Automating OBDD proofs is NP-hard

Dmitry Itsykson
Steklov Institute of Mathematics at St. Petersburg, Russia

Artur Riazanov
Steklov Institute of Mathematics at St. Petersburg, Russia

Abstract

We prove that the proof system OBDD($\wedge$, weakening) is not automatable unless P = NP. The proof is based upon the celebrated result of Atserias and Müller [5] about the hardness of automatability for resolution. The heart of the proof is lifting with multi-output indexing gadget from resolution block-width to dag-like multiparty number-in-hand communication protocol size with $o(n)$ parties, where n is the number of variables in the non-lifted formula. A similar lifting theorem for protocols with $n + 1$ participants was proved by Göös et. el. [12] to establish the hardness of automatability result for Cutting Planes.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases proof complexity, OBDD, automatability, lifting, dag-like communication

Digital Object Identifier 10.4230/LIPIcs.MFCS.2022.59

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2022/046/`

Funding The research is supported by Russian Science Foundation (project 18-71-10042).

Acknowledgements The authors thank Anastasia Sofronova and Michal Garlı́k for fruitful discussions, Anastasia Sofronova for her comments on the draft. They also thank anonymous referees for their feedback. Artur Riazanov additionally thanks Dmitry Sokolov for his patient explanations of the lifting technique in [10].

1 Introduction

Boolean satisfiability is one of the central problems in Computer Science. The input to this problem is a CNF formula and the goal is to determine whether it is satisfiable or not. This is a standard example of an NP-complete problem, and it has been very thoroughly studied. While the consensus is that there is no polynomial algorithm for satisfiability, contemporary SAT-solvers have been quite successful in solving it for many instances appearing “in practice”.

SAT-solvers are tightly connected to proof complexity. A propositional proof system is a formal way of certifying that a CNF formula is unsatisfiable. The execution log of an SAT-solver running on an unsatisfiable input φ can serve as a certificate of unsatisfiability of φ. Then SAT-solvers face the following trade-off: on the one hand, their underlying proof system must be sufficiently strong to have short proofs of all formulas of interest, on the other hand, it must be sufficiently weak so short proofs can be found fast. This tradeoff is witnessed by the success of CDCL-solvers, which are based on (subsystems of) Resolution which is a pretty weak proof system. Nevertheless, so far SAT-solvers based on stronger proof systems have not enjoyed the widespread success of resolution-based solvers.

A propositional proof system Π is called automatable (quasi-automatable) if there exists an algorithm $\mathcal{E}$ that given an unsatisfiable CNF φ produces a Π-proof of φ in time polynomial (quasi-polynomial) in size of φ plus the size of the shortest Π-proof of φ.

However, for many non-trivial proof systems, there are known pieces of evidence that they likely are not automatable or quasi-automatable. A long line of results on resolution automatability [15, 19, 2, 3] is concluded with the recent breakthrough result by Atserias and Müller [5] stating that resolution is not automatable unless P = NP and not quasi-automatable under a stronger assumption. This result sparked a series of follow-up results that establish the hardness of automating for many other proof systems; these results are either based on Atserias-Müller's result directly or follow their plan closely. If P $\neq$ NP, then the following proof systems are not automatable: Nullstellensatz and Polynomial Calculus [13]; Cutting Planes [12]; Res(2) [11]. Under stronger assumptions one can show non-automatability of Frege systems [17, 7, 6].

We continue this line of research and study the automatability of OBDD-based systems. OBDD (or ordered binary decision diagram) is a simple but rather expressive way to represent Boolean functions introduced by Bryant [8]. An OBDD is a very restricted case of a branching program, wherein for all paths from the source to a sink, variables appear in the same order. It is known that branching programs are at least as powerful as Boolean formulas, hence proving superpolynomial lower bounds on size of branching programs for explicit functions is an extremely hard problem. But exponential lower bounds are known for restricted versions of branching programs, including OBDDs. However, this restriction allows performing many important operations with OBDDs very efficiently: testing satisfiability, computing binary operations, applying restrictions, minimization, and so on. These properties have paved the way for OBDD-based propositional proof systems introduced by Atserias, Kolaites, and Vardi [4] to serve as a base for OBDD based SAT-solvers [1, 20].

An OBDD($\wedge$, weakening) refutation of a CNF φ is a sequence of OBDDs that query variables in the same order; the last OBDD in the sequence is identically false and each of those diagrams either represents a clause of φ or follows semantically from two OBDDs that appear earlier in the sequence (formally there are two rules: by the first ($\wedge$) we can derive conjunction of two OBDDs and by the second (weakening) we can derive any semantic implication of a single OBDD). The correctness of application of these rules can be efficiently verified since binary operations for two OBDDs with the same order of variab can be computed in polynomial time. This system simulates Resolution and CP* (Cutting Planes with unary coefficients); it has short refutations of unsatisfiable linear systems over $\mathbb{F}_2$ [4] and clique-coloring tautologies [9] (the latter are hard for Cutting Planes [22]).

Atserias-Müller's approach for establishing hardness of automatability requires proving a lower bound on the proof size of some specific CNF-formula. Unfortunately the tools for proving lower bounds on OBDD($\wedge$, weakening) are quite limited and related to monotone circuit complexity. All known lower bound proofs consist of two steps.

1. To prove the lower bound for a fixed order of variables in OBDDs. Such a lower bound was proved by Atserias et. al. [4]; an exponential lower bound on the size of OBDD($\wedge$, weakening) refutations of clique-coloring tautologies with a particular order of variables follows from monotone interpolation.
2. To transform a formula that is hard for one order into a formula that is hard for all orders. The first transformation of this kind was devised by Krajićek [16]: formulas are equipped with additional variables that parameterize a permutation of main variables such that by fixing these additional variables we can get the initial formula, where variables are permuted by any desired permutation. Segerlind [23, 24] invented a more concise transformation using 2-independent permutation family together with orification of variables; Segerlind used it to prove that OBDD($\wedge$, weakening) may require exponentially longer proofs than Res($O(\log n)$).

Our contribution. Our main result is the following theorem:

▶ **Theorem 1.** *There exist a constant α and a polynomially computable function $\mathcal{R}$ mapping CNF formulas to CNF formulas with the following properties. For any 3-CNF φ with n variables such that: if φ is satisfiable, then $\mathcal{R}(\varphi)$ has a resolution refutation of size at most n^α; if φ is unsatisfiable, then any* OBDD($\wedge$, *weakening*) *refutation of $\mathcal{R}(\varphi)$ has size $2^{\Omega(n)}$.*

Since $\mathrm{OBDD}(\wedge, \mathrm{weakening})$ simulates resolution, any automation algorithm for $\mathrm{OBDD}(\wedge, \mathrm{weakening})$ can be used to solve 3-SAT: if it finds proofs in fixed polynomial time, then the input formula is satisfiable, otherwise, it is unsatisfiable.

Our technique can be applied to other proof systems as well since the only thing that we use about OBDDs is that the value of an OBDD of size S can be computed using $O(\ell \log S)$ bits of communication in the ℓ-party number-in-hand communication model if the partition of variables agrees with the order. For example, this property holds for k-OBDDs for small k, hence our technique can be applied for proof system k-OBDD($\wedge$, weakening) [14].

Technique. The proof consists of two parts:

1. Prove the weaker version of Theorem 1, where the lower bound holds only for refutations that consist of OBDDs in some particular order π.
2. Devise a polynomial-time algorithm that transforms formulas with short resolution refutations to formulas with short resolution refutations; and transforms formulas that are hard for $\mathrm{OBDD}(\wedge, \mathrm{weakening})$ with a specific order to formulas that are hard for $\mathrm{OBDD}(\wedge, \mathrm{weakening})$ for all orders.

To implement the second part we use Segerlind's transformation. It almost suits our case, but the property for resolution works only with an additional condition: if a formula has a short resolution proof with at most constant number negative literals in every clause (we say that the *negative width* of the proof is $O(1)$), then the result of Segerlind's transformation has a short resolution proof.

