# Complexity of read-once branching programs for satisfiable and unsatisfiable Tseitin formulas


**Ludmila Glinskih**
St. Petersburg Department of V.A. Steklov Institute of Mathematics of the Russian Academy of Sciences, Fontanka 27, St. Petersburg, Russia
lglinskih@gmail.com

**Dmitry Itsykson**
St. Petersburg Department of V.A. Steklov Institute of Mathematics of the Russian Academy of Sciences, Fontanka 27, St. Petersburg, Russia
dmitrits@pdmi.ras.ru

**Artur Riazanov**
St. Petersburg Department of V.A. Steklov Institute of Mathematics of the Russian Academy of Sciences, Fontanka 27, St. Petersburg, Russia
St. Petersburg Academic University, Khlopina 8/3, St. Petersburg, Russia
aariazanov@gmail.com

**Petr Smirnov**
St. Petersburg State University, 7-9 Universitetskaya Emb., St. Petersburg, Russia
comradepetr@gmail.com



## Abstract

We study the read-once branching program complexity of two natural problems based on Tseitin formulas. We show that the complexity of computing of the value of a satisfiable Tseitin formula based on a graph $G$ and the complexity of finding a vertex with violated parity condition for an unsatisfiable Tseitin formula based on the same graph $G$ are quasipolynomialy related for read-once branching programs. Namely, if a minimum-size 1-BP for the vertex search problem has size $S$ then the smallest possible size of a 1-BP computing the value of a Tseitin formula based on the same graph is at least $S/n$ and at most $S^{\mathcal{O}(\log n)}$, where $n$ is the number of vertices in the graph.



**2012 ACM Subject Classification** Theory of computation → Computational complexity and cryptography

**Keywords and phrases** Tseitin formula, read-once branching program

**Digital Object Identifier** 10.4230/LIPIcs...

**Acknowledgements** The authors thank Alexander Smal for fruitful discussions.

## 1 Introduction

Let $\varphi$ be an unsatisfiable CNF-formula. We consider the search problem $\mathrm{Search}_\varphi$: given the values of the variables of $\varphi$, find a clause $\varphi$ which is falsified by these values. The search problem $\mathrm{Search}_\varphi$ is very important for the propositional proof complexity. For instance, it is known that the length of the shortest tree-like resolution refutation of $\varphi$ is equal to the size of the smallest decision tree for $\mathrm{Search}_\varphi$ and the length of the shortest regular resolution proof equals the size of the smallest read-once branching program for $\mathrm{Search}_\varphi$ [13]. Lower bounds for the randomized communication complexity of $\mathrm{Search}_\varphi$ imply the lower bounds for the proof complexity of $\varphi$ in tree-like proof systems where proof lines have small randomized communication complexity [1].

As mentioned before, regular resolution refutations of $\varphi$ are equivalent to read-once branching programs computing $\mathrm{Search}_\varphi$. Sometimes an unsatisfiable formula $\varphi$ naturally corresponds to a satisfiable formula $\varphi'$ which is obtained from $\varphi$, for example, by removing several clauses. It is interesting to study connections between read-once branching program complexities of computing $\mathrm{Search}_\varphi$ and computing $\varphi'$. Consider, for example, the pigeonhole principle $\mathrm{PHP}_n^{n+1}$ that states that it is possible that $n+1$ pigeons fly into $n$ holes such that no two pigeons fly into one hole. A natural satisfiable equivalent for $\mathrm{PHP}_n^{n+1}$ is $\mathrm{PHP}_n^n$ (the same statement for $n$ pigeons and $n$ holes). It is known that $\mathrm{PHP}_n^{n+1}$ requires resolution proofs of size $2^{\Theta(n)}$ [9] and that $\mathrm{PHP}_n^n$ requires read-once branching program of size $2^{\Theta(n)}$ [12].

In this paper we study Tseitin formulas constructed by an undirected graph $G(V,E)$ and a charge function $f : V \to \{0,1\}$, the variables of $\mathrm{Ts}_{G,f}$ correspond to the edges of the graph, the formula itself is the conjunction of the parity conditions for the vertices of $G$. A parity condition for a vertex $v \in V$ states that the sum of the values of the variables corresponding to the edges incident to $v$ equals $f(v)$ modulo 2. A Tseitin formula $\mathrm{Ts}_{G,f}$ is satisfiable iff for every connected component of $G$ the sum of the values of $f(v)$ is even [16]. Unsatisfiable Tseitin formulas are important hard examples for various propositional proof systems [15, 16, 3, 14, 10, 5, 8]. It is known that unsatisfiable Tseitin formulas based on an expander graph with $n$ vertices require resolution proofs of size $2^{\Theta(n)}$ [16], the expander-based satisfiable Tseitin formulas require read-once branching programs of size $2^{\Theta(n)}$ [6]. In contrast to the pigeonhole principle the complexity of a Tseitin formula may vary for different graphs. Thus it is interesting to establish a connection between the 1-BP complexity of a satisfiable Tseitin formula and the regular resolution complexity problem of an unsatisfiable Tseitin formula based on the same graph.

For an unsatisfiable Tseitin formula $\mathrm{Ts}_{G,c}$ we introduce an auxiliary search problem $\mathrm{SearchVertex}_{G,c}$: for the given values of the variables of $\mathrm{Ts}_{G,c}$ find a vertex of $G$ with violated parity condition. The problem $\mathrm{SearchVertex}_{G,c}$ is not harder than $\mathrm{Search}_{\mathrm{Ts}_{G,c}}$ since given a falsified clause it is easy to find a vertex with violated parity condition. Consider a complete graph on $\log n$ vertices $K_{\log n}$. Using the technique connecting the expansion of a graph with the resolution width of the corresponding Tseitin formula [2] it is easy to show that the length of the shortest resolution refutation of $\mathrm{Ts}_{K_{\log n},c}$ is $2^{\Omega(\log^2 n)}$ (see [4] for details). On the other hand 1-BP complexity of $\mathrm{Ts}_{K_{\log n},c'}$ is $\mathrm{poly}(n)$ for any $c'$ [6] and, as it follows from our results, the 1-BP complexity of $\mathrm{SearchVertex}_{K_{\log n},c}$ is $\mathrm{poly}(n)$ as well. In the context of 1-BP the problem $\mathrm{SearchVertex}_{G,c}$ is interesting even for graphs with large degrees, while in this case the complexity of $\mathrm{Search}_{\mathrm{Ts}_{G,c}}$ is automatically large since the number of clauses is exponential in the maximum degree of the graph.

It is easy to find an example of a family of graphs $G_n$ such that there is a polynomial-sized decision tree computing $\mathrm{SearchVertex}_{G_n,c}$ (if $\mathrm{Ts}_{G_n,c}$ is satisfiable) but every decision tree for a satisfiable formula $\mathrm{Ts}_{G_n,c'}$ has exponential size. Consider a path of length $n$ with an additional parallel edges between every pair of the consecutive vertices of the path and denote the resulting graph by $P_n$. Let the charge function be zero for all the vertices. The resulting Tseitin formula is satisfiable and has exactly $2^n$ satisfying assignments (an assignment satisfies the formula iff for every pair of parallel edges the values assigned to them are the same). Since each path from the root to an accepting leaf of a decision tree computing a satisfiable Tseitin formula has to test all edges of the graph, no two satisfying assignments correspond to the same leaf. Thus any decision tree for $P_n$ has size at least $2^n$. The Tseitin formula on the same graph and with exactly one of the charges equal to 1 is unsatisfiable and there exists a decision tree of size $\mathcal{O}(n^2)$ computing SearchVertex for this formula. Indeed, we can branch on the values of two central edges and for each of the four possible substitutions only one of the two connected parts of $P_n$ is unsatisfiable, so we will search the vertex with violated parity condition in a graph that has twice smaller size. The size of the resulting decision tree can be determined by the recurrence $S(n) = 4S(n/2)$ and hence $S(n) = \mathcal{O}(n^2)$.

