Electronic Colloquium on Computational Complexity, Report No. 46 (2022)

Automating OBDD proofs is NP-hard

Dmitry Itsykson* and Artur Riazanov†

St. Petersburg Department of V.A. Steklov Institute of Mathematics
of the Russian Academy of Sciences
Fontanka 27, St. Petersburg, 191023 Russia ‡

April 4, 2022

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 Muller [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. [16] to establish the hardness of automatability result for Cutting Planes.

1 Introduction

Boolean satisfiability is one of the central problems in Computer Science. Input in 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 that is pretty weak. 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.

An example of a non-trivial quasi-automatable proof system is tree-like resolution [7]; it is shown by de Rezende [13] that under ETH the automation in time $n^{O(\log n)}$ is optimal.

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 [2, 3, 21, 25] is concluded with the recent breakthrough result by Atserias and Muller [6] stating that resolution is not

*dmitrits@pdmi.ras.ru
†aariazanov@gmail.com
‡The research is supported by Russian Science Foundation (project 18-71-10042).

ISSN 1433-8092

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-Muller's result directly or follow their plan closely. If P $\neq$ NP, then the following proof systems are not automatable:

- Nullstellensatz, Polynomial Calculus, and Sherali-Adams [17];
- Cutting Planes [16];
- Res(2) [15].

Under stronger assumptions one can show non-automatability of Frege systems [8, 9, 23].

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 [10]. 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. 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, 26].

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 first ($\wedge$) we can derive conjunction of two OBDDs and by second (weakening) we can derive any semantic implication of a single OBDD). 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 [12] (the latter are hard for Cutting Planes [27]).

Atserias-Muller'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 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. First such transformation was devised by Krajićek [22]: 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 [28, 29] 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($\mathcal{O}(\log n)$).

Buss et. al. [12] used lifting theorem for dag-like communication by [14] (for the first step) combined with Segerlind's transformation in order to show that OBDD($\wedge$, weakening) does not simulate OBDD($\wedge$, reordering).

1.1 Our contribution

Our main result is the following theorem:

Theorem 1.1. *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*

- *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 OBDD($\wedge$, weakening) simulates resolution, any automation algorithm for OBDD($\wedge$, 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 theorem can be applied for any proof system that simultaneously can be simulated by OBDD($\wedge$, weakening) and simulates resolution. Among such systems we can mention:

- The proof system OBDD($\wedge, \exists$) [4] that uses the projection rule instead of the weakening rule (for OBDDs the projection rule is a partial case of the weakening rule).
- The system Res($\oplus, \leq k$) [20] operating with the disjunctions of linear equalities over $\mathbb{F}_2$, where all but k equations depend on one variable. It was shown in [20] that any proof Res($\oplus, \leq k$) of size S can be simulated by an OBDD($\wedge$, weakening) proof of size $2^{\mathcal{O}(k)}S$. So our result implies that automation of Res($\oplus, \leq k$) is NP-hard if $k \leq n^c$ for some constant c.

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) [19].

1.2 Technique

The proof consists of two parts:

1. Prove the weaker version of Theorem 1.1, 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;
   - formulas that are hard for OBDD($\wedge$, weakening) with a specific order to formulas that are hard for OBDD($\wedge$, 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 *negative width* of the proof is $\mathcal{O}(1)$), then the result of Segerlind's transformation has a short resolution proof. So we devise an additional polynomial-time algorithm that transforms

- formulas with short resolution refutations to formulas with short resolution refutations of constant negative width;
- formulas that are hard for OBDD($\wedge$, weakening) with a specific order to formulas that are hard for OBDD($\wedge$, weakening) with (another) specific order,

and apply it before Segerlind's transformation.

The first part is much more involved. The construction is built on the following theorem of Atserias and Muller.

Theorem 1.2 (Atserias, Muller, 2019 [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. [14] 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 $\mathcal{E}(\varphi)$ from Theorem 1.2) 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 OBDD($\wedge$, weakening) refutations. Göös et. al. [16] 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 [14] does not work for block-width, so Göös et. al. [16] 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 [14] and extend them to show the structural properties of boxes.

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 $S^{\mathcal{O}(k)}$. 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.

Further directions. Prove some non-automatability result for OBDD($\wedge$). It does not simulate resolution [11], so the non-automatability can not be shown in the Atserias-Muller style.

We believe that studying multiparty number-in-hand protocols can be useful outside of the realm of automatability. This model is weaker than the 2-party variant, hence proving lower bounds may be feasible. Is it easier to prove a superpolynomial lower bound on the size of randomized multiparty number-in-hand dag-like protocols than in a two-party case? It is still sufficient to get as a corollary a lower bound for Res($\oplus$) (resolution over parities).

2 Preliminaries

Notation. We use the standard notation $[n] = \{1, \dots, n\}$. Vars(φ) 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, 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 *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: $|\{i \in [k] \mid \text{Vars}(C) \cap U_i \neq \emptyset\}|$. block-width of a resolution refutation is the maximum block-width of a clause in i5t. For an unsatisfiable CNF φ we denote bw(φ) 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 $f_{0\text{-sink}} \equiv 0$, $f_{1\text{-sink}} \equiv 1$, $f_u(x) := \begin{cases} f_v(x) & x_i = 0 \\ f_w(x) & x_i = 1 \end{cases}$ 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 $x_{\pi(i_1)}, x_{\pi(i_2)}, \ldots, x_{\pi(i_k)}$ where $1 \le i_1 < i_2 < \cdots < i_k \le n$.

OBDD refutations. π-OBDD-refutation of a CNF formula φ is a sequence of π-OBDDs $D_1, \ldots, D_s$ 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. It is known 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: OBDD($\wedge$) where only the conjunction rule is allowed, OBDD($\wedge, \exists$) where the conjunction and the projection rules are allowed, and OBDD($\wedge$, 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 π-OBDD(φ) the size of the smallest π-OBDD($\wedge$, weakening) refutation of φ and by OBDD(φ) the size of the smallest OBDD($\wedge$, weakening) refutation of φ.