The first part is much more involved. The construction is built on the following result proved by Atserias and Müller [5]. There exists an algorithm $\mathcal{E}$ that given a 3-CNF formula φ produces in polynomial time another CNF formula $\mathcal{E}(\varphi)$ such that

- if φ is satisfiable, $\mathcal{E}(\varphi)$ admits a polynomial-size resolution refutation;
- if φ is unsatisfiable, the shortest refutation of $\mathcal{E}(\varphi)$ has size $2^{|\varphi|^{\Omega(1)}}$.

We get our result by applying lifting to $\mathcal{E}(\varphi)$. Lifting is a technique to obtain lower bounds for strong computational models from lower bounds for weaker models. Recently, Garg, et. al. [10] proved two similar lifting theorems lifting from resolution width to refutation size in (1) any semantic proof system operating with proof lines of small 2-party communication complexity and (2) cutting planes (precisely it works for proof systems, where proof lines can be computed by 1-round real communication protocol).

The first lifting theorem (applied to $\mathcal{E}(\varphi)$) seems enticing for us since a function computable by an OBDD can be computed with small 2-party communication. Unfortunately, we can not apply this theorem directly since $\mathcal{E}(\varphi)$ can have large resolution width even for a satisfiable φ so after the application of lifting the resulting CNF might have only exponential-size $\mathrm{OBDD}(\wedge, \mathrm{weakening})$ refutations. Göös et. al. [12] face the same problem for Cutting Planes and deal with it by lifting from block-width instead of the plain width. However the lifting theorem in [10] does not work for block-width, so Göös et. al. [12] prove a weaker version of it: they lift from resolution block-width to k-dimensional simplex-dags, where k is the number of variables in the lifted formula plus one. Cutting planes refutations can be converted to k-dimensional simplex-dags of the same size. However, for OBDD($\wedge$, weakening) refutations, the size is raised to the power of k, hence we need a lifting theorem for a smaller value of k.

We prove another lifting theorem: we lift from resolution block-width to k-dimensional box dag size, where k is the size of the largest block in the partition w.r.t. which the block-width is computed, plus one. In our proof, we use the structural properties of rectangles from [10] and extend them to show the structural properties of boxes. The same theorem seems to hold for simplex-dags (the proof in [12] can be adapted as well), but it is not clear whether there exist context where such change in the dimension matters.

We also show that OBDD($\wedge$, weakening) refutations with a specific order of variables of size S can be converted to k-dimensional box dags of size $S^{O(k)}$. In actuality, we prove it for every proof system that operates with proof lines that can be computed by k party communication protocols in the number-in-hand model with a small cost.

2 Preliminaries

Notation. We use the standard notation $[n] = \{1, \dots, n\}$. $\mathrm{Vars}(\varphi)$ denotes the set of propositional variables of a formula φ. We refer to a uniform distribution over a set X by $\mathcal{U}(X)$.

Resolution. A resolution refutation of an unsatisfiable CNF φ is a sequence of clauses ending with the empty clause such that each clause of the sequence is either a clause of φ or is derived from the previous clauses in the sequence with a resolution rule: $\frac{A \vee x \quad B \vee \neg x}{A \vee B}$.

The *width* of a clause is the number of literals in it, and the width of a formula is the maximum width of a clause in it. The *size* of a resolution refutation is the number of clauses in it. The width of a resolution refutation is the largest width of a clause in it.

Let X be a set of propositional variables and $U = U_1, \dots, U_k$ be a partition of X. Let us define the *block-width* of a clause C over variables X as the number of blocks among $U_1, \dots, U_k$ that contain variables of C: $|\{i \in [k] \mid \mathrm{Vars}(C) \cap U_i \neq \emptyset\}|$. The block-width of a resolution refutation is the maximum block-width of a clause in it. For an unsatisfiable CNF φ we denote $\mathrm{bw}(\varphi)$ as the smallest block-width of a resolution refutation of φ.

Ordered Binary Decision Diagrams. A branching program (BP) is a directed acyclic graph with a single source and two sinks: 0-sink and 1-sink. Each of the nodes of the BP except the sinks is labeled with a variable x_i for $i \in [n]$ and has two outgoing edges, one labeled with 1 and another with 0. Let us define the function computed by a BP. For a node u in a BP let $f_u : \{0,1\}^n \to \{0,1\}$ be a function computed by it. We then define $f_{\text{0-sink}} \equiv 0$, $f_{\text{1-sink}} \equiv 1$, $f_u(x) := f_v(x)$ if $x_i = 0$ and $f_u(x) := f_w(x)$ if $x_i = 1$ where u is labeled with the variable x_i, v is 0-successor of u and w is the 1-successor of u. Then we define the function computed by the BP itself as the function computed by its source.

A π-OBDD where $\pi \in S_n$ is a BP computing a function $f : \{0,1\}^n \to \{0,1\}$ such that for any path from the source to a sink each of the node labels appears at most once and the order of the labels appearing in the path respects π. That is, the labels appearing on the path always have form $x_{\pi(i_1)}, x_{\pi(i_2)}, \dots, x_{\pi(i_k)}$ where $1 \leq i_1 < i_2 < \dots < i_k \leq n$.

OBDDs have the following nice property: for every order of variables every Boolean function has a unique minimal OBDD. For a given order π, the minimal π-OBDD of a function f may be constructed in polynomial time from any π-OBDD for the same function [18]. There are also known polynomial-time algorithms that efficiently perform all the Boolean binary operations, negation and projection (elimination of the existential quantifier) to π-OBDDs [18] (we refer to [26] for an introduction to OBDDs).

OBDD refutations. A π-OBDD-refutation of a CNF formula φ is a sequence of π-OBDDs $D_1, \dots, D_s$ such that D_s computes the identically false function and each D_i either computes a clause of φ or is obtained from the previous diagrams in the sequence by one of the rules below.

conjunction rule ($\wedge$) D_i computes the conjunction of D_j and D_k for $j, k < i$;

weakening rule D_i computes a function implied by D_j where $j < i$;

projection rule ($\exists$) D_i computes a function $\exists x f$ where f is computed by D_j with $j < i$, and $x \in \mathrm{Vars}(\varphi)$.

The size of an π-OBDD-refutation is the sum of sizes of all diagrams in it. Using the properties of OBDD it is easy to see that the correctness of a π-OBDD-refutation can be verified in time polynomial in its size and the size of the refuted formula [4]. An OBDD refutation is a π-OBDD refutation for some order π.

Depending on the set of the allowed rules we have several different propositional proof systems: $\mathrm{OBDD}(\wedge)$ where only the conjunction rule is allowed, $\mathrm{OBDD}(\wedge, \exists)$ where the conjunction and the projection rules are allowed, and $\mathrm{OBDD}(\wedge, \text{weakening})$ where the conjunction and the weakening rules are allowed. Since the projection rule is a special case of the weakening rule, we do not include both of them simultaneously.

For an unsatisfiable CNF φ we denote by $\pi\text{-}\mathrm{OBDD}(\varphi)$ the size of the smallest π-$\mathrm{OBDD}(\wedge, \text{weakening})$ refutation of φ and by $\mathrm{OBDD}(\varphi)$ the size of the smallest $\mathrm{OBDD}(\wedge, \text{weakening})$ refutation of φ.

▶ **Proposition 2** ([4]). $\mathrm{OBDD}(\wedge, \exists)$ *(and, thus,* $\mathrm{OBDD}(\wedge, \text{weakening})$*) polynomially simulates resolution: if an unsatisfiable CNF has a resolution refutation of size S, then it has an* $\mathrm{OBDD}(\wedge, \exists)$ *refutation of size* $\mathrm{poly}(S)$*.*

Search$_\varphi$. Search_φ is the following search problem: given an assignment to the variables of the unsatisfiable CNF φ, find a clause that is falsified by this assignment. Formally it can be defined as a relation $\{(x, C) \mid x \in \{0,1\}^{\mathrm{Vars}(\varphi)};\ C \in \varphi;\ C(x) = 0\}$.

