Tight Bounds for Tseitin Formulas

Dmitry Itsykson
St. Petersburg Department of V.A. Steklov Mathematical Institute of RAS, Russia

Artur Riazanov
St. Petersburg Department of V.A. Steklov Mathematical Institute of RAS, Russia
The Henry and Marylin Taub Faculty of Computer Science, Technion, Israel

Petr Smirnov
HSE University at Saint Petersburg, Russia
St. Petersburg Department of V.A. Steklov Mathematical Institute of RAS, Russia

Abstract

We show that for any connected graph G the size of any regular resolution or OBDD($\wedge$, reordering) refutation of a Tseitin formula based on G is at least $2^{\Omega(\mathrm{tw}(G))}$, where $\mathrm{tw}(G)$ is the treewidth of G. These lower bounds improve upon the previously known bounds and, moreover, they are tight.

For both of the proof systems, there are constructive upper bounds that almost match the obtained lower bounds, hence the class of Tseitin formulas is almost automatable for regular resolution and for OBDD($\wedge$, reordering).

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases Proof complexity, Tseitin formulas, treewidth, resolution, OBDD-based proof systems

Digital Object Identifier 10.4230/LIPIcs.SAT.2022.6

Funding *Artur Riazanov*: The author is supported under the agreement 075-15-2022-289 and has also received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 802020-ERC-HARMONIC.
Petr Smirnov: The author is supported under the agreement 075-15-2022-289.

1 Introduction

The development of solvers for the Boolean satisfiability problem is tightly connected with the study of propositional proof systems. Every SAT solver corresponds to a proof system. Roughly speaking, the execution log of every SAT solver running on an unsatisfiable formula φ may serve as a certificate of unsatisfiability of φ. The size of the shortest proof of a formula φ is a lower bound on the running time of a solver executed on φ. On the one hand, we need that an underlying proof system is sufficiently strong to have short refutations of important formulas, on the other hand, the underlying proof system should be sufficiently weak such that short proofs can be efficiently found.

A propositional proof system Π is automatable (quasi-automatable) on a class of unsatisfiable formulas $\mathcal{F}$ [1] if there is an algorithm that finds a refutation of any formula $\varphi \in \mathcal{F}$ in time $\mathrm{poly}(|\varphi|, S) = 2^{\mathcal{O}(\log|\varphi| + \log S)}$ (and $2^{\mathrm{poly}(\log|\varphi|, \log S)}$ in quasi-automatable case), where S is the size of the shortest Π-refutation of φ. The series of results about the hardness of automatability [2, 14, 19, 16] roughly speaking means that it is unlikely that some of the commonly used proof systems are automatable or quasi-automatable on the class of all unsatisfiable CNF formulas. However, it is possible that a proof system is automatable on important formula classes.

In this paper, we consider the class of Tseitin formulas [30] encoding in CNF the following parity principle: any graph has an even number of vertices with an odd degree. For an undirected graph $G = (V, E)$ and a charge function $c \colon V \to \{0, 1\}$ let a Tseitin formula

$\mathrm{T}(G, c)$ be defined as follows. The variables of $\mathrm{T}(G, c)$ correspond to the edges of the graph. The formula itself is the conjunction of *the parity conditions* of the vertices of G stating that the sum of edges incident to v equals $c(v)$ modulo 2. It is known that a Tseitin formula is satisfiable if and only if for every its connected component the sum of charges of its vertices is even [31]. Unsatisfiable Tseitin formulas based on special families of graphs (expanders and grids) are hard for many proof systems [31, 3, 21, 10].

In order to show that a proof system Π is automatable on the class of Tseitin formulas, first, we have to devise an algorithm that produces refutations of Tseitin formulas. Second, we need to bound from below the refutation size of Tseitin formulas such that this bound is close to the running time of the proof search algorithm. We emphasize that the lower bounds must hold for all graphs, this is the main difficulty of the second step.

The first lower bounds on the refutation size of Tseitin formulas for all graphs used the improved Grid Minor Theorem [8]. It states that any graph G has a grid graph of size $t \times t$ as a minor, where $t = \Omega(\mathrm{tw}(G)^\lambda)$, $\mathrm{tw}(G)$ is the treewidth of G and $\lambda \geq \frac{1}{10}$ is a constant; it is known, however, that the theorem is false for $\lambda \geq \frac{1}{2}$. The strategy of the lower bound proofs is to first show lower bounds for the grid graphs, and then extend them to all graphs using the Grid Minor Theorem. Using this method, Glinskih and Itsykson [18] proved a lower bound $2^{\Omega\left(\mathrm{tw}(G)^\lambda\right)}$ on the size of OBDD($\wedge$, reordering) refutations of $\mathrm{T}(G, c)$; they also give a non-matching upper bound $\mathcal{O}(2^{\mathrm{pw}(G)}|E|)$. Galesi et al. [15] proved a lower bound $2^{\mathrm{tw}(G)^{\Omega(1/d)}}$ on the size of depth-d Frege refutations of $\mathrm{T}(G, c)$ using Håstad's lower bound for the grids [21] and the Grid Minor Theorem. This very general approach yields bounds that are very far from being optimal, but such results motivate searching for more precise lower bounds.