**Our results.** In Theorem 5 we show that the existence of 1-BP of size $S$, computing a satisfiable Tseitin formula $\mathrm{Ts}_{G,c}$ based on a connected graph $G$ implies the existence of 1-BP of size $nS$, computing $\mathrm{SearchVertex}_{G,c'}$ where $\mathrm{Ts}_{G,c'}$ is unsatisfiable.

In Theorem 8 we show that the existence of a 1-BP of size $S$ computing $\mathrm{SearchVertex}_{G,c}$ for an unsatisfiable Tseitin formula $\mathrm{Ts}_{G,c}$ based on a connected graph implies the existence of 1-BP of size $S^{\mathcal{O}(\log n)}$ computing $\mathrm{Ts}_{G,c'}$.

The proofs of Theorems 5 and 8 are based on the structural characterizations of 1-BP computing a Tseitin formula or a SearchVertex problem of an unsatisfiable Tseitin formula. Lemma 6 states that each node of a minimum-size 1-BP, computing a Tseitin formula, computes the Tseitin formula obtained from the initial one by the substitution according to any path from the source to the node. This lemma is relatively simple and was known before. Lemma 9 states that if $D$ is a minimum-size 1-BP computing $\mathrm{SearchVertex}_{G,c}$, then each node $s$ of $D$ computes $\mathrm{SearchVertex}_{H,f}$ such that for every substitution $\alpha$ corresponding to a path from the source to $s$, if we apply $\alpha$ to $\mathrm{Ts}_{G,c}$ the resulting Tseitin formula contains exactly one unsatisfiable connected component (i.e. connected component with odd sum of charges) $H$ and $f$ is a restriction of its charge function to $H$. This Lemma is not trivial and requires a careful proof. Therefore, each node $s$ of a minimum-size 1-BP for each of the two problems corresponds to a graph $G_s$ and a charge function $c_s$ such that the node $s$ computes the same problem as the whole 1-BP but for $(G_s, c_s)$.

The transformation of 1-BP for one of the problems into a 1-BP of the other problem is possible because the dependency between a node and its direct successors behaves in a similar way for both problems. Namely, if a node labeled with $x_e$, where $e$ is not a bridge of the graph corresponding to the node, then for both problems the graphs and the charges of the direct successors differs from the parent's graph and charge in the exact same way. If $e$ is a bridge of the graph $G_s$ corresponding to the node of a 1-BP computing the value of a satisfiable formula, the substitution of one of the values of $x_e$ makes the formula unsatisfiable and thus the corresponding edge goes to 0-sink, the other edge goes to the node corresponding to the graph $G_s - e$ that has one more connected component with respect to $G_s$. If $s$ is a node of a 1-BP computing a SearchVertex problem, then $G_s$ is connected graph. Thus if $e$ is a bridge of $G_s$, then one of the successors corresponds to the first connected component of $G_s - e$ and the other successor corresponds to the another component.

**Further research.** In the recent work Glinskih and Itsykson [7] proved that there exists a constant $\delta > 0$ such that the 1-BP complexity of computing a satisfiable Tseitin formula

$\mathrm{Ts}_{G,c}$ is at least $2^{\mathrm{tw}(G)^\delta}$, where $\mathrm{tw}(G)$ is the tree-width of the graph $G$. Our results imply the same lower bound but with a smaller $\delta$ holds for $\mathrm{SearchVertex}_{G,c'}$ and therefore for the regular resolution refutations of $\mathrm{Ts}_{G,c'}$.

Open questions: is it possible to prove a polynomial equivalence of computing SearchVertex and computing a Tseitin formula for the same graph by 1-BPs? Is it possible to separate the complexity of the search for a falsified clause from the complexity of the search for a falsified vertex for bounded degree graphs?

## 2 Preliminaries

### 2.1 Tseitin formulas

Let $G(V, E)$ be an undirected graph without loops but possibly with parallel edges. Let $c : V \to \{0, 1\}$ be a charge function. A Tseitin formula $\mathrm{Ts}_{G,c}$ depends on the propositional variables $x_e$ for $e \in E$. For each vertex $v \in V$ we define the parity condition in $v$ as $P_v \coloneqq (\sum_{e \text{ is incident to } v} x_e = c(v) \bmod 2)$. The Tseitin formula $\mathrm{Ts}_{G,c}$ is the conjunction of all the parity conditions of all the vertices $\bigwedge_{v\in V} P_v$.

In this paper we define a connected component of a graph $G$ as a maximal inclusive connected subgraph of $G$. Assume that $G$ consists of connected components $H_1, H_2, \dots, H_t$. Then a Tseitin formula $\mathrm{Ts}_{G,c}$ is equivalent to the conjunction $\bigwedge_{i=1}^t \mathrm{Ts}_{H_i,c}$. In the last formula we abuse the notation in the following way: we allow that the charge function $c$ is defined not only on the vertices of the graph and we implicitly use the corresponding restriction on the set of vertices.

▶ **Lemma 1** ([16]). *A Tseitin formula $\mathrm{Ts}_{G,c}$ is satisfiable if and only if for every connected component with a set of vertices $U$ of the graph $G$ the condition $\sum_{u\in U} c(u) = 0 \bmod 2$ holds.*

▶ **Lemma 2** ([11], Lemma 2.3). *Let $G(V, E)$ be a connected graph and let $c : V \to \{0, 1\}$ be a charge function. Let $U \subsetneq V$ and $\Phi = \bigwedge_{v\in U} P_v$ be the conjunction of the parity conditions for all vertices from $U$. Then $\Phi$ is satisfiable.*

▶ **Lemma 3.** *The result of the substitution $x_e \coloneqq b$ to $\mathrm{Ts}_{G,c}$ where $b \in \{0, 1\}$ is a Tseitin formula $\mathrm{Ts}_{G',c'}$ where $G' = G - e$ and $c'$ differs from $c$ on the endpoints of the edge $e$ by $b$ and equals $c$ for every other vertex.*

**Proof.** The proof is straightforward. ◀

▶ **Lemma 4.** *Let $G(V, E)$ be a connected graph and let $c_1, c_2 : V \to \{0, 1\}$ be charge functions. If Tseitin formulas $\mathrm{Ts}_{G,c_1}$ and $\mathrm{Ts}_{G,c_2}$ are both satisfiable or both unsatisfiable, then one of them can be obtained from another by replacing some variables with their negations.*

**Proof.** See Appendix A. ◀

For a graph $G(V, E)$ and a charge function $c : V \to \{0, 1\}$ we define a relation $\mathrm{SearchVertex}_{G,c}$ consisting of the pairs $(\sigma, v)$ where $\sigma : \{x_e \mid e \in E\} \to \{0, 1\}$ and $v \in V$ such that $\sum_{e \text{ is incident to } v} \sigma(x_e) \neq c(v) \bmod 2$. If a Tseitin formula $\mathrm{Ts}_{G,c}$ is unsatisfiable then the relation $\mathrm{SearchVertex}_{G,c}$ is total i.e. for every $\sigma : \{x_e \mid e \in E\} \to \{0, 1\}$ there exists $v \in V$ such that $(\sigma, v) \in \mathrm{SearchVertex}_{G,c}$. We consider this relation as the following search problem: given the values of the variables find a vertex with violated parity condition.

## 2.2 Branching programs

A branching program is a way to represent a function $f : \{0,1\}^n \to K$, where $K$ is a finite set. The function $f(x_1, x_2, \dots, x_n)$ is represented by a directed acyclic graph with $|K|$ sinks, sinks are labeled with different elements of the set $K$, each of the remaining nodes is labeled with a variable from $\{x_1, x_2, \dots, x_n\}$ and has exactly two outgoing edges, the first is labeled with 0, the second is labeled with 1. Each node $v$ of a branching program computes a function $f_v : \{0,1\}^n \to K$. The sink labeled with $k \in K$ computes the constant $k$. Assume that a node $v$ is labeled with $x_i$, the outgoing edge from $v$ labeled with 0 ends in a node $v_0$ and the outgoing edge labeled with 1 ends in a node $v_1$. Then $f_v(x_1, \dots, x_n)$ equals $f_{v_1}(x_1, \dots, x_n)$ if $x_i = 1$ and equals $f_{v_0}(x_1, \dots, x_n)$ if $x_i = 0$. It is usually assumed that a branching program has only one source, in that case we say that the branching program computes the function computed in its source. We refer to a sink labeled with $k \in K$ as $k$-sink.

