Electronic Colloquium on Computational Complexity, Report No. 179 (2017)

# IPS-like Proof Systems Based on Binary Decision Diagrams

Alexander Knop
aknop@ucsd.edu
Department of Mathematics
University of California, San Diego
La Jolla, CA, USA

**Abstract**

It is well-known that there is equivalence between ordered resolution and ordered binary decision diagrams (OBDD) [25]; i.e., for any unsatisfiable formula $\phi$, the size of the smallest ordered resolution refutation of $\phi$ equal to the size of the smallest OBDD for the canonical search problem corresponding to $\phi$. But there is no such equivalence between resolution and branching programs (BP).

In this paper, we study different proof systems equivalent to classes of branching programs between BP and OBDD. These proof systems are similar to roABP-**IPS**, an algebraic proof system defined by Forbes et al. [14] and based on the ideal proof system introduced by Grochow and Pitassi [17].

In the paper, we show that proof systems equivalent to $k$-OBDD are not comparable with resolution and cutting planes. We also prove exponential lower bounds for these proof systems on Tseitin formulas. Additionally, we show that proof systems equivalent to $(1,+k)$-BP are strictly stronger than regular resolution.

## 1 Introduction

In 1991 Lovász et al. [25] defined a search problem $\text{Search}_\phi$ associated with an unsatisfiable CNF $\phi$: given a substitution to all the variables of $\phi$, find a falsified clause of $\phi$. In the paper, they also mentioned an unpublished result of Chvátal and Szemerédi that says that the minimal size of a read-once branching program for $\text{Search}_\phi$ is equal to the minimal size of a regular resolution proof of $\phi$. It is easy to see that the same equivalence holds for decision trees and tree-like resolution proofs and for ordered binary decision diagrams and ordered resolution proofs.

Later the complexity of this relation was extensively studied in applications of communication complexity to proof complexity [4, 11, 16, 19], and in interpolation techniques in proof complexity [13, 33].

However, the study of a relationship between the proof complexity of formulas and the complexity of branching programs for corresponding search problems was stuck and nothing is known for bigger classes of diagrams.

In this paper, we try to revitalize this research in the light of **IPS**-like proof systems. The **IPS** proof system is an algebraic proof system defined recently by Grochow and Pitassi [17].

Let $f_1,\dots,f_m \in \mathbb{F}[x_1,\dots,x_n]$ be a system of polynomials. An **IPS** proof, showing that the polynomials $f_1,\dots,f_m$ do not have a common solution in $\{0,1\}^n$, is an algebraic circuit $C(\bar{x},\bar{y},\bar{z}) \in \mathbb{F}[x_1,\dots,x_n,y_1,\dots,y_m,z_1,\dots,z_n]$ [1], such that

1. $C(x_1,\dots,x_n,0,\dots,0)=0$ and
2. $C(x_1,\dots,x_n,f_1(x_1,\dots,x_n),\dots,f_m(x_1,\dots,x_n),x_1^2-x_1,\dots,x_n^2-x_n)=1$.

If the circuit $C$ comes from a restricted class of circuits $\mathcal{C}$ we call such a refutation $\mathcal{C}$-**IPS** *refutation.*

[1] In the following we denote by $\bar{x}$ the vector of the variables $x_1,\dots,x_n$ and we use the same notation for $\bar{y}$ and $\bar{z}$.



ISSN 1433-8092

This proof system is in the spotlight because superpolynomial lower bounds for this proof system imply **VP** $\neq$ **VNP**. Unfortunately, **IPS** is a very strong proof system. For example, it simulates the Extended Frege proof system (but the opposite direction is unknown).

In 2016 Forbes et al. [14] considered $\mathcal{C}$-**IPS** proof systems for several classes of circuits $\mathcal{C}$ and proved exponential lower bounds for some of them including the **IPS**-like proof system based on algebraic read-once oblivious branching programs (roABP) defined by Nisan [27].

An algebraic read-once oblivious branching program (roABP) on the variables $x_1, \ldots, x_n$ over a field $\mathbb{F}$, is a directed acyclic graph such that:

1. the vertices are partitioned into $n+1$ layers $V_0, \ldots, V_n$, so that if $s$ is a source node and $t$ is a sink node then $V_0 = \{s\}$, $V_n = \{t\}$, and each edge goes from $V_{i-1}$ to $V_i$ for some $0 < i \leq n$;
2. each edge $e$ is labeled by an univariate polynomial over $\mathbb{F}$;
3. each variable $x_i$ appears exactly on edges in edge labels in an exactly one layer.

Each $s$-$t$ path in the roABP is said to compute the polynomial equal to the product of the labels of its edges and the roABP computes the sum over all $s$-$t$ paths of such polynomials.

Note that layers define a variable order. Thus, roABPs are very similar to OBDDs and roABPs over $\mathbb{F}_2$ are also known as $\oplus$-OBDDs. In order to stress this similarities we give the equivalent definition of $\oplus$-OBDD.

A parity ordered binary decision diagram ($\oplus$-OBDD) on the variables $x_1, \ldots, x_n$ is a directed acyclic graph such that:

1. each vertex in this graph is labeled either by a variable $x_i$ for $i \in [n]$, or by $\oplus$, or by a Boolean constant;
2. If a vertex is labeled by a variable, then it has exactly two outgoing edges: one edge is labeled by 0 and the other one is labeled by 1, additionally if a vertex is labeled by a constant it is a sink;
3. there is an order $\pi$ such that on any path in the graph all variables that appear as labels of vertices among the path are ordered in this order.

Every such a diagram defines a Boolean function $\{0,1\}^n \to \{0,1\}$. The value of this function may be computed in a recursive manner: for a sink the value is equal to the label of the sink, for a vertex labeled by a variable the value is equal to the value of a diagram along the edge that corresponds to the value of the variable, and for a vertex labeled by $\oplus$ the value is equal to the parity of the values of its children.

Because of the similarities between roABP and $\oplus$-OBDD, it is reasonable to consider proof systems similar to roABP-**IPS**, but based on classical branching programs.

In the following we consider three types of branching programs: $\oplus$-OBDD, $(1,+b)$-BP, and $b$-OBDD (for formal definitions see Section 2). Based on them we define three types of proof systems: $\oplus$-OBDD-$\mathbf{PS}_a$, $(1,+b)$-BP-$\mathbf{PS}_a$, and $b$-OBDD-$\mathbf{PS}_a$, where proof of $\phi(x_1,\ldots,x_n) = \bigwedge_{i=1}^{m} C_i(\bar{x})$ in $\mathcal{C}$-$\mathbf{PS}_a$ ($\mathcal{C}$ is a class of branching programs) is a $\mathcal{C}$ branching program $D$ on the variables $x_1, \ldots, x_n, y_1, \ldots, y_m$ such that

1. $D(x_1,\ldots,x_n,1,\ldots,1) = 1$,
2. $D(x_1,\ldots,x_n,C_1(x_1,\ldots,x_n),\ldots,C_m(x_1,\ldots,x_n)) = 0$, and
3. on any path in $D$ the variables $y_1, \ldots, y_m$ occur at most $a$ times in total altogether.

## 1.1 Our results and Structure of the Paper

In Section 3, we show that for the new proof systems based on $(1,+b)$-BP and $b$-OBDD the size of the smallest proof of $\phi$ is equal to the smallest size of $(1,+b)$-BP and $b$-OBDD diagram for $\text{Search}_\phi$, respectively and prove that $\oplus$-OBDD-$\mathbf{PS}_1$ $p$-simulates $b$-OBDD-$\mathbf{PS}_1$ for any constant $b > 0$.

In Section 4, we construct a transformation that maps a formula $\phi$ to a formula $\psi$ such that if $\text{Search}_\phi$ has large communication complexity, then $\text{Search}_\psi$ has large best-communication complexity. Using this

transformation and result of Göös and Pitassi [16] we construct hard formulas for $b$-OBDD-$\mathbf{PS}_1$ and moreover, prove that $b$-OBDD-$\mathbf{PS}_1$ does not $p$-simulate **RegRes**. However, we show that $\mathbf{TreeTh}(k)$ does not $p$-simulate OBDD-$\mathbf{PS}_1$.

In Section 5, using lower bounds on the communication with a limited number of rounds, we prove $2^{\Omega\left(n^{1/2b}\right)}$ lower bound for size of $b$-OBDD-$\mathbf{PS}_1$ proofs of Tseitin formulas. Moreover, we show that this lower bound is almost tight and as a result, show that **Res** does not $p$-simulates $b$-OBDD-$\mathbf{PS}_1$ for $b \geq 2$. Additionally, we generalize this proof and using this generalization show that Clique–Coloring principle has $b$-OBDD-$\mathbf{PS}_1$ proof of size $2^{\Omega(n^{c/b}\log n)}$ for some absolute constant $c > 0$. Hence, we separate semantic **CP** and $b$-OBDD-$\mathbf{PS}_1$ for large enough $b$. In addition, we prove a polynomial upper bound for size of $\oplus$-OBDD-$\mathbf{PS}_1$ proof of Tseitin formulas. Hence, we prove that $b$-OBDD-$\mathbf{PS}_1$ does not $p$-simulate $\oplus$-OBDD-$\mathbf{PS}_1$.

In the last section, we extend separation between **Res** and 2-OBDD-$\mathbf{PS}_1$. We also show that **RegRes** does not $p$-simulates $(1,+1)$-BP-$\mathbf{PS}_1$.

For all the listed results see Figure 1.

# 2 Preliminaries

## 2.1 Branching programs

The proof systems studied in the paper are based on different types of branching programs. This section presents a formal definition of them and gives some important properties of them.

Let $\Lambda$ be a set and $\Omega$ be a set of functions such that any function $f \in \Omega$ maps $\Lambda^l$ to $\Lambda$ for some $l$. An $\Omega$-nondeterministic branching program or $\Omega$-nondeterministic binary decision diagram is a data structure that represents a $\{0,1\}^n \to \Lambda$ function. Let $\Gamma = \{x_1, \ldots, x_n\}$ be a set of propositional variables. An $\Omega$-nondeterministic binary decision diagram is a directed acyclic graph with one source. Every vertex of the graph is labeled by a variable from $\Gamma$, or a function from $\Omega$, or by an element of $\Lambda$.

- If a vertex is labeled by an element of $\Lambda$, then it is a sink (a sink is a vertex with out-degree 0). We call a vertex labeled by $\lambda \in \Lambda$ a $\lambda$-sink.
- If a vertex is labeled by a variable, then it has exactly two outgoing edges: one edge is labeled by 0 and the other one is labeled by 1.
- If a vertex is labeled by a function $\{0,1\}^l \to \{0,1\}$, then vertex has $l$ outgoing edges, labeled by numbers $1, \ldots, l$.

Every binary decision diagram defines a function $\{0,1\}^n \to \Lambda$. The value of the function for given values of $x_1, \ldots, x_n$ is computed recursively. For each vertex $v$ of a diagram corresponds a value from $\Lambda$ computed in the following manner:

- if $v$ is a sink, then the value is equal to a label in the vertex,
- if $v$ is labeled by a variable $x_i$ and $u$ is a vertex such that $v$ and $u$ are connected by an edge labeled by the value of $x_i$, then the value in the vertex $v$ is equal to the value in the vertex $u$,
- if $v$ is labeled by a function $f : \Lambda^l \to \Lambda$, $u_1, \ldots, u_l$ are vertices such that $v$ and $u_i$ are connected by an edge labeled by $i$, and $a_1, \ldots, a_l$ are the values in the vertices $u_1, \ldots, u_l$ respectively, then the value in $v$ is equal to $f(a_1, \ldots, a_l)$.

In addition, if $R \subseteq \{0,1\}^n \times \Lambda$, then we say that a branching program $D$ is a branching program for $R$ iff $D$ represents a function $f : \{0,1\}^n \to \Lambda$ such that, $(\bar{x}, f(\bar{x})) \in R$ for all $\bar{x} \in \{0,1\}^n$.

In the text we consider three types of branching programs.

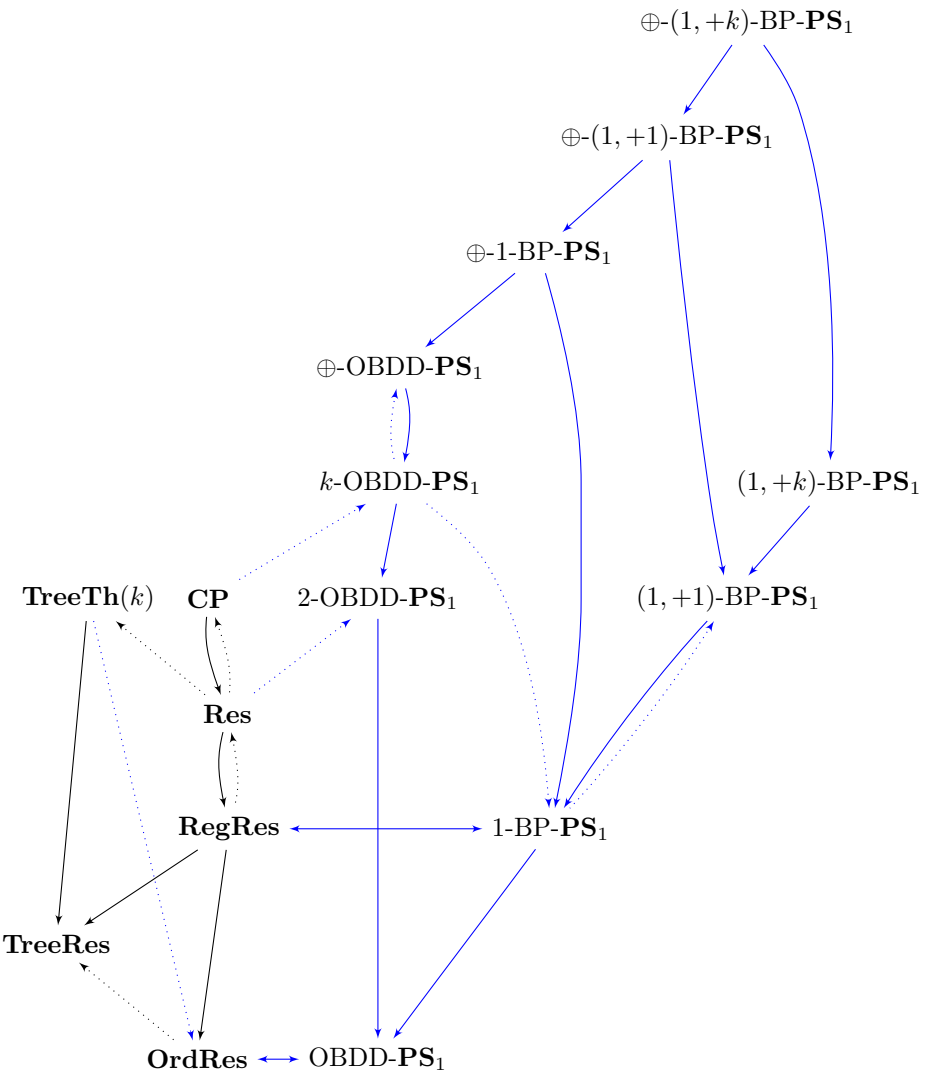
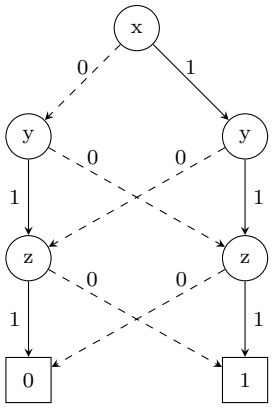
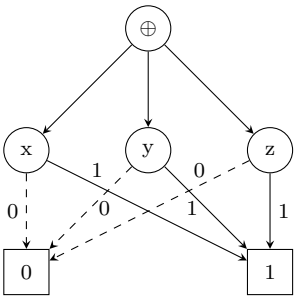



Figure 1: $C_1 \longrightarrow C_2$ denotes $C_1$ $p$-simulates $C_2$, and $C_1 \dashrightarrow C_2$ denotes $C_1$ does not $p$-simulate $C_2$. Blue indicates new results.




(a) OBDD for $x \oplus y \oplus z$




(b) $\oplus$-OBDD for $x \oplus y \oplus z$

- Let $\oplus_n : \{0,1\}^n \to \{0,1\}$ be the parity function and $\bigoplus = \bigcup_{n \in \mathbb{N}} \{\oplus_n\}$. A $\bigoplus$-nondeterministic branching program $D$ is called a $\oplus$-$(1,+b)$-BP if on any path in $D$ there are at most $b$ variables that occur more than once. If $b$ is equal to 0 we denote such a diagram $\oplus$-1-BP.
- A $\bigoplus$-nondeterministic branching program $D$ is called a read-$b$ $\oplus$-OBDD if there is an order $\pi$ and a partition of vertices of $D$ to $b$ layers $D_1, \ldots, D_b$ such that on any path in $D_i$ variables are ordered in this order and edges leaving $D_i$ go to $D_{i+1}, \ldots, D_b$. If $b$ is equal to 1 we denote such a diagram $\oplus$-OBDD. Sometimes we need to write a read-$b$ $\oplus$-OBDD as a string, in this case we write an order over the variables, a partition of the graph into layers, and a graph with all labels.
- A read-$b$ $\oplus$-OBDD $D$ is called a $b$-OBDD if $D$ is an $\emptyset$-nondeterministic branching program. If $b$ is equal to 1 we call such a diagram an OBDD.
- Additionally a $\oplus$-$(1,+b)$-BP $D$ is called a $(1,+b)$-BP if $D$ is an $\emptyset$-nondeterministic branching program. If $b$ is equal to 0 we call such a diagram a 1-BP.