Proposition 2.1 ([4])**.** OBDD($\wedge, \exists$) *(and, thus,* OBDD($\wedge$, weakening)*) polynomially simulates resolution: if an unsatisfiable CNF has a resolution refutation of size S, then it has an* OBDD($\wedge, \exists$) *refutation of size* poly(S).

Search$_\varphi$. Search$_\varphi$ 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 2.2 ([31])**.** *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 its 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 $[n]$. A 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 2.3 ([14]). *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 2.4. *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 $S^{\mathcal{O}(k)}$.*

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 1.1

Our starting point is the following theorem that is essentially proved in [17].

Theorem 3.1 (Lemma 2.2 from [17]). *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 $\mathcal{O}(\log n)$-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 $|B_1| = |B_2| = \cdots = |B_k| = \mathcal{O}(n)$ and $k = \mathcal{O}(n^{c+1})$ 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 $|\pi| = n^{\mathcal{O}(c)}$ and $\mathrm{bw}(\pi) = \mathcal{O}(1)$ w.r.t. partition $B_1, \ldots, B_k$;*

- *if* $\varphi \notin$ 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 3.1 is slightly different from one explicitly stated in [17]. First, it is not stated that all blocks B_i have equal sizes and their sizes are $\mathcal{O}(n)$, but this is clear from the definition in Section 3.1 of [17]. 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 [17] for details).

To prove Theorem 1.1 we follow the plan below:

Lifting with multi-output indexing function In Section 4 we define a block-wise indexing function $\text{IND}_{\ell\times m}$ and its composition with relations and formulas. In Section 4.1 we will see that if a CNF formula φ has short resolution refutation of constant block-width then $\varphi \circ \text{IND}^n_{\ell\times m}$ 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 $\text{Search}_\varphi \circ \text{IND}^n_{\ell\times m}$ and consequently $\text{Search}_{\varphi\circ\text{IND}^n_{\ell\times m}}$ 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 [29] 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 1.1.

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 4.1 (Block-wise indexing, [16])**.** $\text{IND}_{\ell\times m} : [m] \times \{0,1\}^{\ell\times m} \to \{0,1\}^\ell$ *is defined as* $\text{IND}_{\ell\times m}(x, y) = (y_{1,x}, y_{2,x}, \ldots, y_{\ell,x})$ *i.e. given an index* $x \in [m]$ *and a matrix* $y \in \{0,1\}^{\ell\times m}$*, it returns the* x*th column of* y*. For a set* $R \subseteq [m]^n \times (\{0,1\}^{\ell\times m})^n$ *we define* $\text{IND}^n_{\ell\times m}(R) = \{(\text{IND}_{\ell\times m}(x_1, y_1), \ldots, \text{IND}_{\ell\times m}(x_n, y_n)) \in \{0,1\}^{n\ell} \mid (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n) \in R\}$.

4.1 Upper bound for lifted formula

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 $\psi = \varphi \circ \text{IND}^n_{\ell\times m}$. First let us define $C \circ \text{IND}^n_{\ell\times m}$ for a clause C. The resulting CNF formula will compute the function $C \circ \text{IND}^n_{\ell\times m} = C(\text{IND}_{\ell\times m}(x_1, y_1), \ldots, \text{IND}_{\ell\times m}(x_n, y_n))$. Then we define $\varphi \circ \text{IND}^n_{\ell\times m} := \bigwedge_{i=1}^t \left(C_i \circ \text{IND}^n_{\ell\times m}\right)$.

Now let us construct a CNF representation of $C \circ \text{IND}^n_{\ell\times m}$. Let $z_{i,j}$ for $i \in [n]$, $t \in [\ell]$ be the tth variable of the ith block of φ. Let $i_1, \ldots, i_b \in [n]$ be indices of the blocks that are touched by C and let C_j for $j \in [b]$ be the part of variables of C from the i_jth block: $C = C_1 \vee \cdots \vee C_b$. 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 $C \circ \text{IND}^n_{\ell\times m}(x_1, y_1, \ldots, x_n, y_n)$ 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 $\alpha_1, \ldots, \alpha_b \in [m]$.

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 4.2 (the last inequality in Theorem 4 from [16]). *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 $m^{\mathcal{O}(b)} \cdot s$.*

4.2 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_ℓ, 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 [16], but for box dags instead of simplex dags and, crucially, for a smaller number of parties, $\ell+1$ instead of $n\ell+1$.

Theorem 4.3. *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 $m^{\Omega(\mathrm{bw}(S))} \le \mathsf{box\text{-}dag}_{A,B_1,\dots,B_\ell}(S \circ \textsc{Ind}^n_{\ell\times m})$, where block partition of inputs of S is the natural partition into n blocks of size ℓ.*

Let us outline the proof of Theorem 4.3. 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 blockwidth $\mathcal{O}(\log|\mathbb{B}|/\log m)$. 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 [16], 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 [14]. In Subsection 4.3 we formulate the properties of structured rectangles that we need, in Subsection 4.4 we define and prove the analogous properties for structured boxes, in Subsection 4.5 we construct the decision dag solving S.

4.3 Structured Rectangles

Lifting theorems from [14] 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 4.4. *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 4.5. *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:*
   - $\mathcal{W}(R_i, \rho_i)$ *and* $|\mathrm{fix}(\rho_i)| = \mathcal{O}(k/\log n)$;
   - 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 4.6. *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 4.7 (Lemma 4.4, Lemma 4.5 from [14])**.** *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*;* is partitionable*;* respects largeness[2]*. We say that a rectangle* R *is* ρ-structured *iff* $\mathcal{W}(R, \rho)$ *holds.*

Although Lemma 4.4 of [14] is not stated in strong enough form to satisfy Definition 4.4, the needed property is actually proved in Section 9 of [14].