We say that a branching program computes a relation $Q \subseteq \{0,1\}^n \times K$ if it computes a function $f : \{0,1\}^n \to K$ such that for every $x \in \{0,1\}^n$ the condition $(x, f(x)) \in Q$ holds.

If $D$ is a branching program computing a function $f(x_1, x_2, \dots, x_n)$ then for every full assignment $\sigma : \{x_1, \dots, x_n\} \to \{0,1\}$ we denote $D(\sigma) = f(\sigma(x_1), \dots, \sigma(x_n))$.

A branching program is called read-once (1-BP) if for every path in the program the nodes have distinct labels.

# 3 $\mathrm{SearchVertex}_{G,c}$ is not harder than $\mathrm{Ts}_{G,c'}$

In this section we prove the following theorem:

▶ **Theorem 5.** *Let $G(V,E)$ be an undirected connected graph and a Tseitin formula $\mathrm{Ts}_{G,c}$ be satisfiable and $\mathrm{Ts}_{G,c'}$ be unsatisfiable. Assume that there exists a* 1-BP *computing $\mathrm{Ts}_{G,c}$ of size $S$. Then there exists a* 1-BP *computing $\mathrm{SearchVertex}_{G,c'}$ of size at most $|V|S$.*

We need the following lemma characterizing functions computed in nodes of 1-BP representing a Tseitin formula.

▶ **Lemma 6** (partial case of ([6], Claim 15))**.** *Let $D$ be a* 1-BP *computing a Tseitin formula $\mathrm{Ts}_{G,c}$. Then for every node $s$ of $D$, such that* 1*-sink is reachable from $s$ and for every two paths $p_1$ and $p_2$ from the source to $s$ the following conditions hold:*

1. *The sets of labels of the nodes of $p_1$ and $p_2$ are equal.*
2. *Let $\alpha_i$ be a partial assignment according to the path $p_i$ for $i \in \{1,2\}$. Then Tseitin formulas $\mathrm{Ts}_{G,c}|_{\alpha_1}$ and $\mathrm{Ts}_{G,c}|_{\alpha_2}$ are equal.*

**Proof of Theorem 5.** Let $D$ be a minimal 1-BP computing Tseitin formula $\mathrm{Ts}_{G,c}$ and let $S$ be the size of $D$ (the number of nodes including the sinks). Let $n = |V|$.

We enumerate all nodes of $D$, except the 0-sink, in a reverse topological order: $u_1, u_2, \dots, u_{S-1}$ (i.e. any edge of $D$ is directed from a node with the greater number to a node with the less number).

Since $D$ is a minimal 1-BP, 1-sink is reachable from all nodes except the 0-sink, thus by Lemma 6 every node of $D$, except the 0-sink, computes the Tseitin formula obtained from $\mathrm{Ts}_{G,c}$ by a substitution according to any path from the source to this node. We denote the Tseitin formula corresponding to $u_i$ by $\mathrm{Ts}_{G_i,c_i}$.

▶ Claim 7. For every $k \in [S-1]$ there exists a 1-BP $D^{(k)}$ of size at most $kn$ (sinks are included in the number of nodes), such that:

— for every $i \in [k]$

— for every connected component $H$ of $G_i$
— for every charge function $c'_i$, which differs from $c_i$ in exactly one vertex of $H$
there exists a node of $D^{(k)}$ computing $\mathrm{SearchVertex}_{H,c'_i}$.

Moreover, each node $s$ of $D^{(k)}$ computing $\mathrm{SearchVertex}_{H,f}$ for a connected component $H$ and a charge function $f$ such that the Tseitin formula $\mathrm{Ts}_{H,f}$ is unsatisfiable. If the node $s$ is labeled with $x_e$ then $e$ is an edge of $H$ and each of the direct successors of the node $s$ computes $\mathrm{SearchVertex}_{H',f'}$, where $H'$ is a subgraph of $H - e$. The latter statement implies that $D^{(k)}$ is a 1-BP.

Let us stress that Claim 7 for $k = S - 1$ implies the statement of Theorem 5 since $G$ is connected and thus the diagram $D^{(S-1)}$ contains a node computing $\mathrm{SearchVertex}_{G,c''}$, where $c$ and $c''$ differ in exactly one vertex. By Lemma 4 there exists a 1-BP of size at most $nS$ computing $\mathrm{SearchVertex}_{G,c'}$.

**Proof of Claim 7.** The proof is by induction on $k$. Base case: $k = 1$, then $u_1$ is the 1-sink, $G_{u_1}$ is the empty graph and $c_1$ is identically 0. In this case the program $D^{(1)}$ that consists of $n$ sinks labeled with the vertices from $V$ satisfies the conditions of the claim.

Induction step from $k-1$ to $k$. By the induction hypothesis there exists $D^{(k-1)}$ satisfying the conditions for $i \in [k-1]$. Let us add some nodes to $D^{(k-1)}$ such that the condition will by satisfied for $i = k$ as well.

For a graph $H$ and its vertex $v$ we denote by $\mathrm{CC}(H, v)$ the connected component of $H$ containing $v$. For a charge function $c : V \to \{0, 1\}$ and a vertex $j \in V$ denote by $c^{\oplus j} : V \to \{0, 1\}$ the charge function which differs from $c$ on the vertex $j$ and nowhere else.

Let us assume the node $u_k$ of $D$ is labeled with $x_e$. Let $u_{k_0}$ be the endpoint of the edge outgoing from $u_k$ labeled with 0 and $u_{k_1}$ be the endpoint of the edge outgoing from $u_k$ labeled with 1.

By the induction hypothesis for $k_0$ and $k_1$ for every $l \in [n]$ there exists a node $s^0_l$ of the program $D^{(k-1)}$ computing $\mathrm{SearchVertex}_{\mathrm{CC}(G_{k_0},l),c^{\oplus l}_{k_0}}$, and a node $s^1_l$ computing $\mathrm{SearchVertex}_{\mathrm{CC}(G_{k_1},l),c^{\oplus l}_{k_1}}$.

We need to construct a program $D^{(k)}$ such that for every vertex $j \in V$ there exists a node $s_j$ of $D^{(k)}$ computing $\mathrm{SearchVertex}_{\mathrm{CC}(G_k,j),c^{\oplus j}_k}$. For some values of $j \in V$, $D^{(k-1)}$ already contains such a node and for some values of $j$ we will add a node $s_j$ explicitly.

We consider three cases:

***e* is outside of $\mathbf{CC}(G_k, j)$.** In this case the components $\mathrm{CC}(G_k, j)$ and $\mathrm{CC}(G_{k_0}, j)$ are equal, and $c_k$ and $c_{k_0}$ are equal on the vertices of this connected component. Then we may take $s_j = s^0_j$.

***e* is in $\mathbf{CC}(G_k, j)$, and *e* is not a bridge.** In this case we add a new node $s_j$ labeled with $x_e$ and add an edge from $s_j$ to $s^0_j$ labeled with 0 and an edge to $s^1_j$ labeled with 1. Since $e$ is not a bridge, $\mathrm{CC}(G_{k_0}, j)$ and $\mathrm{CC}(G_{k_1}, j)$ are equal to each other and to $\mathrm{CC}(G_k, j) - e$. Thus the node $s^0_j$ computes $\mathrm{SearchVertex}_{\mathrm{CC}(G_k,j)-e,c^{\oplus j}_{k_0}}$, and $s^1_j$ computes $\mathrm{SearchVertex}_{\mathrm{CC}(G_k,j)-e,c^{\oplus j}_{k_1}}$. By Lemma 3 the charge function $c_{k_0}$ equals $c_k$ and $c_{k_1}$ can be obtained from $c_k$ by flipping the values on the endpoints of the edge $e$. Hence $s_j$ computes $\mathrm{SearchVertex}_{\mathrm{CC}(G_k,j),c^{\oplus j}_k}$.