It is easy to see, that any function has OBDD representation of size at most $2^{n+1}$ and as a result it has $b$-OBDD, $(1,+b)$-BP, $\oplus$-$(1,+b)$-BP, and $\oplus$-OBDD of size at most $2^{n+1}$.

One of the most important property of these classes of branching programs is the following.

**Theorem 2.1** ([35]). *There are polynomial-time algorithms for satisfiability testing of* $b$-OBDD.

**Theorem 2.2** ([6]). *There are polynomial-time algorithms for satisfiability testing of* $\oplus$-OBDD.

**Theorem 2.3** ([31]). *For any constant $b$ there is a polynomial-time algorithms for satisfiability testing of* $(1,+b)$-BP *and moreover there is a polynomial-time one-side error probabilistic algorithm for satisfiability testing of* $\oplus$-$(1,+b)$-BP.

Additionally, we need the following folklore property.

**Theorem 2.4.** *Let $D$ be a* $(1,+b)$-BP, *or a* $\oplus$-$(1,+b)$-BP, *or a* $b$-OBDD, *or a* $\oplus$-OBDD *representing $f$, a* $(1,+b)$-BP, *or a* $\oplus$-$(1,+b)$-BP, *or a* $b$-OBDD, *or a* $\oplus$-OBDD *respectively that represents $\neg f$ can be computed in polynomial in $|D|$ time.*

**Theorem 2.5** ([8]). *For any $k$, there is a polynomial $p$ such that for any function representable by a read-$k$* $\oplus$-OBDD *of size $S$ there is a* $\oplus$-OBDD *representation of size $p(S)$.*

*Moreover, for any fixed $b$ there is a polynomial-time algorithm that constructs a read-*1 $\oplus$-OBDD *by a read-$b$* $\oplus$-OBDD.

## 2.2 Proof systems

### 2.2.1 Resolution based proof systems

The *resolution principle* says that if $C$ and $D$ are clauses and $x$ is a variable, then any assignment that satisfies both of the clauses $C \vee x$ and $D \vee \neg x$ also satisfies $C \vee D$. The clause $C \vee D$ is said to be a *resolvent* of the clauses $C \vee x$ and $D \vee \neg x$ derived by *resolving on* the variable $x$.

This definition leads us to a definition of a *resolution* proof system (**Res**). A resolution derivation of a clause $C$ from a CNF $\phi$ is a sequence of clauses in which each clause is either a clause of $\phi$, or is a resolvent of two previous clauses, and $C$ is the last clause in the sequence; it is a refutation of $\phi$ if $C$ is the empty clause.

It is possible to represent a resolution refutation as a directed acyclic graph where the vertices are the clauses in the refutation, each clause of $\phi$ has out-degree 0, and any other clause has two edges go to the two clauses that produced it. The empty clause is the only source in this graph. The edges go to $C \vee x$ and $D \vee \neg x$ are labeled with the literals $x$ and $\neg x$ respectively.

In this paper we mostly interested in three subclasses of resolution proofs. A *regular resolution* (**RegRes**) refutation of $\phi$ is a resolution refutation such that on any path from the empty clause to a clause in $\phi$, no variable occurs more than once as an edge label. We call a regular resolution refutation *ordered* (**OrdRes**) if every sequence of variables labelling a path from the empty clause to a clause in $\phi$ respects the same ordering on the variables.

Finally, a *tree-like resolution* (**TreeRes**) refutation is one in which the underlying graph is a tree.

### 2.2.2 Proof systems based on binary decision diagrams

Let us now define a new family of proof systems studied in this paper.

Let $\phi(\bar{x}) = \bigwedge_{i=1}^{m} C_i(\bar{x})$ be an unsatisfiable formula in CNF, $\Omega$ be a set of Boolean functions, and $\mathcal{C}$ be a class of $\Omega$-nondeterministic branching programs. A $\mathcal{C}$-$\mathbf{PS}_a$ proof of formula $\phi$ is a diagram $D \in \mathcal{C}$ such that

- $D$ depends on variables $x_1, \ldots, x_n, y_1, \ldots, y_m$,
- $D(\bar{x}, 1, \ldots, 1) = 1$ for all $\bar{x} \in \{0,1\}^n$,
- $D(\bar{x}, C_1(\bar{x}), \ldots, C_m(\bar{x})) = 0$ for all $\bar{x} \in \{0,1\}^n$,
- on any path in $D$ the variables $y_1, \ldots, y_m$ occur not greater than $a$ times in total altogether[2].

It is easy to see that these proof systems is sound; completeness is little bit trickier and depends on a class $\mathcal{C}$, but is also true since any function has an OBDD representation. Now let us prove efficiency for $b$-OBDD-$\mathbf{PS}_a$, $\oplus$-OBDD-$\mathbf{PS}_a$, and $(1,+b)$-BP-$\mathbf{PS}_a$ using properties of these classes of branching programs.

**Theorem 2.6.** *For any constants $a > 0$ and $b > 0$, $b$-OBDD-$\mathbf{PS}_a$ and $\oplus$-OBDD-$\mathbf{PS}_a$ are proof systems in the Cook and Reckhow sense. Also for any constant $k > 0$, $(1,+b)$-BP-$\mathbf{PS}_a$ is a proof system for $k$-CNFs in the Cook and Reckhow sense and $\oplus$-$(1,+b)$-BP-$\mathbf{PS}_a$ is a probabilistic proof system for $k$-CNFs in the following sense (for similar proof systems see [17, 29, 30]): there is a polynomial-time randomized algorithm $V(x,y)$ such that*

- *for any unsatisfiable formula $\phi$ in $k$-CNF and any $\oplus$-$(1,+b)$-BP-$\mathbf{PS}_a$ proof $P$ of $\phi$,*

$$\Pr[V(\phi, P) = 1] = 1.$$

- *for any formula $\phi$ if $P$ is not a $\oplus$-$(1,+b)$-BP-$\mathbf{PS}_a$ proof of $\phi$, then*

$$\Pr[V(\phi, P) = 1] \leq \frac{1}{4}.$$

[2] Note that when $\mathcal{C} = (1,+b)$-BP there are two constraints on variables $y_1, \ldots, y_m$: on any path they occur at most $a$ times and at most $b$ of them occur more than once.

For proving this theorem we need the following Lemmas.

**Lemma 2.1** ([31]). *The test if a branching program is a* $(1,+b)$-BP *can be done in polynomial time.*

Additionally using ideas of Lemma 2.1 we may prove the following lemma.

**Lemma 2.2.** *For any constant* $b > 0$ *the test if a branching program is a* $\oplus$-$(1,+b)$-BP *can be done in polynomial time.*

*Proof.* Without loss of generality we may assume that given branching program $D$ is a branching program over the variables $x_1, \ldots, x_n$. Let us denote as $E$ the set of edges of $D$.

For any $I \subseteq [n]$ we assign to every path $p$ a function $\alpha_{I,p} : [n] \to \{0,1,2\}$ such that for all $i \in I$ $\alpha_p(i) = \min(s_i, 2)$ where $s_i$ is a number of occurrences of $x_i$ in $p$.

By $T_I^\alpha(v)$ we denote a truth value of the statement that there is a path $p$ from $v$ to the sink such that $\alpha_{I,p} = \alpha$. If $i \in I$ let us also denote by $\alpha + i$ a function $\beta : I \to \{0,1,2\}$ such that $\beta(i) = \min(\alpha(i) + 1, 2)$.

Let $v$ be a node labeled by $x_i$. It is easy to see, that if $i \in I$, and $v_0$ and $v_1$ are 0-successor and 1-successor respectively, then $T_I^\alpha(v) = \bigvee\limits_{\beta:\beta+i=\alpha} T_I^\beta(v_0) \vee T_I^\beta(v_1)$ and if $i \notin I$ or if $v$ is a nondetermenistic node, then $T_I^\alpha = \bigvee\limits_{(v,u)\in E} T_I^\alpha(u)$.

Let $s$ be the sink of $D$. In order to test the required property it is sufficient to check if for every $I$ of size $b+1$ and every $\alpha : I \to \{0,1,2\}$ such that $\alpha(i) = 2$, $T_I^\alpha(s)$ is false. Note that using the formula for $T_I^\alpha$ it is possible to compute each $T_I^\alpha$ in polynomial time and we have to compute it for $\binom{n}{k}$ different choices of $I$. As a result, we can check the required property in polynomial time. □

**Lemma 2.3.** *Let* $\mathfrak{G}_a$ *be a set of all directed labeled graphs* $G$ *such that:*

1. $G$ *does not have cycles,*
2. $G$ *has one source,*
3. *the vertices of* $G$ *are labeled by variables* $y_1, \ldots, y_m$*, and*
4. *on any path in from the source to a sink there at most* $a$ *vertices labeled by* $y_1, \ldots, y_i$ *in total.*

*For any* $a$*, the test if a graph* $G$ *belongs to* $\mathfrak{G}$ *can be done in polynomial time.*

*Proof.* The proof of this lemma is similar to the proof of Lemma 2.2. Let $G$ be a given graph. It is easy to see that we can check first four constraints in polynomial time. But we also want to check that on any path in this graph the variables $y_1, \ldots, y_m$ occur not greater than $a$ times in total. In order to to it we mark each vertex $v$ by the minimal number $O_v$ such that on any path $p$ from the source to $v$ the variables $y_1, \ldots, y_m$ occur on not greater than $O_v$ times in total altogether.

In order to do it, let us consider in the topological ordering of the graph $G$. It is easy to see that for the source $s = v_1$ of the graph $O_s = 0$ and for any vertex $v$ with successors $u_1, \ldots, u_d$ if $v$ is labeled by $y_i$ for some $i \in [m]$, then $O_v = \min\limits_{i\in[d]} O_{u_i} + 1$ and if $v$ is not labeled by $y_i$ for any $i \in [m]$, then $O_v = \min\limits_{i\in[d]} O_{u_i}$. As a result we can check that $O_v$ is at most $a$ for all vertices $v$. □

*Proof of Theorem 2.6.* Let $\phi = \bigwedge\limits_{i=1}^{m} C_i$ be a CNF, $\Omega$ be a set of all parity functions, and $D$ be a $\Omega$-nondeterministic branching program that is a candidate for $\oplus$-OBDD-$\mathbf{PS}_a$ proof of $\phi$.

In order to check that $D$ is a $\oplus$-OBDD-$\mathbf{PS}_a$ proof of $\phi$ we have to check five properties.

1. $D$ depends only on the variables $x_1, \ldots, x_n, y_1, \ldots, y_m$;
2. $D(\bar{x}, 1, \ldots, 1) = 1$ for all $\bar{x} \in \{0,1\}^n$;
3. $D(\bar{x}, C_1(\bar{x}), \ldots, C_m(\bar{x})) = 0$ for all $\bar{x} \in \{0,1\}^n$;

4. on any path in $D$ the variables $y_1, \ldots, y_m$ occur not greater than $a$ times in total;

5. $D$ is a $\oplus$-OBDD.

It is easy to see that property 1 can be checked by lookup on all vertices of $D$. We can check property 5 since encoding of a $\oplus$-OBDD includes a candidate for order over variables and we can check if all labels are ordered in this order by lookup on all vertices and their predecessors. Property 4 can be checked using Lemma 2.3.

Let us check now property 2. This property is equivalent to the property that $\neg D(\bar{x}, 1, \ldots, 1)$ is unsatisfiable. Using Theorem 2.4 and Theorem 2.2 we can check this property in polynomial time. In order to check property 3 we have to check that $D(\bar{x}, C_1(\bar{x}), \ldots, C_m(\bar{x}))$ is unsatisfiable. Note that if we replace each $y_i$ by a $\oplus$-OBDD representing a clause $C_i$ we get a read-$(a+2)$ $\oplus$-OBDD represents $D(\bar{x}, C_1(\bar{x}), \ldots, C_m(\bar{x}))$ of size at most $|D| \cdot n$ and using Theorem 2.5 and Theorem 2.2 we can check if it is unsatisfiable in polynomial time. Hence we prove that $\oplus$-OBDD-$\mathbf{PS}_a$ is a proof system in the Cook and Reckhow sense. For $b$-OBDD-$\mathbf{PS}_a$ the proof is the same but we also have to check that given branching program is deterministic.

For the case of $(1, +b)$-BP-$\mathbf{PS}_a$ ($\oplus$-$(1, +b)$-BP-$\mathbf{PS}_a$) we need the additional requirement: $\phi$ should be a $k$-CNF formula. In this case after the substitution of the clause $C_i$ instead of $y_i$ we get a $(1, +(b + ak))$-BP ($\oplus$-$(1, +(b + ak))$-BP-$\mathbf{PS}_a$) and we can use Theorem 2.3 and do everything as in the previous case. Also, for testing if a branching program is a $(1, +b)$-BP we use Lemma 2.1 (for testing if a branching program is a $\oplus$-$(1, +b)$-BP we use Lemma 2.2). □

# 3 The Canonical Search Problem

Let $\phi(x_1, \ldots, x_n) = \bigwedge_{i=1}^{m} C_i(x_1, \ldots, x_n)$ be an unsatisfiable formula in CNF. Let us define a relation $\text{Search}_\phi \subseteq \{0,1\}^n \times [m]$ such that $(x_1, \ldots, x_n, i) \in \text{Search}_\phi$ iff $C_i(x_1, \ldots, x_n) = 0$. In the paper [25] was proven that, the size of the smallest regular resolution proof (ordered resolution proof) of $\phi$ is equal to the size of the smallest read-once branching program (ordered binary decision diagram) for the relation $\text{Search}_\phi$.

In this paper we for each $a \in \mathbb{N}$ we consider the following generalization of this relation: $a\text{-Search}_\phi \subseteq \{0,1\}^n \times [m]^a$ such that $((x_1, \ldots, x_n), (i_1, \ldots, i_a)) \in a\text{-Search}_\phi$ iff these exists $j \in [a]$ such that $C_{i_j}(x_1, \ldots, x_n) = 0$.

We show an equivalence between the smallest size of a $b$-OBDD ($(1, +b)$-BP) for $a\text{-Search}_\phi$ and the smallest $b$-OBDD-$\mathbf{PS}_a$ ($(1, +b)$-BP-$\mathbf{PS}_a$) proof of $\phi$ respectively.

**Theorem 3.1.** *Let $\phi = \bigwedge_{i=1}^{m} C_i$ be an unsatisfiable formula in CNF in $n$ variables.*

1. *If there is a $(1, +b)$-BP ($b$-OBDD) for $a\text{-Search}_\phi$ of size $S$, then there is a $(1, +b)$-BP-$\mathbf{PS}_a$ proof ($b$-OBDD-$\mathbf{PS}_a$ proof) of $\phi$ of size $(a + 2) \cdot S$.*

2. *If there is a $(1, +b)$-BP-$\mathbf{PS}_a$ ($b$-OBDD-$\mathbf{PS}_a$) proof of $\phi$ of size $S$, then there is a $(1, +b)$-BP ($b$-OBDD) for $a\text{-Search}_\phi$ of size $\left(\sum_{i=0}^{a} \binom{n}{i}\right) \cdot S$.*

*Proof.* The proof of the first statement is relatively easy. Let $D$ be a diagram for $a\text{-Search}_\phi$. We replace each sink of the diagram labeled by $\{i_1, \ldots, i_a\}$ by a diagram representing the conjunction of $y_{i_1}, \ldots, y_{i_a}$ and get a diagram $D'$. It is easy to see, that

- $D'(x_1, \ldots, x_n, 1, \ldots, 1) = 1$ for all $x_1, \ldots, x_n$,
- $D'(x_1, \ldots, x_n, C_1(x_1, \ldots, x_n), \ldots, C_m(x_1, \ldots, x_n)) = 0$ for all $x_1, \ldots, x_n$, and
- the size of $D'$ is at most $2a \cdot |D|$.

The proof of the second statement is little bit more involved. But the idea behind it is simple: we somehow reverse the transformation from the proof of the first statement. Let us fix some $b$-OBDD-$\mathbf{PS}_a$ ($(1,+b)$-BP-$\mathbf{PS}_a$) proof $D$ of the formula $\phi$. Let $p$ be a consistent path from source to a vertex $v$ and $Y_p = \{i \in [m] \ : \ y_i \text{ occur on the path } p\}$. Note that if $v$ is a 0-sink, and $\bar{x} \in \{0,1\}^n$ is an input such that $\bar{x}, C_1(\bar{x}), \ldots, C_m(\bar{x})$ activates the path $p$, and for all $i \in Y_p$, $C_i(\bar{x}) = 0$ holds, then $\bar{x}, 1, \ldots, 1$ also activates the path $p$ but it is a contradiction with a constraint 2 in the definition of $b$-OBDD-$\mathbf{PS}_a$ ($(1,+b)$-BP-$\mathbf{PS}_a$). Hence for any input $\bar{x}$ if $\bar{x}, C_1(\bar{x}), \ldots, C_m(\bar{x})$ activates the path $p$ there is $i \in Y_p$ such that $C_i(x_1, \ldots, x_n) = 0$.