Dags solving relations.

▶ **Definition 3** ([25]). *Let $\mathcal{F}$ be a family of subsets of a finite set $\mathcal{X}$ and $S \subseteq \mathcal{X} \times \mathcal{O}$ be a relation. Let $\mathcal{D}$ be a single-source (which we refer to as root) acyclic graph. We call $\mathcal{D}$ an* $\mathcal{F}$-dag *solving S if for every node u there exists a set $R_u \in \mathcal{F}$ such that:*

(root condition) *for the root r of the dag $R_r = \mathcal{X}$;*

(leaf condition) *for each leaf (sink) ℓ of the dag there exists $o \in \mathcal{O}$ such that for all $x \in R_\ell$, $(x, o) \in S$;*

(local condition) *each inner node u has out-degree 2 and its two descendants v and w satisfy the property $R_u \subseteq R_v \cup R_w$.*

The size of an $\mathcal{F}$-dag *is the number of nodes in it. We denote the smallest size of* $\mathcal{F}$-dag *solving S by* $\mathcal{F}$-dag(S)*. We usually identify the nodes of an* $\mathcal{F}$-dag *with the sets R_u.*

Now we define several special cases of this general definition.

Decision dag. Assume that we have Boolean domain $\mathcal{X} = \{0,1\}^n$ that we view as a set of values of n propositional variables. A partial assignment is an element of $\{0,1,*\}^n$, where $*$ means that the corresponding variable is not assigned. Let $\mathrm{fix}(\rho) = \rho^{-1}(\{0,1\})$ be the set of assigned variables. If $\mathrm{fix}(\rho) = [n]$ then ρ is a full assignment.

Any partial assignment defines a subcube $\mathrm{Cube}(\rho) = \{\alpha \in \{0,1\}^n \mid \forall i \in \mathrm{fix}(\rho) : \rho(i) = \alpha(i)\}$ that is the set of all full assignments agreeing with ρ.

Let $S \subseteq \{0,1\}^n \times \mathcal{O}$ be a relation and $\mathcal{F}$ be a set of all subcubes $\{\mathrm{Cube}(\rho) \mid \rho \in \{0,1,*\}^n\}$, then we call an $\mathcal{F}$-dag for S a decision dag. We denote the smallest size of a decision dag solving S by $\mathsf{dec\text{-}dag}(S)$.

Observe that a decision tree is a decision dag: a node u of a decision tree can be labeled with a set $\mathrm{Cube}(\rho)$, where ρ is a partial assignment corresponding to the path from the root to u. Hence, since for any total relation there exists a decision tree solving it, any total relation has a decision dag as well.

Let $U = U_1, \ldots, U_k$ be a partition of $[n]$. The block-width of a decision dag is defined as follows: for a node labeled with $\mathrm{Cube}(\rho)$ we compute $|\{i \in [k] \mid U_i \cap \mathrm{fix}(\rho) \neq \emptyset\}|$, the blockwidth of a decision dag is the maximum of this value among the nodes. For a relation S we denote the smallest block-width of a decision dag that solves it as $\mathrm{bw}(S)$.

Observe that a resolution refutation of an unsatisfiable CNF φ can be converted to a decision dag solving Search_φ of the same size: the topology of the dag is the topology of the resolution refutation, a node corresponding to a clause C is labeled with a set $C^{-1}(0) = \{x \in \{0,1\}^n \mid C(x) = 0\}$. It is easy to see that this set is a subcube. If C is derived from D and E via a resolution rule then C is implied by the conjunction of D and E thus $C^{-1}(0) \subseteq (D \wedge E)^{-1}(0) = D^{-1}(0) \cup E^{-1}(0)$. Clearly the root and the leaf properties of the constructed decision dag also hold: for a leaf ℓ labeled with $C^{-1}(0)$ for $C \in \varphi$ every point in $C^{-1}(0)$ falsifies φ by definition; the root corresponds to the empty clause so it is labeled with $\{0,1\}^n$. The reverse also holds, one can convert a decision dag solving Search_φ to a resolution refutation of φ of the same size.

▶ **Proposition 4** ([10]). *There exists a resolution refutation of φ of size S and block-width b if and only if there exists a decision dag solving* Search_φ *of size S and block-width b.*

Box dag. Let $S \subseteq \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_k \times \mathcal{O}$ be a relation. Let $\mathcal{F}$ be a set of boxes $\{A_1 \times A_2 \times \cdots \times A_k \mid A_1 \subseteq \mathcal{X}_1, A_2 \subseteq \mathcal{X}_2, \ldots, A_k \subseteq \mathcal{X}_k\}$. Then we call an $\mathcal{F}$-dag a box dag. Let $U = U_1, \ldots, U_k$ be a partition of $[n]$. If $\mathcal{X}_i = \{0,1\}^{U_i}$ for all $i \in [k]$, then we denote the class of box dags as $\mathsf{box\text{-}dag}_U$ or $\mathsf{box\text{-}dag}_{U_1,\ldots,U_k}$.

▶ Remark 5. We can convert a π-OBDD refutation of a formula φ of size S to an $\mathcal{F}$-dag for Search_φ, where $\mathcal{F}$ consists of zero-sets of π-OBDDs of size at most S. In Section 5 we show that if a partition of variables into k parts agrees with an order π, such a dag can be converted to a box dag of size $S^{O(k)}$.

Automatability. A propositional proof system Π is called *automatable* if there exists an algorithm $\mathcal{A}_\Pi$ that given an unsatisfiable CNF φ produces its refutation in Π in time polynomial in $|\varphi|$ and the size of the shortest refutation of φ in Π.

3 The outline of the proof of Theorem 1

Our starting point is the following theorem that is essentially proved in [13].

▶ **Theorem 6** (Lemma 2.2 from [13]). *For any constant $c \geq 2$ there exists a polynomial-time algorithm $\mathcal{E}$ such that given a 3-CNF formula φ of size n it produces a $O(\log n)$-CNF formula $\mathcal{E}(\varphi)$ such that*

- *there exists a partition $B_1, \ldots, B_k$ of the variables of $\mathcal{E}(\varphi)$ such that $|B_1| = |B_2| = \cdots = |B_k| = O(n)$ and $k = O(n^{c+1})$ and this partition can be computed in polynomial time;*
- *if $\varphi \in \mathrm{SAT}$ then $\mathcal{E}(\varphi)$ has a resolution refutation π such that $|\pi| = n^{O(c)}$ and* $\mathrm{bw}(\pi) = O(1)$ *w.r.t. partition $B_1, \ldots, B_k$;*
- *if $\varphi \notin \mathrm{SAT}$ then any resolution refutation of $\mathcal{E}(\varphi)$ has block-width at least n^{c-1} w.r.t. $B_1, \ldots, B_k$.*

Notice that the statement of Theorem 6 is slightly different from one explicitly stated in [13]. First, it is not stated that all blocks B_i have equal sizes and their sizes are $O(n)$, but this is clear from the definition in Section 3.1 of [13]. Second, the theorem is stated and proved only for $c = 2$ but essentially the same proof holds for larger c, the only change is that we should reduce from rPHP_{n^c} instead of rPHP_{n^2} (see Section 5 of [13] for details).

To prove Theorem 1 we follow the plan below:

Lifting with multi-output indexing function. In Section 4 we define a block-wise indexing function $\textsc{Ind}_{\ell\times m}$ and its composition with relations and formulas. In Section 4 we will see that if a CNF formula φ has short resolution refutation of constant block-width then $\varphi \circ \textsc{Ind}^n_{\ell\times m}$ has a short resolution refutation. In the remainder of Section 4 we show that if a CNF formula φ with variables partitioned into n blocks of size ℓ requires resolution refutations of block-width at least b, then $\mathrm{Search}_\varphi \circ \textsc{Ind}^n_{\ell\times m}$ and consequently $\mathrm{Search}_{\varphi\circ\textsc{Ind}^n_{\ell\times m}}$ requires large $(\ell + 1)$-dimensional box dags.

