Breaking Through the Reordering Obstacle in OBDD Proof Systems

Sam Buss[3], Dmitry Itsykson[2], Alexander Knop[2, 3], and Dmitry Sokolov[1]

[1]KTH Royal Institute of Technology
[2]St. Petersburg Department of V.A. Steklov Institute of Mathematics of the Russian Academy of Sciences
[3]University of California, San Diego
sbuss@ucsd.edu, dmitrits@pdmi.ras.ru, aknop@ucsd.edu, sokolovd@kth.se

October 19, 2018

Abstract

A refutation of a CNF formula ϕ in 1-NBP($\wedge$)-calculus is a sequence of nondeterministic read-once branching programs $D_1, D_2, \ldots, D_s$ such that for every i, D_i represents either a clause of ϕ or the conjunction of two 1-NBPs with smaller numbers, and D_s represents the identically false function. We show that Tseitin formulas and Perfect matching principle based on algebraic expanders with small enough second eigenvalue require 1-NBP($\wedge$) refutations of size at least $2^{\Omega(n)}$, where n is the number of variables. Derivations in the proof system OBDD($\wedge$, reordering) proposed by Itsykson et al. [10] are partial cases of 1-NBP($\wedge$) refutations. There is a polynomial size 1-NBP($\wedge$) refutation of the pigeonhole principle, hence 1-NBP($\wedge$) is strictly stronger than OBDD($\wedge$, reordering). We also present a family of formulas that has polynomial size tree-like resolution refutation but requires 1-NBP($\wedge$) refutations of size at least $2^{\Omega(n)}$. As a corollary, we get that the poof system OBDD($\wedge$, reordering) does not simulate tree-like resolution, this resolves the open question formulated in the paper of Buss et al. [6].

We consider the proof system OBDD($\wedge$, weakening, reordering$_\ell$) that is a subsystem of the proof system OBDD($\wedge$, weakening, reordering), where proofs may contain OBDDs in at most ℓ different orders. For some constant C we show an exponential lower bound on the complexity of tree-like derivations in the proof system OBDD($\wedge$, weakening, reordering$_k$) for $\ell = C \log n$, where n is the number of variables.

1 Introduction

The proof system OBDD($\wedge$, weakening) was introduced in 2004 by Atserias et al. [2]. A refutation of an unsatisfiable CNF formula ϕ in in this proof system is a sequence $D_1, \ldots, D_\ell$ of OBDDs in some order π such that D_ℓ represents the constant false function and for each $i \in [\ell]$, D_i either represents a clause of the formula ϕ or is derived using one of the following rules.

Conjunction (or join) rule: D_i represents the conjunction of D_j and D_k for $j, k < i$.

Weakening rule: there is $j < i$ such that D_j semantically implies D_i. I.e. every assignment that satisfies D_j also satisfies D_i.

The proof system OBDD($\wedge$, weakening) polynomially simulates CP* (cutting planes with polynomially bounded coefficients) since linear inequalities with polynomially bounded coefficients can be efficiently represented by OBDDs. In 1994 Pudlak [16] showed that Clique–Coloring principle is hard for cutting planes; Buss at el. recently showed that there is a polynomial sized OBDD($\wedge$, weakening)-proof of the Clique–Coloring principle [6]. Thus, OBDD($\wedge$, weakening) is strictly stronger than CP*.

Though OBDD($\wedge$, weakening) is a powerful proof system, there are known exponential lower bounds on size of OBDD($\wedge$, weakening)-refutations: in 2007 Segerlind [17] proved an exponential lower bound on size of tree-like OBDD($\wedge$, weakening)-refutations and in 2008 Krajíček [13] proved an exponential lower bound on size of dag-like refutations. Both their proofs consist of two steps.

1. The first step of the proof is the construction of a family of formulas ϕ_n and orders π_n such that any tree-like (dag-like) OBDD($\wedge$, weakening)-refutations of ϕ_n in the order π_n has exponential size. This step may be accomplished by two general techniques: interpolation (a reduction from monotone circuit lower bounds) in the dag-like case [12] and communication complexity (a reduction from the lower bound on the randomised communication complexity of disjointness) in the tree-like case [3]. Both techniques may be applied to semantic proof systems which operate with proof lines that can be computed with small (randomised in tree-like case and deterministic in dag-like case) communication if values of some specific variables are known by Alice and values of other variables are known by Bob. It is well known that if an OBDD D in order π has size S, then it can be computed with communication at most $\log S + 1$, if Alice knows the values of the first variables according to π and Bob knows all other variables. This step significantly uses that all OBDDs in the proof have the same order since we need a partition of variables between Alice and Bob that is consistent with all OBDDs in the proof.
2. The second step of the proof is a construction of the transformation that maps formulas that do not have short refutations for one order to formulas that have no short refutations for all orders.

Itsykson et al. [10] proposed the generalised proof system OBDD($\wedge$, weakening, reordering) that uses the additional derivation rule:

Reordering rule: D_i can be derived it it represents the same function as D_j for $j < i$ (but the orders of these OBDDs may be different).

Notice that the conjunction rule is allowed to apply only for two OBDDs in the same order, since it is an **NP**-hard problem to check whether the conjunction of two OBDDs is equal to the third one if it is not provided that all the OBDDs are in the same order [15, Lemma 8.14].

Buss et al. [6] showed that OBDD($\wedge$, weakening, reordering) is strictly stronger than OBDD($\wedge$, weakening). At this moment it is still an open question to prove a superpolynomial lower bound on size of OBDD($\wedge$, weakening, reordering)-refutations.

Itsykson et al. [10] also studied the proof system OBDD($\wedge$, reordering), a subsystem of OBDD($\wedge$, weakening, reordering) that does not use the weakening rule. It was shown that the pigeonhole principle and Tseitin formulas are hard for OBDD($\wedge$, reordering). The lower bound proofs essentially explore that the join rule cannot be applied to OBDDs in different orders. Although this restriction is natural for efficient verification of proofs, it seems too artificial that it is necessary for lower bound proofs.

1.1 Our Results

In this paper we show that lower bounds may be also proved if we relax restrictions on orders of OBDDs in derivations. In Section 3.2 we study a calculus generalising OBDD($\wedge$, reordering) (we use the word "calculus" in order to emphasise that this generalisation is not necessary a proof system since it is NP-hard to verify the correctness of derivations); in this generalisation we allow to apply the conjunction rule to any previously derived OBDDs. Moreover, we allow to use nondeterministic read-once branching programs instead of OBDDs; we call this generalization 1-NBP($\wedge$)-calculus.

Theorem 3.1 shows a $2^{\Omega(n)}$ lower bound on the size of 1-NBP($\wedge$) refutations of the perfect matching principle and Tseitin formulas based on algebraic expanders, where n is the number of vertices in the graph. The plan of the proof is the following.

1. We choose special clauses of the formula and show that if a conjunction of several clauses of the formula is unsatisfiable, then it contains $\Omega(n)$ special clauses.

2. By the previous step, every 1-NBP($\wedge$) refutation of the formula contains a 1-NBP representing the conjunction of $\theta(n)$ special clauses. We show that this 1-NBP has exponential size.

In Section 5 we show that formulas which are "based on a bipartite graph" have short 1-NBP($\wedge$) refutations; as a corollary we get a family of formulas that are easy for 1-NBP($\wedge$) and exponentially hard for cutting planes.

We apply these results to resolve the question of whether OBDD($\wedge$) simulates resolution or not. The history of this question is quite entangled; the paper [19] claimed that OBDD($\wedge$) does not simulate resolution and the paper [11] claimed that OBDD($\wedge$) does not simulate even tree-like resolution. However, Buss et al. [6] noticed that proofs in the mentioned papers were not complete and showed that tree-like OBDD($\wedge$) does not simulate tree-like resolution. Nonetheless, the question about dag-like OBDD($\wedge$) was left open. Section 4 shows that proofs from [11, 19] are not just incomplete but there is a counterexample to the main statements from them. In the same section we show that 1-NBP($\wedge$) does not simulate tree-like resolution; the key difference in our proof is considering nondeterministic branching programs which allow us to close the gap in the previous proofs.

In Section 6 we consider a subsystem of OBDD($\wedge$, weakening, reordering) that allows the use of at most k different orders of OBDDs in every proof. For some constant C we show an exponential lower bound on the complexity of derivations in this proof system for $\ell = C \log n$, where n is the number of variables. Additionally, we note that the full system OBDD($\wedge$, weakening, reordering) has short proofs of the hard formulas. We prove this statement in three steps.

1. For any unsatisfiable CNF formula ϕ we consider the search problem Search_ϕ: given an assignment σ of variables of ϕ to find a clause of ϕ falsified by σ. We show that if there is a proof of a formula ϕ in OBDD($\wedge$, weakening, reordering) of size S using $\ell = (k-1)$ orders, then the k-party communication complexity of Search_ϕ in the NOF model is $O(\log^2 S)$ for some balanced partition of the values of variables between k participants.

2. We construct a transformation $\mathcal{T}$ of unsatisifiable CNF formulas satisfying the following property. If the k-party communication complexity of Search_ϕ is big with respect to some partition of the values of variables between k participants, then the k-party communication complexity of $\mathrm{Search}_{\mathcal{T}(\phi)}$ for every balanced partition is big.

3. Using a family of formulas ϕ_n such that Search_{ϕ_n} requires $\Omega(\sqrt{n}/2^k k)$ bit of k-party communication for some partition of the values of variables between k participants. We apply $\mathcal{T}$ to ϕ_n and from the first step we get that size of any OBDD($\wedge$, weakening, reordering) derivation using $k-1$ orders is at least $2^{\Omega\left(\sqrt{\frac{\sqrt{n}}{2^k k}}\right)}$.

All the proven and known before results are shown on the Figure 1.

1.2 Open Questions

The main open question is to prove a superpolynomial lower bound on the proof system OBDD($\wedge$, weakening, reordering). In dag-like case it is still open to prove lower bound even for two different orders.