4.4 Structured Boxes

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

Definition 4.8. *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* i*th 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 4.4, 4.5, and 4.6.

Lemma 4.9. *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* $\textsc{Ind}^n_{\ell\times m}(\{x\} \times Y_1 \times \cdots \times Y_\ell) = \mathrm{Cube}(\rho)$.

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 4.4.

Now let us prove the second statement. By Definition 4.4 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

$$\Pr_{x\leftarrow\mathcal{U}(X)}\left[\textsc{Ind}^n_{\ell\times m}(\{x\} \times Y_1 \times Y_2 \times \cdots \times Y_\ell) \neq \mathrm{Cube}(\rho)\right] \leq$$

$$\sum_{i=1}^{\ell} \Pr_{x\leftarrow\mathcal{U}(X)}\left[\textsc{Ind}^n_{1\times m}(\{x\} \times Y_i) \neq \mathrm{Cube}(\rho_i)\right] \leq \frac{2\ell}{n} < 1.$$

□

[1] This property is not explicitly stated in [14], 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] This property is implicit in [14], see Appendix A for details.

Lemma 4.10. *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 4.6 we have that each of the $X \times Y_i$ is $*^n$-structured which by definition implies $*^{n\ell}$-structuredness of R. □

Lemma 4.11. *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 $\mathcal{O}(k/\log n)$ 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 $\mathcal{O}(k/\log n)$ blocks (i.e. it either contains all the indices from a block or none) such that $x \in X^i$ implies $\mathrm{fix}(\rho^i) \subseteq I_x$.

Proof. Consider the partition $X = \bigsqcup_{j \in J} \tilde{X}_j$ from Definition 4.5 and a family $\{F_u\}_{u \in U}$, $F_u \subseteq [n]$ (both do not depend on Y). For each $t \in [\ell]$ apply the partitionable property to $X \times Y_t$: let $X \times Y_t = \bigsqcup_i R_i^{(t)}$ be the partition and let $\bar{X}_t^{err}$, $\bar{Y}_t^{err}$ be the error sets for Definition 4.5. Set $X^{err} := \bigcup_{t=1}^{\ell} \bar{X}_t^{err}$, $Y_t^{err} := \bar{Y}_t^{err}$. It is easy to see that property (1) holds.

Now let us construct the partition $\{R_i\}_{i \in I}$ of R:

- for each $\tilde{X}_j$ and $t \in [\ell]$ let $\mathcal{Z}_{j,t}$ be the set of all rectangles from the partition of $X \times Y_t$ that have projection $\tilde{X}_j$ onto X, in other words $\mathcal{Z}_{j,t} := \{Z \subseteq Y_t \mid \exists i \colon \tilde{X}_j \times Z = R_i^{(t)}\}$;
- let $\{R_i\}_{i \in I}$ consist of boxes $\tilde{X}_j \times Z_1 \times \cdots \times Z_\ell$ for all $j \in J$, $Z_1 \in \mathcal{Z}_{j,1}$, $Z_2 \in \mathcal{Z}_{j,2}$, …, $Z_\ell \in \mathcal{Z}_{j,\ell}$.

First we need to show that $\{R_i\}_{i \in I}$ is indeed a partition of $X \times Y_1 \times \cdots \times Y_\ell$. Consider an arbitrary point $(x, y_1, \ldots, y_\ell) \in X \times Y_1 \times \cdots \times Y_\ell$. Let $\tilde{X}_j$ be the unique part of $\{\tilde{X}_j\}_{j \in J}$ containing x, and for each $t \in [\ell]$ let Z_t be the unique element of $\mathcal{Z}_{j,t}$ containing y_t. Then $\tilde{X}_j \times Z_1 \times \cdots \times Z_\ell$ is the unique element of $\{R_i\}_{i \in I}$ containing $(x, y_1, \ldots, y_\ell)$.

We now proceed to verify the property (2). Consider $i \in I$ and let $R_i := X^i \times Y_1^i \times Y_2^i \times \cdots \times Y_\ell^i$. If it is not covered by error sets, then each of the rectangles $X^i \times Y_t^i$ is not covered by the error sets $\bar{X}_t^{err}$ and $\bar{Y}_t^{err}$. Thus, all of these rectangles are $\rho^{i,t}$-structured for some assignment $\rho^{i,t} \in \{0, 1, *\}^n$ such that $|\mathrm{fix}(\rho^{i,t})| = \mathcal{O}(k/\log n)$. Let $\rho^i \in \{0, 1, *\}^{n\ell}$ be constructed from $\rho^{i,1}, \ldots, \rho^{i,\ell}$ as follows $\rho^i_{(j-1)\ell+t} := \rho^{i,t}_j$. By Definition 4.5 for each $i \in I$ there exists $u(i) \in U$ such that $\mathrm{fix}(\rho^{i,t}) = F_{u(i)}$ for each $t \in [\ell]$. Thus, ρ^i assigns variables from $|F_{u(i)}|$ blocks. By Definition 4.5, $|F_{u(i)}| = \mathcal{O}(k/\log n)$. The property (3) holds because $u(i) = u(i')$ whenever $X^i = X^{i'}$, i.e., it is the same for all boxes with fixed projection onto X. □

4.5 Proof of Theorem 4.3

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 $\mathrm{bw}(S) = \mathcal{O}\left(\log \textsf{box-dag}_{A,B_1,\ldots,B_\ell}(S \circ \textsc{Ind}^n_{\ell \times m})/\log m\right)$.

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 $\mathcal{O}(\log |\mathbb{B}|/\log m) = \mathcal{O}(\log |\mathbb{B}|/\log n)$.