We would like to label each 0-sink $v$ of $D$ by a corresponding $Y_p$ (where $p$ is some path from the source to $v$). The problem is that for different paths $p$ and $q$ from the source to the 0-sink two sets $Y_p$ and $Y_q$ may be different. To solve this issue we construct a $b$-OBDD-$\mathbf{PS}_a$ ($(1,+b)$-BP-$\mathbf{PS}_a$) proof $D'$ of $\phi$ such that for any vertex $v$ of $D'$ and two paths $p$ and $q$ from the source of $D'$ to $v$ $Y_p = Y_q$ holds. In order to do it we create $\sum_{i=0}^{a} \binom{n}{i}$ copies of each vertex of the diagram $D$ (the diagram $D'$ splits in a copy of a vertex $v$ by the same variable as in the vertex $v$), each of them is marked by a subset of $[n]$ of size $i \in \{0, \ldots, a\}$. Let $u'$ be a copy of $u$ and $v'$ be a copy of a $v$. Let us assume that $u'$ is marked by $Y^{u'}$ and $v'$ is marked by $Y^{v'}$. If there is an edge from $u$ to $v$, and $Y^{u'} = Y^{v'}$, and $u$ is labeled by $x_i$ for some $i \in [n]$, then we create an edge from $u'$ to $v'$. Additionally, if there is an edge from $u$ to $v$, and $Y^{u'} \cup \{i\} = Y^{v'}$, and $u$ is labeled by $y_i$ for some $i$, then we create an edge from $u'$ to $v'$. Unfortunately, this diagram has more than one source, in order to solve this problem we delete all vertices not reachable from the copy of the source of $D$ marked by the empty set.

Note that for any path $p$ in $D'$ from the source to a vertex $v$ the set $Y_p = Y^v$. Now let us transform this proof into a a $b$-OBDD ($(1,+b)$-BP) $D''$ that returns a subset of $[n]$ of size at most $a$ such that if an input $\bar{x} \in \{0,1\}^n$ and $\bar{y} \in \{0,1\}^m$ activates a path that goes through a vertex $v$ labeled by $y_i$, then $i \in D''(\bar{x}, \bar{y})$ and for any input $\bar{x} \in \{0,1\}^n$ there is $i \in D''(\bar{x}, C_1(\bar{x}), \ldots, C_m(\bar{x}))$ such that $C_i(\bar{x}) = 0$. In order to do it, we copy a diagram $D'$ and say that in each sink $v$ of $D'$ the diagram returns the value $Y^v$.

Finally, we transform $D''$ into a $b$-OBDD ($(1,+b)$-BP) $D'''$ for $a$-Search$_\phi$ by eliminating splittings by the variables $y_i$. For each vertices $u$, $v$, and $w$ such that there is an $l$-edge from $u$ to $v$ ($l \in \{0,1\}$), and $v$ is labeled by $y_i$, and there is a 1-edge from $v$ to $w$, we remove these edges and create an $l$-edge from $u$ to $w$. Let us prove that the resulting diagram is a diagram for $a$-Search$_\phi$. Consider some input $\bar{x} \in \{0,1\}^n$ and a path $p$ in $D''$ that is activated by $\bar{x}, C_1(\bar{x}), \ldots, C_m(\bar{x})$. Note that there is $i \in D''(\bar{x}, C_1(\bar{x}), \ldots, C_m(\bar{x}))$ such that $C_i(\bar{x}) = 0$ and there is a vertex $v$ in the path $p$ labeled by $y_i$. Let us consider the first vertex $u$ in the path $p$ such that $u$ is labeled by a variable $y_j$ such that $C_j(\bar{x}) = 0$. Note that $j \in D'''(\bar{x})$. Also note that size of $D'''$ is at most $\left(\sum_{i=0}^{a} \binom{n}{i}\right) \cdot S$. □

**Corollary 3.1.**
- *The proof system* 1-BP-$\mathbf{PS}_1$ *is polynomially equivalent to regular resolution.*
- *The proof system* OBDD-$\mathbf{PS}_1$ *is polynomially equivalent to ordered resolution.*

*Proof.* The result follows immediately from results in [25]. The paper [25] proved that for any unsatisfiable formula $\phi$ in CNF the minimal size of a regular resolution (ordered resolution) proof of $\phi$ is equal to the minimal size of a 1-BP (OBDD) for Search$_\phi$. Hence using Theorem 3.1 we prove the equivalence. □

**Corollary 3.2.** *Let $b > 0$ and $a > 0$ be constants and $\phi$ be an unsatisfiable formula in CNF. If there is a $b$-OBDD-$\mathbf{PS}_a$ proof of $\phi$ of size $S$, then there is an order $\pi$ and a $(b+1)$-OBDD-$\mathbf{PS}_a$ proof $D$ of $\phi$ such that*

- $|D| = \left(\sum_{i=0}^{a} \binom{n}{i}\right) \cdot (a+2) \cdot S$,
- *$D$ is a $\pi$-$(b+1)$-OBDD,*
- *the variables $x_1, \ldots, x_n$ precedes the variables $y_1, \ldots, y_m$ in the order $\pi$, and*
- *all the variables $y_1, \ldots, y_m$ occur in $D$ only in the last layer.*

*Proof.* Let us consider a $b$-OBDD-$\mathbf{PS}_a$ proof $D''$ of $\phi$ of size $S$. We transform $D''$ into a $b$-OBDD $D'$ for $\mathrm{Search}_\phi$ of size $\left(\sum_{i=0}^{a}\binom{n}{i}\right)S$. After that we replace each sink of $D'$ labeled by $\{i_1,\dots,i_a\}$ by a diagram for $y_{i_1}\wedge\dots\wedge y_{i_a}$ and get a diagram $D$.

It is easy to see that $D$ satisfies constraints of this conjecture. □

Moreover, using Corollary 3.2 we can show that $\oplus$-OBDD-$\mathbf{PS}_a$ polynomially simulates $b$-OBDD-$\mathbf{PS}_a$.

**Theorem 3.2.** *Let $b>0$ be a constant. There is a polynomial $p$ such that if for some CNF $\phi$ there is a $b$-OBDD-$\mathbf{PS}_a$ proof of size $S$, then there is a $\oplus$-OBDD-$\mathbf{PS}_a$ proof of size $p(S)$.*

**Lemma 3.1** ([15])**.** *Let $f:\{0,1\}^n\to\{0,1\}$, $g:\{0,1\}^n\to\{0,1\}$, and $F$ and $G$ be $\oplus$-OBDDs that represent functions $f$ and $g$ respectively. Then*

- *there is a $\oplus$-OBDD that represents $f(x)\wedge g(x)$ of size $|F|\cdot|G|$;*
- *there is a $\oplus$-OBDD that represents $f(x)\oplus g(x)$ of size $|F|+|G|+1$;*

*Proof of Theorem 3.2.* Let $\phi$ be an unsatisfiable formula in CNF, $D$ be its $(b+1)$-OBDD-$\mathbf{PS}_a$ proof with the corresponding order $\pi$ from Lemma 3.2, $D_1,\dots,D_{b+1}$ be a partition of $D$ into layers from the definition of $(b+1)$-OBDD, and $V_1,\dots,V_{b+1}$ be sets of vertices corresponding to $D_1,\dots,D_{b+1}$.

For $v\in V_{b+1}$ let $D(v)$ be an OBDD obtained from $D$ by choosing $v$ as a source and eliminating all vertices not reachable from $v$. For $v\in V_i$ and $w\in V_j$ $(1\le i<j\le b)$, let $D(v,w)$ be an OBDD obtained from $D$ by choosing $v$ as a source, replacing each node except $w$ from layers $V_l$ for $l>i$ by 0-sink and $w$ by 1-sink, and eliminating all nodes not reachable from $v$.

Let $v_1$ be a source of $D$ and consider a consistent path starting at $v_1$ in $D_1$ and leading through layers $1=l(1)<l(2)\dots<l(r)=b+1$ such that layer $D_{l(i)}$ reached the first time at a vertex $v_i$. Note that a condition $\bar z$ is a satisfying input and this path is activated by $\bar x$ is equivalent to a condition that $D(v_i,v_{i+1})$ is satisfied by $\bar z$ for all $1\le i<r$ and $D(v_r)$ is also satisfied by $\bar z$. Since $r$, $l(1),\dots,l(r)$, $v_1,\dots,v_r$ are fixed by the path and the path is fixed by the input $\bar z$ we have the following claim.

**Claim 3.2.1.** *An input $\bar z$ is satisfiable input of $D$ iff there are unique $r$, $1=l(1)<l(2)\dots<l(r)=b+1$, $v_2\in V_{l(2)},\dots,v_r\in V_{l(r)}$ such that $\bigwedge\limits_{1\le i<r}D(v_i,v_{i+1})\wedge D(v_r)$ is satisfied by $\bar z$.*

Also note that number of different $r$, $v_1,\dots,v_r$ is bounded by $(k+1)\cdot|D|^b$ and for each choice of them the $\pi$-OBDDs $D(v_1,v_2),\dots,D(v_{r-1},v_r)$, and $D(v_r)$ have size at most $|D|$. Hence if we consider $\bigwedge\limits_{1\le i<r}D(v_i,v_{i+1})$ by Lemma 3.2 we get a $\pi$-OBDD of size at most $|D|^b$, additionally note that $\bigwedge\limits_{1\le i<r}D(v_i,v_{i+1})$ does not split by the variables $y_1,\dots,y_m$. Since all the variables $y_1,\dots,y_m$ precedes the variables $x_1,\dots,x_n$ in the order $\pi$ and $D$ is a $b$-OBDD-$\mathbf{PS}_1$ proof of $\phi$, an OBDD representation of

$$\bigwedge_{1\le i<r}D(v_i,v_{i+1})\wedge D(v_r)$$

has size at most $|D|^{b+1}$ and the variables $y_1,\dots,y_m$ occur at most $a$ times in total on any path in this diagram.

As a result, by Claim 3.2.1 and by Lemma 3.2 a $\oplus$-OBDD representation of

$$\bigoplus_{r,v_1,\dots,v_r}\left(\bigwedge_{1\le i<r}D(v_i,v_{i+1})\wedge D(v_r)\right)$$

is a $\oplus$-OBDD-$\mathbf{PS}_a$ proof of $\phi$ of size $O(|D|^{2b+2})$. □

# 4 Lower Bounds via Best-Communication Complexity

In this section we prove a lower bound for $b$-OBDD-$\mathbf{PS}_1$ and technique we use for the lower bound we prove a separation between $b$-OBDD-$\mathbf{PS}_1$ and 1-BP-$\mathbf{PS}_1$.

## 4.1 Lower Bound for $b$-OBDD-$\mathbf{PS}_1$

**Theorem 4.1.** *There is a family $\{\phi_n\}_{n\in\mathbb{N}}$ of formulas in CNF such that for any $b > 0$*

- $|\phi_n| = \mathrm{poly}(n)$;
- *any $b$-OBDD-$\mathbf{PS}_1$ proof of $\phi_n$ has size at least $2^{\Omega\left(\frac{n}{b\log n}\right)}$.*

For proving this theorem we need to introduce a notion of communication complexity introduced by Yao [38]. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. We have two players called Alice and Bob, who have to compute $f(s)$ with partition $\Pi = (\Pi_0, \Pi_1) \subseteq [n]^2$ ($\Pi_0 \cap \Pi_1 = \emptyset$). The function $f$ is known by both of them. However Alice knows only bits of $s$ with indices from $\Pi_0$ and Bob knows only bits of $s$ with indices from $\Pi_1$. They have a two-sided communication channel. On each round of their communication one of them sends a string and Alice and Bob are trying to minimize two parameters: total number of sent bits and number of rounds.

In a more general situation, we have a relation $R \subseteq \{0,1\}^n \times Z$ and Alice knows bits with indices from $\Pi_0$ and Bob knows with indices from $\Pi_1$ and they wish to find $z \in Z$ such that $(s, z) \in R$.

More formally, a communication protocol is a tree $T$ where each internal node $v$ is labeled either by a function $a_v : \{0,1\}^n \to \{0,1\}^{l_v}$ depending on bits with indices from $\Pi_0$ or by a function $b_v : \{0,1\}^n \to \{0,1\}^{l_v}$ depending on bits with indices from $\Pi_1$, each leaf is labeled by an element $z \in Z$, each node has $2^{l_v}$ children, and each edge is labeled by a string from $\{0,1\}^{l_v}$. The value of the protocol $T$ on input $s$ is the label of the leaf reached starting from the root, and walking on the tree, at each internal node labeled by $m$ we go by the edge labeled by $m(s)$.

The cost of the protocols on input $s$ is the sum of lengths of strings on edges from the path, and the number of rounds on input $s$ is the length of the path. The cost (number of rounds) of $T$ is a maximum of costs (numbers of rounds) over all inputs.

The communication complexity denoted $\mathbf{D}^{\Pi}(R)$ of a relation $R$ is the cost of the best protocol for this relation with partition $\Pi$. Additionally, we denote by $\mathbf{D}^{(k),\Pi}(R)$ the cost of the best protocol with $k$ rounds for this relation with partition $\Pi$.

Sometimes, we consider communication complexity of functions $f : A \times B \to \Lambda$, in this case we omit partitions and assume that Alice knows value $a \in A$ and Bob knows value $b \in B$.

It is easy to see that for any relation $R \subseteq \{0,1\}^n \times Z$, partition of input bits $\Pi$, and order $\pi$ over variables $x_1, \ldots, x_n$ such that for any $i \in \Pi_0$ and $j \in \Pi_1$ $i$ precedes $j$ in order $\pi$, $\mathbf{D}^{\Pi}(R) \leq \lceil \log S \rceil$, where $S$ is the size of the smallest $\pi$-OBDD for $R$. Indeed, Alice simulates first $|\Pi_0|$ variables of the $\pi$-OBDD for $R$, she then sends a description of a node from which the simulation should continue and Bob can finish the simulation and compute the output.

Unfortunately, if we do not fix an order, classical communication complexity is not enough for describing OBDD complexity. Let us consider for instance a function $\mathtt{EQ}_n : \{0,1\}^{2n} \to \{0,1\}$ such that $\mathtt{EQ}_n(x, y) = 1$. It is known that communication complexity of $\mathtt{EQ}_n$ with partition $(\{1, \ldots, n\}, \{n+1, \ldots, 2n\})$ is equal to $n + 1$, but there is an OBDD of size $2n$ for it.

However, if we consider the best-communication complexity, then the situation would be much better. The best-communication complexity $\mathbf{D}^{best}(R)$ of a relation $R \subseteq \{0,1\}^{2n} \times Z$ is the minimal communication complexity over all the partitions $\Pi$ of input bits between Alice and Bob such that $|\Pi_0| = |\Pi_1| = n$. Besides, we denote by $\mathbf{D}^{(k),best}(R)$ the best-communication complexity with $k$ rounds.

**Lemma 4.1** (see for example [23, Lemma 12.12])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. For any $b > 0$, if there is an $b$-OBDD of size $S$ for $f$, then $\mathbf{D}^{best}(f) \leq (2b-1)\lceil \log S \rceil$.*

For proving lower bounds for $b$-OBDD-$\mathbf{PS}_1$ it is sufficient to construct a formula $\phi$ in CNF such that $\mathrm{Search}_\phi$ has big best-communication complexity. Our construction is based on a CNF $\phi$ such that $\mathrm{Search}_\phi$ has big communication complexity for at least one partition.

Such a result was proven in the paper [16]. Actually the have proven stronger lower bound, they have proven a lower bound for probabilistic communication complexity. However, deterministic complexity is enough for our case.

**Lemma 4.2** ([16]). *There is a family $\phi_n$ of formulas in $k$-CNF ($k$ is a constant) such that*

- $|\phi_n| = \mathrm{poly}(n)$ *and* $\phi_n$ *depends on* $O(n)$ *variables;*
- *there is a partition* $\Pi$ *of variables of* $\phi$ *such that* $\mathbf{D}^{\Pi}(\mathrm{Search}_\phi) = \Omega\left(\frac{n}{\log n}\right)$

Now we need to transform this formula to a formula with large best-communication complexity of the corresponded search problem. There is a construction of such a transformation introduced by Lam and Ruzzo [24], but their transformation does not fit out our situation. In construction of Lam and Ruzzo it is necessary that hard problem is paddable, but $\mathrm{Search}_\phi$ for $\phi$ from Lemma 4.2 is not paddable. So we introduce a new transformation based on ideas of Segerlind [32].

**Theorem 4.2.** *There is a transformation $\mathcal{F}$ of CNFs into CNFs such that for any large enough $n$, constant $k$, formula $\phi$ in $k$-CNF depending on $n$ variables, and partition of inputs* $\Pi$, $\mathbf{D}^{best}\left(\mathrm{Search}_{\mathcal{F}(\phi)}\right) \geq \mathbf{D}^{\Pi}(\mathrm{Search}_\phi)$ *and* $|\mathcal{F}(\phi)| = \mathrm{poly}(n, |\phi|)$ *hold.*

Idea of the transformation may be explained in three steps. Let $\phi$ be an unsatisfiable formula in CNF with variables $x_1, \ldots, x_n$.