Another question is to study automatisability of OBDD based proof systems. It is known that the Clique–Coloring principle has polynomial OBDD($\wedge$, weakening) refutations [6]. Hence, providing that the separator that accepts k-cliques and rejects $(k+1)$-colorable graphs has big circuit complexity, OBDD($\wedge$, weakening) is not automatisable [5]. However, even the pigeonhole principle requires OBDD($\wedge$, reordering) proofs of exponential size. So it is interesting to investigate whether OBDD($\wedge$, reordering) is automatisable or not.

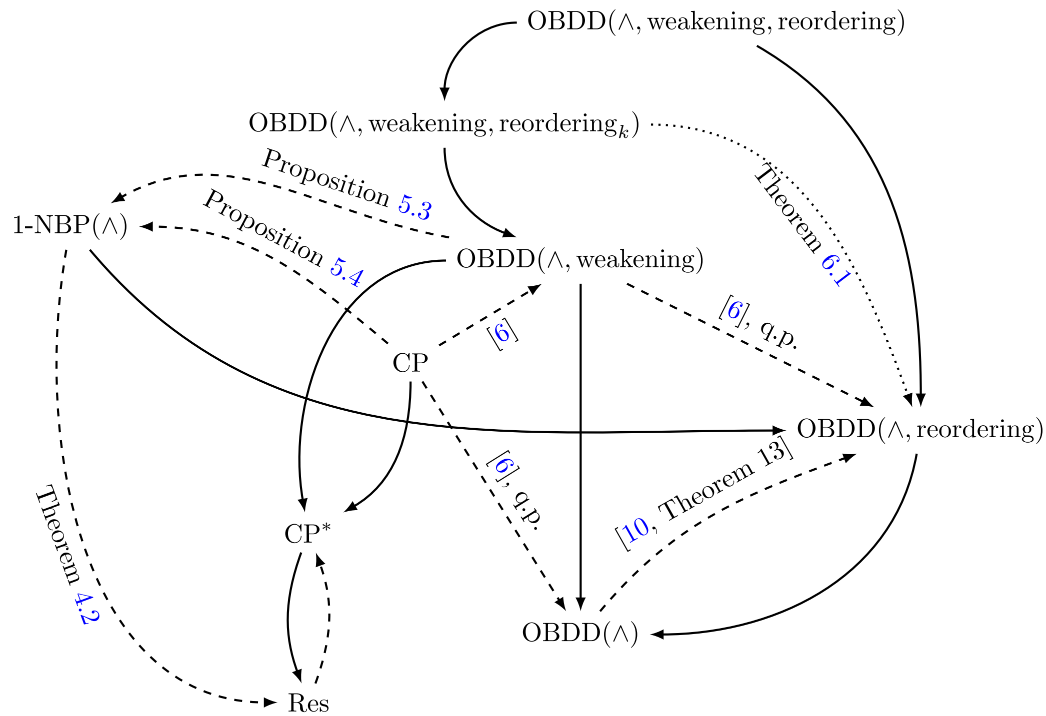

Figure 1: $C_1 \longrightarrow C_2$ denotes C_1 p-simulates C_2, and $C_1 \dashrightarrow C_2$ denotes C_1 does not p-simulate C_2. All the results except one are for the dag-like versions of the systems. The dotted line denotes the separation that is known only for tree-like version. New results are labelled with the relevant theorem. All the separations on the picture are exponential, except the two separations labeled by "q.p" for "quasipolynomial".

2 Preliminaries

2.1 Branching Programs

A deterministic branching program (BP) represents a Boolean function $\{0,1\}^n \to \{0,1\}$ by a directed acyclic graph with exactly one source and two sinks. All the nodes except the sinks are labeled with a variable; every internal node has exactly two outgoing edges: one is labeled with 1 and the other is labeled with 0. One of the sinks is labeled with 1 and the other is labeled with 0. The value of the function for a given values of variables is evaluated as follows: we start a path from the source such that for every node on its path we go along the edge that is labeled with the value of the corresponding variable. This path will end in a sink. The label of this sink is the value of the function.

A nondeterministic branching program (NBP) differs from a deterministic in the way that we also allow guessing nodes that are unlabelled and have two outgoing unlabelled edges. So nondeterministic branching program may have three type of nodes: guessing nodes, nodes labeled with a variable (we call them just labeled nodes) and two sinks; the source may be either a guessing node or a labeled node. The values of a function represented by a nondeterministic branching program equals 1, if there exists at least one path from the source to the sink labeled with 1 such that for every node labeled with a variable on its path we go along the edge that is labeled with the value of the corresponding variable, while for guessing nodes we are allowed to choose any of two outgoing edges.

Note that deterministic branching programs constitute a special case of nondeterministic branching programs.

A deterministic or nondeterministic branching program is (syntactic) read-k (k-BP or k-NBP) if every path from the source to a sink contains at most k occurrences of every variable.

An ordered binary decision diagram (OBDD) is a partial case of 1-BP, where on every path from the source to a sink all the variables appear in the same order.

We say that π is an order over the variables $x_1, \ldots, x_n$ if π is a bijection from $[n]$ to $\{x_1, \ldots, x_n\}$. We denote the set $\{\pi(1), \ldots, \pi(s)\}$ by $\pi[\leq s]$ and the set $\{\pi(s+1), \ldots, \pi(n)\}$ by $\pi[> s]$.

2.2 Graph Based Formulas

Definition 2.1. *Let $G(V, E)$ be an undirected graph. For every edge $e \in E$, let x_e be a propositional variable. A formula* based on G *has the following structure: $\bigwedge_{v \in V} \phi_v$, where ϕ_v is a CNF formula that depends on only the variables x_e such that e is incident to v.*

The following formulas are most important graph-based formulas:

1. Tseitin formulas. The Tseitin formula $\mathrm{TS}_{G,f}$ is defined by an undirected graph $G(V, E)$ and a function $f : V \to \{0, 1\}$, $\mathrm{TS}_{G,f} = \bigwedge_{v \in V} \phi_v$, where ϕ_v is a CNF representation of the following equation: $\sum_{e \in E: v \text{ is incident to } e} x_e \equiv f(v) \pmod 2$. It is well known [20] that $\mathrm{TS}_{G,f}$ is satisfiable if and only if for every connected component S of G, $\sum_{v \in S} f(v)$ is even.

2. Perfect matching principle. The formula PMP_G is based on an undirected graph G as follows: $\mathrm{PMP}_G = \bigwedge_{v \in V} \phi_v$, where ϕ_v encodes in CNF that among values of x_e for e that are incident to v, exactly one variable has the value 1 and all other variables have the value 0. The formula PMP_G is satisfiable if and only of G has a perfect matching.

3. Graph pigeonhole principle. The formula PHP_G is based on a bipartite graph $G(V, E)$, where the set of vertices V is split in two disjoint parts P (pigeons) and H (holes), and every edge connects a pigeon with a hole. For every $v \in P$ a formula $\phi_v = \bigvee_{e \in E: v \text{ is incident to } e} x_e$ (every pigeon flies to at least one hole). For every $v \in H$ a formula ϕ_v encodes that at most one x_e such that v is incident e has value 1 (at most one pigeon can fly to v). The standard pigeonhole principle PHP_n^{n+1} is precisely $\mathrm{PHP}_{K_{n+1,n}}$, where $K_{n+1,n}$ is the complete bipartite graph with $n+1$ and n vertices in the parts.

2.3 OBDD-based proof systems

Let φ be an unsatisfiable CNF formula. An OBDD proof of φ is a sequence $D_1, D_2, \ldots, D_t$ of OBDD's and permutations $\pi_1, \ldots, \pi_t$ such that D_t is an π_t-OBDD that represents the constant false function, and such that each D_i is either an π_i-OBDD which represents a clause of φ or can be obtained from previous OBDD's by one of the following inference rules:

conjunction or join: D_i represents the Boolean function $D_k \wedge D_\ell$ for $1 \leq \ell, k < i$, where D_i, D_k, D_ℓ have the same order $\pi_i = \pi_k = \pi_\ell$;

weakening: there exists a $j < i$ such that D_i and D_j have the same order $\pi_i = \pi_j$, and D_j semantically implies D_i. The latter means that every assignment that satisfies D_j also satisfies D_i;

reordering: D_i is an π_i-OBDD that is equivalent to a π_j-OBDD D_j with $j < i$.

Note that although we use terminology "OBDD proof", it is actually a *refutation* of φ. It is well known that there is a polynomial time algorithm which recognizes whether a given $D_1, \ldots, D_t$ and $\pi_1, \ldots, \pi_t$ is a valid OBDD proof of a given φ. The size of this proof is equal to $\sum_{i=1}^{t} |D_i|$.

We use several different OBDD proof systems with different sets of allowed rules. For example, the OBDD($\wedge$, weakening) proof system uses conjunction and weakening rules; hence, all OBDDs in such a proof have the same order π. We use the notation π-OBDD($\wedge$) proof and π-OBDD($\wedge$, weakening) proof to explicit indicate the ordering. We say that OBDD($\wedge$, weakening, reordering) refutation is OBDD($\wedge$, weakening, reordering$_k$) refutation if there are at most k orders $\pi_1, \ldots, \pi_\ell$ such that all the OBDDs in the refutation are in these orders.

If every D_i is used as a premise of the inference rule at most once, then the proof *tree-like*.

2.4 Calculus of Branching Programs

Let $\phi(x)$ be an unsatisfiable CNF formula with n variables $x_1, x_2, \ldots, x_n$, $\phi = \bigwedge_{i \in I} C_i$, where C_i is a clause of ϕ for $i \in I$. A formal derivation of the contradiction from $\phi(x)$ is a sequence of Boolean functions $f_1(x), f_2(x), \ldots, f_s(x)$, where f_s is identically false function and for all $i \in [s]$, f_i is either represents a clause C_j for some $j \in I$, or $f_i(x) = f_k(x) \wedge f_\ell(x)$, where $k, \ell < i$.

Let $\mathfrak{C}$ be a way of representation of Boolean functions, for example, 1-BP or 1-NBP. If all Boolean functions in a formal derivation of the contradiction are represented as $\mathfrak{C}$, then we call it a derivation in $\mathfrak{C}(\wedge)$-calculus. Thus we define a derivation in the following calculi: 1-NBP, 1-BP.