Itsykson et. al. [23] proved a lower bound $2^{\Omega(\mathrm{tw}(G)/\log|V|)}$ on the size of regular resolution refutations of $\mathrm{T}(G, c)$ for any connected graph G. The heart of this proof is the reduction from satisfiable Tseitin formulas. Namely, if there exists a regular resolution refutation of a Tseitin formula $\mathrm{T}(G, c)$ of size S, then it can be transformed into a read-once branching program (1-BP) computing a satisfiable formula $\mathrm{T}(G, c')$ of size $S^{\mathcal{O}(\log|V|)}$. It is shown in [23] that the size of the minimal 1-BP computing a satisfiable Tseitin formula $\mathrm{T}(G, c')$ is at least $2^{\Omega(\mathrm{tw}(G))}$.

De Colnet and Mengel considered a computational model that is stronger than 1-BP: DNNF (decomposable negation normal form) is a special kind of Boolean circuit in the basis $\{\wedge, \vee, \neg\}$, where negations are applied only to variables and for every $\wedge$-gate variables from two subcircuits of its children do not intersect. De Colnet and Mengel [12] have proved a lower bound $2^{\Omega(\mathrm{tw}(G)/\Delta(G))}$ on the size of DNNF computing satisfiable Tseitin formula $\mathrm{T}(G, c')$, where $\Delta(G)$ denotes the maximum degree of G. Similarly to [23] a regular resolution refutation of a Tseitin formula $\mathrm{T}(G, c')$ of size S can be transformed into a DNNF of size $\mathcal{O}(S|V|)$ computing a satisfiable Tseitin formula. This implies a lower bound $2^{\Omega(\mathrm{tw}(G)/\Delta(G))}/|V|$ on the size of regular resolution refutations of $\mathrm{T}(G, c)$ for a connected graph G. For constant-degree graphs this bound is tight up to a constant factor in the exponent, but for graphs with $\Delta(G) = \omega(\log|V|)$ the bound from [23] is stronger.

Our results. In this paper, we study the complexity of refutations of Tseitin formulas in two proof systems: OBDD($\wedge$, reordering) ([22]) and regular resolution. These two proof systems are very different; it is known that they do not simulate each other [7, 6]. However, the known proofs of lower bounds on Tseitin formulas use similar techniques since in both cases they are based on the complexity of satisfiable Tseitin formulas. Our results imply that the minimal refutation sizes of Tseitin formulas in these proof systems are very close, however, the only known example of a formula that requires resolution refutations of size superpolynomially larger than the size of the shortest OBDD($\wedge$) refutation is a Tseitin formula on the complete graph [7].

We prove a lower bound $2^{\Omega(\mathrm{tw}(G))}$ on the size of OBDD($\wedge$, reordering) refutation of $\mathrm{T}(G, c)$. This lower bound matches the upper bound $\mathcal{O}(2^{\mathrm{pw}(G)}|E|)$ [18] for a large family of graphs with $\mathrm{pw}(G) = \Theta(\mathrm{tw}(G))$ (this family includes grids, complete graphs, expanders, etc.). Our approach is rethinking of the techniques from [22, 18] supplied with new ideas.

We prove a lower bound $2^{\Omega(\mathrm{tw}(G))}$ on the complexity of DNNF computing satisfiable Tseitin formula $\mathrm{T}(G, c')$, thereby improving the result from [12]. Our proof is highly based on [12], we surgically remove the $\frac{1}{\Delta(G)}$ factor from the exponent. We also prove a matching upper bound $2^{\mathcal{O}(\mathrm{tw}(G))}$. As a corollary of the lower bound using a reduction from [12], we get a stronger lower bound $2^{\Omega(\mathrm{tw}(G))}$ on the size of regular resolution refutations of Tseitin formulas that improve both the lower bounds from [23] and [12]. For regular resolution there is also a known upper bound $\mathrm{poly}(\mathrm{T}(G, c))2^{\mathcal{O}(\mathrm{tw}(L(G)))}$, where $L(G)$ is the line graph of G [1]. There is a family of graphs $G_{n,k}$ on $\Theta(n^2k^2)$ vertices with $\mathrm{tw}(L(G_{n,k})) = 4n + \mathcal{O}(k^3)$, $\mathrm{tw}(G_{n,k}) \geq n$ and $\Delta(G_{n,k}) = k$ [20, Section 7], thus for $k < n^{1/3}$ our lower bound is tight. Our upper bound on the size of DNNF implies that the method from [12] can not give a better bound (for example, we can not prove $2^{\Omega(\mathrm{tw}(L(G)))}$ using this method).

Almost automatability. We say that a propositional proof system Π is *almost automatable* on a class of formulas $\mathcal{F}$ if there exists an algorithm A such that for any $\varphi \in \mathcal{F}$, $A(\varphi)$ produces a Π-refutation of φ in time $S^{\mathcal{O}(\log|\varphi|)} = 2^{\mathcal{O}(\log|\varphi| \cdot \log S)}$, where S is the size of the shortest Π-refutation of φ. Notice that if $\varphi = \mathrm{T}(G, c)$, then $\Omega\left(|V| + 2^{\Delta(G)}\right) \leq |\varphi| \leq \mathcal{O}(|V|2^{\Delta(G)})$, hence $\log(|\varphi|) = \Theta(\log|V| + \Delta(G))$.

Our results imply that regular resolution and OBDD($\wedge$, reordering) are almost automatable on the class of Tseitin formulas.