Making box dags out of π-OBDD refutations. In Section 5 we show that if Search_φ requires k-dimensional box dags of size S, then it requires π-OBDD$(\wedge,$ weakening$)$ refutations of size $S^{\Omega(1/k)}$ for some fixed π.

Making all orders hard. In Section 6 we adapt Segerlind's transformation from [24] to show that there exists a CNF-to-CNF mapping that maps CNF formulas with polynomial resolution size to CNF formulas with polynomial resolution size and maps CNF formulas that are hard for π-OBDD$(\wedge,$ weakening$)$ with a fixed π to CNF formulas that are hard for OBDD$(\wedge,$ weakening$)$.

Putting the pieces together. In Section 7 we compose $\mathcal{E}_c$ with the two mappings above to obtain Theorem 1.

4 Lifting with multi-output indexing function

In this section, we prove the lifting theorem for box dags. First, let us formally define the gadget we are going to lift with.

▶ **Definition 7** (Block-wise indexing, [12]). *$\textsc{Ind}_{\ell\times m} : [m] \times \{0,1\}^{\ell\times m} \to \{0,1\}^\ell$ is defined as $\textsc{Ind}_{\ell\times m}(x, y) = (y_{1,x}, y_{2,x}, \dots, y_{\ell,x})$ i.e. given an index $x \in [m]$ and a matrix $y \in \{0,1\}^{\ell\times m}$, it returns the xth column of y. For a set $R \subseteq [m]^n \times (\{0,1\}^{\ell\times m})^n$ we define $\textsc{Ind}^n_{\ell\times m}(R) = \{(\textsc{Ind}_{\ell\times m}(x_1, y_1), \dots, \textsc{Ind}_{\ell\times m}(x_n, y_n)) \in \{0,1\}^{n\ell} \mid (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) \in R\}$.*

Let $\varphi = \bigwedge_{i=1}^t C_i$ be an unsatisfiable CNF with $n\ell$ variables that are partitioned into n blocks of size ℓ. Let us define a CNF $\psi = \varphi \circ \textsc{Ind}^n_{\ell\times m}$. First let us define $C \circ \textsc{Ind}^n_{\ell\times m}$ for a clause C. The resulting CNF formula will compute the function $C \circ \textsc{Ind}^n_{\ell\times m} = C(\textsc{Ind}_{\ell\times m}(x_1, y_1), \dots, \textsc{Ind}_{\ell\times m}(x_n, y_n))$. Then we define $\varphi \circ \textsc{Ind}^n_{\ell\times m} := \bigwedge_{i=1}^t \left(C_i \circ \textsc{Ind}^n_{\ell\times m}\right)$.

Now let us construct a CNF representation of $C \circ \textsc{Ind}^n_{\ell\times m}$. Let $z_{i,j}$ for $i \in [n]$, $t \in [\ell]$ be the tth variable of the ith block of φ. Let $i_1, \dots, i_b \in [n]$ be indices of the blocks that are touched by C and let C_j for $j \in [b]$ be the part of the variables of C from the i_jth block: $C = C_1 \vee \dots \vee C_b$. Let $P_j := \{k \in [\ell] \mid z_{i_j,k} \in C\}$ be the indices (inside a block) of positive literals in C_j and $N_j := \{k \in [\ell] \mid \neg z_{i_j,k} \in C\}$ be the indices of negative literals in C_j. Then the CNF representation of $C \circ \textsc{Ind}^n_{\ell\times m}(x_1, y_1, \dots, x_n, y_n)$ consists of clauses of form $\left(\left(\bigwedge_{j=1}^b (x_{i_j} = \alpha_j)\right) \to \left(\bigvee_{j=1}^b \left(\bigvee_{k\in P_j} y_{k,\alpha_j} \vee \bigvee_{k\in N_j} \neg y_{k,\alpha_j}\right)\right)\right)$ for each $\alpha_1, \dots, \alpha_b \in [m]$.

The size of this representation is $|\varphi| \cdot m^b$ where b is the largest block-width of a clause in φ, so this representation is short for formulas of constant block-width.

▶ **Theorem 8** (the last inequality in Theorem 4 from [12]). *Let φ be an unsatisfiable CNF with $n\ell$ variables that are partitioned into n blocks of size ℓ such that there exists a resolution refutation of φ of size s and block-width b. Then there exists a resolution refutation of $\varphi \circ \textsc{Ind}^n_{\ell\times m}$ of size $m^{O(b)} \cdot s$.*

4.1 Lifting theorem

For a relation $S \subseteq (\{0,1\}^\ell)^n \times \mathcal{O}$ its composition with block-wise indexing is defined as

$$S \circ \textsc{Ind}^n_{\ell\times m} := \left\{ (x_1,\dots,x_n,y_1,\dots,y_n,o) \;\middle|\; \begin{array}{l} x_i \in [m];\ y_i \in \{0,1\}^{\ell\times m};\ o \in \mathcal{O}; \\ (\textsc{Ind}_{\ell\times m}(x_1,y_1),\dots,\textsc{Ind}_{\ell\times m}(x_n,y_n),o) \in S \end{array} \right\}.$$

This is a direct analog of the composition of two functions: we first plug the output of indexing to each ℓ-bit block of the input of S and then "compute" S on the resulting input.

We assume that m is a power of 2 so the relation $S \circ \textsc{Ind}^n_{\ell\times m}$ can be viewed as defined on a binary domain $\{0,1\}^{n\log_2 m+\ell nm}$.

Let us define a partition of the input bits of relation $S \circ \textsc{Ind}^n_{\ell\times m}$. Consider an element of the input domain $(x_1,\dots,x_n,y_1,\dots,y_n) \in [m]^n \times \left(\{0,1\}^{\ell\times m}\right)^n$ where $x_1,\dots,x_n \in [m]$ and $y_1,\dots,y_n$ are matrices in $\{0,1\}^{\ell\times m}$. Let A consist of bits corresponding to of $x_1,\dots,x_n$, (in other words A corresponds to the first $n\log_2 m$ bits of the input), B_j for $j \in [\ell]$ consists of bits corresponding to jth rows of all the matrices $y_1,\dots,y_n$. We are going to imagine that we have $\ell+1$ parties: Alice who receives the bits A of the input, Bob$_1$, Bob$_2$, …, Bob$_\ell$, where Bob$_j$ receives the bits B_j of the input.

Then let $\mathcal{A} := \{0,1\}^A = [m]^n$ be the set of Alice's inputs and let $\mathcal{B}_j := \{0,1\}^{B_j} = \{0,1\}^m$ be the set of Bob$_j$'s inputs.

The following theorem is similar with Theorem 8 form [12], but for box dags instead of simplex dags and, crucially, for a smaller number of parties, $\ell+1$ instead of $n\ell+1$.

▶ **Theorem 9.** *Let Δ be a large enough integer constant. Let $S \subseteq (\{0,1\}^\ell)^n \times \mathcal{O}$ be a total relation where $\ell < \frac{n}{2}$ and $m = (n\ell)^\Delta$. Then $m^{\Omega(\mathrm{bw}(S))} \leq \textsf{box-dag}_{A,B_1,\dots,B_\ell}(S \circ \textsc{Ind}^n_{\ell\times m})$, where the block partition of inputs of S is the natural partition into n blocks of size ℓ.*

Let us outline the proof of Theorem 9. The proof is constructive, i.e., we take a box dag $\mathbb{B}$ solving $S \circ \textsc{Ind}^n_{\ell\times m}$ and extract from it a decision dag solving S of block-width $O(\log|\mathbb{B}|/\log m)$. The idea is to split boxes in the box dag into "structured" boxes that naturally correspond to partial assignments from $\{0,1,*\}^n$ (notice that there is a one-to-one correspondence between partial assignments and subcubes). We then take the assignments that our structured boxes correspond to and construct a decision dag for S out of them (we will need some auxiliary partial assignments as well). A first attempt to formulate what this "structuredness" could mean is the following: a box B is ρ-like if $\textsc{Ind}^n_{\ell\times m}(B) = \mathrm{Cube}(\rho)$. It turns out that we actually can (with some caveats) partition any box in $\mathcal{A} \times \mathcal{B}_1 \times \cdots \times \mathcal{B}_\ell$ into boxes that are ρ-like for some assignments ρ. Unfortunately, we need some additional properties of these boxes to be able to connect them into a valid decision dag.