Note that the proof system OBDD($\wedge$, reordering)-proofs is a partial case of 1-BP and 1-NBP calculus.

The size of a derivation in $\mathfrak{C}(\wedge)$-calculus is the sum of sizes of all $\mathfrak{C}$-representations of all the Boolean functions from the derivation.

2.5 Algebraic Expanders

Let $G(V, E)$ be an undirected graph without loops but possibly with multiple edges. The graph G is an algebraic (n, d, α)-expander if G is d-regular, $|V| = n$ and the absolute value of the second largest eigenvalue of the adjacency matrix of G is not greater than αd.

It is well known that for all $1 > \alpha > 0$ and all large enough constants d there exist a natural number n_0 and a family $\{G_n\}_{n=n_0}^{\infty}$ of (n, d, α)-algebraic expanders. There are explicit constructions such that G_n can be constructed in poly(n) time [14]. Also, it is known that a random d-regular graph is an expander with high probability.

Let us denote by $E(A, B)$ a multiset of edges that have one end in A and another end in B. Note that in the case where both ends of an edge are simultaneously in A and in B, we count this edge twice.

Lemma 2.2 (Expander mixing lemma [1])**.** *Let $G(V, E)$ be an (n, d, α)-expander, $A, B \subseteq V$. Then* $\left| |E(A, B)| - \frac{d|A||B|}{n} \right| \leq \alpha d \sqrt{|A||B|}$.

Lemma 2.3 ([8])**.** *Let $G(V, E)$ be an (n, d, α)-expander. Then for every set $S \subseteq G$ the following inequality is satisfied: $|E(S, V \setminus S))| \geq d|S|(1 - \alpha - \frac{|S|}{n})$.*

Proof. $|E(S, V \setminus S)| = d|S| - |E(S, S)| \geq d|S| - \frac{d}{n}|S|^2 - \alpha d \sqrt{|S|^2} = d|S|(1 - \alpha - \frac{|S|}{n})$. The inequality is followed from Lemma 2.2. ∎

2.6 Communication Complexity

We use the "number on forehead" model of communication complexity. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. We have k players who have to compute $f(s)$. The function f is known by all of them. Let $\Pi = (\Pi_1, \Pi_2, \ldots, \Pi_k)$ be a partition of $[n]$ ($\Pi_i \cap \Pi_j = \emptyset$ for every $i \neq j \in [k]$). The player i knows only bits of s with indices from $[n] \setminus \Pi_i$. They have a communication channel. On each round of their communication one of them sends a string to everyone else and they are trying to minimise the total number of sent bits.

In a more general situation, they have a relation $R \subseteq \{0,1\}^n \times Z$ and the player i knows only bits of s with indices from $[n] \setminus \Pi_i$ and they wish to find $z \in Z$ such that $(s, z) \in R$.

More formally, a communication protocol with respect to a partition Π is a tree T where each internal node v is labeled by a function $f_v : \{0,1\}^{[n]\setminus\Pi_i} \to \{0,1\}$, each leaf is labeled by an element $z \in Z$, each node has 2 children, and edges from a node to its children are labeled by different Boolean values. The value of the protocol T on an input s is the label of the leaf reached starting from the root, and walking on the tree, at each internal node labeled by $f : \{0,1\}^{[n]\setminus\Pi_i} \to \{0,1\}$ we go by the edge labeled by $f(s|_{[n]\setminus\Pi_i})$.

The cost of the protocol is the depth of T.

The communication complexity denoted $\mathbf{D}(R,\Pi)$ of a relation R is the minimal cost among the protocols for this relation with respect to the partition Π.

For $\delta \in (0,1)$ we say that a partition Π of the input of a relation $R : \{0,1\}^n \times Z$ into k subsets is δ-balanced if $|\Pi_i| \geq \lfloor \delta n \rfloor$ for every $i \in [k]$.

3 Lower Bounds on 1-NBP Calculus

In this section we prove lower bounds on 1-NBP($\wedge$) refutations, in Section 3.1 we define hard formulas and in Section 3.2 we show the lower bounds.

3.1 Hard formulas

We consider a formula based on an undirected graph $G(V,E)$. Racall that this means a formula $\Phi = \bigwedge_{v\in V} \phi_v$ such that for every $v \in V$ the formula ϕ_v depends on variables $x_{e_1}, x_{e_2}, \dots, x_{e_d}$, where $\{e_1, e_2, \dots, e_d\}$ is a set of edges incident to v. We say that Φ is a *matching* formula if the following properties are satisfied:

- $\phi_v = (x_{e_1} \vee x_{e_2} \vee \cdots \vee x_{e_d}) \wedge \phi'_v$, where ϕ'_v has value 1 if we substitute all zeros to its variables (note that ϕ_v would be 0 if we substitute all zeros to its variables).
- ϕ_v is satisfied by any assignment of its variables (i.e. $\{x_{e_1}, x_{e_2}, \dots, x_{e_d}\}$) such that the value of one variable is 1 and the values of all other variables is 0.

Examples of matching formulas:

1. Perfect matching principle PMP_G is a matching formula.
2. Tseitin formula $\mathrm{TS}_{G,f}$ is a matching formula if $f \equiv 1$.

3.2 Lower Bound

Theorem 3.1. *Let Φ be an unsatisfiable matching formula based on (n,d,α)-algebraic expander $G(V,E)$, where $\alpha < \frac{1}{161+\sqrt{8}}$. Then the size of any* 1-NBP($\wedge$) *refutation of Φ has size at least $2^{\Omega(n)}$.*

Let Ψ be a conjunction of several clauses of formula $\Phi = \bigwedge_{v\in V} \phi_v$. We assume that $\Psi = \bigwedge_{v\in V} \psi_v$, where ψ_v is the conjunction of some (may be empty) set of clauses of ϕ_v. We say that a vertex $v \in V$ is *active* in Ψ, if ψ_v equals zero if we substitute zeros to all its variables. In other words a vertex $v \in V$ that is incident to edges $e_1, \dots, e_d$ is active if ψ_v contains the clause $(x_{e_1} \vee x_{e_2} \vee \cdots \vee x_{e_d})$.

The proof of Theorem 3.1 is based on the following lemmas that will be proved later.

Lemma 3.2. *Let Ψ be a conjunction of several clauses of the formula $\Phi = \bigwedge_{v\in V} \phi_v$. If Ψ contains at most γn active vertices, then Ψ is satisfiable, where $\gamma = \frac{9}{10}(\frac{1}{2} - \alpha)$.*

Lemma 3.3. *Let Ψ be a conjunction of several clauses of the formula $\Phi = \bigwedge_{v\in V} \phi_v$. Let $\beta = \frac{1-\alpha(\sqrt{8}+1)}{9} \cdot \frac{9}{10}$. If Ψ is satisfiable and contains at least $\beta n/2$ and at most βn active vertices, then every* 1-NBP *representation of Ψ has size at least $2^{\Omega(n)}$.*

Proof of Theorem 3.1. Consider some 1-NBP($\wedge$)-refutation of the formula Φ: $D_1, D_2, \dots, D_s$. For every $i \in [s]$, D_i represents the conjunction of a subset of clauses of Φ. By Lemma 3.2, a formula that corresponds to D_s contains more than γn active vertices. Note that $\gamma \geq \beta$, hence $\gamma n \geq \beta n$, where γ and β are constants from Lemma 3.2 and Lemma 3.3.

Let Ψ_1 and Ψ_2 be the conjunctions of several clauses of the formula Φ. Then the number of active vertices in $\Psi_1 \wedge \Psi_2$ is at most the sum of the numbers of active vertices in Ψ_1 and in Ψ_2. Indeed, a vertex v is active in a formula Ψ if and only if Ψ contains the clause $\bigvee_{e \in E_v} x_e$, where E_v is set of edges that are adjacent to v. Hence, there exists $j \in [s-1]$ such that D_j corresponds to a formula with at least $\frac{\beta n}{2}$ and at most βn active vertices. By Lemma 3.2, D_j is satisfiable. Hence, by Lemma 3.3 the size of D_j is at least $2^{\Omega(n)}$.

In order to complete the proof of Theorem 3.1 we only need to prove Lemma 3.2 and Lemma 3.3. □

3.3 Expanders and Matchings

For a set of vertices S of undirected graph $G(V, E)$ we denote by $\Gamma(S)$ the set of vertices that are adjacent to at least one vertex from S. We also denote the set of external neighbours of S by $\delta(S) = \Gamma(S) \setminus S$.

Lemma 3.4. *Let $G(V, E)$ be an algebraic (n, d, α)-expander. Let $k > 0$ and $\beta \in (0, 1)$ satisfy $\alpha(\sqrt{k} + 1) + \beta(k + 1) < 1$. Then for every set $S \subseteq V$, if $|S| \leq \beta n$, then $|\delta(S)| > k|S|$.*

Proof. Assume that $|S| \leq \beta n$ and $|\delta(S)| \leq k|S|$. Then by Lemma 2.2, $|E(S, \overline{S})| = |E(S, \delta(S))| \leq \frac{d}{n}|S||\delta(S)| + \alpha d\sqrt{|S||\delta(S)|} \leq \frac{d}{n}k|S|^2 + \alpha d\sqrt{k}|S|$. By Lemma 2.3, $|E(S, \overline{S})| \geq d|S|(1 - \frac{|S|}{n} - \alpha)$. The latter two inequalities imply that $\alpha(\sqrt{k} + 1) + \frac{|S|}{n}(k + 1) \geq 1$, but this is a contradiction since $\frac{|S|}{n} \leq \beta$ and by the condition of the Lemma $\alpha(\sqrt{k} + 1) + \beta(k + 1) < 1$. □

The following partial cases of Lemma 3.4 are important for us:

Corollary 3.5. *Let $G(V, E)$ be an algebraic (n, d, α)-expander.*

1. *Let $\alpha < \frac{1}{2}$ and $\beta = (1/2 - \alpha)\frac{9}{10}$. Then for every set $S \subseteq V$, if $|S| \leq \beta n$, then $|\delta(S)| > |S|$.*
2. *Let $\alpha < \frac{1}{4}$ and $\beta = \frac{1 - \alpha(\sqrt{8}+1)}{9} \cdot \frac{9}{10}$. Then for every set $S \subseteq V$, if $|S| \leq \beta n$, then $|\delta(S)| > 8|S|$.*

We will use Tutte's classical criterion of the existence of perfect matching:

Theorem 3.6 (Tutte, 1947)**.** *A graph G has a perfect matching iff for any set $S \subseteq V$:*

$$o(G - S) \leq |S|$$

where $G - S$ denotes the graph G without the vertices from the set S and $o(G - S)$ denotes the number of connected components with odd cardinality in the obtained graph.

