2. *The number of variables in $ColGen_d^m$ is $\Theta(md^{90})$, and the number of clauses in $ColGen_d^m$ is $O(m^3 d^{92})$.*
3. *The number of variables in $perm(ColGen_d^m)$ is $\Theta(md^{90})$, and $perm(ColGen_d^m)$ contains $O(m^5 d^{272})$ many clauses.*

6.1 Lower bound for tree-like OBDD refutations

Lemma 18 *Let $d \in \mathbb{N}$ be given. Set $n = \binom{d+1}{2}^{15} + 2$, set $N = |Vars(ColGen_d)|$, and and set $m = 4N^3$. If there is a size S tree-like OBDD refutation of $perm(ColGen_d^m)$, then there is a size $S^{O(\log N)}$ monotone Boolean formula that outputs 1 on instances where t generates n in a depth d pyramidal fashion, and outputs 0 on all instances where t does not generate n.*

Proof: Because $\frac{2N^3}{m} + \frac{N^2}{mN-1} = \frac{2N^3}{4N^3} + \frac{N^2}{4N^4-1} < 1$, we may apply Lemma 14 and conclude that there is a size $\leq S$ tree-like OBDD refutation of $ColGen_d$ with respect to an order that places all $q_{i,j,a}$'s before all $p_{i,j,k}$'s, and all $p_{i,j,k}$'s before all r_a's. By Theorem 4 there is a monotone Boolean formula F of size at most $S^{O(\log n)}$ that separates $Col_d(\vec{p}, \vec{r})$ and $Gen_d(\vec{p}, \vec{q})$. We now show that the formula F must output 1 on instances where t generates n in a depth d pyramidal fashion, and output 0 on all instances where t does not generate n. Consider an assignment $t \in \{0,1\}^{n^3}$ to the variables of $\vec{p}$. If n generated by a depth d pyramid according to $\vec{t}$, then $Gen(\vec{p}, \vec{r})$ is satisfiable. So $Col(\vec{p}, \vec{r})$ is unsatisfiable, and thus $F(\vec{t}) = 1$. On the other hand, if $GEN_n(\vec{t}) = 0$ then $Col(\vec{t}, \vec{r})$ can be satisfied by assigning color 0 to the elements that can be generated under $\vec{t}$. Thus $Gen(\vec{t}, \vec{q})$ is unsatisfiable, so $F(\vec{t}) = 0$. ∎

Combining Lemma 18 with Theorem 16:

Lemma 19 *Let d be an integer tending to infinity and let N be the number of variables in $ColGen_d$. All tree-like OBDD refutations of $perm(ColGen_d^{4N^3})$ have size $2^{N^{\Omega(1)}}$.*

By Lemma 17, when $N = |Vars(ColGen_d)|$, the number of variables in $perm(ColGen_d^{4N^3})$ is polynomial in the number of variables in $ColGen_d$. Therefore, we have as a corollary:

Corollary 20 *Let d be an integer tending to infinity and let N be the number of variables in $ColGen_d$. Let M be the number of variables in $perm(ColGen_d^{4N^3})$. All tree-like OBDD refutations of $perm(ColGen_d^{4N^3})$ have size $2^{M^{\Omega(1)}}$.*

6.2 Upper bound for DAG-like resolution refutations

A resolution refutation of $ColGen_d$ of size $d^{O(1)}$ in presented in [8], and although they were not concerned with negative width, an inspection of their refutation reveals that it has negative width four. We put the argument in the Appendix for completeness.

Theorem 21 *[8] Let $d \in \mathbb{N}$ tend to infinity, and let $N = N(d)$ be the number of variables in $ColGen_d$. The CNF $ColGen_d(\vec{p}, \vec{q}, \vec{r})$ has a DAG-like resolution refutation of size $N^{O(1)}$ and negative width 4.*

Lemma 22 *Let $d \in \mathbb{N}$ tend to infinity, and let $N = N(d)$ be the number of variables in $ColGen_d$. Let $m \in \mathbb{N}$ be given. The CNF $perm(ColGen_d^m)$ has a a resolution refutation of of size $(Nm)^{O(1)}$.*

Proof: By Theorem 21 there is a resolution refutation of $ColGen_d$ of size $d^{O(1)}$ and negative width 4, so by Corollary 9, there is a resolution refutation of $ColGen_d^m$ of size at most $m^5 N^{O(1)}$. So by Lemma 12, there is a resolution refutation of $perm(ColGen_d^m)$ of size at most $4N^2 m^5 N^{O(1)}$. ∎