Our definition of structured boxes is different from the one given in [12], we formulate it in a different way reducing the structuredness of boxes to the structuredness of rectangles (2-dimensional boxes) that is used to prove the lifting theorem in [10]. In Subsection 4.2 we formulate the properties of structured rectangles that we need, in Subsection 4.3 we define and prove the analogous properties for structured boxes, and in Subsection 4.4 we construct the decision dag solving S.

4.2 Structured Rectangles

Lifting theorems from [10] rely heavily on the notion of *structuredness* of rectangles. To simplify things we will not define it explicitly, but instead, state its properties that we are going to use.

Let $\mathsf{Rect}_{m,n}$ be the set of subrectangles of $[m]^n \times \left(\{0,1\}^{1\times m}\right)^n$: $\{A \times B \mid A \subseteq [m]^n;\ B \subseteq \left(\{0,1\}^{1\times m}\right)^n\}$. We are going to define several properties of predicates on $\mathsf{Rect}_{m,n} \times \{0,1,*\}^n$ i.e. predicates on pairs of form (rectangle, partial assignment). Let $\mathcal{W}$ be a predicate on $\mathsf{Rect}_{m,n} \times \{0,1,*\}^n$.

▶ **Definition 10.** *We say that* $\mathcal{W}$ observes row-structure *if* $\mathcal{W}(X \times Y, \rho)$ *implies that for all* $x \in X$, $\textsc{Ind}^n_{1\times m}(\{x\} \times Y) \subseteq \mathrm{Cube}(\rho)$, *and* $\Pr_{x \leftarrow \mathcal{U}(X)}\left[\textsc{Ind}^n_{1\times m}(\{x\} \times Y) \neq \mathrm{Cube}(\rho)\right] \leq \frac{2}{n}$.

▶ **Definition 11.** *We say that* $\mathcal{W}$ *is* partitionable *if for every* $X \subseteq [m]^n$ *there exist a partition* $X := \bigsqcup_{j \in J} \tilde{X}_j$ *and a family* $\{F_j\}_{j \in J}$, $F_j \subseteq [n]$, *and for every* $R = X \times Y \in \mathsf{Rect}_{m,n}$, *for every parameter* $k \leq n \log n$ *there exists a partition* $R = \bigsqcup_{i \in I} R_i$, *where* $R_i = X_i \times Y_i \in \mathsf{Rect}_{m,n}$, *a family of assignments* $\{\rho_i\}_{i \in I}$, *and sets* $X_{err} \subseteq X, Y_{err} \subseteq Y$ *such that* $|X_{err}| \leq m^n/2^k$, $|Y_{err}| \leq 2^{mn-k}$ *and the following properties hold:*

1. *for each* i *one of the following holds: either* $\mathcal{W}(R_i, \rho_i)$ *and* $|\mathrm{fix}(\rho_i)| = O(k/\log n)$; *or* R_i *is covered by* $X_{err} \times \left(\{0,1\}^{1\times m}\right)^n \cup [m]^n \times Y_{err}$.
2. *For every* $i \in I$ *there exists* $j \in J$ *such that* $\tilde{X}_j = X_i$ *and* $\mathrm{fix}(\rho_i) = F_j$ [1].

▶ **Definition 12.** *We say that* $\mathcal{W}$ respects largeness *if for all* $X \times Y$ *such that* $|X| \geq m^n \cdot 0.99$ *and* $|Y| \geq 2^{mn} \cdot 0.99$ $\mathcal{W}(X \times Y, *^n)$ *holds.*

▶ **Theorem 13** (Lemma 4.4, Lemma 4.5 from [10])**.** *There exists a constant* Δ *such that for any* $m \geq n^\Delta$ *there exists a predicate* $\mathcal{W}$ *on* $\mathsf{Rect}_{m,n} \times \{0,1,*\}^n$ *such that it* observes row-structure[2]; is partitionable; respects largeness[3]. *We say that a rectangle* R *is* ρ-structured *iff* $\mathcal{W}(R, \rho)$ *holds.*

4.3 Structured Boxes

Now let us generalize the notion of structuredness from rectangles to boxes.

▶ **Definition 14.** *Let* $R = X \times Y_1 \times \cdots \times Y_\ell$, *where* $X \subseteq \mathcal{A} = [m]^n$, $Y_j \subseteq \mathcal{B}_j = (\{0,1\}^{1\times m})^n$ *be a box and* $\rho \in \{0,1,*\}^{n\ell}$ *be a partial assignment. We view* ρ *as an assignment to variables of input to* $S \subseteq (\{0,1\}^\ell)^n \times \mathcal{O}$ *that are partitioned into* n *blocks of size* ℓ. *Let* $\rho_i \in \{0,1,*\}^n$ *for* $i \in [\ell]$ *be the marginal assignment of* ρ *assigning the ith variable of each block in the partition of variables of* S. *We say that* R *is a* ρ-structured box *if for each* $i \in [\ell]$ *the rectangle* $X \times Y_i$ *is* ρ_i*-structured.*

We now show that our definition of the structuredness satisfies the analogues of conditions from Definitions 10, 11, and 12.

▶ **Lemma 15.** *Assume that* $n > 2\ell$. *Let* $R = X \times Y_1 \times \cdots \times Y_\ell \subseteq \mathcal{A} \times \mathcal{B}_1 \times \cdots \times \mathcal{B}_\ell$ *be a* ρ*-structured box where* $\rho \in \{0,1,*\}^{n\ell}$. *Then for all* $x \in X$, $\textsc{Ind}^n_{\ell\times m}(\{x\} \times Y_1 \times \cdots \times Y_\ell) \subseteq \mathrm{Cube}(\rho)$ *and there exists* $x \in X$ *such that* $\textsc{Ind}^n_{\ell\times m}(\{x\} \times Y_1 \times \cdots \times Y_\ell) = \mathrm{Cube}(\rho)$.

[1] This property is not explicitly stated in [10], although it is clear from the Rectangle Scheme that generates the partition: first X is partitioned and then each part $X_i \times Y$ is partitioned separately.

[2] Although Lemma 4.4 of [10] is not stated in strong enough form to satisfy Definition 10, the needed property is actually proved in Section 9 of [10].

[3] This property is implicit in [10], see the full version of the paper for the details.

Proof. If there exist $x \in X, y_1 \in Y_1, \ldots, y_\ell \in Y_\ell$ such that $\alpha := \textsc{Ind}^n_{\ell\times m}(x, y_1, \ldots, y_\ell)$ does not agree with ρ, then there exists $i \in [\ell]$ such that $\textsc{Ind}^n_{1\times m}(x, y_i)$ does not agree with ρ_i which violates Definition 10.