Lemma 3.7. *Let $G(V, E)$ be an undirected graph with n vertices. Assume that for some subset $S \subseteq V$ for all its subsets $A \subseteq S$ the inequality $|\delta(A)| > |A|$ holds, then there exists a matching in G that covers all the vertices from S (i.e. for every $v \in S$ there exists an edge from the matching that is incident to v).*

Proof. We define a new graph G' that is obtained from a subgraph of G induced by the set $S \cup \delta(S)$ by adding all the edges between all the pairs of vertices from $\delta(S)$. If G' contains an odd number of vertices, we add to it one more vertex v_0 and connect v_0 with all the vertices from $\delta(S)$. Now G' has an even number of vertices. Let $B = \delta(S) \cup \{v_0\}$ if $|S \cup \delta(S)|$ is odd and $B = \delta(S)$ otherwise. We are going to prove that the graph G' has a perfect matching M. Notice that a set of edges from M that are incident to vertices from S is a matching in G that covers S.

We will show that G' satisfies the conditions of Tutte's theorem. We know that $|\delta(S)| > |S| > 0$. We claim that in the graph G every vertex $v \in S$ is connected by a path with a vertex from $\delta(S)$. Indeed, assume that there is a vertex $v \in S$ that is not connected with $\delta(S)$. Then the connected component U that contains v lies in S, hence, $\delta(U) = \emptyset$; the latter contradicts the inequality $|\delta(U)| > |U|$. Also it is easy to see that for every $v \in S$ there is a path connecting v and $\delta(S)$ in G such that all the vertices in this path except the last are in S. Indeed, if we consider some path that connects v in $\delta(S)$, then the first vertex in this path that is not in S should be in $\delta(S)$. The latter observation implies that G' is connected, since every vertex from S is connected with some vertex in $\delta(S) \subseteq B$ and B is a clique in G'. The number of vertices in G' is even, hence. G' does not contain odd connected components.

Consider arbitrary non-empty set $A \subseteq S \cup B$. For the sake of contradiction assume that after removing vertices A from G' we get a graph with at least $|A| + 1$ odd connected components. Since B is a clique in G', all remaining vertices from B should be in one connected component of $G' - A$. Hence, there exist at least $|A|$ connected components of $G' - A$ that contains only vertices from S. Let U denote the union of all connected components of $G' - A$ that contains only vertices from S. Since every such a component has at least one vertex, we get that $|U| \geq |A|$. We know that $\delta(U) > |U|$ in the graph G, hence, $|\delta(U)| > |U| \geq |A|$. All vertices of $\delta(U)$ are in G'; and $|\delta(U) \setminus A| \geq 1$. We get a contradiction since there is a vertex in $\delta(U) \setminus A$ that is connected with U; the later is impossible since U is the union of several connected components in $G' - A$. Thus G' satisfies the conditions of Tutte's theorem and G' has a perfect matching. □

3.4 Proof of Lemma 3.2

Proof of Lemma 3.2. The graph G is an algebraic (n, d, α)-expander with $\alpha < \frac{1}{2}$, hence, by Lemma 3.4 for $\beta = \gamma = \frac{9}{10}(\frac{1}{2} - \alpha)$ and $k = 1$ (i.e. by Corollary 3.5), we get that for any set $S \subseteq V$, if $|S| \leq \beta n$, then $|\delta(S)| > |S|$. Then by Lemma 3.7, for every $S \subseteq V$, if $|S| \leq \gamma n$, then there exists a matching in G that covers S.

Let S be the set of active vertices in Ψ. Consider some matching of the graph G that covers S. Let ρ be an assignment of variables that corresponds to this matching (edges from the matching have value 1 and other edges have value 0). For every vertex $v \in V$ the assignment ρ assigns either all zeros to variables corresponding edges that are incident to v if v is not active, or one to one variable and zeros to other variables if v is active. If v is active, then v is covered by the matching, hence one of the variables corresponding edges that are incident to v has the value 1 and other variables have the value 0 in ρ (hence, ρ satisfies ϕ_v and hence ψ_v). Thus ρ satisfies Ψ. □

3.5 Lower Bound on The Size of 1-NBP

In this subsection we prove Lemma 3.3.

Let a graph $G(V, E)$ be an algebraic (n, d, α)-expander, where $\alpha < \frac{1}{2}$. Let Φ be a matching formula based on the graph G. Let Ψ be a conjunction of several clauses of Φ and S be the set of active vertices of Ψ.

We say that an assignment of all variables of formula Φ is *good* if it corresponds to a matching that covers the set S. More precisely a full assignment ρ is good if there exists a matching M in the graph G that covers all the vertices from the set S and $\rho(x_e) = 1$ iff $e \in M$ for all $e \in E$. Note that all the good assignments satisfy Ψ.

Lemma 3.8. *Let a* 1-NBP *D represent the formula Ψ. Let p_1 and p_2 be two paths in D from the source to the sink labeled with* 1 *that are consistent with good assignments. Assume that p_1 and p_2 have a common node v. For every $i \in \{1, 2\}$ we consider a matching ρ_i that corresponds to the part of the path p_i from the source to v. If ρ_i covers a subset $S_i \subseteq S$ for $i \in \{1, 2\}$, then $S_1 = S_2$.*

Proof. Let us denote by $p_{i,1}$ and $p_{i,2}$ parts of the path p_i from the source to v and from v to the sink labeled by 1, respectively. Note that the path $p_{1,1}p_{2,2}$ (i.e. we go along p_1 from the sink to v and go along p_2 from v to the sink labeled with 1) is also an accepting path of D. Since the set S is active in Ψ, for every $u \in S$ there exists e incident to u such that the path $p_{1,1}p_{2,2}$ substitutes $x_e := 1$. Hence, a matching that corresponds to $p_{2,2}$ covers all the vertices from $S \setminus S_1$. On the other hand, p_2 corresponds to a matching, hence, a matching corresponding to $p_{2,2}$ covers exactly vertices $S \setminus S_2$ from S. Hence $S \setminus S_2 \subseteq S \setminus S_1$. Analogously, $S \setminus S_1 \subseteq S \setminus S_2$. Thus, $S_1 = S_2$. □

In the proof of Lemma 3.3 we will use the following lemma:

Lemma 3.9. *Let $G(V,E)$ be an algebraic (n,d,α)-expander with $\alpha < \frac{1}{161+\sqrt{8}}$. Let $S \subseteq V$ and $\beta n/2 \le |S| \le \beta n$, where $\beta = \frac{1-\alpha(\sqrt{8}+1)}{9} \cdot \frac{9}{10}$. Then there exists a probabilistic distribution $\mathcal{D}$ on matchings covering S such that for every subset $A \subseteq S$ if $|A| \in \{\lceil |S|/2 \rceil, \lceil |S|/2 \rceil - 1\}$, then a random matching distributed according $\mathcal{D}$ does not contain edges connecting vertices from A with vertices from $S \setminus A$ with probability at most $2^{-\Omega(n)}$.*

We will prove Lemma 3.9 in the next subsection.

Proof of Lemma 3.3. Consider a 1-NBP for the formula Ψ. Let S be the set of active vertices of Ψ. Consider some good assignment; let p be a path from the source to the sink labeled with 1 corresponding to this assignment. By Lemma 3.8, every vertex of path p corresponds to a subset of S. Since the path p corresponds to a matching and going along an edge of the path p we increase the number of covered vertices from S by at most two, there exists a vertex v_p such that the part of the path p from the source to v_p covers exactly vertices $A_p \subseteq S$ from S, where $|A_p| \in \{\lceil |S|/2 \rceil, \lceil |S|/2 \rceil - 1\}$. Notice that for all paths through v_p that correspond to good assignments, their first parts (from the source to v_p) correspond to matchings that do not contain edges from $E(A_p, S \setminus A_p)$.

Consider the distribution $\mathcal{D}$ from Lemma 3.9. Consider a random matching distributed according $\mathcal{D}$ and accepting path p that corresponds to this matching. The path p defines the vertex v_p and the set A_p. By Theorem 3.9, the probability of the event that the random accepting path that corresponds to a random matching distributed according $\mathcal{D}$ passes v_p is at most $2^{-\Omega(n)}$. Since every such path p passes v_p with probability 1, the number of different vertices v_p is at least $2^{\Omega(n)}$. □

3.6 Distribution on The Matchings

In this subsection we give a proof of Lemma 3.9. We start with an informal idea of the proof. Since the size of S is large enough, there are $\Omega(n)$ edges connecting two vertices from S. We show that the set of edges connecting A and $S \setminus A$ consists of a constant fraction of all edges connecting two vertices from S. Consider the following randomised process:

1. Let I be the set of edges connecting two vertices from S.
2. $M := \emptyset$
3. While I is not empty
   - Take $e \leftarrow I$ at random;
   - $M := M \cup \{e\}$;
   - Remove e and all the edges that have common endpoints with e from I;
4. Try to cover all the uncovered by M vertices of S using a matching N from exterior edges (i.e. edges connecting S and $V \setminus S$)
5. Return $M \cup N$

The described algorithm has the following problem: it may be impossible to implement the 4th step of the process since there are no corresponding matching from exterior edges. Thus we change the algorithm and at first we choose a set of *bad* vertices $B \subset S$ that has relatively small number of edges connecting B and $V \setminus S$. We argue that B is small and we cover B before generating of a random matching using Lemma 3.7. Let B' be a set of vertices covered by the choosing matching. $|B'| \le 2|B|$, hence B' is also small. Since B' is small, the number of edge between $A \setminus B'$ and $S \setminus (A \cup B')$ is also $\Omega(n)$.