***e* is a bridge of $\mathbf{CC}(G_k, j)$.** Let $A$ and $B$ be two components of the graph $\mathrm{CC}(G_k, j)-e$, such that $A$ contains the vertex $j$. Let $a \in A$, $b \in B$ be the endpoints of the edge $e$.

Since $\mathrm{Ts}_{G_k,c_k}$ is satisfiable, then Lemma 1 implies that there exists exactly one value $\gamma \in \{0, 1\}$ such that $\mathrm{Ts}_{G_k,c_k}|_{x_e:=\gamma}$ is satisfiable. Since $D$ is minimal, the edge from $u_k$ labeled with $1 - \gamma$ goes to the 0-sink.

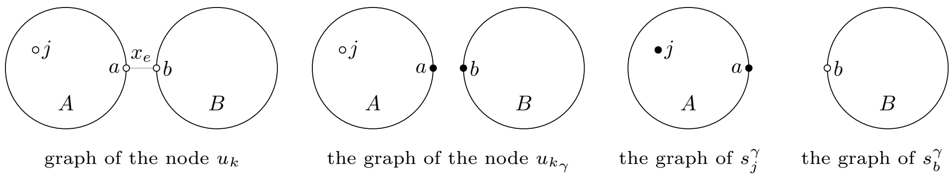


**Figure 1** The graphs of the nodes (for $\gamma = 1$); nodes with the charge different from $c_k$ are black, nodes with the same charge as in $c_k$ are white.

We add a new node $s_j$ labeled with $x_e$, an edge from $s_j$ to $s_j^\gamma$ labeled with $\gamma$ and an edge from $s_j$ to $s_b^{1-\gamma}$ labeled with $1-\gamma$. (See Fig. 1.)

It is easy to see that $A = \mathrm{CC}(G_{k_\gamma}, j)$. Thus $s_j^\gamma$ computes $\mathrm{SearchVertex}_{A,c_{k_\gamma}^{\oplus j}}$ and $c_{k_\gamma}$ equals $c_k$ on the vertices of the graph $A$ except maybe the vertex $a$ and $c_{k_\gamma}(a) = c_k(a) + \gamma \mod 2$ (see Figure 2 for clarification). Since $b \in B$ the vertex $s_b^\gamma$ computes $\mathrm{SearchVertex}_{B,c_{k_\gamma}^{\oplus b}}$ and $c_{k_\gamma}$ equals $c_k$ on the vertices of $B$ except maybe the vertex $b$ and $c_{k_\gamma}(b) = c_k(b) + \gamma \mod 2$. Thus $c_{k_\gamma}^{\oplus b}$ equals $c_k$ on the vertices of $B$ except maybe the vertex $b$ and $c_{k_\gamma}^{\oplus b}(b) = c_k(b) + \gamma + 1 = c_k(b) + (1-\gamma) \mod 2$. This implies that $s_j$ computes $\mathrm{SearchVertex}_{\mathrm{CC}(G_k,j),c_k^{\oplus j}}$. ◀

◀

## 4 $\mathrm{Ts}_{G,c}$ is at most quasipolynomialy harder than $\mathrm{SearchVertex}_{G,c'}$

In this section we prove the following theorem:

▶ **Theorem 8.** *Let $G(V,E)$ be an undirected connected graph and a Tseitin formula $\mathrm{Ts}_{G,c}$ be satisfiable and $\mathrm{Ts}_{G,c'}$ be unsatisfiable. Assume that there exists a 1-BP computing $\mathrm{SearchVertex}_{G,c'}$ of size $S$. Then there exists a 1-BP computing $\mathrm{Ts}_{G,c}$ of size at most $S^{\mathcal{O}(\log|V|)}$.*

In the following subsection we study the structure of 1-BPs computing SearchVertex. In the subsection 4.2 we prove Theorem 8 itself.

### 4.1 The structure of 1-BP computing SearchVertex

Let $D$ be a 1-BP that computes $\mathrm{SearchVertex}_{G,c}$, where $G(V,E)$ is a connected graph and $\mathrm{Ts}_{G,c}$ is unsatisfiable. For any internal node $s$ of $D$ we denote by $h(s)$ the set of labels of sinks reachable from $s$. We denote by $P(s)$ the set of partial assignments corresponding to the paths from the source of $D$ to $s$.

Let $F(V,E)$ be an undirected (not necessary connected) graph and let $H$ be a connected component of $F$. We call $H$ *a satisfiable component* of a formula $\mathrm{Ts}_{F,c}$ if the formula $\mathrm{Ts}_{H,c}$ is satisfiable. Otherwise we call $H$ an unsatisfiable component of the formula $\mathrm{Ts}_{F,c}$.

In this subsection we prove the following Lemma.

▶ **Lemma 9.** *Let $G(V,E)$ be a connected graph and let $c$ be such that $\mathrm{Ts}_{G,c}$ is unsatisfiable. Let $D$ be a minimum-size 1-BP computing $\mathrm{SearchVertex}_{G,c}$. Then every node $s$ of $D$ computes $\mathrm{SearchVertex}_{H,f}$ such that for every $\alpha \in P(s)$, $H$ is the only unsatisfiable component of the formula $\mathrm{Ts}_{G,c}|_\alpha$ and $f$ is the restriction of the charge function of $\mathrm{Ts}_{G,c}|_\alpha$ to the vertices of $H$.*

▶ Proposition 10. Let $s$ be an internal node of $D$. Let $\alpha_1$ and $\alpha_2$ be the assignments from $P(s)$. Then the following conditions hold:

1. For every edge $e \in E$ incident to a vertex in $h(s)$, $\alpha_1$ assigns a value to the variable $x_e$ iff $\alpha_2$ does.
2. For every $v \in h(s)$ the charge of $v$ in $\mathrm{Ts}_{G,c}|_{\alpha_1}$ is equal to the charge of $v$ in $\mathrm{Ts}_{G,c}|_{\alpha_2}$.

**Proof.** Consider a vertex $v \in h(s)$ and consider some sink $t$ labeled with $v$ that is reachable from $s$. Let $\beta$ be a partial assignment corresponding to a path from $s$ to $t$. Notice that the set of variables assigned by $\beta$ does not intersect the set of variables assigned by $\alpha_i$ for $i \in \{1, 2\}$ since $D$ is a 1-BP. Let us define $\rho_i = \alpha_i \cup \beta$ for $i \in \{1, 2\}$.

Both the assignments $\rho_1$ and $\rho_2$ falsify the vertex $v$. Thus, for every edge $e$ incident to $v$ the value for $x_e$ is assigned by $\rho_1$ and by $\rho_2$. Thus $\alpha_1$ and $\alpha_2$ assign values to the same subset of variables among $\{x_e \mid e \text{ is incident to } v\}$ and the sums modulo 2 of the values assigned to these variables by $\alpha_1$ and $\alpha_2$ are the same. ◀

By Lemma 3 the result of the substitution of a value to a variable of a Tseitin formula is a Tseitin formula as well. For an arbitrary assignment $\alpha$ from $P(s)$ we denote by $G_{s,\alpha}$ and $c_{s,\alpha}$ a graph and a charge function such that $\mathrm{Ts}_{G,c}|_\alpha$ is precisely $\mathrm{Ts}_{G_{s,\alpha},c_{s,\alpha}}$.

Notice that if for some $\alpha \in P(s)$, $C$ is an unsatisfiable component of $\mathrm{Ts}_{G_{s,\alpha},c_{s,\alpha}}$ and all its vertices are contained in $h(s)$, then by Proposition 10, $C$ is an unsatisfiable component with respect to all partial assignments from $P(s)$. Let $U(s)$ be the set of all unsatisfiable components of $\mathrm{Ts}_{G_{s,\alpha},c_{s,\alpha}}$ contained in $h(s)$, where $\alpha$ is some partial assignment from $P(s)$. By the remark above $U(s)$ does not depend on $\alpha$.