1. Let us consider the following transformation:

$$\mathrm{perm}_{S_n}(\phi)(z_1, \ldots, z_l, x_1, \ldots, x_n) = \bigwedge_{\alpha \in S_n} (\mathrm{enc}(z_1, \ldots, z_l) = \alpha) \to \phi(x_{\alpha(1)}, \ldots, x_{\alpha(n)}),$$

   where $S_n$ is a set of permutations over $[n]$, $l = \lceil \log |S_n| \rceil$ and $\mathrm{enc} : \{0,1\}^l \to S_n$ is some surjective function. If $\Pi = (\Pi_0, \Pi_1)$ is a partition of the variables $x_1, \ldots, x_n$, then for any partition $\Gamma$ of the variables $x_1, \ldots, x_n, z_1, \ldots, z_l$ such that $|\Pi_0| = |\{x_i : x_i \in \Gamma_0\}|$ holds $\mathbf{D}^{\Gamma}\left(\mathrm{Search}_{\mathrm{perm}_{S_n}(\phi)}\right) \geq \mathbf{D}^{\Pi}(\mathrm{Search}_\phi)$. Indeed, let $\alpha$ be a permutation over $[n]$ such that $\Pi_0 = \alpha(\{x_i : x_i \in \Gamma_0\})$; we consider a protocol for $\mathrm{Search}_{\mathrm{perm}_{S_n}(\phi)}$ and run this protocol for some values of the variables $x_1, \ldots, x_n$ and the values of the variables $z_1, \ldots, z_l$ such that $\mathrm{enc}(z_1, \ldots, z_l) = \alpha$. By definition we get a protocol for $\mathrm{Search}_\phi$.

   The problem of this construction is the length of $\mathrm{perm}_{S_n}(\phi)$ it is exponential in $n$.

2. Second idea is to consider some "good" set of permutations $P$ instead of $S_n$. Unfortunately, in this case permutation $\alpha$ such that $\Pi_0 = \alpha(\{x_i : x_i \in \Gamma_0\})$ does not necessary belongs to $P$.

3. Finally, third idea is to replace variables of $\phi$ by or of many fresh variables. Consider

$$\mathrm{perm}_P(\phi)(z_1, \ldots, z_l, y_1, \ldots, y_{nm}) = \bigwedge_{\alpha \in P} \left[ (\mathrm{enc}(z_1, \ldots, z_l) = \alpha) \to \phi\left( \bigvee_{i=1}^{m} y_{\alpha(i)}, \ldots, \bigvee_{i=(n-1)m+1}^{nm} y_{\alpha(i)} \right) \right],$$

   where $P \subseteq S_{nm}$, $l = \lceil \log P \rceil$, and $\mathrm{enc} : \{0,1\}^l \to P$ is some surjective function.

Now let us explain how to encode the result of this transformation in CNF. First of all, we need to define a composition of formulas. Let $\phi$ be a $k_1$-CNF depending on $x_1, \ldots, x_n$ and $g$ be a $k_2$-CNF depending on $m$ variables. We denote by $\phi \diamond g$ a $k_1 \cdot k_2$-CNF of size $|g|^{k_1}|\phi|$ depending on variables the $y_{1,1}, \ldots, y_{m,n}$ such that $\phi \diamond g$ is a result of substitutions $x_i$ for $g(y_{i,1}, \ldots, y_{i,m})$ and applications of distributive law. If $g = \bigvee_{i=1}^{m} y_i$, then we denote $\phi \diamond g$ by $\phi^m$.

Let $t \in \mathbb{N}$ be given, $\mathbb{F}$ be a field of size $2^t$. Define the set $P_t$ to be a set of all mappings given by $x \mapsto ax+b$ with $a, b \in \mathbb{F}$ and $a \neq 0$. Let $\phi$ be a CNF in the variables $x_1, \ldots, x_n$. Note that $P_t$ is not a subset of $S_n$, but subset of $S_{2^t}$, hence we have to add fresh variables. Let us consider $t = \lceil \log n \rceil$ and $x_{n+1}, \ldots, x_{2^t}$, $z_1, \ldots, z_{2t}$ are new variables. We use $z_1, \ldots, z_{2t}$ to encode the permutations of $P_t$ in some surjective manner. The CNF $\mathrm{perm}(\phi)$ is the CNF obtained as follows: for each assignment $\alpha \in \{0,1\}^{2t}$ to $z_1, \ldots, z_{2t}$, let $\pi$ denotes permutation encoded by $\alpha$. For every clause $\bigvee_{i \in I} x_i^{\sigma_i}$ from $\phi$ there is a clause $\bigvee_{i \in [2t]} z_i^{1-\alpha_i} \vee \bigvee_{i \in I} x_{\pi(i)}^{\sigma_i}$ in $\mathrm{perm}(\phi)$.

For proving this theorem we also need two technical results.

**Lemma 4.3** ([37]). *For any $t$, $|P_t| = 2^t \cdot (2^t - 1)$, every mapping from $P_t$ 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 $\Pr_{\pi \in P_t}[\pi(x_1) = y_1, \pi(x_2) = y_2] = \frac{1}{2^t(2^t-1)}$.*

**Lemma 4.4** (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 4.2.* Let $m = 100n$, $\mathcal{F}(\phi) = \mathrm{perm}(\phi^m)$, and $N$ be a number of the variables in $\phi^m$. Fix two arbitrary balanced partitions $\Gamma$ and $\Pi$ of the variables $\mathcal{F}(\phi)$ and $\phi$ respectively. We prove that if there is a protocol for $\mathrm{Search}_{\mathrm{perm}(\phi^m)}$ and the partition $\Gamma$ with communication complexity $S$, then there is a protocol for $\mathrm{Search}_{\phi}$ and the partition $\Pi$ with communication complexity $S$.

Let us assume that there is a permutation $\pi \in P_{\lceil \log N \rceil}$ such that for any $i \in [n]$ and $k \in \{0,1\}$ there is $j$, such that $y_{i,j}$ is mapped to a variable from $\Gamma_k$ by $\pi$.

Let $V = \{v_1, \ldots, v_{2\lceil \log N \rceil}\}$ be a set of the variables of $\mathrm{perm}(\phi^m)$ except variables $z_1, \ldots, z_{2\lceil \log N \rceil}$ encoding permutation and let for every $i \in [n]$ and $k \in \{0,1\}$, $v_{r(i,k)}$ denotes some variable $y_{i,j}$ that is mapped to a variable from $\Gamma_k$ by the permutation $\pi$. The protocol will be the following: on input $x_1, \ldots, x_n$ it runs the protocol for $\mathrm{Search}_{\mathrm{perm}(\phi^m)}$ with substitution to input values such that $z_1, \ldots, z_{2\lceil \log N \rceil}$ encode $\pi$, all the variables from $V \setminus \{v_{r(i,k)} : i \in [n], k \in \{0,1\}\}$ are equal to zero, and

1. $v_{r(i,k)} = x_i$ and $v_{r(i,1-k)} = 0$ if $x_i \in \Pi_k$ and
2. $v_{r(i,k)} = 0$ and $v_{r(i,1-k)} = x_i$ otherwise.

It is easy to see that the communication complexity of this protocol is equal to $S$ on the partition $\Gamma$. Thus, $\mathbf{D}^{\Pi}(\mathrm{Search}_{\phi}) \leq \mathbf{D}^{\Gamma}\left(\mathrm{Search}_{\mathrm{perm}(\phi^m)}\right)$.

In the rest we prove existence of such a permutation $\pi$.

In the following we denote $v_{\pi(i)}$ by $\pi(v_i)$. Let $\Gamma'$ be a partition induced by $\Gamma$ on $V$. Note that $\Gamma'$ is almost balanced i.e. $\frac{N}{2} \geq |\Gamma'_k| \geq \frac{N}{2} - 2\lceil \log N \rceil$.

Choose uniformly random $\pi \in P_{\lceil \log N \rceil}$ and for $k \in \{0,1\}$ let us consider random variables such that $\chi^k_{i,j} = 1$ iff $y_{i,j}$ is mapped by $\pi$ into $\Gamma'_k$ and $Y^k_i = \sum_{j=1}^{m} \chi^k_{i,j}$.

By Lemma 4.3 $\chi^k_{i,j}$ has expectation equals $\frac{|\Gamma'_k|}{N}$ and by additivity of expectation, expectation of $Y^k_i$ is

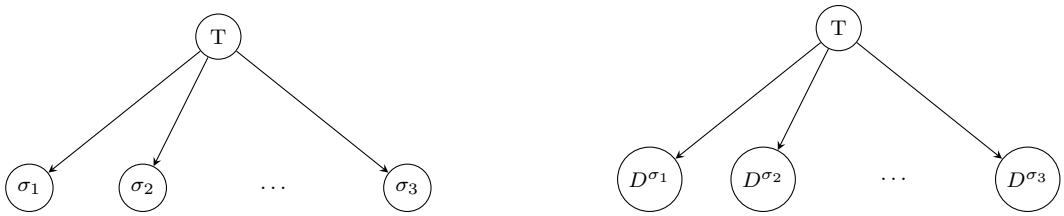



Figure 3: On the right side presented a diagram for $\mathrm{perm}(\phi)$ where $T$ is a complete decision tree for enc and $D^\sigma$ is a $(1,+k)$-BP for $a$-$\mathrm{Search}_{\phi(x_{\sigma(1)},\dots,x_{\sigma(n)})}$

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 4.4

$$\begin{aligned}\Pr[Y^k_i=0] &\le \frac{\mathbb{E}[Y^k_i] + \sum\limits_{i\neq j\in[n]}\mathrm{Cov}(\chi^k_{i,j_0},\chi^k_{i,j_1})}{\left(\mathbb{E}[Y^k_i]\right)^2}\\ &\le \frac{N}{m|\Gamma'_k|} + \frac{m(m-1)}{m^2(N-1)} \le \frac{N}{m\left(\frac{N}{2}-2\log N\right)} + \frac{1}{N-1}\\ &\le \frac{N}{m\frac{N}{4}} + \frac{1}{N-1} = \frac{4}{m} + \frac{1}{nm-1}\end{aligned}$$

Therefore, by union bound $\Pr\left[\exists i,k\ Y^k_i=0\right] \le \frac{8n}{m} + \frac{2n}{nm-1} \le 1$. As a result, there is a permutation $\pi \in P_{\lceil \log N\rceil}$ such that for any $i\in[n]$ and $k\in\{0,1\}$ there is a $j\in[m]$ and $v_{r(i,k)}\in\Gamma'_k$ such that $v_{r(i,k)} = \pi\left(y_{i,j}\right)$. □

*Proof of the Theorem 4.1.* Let $\phi_n$ be a family of formulas from Lemma 4.2. By Lemma 4.1 and Theorem 4.2 there is $\mathcal{F}$ such that any $k$-OBDD for $\mathrm{Search}_{\mathcal{F}(\phi_n)}$ has size $2^{\Omega\left(\frac{n}{k\log n}\right)}$ and $|\mathcal{F}(\phi_n)| = \mathrm{poly}(n)$.

As a result, any $b$-OBDD-$\mathbf{PS}_1$ proof of $\mathcal{F}(\phi_n)$ has size $2^{\Omega\left(\frac{n}{b\log n}\right)}$ by Theorem 3.1. □

## 4.2 Separation Between $b$-OBDD-$\mathbf{PS}_1$ and 1-BP-$\mathbf{PS}_1$

It is easy to see that in previous theorem $\mathrm{perm}(\phi^m)$ is hard for $k$-OBDD-$\mathbf{PS}_1$ mostly because for different substitutions for $z_1,\dots,z_l$ we need to use different order on variables. However, for such proof systems as 1-BP-$\mathbf{PS}_1$ it is not a problem. Using this difference between these two proof systems we may separate them.

**Theorem 4.3.** *There are a family $\{\phi_n\}_{n\in\mathbb{N}}$ of formulas in CNF and a constant $c>0$ such that*

- $|\phi_n| = \mathrm{poly}(n)$ *and* $\phi_n$*;*

- *any* $k$-OBDD-$\mathbf{PS}_1$ *proof of* $\phi_n$ *has size* $2^{\Omega(n^c)}$*;*
- *there is a* 1-BP-$\mathbf{PS}_1$ *proof of* $\phi_n$ *of size* $\mathrm{poly}(n)$*.*

At first prove that $\mathrm{perm}(\phi)$ is not harder than $\phi$ for $(1,+b)$-BP-$\mathbf{PS}_a$.

**Theorem 4.4.** *Let* $\phi$ *be a CNF on variables* $x_1, \ldots, x_n$ *and* $k$ *be a constant. If there is a proof of* $\phi$ *in* $(1,+b)$-BP-$\mathbf{PS}_a$ *of size* $S$*, then there is a proof of* $\mathrm{perm}(\phi)$ *of size* $\mathrm{poly}(S,n)$*.*

*Proof.* Let $D$ be a $(1,+k)$-BP for $a$-Search$_\phi$ of size $S$. It is easy to see that there are diagrams $D^\pi$ for $a$-Search$_{\phi(x_{\pi(1)},\ldots,x_{\pi(n)})}$ of size $S$ (it may be constructed from $D$ by replacement $x_i$ by $x_{\pi(i)}$ for all $i \in \{1,\ldots,n\}$).

Now let us construct a $(1,+b)$-BP for $a$-Search$_{\mathrm{perm}(\phi)}$. Let $t = \lceil \log(n) \rceil$ and $m = |P_t|$. Let us consider a complete decision tree for the function enc and replace leaf of this tree labeled by $\pi$ by the root of a diagram $D^\pi$ (see Figure 3).

It is easy to prove that this is a diagram for $a$-Search$_{\mathrm{perm}(\phi)}$ and its size is at most $2^{2\lceil \log(n) \rceil} S = \mathrm{poly}(S,n)$. So by Theorem 3.1 there is a $(1,+b)$-BP-$\mathbf{PS}_a$ proof of $\mathrm{perm}(\phi)$ of size $\mathrm{poly}(n,S)$. □

Now we need a formula $\phi$ and a partition $\Pi$ such that $\mathbf{D}^\Pi(\mathrm{Search}_\phi)$ however, $\phi^m$ is not hard for 1-BP-$\mathbf{PS}_1$ for any $m$. For this reason we consider the pebbling contradictions. *Pebbling contradiction* $\mathtt{Peb}_G$ for a directed acyclic graph $G$ (for simplicity we consider such a graph with only one sink $t$, vertex with out degree 0) is a conjunction of the following clauses:

**sink axiom:** $\neg x_t$;

**propagation axiom:** $\left(\bigwedge_{i=1}^{l} x_{p_i}\right) \to x_v$, where $p_1, \ldots, p_l$ is a least of all predecessors of $v$ for every $v$.

In 2014 Göös and Pitassi have proven that there is a gadget $g$ such that the communication complexity of Search$_{\mathtt{Peb}_G \diamond g}$ is high.

**Lemma 4.5** ([16])**.** *There is a family* $G_n$ *of graphs of constant degree* $d$ *and a CNF* $g$ *such that there is a partition* $\Pi$ *of variables of* $\mathtt{Peb}_G \diamond g$ *such that* $\mathbf{D}^\Pi\left(\mathrm{Search}_{\mathtt{Peb}_{G_n} \diamond g}\right) = \Omega\left(\sqrt{n}\right)$ *and* $G_n$ *is a graph with* $O(n)$ *vertices.*

However, using techniques similar to techniques from [9] we can show that for any CNF $g$ and graph $G$ there is an OBDD-$\mathbf{PS}_1$ proof of $\mathtt{Peb}_G \diamond g$.

**Theorem 4.5.** *For any oriented graph* $G$ *on* $n$ *vertices with maximal degree not greater than* $d$ *and CNF* $g$ *depends on* $k$ *variables there is an* OBDD-$\mathbf{PS}_1$ *proof of* $\mathtt{Peb}_G \diamond g$ *of size* $\mathrm{poly}(n, 2^d, 2^k)$*.*

We prove this Theorem using equivalence between OBDD-$\mathbf{PS}_1$ and ordered resolution.

**Lemma 4.6.** *For any graph* $G$ *on* $n$ *vertices with maximal degree not greater than* $d$*, CNF* $g$ *depends on* $k$ *variables,* $v$ *be a vertex of* $G$*, and* $p_1, \ldots, p_l$ *all predecessors of* $v$*.*

*For any order over variables of* $\mathtt{Peb}_G \diamond g$ *and clause* $C$ *of* $x_v \diamond g$ *there is a polynomial size ordered resolution derivation of* $C$ *from clauses of formulas* $x_{p_1} \diamond g, \ldots, x_{p_l} \diamond g$*, and* $\left(\left(\bigwedge_{i=1}^{l} x_{p_i}\right) \to x_v\right) \diamond g$ *respecting this order. Moreover, this derivation does not resolve on the variables of* $x_v \diamond g$*.*

*Proof.* Consider the clauses of formulas $x_{p_1} \diamond g, \ldots, x_{p_l} \diamond g$, and $\left(\bigwedge_{i=1}^{l} \neg x_{p_i}\right) \diamond g$. Their conjunction us unsatisfiable, hence there is an ordered resolution refutation of size $2^{dk}$.