Corollary 23 *Let $d \in \mathbb{N}$ tend to infinity, let $N = N(d)$ be the number of variables in $ColGen_d$, and let $M = M(d)$ be the number of variables in $perm(ColGen_d^{4N^3})$. The CNF $perm(ColGen_d^{4N^3})$ has a a resolution refutation of of size $M^{O(1)}$.*

Let $d \in \mathbb{N}$ be given, let N be the number of variables in $ColGen_d$, and let $M = M(d)$ be the number of variables in $perm(ColGen_d^{4N^3})$. Corollary 20 shows that tree-like OBDD refutations size $2^{M^{\Omega(1)}}$ to refute $perm(ColGen_d^{4N^3})$, and Corollary 23 shows that resolution can refute $perm(ColGen_d^{4N^3})$ in size $M^{O(1)}$. Combining these two facts proves Theorem 1: Tree-like OBDD refutations cannot p-simulate DAG-like resolution.

7 DAG-like OBDD refutations do not p-simulate Res$(O(\log n))$

Definition 7.1 *[27, 41] Let $l, m, n \in \mathbb{N}$ be given. Let there be propositional variables p_e, for all $e \in \binom{[n]}{2}$, $q_{u,i}$ for all $u \in [m]$, $i \in [n]$, and $r_{i,a}$ for all $i \in [n]$ and $a \in [l]$. The set of clauses $Clique_{m,n}(\vec{p}, \vec{q})$ is given by:*

1. $\bigvee_{i \in [n]} q_{u,i}$ *for all* $u \in [m]$.
2. $\neg q_{u,i} \vee \neg q_{v,i}$ *for all* $u, v \in [m]$ *with* $u \neq v$ *and all* $i \in [n]$.
3. $\neg q_{u,i} \vee \neg q_{v,j} \vee p_{i,j}$ *for all* $u, v \in [m]$ *with* $u \neq v$ *and all* $\{i, j\} \in \binom{[n]}{2}$.

The set of clauses $Color_{n,l}(\vec{p}, \vec{r})$ is given by:

1. $\bigvee_{a \in [l]} r_{i,a}$ *for all* $i \in [n]$.
2. $\neg r_{i,a} \vee \neg r_{i,b}$ *for all* $a, b \in [l]$ *with* $a \neq b$, *and all* $i \in [n]$.
3. $\neg r_{i,a} \vee \neg r_{j,a} \vee \neg p_{i,j}$ *for all* $a \in [l]$, *and all* $\{i, j\} \in \binom{[n]}{2}$.

For each n, set $l(n) = \lfloor \left(\frac{n}{8 \log n} \right)^{2/5} \rfloor$ and $m(n) = l^2$. Let $CliqCol_n$ be the union of $Clique_{m,n}(\vec{p}, \vec{q})$ and $Color_{n,l}(\vec{p}, \vec{r})$.

The following facts follow immediately from Definition 7.1, and applications of Corollary 7 and Lemma 11.

Lemma 24 *Let n tend to infinity, with $k = k(n) \in \mathbb{N}$.*

1. *The number of variables in* $CliqCol_n$ *is* $\Theta(n^2)$*, the number of clauses in* $CliqCol_n$ *is* $O\left(n^2\left(n/\log n\right)^{8/5}\right)$*, and the negative width of* $CliqCol_n$ *is* 3*.*

2. *The number of variables in* $CliqCol_n^k$ *is* $\Theta(kn^2)$*, and the number of clauses in* $CliqCol_n^k$ *is* $O\left(k^3n^2\left(n/\log n\right)^{8/5}\right)$*.*

3. *The CNF* $perm(CliqCol_n^k)$ *contains* $\Theta(kn^2)$ *many variables, and* $perm(CliqCol_n^k)$ *contains* $O\left(k^5n^6\left(n/\log n\right)^{8/5}\right)$ *many clauses.*

7.1 Lower bound for OBDD refutations

Lemma 25 *Let* $n \in \mathbb{N}$ *be given and let* N *be the number of variables in* $CliqCol_n$*. If there is a size* S *OBDD refutation of perm* $\left(CliqCol_n^{4N^3}\right)$*, then there is a size* $S^{O(\log N)}$ *monotone Boolean circuit that separates* $Clique_{n,m}(\vec{p},\vec{q})$ *from* $Color_{n,l}(\vec{p},\vec{r})$*, where* $l = \lfloor\left(\frac{n}{8\log n}\right)^{2/5}\rfloor$ *and* $m = l^2$*.*