Now let us prove the second statement. By Definition 10 for each $i \in [\ell]$ we have $\Pr_{x\leftarrow\mathcal{U}(X)}\left[\textsc{Ind}^n_{1\times m}(\{x\}\times Y_i) \neq \mathrm{Cube}(\rho_i)\right] \leq \frac{2}{n}$. Then $\Pr_{x\leftarrow\mathcal{U}(X)}\left[\textsc{Ind}^n_{\ell\times m}(\{x\}\times Y_1 \times Y_2 \times \cdots \times Y_\ell) \neq \mathrm{Cube}(\rho)\right]$ is bounded by $\sum_{i=1}^{\ell} \Pr_{x\leftarrow\mathcal{U}(X)}\left[\textsc{Ind}^n_{1\times m}(\{x\}\times Y_i) \neq \mathrm{Cube}(\rho_i)\right] \leq \frac{2\ell}{n} < 1$. ◀

▶ **Lemma 16.** *If $R = X \times Y_1 \times \cdots \times Y_\ell \subseteq \mathcal{A} \times \mathcal{B}_1 \times \cdots \times \mathcal{B}_\ell$ is such that $|X| \geq m^n \cdot 0.99$ and $|Y_i| \geq 2^{mn} \cdot 0.99$ for each $i \in [\ell]$, then R is $*^{n\ell}$-structured.*

Proof. By Definition 12 we have that each of the $X \times Y_i$ is $*^n$-structured which by definition implies $*^{n\ell}$-structuredness of R. ◀

▶ **Lemma 17.** *Let $R = X \times Y_1 \times \cdots \times Y_\ell \subseteq \mathcal{A} \times \mathcal{B}_1 \times \cdots \times \mathcal{B}_\ell$ be an arbitrary box and $k \leq n \log n$ be a parameter. Then there exist sets $X^{err} \subseteq \mathcal{A}, Y_1^{err} \subseteq \mathcal{B}_1, \ldots, Y_\ell^{err} \subseteq \mathcal{B}_\ell$, a partition $R = \bigsqcup_{i\in I} R_i$, and a family of partial assignments $\{\rho^i\}_{i\in I}$, where $R_i = X^i \times Y_1^i \times \cdots \times Y_\ell^i$ is a box and $\rho^i \subseteq \{0, 1, *\}^{n\ell}$ satisfying the following conditions.*

(1*) $|X^{err}| \leq \frac{m^n\cdot\ell}{2^k}$, $|Y_i^{err}| \leq 2^{nm-k}$.

(2*) *For each $i \in I$ at least one of the following statements holds:*
- *R_i is ρ^i-structured and ρ^i assigns $O(k/\log n)$ blocks from the standard partition of $[n\ell]$ into n blocks of size ℓ;*
- *R_i is covered by one of the error sets i.e. $X^i \subseteq X^{err}$ or there exists $j \in [\ell]$ such that $Y_j^i \subseteq Y_j^{err}$.*

(3*) *For each $x \in X \setminus X^{err}$ there exists a set $I_x \subseteq [n\ell]$ that is a union of $O(k/\log n)$ blocks (i.e. it either contains all the indices from a block or none) such that $x \in X^i$ implies $\mathrm{fix}(\rho^i) \subseteq I_x$.*

4.4 Proof of Theorem 9

Recall that the inequality we are to prove is $m^{\Omega(\mathrm{bw}(S))} \leq \textsf{box-dag}_{A,B_1,\ldots,B_\ell}(S \circ \textsc{Ind}^n_{\ell\times m})$. It is equivalent to $\mathrm{bw}(S) = O\left(\log \textsf{box-dag}_{A,B_1,\ldots,B_\ell}(S \circ \textsc{Ind}^n_{\ell\times m})/\log m\right)$.

Consider the smallest $\textsf{box-dag}_{A,B_1,\ldots,B_\ell}$ $\mathbb{B}$ solving $S \circ \textsc{Ind}^n_{\ell\times m}$. We construct a decision dag solving S of block-width $O(\log|\mathbb{B}|/\log m) = O(\log|\mathbb{B}|/\log n)$.

Similarly to [10] we first assume that partitions yielded by Lemma 17 are always errorless, i.e. $X^{err} = Y_1^{err} = \cdots = Y_\ell^{err} = \emptyset$. Then we will fix the proof so it works without this assumption, this part of the proof repeats the argument from Section 5.3 in [10] more or less verbatim, so we omit it in this version of the paper. We apply Lemma 17 to each of the boxes in $\mathbb{B}$ with some parameter k that we fix later to achieve the needed lower bound.

Let us construct a decision dag $\mathbb{D}$ that solves S. Each node of a decision dag labeled with function f naturally corresponds to a partial assignment ρ_f such that $\mathrm{Cube}(\rho_f) = f^{-1}(0)$. We will identify nodes of a decision dag with the assignments corresponding to them. That suggests the construction of $\mathbb{D}$: for each of the nodes of $\mathbb{B}$ we apply Lemma 17 to it and for each ρ-structured box in the resulting partition add the node ρ to $\mathbb{D}$. To turn this collection of nodes into a correct decision dag, we need to locate the root, the leaves, and connect (via auxiliary nodes) the nodes between each other such that the conditions on dags are met. More precisely, it is sufficient to show that:

1. The partition of the root of $\mathbb{B}$ consists of a single $*^{n\ell}$-structured box.
2. If an o-labeled leaf of $\mathbb{B}$ contains a ρ-structured box in its partition, then for every $x \in \mathrm{Cube}(\rho)$, $(x, o) \in S$.
3. Suppose a node u in $\mathbb{B}$ has direct descendants v_1 and v_2. Then let $\rho^u_1, \dots, \rho^u_{t_u}$ be the assignments yielded by the partition of the box u, $\rho^{v_q}_1, \dots, \rho^{v_q}_{t_{v_q}}$ be the assignments yielded by the partition of the box v_q for $q \in \{1, 2\}$. Then there exists a assignment-labeled dag with sources $\rho^u_1, \dots, \rho^u_{t_u}$, leaves $\rho^{v_q}_1, \dots, \rho^{v_q}_{t_{v_q}}$ for $q \in \{1, 2\}$ that satisfies the local condition of a decision dag having block-width $O(k/\log n)$.

Proof of 1. By Lemma 16 we have that the entire root of $\mathbb{B}$ is $*^{n\ell}$-structured, thus we may assume that its partition is a single box.

Proof of 2. Let u be an o-labeled leaf of $\mathbb{B}$. Suppose that $B = X \times Y_1 \times \dots \times Y_\ell$ is a ρ-structured box in the partition of u. By Lemma 15 there exists x_0 such that $\textsc{Ind}^n_{\ell\times m}(\{x_0\} \times Y_1 \times \dots \times Y_\ell) = \mathrm{Cube}(\rho)$, i.e. for every $\alpha \in \mathrm{Cube}(\rho)$ there exist $y_1, \dots, y_\ell$ such that $(x_0, y_1, \dots, y_\ell) \in B$ and $\textsc{Ind}^n_{\ell\times m}(x_0, y_1, \dots, y_\ell) = \alpha$. Then since $\mathbb{B}$ is a box-dag for $S \circ \textsc{Ind}^n_{\ell\times m}$, $(\alpha, o) \in S$.

Proof of 3. It is sufficient to construct a separate dag with local property rooted in ρ^u_i with leaves from $\mathcal{L} := \{\rho^{v_q}_p\}_{q\in\{1,2\},\, p\in[t_{v_q}]}$ of block-width $O(k/\log n)$.