Proof of Lemma 3.9. Let us fix some $A \subseteq S$ such that $|A| \in \{\lceil |S|/2 \rceil, \lceil |S|/2 \rceil - 1\}$.

Consider an inclusion-wise maximal set of vertices $B \subseteq S$ such that the set of all vertices $V \setminus S$ that are connected by edges to vertices from B has less than $|B|$ vertices.

Notice that for any subset $T \subseteq S \setminus B$, the number of neighbors of T among $V \setminus S$ is at least $|T|$. Since otherwise we may choose $T \cup B$ instead of B and get a contradiction with the maximality of B.

By Lemma 3.4, for $k = 8$ (i.e. by Corollary 3.5), if $|B| \leq \beta n$, then $|\delta(B)| > 8|B|$. Since $|\delta(B) \cap (V \setminus S)| < |B|$, $|S \cap \delta(B)| \geq 7|B|$. Hence, $|S| \geq 8|B|$ or $|B| \leq |S|/8$.

We cover the set B as follows: by Lemma 3.7, there exists a matching N that covers B. Let N' be a subset of N that consists of all edges from N that are incident to vertices from B. Let S' be obtained from S by the deletion of all endpoints of edges from N'; $|S'| \geq |S| - |N'| \geq |S| - 2|B| \geq \frac{3}{4}|S|$. Let $A' = S' \cap A$.

The number of edges between A' and $S' \setminus A'$ can be estimated by the Expander Mixing Lemma. We know $|A'| \leq |S|/2 \leq \frac{2}{3}|S'|$ and $|A'| \geq |S|/2 - 3/2 \geq |S'|/2 - 3/2$. Assume that $|S'| = \tau n$, we know that $\tau \geq \frac{3}{8}\beta$. For large enough n the following inequalities are satisfied: $2\tau n/3 \geq |A'| \geq \tau n/3$ and $2\tau n/3 \geq |S' \setminus A'| \geq \tau n/3$.

By the Expander Mixing Lemma $|E(A', S \setminus A')| \geq \frac{d}{n}|A'||S \setminus A'| - \alpha d\sqrt{|A'||S \setminus A'|} \geq dn(\tau^2/9 - 2/3\alpha\tau)$ for large enough n. We need that $\tau^2/9 - 2/3\alpha\tau > 0$ or $\tau > 6\alpha$. $\tau \geq \frac{3}{4}\frac{\beta}{2}$, hence, it is sufficient to have $\beta > 16\alpha$. The later inequality is satisfied since $\alpha < \frac{1}{161+\sqrt{8}}$.

For the chosen τ we have $|E(A', S \setminus A')| = \Omega(n)$; in other words, the edges from this cut constitutes a constant fraction. Consider the following randomised process:

1. Let I be the set of the edges connecting two vertices from S'.
2. $M := \emptyset$
3. While I is not empty
   - Take $e \leftarrow I$ at random;
   - $M := M \cup \{e\}$;
   - Remove e and all edges that have common endpoints with e from I;

We claim that this probability $1 - 2^{-\Omega(n)}$ the set M contains edges from $E(A', S \setminus A')$. Indeed, initially I contains all edges from $E(A', S \setminus A')$ and edges from $E(A', S \setminus A')$ constitutes a constant fraction from I. In every execution of the loop on the 3rd step of the randomised process we remove at most $2d$ edges from I, hence among the first $\Omega(n)$ executions of the loop on the 3rd step $|I \cap E(A', S \setminus A')| = \Omega(n)$. Every time with constant probability $e \in E(A', S \setminus A')$. Thus the probability that the final value of M does not contain an edges from $E(A', S \setminus A')$ is at most $2^{-\Omega(n)}$.

Some of the vertices from $T \subseteq S'$ may be still non covered. We will cover them using the Hall's theorem. Unfortunately, the part of vertices from $V \setminus S$ may be already used for the matching covered B. Let us assume that there exists a set $X \subseteq T$ that does not satisfy the condition of the Hall's theorem: X has fewer than $|X|$ neighbours in $V \setminus S$ that are not covered by N'. This contradicts the maximality of B, since we may use $B \cup X$ instead of B. Hence it is possible to extend $N' \cup M$ to a matching covering S. We return this matching. □

4 1-NBP Calculus Does Not Polynomially Simulate Tree-like Resolution

A couple earlier papers have claimed that resolution is not simulated by OBDD($\wedge$), see Theorem 5 of [19] and Corollary 4 of [11], but we have been unable to verify their proofs. The difficult point in the proofs is in Lemma 8 of [19] and in Lemma 4 of [11]. In the former, it is shown that two distinct nodes in an OBDD $B(F, \prec)$ correspond to two distinct nodes in another OBDD $B(F \cup G, \prec)$; however, it does not follow from this that n distinct nodes in $B(F, \prec)$ correspond to n distinct nodes in $B(F \cup G, \prec)$. A similar technique is implicitly used in the latter paper.

Let $\phi(x_1, x_2, \ldots, x_n)$ be a CNF formula. The extension rule is an operation that adds to ϕ new clauses that represent $z = f(x_1, x_2, \ldots, x_n)$, where z is a fresh variable and f is an arbitrary Boolean function. We say that a formula ψ is an extension of ϕ if ψ may be obtained from ϕ by several applications of the extension rule. We may assume that $\psi = \phi \wedge E$, where E is the conjunction of all the added clauses.

Lemma 4 of [11] claims that for every CNF ϕ if $\psi = \phi \wedge E$ is an extension of ϕ, then for every order π the minimal size of OBDD in the order π computing ϕ is at most the the minimal size of OBDD in the order π computing ψ. Lemma 8 of [19] is formulated as a very partial case Lemma 4 of [11] but indeed its "proof" never used the specifics of this case.

We start from a counter example for Lemma 4 of [11].

We give an example of a Boolean function $f(\vec{a})$ which has a short representation as a π-OBDD with the addition of an extension, but requires an exponentially long τ-OBDD representation over the original variables $\vec{a}$, where τ is the restriction of the order π to the original variables.

Let $t \geq 1$ and $m = 2^t$. Let $f(y_1, \ldots, y_m, x_1, \ldots x_t)$ be the *index* function defined as

$$f(y_1, \ldots, y_m, x_1, \ldots x_t) \;=\; y_{\mathrm{bin}(\vec{x})},$$

where $\mathrm{bin}(\vec{x})$ means the integer with binary representation given by $x_1, \ldots, x_t$. (We could have also called f a "selection" function or "look up" function.)

We add extension variables $z_1, \ldots, z_t$ with the (rather trivial) extension definitions $z_i \leftrightarrow x_i$. For this, define $g(\vec{z}, \vec{y}, \vec{z})$ by

$$g(z_1, \ldots, z_t, y_1, \ldots, y_m, x_1, \ldots x_t) \;=\; \Big(y_{\mathrm{bin}(\vec{x})} \wedge \bigwedge_i (z_i \leftrightarrow x_i)\Big)$$

Let τ be the linear order placing all y_i's before all x_i's. Let π be the linear order placing all z_i's before all y_i's, and all y_i's before all x_i's, so that π extends τ.

Proposition 4.1. *1. f has $m + t = 2^t + t$ many inputs and g has $m + 2t = 2^t + 2t$ inputs.*

2. Any τ-OBDD for f requires size at least $2^m = 2^{2^t}$.

3. There is a π-OBDD for g of size $O(2^t)$.

Proof. Part 1. is immediate from the definitions.

Part 2. is easy to prove by observing that once the τ-OBDD has queried all the y_i's, it must remember all m of the values of $y_1, \ldots, y_m$: To prove this, note that each setting to $y_1, \ldots, y_m$ gives a different function of $x_1, \ldots, x_t$.

Part 3. is proved by constructing the π-OBDD. The first stage uses $2^{t+1} - 1$ many nodes to query the variables z_i and remember all their values. The second stage uses exactly 2^t nodes, one per y_i, to query the needed value of y_i with $i = \mathrm{bin}(\vec{z})$. If the queried y_i has value 0 (False), the OBDD outputs 0. Otherwise, the third stage checks the values of each x_i to see if it is equal to the corresponding z_i. It is obvious that this can be done with $t \cdot 2^t$ nodes, but by collapsing nodes, the third stage can even be done with $2^{t+1} - 1$ many nodes. The overall size of the π-OBDD is less than $5 \cdot 2^t = O(2^t)$. □

The number of inputs n to f is $m + t = 2^t + t$. Thus the π-OBDD for g has size $O(n)$, and the τ-OBDD for f requires size at least $2^{n - \log n} = 2^n/n$.

4.1 Proof of The Separation

In this subsection we show that it is possible to fix the problem in the previous "proofs" by switching from OBDDs to 1-NBPs. Namely we prove the following theorem.

Theorem 4.2. *There is a family of formulas $\{\Psi_n\}_{n\in\mathbb{N}}$ of size $\mathrm{poly}(n)$ such that any $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ refutation of Ψ_n has size at least $2^{\Omega(n)}$ and there is a tree-like resolution refutation of Ψ_n of size $\mathrm{poly}(n)$.*

The correct analogue of Lemma 4 of [11] is the following.