Proof: Because $\frac{2N^3}{4N^3} + \frac{N^2}{4N^4-1} < 1$, we may apply Lemma 14 and conclude that there is a size $\leq S$ tree-like OBDD refutation of $CliqCol_{l,m,n}$ with respect to an order that places all $q_{u,i}$'s before all $p_{i,j}$'s, and all $p_{i,j}$'s before all $r_{i,a}$'s. By Theorem 4 there is a monotone Boolean circuit F of size at most $S^{O(\log n)}$ that separates $Clique_{n,m}(\vec{p},\vec{q})$ from $Color_{n,l}(\vec{p},\vec{r})$. ∎

Theorem 26 *[27] There exists a constant* $c > 0$ *so that for all* l, m, n *satisfying* $3 \leq l < m$ *and* $m\sqrt{l} \leq \frac{n}{8\log n}$*, every monotone Boolean circuit that separates* $Clique_{n,m}(\vec{p},\vec{q})$ *from* $Color_{n,l}(\vec{p},\vec{r})$ *has size at least* $2^{c\sqrt{l}}$*.*

Combining Lemma 25 with Theorem 26 yields:

Lemma 27 *Let* $n \in \mathbb{N}$ *tend to infinity, and let* N *be the number of variables in* $CliqCol_n$*. All OBDD refutations of* $perm(CliqCol_n^{4N^3})$ *have size* $2^{N^{\Omega(1)}}$*.*

Proof: Let S be the size of smallest OBDD refutation of $perm(CliqCol_n^{4N^3})$. By Lemma 25, there is a monotone circuit of size at most $S^{O(\log N)}$ that separates $Clique_{n,m}(\vec{p},\vec{q})$ from $Color_{n,l}(\vec{p},\vec{r})$, where $l = \lfloor\left(\frac{n}{8\log n}\right)^{2/5}\rfloor$ and $m = l^2$. Because $m\sqrt{l} = l^{5/2} \leq \left(\left(\frac{n}{8\log n}\right)^{2/5}\right)^{5/2} = \frac{n}{8\log n}$ we may apply Theorem 26 and conclude that every monotone Boolean circuit that separates $Clique_{n,m}(\vec{p},\vec{q})$ from $Color_{n,l}(\vec{p},\vec{r})$ has size at least $2^{\Omega(\sqrt{l})} = 2^{\Omega(\sqrt[5]{n/\log n})}$. Therefore, $S = 2^{N^{\Omega(1)}}$. ∎

By Lemma 24, when $N = |Vars(CliqCol_n)|$, the number of variables in $perm(CliqCol_n^{4N^3})$ is polynomial in the number of variables in $ColGen_n$. Therefore, we have as a corollary:

Corollary 28 *Let* $n \in \mathbb{N}$ *tend to infinity, let* N *be the number of variables in* $CliqCol_n$*, let* M *be the number of variables in* $perm(CliqCol_n^{4N^3})$*. All OBDD refutations of* $perm(CliqCol_n^{4N^3})$ *have size* $2^{M^{\Omega(1)}}$*.*

7.2 Quasipolynomial size Res($O(\log n)$) refutations

Lemma 29 *Let $n \in \mathbb{N}$ tend to infinity, and let N be the number of variables in $CliqCol_n$. There is a Res($O(\log N)$) refutation of $CliqCol_n$ of size $N^{O(\log N)}$.*

Proof: First, we reduce the task of refuting $CliqCol_n$ to the task of refuting PHP_l^m, where $l = \lfloor \left(\frac{n}{8 \log n}\right)^{2/5} \rfloor$ and $m = l^2$. We simply define the $PHP_l^{l^2}$ variables using a 2-DNF substitution, as was done in [41, 9], and then apply Lemma 8 using the Res($O(\log l)$) refutations of $PHP_l^{l^2}$ demonstrated by Maciel, Pitassi and Woods [31].