If we add literals of the clause $C$ to clauses of $\left(\bigwedge_{i=1}^{l} \neg x_{p_i}\right) \diamond g$ and their descendants in the proof, then we get a proof of $C$ that does not resolve on the variables of $\mathtt{Peb}_G \diamond g$. □

*Proof of Theorem 4.5.* Let $t$ be the only sink of the graph $G$. Let us fix an order over the variables of $\mathtt{Peb}_G \diamond g$ that respects a topological ordering of a graph $G$. To construct the refutation, we start with the graph $G$. Label each source $v$ of G with the axiom $x_v \diamond g$. For each non-source vertex $v$ of $G$ with predecessors $p_1, \ldots, p_l$, replace $v$ with the ordered derivation of $x_v \diamond g$ guaranteed by Lemma 4.6 that respects the ordering. The result is an ordered derivation of the clause $x_t \diamond g$ that has not resolved on the variables of $x_t \diamond g$. Resolve this clause with the axioms $\neg x_t \diamond g$ such a way that respects the fixed ordering. □

*Proof of Theorem 4.3.* Let us consider a family graphs $G_n$ and a function $g$ from Lemma 4.5. By Theorem 4.5 $\mathbf{D}^{best}\left(\mathrm{Search}_{\mathtt{Peb}_{G_n} \diamond g^m}\right) = \Omega(\sqrt{n})$. Hence, because of equivalence between the complexity of $\mathrm{Search}_{\mathrm{perm}(\mathtt{Peb}_{G_n} \diamond g)}$ and the $b$-OBDD-$\mathbf{PS}_1$ proof complexity of $\mathrm{perm}(\mathtt{Peb}_{G_n} \diamond g^m)$ the size of any $b$-OBDD-$\mathbf{PS}_1$ proof of $\mathrm{perm}(\mathtt{Peb}_{G_n} \diamond g^m)$ is at least $2^{\Omega(\sqrt{n})}$. However, by Theorem 4.5 and Theorem 4.4 there is a 1-BP-$\mathbf{PS}_1$ proof of $\mathrm{perm}(\mathtt{Peb}_{G_n} \diamond g^m)$ of size at most $\mathrm{poly}(n)$. □

Additionally, using this techniques we can show separation between OBDD-$\mathbf{PS}_1$ and $\mathbf{TreeTh}(k)$.

**Corollary 4.1.** *There are a family $\{\phi_n\}_{n\in\mathbb{N}}$ of formulas in CNF and a constant $c > 0$ such that*

- $|\phi_n| = \mathrm{poly}(n)$;
- *any* $\mathbf{TreeTh}(k)$ *proof of* $\phi_n$ *has size* $2^{\Omega(n^c)}$;
- *there is an* OBDD-$\mathbf{PS}_1$ *proof of* $\phi_n$ *of size* $\mathrm{poly}(n)$.

*Proof.* Göös and Pitassi have proven [16] that there is a family of graphs $G_n$ on $n$ vertices with constant degree and a function $g$ such that any $\mathbf{TreeTh}(k)$ proof of $\mathtt{Peb}_{G_n} \diamond g$ has size at least $2^{\Omega(n^c)}$ for some $c > 0$. In the same time by Theorem 4.5 there is a proof of $\mathtt{Peb}_{G_n} \diamond g$ of size at most $\mathrm{poly}(n)$. Hence we can choose $\phi = \mathtt{Peb}_{G_n} \diamond g$. □

# 5 Communication Complexity with Bounded Number of Rounds

In the previous section we proved several lower bounds for $b$-OBDD-$\mathbf{PS}_a$, unfortunately, these formulas are artificial. In this section we prove lower bounds for unsatisfiable Tseitin formulas.

For a graph $G$, in the sequel we denote by $V(G)$ the set of vertices of $G$ and by $E(G)$ the set of edges of $G$. Additionally, if $M \subseteq E(G)$ we denote by $G[M]$ the graph whose vertex set is $V(G)$ and edge set is $M$.

Let $G$ be an undirected graph with degree bounded by a constant $d$ and $c : V(G) \to \{0, 1\}$ be a function, we call it the labelling function, then a Tseitin formula $\mathtt{TS}_{G,c}$ for the graph $G$ and the labelling function $c$ be the following. Every edge $e \in E(G)$ has the corresponding propositional variable $p_e$ (in this section we assume that $G$ does not contain loops). For every vertex $v \in V$ we write down a formula in CNF that encodes $\sum_{u\in V(G):(u,v)\in E(G)} p_{(u,v)} \equiv c(v) \pmod 2$. The conjunction of the formulas described above is called a Tseitin formula. If $\sum_{v\in U} c(v) \equiv 1 \pmod 2$ for some connected component $U \subseteq V(G)$, then the Tseitin formula is unsatisfiable. Indeed, if we sum up all equalities stated in the vertices from $U$ we get $0 \equiv 1 \pmod 2$ since every variable has exactly 2 occurrences. If $\sum_{v\in U} c(v) \equiv 0 \pmod 2$ for every connected component $U$, then the Tseitin formula is satisfiable ([34, Lemma 4.1]).

However, the communication complexity method does not work for Tseitin formulas.

**Proposition 5.1.** *Let $G$ be a graph with a constant degree, $c : V(G) \to \{0, 1\}$ be a labeling function, and $\Pi$ be a partition of the variables of $\mathtt{TS}_{G,c}$. If $\mathtt{TS}_{G,c}$ is unsatisfiable, then $\mathbf{D}^{\Pi}\left(\mathrm{Search}_{\mathtt{TS}_{G,c}}\right) = O(\log(|V(G)|))$.*

*Proof.* Let us consider a protocol with $l = O(\log(|V|))$ stages such that on the stage with number $i$ Alice and Bob consider a subset $V_i \subseteq V(G)$ such that $\sum_{v\in V_i, u\in V:(v,u)\in E(G)} p_{(v,u)} \neq \sum_{v\in V_i} c_v$ and $|V_l| = 1$ hold.

Initially $V_1 = V(G)$ and if $|V_i| \neq 1$ Alice and Bob split $V_i$ into two arbitrary parts $V_i^1$ and $V_i^2$ such that $0 \leq |V_i^1| - |V_i^2| \leq 1$ and if $\sum\limits_{v \in V_i^1, u \in V(G): (v,u) \in E(G)} p_{(v,u)} = \sum\limits_{v \in V_i^1} c_v$ set $V_{i+1} = V_i^2$, otherwise set $V_{i+1} = V_i^1$.

Note that if Alice and Bob know the variable $v$ such that $\sum\limits_{u \in V(G): (v,u) \in E(G)} p_{(v,u)} \neq c_v$, then the can find a falsified clause using constant communication.

Additionally, it is easy to see that regardless of the partition of the variables Alice and Bob can check the equality $\sum\limits_{v \in V_i^1, u \in V(G): (v,u) \in E(G)} p_{(v,u)} = \sum\limits_{v \in V_i^1} c_v$ using two bits of communication. As a result, this is a protocol for $\mathrm{Search}_{\mathtt{TS}_{G,c}}$ with the cost equal to $O(\log(|V(G)|))$. □

Nevertheless, if we bound number of round in protocols we can get a lower bound for $\mathrm{Search}_{\mathtt{TS}_{G,c}}$.

**Lemma 5.1** ([23, Lemma 12.12])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. For any $b > 0$, if there is a $b$-OBDD representation of $f$ of size $S$, then $\mathbf{D}^{(2b-1),best}(f) \leq (2b-1)\lceil \log(S) \rceil$.*

Now, if we prove a lower bound for the communication complexity of $\mathrm{Search}_{\mathtt{TS}_{G,c}}$ with bounded number of rounds, using equivalence between $b$-OBDD complexity of $\mathrm{Search}_{\mathtt{TS}_{G,c}}$ and $b$-OBDD-$\mathbf{PS}_1$ proof complexity of $\mathtt{TS}_{G,c}$ we prove a lower bound for the size of $b$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G,c}$.

**Theorem 5.1.** *There is a family $\{G_n\}_{n \in \mathbb{N}}$ of graphs with degree less than $d$ and a family of labeling functions $\{c_n : V(G_n) \to \{0,1\}\}_{n \in \mathbb{N}}$ such that any $b$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G_n,c_n}$ has size at least $2^{\Omega\left(n^{1/(2b-1)}\right)}$.*

To prove this theorem we consider Karchmer–Wigderson communication games [21]. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. By $\mathrm{KW}(f)$ we denote the Karchmer–Wigderson game for the function $f$. The Karchmer–Wigderson game for the function $f$ is the relation $R \subseteq f^{-1}(0) \times f^{-1}(1) \times [n]$ such that

$$(x, y, i) \in R \iff x_i \neq y_i.$$

By $\mathbf{L}^{(b)}(f)$ we denote the minimal size of a formula computing $f$ with $b$ alternating levels of unbounded fan-in AND and OR gates. Nissan and Wigderson [28] in 1993 noticed that $\mathbf{D}^{(b)}(\mathrm{KW}(f)) = \Theta(\log(\mathbf{L}^{(b)}(f)))$.

**Theorem 5.2** ([18])**.** *Let $\oplus_n : \{0,1\}^n \to \{0,1\}$ be the parity function. For any $b > 1$ holds $\mathbf{D}^{(b)}(\mathrm{KW}(\oplus_n)) = \Omega\left(n^{1/(b-1)}\right)$.*

*Proof.* Håstad in 1983 [18] have proven that for any $b > 1$, $\mathbf{L}^{(b)}(\oplus_n) = 2^{\Omega\left(n^{1/(b-1)}\right)}$. Using connection between $\mathbf{L}^{(b)}(\oplus_n)$ and $\mathbf{D}^{(b)}(\mathrm{KW}(\oplus_n))$ we get the desired statement. □

**Theorem 5.3.** *There are a family of graphs $\{G_n\}_{n \in \mathbb{N}}$ of constant degree and a family of labeling functions $\{c_n : V(G_n) \to \{0,1\}\}_{n \in \mathbb{N}}$ such that $\mathtt{TS}_{G_n,c_n}$ is unsatisfiable for any $n$ and $\mathbf{D}^{(b),best}\left(\mathrm{Search}_{\mathtt{TS}_{G_n,c_n}}\right) = \Omega(n^{1/(b-1)})$.*

Before we start proving this theorem let us illustrate its proof on a simpler example.

Let $G$ be a grid $n \times n$, $c : V(G) \to \{0,1\}$ be a labeling function such that only the left top corner of $G$ is labeled by 1, and $\Pi$ be a partition such that Alice knows the variables corresponding to the lower triangle (edges to the left to the main diagonal of $G$) and Bob knows the variables corresponding to the upper triangle. We prove that $\mathbf{D}^{(b),\Pi}\left(\mathrm{Search}_{\mathtt{TS}_{G,c}}\right) \geq \mathbf{D}^{(b)}(\mathrm{KW}(\oplus_n))$ by constructing a reduction $\mathrm{KW}(\oplus_n)$ to $\mathrm{Search}_{\mathtt{TS}_{G,c}}$. Let $A \in \oplus_n^{-1}(0)$ and $B \in \oplus_n^{-1}(1)$ be Alice and Bob's inputs of $\mathrm{KW}(\oplus_n)$, we construct without communication an instance $(x, y)$ of $\mathrm{Search}_{\mathtt{TS}_{G,c}}$ (with respect to the partition $\Pi$) such that Alice and Bob can obtain an answer for $\mathrm{KW}(\oplus_n)(A, B)$ from the answer of $\mathrm{Search}_{\mathtt{TS}_{G,c}}(x, y)$ without communication.

Let $v_1, \ldots, v_n$ be all the vertices from the main diagonal of $G$, and for each $i, j \in [n]$, $q^u_{i,j}$ be a path from $v_i$ to $v_j$ in the upper triangle and $q^\ell_{i,j}$ be a path from $v_i$ to $v_j$ in the lower triangle.

We construct an instance $(x, y)$ in three steps.

- Initially, Alice and Bob have all zero instance $(x, y)$ of $\mathrm{Search}_{\mathtt{TS}_{G,c}}$.

- Note that $\sum_{i=1}^{n} A_i \equiv 0 \pmod 2$, since $A \in \oplus_n^{-1}(0)$. Hence, the set $\{i \in [n] \ : \ A_i = 1\}$ has even size and Alice can split it into pairs $(l_1, r_1), \ldots, (l_t, r_t)$. After that Alice inverts all the edges along the paths $q^u_{l_1,r_1}, \ldots, q^u_{l_t,r_t}$ (it is possible to do without communication since paths $q^u_{i,j}$ are paths in the upper triangle).

- Similarly $\sum_{i=1}^{n} B_i \equiv 1 \pmod 2$, since $B \in \oplus_n^{-1}(1)$. Hence, the set $\{i \in [n] \ : \ B_i = 1\} \Delta \{1\}$ has even size and Bob can split it into pairs $(l_1, r_1), \ldots, (l_t, r_t)$. After that Bob inverts all the edges along the paths $q^\ell_{l_1,r_1}, \ldots, q^\ell_{l_t,r_t}$ (it is possible to do without communication since paths $q^\ell_{i,j}$ are paths in the lower triangle).

It is clear, that all the clauses corresponded to the vertices outside from the main diagonal are satisfied since when we invert values of the edges along the path we invert even number of incident edges. Note that by construction if $E^\ell_i \subseteq E(G)$ is a set of all edges incident to a vertex $v_i$ in the lower triangle, then $\sum_{e \in E^\ell_i} x(e) = A_i$ and if $E^u_i \subseteq E(G)$ is a set of all edges incident to a vertex $v_i$ in the upper triangle, then $\sum_{e \in E^u_i} y(e) = B_i + c(v_i)$. Hence, a clause corresponds to a vertex $v_i$ is falsified if $A_i \neq B_i$, and moreover if $A_i \neq B_i$, then there is a clause corresponds to a vertex $v_i$ that is falsified. Hence we get a reduction $\mathrm{KW}(\oplus_n)$ to $\mathrm{Search}_{\mathtt{TS}_{G,c}}$.

In this reduction we use that Bob knows values of all edges along the path between the vertex labeled by 1 and $v_i$ for each $i \in [n]$. However if we do not fix the partition we can not guarantee this property. In order to solve this problem we prove that for all labeling functions $c$ for the fixed graph $G$ complexities of Tseitin formulas $\mathtt{TS}_{G,c}$ are the same.

**Lemma 5.2.** *If $G$ is a connected graph, $c$ and $c'$ are labeling functions such that $\mathtt{TS}_{G,c}$ and $\mathtt{TS}_{G,c'}$ are unsatisfiable, then for any constant $b > 0$ and partition of the variables $\Pi$ holds $\mathbf{D}^{(b),\Pi}\left(\mathrm{Search}_{\mathtt{TS}_{G,c}}\right) = \mathbf{D}^{(b),\Pi}\left(\mathrm{Search}_{\mathtt{TS}_{G,c'}}\right)$.*

*Proof.* Let Alice and Bob get some instance $I : \{p_e \ : \ e \in E(G)\} \to \{0,1\}$ of $\mathrm{Search}_{\mathtt{TS}_{G,c}}$ we show that they can construct without communication an instance $I'$ of $\mathrm{Search}_{\mathtt{TS}_{G,c'}}$ such that if they know a clause of $\mathtt{TS}_{G,c'}$ falsified by $I'$, then they can find a clause of $\mathtt{TS}_{G,c}$ falsified by $I$. As a result, we get the following inequality $\mathbf{D}^{(b),\Pi}\left(\mathrm{Search}_{\mathtt{TS}_{G,c}}\right) \leq \mathbf{D}^{(b),\Pi}\left(\mathrm{Search}_{\mathtt{TS}_{G,c'}}\right)$ and using symmetry of the statement we prove the theorem.

Let $\mathbb{1}_c = \{v \in V(G) \ : \ c(v) = 1\}$ and $\mathbb{1}_{c'} = \{v \in V(G) \ : \ c'(v) = 1\}$. Note that, $\mathbb{1}_c$ and $\mathbb{1}_{c'}$ have odd number elements since $\mathtt{TS}_{G,c}$ and $\mathtt{TS}_{G,c'}$ are unsatisfiable. Hence $\mathbb{1}_c \Delta \mathbb{1}_{c'}$[3] has even number of elements. We split $\mathbb{1}_c \Delta \mathbb{1}_{c'}$ into pairs $(x_1, y_1), \ldots, (x_t, y_t)$ and consider paths $q_1, \ldots, q_t$ such that $q_i$ is a path between $x_i$ and $y_i$.