Lemma 4.3. *Let $f(x_1, x_2, \dots, x_n)$ be a Boolean function and $g(z, x_1, \dots, x_n)$ be a Boolean function such that $g(h(x_1, \dots, x_n), x_1, x_2, \dots, x_n)$ is the constant 1, where $h : \{0,1\}^n \to \{0,1\}$ is a Boolean function. Then for every* 1-NBP *for $f(x_1, x_2, \dots, x_n) \wedge g(z, x_1, \dots, x_n)$ of size S there exists a* 1-NBP *for $f(x_1, x_2, \dots, x_n)$ of size at most S.*

Proof. Consider a 1-NBP D for the function $f(x_1, x_2, \dots, x_n) \wedge g(z, x_1, \dots, x_n)$ and change all the nodes labeled with z to guessing nodes and erase labellings of outgoing edges. We denote the resulting labeled graph by D'. On every path from the source to a sink of D' every variable appears at most once as a label of a node. We claim that D' is a correct 1-NBP for f. Consider $\alpha \in \{0,1\}^n$ and assume that $f(\alpha) = 0$ and D' has a path from the source to the sink 1 that is consistent with α. Consider on this path in D, it must finish at the sink 0 since $f(\alpha) \wedge g(z, \alpha)$ equals 0 for every value of the variable z. Assume that $f(\alpha) = 1$; since $f(\alpha) = 1$ and $g(h(\alpha), \alpha) = 1$, there is a path in D from the source to the sink labeled with 1 that is consistent with substitution $x = \alpha, z = h(\alpha)$. This path in D' is consistent with $x = \alpha$ and also finishes in the sink 1. ◻

Lemma 4.4. *Let ϕ be an unsatisfiable CNF formula and $\psi = \phi \wedge E$ be an extension of ϕ. Then for every* 1-NBP($\wedge$) *refutation of ψ of size s there exist a* 1-NBP($\wedge$) *refutation of ϕ of size at most s.*

Proof. It is sufficient to prove for the case then ψ is obtained from ϕ by one extension operation. Let ψ represent $\phi(x_1, \dots, x_n) \wedge z = f(x_1, x_2, \dots, x_n)$. Consider 1-NBP($\wedge$) refutation of ψ: D_1, D_2, …, D_s. Every D_i represents the conjunction of several clauses of ψ. Consider a sequence D'_1, D'_2,…, D'_s, where D'_i is the shortest 1-NBP representation of the conjunction of clauses from ϕ that are from the conjunction represented by D_i. If D'_i is an empty conjunction, then it is just constant 1. We claim that D'_s represents constant 0. Indeed, assume α is a satisfying assignment of D'_s, then $x = \alpha$, $z = f(\alpha)$ satisfies D_s, but D_s is constant 0.

For every $i \in [s]$ the function D_i is the conjunction of D'_i and a function that becomes 1 after the substitution $z = f(x_1, x_2, \dots, x_n)$. Hence, by Lemma 4.3, size of D'_i is at most size of D_i. Every D'_i represents either a clause of ϕ, or the constant 1, or $D'_j \wedge D'_k$, where $j, k < i$. If we drop off all constants 1 and repetitions (that may come as a conjunction with the constant 1) we get a 1-NBP($\wedge$) refutation of ϕ of size at most s. ◻

It is well known that for every constant-degree graph $G(V, E)$ on n vertices and every function $f : V \to \{0,1\}$, if the Tseitin formula $\mathrm{TS}_{G,f}$ is unsatisfiable, then there exists a tree-like derivation of $\neg\mathrm{TS}_{G,f}$ in Extended Frege proof system of poly(n) size. Hence, there exists a tree-like proof of $\mathrm{TS}_{G,f}$ in the Extended resolution of poly(n) size. The latter implies the following statement:

Lemma 4.5. *Let G_n be undirected graph with n vertices, degrees of all vertices do not exceed d. Let $\mathrm{TS}_{G,f}$ be unsatisfiable Tseitin formula. Then there exists a CNF formula Ψ_n of size* poly(n) *that is an extension of TS_{G_n,f_n} and there is a tree-like resolution refutation of Ψ_n of size* poly(n).

Proof of Theorem 4.2. Let G_n be an (n, d, α)-algebraic expander with $\alpha < \frac{1}{161+\sqrt{8}}$ and let f_n be a labeling function for G_n such that TS_{G_n,f_n} is unsatisfiable. Then by Theorem 3.1 size of any 1-NBP($\wedge$) refutation of TS_{G_n,f_n} has size at least $2^{\Omega(n)}$ and as a result any OBDD($\wedge$, reordering) refutation of TS_{G_n,f_n} has size at least $2^{\Omega(n)}$. The formula Ψ_n from Lemma 4.5 is an extension of TS_{G_n,f_n}, hence by Lemma 4.4 any 1-NBP($\wedge$) refutation of Ψ_n has size at least $2^{\Omega(n)}$. By Lemma 4.5 the formula Ψ_n has tree-like resolution refutation of size poly(n). ◻

5 Upper Bounds on 1-NBP Calculus

In this section we show that 1-NBP($\wedge$)-calculus (and even 1-BP-calculus) has short refutations of several non-trivial families of formulas.

Theorem 5.1. *Let $\phi = \bigwedge_{v \in V} \phi_v$ be unsatisfiable formula based on bipartite graph $G(V, E)$. Suppose that, for all $v \in V$ there is a* 1-BP($\wedge$) *derivation of ϕ_v from its clauses of size at most S. Then there exists a* 1-BP($\wedge$) *refutation of ϕ of size* poly($|V|, S$).

Proof. Let V_1 and V_2 be the two parts of the bipartite graph G. We show that for every $i \in \{1, 2\}$ there is a 1-BP($\wedge$) derivation of $\bigwedge_{v \in V_i} \phi_v$ of size $\text{poly}(|V|, S)$. Note that for $i \in \{1, 2\}$, distinct formulas ϕ_v for $v \in V_i$ do not share variables. For all $v \in V_i$ we derive ϕ_v; the total size of this derivation is at most $|V_i|S$. Then we consequently derive conjunctions ϕ_v from $v \in V_i$: the conjunction of the first two, of the first three and etc. If two 1-BPs do not share variables, the size of their conjunction is at most the sum of sizes of initial 1-BPs. Hence size of the derivation of $\bigwedge_{v \in V_i} \phi_v$ is at most $O(|V_i|^2 S)$.

When we derive $\bigwedge_{v \in V_i} \phi_v$ for $i = 1$ and $i = 2$, we apply the conjunction rule and get the constant 0. ☐

Corollary 5.2. *If G is a bipartite graph and formulas PMP_G, $\text{TS}_{G,f}$, PHP_G are unsatisfiable, then there exists polynomial size* 1-BP($\wedge$) *refutations of them.*

Buss et al. [6] proved that there exists two families of formulas ϕ_n and ψ_n that polynomial size OBDD($\wedge$, reordering) proofs but ϕ_n requires superpolynomial OBDD($\wedge$, reordering) refutations and ψ_n requires superpolynomial cutting planes refutation. The proof had the following structure: at first they presented a family of formulas that has $\Omega(\log^2 n)$ resolution width and has polynomial size OBDD($\wedge$) refutation and then applied the dag-like lifting developed by Garg et al. [7].

It is possible to apply the same approach to the graph pigeonhole principle. It is known that there is a sequence of bipartite graphs G_n such that G_n has linearly many edges, PHP_{G_n} is a $O(1)$-CNF formula and it requires $\Omega(n)$ resolution width [4]. By Corollary 5.2 PHP_{G_n} has $\text{poly}(n)$ size 1-NBP refutations. Using the same approach as in the paper [6] we may get the following statements:

Proposition 5.3 (see Section 4.4 of [6])**.** *There is a family of formulas $\{\phi_n\}_{n \in \mathbb{N}}$ of size* $\text{poly}(n)$ *such that any* OBDD($\wedge$, weakening) *refutation of ϕ_n has size at least $2^{n^{\Omega(1)}}$ but there is a* 1-BP($\wedge$) *refutation of ϕ_n of size* $\text{poly}(n)$.

Proposition 5.4 (see Section 4.3 of [6])**.** *There is a family of formulas $\{\psi_n\}_{n \in \mathbb{N}}$ of size* $\text{poly}(n)$ *such that any cutting planes refutation of ψ_n has size at least $2^{n^{\Omega(1)}}$ but there is a* 1-BP($\wedge$) *refutation of ψ_n of size* $\text{poly}(n)$.

6 Lower Bounds on $\text{OBDD}(\wedge, \text{weakening}, \text{reordering}_\ell)$

Theorem 6.1. *For any $\ell > 0$, there is a family of formulas $\{\phi_n\}_{n \in \mathbb{N}}$ such that $|\phi_n| = \text{poly}(n, \ell)$, every tree-like* $\text{OBDD}(\wedge, \text{weakening}, \text{reordering}_\ell)$ *proof of ϕ_n has size at least $2^{\Omega\left(\sqrt{\sqrt{n}/2^{\ell+1}(\ell+1)}\right)}$, and there is a tree-like* OBDD($\wedge$, reordering) *proof of ϕ_n of size* $\text{poly}(n)$

The proof of this theorem is based on two following theorems.

Theorem 6.2. *Let ϕ be an unsatisfiable CNF. If there is a tree-like* $\text{OBDD}(\wedge, \text{weakening}, \text{reordering}_\ell)$ *proof of ϕ of size S, then there is a $\frac{1}{\ell+1}$-balanced partition Π of the variables of ϕ into $\ell + 1$ subsets such that* $\mathbf{D}(\text{Search}_\phi, \Pi) = O(\log^2 S)$.

Theorem 6.3. *For any $k, \delta > 0$, there is a family of formulas $\{\phi_n\}_{n \in \mathbb{N}}$ such that $|\phi_n| = \text{poly}(n, k, \frac{1}{\delta})$ and* $\mathbf{D}(\text{Search}_{\phi_n}, \Pi) = \Omega(\sqrt{n}/2^k k)$ *for every δ-balanced partition Π into k subsets.*

Moreover, there is a tree-like OBDD($\wedge$, reordering) *proof of ϕ_n of size* $\text{poly}(n)$.

Proof of Theorem 6.1. Let us consider the family of formulas $\{\phi_n\}_{n \in \mathbb{N}}$ from Theorem 6.3 for $\delta = \frac{1}{\ell+1}$ and $k = \ell + 1$.

We prove that every $\text{OBDD}(\wedge, \text{weakening}, \text{reordering}_\ell)$ proof of ϕ_n has size at least $2^{\Omega\left(\sqrt{\sqrt{n}/2^{\ell+1}(\ell+1)}\right)}$. Assume that there is a tree-like $\text{OBDD}(\wedge, \text{weakening}, \text{reordering}_\ell)$ proof of ϕ_n of size S. Hence, by Theorem 6.2, there is a $\frac{1}{\ell+1}$-balanced partition Π_n of the variables of ϕ_n such that $\mathbf{D}(\text{Search}_{\phi_n}, \Pi_n) = O(\log^2 S)$.