Here is the reduction from $CliqCol_n$ to PHP_l^m: For $u \in [m]$, $v \in [l]$, define $P_{u,v} = \bigvee_{i=1}^{n} q_{u,i} r_{i,v}$. Note that for each $u \in [m]$, $\bigvee_{v \in [l]} P_{u,v} = \bigvee_{v \in [l]} \bigvee_{i=1}^{n} q_{u,i} r_{i,v}$ is a positive 2-DNF. Furthermore, each $\bigvee_{v \in [l]} P_{u,v}$ it can be derived from the $Clique_{m,n}$ clauses of type 1 and the $Color_{l,n}$ claues of type 1, using an $n^{O(1)}$ size Res(2) derivation. For $u, u' \in [m]$ with $u \neq u'$, and $v \in [l]$, $\neg P_{u,v} \vee \neg P_{u',v}$, can be expressed as a conjunction n^2 many clauses, each of negative width four: For each $i, j \in [n]$, $\neg q_{u,i} \vee \neg r_{i,v} \vee \neg q_{u',j} \vee \neg r_{j,v}$. These can be derived as from the clauses of $Clique_{m,n}$ of types 2 and 3 and the clauses of $Col_{l,n}$ of types 2 and 3, using a $n^{O(1)}$ size Res(2) derivation.

Maciel Pitassi and Woods have demonstrated that $PHP_l^{l^2}$ has a Res($O(\log l)$) refutation of size $l^{O(\log l)}$ [31]. Furthermore, a careful inspection of their proof reveals that it contains no negative literals in compound terms, and that every line contains at most $O(\log l)$ many negative literals (see Theorem 31 of the Appendix). Because the 2-DNFs $P_{u,v}$ are positive, and each contains at most n many terms, by Lemma 8, there is a Res($O(\log l)$) refutation of $PHP_l^{l^2}(P)$ of size at most $n^{O(\log l)}$.

Thus there is a Res($O(\log l)$) refutation of $CliqCol_n$ of size at most $n^{O(\log l)} = N^{O(\log N)}$.

■

Corollary 30 *Let $n \in \mathbb{N}$ tend to infinity, let N be the number of variables in $CliqCol_n$, and let M be the number of variables in $perm(CliqCol_n^{4N^3})$. There is a Res($O(\log M)$) refutation of $perm(CliqCol_n^{4N^3})$ of size $M^{O(\log M)}$.*

Let $n \in \mathbb{N}$ be given, let N be the number of variables in $CliqCol_n$, and let M be the number of variables in $perm(CliqCol_n^{4N^3})$. Corollary 28 shows that OBDD refutations require size $2^{M^{\Omega(1)}}$ to refute $perm(ColGen_d^{4N^3})$, and Corollary 23 shows that Res($O(\log M)$) can refute $perm(CliqCol_n^{4N^3})$ in size $M^{O(1)}$. Combining these two facts proves Theorem 2: OBDD refutations of M-variable CNFs cannot p-simulate Res($O(\log M)$).

References

[1] A. Aguirre and M. Vardi. Random 3-SAT and BDDs: The plot thickens further. In *Principles and Practice of Constraint Programming*, pages 121–136, 2001.

[2] N. Alon and J. Spencer. *The Probabilistic Method*. Interscience Series in Discrete Mathematics and Optimization. John Wiley, 2000.

[3] F. Aloul, M. Mneimneh, and K. Sakallah. ZBDD-based backtrack search SAT solver. In *Eleventh IEEE/ACM Workshop on Logic & Synthesis*, pages 131–136, 2002.

[4] A. Atserias, P. Kolaitis, and M. Vardi. Constraint propagation as a proof system. In *Tenth International Conference on Principles and Practice of Constraint Programming*, pages 77–91, 2004.

[5] P. Beame, R. Karp, T. Pitassi, and M. Saks. The efficiency of resolution and Davis–Putnam procedures. *SIAM Journal on Computing*, 31(4):1048–1075, August 2002. Preliminary versions in FOCS 1996 and STOC 1998.

[6] E. Ben-Sasson, R. Impagliazzo, and A. Wigderson. Near optimal separation of tree-like and general resolution. *Combinatorica*, 24(4):585–603, 2004.

[7] E. Ben-Sasson and A. Wigderson. Short proofs are narrow — resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.

[8] M. L. Bonet, J. L. Esteban, N. Galesi, and J. Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM Journal on Computation*, 30(5):1462–1484, 2000.

[9] M. L. Bonet, T. Pitassi, and R. Raz. Lower bounds for cutting planes proofs with small coefficients. *Journal of Symbolic Logic*, 62(3):708–728, 1997.

[10] R. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, 1986.

[11] R. Bryant. Symbolic boolean manipulation with ordered binary decision diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.

[12] P. Chatalic and L. Simon. Multi-resolution on compressed sets of clauses. In *Proceedings of the Twelfth International Conference on Tools with Artificial Intelligence*, pages 2–10, 2000.

[13] P. Chatalic and L. Simon. Zres: The old Davis-Putnam procedures meets ZBDDs. In *Proceedings of the Seventeenth International Conference on Automated Deduction*, pages 449–454, 2000.