- Alekhnovich and Razborov [1] developed an algorithm (Branch-Width Based Automated Theorem Prover or BWBATP) searching for regular resolution refutations of CNF formulas. BWBATP finds a regular resolution refutation of a Tseitin formula $\mathrm{T}(G, c)$ in $2^{\mathcal{O}(\mathrm{tw}(L(G))}\mathrm{poly}(|\mathrm{T}(G, c)|)$ steps. Since $\mathrm{tw}(L(G)) = \mathcal{O}(\mathrm{tw}(G)\Delta(G))$, $2^{\mathcal{O}(\mathrm{tw}(L(G))}\mathrm{poly}(|\mathrm{T}(G, c)|) = 2^{\mathcal{O}(\mathrm{tw}(G)\log|\mathrm{T}(G,c)|)}$.
- We show in Section 3.2 that an OBDD($\wedge$) refutation of a Tseitin formula $\mathrm{T}(G, c)$ can be constructed in $2^{\mathcal{O}(\mathrm{tw}(G)\log|V|)}\mathrm{poly}(\mathrm{T}(G, c)) = 2^{\mathcal{O}(\mathrm{tw}(G)\log|\mathrm{T}(G,c)|)}$ steps.

New preprint of de Colnet and Mengel. While preparing this paper we became aware of the new preprint of de Colnet and Mengel [13]. Results of that paper imply a lower bound $2^{\Omega(\mathrm{tw}(G)/\Delta^3(G))}$ on the size of OBDD($\wedge$, reordering) refutations of $\mathrm{T}(G, c)$. In fact, in [13] the authors deal with a slightly stronger model, where instead of OBDD they use structural DNNF. This model is not a propositional proof system since it is NP-hard to verify such proofs [26]. However, our proof works for their model without any changes, and using our DNNF lower bound we get a lower bound $2^{\Omega(\mathrm{tw}(G))}$ on the model used in [13], that is better for graphs with non-constant degrees (see Remark 3.7 for details). Our proof and the proof from [13] use different strategies, while ours seems to be much simpler.

2 Preliminaries

Boolean functions and formulas. Let X be a set of propositional variables. A *partial assignment* is a set of elementary assignments $x := a$, where $x \in X$ and $a \in \{0, 1\}$ such that every variable appears in at most one elementary assignment. The *support* of a partial assignment ρ is the set of variables on the left-hand side of the elementary assignments.

Let σ and τ be partial assignments. We write $\sigma \subseteq \tau$, if the support of σ is a subset of the support of τ, and they agree on the support of σ. If the supports of σ and τ do not intersect, we denote by $\sigma \cup \tau$ a partial assignment that coincides with σ on the support of σ and with τ on the support of τ.

For a Boolean function $f : \{0,1\}^n \to \{0,1\}$ we use a notation $\mathrm{sat}(f) = \{x \in \{0,1\}^n \mid f(x) = 1\}$.

We identify a CNF formula φ with the set of its clauses. For example, if φ and ψ are CNF formulas, then $\varphi = \psi$ means that their sets of clauses are equal, and $\varphi \subseteq \psi$ means that every clause of φ occurs in ψ.

OBDD. An *ordered binary decision diagram* (OBDD) is used to represent a Boolean function [5]. Let $X = \{x_1, \ldots, x_n\}$ be a set of propositional variables. A binary decision diagram (BDD) is a directed acyclic graph with one source. Each node of the graph is labeled by a variable from X or by a constant 0 or 1. If a node is labeled by a constant, then it is a sink (has out-degree 0). If a node is labeled by a variable, then it has exactly two outgoing edges: one edge is labeled by 0 and the other edge is labeled by 1. Every binary decision diagram defines a Boolean function $\{0,1\}^n \to \{0,1\}$. The value of the function for given values of $x_1, \ldots, x_n$ is computed as follows: we start a path at the source and at every step follow the edge that corresponds to the value of the variable labeling the current node. Every such path reaches a sink, which is labeled either 0 or 1: this constant is the value of the function.

Let π be a permutation of the set $[n] = \{1, \ldots, n\}$. A π-ordered binary decision diagram ($\pi-$OBDD) is a binary decision diagram such that on every path from the source to a sink every variable has at most one occurrence and the variable $x_{\pi(i)}$ can not appear before $x_{\pi(j)}$ if $i > j$. An ordered binary decision diagram (OBDD) is a π-ordered binary decision diagram for some permutation π. By convention, every OBDD is associated with a single fixed permutation π. This π puts a total order on all the variables, even if the OBDD does not query all variables.

The *size* of an OBDD is the number of nodes in it.

▶ **Lemma 2.1** ([32, Theorem 3.3.1]). *Let D be a π-OBDD, and ρ be a partial assignment to variables of D. Then there is a π-OBDD computing $D|_\rho$ of size at most $|D|$.*

▶ **Lemma 2.2** ([32, Theorem 3.3.6]). *Let D_1 and D_2 be π-OBDDs over the same set of variables. Then there is a π-OBDD of size $\mathcal{O}(|D_1||D_2|)$ computing $D_1 \wedge D_2$.*

DNNF. A Boolean circuit in the negation normal form (NNF) is a circuit in the de Morgan basis $\{\wedge, \vee, \neg\}$ with binary conjunctions and disjunctions, where all negations only apply to variables. For a gate g of an NNF Boolean circuit, we define $\mathrm{var}(g)$ as a set of variables in its subcircuit.