Hence, $\log^2 S = \Omega\left(\sqrt{n}/2^{\ell+1}(\ell+1)\right)$ by the properties of ϕ_n; i.e. $S = 2^{\Omega\left(\sqrt{\sqrt{n}/2^{\ell+1}(\ell+1)}\right)}$.

However, by Theorem 6.3, there is a tree-like OBDD($\wedge$, reordering) proof of ϕ_n of size $\text{poly}(n)$. ☐

6.1 Lower Bounds for Multiparty Best-communication Complexity

This section proves Theorem 6.3. To prove it we use a lower bound for communication complexity proven by Göös and Pitassi [9]. Their construction uses a pebbling contradiction.

Definition 6.4. *Let G be a directed acyclic graph with one sink t. The CNF formula* Peb_G *(*pebbling contradiction *for a graph G), uses a variable x_v for each vertex v of G and has the following clauses:*

- $\neg x_t$*;*
- *for each vertex v, the clause $x_v \vee \bigvee_{i=1}^{d} \neg x_{p_i}$ where $p_1, \dots, p_d$ are all the immediate predecessors of v ($d = 0$ if v is a source).*

It is not hard to see that Peb_G has short tree-like OBDD($\wedge$) proofs.

Theorem 6.5 ([6]). *For any directed acyclic graph $G(V, E)$ with n vertices and maximum in-degree d there is a tree-like* OBDD($\wedge$) *proof of* Peb_G *of size* $\mathrm{poly}(n)$.

Additionally, we need a concept of composition of CNFs. Let $\phi \circ g(y_{1,1}, \dots, y_{n,m})$ denote the formula that can be obtained from $\phi(x_1, \dots, x_n)$ by replacement of each variable x_i by $g(y_{i,1}, \dots y_{i,m})$, where g is a CNF.

Theorem 6.6 ([9]). *For every $k > 0$, there is a family of directed acyclic graphs $\{G_n\}_{n\in\mathbb{N}}$ with constant degree such that G_n has n vertices, a CNF formula g on $O(1)$ variables, and a family of partitions $\{\Pi_n\}_{n\in\mathbb{N}}$ such that*

- $|\mathrm{Peb}_{G_n} \circ g| = \mathrm{poly}(n)$*;*
- Π_n *is a partition of the variables of* $\mathrm{Peb}_{G_n} \circ g$ *into k subsets;*
- $\mathbf{D}(\mathrm{Search}_{\mathrm{Peb}_{G_n}\circ g}, \Pi_n) = \Omega(\sqrt{n}/2^k k)$.

This theorem is almost the same as Theorem 6.3 but the key difference that that theorem holds only for one partition. Thus, now we need to transform that lower bound into a best-communication lower bound i.e. the lower bound that holds for all balanced partitions. In order to do this, let us define the transformation based on the transformation introduced by Segerlind [18].

Let $t = \lceil \log(n) \rceil$ and $N = 2^t$, and $\mathbb{F}$ be the field $\mathrm{GF}(N)$. Define $\mathcal{P}_t$ to be the set of all mappings given by $x \mapsto ax + b$ with $a, b \in \mathbb{F}$ and $a \neq 0$. Elements of $\mathcal{P}_n$ may be represented by binary strings of length $2t$ such that the first t bits are not all zero; we denote the representation of $\alpha \in \mathcal{P}_t$ as $\mathrm{rep}(\alpha)$. Note that $\mathcal{P}_t \subseteq S_N$ so we have to add new variables, $x_{n+1}, \dots, x_N$. Then define

$$\mathrm{perm}(\phi)(z_1, \dots, z_{2t}, x_1, \dots, x_N) = \bigwedge_{\sigma\in\mathcal{P}_t} \left[\left(\bigwedge_{i=1}^{2t} z_i = \mathrm{rep}(\sigma)_i \right) \to \phi(x_{\sigma(1)}, \dots, x_{\sigma(n)}) \right] \wedge \bigvee_{i=1}^{t} z_i.$$

Additionally, let $\phi^{\vee_m}(y_{1,1}, \dots, y_{n,m})$ denote the formula that is obtained from $\phi(x_1, \dots, x_n)$ by replacing each variable x_i by the disjunction of m fresh variables $y_{i,1}, \dots y_{i,m}$ (note that it is a composition of g and $\bigvee_{i=1}^{m} z_i$).

Now we can define the transformation $\mathcal{T}$. Let ϕ be a formula on n variables, $K > 0$ be an integer, and $m(K, \delta, n)$ be the least integer m such that $\frac{2Kn}{\delta m} + \frac{Kn}{mn-1} < 1$, so $m(K, \delta, n) = O(\frac{Kn}{\delta})$. Then $\mathcal{T}_{K,\delta}(\phi) = \mathrm{perm}(\phi^{\vee_{m(K,\delta,n)}})$, where n is the number of variables of ϕ.

Theorem 6.7. *Let $K, \delta > 0$. For every ϕ with big enough number of variables, for every partition Π of the variables of* Search_ϕ *into K subsets, and for every δ-balanced partition Γ of the variables of* $\mathrm{Search}_{\mathcal{T}_{K,\delta}(\phi)}$ *into K subsets,* $\mathbf{D}(\mathrm{Search}_{\mathcal{T}_{K,\delta}(\phi)}, \Gamma) \geq \mathbf{D}(\mathrm{Search}_\phi, \Pi)$.

For proving this theorem we also need two technical results.

Lemma 6.8 ([21])**.** *For any t, $|\mathcal{P}_t| = 2^t \cdot (2^t - 1)$, every mapping from $\mathcal{P}_t$ is a permutation, and for any $x_1, x_2, y_1, y_2 \in [2^t]$ if $x_1 \neq x_2$ and $y_1 \neq y_2$, then $\Pr_{\pi \in \mathcal{P}_t}[\pi(x_1) = y_1, \pi(x_2) = y_2] = \frac{1}{2^t(2^t-1)}$.*

Lemma 6.9 (Chebyshev's inequality)**.** *If $X_1, \ldots, X_t$ are random Boolean variables and $Y = \sum_{i=1}^{t} X_i$, then*

$$\Pr[Y = 0] \leq \frac{\mathbb{E}Y + \sum_{i \neq j \in [t]} \mathrm{Cov}(X_i, X_j)}{(\mathbb{E}Y)^2}.$$

Proof of Theorem 6.7. Let $\Pi = (\Pi_0, \ldots, \Pi_{K-1})$ be an arbitrary partition into K subsets. We prove that if there is a protocol for $\mathrm{Search}_{\mathcal{T}_{K,\delta}(\phi)}$ and a δ-balanced partition Γ with communication complexity S, then there is a protocol for Search_ϕ and the partition Π with communication complexity S.

Let n be the number of variables of ϕ, and $m = m(K, \delta, n)$. Let $N = n \cdot m$ be the number of the variables of $\phi^{\vee_m}$, and $t = \lceil \log N \rceil$.

Claim 6.9.1. *There is a permutation $\pi \in \mathcal{P}_t$ such that for any $i \in [n]$ and $k \in [K]$ there is $j \in [m]$, such that $y_{i,j}$ is mapped to a variable from Γ_k by π.*

Let $V = \{v_1, \ldots, v_{2^\ell}\}$ be a set of variables of $\mathcal{T}_{K,\delta}(\phi) = \mathrm{perm}(\phi^{\vee_m})$ not including any of the variables z_1, $\ldots$, z_{2t} encoding a permutation. For every $i \in [n]$ and $k \in [K]$, let $v_{r(i,k)}$ denote some variable $y_{i,j}$ that is mapped to a variable from Γ_k by the permutation π. The protocol will be the following: on input x_1, $\ldots$, x_n it runs the protocol for $\mathrm{Search}_{\mathrm{perm}(\phi^{\vee_m})}$ with substitution to input values such that z_1, $\ldots$, z_{2t} encode π, all the variables from $V \setminus \{v_{r(i,k)} : i \in [n], k \in [K]\}$ are equal to zero, and

- $v_{r(i,k)} = x_i$ if $x_i \in \Pi_k$ and
- $v_{r(i,k)} = 0$ otherwise.

It is easy to see that the communication complexity of this protocol is equal to S on the partition Γ. Thus, $\mathbf{D}(\mathrm{Search}_\phi, \Pi) \leq \mathbf{D}(\mathrm{Search}_{\mathcal{T}_{K,\delta}(\phi)}, \Gamma)$.

We now prove Claim 6.9.1. We write $\tau(v_i)$ to denote $v_{\tau(i)}$, where τ is a permutation in $\mathcal{P}_t$. Let Γ' be the partition induced by Γ on V. Note that $|\Gamma'_k| \geq \lfloor \delta N \rfloor - 2\lceil \log N \rceil$, so Γ'_k is nonempty.

For $k \in K$ and choosing $\pi \in \mathcal{P}_t$ uniformly at random, let $\chi^k_{i,j}$ be the Boolean random variables such that $\chi^k_{i,j} = 1$ iff $y_{i,j}$ is mapped by π into Γ'_k. Set $Y^k_i = \sum_{j=1}^{m} \chi^k_{i,j}$. By Lemma 6.8, $\chi^k_{i,j}$ has expectation equals $\frac{|\Gamma'_k|}{N}$ and by additivity of expectation, the expectation of Y^k_i is equal to $\frac{m|\Gamma'_k|}{N}$. Note that

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

Hence, by Lemma 6.9,

$$\begin{aligned}
\Pr[Y^k_i = 0] &\leq \frac{\mathbb{E}Y^k_i + \sum_{j_0 \neq j_1 \in [m]} \mathrm{Cov}(\chi^k_{i,j_0}, \chi^k_{i,j_1})}{\left(\mathbb{E}Y^k_i\right)^2} \leq \frac{N}{m|\Gamma'_k|} + \frac{m(m-1)}{m^2(N-1)} \\
&\leq \frac{N}{m\left(\lfloor \delta N \rfloor - 2\log N\right)} + \frac{1}{N-1} \leq \frac{N}{m\frac{\delta N}{2}} + \frac{1}{N-1} = \frac{2}{\delta m} + \frac{1}{nm-1}.
\end{aligned}$$

Therefore, by union bound $\Pr\left[\exists i,k\ Y_i^k = 0\right] \le \frac{2Kn}{\delta m} + \frac{Kn}{nm-1} \le 1$. As a result, there is a permutation $\pi \in P_\ell$ such that for any $i \in [n]$ and $k \in [K]$ there is a $j \in [m]$ and $v_{r(i,k)} \in \Gamma'_k$ such that $v_{r(i,k)} = \pi(y_{i,j})$. ☐

Additionally, in order to prove the upper bound we may use the following result from [6].