[14] V. Chvátal and E. Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

[15] C. Coarfa, D. Demopoulos, A. S. M. Aguirre, D. Subramanian, and M. Vardi. Random 3-SAT: The plot thickens. *Constraints*, 8(3):243–261, 2003.

[16] S. Cook and A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.

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

[18] E. Goldberg and Y. Novikov. Berkmin: a fast and robust SAT solver. In *Proceedings of DATE 2002*, 2002.

[19] J. F. Groote. Hiding propositional constants in BDDs. *Formal Methods in System Design: an International Journal*, 8(1):91–96, 1996.

[20] J. F. Groote and H. Zantema. Resolution and binary decision diagrams cannot simulate each other polynomially. *Discrete Applied Mathematics*, 130(2), 2003.

[21] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

[22] J. Huang and A. Darwiche. Toward good elimination ordering for symbolic SAT solving. In *Proceedings of the Sixteenth IEEE Conference on Tools with Artificial Intelligence*, pages 566–573, 2004.

[23] R. Impagliazzo, T. Pitassi, and A. Urquhart. Upper and lower bounds for tree-like cutting planes proofs. In *Proceedings Ninth Annual Symposium on Logic in Computer Science*, pages 220–228, 1994.

[24] T. Jussila, C. Sinz, and A. Biere. Extended resolution proofs for symbolic SAT solving with quantification. In *Proceedings of the Ninth International Conference on Theory and Applications of Satisfiability Testing*, pages 54–60, 2006.

[25] J. Krajíček. Lower bounds to the size of constant-depth propositional proofs. *Journal of Symbolic Logic*, 59(1):73–86, 1994.

[26] J. Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory.* Cambridge University Press, 1995.

[27] J. Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *Journal of Symbolic Logic*, 62(2):457–486, 1997.

[28] J. Krajíček. On the weak pigeonhole principle. *Fudamenta Mathematicae*, 170:123–140, 2001.

[29] J. Krajíček. An exponential lower bound for a constraint propagation proof system based on ordered binary decision diagrams. Technical Report 7, Electronic Colloquium on Computational Complexity, 2007. To appear in the Journal of Symbolic Logic.

[30] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[31] A. Maciel, T. Pitassi, and A. Woods. A new proof of the weak pigeonhole principle. *Journal of Computer and System Sciences*, 64(3):843–872, 2002.

[32] J. Marques-Silva and K. Sakallah. GRASP a new search algorithm for satisfiability. In *Proceedings of IEEE/ACM Internation Conference on Computer-Aided Design*, 1996.

[33] C. Meinel and T. Theobald. *Algorithms and Data Structures in VLSI Design.* Springer-Verlag, 1998.

[34] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of 2001 Design Automation Conference*, 2001.

[35] D. Motter and I. Markov. A compressed breadth-first search for satisfiability. In *Fourth International Workshop on Algorithms Engineering and Experiments (ALENEX)*, pages 29–42, 2002.

[36] D. Motter and I. Markov. Overcoming resolution-based lower bounds for SAT solvers. In *Eleventh IEEE/ACM Workshop on Logic and Synthesis*, pages 373–378, 2002.

[37] D. Motter, J. Roy, and I. Markov. Resolution cannot polynomially simulate compressed-BFS. *Annals of Mathematics and Artificial Intelligence*, 44(1–2):121–156, 2005.

[38] G. Pan and M. Vardi. Search vs. symbolic techniques in satisfiability solving. In *The Seventh International Conference on Theory and Applications of Satisfiability Testing*, 2004.

[39] R. Raz. Resolution lower bounds for the weak pigeonhole principle. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 553–562, 2002.

[40] R. Raz and P. McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999.

[41] A. Razborov. Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic. *Izvestiya of the Russian Academy of Science, Mathematics*, 59(1):201–224, 1995.

[42] J. Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, January 1965.

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

[44] C. Sinz and A. Biere. Extended resolution proofs for conjoining BDDs. In *First International Computer Science Symposium in Russia*, pages 600–611, 2006.

[45] T. Uribe and M. Stickel. Ordered binary decision diagrams and the Davis-Putnam procedure. In *Proceedings of the First International Conference on Constraints in Computational Logics*, pages 34–49, 1994.

[46] A. Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, 1987.

[47] M. N. Wegman and J. L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22(3):265–279, 1981.