Similarly to [14] we first assume that partitions yielded by Lemma 4.11 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 [14] more or less verbatim so we place it in the Appendix 4.6. We

apply Lemma 4.11 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 4.11 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 $\rho^u_1, \ldots, \rho^u_{t_u}$ be the assignments yielded by the partition of the box u, $\rho^{v_q}_1, \ldots, \rho^{v_q}_{t_{v_q}}$ 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 $\rho^u_1, \ldots, \rho^u_{t_u}$, leaves $\rho^{v_q}_1, \ldots, \rho^{v_q}_{t_{v_q}}$ for $q \in \{1, 2\}$ that satisfies the local condition of a decision dag having block-width $\mathcal{O}(k/\log n)$.

Proof of 1. By Lemma 4.10 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 $B = X \times Y_1 \times \cdots \times Y_\ell$ is a ρ-structured box in the partition of u. By Lemma 4.9 there exists x_0 such that $\textsc{Ind}^n_{\ell\times m}(\{x_0\} \times Y_1 \times \cdots \times Y_\ell) = \mathrm{Cube}(\rho)$, i.e. for every $\alpha \in \mathrm{Cube}(\rho)$ there exist $y_1, \ldots, y_\ell$ such that $(x_0, y_1, \ldots, y_\ell) \in B$ and $\textsc{Ind}^n_{\ell\times m}(x_0, y_1, \ldots, y_\ell) = \alpha$. 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 $\mathcal{O}(k/\log n)$.

Recall that we abuse notation by identifying nodes of a box dag with their underlying boxes. Let $B = X \times Y_1 \times \cdots \times Y_\ell$ be a ρ^u_i-structured box from the partition of u. And let $x \in X$ be such that $\textsc{Ind}^n_{\ell\times m}(\{x\} \times Y_1 \times \cdots \times Y_\ell) = \mathrm{Cube}(\rho^u_i)$. By the property of a box-dag, B is covered by the union of boxes v_1 and v_2. Thus $\{x\} \times Y_1 \times \cdots \times Y_\ell$ 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 4.11. 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 4.11, $I^{v_1}_x$ and $I^{v_2}_x$ are unions of $\mathcal{O}(k/\log n)$ blocks and $\mathrm{fix}(\rho^u_i)$ touches $\mathcal{O}(k/\log n)$ blocks. Thus block-width of the resulting dag is also $\mathcal{O}(k/\log n)$. It is sufficient to show that for any leaf $\theta \in \{0, 1, *\}^{n\ell}$ of the decision tree we can find a node ω in $\mathcal{L}$ such that $\mathrm{Cube}(\omega) \supseteq \mathrm{Cube}(\theta)$. Then we can direct both edges from ω to θ.

Consider any leaf of the decision tree $\theta \in \{0, 1, *\}^{n\ell}$. Since $\textsc{Ind}^n_{\ell\times m}(\{x\} \times Y_1 \times \cdots \times Y_\ell) = \mathrm{Cube}(\rho^u_i)$ and θ extends ρ^u_i (i.e., $\mathrm{Cube}(\theta) \subseteq \mathrm{Cube}(\rho^u_i)$), there exist $y_1 \in Y_1, \ldots, y_\ell \in Y_\ell$ such that $\textsc{Ind}^n_{\ell\times m}(x, y_1, \ldots, y_\ell) \in \mathrm{Cube}(\theta)$. Then consider an ω-structured box B_0 from a partition of v_1 or v_2 for $\omega \in \mathcal{L}$ that contains $(x, y_1, \ldots, y_\ell)$. 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 4.11, the second holds by the construction of the decision tree. Since by Lemma 4.9, $\textsc{Ind}^n_{\ell\times m}(x, y_1, \ldots, y_\ell) \in \mathrm{Cube}(\omega)$, $\mathrm{Cube}(\omega)$ and $\mathrm{Cube}(\theta)$ have a point in common, then $\mathrm{fix}(\omega) \subseteq \mathrm{fix}(\theta)$ implies $\mathrm{Cube}(\omega) \supseteq \mathrm{Cube}(\theta)$. That finishes the proof under the errorless assumption.

4.6 Getting rid of the errorless assumption in Theorem 4.3

This section repeats the argument from Section 5.3 in [14] more or less verbatim.

Instead of constructing the decision dag from top to bottom we process the boxes in $\mathbb{B}$ in a reverse topological order i.e. from leaves to the root. This guarantees that whenever a box u is processed, its two

descendants v and w have already been processed. Next we describe the process of removing error sets from the boxes.

First, initialize cumulative error sets $\tilde{X}^{err}, \tilde{Y}_1^{err}, \ldots, \tilde{Y}_\ell^{err} := \emptyset$. Then for each box $B_i = X^i \times Y_1^i \times \cdots \times Y_\ell^i \in \mathbb{B}$ in the reverse topological order we do the following

1. If B_i is not a leaf remove all points with error coordinates from the box:

$$\bar{B}_i := B_i \cap \left((\mathcal{A} \setminus \tilde{X}^{err}) \times \prod_{j=1}^{\ell} \left(\mathcal{B}_j \setminus \tilde{Y}_j^{err} \right) \right);$$

   otherwise $\bar{B}_i := B_i$.

2. Let $X^{err}, Y_1^{err}, \ldots, Y_\ell^{err}$ be the error sets yielded by Lemma 4.11 applied to $\bar{B}_i$.

3. Update $\tilde{X}^{err} := \tilde{X}^{err} \cup X^{err}$ and $\tilde{Y}_j^{err} := \tilde{Y}_j^{err} \cup Y_j^{err}$ for $j \in [\ell]$.