Let g be a gate of an NNF with direct predecessors g_l and g_r. The gate g is *decomposable* if $\mathrm{var}(g_l) \cap \mathrm{var}(g_r) = \emptyset$. The gate g is *complete* if $\mathrm{var}(g_l) = \mathrm{var}(g_r)$. An NNF is called *decomposable* (DNNF) if each $\wedge$-gate is decomposable. An NNF is *complete* if each $\vee$-gate is complete.

▶ **Lemma 2.3** ([11]). *Let D be a DNNF. Then there is a complete DNNF computing the same function as D and it can be constructed in time* $\mathrm{poly}(|D|)$.

Resolution. A *resolution refutation* of an unsatisfiable CNF formula φ is a sequence of clauses $C_1, C_2, \dots, C_s$ such that C_s is the empty clause (identically false), and for all $i \in [s]$, the clause C_i is either a clause of φ, or can be obtained by the resolution rule from two preceding clauses, where the resolution rule allows to derive $A \vee B$ from $A \vee x$ and $B \vee \neg x$.

A resolution refutation is *regular* if for every increasing sequence $1 \le i_1 < i_2 < \dots < i_k \le s$ such that for all $j \in \{2, \dots, k\}$ the clause C_{i_j} is obtained by the resolution rule applied to $C_{i_{j-1}}$ as one of the premises (let x_j denote the resolved variable), all variables x_j for $j \in \{2, \dots, k\}$ are distinct.

The number s is the *size* of the resolution refutation.

OBDD-based proof systems. An OBDD($\wedge$, reordering) *refutation* of an unsatisfiable CNF formula φ is a sequence of OBDDs $D_1, D_2, \dots, D_s$ such that D_s computes the identically false function, and for all $i \in [s]$, the OBDD D_i is either computes a clause of φ, or can be obtained from previous OBDDs by the following inference rules:

- Conjunction: If $j, k < i$ and D_j and D_k are π-OBDDs for some order π, then we can infer OBDD $D_i = D_j \wedge D_k$ with the same order π.
- Reordering: If $j < i$, then we can infer OBDD D_i computing the same function as D_j, and variable orders of D_i and D_j can be different.

The *length* of the refutation is s and its *size* is the sum of the sizes of OBDDs in it, i.e., $|D_1| + |D_2| + \ldots + |D_s|$.

OBDD($\wedge$) refutation is an OBDD($\wedge$, reordering) refutation that does not contain the reordering rule. A refutation is tree-like if every OBDD in it is used at most once as a premise of a rule.

▶ **Lemma 2.4** ([18, Lemma 5.4]). *Let φ be an unsatisfiable CNF formula that has an* OBDD($\wedge$, reordering) *refutation of size S. Let ρ be a partial assignment of values of the formula φ. Then $\varphi|_\rho$ has an* OBDD($\wedge$, reordering) *refutation of size at most S.*

Graph basics. Throughout the paper, we consider undirected graphs possibly with self-loops and parallel edges. We use $G = (V, E)$ to denote a graph G with a vertex set V and an edge set E. An undirected edge $e \in E$ incident to vertices $v \in V$ and $u \in V$, we denote by (v, u) or (u, v). For a vertex v, we denote the set of edges incident to v by $E(v)$: $E(v) = \{(v, u) \in E\}$.

By $\Delta(G)$ we denote the maximum degree of a graph G, and by $\#G$ we denote the number of connected components in G.

For $V' \subseteq V$ we denote by $G[V']$ the subgraph of G, induced by vertices of V'. In particular, if $V' = \{v_1, v_2, \dots, v_k\}$, we write $G[v_1, v_2, \dots, v_k]$ meaning the same graph $G[V']$. We denote by $G \setminus V'$ the graph $G[V \setminus V']$. For $E' \subseteq E$, we denote by $G \setminus E'$ the graph $G' = (V, E \setminus E')$.

A graph $G = (V, E)$ is k-connected, if it has more than k vertices, and for each vertex subset $S \subseteq V$ of size at most k, the graph $G \setminus S$ is connected.

For any graph H, we denote by $V(H)$ the set of its vertices and by $E(H)$ and the set of its edges.

Graph minors. For $e = (v, u) \in E$, we denote by G/e the graph obtained from G by *contraction* of edge e: we delete the edge e and merge v and u into one vertex. A graph G' is a minor of graph G, if G' can be obtained from G by vertex deletions, edge deletions and edge contractions.

Let a vertex $v \in V$ of degree two has two different neighbors u and w. *Suppression* of v is an operation on graph G, in which we delete the vertex v from G and add an edge (u, w). G' is called a *topological minor* of graph G, if G' can be obtained from G by vertex deletions, edge deletions and vertex suppressions.

Graph decompositions. A *tree decomposition* of an undirected graph $G = (V, E)$ is a tree $T = (V_T, E_T)$ and a family $\{X_t\}_{t \in V_T}$ of subsets of V such that the following properties hold:
1. The union of X_t for $t \in V_T$ equals V.
2. For every edge $(v, u) \in E$ there exists $t \in V_T$ such that $v, u \in X_t$.
3. If a vertex $v \in V$ is contained in the sets X_t and X_s for some $t, s \in V_T$, then it is also contained in X_p for all vertices p on the unique path between s and t in T.