Note that, if we invert in $I$ all the edges along the paths $q_1, \ldots, q_t$ we get an instance $I'$ of $\mathtt{TS}_{G,c'}$ such that $I$ falsify a clause $C$ corresponds to a vertex $v \in V(G)$ iff $I'$ falsify a clause $C'$ corresponds to a vertex $v \in V(G)$ and moreover we can get $C'$ if we invert in $C$ all the variables correspond to edges along the paths $q_1, \ldots, q_t$. □

In Theorem 5.1 we will prove lower bounds for Tseitin formulas on expander graphs

**Definition 5.1.** *A graph $G$ with vertices $V$ and edges $E$ is an $(n, d, \alpha)$-algebraic expander, if $|V| = n$, the degree of any vertex in $V$ equals $d$ and the absolute value of the second largest eigenvalue of the adjacency matrix of $G$ is not greater than $\alpha d$.*

It is well known that for all $\alpha > 0$ and all large enough constants $d$ there exists a family $\{G_n\}_{n \in \mathbb{N}}$ of $(n, d, \alpha)$-algebraic expanders. There are explicit constructions such that $G_n$ can be constructed in $\mathrm{poly}(n)$ time [26]. Also, it is known that a random $d$-regular graph is a good expander with high probability.

[3] From here on after $A \Delta B$ denotes a symmetric difference of $A$ and $B$.

In our sketch we use two important properties: the upper and the lower triangles are connected and have a large intersection. Fortunately, expander graphs have similar property: if $M$ is an arbitrary half of edges of $G$, then $G[M]$ and $G[E(G) \setminus M]$ have connected components $V_1$ and $V_2$, respectively, such that $V_1$ and $V_2$ have large intersection.

**Lemma 5.3** ([2]). *(Expander mixing lemma) Let $G$ be an $(n, d, \alpha)$-expander. For any two subsets $S, T \subseteq V(G)$ the following inequality holds: $||E_{S,T}(G)| - \frac{d|S||T|}{n}| \leq \alpha d\sqrt{|S||T|}$, where $|E_{S,T}(G)| = |\{(u, v) : u \in S, v \in T, (u, v) \in E(G)\}|$*[4].

**Corollary 5.1.** *For any $(n, d, \alpha)$-expander $G$ for $\alpha < \frac{1}{16}$ and $M \subseteq E(G)$ such that $\left\lfloor \frac{dn}{4} \right\rfloor \leq |M| \leq \left\lceil \frac{dn}{4} \right\rceil$ there are two subsets of vertices $V_1, V_2 \subseteq V(G)$ such that $|V_1 \cap V_2| \geq \frac{n}{16^2}$, $V_1$ is a connected component in $G[M]$, and $V_2$ is a connected component in $G[E(G) \setminus M]$.*

*Proof.* First of all, let us prove that for any $M \subseteq E(G)$ of size at least $\left\lceil \frac{dn}{4} \right\rceil$ there is a large connected component in $G[M]$ i.e. there is a subset of vertices $V' \subseteq V(G)$ such that $|V'| \geq \frac{n}{8}$ and $V'$ is a connected component of $G[M]$.

Let us assume that for some $M$ all connected components $S_1, \ldots, S_l$ in $G[M]$ have size less than $\frac{n}{8}$. Note that $\sum_{i=1}^{l} |E_{S_i,S_i}(G)| \geq 2|M|$ and by the expander mixing lemma $|E_{S_i,S_i}(G)| \leq \frac{d|S_i|^2}{n} + \alpha d|S_i|$. Since we assume that $|S_i| < \frac{n}{8}$ we have the following inequality

$$|E_{S_i,S_i}(G)| < \frac{d|S_i|}{8} + \alpha d|S_i| < \frac{3d|S_i|}{16}.$$

As a result, we get a contradiction:

$$2\left\lfloor \frac{dn}{4} \right\rfloor \leq 2|M| \leq \sum_{i=1}^{l} |E_{S_i,S_i}(G)| < \sum_{i=1}^{l} \frac{3d|S_i|}{16} = \frac{3dn}{16}.$$

Now let $V_1$ be any connected component of $G[M]$ of size at least $\frac{n}{8}$ and $V_2$ be any connected component of $G[E \setminus M]$ of size at least $\frac{n}{8}$.

Note that for any vertices $v_1 \in V_1$ and $v_2 \in V_2$ if $(v_1, v_2) \in M$, then $v_2 \in V_1$ since $V_1$ is a connected component in $G[M]$ and if $(v_1, v_2) \in E(G) \setminus M$, then $v_1 \in V_2$ since $V_2$ is a connected component in $G[E(G) \setminus M]$. Hence for any $v_1 \in V_1$ and $v_2 \in V_2$, if $(v_1, v_2) \in E(G)$, then $v_1 \in V_1 \cap V_2$ or $v_2 \in V_1 \cap V_2$. As a result, $\frac{E_{V_1,V_2}(G)}{2d} \leq |V_1 \cap V_2|$. However, by mixing lemma $|E_{V_1,V_2}(G)| \geq \frac{dn^2}{8^2 n} - \alpha d\frac{n}{8} \geq \frac{dn}{16 \cdot 8}$. □

*Proof of Theorem 5.3.* Let us consider a family $\{G_n\}_{n\in\mathbb{N}}$ of $(n, d, \alpha)$-expanders and some family of labeling functions $\{c_n : V(G_n) \to \{0, 1\}\}_{n\in\mathbb{N}}$. At the beginning we prove that for some constant $\epsilon > 0$ the following inequality $\mathbf{D}^{(b)}\left(\mathrm{Search}_{\mathtt{TS}_{G_n,c_n}}\right) \geq \mathbf{D}^{(b)}\left(\mathrm{KW}\left(\oplus_{\epsilon n}\right)\right)$ holds and by Theorem 5.2 it implies, $\mathbf{D}^{(b)}\left(\mathrm{Search}_{\mathtt{TS}_{G_n,c_n}}\right) \geq \Omega\left(n^{1/(b-1)}\right)$.

Let us fix $n \in \mathbb{N}$ and some balanced partition $\Pi$ of the variables of $\mathrm{Search}_{\mathtt{TS}_{G_n,c_n}}$. Note that there is a set of edges $M$ corresponding to $\Pi_0$. By the previous Corollary there are $V_1, V_2 \subseteq V(G_n)$ such that $V_1$ is a connected component in $G[M]$, $V_2$ is a connected component in $G[E(G_n) \setminus M]$, and $|V_1 \cap V_2| \geq \frac{n}{128}$. Let $V_1 \cap V_2 = \{v_1, \ldots, v_k\}$ $(k \geq \frac{n}{256})$ and $c'$ be a labeling function such that the only vertex labeled by 1 belongs to $V_1 \cap V_2$.

By Lemma 5.2, $\mathbf{D}^{(b),\Pi}\left(\mathrm{Search}_{\mathtt{TS}_{G_n,c_n}}\right) = \mathbf{D}^{(b),\Pi}\left(\mathrm{Search}_{\mathtt{TS}_{G_n,c'}}\right)$. Hence, it is enough to prove that $\mathbf{D}^{(b)}\left(\mathrm{KW}\left(\oplus_k\right)\right) \leq \mathbf{D}^{(b)}\left(\mathrm{Search}_{\mathtt{TS}_{G_n,c'}}\right)$.

We prove this inequality using reduction $\mathrm{KW}\left(\oplus_k\right)$ to $\mathrm{Search}_{\mathtt{TS}_{G_n,c'}}$. Let for each $i, j \in [k]$, $q^1_{i,j}$ be a path from $v_i$ to $v_j$ in $G[M]$ and $q^2_{i,j}$ be a path from $v_i$ to $v_j$ in $G[E(G) \setminus M]$. Let Alice gets a vector $a \in \{0, 1\}^k$ such that $\oplus_k(a) = 0$ and Bob gets $b \in \{0, 1\}^k$ such that $\oplus_k(b) = 1$. We construct an instance $(x, y)$ such that Alice and Bob can obtain $\mathrm{KW}\left(\oplus_k\right)(a, b)$ from $\mathrm{Search}_{\mathtt{TS}_{G_n,c'}}$ without communication. Construction consists of three steps and each of them does not involve communication between Alice and Bob.

[4]Note that if $u, v \in S \cap T$, then we count an edge between them twice.

- Initially, Alice and Bob have all zero instance $(x, y)$ of $\text{Search}_{\text{TS}_{G_n,c'}}$;
- Note that $\sum_{i=1}^{k} a_i \equiv 0 \pmod 2$, since $\oplus_k(a) = 0$. Hence, the set $\{i \in [k] \ : \ a_i = 1\}$ has even size and Alice can split it into pairs $(l_1, r_1), \ldots, (l_t, r_t)$. After that Alice inverts all the edges along the paths $q^1_{l_1,r_1}, \ldots, q^1_{l_t,r_t}$.
- Similarly $\sum_{i=1}^{k} b_i \equiv 0 \pmod 2$, since $\oplus_k(b) = 1$. Hence, the set $\{i \in [k] \ : \ b_i = 1\} \Delta \{1\}$ has even size and Alice can split it into pairs $(l_1, r_1), \ldots, (l_t, r_t)$. After that Alice inverts all the edges along the paths $q^2_{l_1,r_1}, \ldots, q^2_{l_t,r_t}$.

It is clear, that all the clauses corresponded to the vertices outside $V_1 \cap V_2$ are satisfied since when we invert values of the edges along the path we invert even number of incident edges. Note that by construction if $E^1_i \subseteq M$ is a set of all edges incident to a vertex $v_i$, then $\sum_{e \in E^1_i} x(e) = a_i$ and if $E^2_i \subseteq E(G) \setminus M$ is a set of all edges incident to a vertex $v_i$, then $\sum_{e \in E^2_i} y(e) = b_i + c'(v_i)$. Hence, a clause corresponds to a vertex $v_i$ is falsified if $a_i \neq b_i$, and moreover if $a_i \neq b_i$, then there is a clause corresponds to a vertex $v_i$ that is falsified. Hence we get a reduction $\text{KW}\,(\oplus_k)$ to $\text{Search}_{\text{TS}_{G_n,c'}}$. □

*Proof of the Theorem 5.1.* Let $\{G_n\}_{n\in\mathbb{N}}$ and $\{c_n : V(G_n) \to \{0,1\}\}_{n\in\mathbb{N}}$ be a family of graphs and a family of labeling functions from Theorem 5.3.

Let us assume that there is a $b$-OBDD diagram for $\text{Search}_{\text{TS}_{G_n,c_n}}$ of size $S$. Lemma 5.1 implies that $\mathbf{D}^{(2b-1),best}\left(\text{Search}_{\text{TS}_{G_n,c_n}}\right) \leq (2b-1) \cdot S$. Hence, by Theorem 5.3 $S = \Omega(n^{1/(2b-1)})$. As a result, using Theorem 3.1 we conclude that any $b$-OBDD-$\mathbf{PS}_1$ proof of $\text{TS}_{G_n,c_n}$ has size $2^{\Omega(n^{1/(2b-1)})}$. □

It is possible to see, that Theorem 5.3 gives almost tight lower bound for communication complexity of Tseitin formulas and we can transform protocol from Proposition 5.2 to $b$-round setting with cost $O(n^{1/b})$.

**Proposition 5.2.** *Let $b > 0$ be a constant, $G$ be a graph with a constant degree, $c : V(G) \to \{0,1\}$ be a labeling function, and $\Pi$ be a partition of the variables of $\text{TS}_{G,c}$. If $\text{TS}_{G,c}$ is unsatisfiable, then $\mathbf{D}^{(b),\Pi}\left(\text{Search}_{\text{TS}_{G,c}}\right) = O(|V(G)|^{1/b})$.*

*Proof.* Let us consider the protocol with $b$ stages such that on the stage with number $i$ Alice and Bob consider a subset $V_i \subseteq V(G)$ such that $\sum_{(u,v)\in E_G(V_i,V(G))} p_{(u,v} \neq \sum_{v\in V_i} c(v)$ and $|V_b| = 1$.

Initially, $V_1 = V(G)$ and if $|V_i| \neq 1$ Alice and Bob split $V_i$ into $l = \left\lceil |V(G)|^{1/b} \right\rceil$ parts $V^1_i, \ldots, V^l_i$ such that $-1 \leq |V^j_i| - |V^k_i| \leq 1$ for all $j, k \in [l]$ and if

$$\sum_{(u,v)\in E_G(V^j_i,V(G))} p_{(u,v)} \neq \sum_{v\in V^j_i} c_v$$

for some $j \in [l]$ set $V_{i+1} = V^j_i$.

Note that, if Alice and Bob know the variable $v$ such that $\sum_{u\in N_G(v)} p_{u,v} \neq c(v)$ ($N_G(v)$ denotes all neighbors of $v$ in the graph $G$), then they can find a falsified clause using constant communication.

Additionally, it is easy to see that regardless of the partition of the variables Alice and Bob can check the equality

$$\sum_{(u,v)\in E_G(V^j_i,V(G))} p_{(v,u)} = \sum_{v\in V^j_i} c_v$$

using two bits of communication. As a result, this is a protocol for $\text{Search}_{\text{TS}_{G,c}}$ with the cost equal to $O(\lceil |V(G)|^{1/b} \rceil)$ (they exchange with $2l = 2\left\lceil |V(G)|^{1/b} \right\rceil$ bits on each step). □

Moreover, there is a $b$-OBDD of size $n^2 2^{(n^{1/b})}$.

**Theorem 5.4.** *Let $b, d, n > 0$ be integers, $G$ be a graph on $n$ vertices with maximal degree $d$, and $c : V(G) \to \{0,1\}$ be a labeling function. If $\mathtt{TS}_{G,c}$ is unsatisfiable, then there is a $b$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G,c}$ of size at most $n^2 2^{(n^{1/b}+d)}$.*

*Proof.* Without loss of generality we may assume that the graph $G$ is connected.

First of all, let us note that for any set $\{f_i(x) = b_i\}_{i=1}^k$ of linear equation on $m$ variables there is an OBDD of size $m2^k$ that represents the function $g : \{0,1\}^m \to \{0,1\}^k$ such that for any $x \in \{0,1\}^m$, $g(x)_i = 1$ iff $f_i(x) = b_i$.

Now, let us construct a $b$-OBDD for $\mathrm{Search}_{\mathtt{TS}_{G,c}}$. Let us fix $k = \lceil n^{1/b} \rceil$ and construct the following tree (with vertices labeled by subsets of $V(G)$):

- the root of this tree is labeled by $V(G_n)$;
- for each node $p$ labeled by a set of size more than 1 we create $k$ children $q_1, \ldots, q_k$ of $p$, split the current label $V_p \subseteq V(G)$ into $k$ almost equal sized sets $V_{q_1}, \ldots, V_{q_k}$, and label $q_1, \ldots, q_k$ by $V_{q_1}, \ldots, V_{q_k}$;
- we terminate this process if current set has size at most 1;

Note that, depth of this tree is at most $b$.

It is easy to see, that for any input $x$ we can go from the root to a leaf of this tree such that on any step we stay in the node $p$ such that $\sum\limits_{(u,v) \in E_G(V_p, V(G))} x_{(u,v)} \neq \sum\limits_{v \in V_p} c(v)$. Indeed, in the root this statement is true because $\mathtt{TS}_{G,c}$ is unsatisfiable and if this statement is true for some node $p$, then there is a child $q$ in the tree such that this statement holds for $q$ too, because $\{V_{q_i}\}_{i=1}^k$ is a partition of $V_p$. Hence, when we reach a leaf, this leaf is labeled by a set $\{v\}$ such that $\sum\limits_{(v,u) \in E(G_n)} x_{(v,u)} \neq c(v)$. We can not reach a leaf labeled by an empty set since the sum over the empty set is always zero.

One can transform this tree into a $b$-OBDD for $\mathrm{Search}_{\mathtt{TS}_{G,c}}$ of size $k^b n(2^k + 2^d) = n^2 2^{(n^{1/b}+d)}$. At first, we construct an OBDD for each node $p$ with children $q_1, \ldots, q_k$ that find a node $q_i$ such that $\sum\limits_{(u,v) \in E_G(V_{q_i}, V(G))} x_{(u,v)} \neq \sum\limits_{v \in V_{q_i}} c(v)$ if $\sum\limits_{(u,v) \in E_G(V_p, V(G))} x_{(u,v)} \neq \sum\limits_{v \in V_p} c(v)$. By previous remark, this OBDDs have size at most $n2^k$. After that, we connect this OBDDs into $b$-OBDD similarly to the tree. It is easy to see that we get a $(b-1)$-OBDD that finds a vertex such that one of the corresponding to it clauses is falsified by $x$ (we have an OBDD for all vertices except leafs). The size of these $(b-1)$-OBDD equals $k^b(n2^k)$ since each vertex has at most $k$ children. Hence if we put an OBDD that finds a falsified clause corresponding to a vertex $v \in V(G)$ in each node labeled by the vertex $v$ we get $b$-OBDD for $\mathrm{Search}_{\mathtt{TS}_{G,c}}$ of size $k^b n(2^k + 2^d)$ since the last OBDD has size at most $2^d$. □

Using this result we can show that $b$-OBDD-$\mathbf{PS}_1$ does not $p$-simulate $(2b)$-OBDD-$\mathbf{PS}_1$

**Corollary 5.2.** *There are a family of graphs $\{G_n\}_{n \in \mathbb{N}}$ and labeling functions $\{c_n\}_{n \in \mathbb{N}}$ such that*

- *$G_n$ has $n$ vertices and a constant degree $d$;*
- *Tseitin formulas $\mathtt{TS}_{G_n,c_n}$ are unsatisfiable;*
- *for any $b > 0$, any $b$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G_n,c_n}$ has size at least $2^{\Omega\left(n^{1/(2b-1)}\right)}$;*
- *for any $b > 0$, there is a $(2b)$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G_n,c_n}$ of size at most $2^{\left(n^{1/(2b)\log n}\right)}$.*

Additionally, using Theorem 5.4 we can show that **Res** does not $p$-simulate $b$-OBDD-$\mathbf{PS}_1$ for $b \geq 2$.