Consider some $\alpha \in P(s)$. Let $H(V_H, E_H)$ be a connected component of $G_{s,\alpha}$ that contains at least one vertex from $h(s)$. Then there are three possible cases (three types of a component $H$ with respect to a node $s$ and a partial assignment $\alpha$):

(1) $V_H \subseteq h(s)$ and $H$ is unsatisfiable connected component of $\mathrm{Ts}_{G_{s,\alpha},c_{s,\alpha}}$. I.e. $H \in U(s)$;

(2) $V_H \subseteq h(s)$ and $H$ is satisfiable connected component of $\mathrm{Ts}_{G_{s,\alpha},c_{s,\alpha}}$;

(3) $V_H \not\subseteq h(s)$.

▶ Proposition 11. Let $s$ be an internal node of $D$. Then $U(s)$ is not empty. In other words, for any partial assignment $\alpha \in P(s)$ the formula $\mathrm{Ts}_{G_{s,\alpha},c_{s,\alpha}}$ contains at least one unsatisfiable component $C$ of type (1).

**Proof.** Assume for the sake of contradiction that all connected components of $G_{s,\alpha}$ intersecting $h(s)$ have type (2) or (3).

Consider a formula $\Psi$ which is a part of $\mathrm{Ts}_{G_{s,\alpha},c_{s,\alpha}}$ and it is the conjunction of the parity conditions of the vertices from $h(s)$. The formula $\Psi$ is the conjunction of several formulas depending on disjoint sets of variables corresponding to connected components of $G_{s,\alpha}$. Since each connected component of the graph $G_{s,\alpha}$ has type (2) or (3), each of these formulas is either a satisfiable Tseitin formula or a subset of a Tseitin formula based on a connected graph. By Lemma 2 a formula in the latter case is also satisfiable. Thus $\Psi$ itself is satisfiable. Let $\beta$ be a full assignment which satisfies $\Psi$ and agrees with $\alpha$. By the construction $\beta$ does not falsify any vertex of $h(s)$.

We get a contradiction as follows. We consider the path in $D$ from the source to a sink corresponding to $\beta$. By the construction the path goes through the node $s$ and hence ends in a sink labeled with an element of $h(s)$. Thus $\beta$ falsifies a vertex from $h(s)$ and this contradicts the construction of $\beta$.

Thus, there exists a connected component $S \subseteq h(s)$ such that $S$ is an unsatisfiable component of $\mathrm{Ts}_{G_{s,\alpha},c_{s,\alpha}}$. Therefore, $U(s)$ is not empty. ◀

▶ Proposition 12. Let $D$ be a *minimum-size* 1-BP that computes $\mathrm{SearchVertex}_{G,c}$. Then any internal node $s$ of $D$ is labeled with an edge incident to $h(s)$.

**Proof.** Assume that for an inner node $s$ labeled with $x_e$ the statement is false i.e. $e$ connects two vertices outside $h(s)$. Let $t_0$ and $t_1$ be the direct successors of $s$ such that the edge $(s, t_i)$ is labeled with $i$. Let us modify $D$ as follows: remove the edge $(s, t_0)$ and contract the edge $(s, t_1)$. We denote the result of the contraction by $s'$ and label it with the label of $t_1$ in $D$. The resulting diagram $D'$ is strictly smaller than $D$. We claim that $D'$ also computes $\mathrm{SearchVertex}_{G,c}$. Consider a full assignment $\beta$. Let $\beta'(x_q) = \begin{cases} \beta(x_q) & q \neq e \\ 1 & q = e \end{cases}$.

If the path in $D$ corresponding to $\beta$ does not pass through $s$, then exactly the same path with the same labels is contained in $D'$ thus $D'(\beta) = D(\beta)$. Hence it is sufficient to consider the case where the path in $D$ corresponding to $\beta$ passes through the node $s$. In this case the path in $D'$ corresponding to $\beta'$ passes through $s$ as well, because among the nodes of any path from the source to $s$ only $s$ is labeled with $x_e$. Then $D(\beta') \in h(s)$. The edge $e$ is not incident to any vertex from $h(s)$ thus $e$ is not incident to the vertex $D(\beta')$. Since the vertex $D(\beta')$ is falsified by $\beta'$ and $e$ is not incident to $D(\beta')$, then $D(\beta')$ is falsified by $\beta$ as well. By the construction of $D'$ the equality $D(\beta') = D'(\beta)$ holds. Thus, $\beta$ falsifies $D'(\beta)$. Therefore, $D'$ correctly computes $\mathrm{SearchVertex}_{G,c}$ and the size of $D'$ is strictly less then the size of $D$, this is a contradiction. ◀

▶ Proposition 13. Let $D$ be a *minimum-size* 1-BP that computes $\mathrm{SearchVertex}_{G,c}$. Then any internal node $s$ of $D$ is labeled with a variable $x_e$, where $e$ connecting two vertices of a component from the set $U(s)$.

**Proof.** By Proposition 12 we may assume that for every node $l$ of $D$ if $l$ is labeled by $x_e$, then $e$ is incident to a vertex of $h(l)$.

Assume that the statement of the proposition is false. Let us fix the deepest (i.e. the farthest from the source) node $s$ of $D$ violating the statement. Let $s$ be labeled by $x_e$. Let $t_0$ and $t_1$ be the direct successors of $s$ and the edge $(s, t_i)$ be labeled with $i$ for $i \in \{0, 1\}$. Let $\alpha$ be an assignment corresponding to some path from the source to $s$.

Since $s$ violates the statement, $e$ connects two vertices of a satisfiable component of $G_{s,\alpha}$ or connects two vertices of a component containing a vertex outside $h(s)$.

Let $C(V_C, E_C)$ be the connected component of $G_{s,\alpha}$ containing the edge $e$.

Let us consider an arbitrary partial assignment $\theta$ which satisfies all vertices of $V_C \cap h(s)$ and does not assign any variable corresponding to an edge of $G_{s,\alpha}$ outside $C$ (if $C$ is satisfiable, $\theta$ exists by definition and if $C$ contains a vertex from the outside of $h(s)$, $\theta$ exists by Lemma 2).

▶ Claim 14. Consider a path $\tau = (\tau_1, \ldots, \tau_m)$ from $t_{\theta(x_e)}$ in $D$ to a sink node. Then for every label $x_{e'}$ of a node of $\tau$, $e'$ is not incident to a vertex from $C$.

**Proof.** Assume for the sake of contradiction that there exists a node violating the statement. Let $i \in [m]$ be the smallest index such that $\tau_i$ is labeled by $x_{e'}$ and $e'$ is incident to a vertex from $C$.

Since $\tau_i$ is a successor of $s$, $h(\tau_i) \subseteq h(S)$. By Proposition 12 the edge $e'$ is incident to $h(\tau_i)$. We are going to show that $e'$ is not contained in an unsatisfiable component inside $h(\tau_i)$ and get a contradiction with the assumption that $s$ is the deepest node violating the statement of the proposition.

Assume that $e'$ is contained in an unsatisfiable component $C' \subseteq h(\tau_i)$. As it was mentioned before the structure of such components is independent of a path from the source to $\tau_i$, thus we choose a path that agrees with $\alpha$ on the path from the source to $s$ and then continues as $\tau_1, \dots, \tau_i$. Let $\mu$ be the partial assignment corresponding to this path. $\mu$ extends $\alpha$, hence the graph $G_{\tau_i,\mu}$ is a subgraph of $G_{s,\alpha}$. $C$ is a connected component of $G_{s,\alpha}$. $C$ and $C'$ has a common edge $e'$. Thus $C'$ is a subgraph of $C$. The assignment $\theta$ satisfies the Tseitin formula corresponding to the connected component $C$ and the charge function $c_{s,\alpha}$. Moreover, $\theta$ and $\mu$ have only one common variable in their domains. That variable is $x_e$ and $\mu$ agrees with $\theta$ on $x_e$ by the construction. Therefore, $\mu$ has a full extension that agrees with $\theta$. But that extension satisfies all parity conditions of the vertices from $C'$ which contradicts unsatisfiability of $C'$ with respect to $\tau_i$. ◀

Consider a diagram $D'$ obtained by the removing the edge $(s, t_{1-\theta(e)})$ and the contraction of the edge $(s, t_{\theta(e)})$ where $t_i$ is as before. Since $D'$ is smaller than $D$ and $D$ is a minimal 1-BP computing $\mathrm{SearchVertex}_{G,c}$, there exists a full assignment $\beta$ such that the vertex $D'(\beta)$ is not falsified by $\beta$. The path in $D'$ corresponding to $\beta$ passes through $s$ since otherwise $D'(\beta) = D(\beta)$ which is falsified by $\beta$.