We refer to $\bar{B}_i$ as *cleaned boxes*. Then we apply the decision dag construction from Section 4.5 to the partitions of cleaned boxes by Lemma 4.11 where all non-structured boxes in the partition (those covered by the error sets) are removed. We refer to this partitions as *cleaned partitions*. We need to show that the analogues of conditions 1-3 from Section 4.5 still hold:

1. The cleaned partition of the cleaned root box 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 cleaned 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 $\rho_1^u, \ldots, \rho_{t_u}^u$ be the assignments yielded by the cleaned partition of the box u, $\rho_1^{v_q}, \ldots, \rho_{t_{v_q}}^{v_q}$ be the assignments yielded by the cleaned partition of the box v_q for $q \in \{1, 2\}$. Then there exists an assignment-labeled dag with sources $\rho_1^u, \ldots, \rho_{t_u}^u$, leaves $\rho_1^{v_q}, \ldots, \rho_{t_{v_q}}^{v_q}$ for $q \in \{1, 2\}$ that satisfies the local condition of a decision dag having blockwidth $\mathcal{O}(k/\log n)$.

Now is time to choose k. If $|\mathbb{B}| \geq n^n/100$, Theorem 4.3 holds since $\mathrm{bw}(S) = \mathcal{O}(n)$ and $m = n^{\mathcal{O}(1)}$, so we may assume that $\log_2 |\mathbb{B}| < n \log_2 n - \log_2 100$. Now we fix $k = \log_2(100|\mathbb{B}|) \leq n \log_2 n$.

Proof of 1. Since $k = \log_2(100|\mathbb{B}|)$, $|\tilde{X}^{err}| \leq m^n \cdot 2^{-k} \cdot |\mathbb{B}| = \frac{m^n}{100}$; $|\tilde{Y}_1^{err}|, \ldots, |\tilde{Y}_\ell^{err}| \leq 2^{mn} \cdot 2^{-k} \cdot |\mathbb{B}| = \frac{2^{mn}}{100}$. Then the cleaned root box satisfies that conditions of Lemma 4.10, thus the cleaned partition of the root consists of a single $*^{n\ell}$-structured box.

Proof of 2. The proof stays the same since $\bar{B}_i = B_i$ for every leaf.

Proof of 3. Let $\tilde{X}^{err,u}, \tilde{Y}_1^{err,u}, \ldots, \tilde{Y}_\ell^{err,u}$ be the error sets at step (2) of the process for a node u of $\mathbb{B}$. Then for a node v and its children w_1, w_2 in $\mathbb{B}$, $\tilde{X}^{err,w_i} \subseteq \tilde{X}^{err,v}$, $\tilde{Y}_j^{err,w_i} \subseteq \tilde{Y}_j^{err,v}$ for each $i \in \{1, 2\}$, $j \in [\ell]$. Recall that the proof of statement 3 with errorless assumption consisted of finding a structured box among the structured boxes of the partitions of w_1 and w_2 covering an arbitrary point in a structured box of the partition of u. Since the cleaned part of u does not intersect error sets of w_1 and w_2, any its point is covered by a structured box from w_1 or w_2 which is sufficient for us. Therefore, the proof with errorless assumption works without any changes.

5 Making box bags out of π-OBDD refutations

5.1 Number-in-hand multiparty communication

Let us define number-in-hand k party communication protocols. These protocols naturally extend classical 2-party communication (for the formal definition of 2-party protocols we refer to [24]). Let $S \subseteq \{0, 1\}^{U_1} \times$

$\{0,1\}^{U_2} \times \cdots \times \{0,1\}^{U_k} \times Y$ be a relation, where $U_1, \dots U_k$ is a partition of $[n]$. In a number-in-hand k-party communication protocol ith party receives an element $x_i \in \{0,1\}^{U_i}$. Then all of these parties take turns broadcasting 1-bit messages (which depend only on their input and the previously broadcasted messages) to all other parties. In the end of the protocol all parties must agree on an element $y \in Y$ such that $(x_1, \dots, x_k, y) \in S$. The cost of the protocol is the maximum number of bits exchanged by the parties.

First we give a formal definition of a number-in-hand protocol. Formally a protocol is a rooted binary tree where each inner node s is labeled with a function $\mathrm{dir}_s : \{0,1\}^{U_i} \to \{0,1\}$ for some $i \in [k]$, the edges outgoing of each node are labeled with 0 and 1, and the leaves are labeled with elements of Y. Each node of the binary tree corresponds to a possible history of communication between the parties, the function $\mathrm{dir}_s : \{0,1\}^{U_i} \to \{0,1\}$ corresponds to the decision of the ith party in this configuration. Then each element $x \in \{0,1\}^n$ uniquely determines a leaf of the protocol tree: let us descend from the root each time going along the edge labeled with $\mathrm{dir}_s(x|_{U_i})$, where s is the current node. We say that a protocol computes S if for each input $x \in \{0,1\}^n$ the leaf corresponding to it is labeled with $y \in Y$ such that $(x, y) \in S$.

As for the classical 2-party communication a number-in-hand communication protocol can be viewed as a box tree which is a special case of a $\mathsf{box\text{-}dag}_{U_1,\dots,U_k}$. This tree has the same topology as the tree from the definition of the protocol. Let us label each node s of this tree with a set U_s which contains all points $x \in \{0,1\}^n$ such that s lies on the path between the root and the leaf uniquely determined by x. By induction from top to bottom one can show that all such labels are actually $(U_1, \dots, U_k)$-boxes. Moreover, if for a node u with 0-successor v and 1-successor w, if u is labeled with $X_1 \times \cdots \times X_k$, then v is labeled with $X_1 \times \cdots \times X_{i-1} \times (\mathrm{dir}_u^{-1}(0) \cap X_i) \times X_{i+1} \times \cdots \times X_k$ and w is labeled with $X_1 \times \cdots \times X_{i-1} \times (\mathrm{dir}_u^{-1}(1) \cap X_i) \times X_{i+1} \times \cdots \times X_k$.