A Blowing up Res(k) refutations

Proof:(of Lemma 8) Let Γ be a Res(k_0) refutation of $\mathcal{F}$ that has size at most S, negative width at most w, and contains no negated literals in compound terms. We simply apply the transformation that replaces each k_0-DNF F of Γ by the (k_0k_1)-DNFs of $\Sigma_D(F)$ and then patch things up to preserve the inferences. By Corollary 6, the initial translation that replaces each F by $\Sigma_D(F)$ increases the number of lines from S to at most m^wS. In the analysis below, we show that each formula in the translation needs at most $2m^{k_0}$ many new lines to patch up its inference, thus the net size of the derivation is at most $2m^{w+k_0}S$.

Axioms: Every k_0-DNF $F \in \mathcal{F}$ is replaced by the (k_0k_1)-DNFs of $\Sigma_D(F)$, so the hypotheses of the new derivation are indeed the DNFs of $\Sigma_D(\mathcal{F})$.

Subsumption inferences: From F with $F \subseteq G$, infer G. The characterization of Lemma 5 guarantees that in this case, every $G' \in \Sigma_D(G)$ is subsumed by some $F' \in \Sigma_D(F)$.

AND-introduction inferences : From $F \vee \bigwedge_{i\in I_1} x_i, F \vee \bigwedge_{i\in I_2} x_i, \ldots F \vee \bigwedge_{i\in I_j} x_i$, infer $F \vee \left(\bigwedge_{i\in I} x_i\right)$, where $I = I_1 \cup \cdots \cup I_j$ and $|I| \leq k$. Note that the literals of the conjunction are all

positive variables because Γ contains no negations inside compound literals. By Lemma 5, we have that:

$$\Sigma_D\left(F \vee \left(\bigwedge_{i\in I} x_i\right)\right) = \{F' \vee \bigvee_{T\in\prod_{i\in I} D_i} \left(\bigwedge_{i\in I} T_i\right) \mid F' \in \Sigma_D(F)\}$$

On the other hand, for each $l = 1, \ldots j$:

$$\Sigma_D\left(F \vee \left(\bigwedge_{i\in I_l} x_i\right)\right) = \{F' \vee \bigvee_{T\in\prod_{i\in I_l} D_i} \left(\bigwedge_{i\in I} T_i\right) \mid F' \in \Sigma_D(F)\}$$

Therefore, we can construct every (k_0k_1)-DNF of $\Sigma_D\left(F \vee \left(\bigwedge_{i=1}^{k} x_i\right)\right)$ by using a tree of at most $\sum_{j=0}^{k} m^j < 2m^k \leq 2m^{k_0}$ many AND-introduction inferences.

AND-elimination inferences: From $F \vee \left(\bigwedge_{i\in I} x_i\right)$ infer $F \vee \bigvee_{i\in J} x_i$, where $J \subseteq I$ and $|I| \leq k_0$. Because Γ contains no negations inside compound terms, we may assume that each x_i is positive.

$$\Sigma_D\left(F \vee \left(\bigwedge_{i\in I} x_i\right)\right) = \{F' \vee \bigvee_{T\in\prod_{i=1}^{k} D_i} \left(\bigwedge_{i\in I} T_i\right) \mid F' \in \Sigma_D(F)\}$$

On the other hand:

$$\Sigma_D\left(F \vee \left(\bigwedge_{i\in J} x_i\right)\right) = \{F' \vee \bigvee_{T\in\prod_{i\in J} D_i} \left(\bigwedge_{i\in J} T_i\right) \mid F' \in \Sigma_D(F)\}$$

Therefore, every DNF of $\Sigma_D(F \vee \bigvee_{i\in J} x_i)$ can be inferred from some DNF of $\Sigma_D\left(F \vee \left(\bigwedge_{i\in I} x_i\right)\right)$ by at most $m^{|I|} \leq m^{k_0}$ applications of AND-elimination.

Cut inferences: From $F \vee x_i$ and $G \vee \neg x_i$, infer $F \vee G$. Every DNF of $\Sigma_D(F \vee G)$ is of the form $F' \vee G'$ where $F' \in \Sigma_D(F)$ and $G' \in \Sigma_D(G)$. On the other hand, for each $F' \in \Sigma_D(F)$, $F' \vee D_i \in \Sigma_D(F \vee x_i)$, and for each DNF $G' \in \Sigma_D(G)$, for each $T \in D_i$, the clause $G' \vee \bigvee_{l\in T} \neg l$ belongs to $\Sigma_D(G \vee \neg x_i)$. We apply cut-inferences with these at most m DNFs with $F' \vee D_i$, and we we obtain $F' \vee G'$.