Let $\beta'(x_q) = \begin{cases} \beta(x_q) & q \neq e \\ \theta(x_e) & q = e \end{cases}$.

Let $v = D'(\beta)$.

▶ Claim 15. $v$ is incident to $e$.

**Proof.** As in the proof of Proposition 12, $D(\beta') = D'(\beta) = v$ and by the correctness of $D$, $\beta'$ falsifies $v$. On the other hand $\beta$ does not falsify $v$ by the choice of $\beta$. Since $\beta'$ and $\beta$ differ only on $e$, $e$ is incident to $v$. ◀

Since $v$ is incident to $e$ and $e$ connects two vertices from $C$, $v \in C$. By Claim 14 the part of $\beta'$ that corresponds to the path from $t_{\theta(x_e)}$ to a sink does not substitute values to edges that are incident to $C$, and since $\beta'$ falsifies $v$, we get that $v$ is a leaf in $G_s$. But the value $\beta'(x_e)$ was chosen according to the assignment $\theta$ satisfying all vertices in $V_C \cap h(s) \ni v$ and thus $\beta'$ satisfies $v$ that leads to a contradiction. ◀

▶ Proposition 16. Let $D$ be a *minimum-size* 1-BP computing $\mathrm{SearchVertex}_{G,c}$, where $\mathrm{Ts}_{G,c}$ is unsatisfiable. Let $s$ be a node of $D$. Let $\alpha$ be a partial assignment from the set $P(s)$. Then each vertex $v$ from $h(s)$ is contained in an unsatisfiable component of $\mathrm{Ts}_{G_{s,\alpha},c_{s,\alpha}}$ which is contained in $h(s)$.

**Proof.** We prove the proposition by induction on the distance $d$ from $s$ to the farthest sink reachable from $s$.

Base case: $d = 0$, i.e. $s$ is a sink. $h(s)$ consists of the only vertex $v$, the label of the node $s$. $v$ is falsified by the assignment $\alpha$, then the component that consists of $v$ is unsatisfiable.

Induction step. Assume for the sake of contradiction that $v$ is a vertex of $h(s)$ contained in a connected component $C(V_C, E_C)$ of type (2) or (3) with respect to the node $s$ and the assignment $\alpha \in P(s)$. Let $t_0$ and $t_1$ be the direct successors of the node $s$. Notice that $h(s) = h(t_0) \cup h(t_1)$ by the definition of $h$, thus there exists $i \in \{0, 1\}$ such that $v \in h(t_i)$. Consider $\beta_i \in P(t_i)$ that extends $\alpha$.

Let $s$ be labeled with a variable $x_e$; by Proposition 13 the edge $e$ is contained in an unsatisfiable component of $\mathrm{Ts}_{G,c}|_\alpha$, thus $e$ is not contained in the component $C$. If $V_C \setminus h(t_i) \neq \varnothing$, then $v$ belongs to a connected component of type (3) with respect to the node $t_i$ and the assignment $\beta_i$. If $V_C \subseteq h(t_i)$, then, since $e$ is not contained in $C$, the connected component $C$ equals the corresponding connected component of $\mathrm{Ts}_{G,c}|_{\beta_i}$ contained in $h(t_i)$, moreover the charges of the vertices of $C$ are the same in the formulas $\mathrm{Ts}_{G,c}|_{\beta_i}$ and $\mathrm{Ts}_{G,c}|_\alpha$.

Thus in this case $v$ is contained in a component of type (2) with respect to the node $t_i$ and the assignment $\beta_i$.

But all vertices of $h(t_i)$ are contained in components of type (1) with respect to the node $t_i$ and the assignment $\beta_i$ by the induction hypothesis. This is a contradiction, hence all the vertices of $h(s)$ are contained in unsatisfiable components. ◀

**Proof of Lemma 9.** We prove by induction on the distance $d$ from the source to $s$ that $U(s)$ consists of a single connected component with the set of vertices $h(s)$. Moreover, for every $\alpha \in P(s)$, the single component from $U(s)$ is the only unsatisfiable component of $\mathrm{Ts}_{G,c}|_{\alpha}$.

Base case: $d = 0$, i.e. $s$ is the source of $D$. Since $G$ is connected, it consists of the only unsatisfiable component of $\mathrm{Ts}_{G,c}$. Lemma 2 implies that for every vertex $v \in V$ there exists a full assignment such that the parity condition of $v$ is violated, but the parity condition of any other vertex is satisfied. Therefore, $h(s) = V$.

Induction step. Let $\alpha$ be a partial assignment from $P(s)$. Let $r$ be the direct predecessor of $s$ according to the path corresponding to $\alpha$. Let $\beta \in P(r)$ agree with $\alpha$. By the induction hypothesis, $U(r)$ consists of a single connected component $C(V_C, H_C)$ of the formula $\mathrm{Ts}_{G,c}|_{\beta}$ with the vertex set $h(r)$. Let $x_e$ be the label of the node $r$. By Proposition 13 the edge $e$ is contained in $C$. We consider the following two cases.

If $e$ is not a bridge of $C$, then for the substitution $x_e \coloneqq \alpha(x_e)$ to $\mathrm{Ts}_{G,c}|_{\beta}$ the resulting formula has the only unsatisfiable component $C - e$. Lemma 2 implies that every vertex of $V_C$ can be the only vertex, where the parity condition of $\mathrm{Ts}_{G,c}|_{\alpha}$ is violated. Therefore, $h(s) = h(r) = V_C$.

Assume that $e$ is a bridge of $C$. Let $A, B$ be the connected components of $C - e$. The result of the substitution $x_e \coloneqq \alpha(x_e)$ to $\mathrm{Ts}_{G,c}|_{\beta}$ (which is $\mathrm{Ts}_{G,c}|_{\alpha}$) has the connected components $A$ and $B$ instead of $C$. Lemma 1 implies that exactly one of the components $A$ and $B$ is unsatisfiable. W.l.o.g. we assume that $A$ is unsatisfiable component of $\mathrm{Ts}_{G,c}|_{\alpha}$. By Lemma 2 every vertex of $A$ can be the only vertex with violated parity condition, thus $h(s)$ contains all vertices of $A$. $h(s)$ does not contain any vertex of $B$ since by Proposition 16 it can only contain vertices of unsatisfiable components. $A$ is the single unsatisfiable component of $\mathrm{Ts}_{G,c}|_{\alpha}$ since the substitution of any value to $x_e$ in $\mathrm{Ts}_{G,c}|_{\beta}$ does not affect any component except $C$. So, the induction step is proved.

Consider arbitrary node $s$ of $D$. We have already established that for every $\alpha \in P(s)$ a Tseitin formula $\mathrm{Ts}_{G_{s,\alpha}, c_{s,\alpha}}$ which is the result of substitution $\alpha$ to $\mathrm{Ts}_{G,c}$ has the only unsatisfiable component $H \in U(s)$ with the set of vertices $h(s)$. Moreover, by Proposition 10, $H$ and the restriction of $c_{s,\alpha}$ to the vertices of $H$ does not depend on the choice of $\alpha$. We denote by $f$ the restriction of $c_{s,\alpha}$ to $h(s)$.

Fix $\alpha \in P(s)$ and consider an arbitrary path from $s$ to a sink in $D$. Let $\mu$ be the partial assignment corresponding to this path. Let $\gamma$ be the union of $\mu$ and $\alpha$. Let $v = D(\gamma)$. Then $\gamma$ falsifies the parity condition in the vertex $v$ of $\mathrm{Ts}_{G,c}$. Thus $\mu$ falsifies the parity condition in the vertex $v$ of the formula $\mathrm{Ts}_{H,f}$ as well. Therefore the node $s$ of $D$ computes $\mathrm{SearchVertex}_{H,f}$. ◀

## 4.2 Proof of Theorem 8

We need the following technical proposition.