**Corollary 5.3.** *There is a family of graphs $\{G_n\}_{n \in \mathbb{N}}$ and labeling functions $\{c_n\}_{n \in \mathbb{N}}$ such that*

- *$G_n$ has $n$ vertices and a constant degree $d$;*
- *Tseitin formulas $\mathtt{TS}_{G_n,c_n}$ are unsatisfiable;*
- *any resolution proof of $\mathtt{TS}_{G_n,c_n}$ has size at least $2^{\Omega(n)}$;*
- *for any $b$ there is a $b$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G_n,c_n}$ of size at most $2^{\left(n^{1/b\log n}\right)}$.*

Theorem 5.4 and Theorem 3.2 show that unsatisfiable Tseitin formulas have $\oplus$-OBDD-$\mathbf{PS}_1$ proofs of sub-exponential size. Moreover, this upper bound for $\oplus$-OBDD-$\mathbf{PS}_1$ may be enhanced and we can prove that Tseitin formulas have $\oplus$-OBDD-$\mathbf{PS}_1$ proofs of polynomial size.

**Theorem 5.5.** *For any graph $G$ on $n$ vertices with maximal degree $d$ and labeling function $c : V(G) \to \{0,1\}$ if $\mathtt{TS}_{G,c}$ is unsatisfiable, then there is a $\oplus$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G,c}$ of size $\mathrm{poly}(n, 2^d)$.*

*Proof.* Without loss of generality we may assume that $G$ is a connected graph.

In the sequel we construct a $\oplus$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G,c}$ for any order of variables. Let us consider the case when $G$ has an odd number of vertices. Note that, since $\mathtt{TS}_{G,c}$ is unsatisfiable, then the cardinality of the set $\left\{ v \in V(G) \ : \sum\limits_{(u,v)\in E(G)} p_{(u,v)} \neq c(v) \right\}$ is odd for any substitution to the variables $p_e$ $(e \in E(G))$. Let us define for each $v \in V(G)$, $S_v \in \{0,1\}$ such that $S_v = 0$ iff $\sum\limits_{(u,v)\in E(G)} p_{(u,v)} \neq c(v)$. It is easy to see that $\bigoplus\limits_{v\in V(G)} S_v = 0$ and $\bigoplus\limits_{v\in V(G)} 1 = 1$ since $V(G)$ has odd number of vertices.

Let $\mathtt{TS}_{G,c} = \bigwedge\limits_{i=1}^{m} C_i$ and $F_v : \{0,1\}^{|E(G)|} \times \{0,1\}^m \to \{0,1\}$ be a function with two types of variables: variables corresponding to edges and variables corresponding to clauses of $\mathtt{TS}_{G,c}$ (placeholder variables), such that $F_v(\bar{p},\bar{y}) = 0$ iff $\sum\limits_{u\in V(G):(u,v)\in E(G)} p_{(u,v)} \neq c(v)$ and a placeholder variable that corresponds to a falsified clause of the subformula $\sum\limits_{u\in V(G):(u,v)\in E(G)} p_{(u,v)} = c(v)$ (note that any substitution to this formula may falsify at most one clause) of the formula $\mathtt{TS}_{G,c}$ is equal to zero. $F_v$ has an OBDD representation of size at most $2^{d+2}$ (since there are at most $d$ variables $p_{(u,v)}$ in this formula); hence $\oplus$-OBDD representation of $\bigoplus\limits_{v\in V(G)} F_v$ has size $n2^{d+2}$. Moreover, since $\bigoplus\limits_{v\in V(G)} F_v(\bar{x}, C_1(\bar{x}), \ldots, C_m(\bar{x})) = \bigoplus\limits_{v\in V(G)} S_v = 0$ for any $x \in \{0,1\}^{|E(G)|}$ and $\bigoplus\limits_{v\in V(G)} F_v(\bar{x}, 1, \ldots, 1) = \bigoplus\limits_{v\in V(G)} 1 = 1$ for any $x \in \{0,1\}^{|E(G)|}$, $\oplus$-OBDD representation of $\bigoplus\limits_{v\in V(G)} F_v$ is a $\oplus$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G,c}$.

For the case when $G$ has even number of vertices, the proof is almost the same. Let $v$ be a vertex such that $G' = G \setminus v$ is connected graph. Note that for any graph $G$ with labeling function $c : V(G) \to \{0,1\}$ we may consider graph $G'$ with labeling function $c^{(f)} : V(G') \to \{0,1\}$ such that $c^{(f)}(u) = c(u)$ for $u$ not incident with $v$ and $c^{(f)}(u) = c(u) \oplus f(u)$ for $u \in N_G(v)$ where $f \in R_v = \left\{ g : N_G(v) \to \{0,1\} \ : \sum\limits_{u\in N_G(v)} g(u) = c(v) \right\}$. It is easy to see that if $\mathtt{TS}_{G,c}$ is unsatisfiable, then $\mathtt{TS}_{G',c^{(f)}}$ is unsatisfiable for any $f \in R_v$. Since graph $G'$ has odd number of vertices there is a $\oplus$-OBDD proof $D^{(f)}$ of $\mathtt{TS}_{G',c^{(f)}}$ of size $\mathrm{poly}(n)$. Now, let us consider the following diagram: it splits by the variables $p_{(u,v)}$ for all $u \in N_G(v)$ and if these variables falsifies a clause $C_i$ of the subformula $\sum\limits_{u\in V(G):(u,v)\in E(G)} p_{(u,v)} = c(v)$ of formula $\mathtt{TS}_{G,c}$ we return the value of $y_i$, otherwise we consider a function $f \in R_v$ such that $f(u) = p_{(u,v)}$ for all $u \in N_G(v)$ and return the value of $D^{(f)}$. Obviously it is a $\oplus$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G,c}$ of size $\mathrm{poly}(n, 2^d)$. □

As a result, using this theorem we can show that $b$-OBDD-$\mathbf{PS}_1$ does not $p$-simulate $\oplus$-OBDD-$\mathbf{PS}_1$.

**Corollary 5.4.** *There are a family of graphs $\{G_n\}_{n\in\mathbb{N}}$ and labeling functions $\{c_n\}_{n\in\mathbb{N}}$ such that*

- *$G_n$ has $n$ vertices and a constant degree $d$;*
- *Tseitin formulas $\mathtt{TS}_{G_n,c_n}$ are unsatisfiable;*
- *for any $b > 0$, any $b$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G_n,c_n}$ has size at least $2^{\Omega\left(n^{1/b}\right)}$;*
- *there is a $\oplus$-OBDD-$\mathbf{PS}_1$ proof of $\mathtt{TS}_{G_n,c_n}$ of size $\mathrm{poly}(n)$.*

The idea behind the proof of the upper bound for $b$-OBDD-$\mathbf{PS}_1$ on $\mathtt{TS}_{G,c}$ may be generalized to the following case.

In standard decision trees, at each node a test on a single variable is made. If the result is 0, one descends into the left subtree, whereas if the result is 1, one descends into the right subtree. The label of the reached leaf is the value of the function (on that particular input). We will now consider decision trees where more general test are allowed. In an OBDD decision tree, at each node one evaluate a value of some $\pi$-OBDD on input. If the result is 0 one descends into the left subtree, whereas if the result is 1, one descends into the right subtree. The cost of such a tree is a maximum among sizes of OBDDs in this tree (note that an order $\pi$ is the same for all nodes of the tree).

**Theorem 5.6.** *If there is an* OBDD*-decision tree for a relation* $\mathrm{Search}_\phi$ *of cost $S$ and depth $d$, then there is a $b$-*OBDD *for* $\mathrm{Search}_\phi$ *of size $S^{2^{\left\lceil \frac{d}{b} \right\rceil}} \cdot 2^d$.*

We prove this theorem in the end of the paragraph. Now let us prove some corollaries of this theorem.

It is easy to see, that the communication protocol for $\mathrm{Search}_{\mathtt{TS}_{G,c}}$ from Proposition 5.2 may be transformed into an OBDD-tree of size $O(n)$ and depth $\log n$ (since on each step of the protocol we compute a parity of some variables). Hence, using Theorem 5.6 we can almost get a result of Theorem 5.4, we can prove that for any graph $G$ on $n$ vertices with maximal degree $d$ and labeling function $c : V(G) \to \{0,1\}$ there is a $b$-OBDD for $\mathrm{Search}_{\mathtt{TS}_{G,c}}$ of size $n^{2^{\log n/b}} 2^{\log n} = 2^{O(n^{1/b} \log n)}$.

It is possible to note that, using standard transformation between proofs and communication protocols, any OBDD($\wedge$, weakening) proof [3,20,22] of a formula $\phi$ of size $S$ can be transformed into an OBDD-decision tree of cost $S$ and depth $O(\log S)$.

Using this argument we want to extend Corollary 4.1 for $k = 1$ and prove that semantic **CP** does not simulate $b$-OBDD-$\mathbf{PS}_1$ for large enough $b$. For this separation we need a formula that is easy for OBDD($\wedge$, weakening) but it is hard for semantic **CP**.

An example of such a formula is the Clique-Coloring principle. The Clique-Coloring principle is a formula encoding the statement that it is impossible that graph is $(m-1)$-colorable and has a $m$-clique in the same time. The Clique-Coloring principle is a formula on variables $\{p_{i,j}\}_{i \neq j \in [n]}$, $\{r_{i,l}\}_{i \in [n], l \in [m-1]}$, and $\{q_{k,i}\}_{k \in [m], i \in [n]}$. Informally $p_{i,j} = 1$ iff there is an edge between vertices $i$ and $j$, $r_{i,l} = 1$ iff vertex $i$ has color $l$, and $q_{k,i} = 1$ iff vertex $i$ is $k$th vertex in the clique.

More formally, the Clique-Coloring principle is a conjunction of the following statements written in CNF.

- $\bigvee_{i=1}^{n} q_{k,i}$ for any $k \in [m]$, this constraint states that there is a vertex in the clique with number $k$;
- $\neg q_{k,i} \vee \neg q_{k',j} \vee p_{i,j}$ for all $i, j \in [n]$ and $k \neq k' \in [m]$, this constraint states that if both $i$ and $j$ are in the clique then there is an edge between them;
- $\bigvee_{l=1}^{m-1} r_{i,l}$ for all $i$, this constraint states that any vertex is colored;
- $\neg p_{i,j} \vee \neg r_{i,l} \vee \neg r_{j,l}$ for all $i \neq j$ and $l$, this constraint states that for any two vertices with the same color it is impossible that there is an edge between them.

We denote by $\text{Clique-Coloring}_{n,m}$ the Clique-Coloring principle for $n$ and $m$.

**Theorem 5.7** ([12]). *There is a constant $c > 0$ such that any semantic* **CP** *proof of* Clique-Coloring$_{n,\sqrt{n}}$ *has size at least* $2^{n^c}$.

However, recently was proven that there is an OBDD($\wedge$, weakening) proof of Clique-Coloring$_{n,\sqrt{n}}$ of size poly($n$) [10].

**Theorem 5.8** (see [10]). *For any $n$ there is an* OBDD*-decision tree with cost $O(n)$ and depth $O(\log n + \log m)$ for* Search$_{\text{Clique-Coloring}_{n,m}}$.

In fact, direct construction of an OBDD-decision tree for Search$_{\text{Clique-Coloring}_{n,m}}$ of cost $O(n)$ and depth $O(\log n + \log m)$ is much easier than construction of OBDD($\wedge$, weakening) proof.

*Proof.* Let us consider the following algorithm that solves Search$_{\text{Clique-Coloring}_{n,m}}$ that can be easily transformd into an OBDD-tree.

1. If $\sum\limits_{i\in[n],k\in[m],l\in[m-1]} q_{k,i}r_{i,l} \geq m$, then go to the step 2, otherwise go to the step 5.
2. On this step we know that $\sum\limits_{l=1}^{m-1}\sum\limits_{i\in[n],k\in[m]} q_{k,i}r_{i,l} \geq m$. Hence there is $l \in [m-1]$ such that $\sum\limits_{i\in[n],k\in[m]} q_{k,i}r_{i,l} > 1$ and we can find this $l$ using the binary search. After we find $l$, go to the step 3.
3. On this step we know that $\sum\limits_{i\in[n],k\in[m]} q_{k,i}r_{i,l} > 1$; hence there are $k_1 \in [m]$, $i_1 \in [n]$, $k_2 \in [m]$, and $i_2 \in [n]$ such that $q_{k_1,i_1} = q_{k_2,i_2} = r_{i_1,l} = r_{i_2,l} = 1$ and $(k_1, i_1) \neq (k_2, i_2)$; We find them using the binary search and go to the step 4.
4. We know that $q_{k_1,i_1} = q_{k_2,i_2} = r_{i_1,l} = r_{i_2,l} = 1$; If $p_{i_1,i_2} = 1$, then return the clause $\neg p_{i,j} \vee \neg r_{i,l} \vee \neg r_{j,l}$, otherwise return the clause $\neg q_{k,i} \vee \neg q_{k',j} \vee p_{i,j}$.
5. We know that $\sum\limits_{k=1}^{m}\sum\limits_{i\in[n],l\in[m-1]} q_{k,i}r_{i,l} < m$. Hence there is $k \in [m]$ such that $\sum\limits_{i\in[n],l\in[m-1]} q_{k,i}r_{i,l} = 0$ and we can find it using the binary search. After we find $k$, go to the step 6.
6. We know that $\sum\limits_{i\in[n],l\in[m-1]} q_{k,i}r_{i,l} = 0$; hence if for some $i \in [n]$, $q_{k,i} = 1$ holds, then $r_{i,l} = 0$ for all $l \in [m-1]$. Using binary search find $i$ such that $q_{k,i} = 1$ if there is not such $i$ return the clause $\bigvee\limits_{i=1}^{n} q_{k,i}$, otherwise return the clause $\bigvee\limits_{l=1}^{m-1} r_{i,l}$.

It is easy to see, that on each step of this algorithm we split by the value of an OBDD of size $O(n)$. Also, number of steps of this algorithm on any input is bounded by $O(\log n)$. Hence, the resulting tree has depth $O(\log n)$ and cost $O(n)$. □

**Corollary 5.5.** *There are a family of formulas $\{\phi\}_{n\in\mathbb{N}}$ and constants $c > 0$ and $d > 0$ such that*

- $|\phi_n| = \text{poly}(n)$ *and $\phi_n$ depends on* poly($n$) *variables;*
- *any semantic* **CP** *proof of $\phi_n$ has size at least $2^{\Omega(n^c)}$;*
- *for any $b$ there is a $b$-*OBDD-$\mathbf{PS}_1$ *proof of $\phi_n$ of size at most $2^{O\left(n^{d/b\log n}\right)}$.*

*Proof.* Let us consider a formula $\phi_n$ = Clique-Coloring$_{n,\sqrt{n}}$. By Theorem 5.7 any semantic **CP** proof of $\phi_n$ has size at least $2^{n^c}$ for some constant $c > 0$. However by Theorem 5.8 there is an OBDD-decision tree of cost $O(n)$ and depth $d\log n$ for some constant $d > 0$; hence using Theorem 5.6 we can get an $b$-OBDD-$\mathbf{PS}_1$ proof of Clique-Coloring$_{n,\sqrt{n}}$ of size $n^{2^{d\log n/b}}2^{d\log n} = 2^{O\left(n^{d/b}\log n\right)}$. □

Now let us prove Theorem 5.6. First of all, we need the following technical Lemmas.

**Lemma 5.4** ([36, Theorem 3.3.6]). *Let $f : \{0,1\}^n \to \Lambda_f$, $g : \{0,1\}^n \to \Lambda_g$, and $p : \Lambda_f \times \Lambda_g \to \Lambda$. If there are* OBDD*s $F$ and $G$ that represent functions $f$ and $g$ respectively, then there is an* OBDD *that represents $p(f(x), g(x))$ of size $|F| \cdot |G|$.*

**Corollary 5.6.** *For any functions $f, g : \{0,1\}^n \to \Lambda$ and function $h : \{0,1\}^n \to \Lambda \times \Lambda$ such that $h(x) = (f(x), g(x))$ holds for any $x \in \{0,1\}^n$ if $F$ and $G$ are* OBDD*s that represents $f$ and $g$, respectively, then there is an* OBDD *$H$ that represents $h$ of size at most $|F| \cdot |G|$.*

*Proof.* We just use Lemma 5.4 with the Identity function $p : \Lambda \times \Lambda \to \Lambda \times \Lambda$. □

**Corollary 5.7.** *For any functions $f : \{0,1\}^n \to \Lambda \times \Lambda$, $g : \{0,1\}^n \to \{0,1\}$ and $h : \{0,1\}^n \to$ such that for any $x \in \{0,1\}^n$ $h(x) = f(x)_{g(x)}$ holds if $F$ and $G$ are* OBDD*s that represents $f$ and $g$, respectively, then there is an* OBDD *$H$ that represents $h$ of size at most $|F| \cdot |G|$.*

*Proof.* We just use Lemma 5.4 with the Identity function $p : \Lambda \times \Lambda \to \Lambda \times \Lambda$. □

**Lemma 5.5.** *If there is an* OBDD*-decision tree for a function $f$ of cost $S$ and depth $d$, then there is an* OBDD *for $f$ of size at most $S^{2^d-1}$.*

*Proof.* We prove this lemma using mathematical induction by $d$. The basis is clear, now let us prove the induction step from $d$ to $d+1$.