■

B A resolution refutation of the $ColGen$ principle

Proof:(of Theorem 21, as per [8]) For each $c \in [n]$, $j \in [d]$, resolve the clauses of the form $\neg q_{d,j,c} \vee p_{1,1,c}$ and $\neg p_{1,1,c} \vee \neg r_c$ to obtain $\neg q_{d,j,c} \vee \neg r_c$. Clearly these steps have negative width two.

Now derive $\neg q_{i,j,c} \vee \neg r_c$ for every $(i,j) \in Pyr_d$ and $c \in [n]$. This is done downwards for d to 1, the starting point is just $\neg q_{d,j,c} \vee \neg r_c$, which we derived previously. After stage $i+1$, we have derived $\neg q_{i+1,j,a} \vee \neg r_a$ and $\neg q_{i+1,j+1,b} \vee \neg r_b$, for each $a, b \in [n]$. For each $c \in [n]$, esolve these against $r_a \vee r_b \vee \neg p_{a,b,c} \vee \neg r_c$ to obtain $\neg q_{i+1,j,a} \vee \neg q_{i+1,j+1,b} \vee \neg p_{a,b,c} \vee \neg r_c$, and resolve these new clauses against $\neg q_{i+1,j,a,} \vee \neg q_{i+1,j+1,b} \vee \neg q_{i,j,c} \vee p_{a,b,c}$ to obtain $\neg q_{i+1,j,a} \vee \neg q_{i+1,j+1,b} \vee \neg q_{i,j,c} \vee \neg r_c$.

For each $a, b \in [n]$, we resolve $\neg q_{i+1,j,a} \vee \neg q_{i+1,j+1,b} \vee \neg q_{i,j,c} \vee \neg r_c$ against $\bigvee_{a\in[n]} q_{i+1,j,a}$, to obtain $\neg q_{i+1,j+1,b} \vee \neg q_{i,j,c} \vee \neg r_c$. Then, for reach $b \in [n]$, we resolve $\neg q_{i+1,j+1,b} \vee \neg q_{i,j,c} \vee \neg r_c$ against $\bigvee_{b\in[n]} q_{i+1,j+1,a}$, to obtain $\neg q_{i,j,c} \vee \neg r_c$. Notice that this phase has negative width a most 4.

Now, we have $\neg q_{1,1,a} \vee \neg r_a$ for all $a \in [n]$. For each $a \in [n]$, resolve these against $\neg p_{a,a,n} \vee r_a$ to obtain $\neg q_{1,1,a} \vee \neg p_{a,a,n}$. Resolve these against $\neg q_{1,1,a} \vee p_{a,a,n}$ to obtain $\neg q_{1,1,a}$. Now resolve these against $\bigvee_{a\in[n]} q_{1,1,a}$ to obtain the empty clause. Clearly this final phase has negative width at most two.

This refutation clearly has size $n^{O(1)}$. ∎

C Refuting $PHP_l^{l^2}$

Theorem 31 *[31] There exists $c > 0$, such that for every $n > 1$, there is a Res($\lceil \log n \rceil + 1$) refutation of $PHP_n^{n^2}$ of size $n^{c \log n}$, such that no negated literal appears in a compound term, and the negative width of every formula is at most $2 \log n$.*

Proof: We just carry out the proof from [31] and keep track of the negative literals.

Let $n > 1$ be given. We show by induction on $l = 0, \ldots \lceil \log n \rceil$, for every k, and every family $\langle Q_{i,j} \mid i \in [n^2],\ j \in [2^l] \rangle$ of positive k-DNFS, the set of k-DNFs $PHP_n^{n^2} \cup PHP_{2^l}^{n^2}(Q)$ has a Res$(l + k)$ refutation of size $n^{O(k+l)}$ with no negative literals in compound terms, and negative width at most $2l + 2k$. Specializing to the case of $k = 1$ and $l = \lceil \log n \rceil$ proves Theorem 31.

For $l = 0$, note that $PHP_1^{n^2}$ has a resolution refutation of five lines with negative width two. It immediately follows that for any positive k-DNFs $\langle Q_{i,j} \mid i \in [n^2],\ j \in [2^l] \rangle$, $PHP_1^{n^2}$ has a Res(k) refutation of size $n^{O(k)}$ with no negative literals in compound terms and negative width at most $2k$.