▶ Proposition 17. Let $G$ be a connected graph and let $H$ and $T$ be two connected subgraphs of $G$ with disjoint sets of vertices. Let $D_H$ be a 1-BP computing a satisfiable Tseitin formula $\mathrm{Ts}_{H,c_H}$ and let $D_T$ be a 1-BP computing a satisfiable Tseitin formula $\mathrm{Ts}_{T,c_T}$. Consider a branching program $D$ which is obtained by redirecting edges of $D_H$ going to 1-sink to the source of the program $D_T$ (and by merging two 1-sinks into a single 1-sink and two 0-sinks into a single 0-sink). Let $s$ be a node of $D_H$ such that in the program $D_H$ it computes the formula $\mathrm{Ts}_{H_s,c_s}$. Then in the new program $D$ the node $s$ computes a formula $\mathrm{Ts}_{K,f}$, where $K$ is the union of the graphs $H_s$ and $T$, and the charge function $f$ equals $c_s$ on the vertices of $H_s$ and equals $c_T$ on the vertices of $T$.

**Proof.** The proof is straightforward. The node $s$ of $D$ checks the parity conditions of $\mathrm{Ts}_{H_s,c_s}$ and then the parity conditions of $\mathrm{Ts}_{T,c_T}$. ◀

**Proof of Theorem 8.** Consider a minimum-size 1-BP $D$ computing $\mathrm{SearchVertex}_{G,c}$ and let $S$ be its size. Enumerate the nodes of $D$ in a reverse topological order $u_1, u_2, \dots, u_S$, i.e. such that every edge of $D$ is directed from a node with the greater number to a node with the less number.

By Lemma 9 every node of $D$ computes $\mathrm{SearchVertex}_{H,c'}$, where $H$ is a connected graph and the formula $\mathrm{Ts}_{H,c'}$ is unsatisfiable. We assume that $u_i$ computes the SearchVertex-relation for a graph $G_i(V_i, E_i)$ and a charge function $c_i$.

For $k$ from 0 to $S$ we iteratively construct a branching program $D^{(k)}$ such that:

- for every $i \in [k]$
- for every charge function $c'_i : V_i \to \{0,1\}$ that differs from $c_i$ for exactly one vertex of $G_i$

there exists a node of $D^{(k)}$ computing $\mathrm{Ts}_{G_i,c'_i}$. Moreover, each node of $D^{(k)}$ computes a Tseitin formula $\mathrm{Ts}_{H,f}$. If a node labeled with $x_e$ then $e$ is an edge of the graph $H$ and each of its successors computes $\mathrm{SearchVertex}_{H',f'}$, where $H'$ is a subgraph of $H - e$. This condition implies that $D^{(k)}$ is a 1-BP.

For $k = 0$, the program $D^{(0)}$ consisting of 0-sink and 1-sink is sufficient.

Assume that $D^{(k-1)}$ is constructed. We show how to add several nodes to $D^{(k-1)}$ such that the resulting diagram $D^{(k)}$ satisfies the conditions.

Consider several cases. Let $u_k$ be a sink labeled with a vertex $v$. Then the graph $G_k$ consists of the vertex $v$ and $c_k(v) = 1$. In that case we do not need to add any nodes to $D^{(k-1)}$ since the 1-sink satisfies the conditions for $u_k$.

Let $u_k$ be a non-sink node labeled with a variable $x_e$. Assume that the edge outgoing from $u_k$ labeled with 0 ends in the node $u_{k_0}$ and the edge outgoing from $u_k$ ends in the node $u_{k_1}$. For every vertex $j$ of the graph $G_k$ we will add to $D^{(k)}$ a node $s_j$ computing $\mathrm{Ts}_{G_k,c_k^{\oplus j}}$. (Recall that for a charge function $c : V \to \{0,1\}$ and a vertex $j \in V$ denote by $c^{\oplus j} : V \to \{0,1\}$ the charge function which differs from $c$ on the vertex $j$ and nowhere else.)

We consider two cases:

***e* is not a bridge of $G_k$.** In this case Lemma 9 implies that the graphs $G_{k_0}$ and $G_{k_1}$ are equal to $G_k - e$. Let $j$ be a vertex $G_k$, then it is a vertex of $G_{k_0}$ and $G_{k_1}$. By the induction hypothesis for $k_0$ and $k_1$ there exists such nodes $s_j^0$ and $s_j^1$ in $D^{(k-1)}$ such that $s_j^0$ computes $\mathrm{Ts}_{G_{k_0},c_{k_0}^{\oplus j}}$ and $s_j^1$ computes $\mathrm{Ts}_{G_{k_1},c_{k_1}^{\oplus j}}$. We add to $D^{(k)}$ a node $s_j$ and label it with $x_e$ and add an edge $(s_j, s_j^0)$ labeled with 0 and an edge $(s_j, s_j^1)$ labeled with 1.

Notice that by Lemma 9, $c_{k_0}$ equals $c_k$, and $c_{k_1}$ differs from $c_k$ only in the endpoints of $e$, thus the same statement is true for the charge functions $c_{k_0}, c_{k_1}$ and $c_k$ with flipped value at the vertex $j$. Therefore $s_j$ correctly computes $\mathrm{Ts}_{G_k,c_k^{\oplus j}}$.

***e* is a bridge of $G_k$.** Let $A$ be the connected component of the graph $G_k - e$ containing the vertex $j$ and let $B$ be the other connected component of $G_k - e$. Let $a \in A$, $b \in B$ be the endpoints of the edge $e$.

Since $\mathrm{Ts}_{G_k,c_k}$ is unsatisfiable, Lemma 1 implies that there exists the unique $\gamma \in \{0,1\}$ such that the formula $\mathrm{Ts}_{G_k,c_k}|_{x_e:=\gamma}$ contains the unique unsatisfiable component $A$ and $\mathrm{Ts}_{G_k,c_k}|_{x_e:=1-\gamma}$ contains the unique unsatisfiable component $B$. By Lemma 9, $G_{k_\gamma} = A$ and

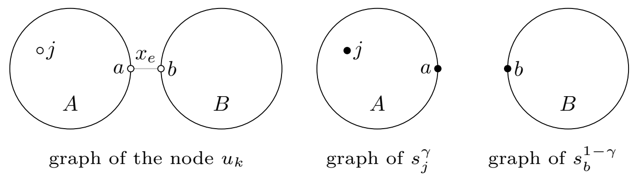


**Figure 2** The graphs of the nodes (for $\gamma = 1$); nodes with the charge different from $c_k$ are black, nodes with the same charge as in $c_k$ are white.

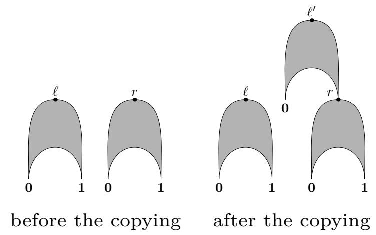


**Figure 3** Copying.

$G_{k_{1-\gamma}} = B$. By the induction hypothesis $D^{(k-1)}$ contains a vertex $s_j^{\gamma}$ computing $\mathrm{Ts}_{A, c_{k_\gamma}^{\oplus j}}$ and a vertex $s_b^{1-\gamma}$, computing $\mathrm{Ts}_{B, c_{k_{1-\gamma}}^{\oplus b}}$ (see Fig. 2).

If the number of vertices in the component $A$ is less or equal than the number of vertices in $B$, we denote $\ell = s_j^{\gamma}$ and $r = s_b^{1-\gamma}$, otherwise we denote $\ell = s_b^{1-\gamma}$ and $r = s_j^{\gamma}$. We copy the subprogram of $\ell$ (i.e. all successors of $\ell$ except the sinks) and add it to $D^{(k)}$. For every edge from the copied nodes to the 1-sink we redirect it to the node $r$. The edges to the 0-sink remain unchanged. We denote the source of the copied subprogram of $\ell$ by $\ell'$ (see Fig. 3).