The sets X_u are called *bags* of the tree decomposition. The *size* of tree decomposition is the number of nodes in T. The *width* of a tree decomposition, denoted by $w(T)$, is the maximum bag size $|X_u|$ for $u \in V_T$ minus one. The *treewidth* of a graph G, denoted by $\mathrm{tw}(G)$, is the minimum width among all tree decompositions of the graph G.

A *path decomposition* of a graph G is a tree decomposition of G such that the underlying tree T is a simple path. The *pathwidth* of a graph G, denoted by $\mathrm{pw}(G)$, is the minimum width among all path decompositions of the graph G.

▶ **Lemma 2.5** (Folklore, see, e.g., [24, Theorem 6]). *For any graph G, $\mathrm{pw}(G) \leq \mathcal{O}(\mathrm{tw}(G) \log |V(G)|)$.*

Moreover, given a tree decomposition T of width k, one can construct a path decomposition of width $\mathcal{O}(k \log |V(G)|)$ in time $\mathrm{poly}(|V(G)|, |E(G)|, |V(T)|)$.

▶ **Lemma 2.6** (Folklore). *Let G be a graph, $A \subseteq V(G)$. Then $\mathrm{tw}(G \setminus A) \geq \mathrm{tw}(G) - |A|$.*

▶ **Lemma 2.7** ([27, Proposition 2.7]). *Let G be a graph and G' be its minor. Then $\mathrm{tw}(G') \leq \mathrm{tw}(G)$.*

▶ **Lemma 2.8** ([9, Chapter 7]). *Let G be a graph and H be obtained from G by vertex suppressions. Then $\mathrm{tw}(H) \geq \mathrm{tw}(G) - 1$.*

▶ **Theorem 2.9** ([4]). *Given a graph G, one can obtain a tree decomposition of width $\mathcal{O}(\mathrm{tw}(G))$ in time $2^{\mathcal{O}(\mathrm{tw}(G))}|V(G)|$.*

▶ **Theorem 2.10** ([12, Lemma 25]). *Let G be a graph with a treewidth of at least 3. Then G has a 3-connected topological minor H with $\mathrm{tw}(H) = \mathrm{tw}(G)$.*

A *branch decomposition* of an undirected graph $G = (V, E)$ is a tree $T = (V_T, E_T)$, each non-leaf node has degree three, and leaves are in bijection with the edges of G. Each edge e of E_T gives an e-separation of the set E into two non-empty parts E_1 and E_2: deleting the edge e from T, we get two trees T_1 and T_2; let E_1 be edges that occur in leaves of T_1, E_2 be edges that occur in leaves of T_2. The width of e-separation is the number of vertices of G incident to edges from both E_1 and E_2. The *width* of branch decomposition T is the maximum width of e-separation over all $e \in E_T$. The *branchwidth* of G, denoted by $\mathrm{bw}(G)$, is the minimum width among all branch decompositions of G.

▶ **Theorem 2.11** ([28, Theorem 5.1]). *For any graph G, $\max(\mathrm{bw}(G), 2) \leq \mathrm{tw}(G) + 1 \leq \max(\lceil \frac{3}{2} \mathrm{bw}(G) \rceil, 2)$.*

Tseitin formulas. Let $G = (V, E)$ be a graph. Let $c\colon V \to \{0, 1\}$ be a *charge function*. A *Tseitin formula* $\mathrm{T}(G, c)$ depends on the propositional variables x_e for $e \in E$. For each vertex $v \in V$ we define the parity condition of v as $P_v \coloneqq \left(\sum_{e\in E(v)} x_e \equiv c(v) \bmod 2\right)$. The Tseitin formula $\mathrm{T}(G, c)$ is the conjunction of parity conditions of all the vertices: $\bigwedge_{v\in V} P_v$. Tseitin formulas are represented in CNF as follows: we represent P_v in CNF in the canonical way for all $v \in V$.

When we write about substitutions to a Tseitin formula, we often identify variables and edges that correspond to them.

Assume that G consists of connected components $H_1, H_2, \dots, H_t$. Then the Tseitin formula $\mathrm{T}(G, c)$ is equivalent to the conjunction $\bigwedge_{i=1}^{t} \mathrm{T}(H_i, c)$. In the last formula we abuse the notation since c is defined not only on the vertices of H_i and, thus, we implicitly use the corresponding restriction on the set of vertices.

Let $c_1 \colon V_1 \to \{0, 1\}, c_2 \colon V_2 \to \{0, 1\}$ be charge functions. We denote by $c_1 + c_2$ the charge function $(c_1 + c_2)\colon V_1 \cup V_2 \to \{0, 1\}$ such that

$$(c_1 + c_2)(v) = \begin{cases} c_1(v), & \text{if } v \in V_1 \setminus V_2; \\ c_2(v), & \text{if } v \in V_2 \setminus V_1; \\ c_1(v) + c_2(v) \bmod 2, & \text{if } v \in V_1 \cap V_2. \end{cases}$$

If V_1 and V_2 do not intersect, we can also write $c_1 \sqcup c_2$ meaning the same charge function $c_1 + c_2$. By $\mathbf{1}_v$, we denote the charge function $\mathbf{1}_v \colon \{v\} \to \{0, 1\}$ such that $\mathbf{1}_v(v) = 1$.