Lemma 5.1. *[a generalization of a similar lemma in [31]] 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 of cost c w.r.t. partition $U_1, \dots, U_k$ of $[n]$. Let $S \subseteq \{0,1\}^{U_1} \times \cdots \times \{0,1\}^{U_k} \times Y$ 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 $\mathcal{O}(|D| \cdot 2^{3c})$ that solves S.*

Proof. First, observe that for every node u in D, $f_u^{-1}(0)$ can be represented as a disjoint union of at most 2^c boxes corresponding to the leaves of the communication protocol solving f_u. D' will consist of the boxes appearing in the representations of $f_u^{-1}(0)$ for all $u \in D$ together with auxiliary nodes connecting these representations to each other. If r is a root of D, then $f_r \equiv 0$; so its representation as a union of boxes contains the only box $\{0,1\}^{U_1} \times \dots \{0,1\}^{U_k} = \{0,1\}^n$ which is the root of D'. For each leaf ℓ of D labeled with y we label all boxes in the representation of $f_\ell^{-1}(0)$ with y in D'.

Now in order to complete the construction we need to connect the boxes from the disjoint representations to each other such that the local condition is satisfied for each inner node u and its successors v and w. In order to do this we add auxiliary boxes to the dag.

Let a be an inner node in D and let b and c be its successors. Let $A_1, \dots, A_n$, $B_1, \dots, B_m$ and $C_1, \dots, C_m$ be the boxes in their respective representations. For each A_i let us copy the box-tree for f_b, where all the boxes are replaced with their intersections with A_i, and root it in A_i. All the empty nodes labeled with empty boxes are removed from the dag, the nodes that lose one of the successors are then replaced with the remaining successor. For all the leaves of the box-tree for f_b labeled with 0 we can find a box B_j that contains the box in the leaf. Then we redirect all edges to such leaves of the box-tree to a suitable B_j. For all the leaves labeled with 1 we root a copy of a box-tree for f_c with the boxes replaced with their intersections with the box in the leaf (removing empty boxes as before). By the local property of $\mathcal{F}$-dag all the leaves of the box-tree for f_c labeled with 1 become empty after the intersection with the box in the leaf of the box-tree for f_b. Then we can find C_j containing the box in the leaf of the box-tree of f_c; we then redirect edges to such leaf to C_j. The number of nodes added to D' per a single box in a disjoint box representation of a node is at most 2^{2c}. Since there are at most 2^c boxes in each of these representations $|D'| \leq |D| 2^{3c}$ as required. □

5.2 From box dags to π-OBDDs

Let X be a set of propositional variables of size n, $\mathcal{V} := (V_1, V_2, \ldots, V_k)$ be a partition of X: $X = V_1 \sqcup \cdots \sqcup V_k$, 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 5.2. *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 $D_1, \ldots, D_t$ be a π-OBDD($\wedge$, weakening) refutation of φ of size S. Then $\mathsf{box\text{-}dag}_{\mathcal{V}}(\mathrm{Search}_\varphi) \leq S^{\mathcal{O}(k)}$.*

First, let us prove the following lemma.

Lemma 5.3. *Let D be a π-OBDD over variables X computing a function f and $\mathcal{V} = (V_1, \ldots, V_k)$ 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. Let the communicating parties put a pebble in the source of D and move it according to the input bits. First, the first party moves the pebble, then the second one does, and so on. If the current party does not know the variable queried in the current node, they broadcast the index of the node where the pebble is, and then the next party proceeds to move it. Since π agrees with $\mathcal{V}$ there will be at most k broadcasts. The last broadcast is the index of a sink so all the parties know the value of f after the protocol. □

Proof of Theorem 5.2. By Lemma 5.3, 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 5.1, there exists a $\mathsf{box\text{-}dag}_{\mathcal{V}}$ of size $S \cdot 2^{3k \log S} = S^{\mathcal{O}(k)}$ 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 6.1 ([30])**.** *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$(\varphi) \leq$ OBDD$(\mathcal{T}_0(\varphi))$ (Lemma 14 from [30]);*
- *If φ has a resolution refutation of size s and* negative width *w, then $\mathcal{T}_0(\varphi)$ has resolution size at most $s \cdot n^{\mathcal{O}(w)}$, (Corollary 9 and Lemma 12 from [30]).*

Lemma 6.2. *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.*

Proof. The construction of φ' is the following. Let $x_1, \ldots, x_n$ be the variables of φ. The variables of φ' are $x_1, \ldots, x_n, y_1, \ldots, y_n$. For each clause $C = \bigvee_{i \in P_C} x_i \vee \bigvee_{i \in N_C} \neg x_i$ of φ we add a clause $C' = \bigvee_{i \in P_C} x_i \vee \bigvee_{i \in N_C} y_i$ to φ' and for each $i \in [n]$ add clauses $x_i \vee y_i$ and $\neg x_i \vee \neg y_i$ to φ'. Let P be the function mapping C to C'.

First, we transform a resolution refutation $C_1, C_2, \ldots, C_s$ of φ into a resolution refutation of φ' with a constant negative width. Let us show how to derive clauses $\mathsf{P}(C_1), \mathsf{P}(C_2), \ldots, \mathsf{P}(C_s)$ from the clauses of φ'. By induction on i we prove that the clauses $\mathsf{P}(C_1), \ldots, \mathsf{P}(C_i)$ can be derived from the clauses of φ' in $2i$ steps with negative width 1. The base case is trivial. By the induction hypothesis the clauses $\mathsf{P}(C_1), \ldots, \mathsf{P}(C_{i-1})$ can be derived in $2i-2$ applications of resolution. Suppose that $C_i = A \vee B$ and it is derived from $C_j = A \vee x_\ell$

and $C_k = B \vee \neg x_\ell$. Then $\mathsf{P}(C_j) = \mathsf{P}(A) \vee x_\ell$ and $\mathsf{P}(C_k) = \mathsf{P}(B) \vee y_\ell$. First let us resolve $\mathsf{P}(C_k)$ and $\neg y_\ell \vee \neg x_\ell$ on y_ℓ getting $\mathsf{P}(B) \vee \neg x_\ell$ and then resolve it with $\mathsf{P}(A)$ on x_ℓ getting $\mathsf{P}(A) \vee \mathsf{P}(B) = \mathsf{P}(A \vee B) = \mathsf{P}(C_i)$. Observe that the negative width of the resulting refutation is 1.