By Proposition 17 the node $\ell'$ computes the Tseitin formula based on the union of graphs $A$ and $B$, the charges of the vertices of $A$ equal $c_{k_\gamma}^{\oplus j}$ and the charges of the vertices of $B$ equal $c_{k_{1-\gamma}}^{\oplus b}$. Similarly, by Proposition 17 each copied node computes a new Tseitin formula: the graph of the new formula can be obtained from the graph of the original one by addition of a new component which by the construction has at least as many vertices as the initial graph.

We add to the program $D^{(k)}$ a node $s_j$ labeled with $x_e$, the edge outgoing from $s_j$ labeled with $1-\gamma$ ends in the 0-sink, and the edge labeled with $\gamma$ ends in the node $\ell'$.

Since $A$ is an unsatisfiable component of $\mathrm{Ts}_{G_k, c_k}|_{x_e := 1-\gamma}$ and $j \in A$, the formula $\mathrm{Ts}_{G_k, c_k^{\oplus j}}|_{x_e := 1-\gamma}$ contains the usnatisfiable component $A$, hence it is unsatisfiable. $\mathrm{Ts}_{G_k, c_k}|_{x_e := \gamma}$ contains unsatisfiable component $A$ and satisfiable component $B$ with the charge $c_{k_\gamma}$. Hence $\mathrm{Ts}_{G_k, c_k^{\oplus j}}|_{x_e := \gamma}$ contains the satisfiable component $A$ with the charge $c_{k_\gamma}^{\oplus j}$ and the satisfiable component $B$ with the charge $c_{k_\gamma}$ that on the vertices of $B$ equals $c_{k_{1-\gamma}}^{\oplus b}$. Therefore, the node $\ell'$ computes $\mathrm{Ts}_{G_k, c_k^{\oplus j}}|_{x_e := \gamma}$, and hence, $s_j$ computes $\mathrm{Ts}_{G_k, c_k^{\oplus j}}$.

▶ Claim 18. The number of nodes in $D^{(S)}$ does not exceed $\log n (nS)^{\log n} = S^{O(\log n)}$, where $n$ is the number of vertices in the graph $G$.

**Proof.** Consider some node of the program $D^{(S)}$ that computes a Tseitin formula $\mathrm{Ts}_{H,f}$. If $H$ consists of $m$ connected components then they could be enumerated as $C_1(V_{C_1}, E_{C_1})$, $C_2(V_{C_2}, E_{C_2}), \ldots, C_m(V_{C_m}, E_{C_m})$ such that $|V_{C_i}| \geq |V_{C_1}| + \cdots + |V_{C_{i-1}}|$ for every $i \in [m]$. Since $H$ is a subgraph of $G$ we have $m \leq \log n$.

By the construction for every $i \in [m]$ there exists a node $u$ of the program $D$ such that $u$ computes $\mathrm{SearchVertex}_{C_i, h}$, where $h$ differs from $f$ in exactly one vertex from $V_{C_i}$. The number of different $m$ does not exceed $\log n$. The number of different pairs $(C_i, f)$ such that there exists a node $u$ of $D$, computing $\mathrm{SearchVertex}_{C_i, h}$, where $f$ and $h$ differs in exactly one vertex of $V_{C_i}$, does not exceed $Sn$. Therefore the number of nodes in $D^{(S)}$ does not exceed $\log n (Sn)^{\log n}$. ◀

◀

## References


1 Paul Beame, Toniann Pitassi, and Nathan Segerlind. Lower bounds for lovász-schrijver systems and beyond follow from multiparty communication complexity. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, pages 1176–1188, 2005. URL: `http://dx.doi.org/10.1007/11523468_95`, `doi:10.1007/11523468_95`.

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

3 Eli Ben-Sasson. Hard examples for bounded depth Frege. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 563–572, 2002. URL: `http://doi.acm.org/10.1145/509907.509988`, `doi:10.1145/509907.509988`.

4 Sam Buss, Dmitry Itsykson, Alexander Knop, and Dmitry Sokolov. Reordering rule makes OBDD proof systems stronger. Technical Report TR18-041, Electronic Colloquium on Computational Complexity (ECCC), 2018. To appear in Proceedings of CCC-2018. URL: `https://eccc.weizmann.ac.il/report/2018/041`.

5 Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes (abstract). In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity, Atlanta, Georgia, USA, May 4-6, 1999*, page 5, 1999. URL: `http://dx.doi.org/10.1109/CCC.1999.766254`, `doi:10.1109/CCC.1999.766254`.

6 Ludmila Glinskih and Dmitry Itsykson. Satisfiable Tseitin formulas are hard for nondeterministic read-once branching programs. In *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*, pages 26:1–26:12, 2017. URL: `https://doi.org/10.4230/LIPIcs.MFCS.2017.26`, `doi:10.4230/LIPIcs.MFCS.2017.26`.

7 Ludmila Glinskih and Dmitry Itsykson. On Tseitin formulas, read-once branching programs and treewidth. *Unpublished manuscript, available at https://logic.pdmi.ras.ru/~dmitrits/papers.php*, 2018.

8 Dima Grigoriev. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. *Theor. Comput. Sci.*, 259(1-2):613–622, 2001. URL: `https://doi.org/10.1016/S0304-3975(00)00157-2`, `doi:10.1016/S0304-3975(00)00157-2`.

9 Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

10 Johan Håstad. On small-depth Frege proofs for Tseitin for grids. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 97–108, 2017. URL: `https://doi.org/10.1109/FOCS.2017.18`, `doi:10.1109/FOCS.2017.18`.

11 D.M. Itsykson and A.A. Kojevnikov. Lower bounds of static Lovasz-Schrijver calculus proofs for Tseitin tautologies. *Zapiski Nauchnykh Seminarov POMI*, 340:10–32, 2006.

12 Stasys Jukna. A note on read-$k$ times branching programs. *ITA*, 29(1):75–83, 1995.

13 László Lovász, Moni Naor, Ilan Newman, and Avi Wigderson. Search Problems in the Decision Tree Model. *SIAM J. Discrete Math.*, 8(1):119–132, 1995. URL: `http://dx.doi.org/10.1137/S0895480192233867`, `doi:10.1137/S0895480192233867`.

14 Toniann Pitassi, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. Poly-logarithmic frege depth lower bounds via an expander switching lemma. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 644–657, 2016. URL: `http://doi.acm.org/10.1145/2897518.2897637`, `doi:10.1145/2897518.2897637`.

**15** G. S. Tseitin. On the complexity of derivation in the propositional calculus. *Zapiski nauchnykh seminarov LOMI*, 8:234–259, 1968. English translation of this volume: Consultants Bureau, N.Y., 1970, pp. 115–125.

**16** A. Urquhart. Hard examples for resolution. *JACM*, 34(1):209–219, 1987.

## A Appendix

**Proof of Lemma 4.** A replacement $x_e$ with $\neg x_e$ in a Tseitin formula corresponds to the flipping of the charges of the endpoints of the edge $e$. Since $G$ is connected and $\mathrm{Ts}_{G,c_1}$ and $\mathrm{Ts}_{G,c_2}$ are both satisfiable or both unsatisfiable, then by Lemma 1 the charge functions $c_1$ and $c_2$ have even number of differences. Let $v_1, v_2, \ldots, v_{2k}$ be the vertices where $c_1$ differs from $c_2$. Let $p_i$ be a simple path connecting $v_{2i-1}$ and $v_{2i}$ for $i \in [k]$. Let us modify $\mathrm{Ts}_{G,c_1}$ in the following way: for each of the paths $p_1, \ldots, p_k$ we replace the variables corresponding to the edges of a path with their negations (if several paths pass through an edge $e$ we will replace $x_e$ with its negation as many times as the number of paths that pass through $e$). The resulting formula is $\mathrm{Ts}_{G,c_2}$ since charges of the ends of the paths (i.e. in the vertices $v_1, \ldots, v_{2k}$) have been changed and charges of all other vertices have not been changed. ◀