Recall that we abuse notation by identifying nodes of a box dag with their underlying boxes. Let $B = X \times Y_1 \times \dots \times Y_\ell$ be a ρ^u_i-structured box from the partition of u. And let $x \in X$ be such that $\textsc{Ind}^n_{\ell\times m}(\{x\} \times Y_1 \times \dots \times Y_\ell) = \mathrm{Cube}(\rho^u_i)$. By the property of a box-dag, B is covered by the union of boxes v_1 and v_2. Thus $\{x\} \times Y_1 \times \dots \times Y_\ell$ is also covered by $v_1 \cup v_2$. Let $I^{v_1}_x, I^{v_2}_x \subseteq [n\ell]$ be the variable sets from Lemma 17. Let our ρ^u_i-rooted decision dag consist of two parts. The first part is a decision tree querying one by one all variables from $I^{v_1}_x \cup I^{v_2}_x \setminus \mathrm{fix}(\rho^u_i)$. From each leaf of this decision tree we direct both edges to one of the nodes of $\mathcal{L}$. Observe that by the part (3) of Lemma 17, $I^{v_1}_x$ and $I^{v_2}_x$ are unions of $O(k/\log n)$ blocks and $\mathrm{fix}(\rho^u_i)$ touches $O(k/\log n)$ blocks. Thus block-width of the resulting dag is also $O(k/\log n)$.Consider any leaf of the decision tree $\theta \in \{0, 1, *\}^{n\ell}$. Since $\textsc{Ind}^n_{\ell\times m}(\{x\} \times Y_1 \times \dots \times Y_\ell) = \mathrm{Cube}(\rho^u_i)$ and θ extends ρ^u_i (i.e., $\mathrm{Cube}(\theta) \subseteq \mathrm{Cube}(\rho^u_i)$), there exist $y_1 \in Y_1, \dots, y_\ell \in Y_\ell$ such that $\textsc{Ind}^n_{\ell\times m}(x, y_1, \dots, y_\ell) \in \mathrm{Cube}(\theta)$. Then consider an ω-structured box B_0 from a partition of v_1 or v_2 for $\omega \in \mathcal{L}$ that contains $(x, y_1, \dots, y_\ell)$. Observe that $\mathrm{fix}(\omega) \subseteq I^{v_1}_x \cup I^{v_2}_x \subseteq \mathrm{fix}(\theta)$. The first inclusion holds by the part (3) of Lemma 17, the second holds by the construction of the decision tree. Since by Lemma 15, $\textsc{Ind}^n_{\ell\times m}(x, y_1, \dots, y_\ell) \in \mathrm{Cube}(\omega)$, $\mathrm{Cube}(\omega)$ and $\mathrm{Cube}(\theta)$ have a point in common, then $\mathrm{fix}(\omega) \subseteq \mathrm{fix}(\theta)$ implies $\mathrm{Cube}(\omega) \supseteq \mathrm{Cube}(\theta)$. Then we can direct both edges from ω to θ. That finishes the proof under the errorless assumption.

5 From Box-Dags to OBDD Refutations

▶ **Lemma 18** (a generalization of a similar lemma in [25]). *Let $U_1, \dots, U_k$ be a partition of $[n]$. Let $\mathcal{F}$ be the class of* functions *that are computable by k-party number-in-hand communication protocol*[4] *of cost c w.r.t. partition $U_1, \dots, U_k$ of $[n]$. Let $S \subseteq \{0,1\}^{U_1} \times \dots \times \{0,1\}^{U_k} \times Y$ be a relation and let D be a $\mathcal{F}$-dag that solves it. Then there exists a* $\mathsf{box\text{-}dag}_{U_1,\dots,U_k}$ *D' of size $O(|D| \cdot 2^{3c})$ that solves S.*

[4] For a formal definition of number-in-hand protocol see e.g. [21].

Let X be a set of propositional variables of size n, $\mathcal{V} := (V_1, \dots, V_k)$ be a partition of X: $X = V_1 \sqcup \dots \sqcup V_k$, and $\pi : [n] \to X$ be a bijection (order on the variables X). We say that a partition $\mathcal{V}$ *agrees with* π if V_1 comes first in the order, then goes V_2 and so on until V_k.

▶ **Theorem 19.** *Let φ be an unsatisfiable CNF over variables X. Let $\pi : [n] \to X$ be an order of variables and $\mathcal{V}$ be a partition of X agreeing with π. Let $D_1, \dots, D_t$ be a π-OBDD($\wedge$, weakening) refutation of φ of size S. Then* $\mathsf{box\text{-}dag}_{\mathcal{V}}(\mathrm{Search}_\varphi) \le S^{O(k)}$.

▶ **Lemma 20.** *Let D be a π-OBDD over variables X computing a function f and $\mathcal{V} = (V_1, \dots, V_k)$ be a partition of X that agrees with π. Then there exists a k-party number-in-hand communication protocol computing f with cost $k\lceil \log_2 |D| \rceil$.*

Proof of Theorem 19. By Lemma 20, a π-OBDD refutation of φ of size $S = \sum_{i=1}^{t} |D_i|$ can be viewed as an $\mathcal{F}$-dag solving Search_φ (for the diagrams derived via the weakening rule we direct both of the outgoing edges to the same node), where $\mathcal{F}$ is the class of functions that can be computed with cost at most $k\lceil \log_2 S \rceil$ by a k-party number-in-hand communication protocol with input partition $\mathcal{V}$. Then by Lemma 18, there exists a $\mathsf{box\text{-}dag}_{\mathcal{V}}$ of size $S \cdot 2^{3k \log S} = S^{O(k)}$ solving Search_φ. ◀

6 Making all orders hard

Let *negative width* of a resolution refutation be the maximal number of negative literals in a clause of the refutation.

▶ **Theorem 21** ([24]). *There exists a polynomial-time algorithm $\mathcal{T}_0$ that given a CNF φ over n variables returns a CNF-formula $\mathcal{T}_0(\varphi)$ such that*

- *for any variable ordering π, π-OBDD$(\varphi) \le$ OBDD$(\mathcal{T}_0(\varphi))$ (Lemma 14 from [24]);*
- *If φ has a resolution refutation of size s and* negative width *w, then $\mathcal{T}_0(\varphi)$ has resolution size at most $s \cdot n^{O(w)}$, (Corollary 9 and Lemma 12 from [24]).*

▶ **Lemma 22.** *If a CNF-formula φ has a resolution refutation of size s and the size of the smallest π-OBDD refutation of φ is t, then there exists polynomial-time algorithm that given φ outputs a formula φ' and a variable order π' such that φ' has a resolution refutation of size $O(s)$ and negative width $O(1)$, and the size of the smallest π'-OBDD refutation of φ' is at least t.*

▶ **Corollary 23.** *There exists a polynomial-time algorithm $\mathcal{T}$ that given a CNF φ over n variables returns a CNF-formula $\mathcal{T}(\varphi)$ such that for any variable ordering π, π-OBDD$(\varphi) \le$ OBDD$(\mathcal{T}(\varphi))$; and if φ has a resolution refutation of size s, then the resolution size of $\mathcal{T}(\varphi)$ is at most $s \cdot n^{O(1)}$.*

Proof. The new algorithm $\mathcal{T}$ first applies the transformation from Lemma 22 to a CNF formula and only then applies the algorithm $\mathcal{T}_0$ from Theorem 21 to it. ◀

7 Putting the pieces together

▶ **Theorem 1.** *There exist a constant α and a polynomially computable function $\mathcal{R}$ mapping CNF formulas to CNF formulas with the following properties. For any 3-CNF φ with n variables such that: if φ is satisfiable, then $\mathcal{R}(\varphi)$ has a resolution refutation of size at most n^α; if φ is unsatisfiable, then any* OBDD($\wedge$, weakening) *refutation of $\mathcal{R}(\varphi)$ has size $2^{\Omega(n)}$.*

Proof. Let $\mathcal{E}$ be the algorithm from Theorem 6 with the parameter $c = 3$, and $\mathcal{T}$ be the algorithm from Corollary 23. Let n be the number of variables of φ and let n_φ be the number of variables in $\mathcal{E}(\varphi)$. Let ℓ_φ be the size of the blocks in the block partition in Theorem 6, $\ell_\varphi = O(n)$. Then let $m_\varphi = (n_\varphi \ell_\varphi)^\Delta$ where Δ is from Theorem 9 and let $\mathcal{R}(\varphi) := \mathcal{T}(\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi})$.

Let us first consider the case of $\varphi \in \mathrm{SAT}$. Then by Theorem 6, $\mathcal{E}(\varphi)$ has a resolution refutation π such that $|\pi| = |\varphi|^{O(1)}$ and $\mathrm{bw}(\pi) = O(1)$. Then applying Theorem 8 we get that there exists a resolution refutation of $\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi}$ of size $|\varphi|^{O(1)}$. Then by Corollary 23 $\mathcal{T}(\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi})$ has a resolution refutation of size $|\varphi|^{O(1)}$.