▶ **Lemma 2.12** (Folklore, see, e.g., [31]). *A Tseitin formula $\mathrm{T}(G, c)$ is satisfiable if and only if for every connected component $C(U, E_U)$ of the graph G, the condition $\sum_{u\in U} c(u) \equiv 0 \bmod 2$ holds.*

Note that if a connected component C consists of an isolated vertex v, then either $c(v) = 1$ and $\mathrm{T}(G, c)$ is unsatisfiable, or $c(v) = 0$ and $\mathrm{T}(G, c) = \mathrm{T}(G \setminus v, c)$. In other words, adding zero-charged isolated vertices does not change a Tseitin formula.

▶ **Lemma 2.13** (Folklore). *Let $G = (V, E)$ be a graph, $\mathrm{T}(G, c)$ be satisfiable, σ be a full assignment for the set of variables of $\mathrm{T}(G, c)$. Then the number of parity conditions falsified by σ is even.*

Proof. See Appendix A for the proof. ◀

▶ **Lemma 2.14** (Folklore). *The result of the substitution $x_e \coloneqq b$ to $\mathrm{T}(G, c)$ where $b \in \{0, 1\}$ is a Tseitin formula $\mathrm{T}(G', c')$ 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.*

▶ **Lemma 2.15** (Folklore, see, e.g., [23, Lemma 2.3]). *Let $G = (V, E)$ be a connected graph and let $c_1, c_2 \colon V \to \{0, 1\}$ be charge functions. If Tseitin formulas $\mathrm{T}(G, c_1)$ and $\mathrm{T}(G, c_2)$ are both satisfiable or both unsatisfiable, then one of them can be obtained from another by replacing some variables with their negations.*

▶ **Lemma 2.16** ([17, Lemma 2]). *If a Tseitin formula $\mathrm{T}(G, c)$ is satisfiable, then it has $2^{|E|-|V|+\#G}$ satisfying assignments.*

▶ **Lemma 2.17** ([18, Lemma 5.5]). *Let $G = (V, E)$ be a connected graph and $G' = (V', E')$ be a connected subgraph of G with $E' \neq \emptyset$ that is obtained from G by the deletion of some vertices and edges. For every unsatisfiable Tseitin formula $\mathrm{T}(G, c)$ there exists an assignment ρ on variables $E \setminus E'$, such that ρ does not falsify any clause of $\mathrm{T}(G, c)$.*

▶ **Lemma 2.18** (Folklore, see, e.g., [18, Lemma 5.2]). *Let G be a 2-connected graph and $\mathrm{T}(G,c)$ be unsatisfiable. Then $\mathrm{T}(G,c)$ is a minimally unsatisfiable formula, i.e. removing any of its clauses makes it satisfiable.*

▶ **Lemma 2.19** ([12, Lemma 24]). *Let H be a topological minor of G. If a satisfiable Tseitin formula $\mathrm{T}(G,c)$ has a DNNF of size s, then any satisfiable formula $\mathrm{T}(H,c')$ also has a DNNF of size s.*

▶ **Theorem 2.20** ([23, Theorem 1.9]). *Let $\mathrm{T}(G,c)$ be a satisfiable Tseitin formula. Then the minimum size of* OBDD *computing $\mathrm{T}(G,c)$ is at least $2^{\Omega(\mathrm{tw}(G))}$.*

3 OBDD($\wedge$, reordering)

3.1 Lower Bound

In this section, we prove the following theorem.

▶ **Theorem 3.1.** *Let G be a connected graph and $\mathrm{T}(G,c)$ be an unsatisfiable Tseitin formula. Then any* OBDD($\wedge$, reordering) *refutation of $\mathrm{T}(G,c)$ has a size of at least $2^{\Omega(\mathrm{tw}(G))}$.*

We say that a graph G *is a subdivision* of H if H can be obtained from G by several suppression operations. We say that a graph G is *almost 3-connected* if it is a subdivision of a 3-connected graph H.

First, we prove Theorem 3.1 only for almost 3-connected graphs.

▶ **Theorem 3.2.** *Let G be an almost 3-connected graph and $\mathrm{T}(G,c)$ be an unsatisfiable Tseitin formula. Then any* OBDD($\wedge$, reordering) *refutation of $\mathrm{T}(G,c)$ has a size of at least $2^{\Omega(\mathrm{tw}(G))}$.*

Let us show how Theorem 3.2 implies Theorem 3.1.

Proof of Theorem 3.1. Let S be the minimal size of OBDD($\wedge$, reordering) refutation of $\mathrm{T}(G,c)$.

If $\mathrm{tw}(G) \le 2$, then the theorem is trivial. Otherwise, by Theorem 2.10, there exists a 3-connected topological minor H of G such that $\mathrm{tw}(G) = \mathrm{tw}(H)$. Consider a sequence of operations that transforms G to H, where all edge and vertex deletions precede suppressions. Let us denote by G' the graph obtained from G after the application of all edge and vertex deletions. By Lemma 2.17 there exists a partial assignment ρ with support corresponding to the edges that are in G but not in G' such that ρ does not falsify any clause of $\mathrm{T}(G,c)$. It is easy to see that $\mathrm{T}(G,c)|_\rho$ coincides with $\mathrm{T}(G',c')$ for some c' and $\mathrm{T}(G',c')$ is unsatisfiable. By Lemma 2.4, there exists OBDD($\wedge$, reordering) refutation of $\mathrm{T}(G',c')$ of size at most S.