W.l.o.g. we may assume that π orders the variables as $x_1, x_2, \ldots, x_n$. Consider an order π' ordering the variables as $x_1, y_1, x_2, y_2, \ldots, x_n, y_n$. We claim that a π'-OBDD refutation of φ' of size t can be transformed into a π-OBDD refutation of φ of size at most t. Let us apply a substitution $y_1 := \neg x_1; y_2 := \neg x_2; \ldots; y_n := \neg x_n$ to each of the π'-OBDDs in the refutation of φ'. Then observe that the resulting refutation is a π-OBDD refutation of φ: the OBDDs corresponding to the clauses of form $x_i \vee y_1$ or $\neg x_i \vee \neg y_i$ compute identical truth after the substitution which allows us to remove them from the refutation; it is easy to see that the conjunction rule and the weakening rule are preserved after the substitution.

Then it is sufficient to show that the size of π'-OBDD does not grow after the substitution $y_i := \neg x_i$ for any $i \in [n]$. The diagram after the substitution can be obtained from the initial one as follows:

- replace the label of each node querying y_i with x_i and swap the labels of the edges going out of it; now it is possible that two consecutive nodes in a path query x_i;
- for each α-labeled edge that goes from a node u labeled with x_i to a node v labeled with x_i redirect it to the endpoint of the α-labeled edge going out of v.

Clearly after this transformation the number of nodes does not increase and the OBDD after the transformation computes the function after the substitution. □

Corollary 6.3. *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$(\mathcal{T}(\varphi))$;*
- *If φ has a resolution refutation of size s, then the resolution size of $\mathcal{T}(\varphi)$ is at most $s \cdot n^{\mathcal{O}(1)}$.*

Proof. The new algorithm $\mathcal{T}$ first applies the transformation from Lemma 6.2 to a CNF formula and only then applies the algorithm $\mathcal{T}_0$ from Theorem 6.1 to it. □

7 Putting the pieces together

Theorem 1.1. *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*

- *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$(\wedge, \text{weakening})$ *refutation of $\mathcal{R}(\varphi)$ has size $2^{\Omega(n)}$.*

Proof. Let $\mathcal{E}$ be the algorithm from Theorem 3.1 with the parameter $c = 3$, and $\mathcal{T}$ be the algorithm from Corollary 6.3. 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 3.1, $\ell_\varphi = \mathcal{O}(n)$. Then let $m_\varphi = (n_\varphi \ell_\varphi)^\Delta$ where Δ is from Theorem 4.3 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 $\varphi \in$ SAT. Then by Theorem 3.1, $\mathcal{E}(\varphi)$ has a resolution refutation π such that $|\pi| = |\varphi|^{\mathcal{O}(1)}$ and $\mathrm{bw}(\pi) = \mathcal{O}(1)$. Then applying Theorem 4.2 we get that there exists a resolution refutation of $\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi}$ of size $|\varphi|^{\mathcal{O}(1)}$. Then by Corollary 6.3 $\mathcal{T}(\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi})$ has a resolution refutation of size $|\varphi|^{\mathcal{O}(1)}$.

Let us proceed with the case $\varphi \notin$ SAT. Suppose $\mathcal{R}(\varphi)$ has a OBDD$(\wedge, \text{weakening})$ refutation of size S. Then by Corollary 6.3 the formula $\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi}$ has a π-OBDD$(\wedge, \text{weakening})$ 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-OBDD($\wedge$, weakening) refutation of $\mathcal{E}(\varphi) \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi \times m_\varphi}$ is at most S which by Theorem 5.2 implies that $\textsf{box-dag}_{A,B_1,\ldots,B_\ell}\left(\text{Search}_{\mathcal{E}(\varphi)\circ\textsc{Ind}^{n_\varphi}_{\ell_\varphi\times m_\varphi}}\right) \leq S^{\mathcal{O}(\ell_\varphi+1)}$.

Then the fact that $\text{Search}_{\mathcal{E}(\varphi)\circ\textsc{Ind}^{n_\varphi}_{\ell_\varphi\times m_\varphi}}$ is at least as hard as $\text{Search}_{\mathcal{E}(\varphi)} \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi\times m_\varphi}$ and the inequality $\textsf{box-dag}_{A,B_1,\ldots,B_\ell}\left(\text{Search}_{\mathcal{E}(\varphi)} \circ \textsc{Ind}^{n_\varphi}_{\ell_\varphi\times m_\varphi}\right) \geq m_\varphi^{\Omega(\text{bw}(\mathcal{E}(\varphi)))}$ implied by Theorem 4.3 together imply that $S \geq m_\varphi^{\Omega(\text{bw}(\mathcal{E}(\varphi))/(\ell_\varphi+1))}$. By Theorem 3.1 using Proposition 2.3 to switch from decision dag to resolution refutation we have $\text{bw}(\mathcal{E}(\varphi)) = \Omega(n^{c-1}) = \Omega(n^2)$ which implies that $S \geq m_\varphi^{\Omega(n)}$ since $\ell_\varphi = \mathcal{O}(n)$. This completes the proof of the theorem since $m_\varphi \geq 2$. □

Corollary 7.1. *If* OBDD($\wedge$, weakening) *is automatable then* P = NP.

Proof. Let $\mathcal{X}$ be an algorithm that automates OBDD($\wedge$, weakening). Given an unsatisfiable CNF φ the algorithm $\mathcal{X}$ runs in time $\mathcal{O}((|\varphi| + |\pi|)^k)$, where π is the shortest OBDD($\wedge$, weakening)-refutation of φ.