Let us proceed with the case $\varphi \notin \mathrm{SAT}$. Suppose $\mathcal{R}(\varphi)$ has a $\mathrm{OBDD}(\wedge, \mathrm{weakening})$ refutation of size S. Then by Corollary 23 the formula $\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi}$ has a π-$\mathrm{OBDD}(\wedge, \mathrm{weakening})$ refutation of size S for any variable order π. Then consider the order of variables π_0 where the variables of $\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi}$ are ordered as follows:

- All the variables corresponding to the indices in an arbitrary order (denote this set by A);
- All the variables from the first rows of the matrices (denote this set by B_1);
- ...
- All the variables from the ℓ_φth rows of the matrices (denote this set by B_{ℓ_φ}).

The size of π_0-$\mathrm{OBDD}(\wedge, \mathrm{weakening})$ refutation of $\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi}$ is at most S which by Theorem 19 implies that $\textsf{box-dag}_{A,B_1,\dots,B_\ell}\left(\mathrm{Search}_{\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi}}\right) \leq S^{O(\ell_\varphi + 1)}$.

Then the fact that $\mathrm{Search}_{\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi}}$ is at least as hard as $\mathrm{Search}_{\mathcal{E}(\varphi)} \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi}$ and the inequality $\textsf{box-dag}_{A,B_1,\dots,B_\ell}\left(\mathrm{Search}_{\mathcal{E}(\varphi)} \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi}\right) \geq m_\varphi^{\Omega(\mathrm{bw}(\mathcal{E}(\varphi)))}$ implied by Theorem 9 together imply that $S \geq m_\varphi^{\Omega(\mathrm{bw}(\mathcal{E}(\varphi))/(\ell_\varphi + 1))}$. By Theorem 6 using Proposition 4 to switch from decision dag to resolution refutation we have $\mathrm{bw}(\mathcal{E}(\varphi)) = \Omega(n^{c-1}) = \Omega(n^2)$ which implies that $S \geq m_\varphi^{\Omega(n)}$ since $\ell_\varphi = O(n)$. This completes the proof of the theorem since $m_\varphi \geq 2$. ◀

▶ **Corollary 24.** *If* $\mathrm{OBDD}(\wedge, \mathrm{weakening})$ *is automatable then* $\mathrm{P} = \mathrm{NP}$.

References

1 Alfonso San Miguel Aguirre and Moshe Y. Vardi. Random 3-sat and bdds: The plot thickens further. In *Principles and Practice of Constraint Programming – CP 2001, 7th International Conference, CP 2001, Paphos, Cyprus, November 26 – December 1, 2001, Proceedings*, pages 121–136, 2001. doi:10.1007/3-540-45578-7_9.

2 Michael Alekhnovich, Samuel R. Buss, Shlomo Moran, and Toniann Pitassi. Minimum propositional proof length is np-hard to linearly approximate. *J. Symb. Log.*, 66(1):171–191, 2001. doi:10.2307/2694916.

3 Michael Alekhnovich and Alexander A. Razborov. Resolution is not automatizable unless W[P] is tractable. *SIAM J. Comput.*, 38(4):1347–1363, 2008. doi:10.1137/06066850X.

4 Albert Atserias, Phokion G. Kolaitis, and Moshe Y. Vardi. Constraint propagation as a proof system. In Mark Wallace, editor, *Principles and Practice of Constraint Programming – CP 2004, 10th International Conference, CP 2004, Toronto, Canada, September 27 – October 1, 2004, Proceedings*, volume 3258 of *Lecture Notes in Computer Science*, pages 77–91. Springer, 2004. doi:10.1007/978-3-540-30201-8_9.

5 Albert Atserias and Moritz Müller. Automating resolution is np-hard. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 498–509. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00038.

6 Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldà, Alexis Maciel, and Toniann Pitassi. Non-automatizability of bounded-depth frege proofs. *Comput. Complex.*, 13(1-2):47–68, 2004. doi:10.1007/s00037-004-0183-5.

7 Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. No feasible interpolation for tc0-frege proofs. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 254–263. IEEE Computer Society, 1997. doi:10.1109/SFCS.1997.646114.

8 Randal E. Bryant. Symbolic Boolean manipulation with ordered binary-decision diagram. *ACM Computing Surveys*, 24(3):293–318, 1992.

9 Sam Buss, Dmitry Itsykson, Alexander Knop, and Dmitry Sokolov. Reordering rule makes OBDD proof systems stronger. In *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, pages 16:1–16:24, 2018. doi:10.4230/LIPIcs.CCC.2018.16.

10 Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 902–911, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3188745.3188838.

11 Michal Garlík. Failure of feasible disjunction property for k-dnf resolution and np-hardness of automating it. *Electron. Colloquium Comput. Complex.*, page 37, 2020. URL: https://eccc.weizmann.ac.il/report/2020/037.

12 Mika Göös, Sajin Koroth, Ian Mertz, and Toniann Pitassi. Automating cutting planes is np-hard. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 68–77, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384248.

13 Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, Dmitry Sokolov, and Susanna F. de Rezende. Automating algebraic proof systems is np-hard. *Electron. Colloquium Comput. Complex.*, 27:64, 2020. URL: https://eccc.weizmann.ac.il/report/2020/064.

14 Dmitry Itsykson, Alexander Knop, Andrei E. Romashchenko, and Dmitry Sokolov. On obdd-based algorithms and proof systems that dynamically change the order of variables. *J. Symb. Log.*, 85(2):632–670, 2020. doi:10.1017/jsl.2019.53.

15 Kazuo Iwama. Complexity of finding short resolution proofs. In Igor Prívara and Peter Ruzicka, editors, *Mathematical Foundations of Computer Science 1997, 22nd International Symposium, MFCS'97, Bratislava, Slovakia, August 25-29, 1997, Proceedings*, volume 1295 of *Lecture Notes in Computer Science*, pages 309–318. Springer, 1997. doi:10.1007/BFb0029974.

16 Jan Krajiček. An exponential lower bound for a constraint propagation proof system based on ordered binary decision diagrams. *Journal of Symbolic Logic*, 73(1):227–237, 2008. doi:10.2178/jsl/1208358751.

17 Jan Krajícek and Pavel Pudlák. Some consequences of cryptographical conjectures for s_2^1 and EF. *Inf. Comput.*, 140(1):82–94, 1998. doi:10.1006/inco.1997.2674.

18 Christoph Meinel and Anna Slobodova. On the complexity of Constructing Optimal Ordered Binary Decision Diagrams. In *Proceedings of Mathematical Foundations of Computer Science*, volume 841, pages 515–524, 1994.

19 Ian Mertz, Toniann Pitassi, and Yuanhao Wei. Short proofs are hard to find. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 84:1–84:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.84.

20 Guoqiang Pan and Moshe Y. Vardi. Symbolic techniques in satisfiability solving. *Journal of Automated Reasoning*, 35(1-3):25–50, 2005. doi:10.1007/s10817-005-9009-7.

21 Jeff M. Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 486–501, USA, 2012. Society for Industrial and Applied Mathematics.

22 Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997. doi:10.2307/2275583.

23 Nathan Segerlind. Nearly-exponential size lower bounds for symbolic quantifier elimination algorithms and OBDD-based proofs of unsatisfiability. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(009), 2007. URL: http://eccc.hpi-web.de/eccc-reports/2007/TR07-009/index.html.

24 Nathan Segerlind. On the relative efficiency of resolution-like proofs and ordered binary decision diagram proofs. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, CCC 2008, 23-26 June 2008, College Park, Maryland, USA*, pages 100–111. IEEE Computer Society, 2008. doi:10.1109/CCC.2008.34.

25 Dmitry Sokolov. Dag-like communication and its applications. In *Computer Science – Theory and Applications – 12th International Computer Science Symposium in Russia, CSR 2017, Kazan, Russia, June 8-12, 2017, Proceedings*, pages 294–307, 2017. doi:10.1007/978-3-319-58747-9_26.

26 I. Wegener. *Branching Programs and Binary Decision Diagrams: Theory and Applications*. Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 2000. URL: https://books.google.ru/books?id=xqqJj42ZoXcC.