Since H is obtained from G' by several suppressions, G' is almost 3-connected. Hence, by Theorem 3.2, $S \ge 2^{\Omega(\mathrm{tw}(G'))}$. Since H is a minor of G' and G' is a minor of G, by Lemma 2.7 we have $\mathrm{tw}(G) \ge \mathrm{tw}(G') \ge \mathrm{tw}(H) = \mathrm{tw}(G)$. Thus, $\mathrm{tw}(G') = \mathrm{tw}(G)$ and, hence, $S \ge 2^{\Omega(\mathrm{tw}(G))}$. ◀

In order to complete the proof of Theorem 3.1, we have to prove Theorem 3.2.

Let G be a subdivision of H. Notice that for every vertex u of H there exists a vertex u' from G that is transformed to u. We call such vertices u' *main vertices*; we call all other vertices of G (that are not main) *interior vertices*. If vertices u and v are adjacent in H, then corresponding main vertices u' and v' are connected by a path in G (possibly of length 1) that is entirely transformed to the edge (u,v). We call such paths in G as *long edges*. Notice that all endpoints of long edges are main vertices, and all other vertices of long edges are interior.

For every subgraph H' of H, we define a corresponding subgraph G' of G in the following way. The set of vertices of G' are 1) main vertices of G that correspond to the vertices of H', and 2) interior vertices of long edges corresponding to the edges of H'. The set of edges of G' consists of all edges from long edges corresponding to edges of H'. It is easy to see that G' is a subdivision of H'.

It is easy to see that every almost 3-connected graph is 2-connected. However, the stronger statement holds.

▶ **Lemma 3.3.** *Let G be an almost 3-connected graph. Let u and v be two vertices that do not belong to the same long edge. Then the graph $G \setminus \{v, u\}$ is connected.*

Proof. See Appendix B for the proof. ◀

Proof of Theorem 3.2. If $\Delta(G) > \mathrm{tw}(G)/10$, then formula $\mathrm{T}(G, c)$ has at least $2^{\mathrm{tw}(G)/10}$ clauses and since G is 2-connected, by Lemma 2.18 all these clauses should be used in a refutation. Hence, any OBDD($\wedge$, reordering) refutation of $\mathrm{T}(G, c)$ has a size of at least $2^{\mathrm{tw}(G)/10}$. So we can assume that $\Delta(G) \leq \mathrm{tw}(G)/10$.

Let G is a subdivision of a 3-connected graph H.

Consider an OBDD($\wedge$, reordering) refutation of $\mathrm{T}(G, c)$ that has the minimal possible size. The last line in the refutation is an identically false OBDD. If this line represents a clause of the initial formula, then G has an isolated vertex and since G is connected, it consists of one vertex, in this case the statement is trivial. Since the refutation is minimal, the last line can not be obtained by the reordering rule. Hence, the last line is obtained by the conjunction rule: $D_1 \wedge D_2 = \Box$, where D_1 and D_2 have the same order that we denote by π. Notice that by the minimality of the refutation, both D_1 and D_2 are satisfiable.

For every $i \in \{1, 2\}$, D_i is the conjunction of several clauses of $\mathrm{T}(G, c)$. Since D_i is satisfiable, this conjunction does not contain all clauses. Let $A_i \subseteq V$ be a set of vertices such that there is a clause from their parity conditions that is not included in D_i. Notice that $A_i \neq \emptyset$ for $i \in \{1, 2\}$.

We consider two cases.

First case. Assume that every two vertices $v \in A_1$ and $u \in A_2$ belong to the same long edge of G.

▷ Claim 3.4. There exists a subgraph G' of G such that $\mathrm{tw}(G') \geq \mathrm{tw}(G) - \max(3, \Delta(G) - 1)$ and at least one of the sets A_1 and A_2 does not intersect with the set of vertices of G'.

Proof. If A_2 contains an interior vertex of a long edge, then all vertices from A_1 belong to this long edge. Let x and y be the endpoints of this long edge. Let $H' = H \setminus \{x, y\}$ and G' be a subgraph of G corresponding to H'. Let us estimate $\mathrm{tw}(G')$:

$$\mathrm{tw}(G') \geq \mathrm{tw}(H') \geq \mathrm{tw}(H) - 2 \geq \mathrm{tw}(G) - 3.$$

In the first inequality we use Lemma 2.7, in the second one we use Lemma 2.6 and in the third one we use Lemma 2.8. Notice also that the set of vertices of G' does not intersect with A_1. The case in which A_1 contains an interior vertex of a long edge is analogous.

Now assume that A_1 and A_2 consist of only main vertices, so we may assume that A_1 and A_2 are vertices of H. Let u be a vertex from A_2. Then in the graph H every vertex from A_1 is either adjacent to u or equal to u, hence $|A_1| \leq \Delta(H) + 1 \leq \Delta(G) + 1$. Then we delete from G all vertices from A_1 and get G'. The set of vertices of G' does not intersect with A_1 and $\mathrm{tw}(G') \geq \mathrm{tw}(G) - \Delta(G) - 1$ by Lemma 2.6. ◁

W.l.o.g. we assume that the set of vertices of G' does not intersect with A_1. Let σ be a satisfying assignment of D_1 and let σ' be the restriction of σ to the all edges in $E(G) \setminus E(G')$. Let us denote $F \coloneqq D_1|_{\sigma'}$.