The algorithm solving SAT with this assumption is the following: Consider the following algorithm solving 3-SAT: given φ, run $\mathcal{X}$ on $\mathcal{R}(\varphi)$ for $n^{k\alpha+1}$ steps. If in that time it finds a refutation of $\mathcal{R}(\varphi)$, then $\varphi \in$ SAT, because if $\varphi \notin$ SAT the second part of Theorem 1.1 implies that $\mathcal{R}(\varphi)$ does not have polynomial size OBDD($\wedge$, weakening) refutations. If $\mathcal{X}$ does not find a refutation in the given time, then there is no refutation of size n^α which implies that $\varphi \notin$ SAT by the first part of Theorem 1.1. □

Acknowledgements

The authors thank Anastasia Sofronova and Michal Garlík 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 [14].

Statement on the situation in Ukraine

According to the military censorship law in Russia, the authors can't publicly state that they strongly condemn the aggressive war started by Vladimir Putin and the war crimes committed on Ukraine's territory. The authors can only express their condolences to all the victims, their relatives, and close friends.

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.

[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.

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

[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.

[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.

[6] Albert Atserias and Moritz Müller. Automating resolution is np-hard. *J. ACM*, 67(5):31:1–31:17, 2020.

[7] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 274–282. IEEE Computer Society, 1996.

[8] 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.

[9] 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.

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

[11] Sam Buss, Dmitry Itsykson, Alexander Knop, Artur Riazanov, and Dmitry Sokolov. Lower bounds on obdd proofs with several orders. *ACM Trans. Comput. Logic*, 22(4), sep 2021.

[12] 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.

[13] Susanna F. de Rezende. Automating tree-like resolution in time $n^{o(\log n)}$ is eth-hard. In Carlos E. Ferreira, Orlando Lee, and Flávio Keidi Miyazawa, editors, *Proceedings of the XI Latin and American Algorithms, Graphs and Optimization Symposium, LAGOS 2021, Online Event / São Paulo, Brazil, May 2021*, volume 195 of *Procedia Computer Science*, pages 152–162. Elsevier, 2021.

[14] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:175, 2017.

[15] Michal Garlík. Failure of feasible disjunction property for k-dnf resolution and np-hardness of automating it. *Electron. Colloquium Comput. Complex.*, page 37, 2020.

[16] 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, page 68–77, New York, NY, USA, 2020. Association for Computing Machinery.

[17] 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.

[18] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. *SIAM Journal on Computing*, 49(4):132–143, 2020. Preliminary version: FOCS'17.

[19] 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.

[20] Dmitry Itsykson and Dmitry Sokolov. Resolution over linear equations modulo two. *Ann. Pure Appl. Log.*, 171(1), 2020.

[21] 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.

[22] 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.

[23] Jan Krajícek and Pavel Pudlák. Some consequences of cryptographical conjectures for s^1_2 and EF. *Inf. Comput.*, 140(1):82–94, 1998.

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

[25] 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.

[26] Guoqiang Pan and Moshe Y. Vardi. Symbolic techniques in satisfiability solving. *Journal of Automated Reasoning*, 35(1-3):25–50, 2005.

[27] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997.

[28] 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.

[29] 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.

[30] Nathan Segerlind. On the relative efficiency of resolution-like proofs and ordered binary decision diagram proofs. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 100–111, 2008.

[31] 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.

A ρ-structuredness respects largeness

In this section we verify that the ρ-structuredness property of rectangles from [14] does actually conform to Definition 4.6.

First let us introduce the definition of ρ-structuredness from [14].

For a random variable $\mathbf{x}$ its *min-entropy* as $H_\infty(\mathbf{x}) = \log \min_x \frac{1}{\Pr[x=\mathbf{x}]}$. For a random variable $\mathbf{x}$ taking values from a set C^n and for $I \subseteq [n]$ let $\mathbf{x}_I$ be the marginal distribution of $\mathbf{x}$ on the coordinates from I.

Definition A.1 (Definition 4.2 in [14], introduced in [18])**.** *A rectangle $R := X \times Y \subseteq [m]^n \times \{0,1\}^{mn}$ is* ρ-structured*, where $\rho \in \{0,1,*\}^n$ if*

1. *the random variable $\mathcal{U}(X)_{\mathrm{fix}(\rho)}$ has one-point support, and every $z \in \mathrm{IND}^n_{1\times m}(R)$ is consistent with ρ, that is, $\mathrm{IND}^n_{1\times m}(R) \subseteq \mathrm{Cube}(\rho)$;*

2. $I \subseteq [n] \setminus \mathrm{fix}(\rho)$, $H_\infty(\mathcal{U}(X)_I) \geq 0.9 \log |[m]^I| = 0.9|I| \log m$;

3. Y *is large enough:* $H_\infty(\mathcal{U}(Y)) \geq mn - n^3$.

Proposition A.2. *For large enough m any rectangle $R = X \times Y$ where $|X| \geq m^n \cdot 0.99$ and $|Y| \geq 2^{mn} \cdot 0.99$ is $*^n$-structured.*

Proof. Since $H_\infty(\mathcal{U}(Y)) = \log |Y| = mn + \log 0.99 > mn - n^3$, the property 3 is satisfied. The property 1 is trivially satisfied since $\mathrm{Cube}(*^n) = \{0,1\}^n$.

Suppose the property 2 is violated for a non-empty set I (for the empty set it is always true). Let $x \in \{0,1\}^I$ be the witness of the violation:

$$\Pr_{\mathbf{x} \leftarrow \mathcal{U}(X)_I}[x = \mathbf{x}] > m^{-0.9|I|}.$$

This implies that $|X_I| < m^{0.9|I|}$ then $|X| < m^{n-|I|+0.9|I|} = \frac{m^n}{m^{0.1|I|}} \leq \frac{m^n}{m^{0.1}}$. For $m^{0.1} > 100$ this contradicts the assumption that $|X| \geq 0.99 m^n$. □

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il