Now, let $l < \log n$ be given, and assume that for every k, and every family of positive k-DNFs $\langle Q_{i,j} \mid i \in [n^2],\ j \in [2^l] \rangle$, $PHP_n^{n^2} \cup PHP_{2^l}^{n^2}(Q)$ has a Res$(l + k)$ refutation of size $n^{O(k+l)}$ with no negative literals in compound terms, and negative width at most $2l + 2k$.

Let k be given and let $\langle Q_{i,j} \mid i \in [n^2],\ j \in [2^{l+1}] \rangle$ be a family of positive k-DNFs. Partition $[n^2]$ into n many sets of size n, $A_1, \ldots A_m$. Partition 2^{l+1} into two sets of size 2^l, B_0 and B_1.

Consider some $m \in [n]$. For each $i \in A_m$, the formula $\bigvee_{j\in[2^{l+1}]} Q_{i,j}$ can be written as $\bigvee_{j\in B_0} Q_{i,j} \vee \bigvee_{j\in B_1} Q_{i,j}$. For each $i \in [n^2]$, $j \in B_0$, let let $R_{i,j}$ be the $(k+1)$-DNF $\bigvee_{t=1}^{n} x_{i,t} Q_{t,j}$. Using AND-introduction inferences and subsumption inferences, we derive $\bigvee_{j\in B_0} R_{i,j}$ for each $i \in [n^2]$, $j \in B_0$. Furthermore, the set of clauses equivalent to $\neg R_{i_0,j} \vee \neg R_{i_1,j}$, for $i_0, i_1 \in [n^2]$, with $i_0 \neq i_1$, and $j \in B_0$, either follows by subsuming some $\neg Q_{t_0,j} \vee \neg Q_{t_1,j}$ (with $t_0, t_1 \in [n]$, $t_0 \neq t_1$) or by subsuming some $\neg x_{i_0,t} \vee \neg x_{i_1,t}$ with $t \in [n]$. Therefore, we may apply the induction hypothesis to refute $PHP_n^{n^2} \cup PHP_{2^l}^{n^2}(Q)$ has a Res$(l + k + 1)$ refutation of size $n^{O(k+1)}$ with no negative literals in compound terms, and negative width at most $2l+2k+2$. By dragging along the side formulas, we derive $\bigvee_{i\in A_m} \bigvee_{j\in B_1} Q_{i,j}$, using a Res$(l + k + 1)$ derivation of size $n^{O(k+1)}$ with no negative literals in compound terms, and negative width at most $2l + 2k + 2$.

Perform the derivation of the preceding paragraph for each $m \in [n]$. This derives $\bigvee_{v\in B_1} \bigvee_{u\in A_m} Q_{u,v}$ for each $m \in [n]$. Thus, we have derived $PHP_{B_1}^{n}(\langle \bigvee_{u\in A_m} Q_{u,v} \mid m \in [n] \rangle)$.

For each $i \in [n^2]$, $j \in B_1$, let $R_{i,j}$ be the $(k + 1)$-DNF $\bigvee_{m=1}^{n} x_{i,m} \left(\bigvee_{u\in A_m} Q_{u,j} \right)$. We derive $PHP_{B_1}^{n^2}(\langle R_{i,j} \mid i \in [n^2],\ j \in [n] \rangle)$ from $PHP_n^{n^2} \cup PHP_{B_1}^{n}(\langle \bigvee_{u\in A_m} Q_{u,v} \mid m \in [n] \rangle)$ as follows: The formulas $\bigvee_{j\in[n]} R_{i,j}$ for each $i \in [n^2]$ follow by applying AND-introduction and subsumption inferences to the formulas $\bigvee_{t=1}^{n} x_{i,t}$ and $\bigvee_{j\in B_1} \bigvee_{u\in A_m} Q_{u,j}$. Each clauses in the deMorgan expansion

of $\neg R_{i_0,j} \vee \neg R_{i_1,j}$ either follows by subsuming some $\neg Q_{t_0,j} \vee \neg Q_{t_1,j}$ (with $t_0, t_1 \in [n]$, $t_0 \neq t_1$) or by subsuming some $\neg x_{i_0,t} \vee \neg x_{i_1,t}$ with $t \in [n]$.

Therefore, we can apply the induction hypothesis to $PHP_n^{n^2} \cup PHP_{B_1}^{n^2}(\langle R_{i,j} \mid i \in [n^2],\ j \in [n]\rangle)$ and we have a $\mathrm{Res}(k+1+l)$ of size $n^{O(k+1)}$ with no negated literals in compound terms, and of negative width at most $k+l+1$. ∎

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/