Let us consider a tree $T$ of depth $d+1$ and cost $S$, let $r$ be its root labeled by $D$, by induction hypothesis we can transform left and right subtrees of $r$ into OBDDs $L$ and $R$, respectively, of size at most $S^{2^d-1}$. By Corollary 5.6 and Corollary 5.7 we get an OBDD for $f$ of size $|D| \cdot |L| \cdot |R| \leq S \cdot S^{2^d-1} \cdot S^{2^d-1} = S^{2^{d+1}-1}$. □

*Proof of Theorem 5.6.* Let $T$ be an OBDD-decision tree for a relation $\text{Search}_\phi$ of cost $S$ and depth $d$. Let us split $T$ (starting from the root) into trees of depth $\lceil \frac{d}{b} \rceil$ and organize them in a natural way as a tree of arity at most $2^{\lceil \frac{d}{b} \rceil}$ and depth $b$.

By Lemma 5.5 we can transform each of these trees into an OBDD of size $S^{2^{\lceil \frac{d}{b} \rceil}}$. Hence we can get a $b$-OBDD that computes by an input the number of a leaf of $T$ corresponding to this input and size of this $b$-OBDD is at most $S^{2^{\lceil \frac{d}{b} \rceil}} 2^d$.

As a result we get a $b$-OBDD for $\text{Search}_\phi$ of size at most $S^{2^{\lceil \frac{d}{b} \rceil}} \cdot 2^d$. □

# 6 Ordering Priniple and its Variations

In the previous paragraph we have proven hierarchy for $b$-OBDD-$\mathbf{PS}_1$, however, for $(1,+b)$-BP-$\mathbf{PS}_1$ hierarchy does not known. In this paragraph we show that at least first two layers are different i.e. 1-BP-$\mathbf{PS}_1$ does not $p$-simulate $(1,+1)$-BP-$\mathbf{PS}_1$. We also enhence a sepration between **Res** and OBDD-$\mathbf{PS}_1$ in this section.

**Theorem 6.1.** *There is a family of unsatisfiable CNFs $\{\phi_n\}_{n\in\mathbb{N}}$ such that*

- $|\phi_n| = \text{poly}(n)$*;*
- *any* 1-BP-$\mathbf{PS}_1$ *proof of $\phi_n$ has size at least $2^{\Omega(n)}$;*
- *there is a* $(1,+1)$-BP-$\mathbf{PS}_1$ *proof of $\phi_n$ of size* $\text{poly}(n)$.

For proving this theorem we use a lower bound proven by Alekhnovich et. al. [1] for the following formula. Let $\rho : [n]^3 \to [n]^2$; the formula $\texttt{GT}'_{n,\rho}$ is the conjunction of the following statements written in CNF:

**antisymmetry:** $x_{i,j} \leftrightarrow \neg x_{j,i}$ for all $i \neq j \in [n]$;

**totality:** $\bigvee_{k\in[n],k\neq j} x_{k,j}$ for all $j \in [n]$;

**positively corrupted transitivity:** $\neg x_{i_1,i_2} \vee \neg x_{i_2,i_3} \vee \neg x_{i_3,i_1} \vee x_{\rho(i_1,i_2,i_3)}$ for all distinct $i_1, i_2, i_3 \in [n]$;

**negatively corrupted transitivity:** $\neg x_{i_1,i_2} \vee \neg x_{i_2,i_3} \vee \neg x_{i_3,i_1} \vee \neg x_{\rho(i_1,i_2,i_3)}$ for all distinct $i_1, i_2, i_3 \in [n]$.

Note that the formula $\mathtt{GT}'_{n,\rho}$ has only variables $x_{1,1}, \ldots, x_{n,n}$.

**Theorem 6.2** ([1]). *For $n$ sufficiently large, there exists $\rho$ such that any regular resolution refutation of $\mathtt{GT}'_{n,\rho}$ has size greater than $2^{n/200}$.*

For proving an upper bound for $\mathtt{GT}'_{n,\rho}$ let us consider an easier formula. The formula $\mathtt{GT}_n$ is the conjunction of the following statements written in CNF:

**antisymmetry:** $x_{i,j} \leftrightarrow \neg x_{j,i}$ for all $i \neq j \in [n]$;

**totality:** $\bigvee_{k\in[n],k\neq j} x_{k,j}$ for all $j \in [n]$;

**transitivity:** $\neg x_{i_1,i_2} \vee \neg x_{i_2,i_3} \vee \neg x_{i_3,i_1}$ for all distinct $i_1, i_2, i_3 \in [n]$.

**Theorem 6.3** ([7]). *For any $n$ there is an* **OrdRes** *proof of $\mathtt{GT}_n$ of size $O(n^3)$.*

Using this upper bound we can prove the upper bound for $\mathtt{GT}'_{n,\rho}$.

*Proof of Theorem 6.1.* To prove Theorem 6.1 it is enough to prove that for any $n$ and $\rho$ there is an $(1,+1)$-BP-$\mathbf{PS}_1$ proof of $\mathtt{GT}'_{n,\rho}$ of size $O(n^3)$.

By Theorem 3.1 it is enough to prove an upper bound for $(1,+1)$-BP for $\mathrm{Search}_{\mathtt{GT}'_{n,\rho}}$.

By Theorem 6.3 and Theorem 3.1 there is an OBDD $D$ for $\mathrm{Search}_{\mathtt{GT}_n}$ of size $O(n^3)$. Let $D'$ be the following modification of the diagram $D$, we replace each leaf of $D$ labeled by a clause $\neg x_{i_1,i_2} \vee \neg x_{i_2,i_3} \vee \neg x_{i_3,i_1}$ by a diagram that splits by a the variable $x_{\rho(i_1,i_2,i_3)}$ and if it is true, then returns $\neg x_{i_1,i_2} \vee \neg x_{i_2,i_3} \vee \neg x_{i_3,i_1} \vee \neg x_{\rho(i_1,i_2,i_3)}$, otherwise $\neg x_{i_1,i_2} \vee \neg x_{i_2,i_3} \vee \neg x_{i_3,i_1} \vee x_{\rho(i_1,i_2,i_3)}$.

Note that the size of the diagram $D'$ equal to $3|D| = O(n^3)$. □

Additionally, using a construction on base of $\mathtt{GT}_n$ we can prove a stronger separation between **Res** and 2-OBDD-$\mathbf{PS}_1$ than in Corollary 5.3.

**Theorem 6.4.** *There are a family of formulas $\{\phi_n\}_{n\in\mathbb{N}}$ and a constant $c > 0$ such that*

- $|\phi_n| = \mathrm{poly}(n)$;
- *there is a* 2-OBDD-$\mathbf{PS}_1$ *proof of $\phi_n$ of size* $\mathrm{poly}(n)$;
- *any resolution proof of $\phi_n$ has size $2^{\Omega(n^c)}$.*

For proving this result we need to introduce a notion of *width*: width of a resolution proof is the maximum size of a clause in the proof.

Alekhnovich and Razborov have proven a connection between the minimal width of a formula and the minimal size of a resolution proof of the xorification of this formula (composition of this formula and a function $x_1 \oplus x_2$).

**Theorem 6.5** ([5]). *There is a constant $c > 0$ such that, for any formula $\phi$ in $k$-CNF, if any resolution proof of $\phi$ has width at least $w$, then any resolution proof of $\phi \diamond \oplus_2$ has size at least $2^{c\cdot(w-k)}$.*

The plan of the proof of Theorem 6.4 is the following.

1. We construct a formula $\psi$ with a small resolution proof but any resolution proof of this formula has big size;

2. We prove that if formula $\phi$ has small ordered resolution proof, then $\phi \diamond \oplus_2$ has small 2-OBDD-$\mathbf{PS}_1$ proof;

3. In the last step we use Theorem 6.5 and two results from previous steps to show that $\psi \diamond \oplus_2$ is suitable for the statement of Theorem 6.4.

For the first step let us consider the following modification of the ordering principle. Formula $\mathtt{MGT}_n$ is the conjunction of the following clauses.

**antisymmetry:** $x_{i,j} \leftrightarrow \neg x_{j,i}$ for all $i \neq j \in [n]$;

**extended totality:**
- $\neg y_{0,j}$ for all $j \in [n]$;
- $y_{i-1,j} \vee x_i \vee \neg y_{i+1,j}$ for all $i \neq j \in [n]$;
- $y_{n,j}$ for all $j \in [n]$;

**transitivity:** $\neg x_{i_1,i_2} \vee \neg x_{i_2,i_3} \vee \neg x_{i_3,i_1}$ for all distinct $i_1, i_2, i_3 \in [n]$.

**Theorem 6.6** ([7])**.** *Width of any resolution proof of* $\mathtt{MGT}_n$ *is at least* $\Omega(n)$*. However, there is an ordered resolution proof of* $\mathtt{MGT}_n$ *of size* $\mathrm{poly}(n)$*.*

Proof of the second step statement is almost straightforward.

**Lemma 6.1.** *Let* $\phi$ *be an unsatisfiable formula in* $k$*-CNF. If there is an ordered resolution proof of* $\phi$ *of size* $S$*, then there is a* 2-OBDD-$\mathbf{PS}_1$ *proof of* $\phi \diamond \oplus_2$ *of size* $2^{2k+1}S$*.*

*Proof.* By Theorem 3.1 it is enough to construct a 2-OBDD for $\mathrm{Search}_{\phi \diamond \oplus_2}$ of size $2^{2k}S$.

Let $D$ be an OBDD of size at most $S$ for $\mathrm{Search}_\phi$ it exists since $\phi$ has ordered resolution proof of size at most $S$. Let $F^{(C)}$ be some OBDD such that if clause $C \diamond \oplus_2$ is false on the input, then $F^{(C)}$ finds which clause of $C \diamond \oplus_2$ is false. Note that for any clause $C$ there is a diagram $F^{(C)}$ of size at most $2^{2k}$. Let $D'$ be the following modification of $D$, we replace each node splitting by a variable $x$ by a diagram splitting by a value of $x \diamond \oplus_2$ and we replace each sink labeled by a clause $C$ by a diagram $F^{(C)}$.

It is clear, that $D'$ is a 2-OBDD for $\mathrm{Search}_{\phi \diamond \oplus_2}$ of size $2^{2k}S$. □

*Proof of Theorem 6.4.* Let us consider $\phi_n = \mathtt{MGT}_n \diamond \oplus_2$. It is easy to see, that size of this formula is $O(n)$. By Theorem 6.6 and Theorem 6.5 size of any resolution proof of $\phi$ is at least $2^{\Omega(n)}$. However, by Theorem 6.6 and Lemma 6.1 there is a 2-OBDD-$\mathbf{PS}_1$ proof of size $\mathrm{poly}(n)$. □

# 7 Acknowledgment

The author is grateful to Edward A. Hirsch for bringing the problem to his attention, to Dmitry Itsykson and Sam Buss for proofreading of the paper, and to Dmitry Sokolov for fruitful discussions about expander graphs.

# 8 Further Directions

1. Lower bounds for **RegRes** imply lower bounds for 1-BP-$\mathbf{PS}_1$, but for any $k > 0$ lower bounds for $(1, +k)$-BP-$\mathbf{PS}_1$ are unknown. Additionally, it is interesting to compare **Res** with $(1, +k)$-BP-$\mathbf{PS}_1$.

2. We proved that $k$-OBDD-$\mathbf{PS}_1$ does not $p$-simulates $(2k)$-OBDD-$\mathbf{PS}_1$. However, it is interesting to study relationships between $k$-OBDD-$\mathbf{PS}_1$ and $(k+1)$-OBDD-$\mathbf{PS}_1$.

3. Prove a superpolynomial lower bound for $\oplus$-OBDD-$\mathbf{PS}_1$ and study the relationship between **Res** and $\oplus$-OBDD-$\mathbf{PS}_1$.

# References


[1] Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart. An exponential separation between regular and general resolution. *Theory of Computing*, 3(1):81–102, 2007.

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

[3] Albert Atserias, Phokion G. Pg. 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.

[4] 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.

[5] Eli Ben-Sasson. Size space tradeoffs for resolution. In John H Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 457–464. ACM, 2002.

[6] Beate Bollig, Martin Sauerhoff, Detlef Sieling, and Ingo Wegener. Hierarchy theorems for k-OBDDs and k-IBDDs. *Theoretical Computer Science*, 205(1-2):45–60, 1998.

[7] Maria Luisa Bonet and Nicola Galesi. Optimality of size-width tradeoffs for resolution. *Computational Complexity*, 10(4):261–276, 2001.

[8] Henrik Brosenne, Matthias Homeister, and Stephan Waack. Nondeterministic ordered binary decision diagrams with repeated tests and various modes of acceptance. *Information Processing Letters*, 98(1):6–10, 2006.

[9] Joshua Buresh-Oppenheim and Toniann Pitassi. The complexity of resolution refinements. *Journal of Symbolic Logic*, 72(4):1336–1352, 2007.

[10] Sam Buss, Dmitry Itsykson, Alexander Knop, and Dmitry Sokolov. Reordering rule makes obdd proof systems stronger. Private communication.

[11] Susanna F. de Rezende, Jakob Nordström, and Marc Vinyals. How limited interaction hinders real communication (and what it means for proof and circuit complexity). In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 295–304. IEEE Computer Society, 2016.

[12] Yuval Filmus, Pavel Hrubes, and Massimo Lauria. Semantic versus syntactic cutting planes. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*, pages 35:1———35:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[13] Noah Fleming, Denis Pankratov, Toniann Pitassi, and Robert Robere. Random CNFs are hard for cutting planes. *Electronic Colloquium on Computational Complexity (ECCC)*, 45:1–19, 2017.

[14] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 32:1–32:17, 2016.

[15] Jordan Gergov and Christoph Meinel. Mod-2-OBDDs - a data structure that generalizes EXOR-sum-of-products and ordered binary decision diagrams. *Formal Methods in System Design*, 8(3):273–282, 1996.

[16] 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.

[17] Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing. *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 110–119, 2014.

[18] Johan Hastad. *Computational limitations of small-depth circuits*. MIT Press, Cambridge, MA, USA, 1987.

[19] Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: amplifying communication complexity hardness to time-space trade-offs in proof complexity. In *Proceedings of the 44th ACM Symposium on Theory of Computing, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 233–248, 2012.

[20] 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.

[21] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require Super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, 1990.

[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] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[24] Tak Wah Lam and Walter L. Ruzzo. Results on communication complexity classes. *Journal of Computer and System Sciences*, 44(2):324–342, 1992.

[25] László Lovász, Moni Naor, Ilan Newman, and Avi Wigderson. Search problems in the decision tree model. *SIAM Journal on Discrete Mathematics*, 8(1):119–132, 1995.

[26] Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[27] Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 410–418, 1991.

[28] Noam Nisan and Avi Wigderson. Rounds in communication complexity revisited. *SIAM Journal on Computing*, 22(1):211–219, 1993.

[29] Toniann Pitassi. Algebraic Propositional Proof Systems. In Neil Immerman and Phokion G Kolaitis, editors, *Descriptive Complexity and Finite Models, Proceedings of a DIMACS Workshop 1996, Princeton, New Jersey, USA, January 14-17, 1996*, volume 31 of {*DIMACS*} *Series in Discrete Mathematics and Theoretical Computer Science*, pages 215–244. DIMACS/AMS, 1996.

[30] Toniann Pitassi. Propositional proof complexity and unsolvability of polynomial equations. In *Proceedings of the International Congress of Mathematicians*, volume 3, pages 10–19, 1998.

[31] Petr Savický. A probabilistic nonequivalence test for syntactic (1,+k)-branching programs. *Electronic Colloquium on Computational Complexity (ECCC)*, 5(51), 1998.

[32] 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.

[33] Dmitry Sokolov. Dag-like communication and its applications. In Pascal Weil, editor, *Computer Science - Theory and Applications - 12th International Computer Science Symposium in Russia, CSR 2017, Kazan, Russia, June 8-12, 2017, Proceedings*, volume 10304 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2017.

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

[35] Stephan Waack. On the descriptive and algorithmic power of parity ordered binary decision diagrams. In Rüdiger Reischuk and Michel Morvan, editors, *STACS 97, 14th Annual Symposium on Theoretical Aspects of Computer Science, Lübeck, Germany, February 27 - March 1, 1997, Proceedings*, volume 1200 of *Lecture Notes in Computer Science*, pages 201–212. Springer, 1997.

[36] Ingo Wegener. *Branching Programs and Binary Decision Diagrams*. SIAM, 2000.

[37] 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.

[38] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 209–213, New York, NY, USA, 1979. ACM.



ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il