Theorem 6.10 (see [6, Lemma 2, Corollary 6, Theorem 5])**.** *Let $\{\phi_n\}_{n\in\mathbb{N}}$ be a family of unsatisfiable formulas in $O(1)$-CNF such that there is a tree-like $\mathrm{OBDD}(\wedge)$ proof of ϕ_n of size $\mathrm{poly}(n)$ and $|\phi_n| = \mathrm{poly}(n)$.*

- *There is a tree-like $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ proof of $\mathrm{perm}(\phi)$ of size $\mathrm{poly}(n)$.*
- *For every m, there is a tree-like $\mathrm{OBDD}(\wedge)$ proof of $\phi^{\vee_m}$ of size $\mathrm{poly}(n,m)$.*
- *For every formula g on $O(1)$ variables, there is a tree-like $\mathrm{OBDD}(\wedge)$ proof of $\phi \circ g$ of size $\mathrm{poly}(n)$.*

Corollary 6.11. *Let $\{\phi_n\}_{n\in\mathbb{N}}$ be a family of unsatisfiable formulas in $O(1)$-CNF such that there is a tree-like $\mathrm{OBDD}(\wedge)$ proof of ϕ_n of size $\mathrm{poly}(n)$ and $|\phi_n| = \mathrm{poly}(n)$. Then there is a tree-like $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ proof of $\mathcal{T}_{K,\delta}(\phi_n)$.*

Proof of Theorem 6.3. Let $\{G_n\}_{n\in\mathbb{N}}$ be a family of graphs, g be a CNF on $O(1)$ variables, and $\{\Pi_n\}_{n\in\mathbb{N}}$ be a family of partitions from Theorem 6.6.

Let $\phi_n = \mathcal{T}_{k,\delta}(\mathrm{Peb}_{G_n} \circ g)$. Note that, by Theorem 6.7, for every δ-balanced partition Γ of the variables of ϕ_n into k subsets, $\mathbf{D}(\phi_n, \Gamma) \ge \Omega(\sqrt{n}/2^k k)$.

However, by Theorem 6.5 there is a tree-like $\mathrm{OBDD}(\wedge)$ proof of Peb_{G_n} of size $\mathrm{poly}(n)$; thus, by Corollary 6.11, there is a $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ proof of ϕ_n of size $\mathrm{poly}(n)$. ☐

6.2 Upper Bounds for Multiparty Communication Complexity

This section proves Theorem 6.2.

Lemma 6.12. *For any orders $\pi_1, \ldots, \pi_\ell$ over the variables $x_1, \ldots, x_n$ there are $s_1, \ldots, s_\ell \in [n]$ and a partition Π of the variables $x_1, \ldots, x_n$ into $\ell + 1$ subsets such that*

- *$|\Pi_i| \ge \lfloor \frac{n}{\ell+1} \rfloor$ for every $i \in [\ell+1]$ i.e. Π is $\frac{1}{\ell+1}$-balanced;*
- *for every $i \in [\ell]$, $\pi_i[\le s_i] \cap \Pi_{i+1} = \emptyset$ and $\pi_i[> s_i] \cap \Pi_i = \emptyset$.*

Proof. Consider the following algorithm that constructs a partition.

- $S_1 := \{x_1, x_2, \ldots, x_n\}$;
- For $i = 1$ to ℓ
  - Let Π_i be the first $\lfloor \frac{n}{\ell+1} \rfloor$ elements of S_i in the order π_i.
  - Let s_i be the maximal number of an element of Π_i in the order π_i.
  - $S_{i+1} := S_i \setminus \Pi_i$.
- $\Pi_{\ell+1} := S_{\ell+1}$

By the construction, $|\Pi_i| = \lfloor \frac{n}{\ell+1} \rfloor$ for $i \in [\ell]$, and hence $|\Pi_{\ell+1}| \ge \lfloor \frac{n}{\ell+1} \rfloor$. Notice that Π_i and s_i are defined such that $\pi_i[> s_i] \cap \Pi_i = \emptyset$ and $\pi_i[\le s_i] \cap S_{i+1} = \emptyset$. Since $\Pi_{i+1} \subseteq S_{i+1}$, we get that $\pi_i[\le s_i] \cap \Pi_{i+1} = \emptyset$. ☐

Lemma 6.13. *Let a function f be computed by a π-OBDD D, $s \in [n]$ be an integer, and Π be a partition of variables of f into k subsets such that $\Pi_a \cap \pi[\le s] = \Pi_b \cap \pi[> s] = \emptyset$ for some $a, b \in [k]$. Then $\mathbf{D}(f, \Pi) \le \lceil \log |D| \rceil + 1$.*

Proof. Player a starts the computation of f according D using the variables she knows (variables outside of Π_a). She reaches a vertex v of D after reading all the variables $\pi[\le s]$ and sends the number of the vertex v; it has at most $\lceil \log |D| \rceil$ bits. Player b continues computing f starting from v using now variables he knows and sends the result of the computation (it is 1 bit). ☐

Proof of Theorem 6.2. Consider a tree-like OBDD($\wedge$, weakening, reordering$_\ell$) proof $D_1, \ldots, D_m$ of the formula ϕ of size S. Since this proof is OBDD($\wedge$, weakening, reordering$_\ell$) there are orders $\pi_1, \ldots, \pi_\ell$ over the variables of ϕ such that for every $i \in [m]$, D_i is a π_j-OBDD for some $j \in [\ell]$.

Let Π be a partition from Lemma 6.12. Based on this proof we construct a $(\ell+1)$-party communication protocol for Search$_\phi$ with respect to the partition Π of complexity at most $O(\log^2 S)$. The protocol consists of $s = O(\log S)$ steps. At each step we consider some tree T_i that is known by all the players. The inner vertices of the tree are labelled with OBDD's in orders $\pi_1, \ldots, \pi_\ell$ and the leaves are labelled with clauses of ϕ or with trivially satisfied clauses.

In the first step, the tree T_1 is the tree of our tree-like proof. $T_i \subseteq T_{i-1}$. At each step, the players know that the OBDD at the root of T_i is falsified by the input assignment, and that there exists some clause at a leaf of T_i that is falsified. In the end, the tree T_s consists of a single vertex; hence it provides a clause of ϕ that is falsified by the input assignment.

Now we describe how we obtain the tree T_{i+1} from the tree T_i. Let v be a vertex of tree T_i such that a subtree T' with root v satisfies the following condition: $\frac{1}{3}|T_i| \le |T'| \le \frac{2}{3}|T_i|$ (the players can find such a vertex v without communication). Let D be the π_j-OBDD labelling v; if the input assignment evaluates diagram D to zero, then T_{i+1} equals T' (the players can evaluate the π_j-OBDD D on the input assignment with at most $\lceil \log |D| + 1 \rceil \le 2 \log S$ bits of communication by Lemma 6.13). Otherwise, $T_{i+1} := T_i \setminus T'$.

It is easy to see that if the value of D equals zero then there is a leaf with falsified clause in the tree T'. Otherwise there is a leaf with falsified clause in the tree $T_i \setminus T'$. Also, at each step the players use at most $2 \log S$ bits of communication and there are at most $O(\log S)$ steps (since $|T_i| \le \frac{2}{3}|T_{i+1}|$). Hence, the players use at most $O(\log^2 S)$ bits of communication. ☐

Acknowledgements The authors thanks Ludmila Glinskih for fruitful discussions.

The research presented in Sections 3, 4.1 and 5 was supported by Russian Science Foundation (project 18-71-10042).

References

[1] Noga Alon and Fan R. K. Chung. Explicit construction of linear sized tolerant networks. *Discrete Mathematics*, 306(10-11):1068–1071, 2006.

[2] 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*, volume 3258 of *Lecture Notes in Computer Science*, pages 77–91. Springer, 2004.

[3] Paul Beame, Toniann Pitassi, and Nathan Segerlind. Lower bounds for Lovasz-Schrijver systems and beyond follow from multiparty communication complexity. *SIAM Journal on Computing*, 37(3):845–869, 2007.

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

[5] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. On interpolation and automatization for frege systems. *SIAM J. Comput.*, 29(6):1939–1967, 2000.

[6] 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.

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

[8] Ludmila Glinskih and Dmitry Itsykson. Satisfiable tseitin formulas are hard for nondeterministic read-once branching programs. In *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*, pages 26:1–26:12, 2017.

[9] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 847–856, 2014.

[10] Dmitry Itsykson, Alexander Knop, Andrei Romashchenko, and Dmitry Sokolov. On OBDD-based algorithms and proof systems that dynamically change order of variables. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPIcs*, pages 43:1–43:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[11] Matti Järvisalo. On the relative efficiency of DPLL and OBDDs with axiom and join. In Jimmy Ho-Man Lee, editor, *Principles and Practice of Constraint Programming - CP 2011 - 17th International Conference, CP 2011, Perugia, Italy, September 12-16, 2011. Proceedings*, volume 6876 of *Lecture Notes in Computer Science*, pages 429–437. Springer, 2011.

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

[13] 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.

[14] A Lubotzky, R Phillips, and P Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[15] Christoph Meinel and Thorsten Theobald. *Algorithms and data structures in VLSI design: OBDD - foundations and applications*. Springer, 1998.

[16] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–998, 1997.

[17] 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.

[18] 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.

[19] Olga Tveretina, Carsten Sinz, and Hans Zantema. Ordered binary decision diagrams, pigeonhole formulas and beyond. *JSAT*, 7(1):35–58, 2010.

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

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