Notice that F computes a satisfiable Tseitin formula $\mathrm{T}(G', c')$ for some charging function c'. By Theorem 2.20, $|F| \geq 2^{\Omega(\mathrm{tw}(G'))}$. $\mathrm{tw}(G') \geq \mathrm{tw}(G) - \max(3, \Delta(G) - 1)$, and we already know that $\Delta(G) \leq \mathrm{tw}(G)/10$, hence $\mathrm{tw}(G') \geq 0.9\,\mathrm{tw}(G) - 3$. Since F is obtained by an application of a substitution from D_1, by Lemma 2.1, $|D_1| \geq |F|$. Hence, $|D_1| \geq 2^{\Omega(\mathrm{tw}(G))}$.

Second (main) case. Assume that there exist $v \in A_1$ and $u \in A_2$ that do not belong to the same long edge of G.

▶ **Theorem 3.5** ([29, Theorem 3.2]). *Let G be a 2-connected graph and v, u are two vertices from G. Then there is a path p connecting v and u such that $\mathrm{tw}(G \setminus V(p)) \geq c\,\mathrm{tw}(G)$, where $V(p)$ is the set of vertices of the path p and $c > 0$ is an absolute constant.*

Let p be a path between u and v in the graph G given by Theorem 3.5. We know that $\mathrm{tw}(G \setminus V(p)) \geq c\,\mathrm{tw}(G)$ for some constant $c > 0$. Since v and u do not belong to the same long edge, the distance between them in G is at least 2. By Lemma 3.3, the graph $G \setminus \{v, u\}$ is connected.

▶ **Lemma 3.6.** *Let $G = (V, E)$ be a 2-connected graph, $v, u \in V$, the distance between v and u be at least 2 and $G \setminus \{v, u\}$ be connected. Let p be a path between v and u, define $G_0 = G \setminus V(p)$.*

Let $\mathrm{T}(G, c)$ be an unsatisfiable Tseitin formula, CNF formulas φ_1 and φ_2 be the conjunctions of several clauses of $\mathrm{T}(G, c)$ such that φ_1 and φ_2 are satisfiable and $\varphi_1 \wedge \varphi_2$ is unsatisfiable. Let φ_1 does not contain a clause from the parity condition of v of formula $\mathrm{T}(G, c)$ and φ_2 does not contain a clause from the parity condition of u.

Then there exists such partial assignments α_1 and α_2 such that $\varphi_1|_{\alpha_1} \wedge \varphi_2|_{\alpha_2} = \mathrm{T}(G_0, c_0)$ and $\mathrm{T}(G_0, c_0)$ is satisfiable.

Using Lemma 3.6 we now complete the second case of the proof and, thus, we complete the whole proof of Theorem 3.2. Indeed, let φ_1 and φ_2 be the conjunctions of clauses included in D_1 and D_2 respectively. By the choice of p, $\mathrm{tw}(G_0) \geq c\,\mathrm{tw}(G)$, hence, the size of any OBDD computing $\varphi_1|_{\alpha_1} \wedge \varphi_2|_{\alpha_2}$ is at least $2^{\Omega(c\,\mathrm{tw}(G))}$ by Theorem 2.20. Hence, by Lemma 2.2 at least one of $\varphi_1|_{\alpha_1}$ or $\varphi_2|_{\alpha_2}$ requires $\pi-$OBDD of size at least $2^{\Omega(c\,\mathrm{tw}(G))/2}$, hence by Lemma 2.1 at least one of D_1 and D_2 has a size of at least $2^{\Omega(c\,\mathrm{tw}(G))/2}$.

Proof of Lemma 3.6. Let C_v be a clause from the parity condition of v that does not appear in φ_1 and C_u be a clause from the parity condition of u that does not appear in φ_2. Let e_v be an edge of p incident to v and e_u be an edge of p incident to u. Notice that $e_v \neq e_u$, since the distance between v and u is at least 2. We denote by $V(p)$ the set of vertices of p, and by $E(p)$ the set of edges of p.

Partial assignments α_1 and α_2 will have the same support corresponding to all edges incident to vertices of $V(p)$. Partial assignments α_1 and α_2 differ exactly on $E(p)$ and coincide on all other edges. We will define α_i in several steps. In the first and the second steps, we construct a partial assignment σ that is defined on the set of edges incident to v or u except e_v and e_u. In the third and fourth steps, we construct an assignment to edges of p. In the fifth step, we conclude with the construction of α_i, and in the final step we verify that substitutions α_i satisfy the required properties.

1. Construction of σ. We define an assignment σ with support $E(v) \sqcup E(u) \setminus E(p)$ such that it does not satisfy C_v and C_u. Such σ exists since $E(u)$ and $E(v)$ are disjoint.

2. Application of σ. Consider the Tseitin formula $\mathrm{T}(G, c + \mathbf{1}_v)$. It is satisfiable since $\mathrm{T}(G, c)$ is unsatisfiable and G is connected. By Lemma 2.14, $\mathrm{T}(G, c + \mathbf{1}_v)|_\sigma$ is also a Tseitin formula; let us denote it by $\mathrm{T}(G', c' + \mathbf{1}_v)$.

We claim that $\mathrm{T}(G', c' + \mathbf{1}_v)$ is satisfiable. By the condition of the lemma, $G \setminus \{v, u\}$ is connected. G' is connected since both v and u have in G' one edge connecting them with $G \setminus \{v, u\}$. The formula $\mathrm{T}(G, c + \mathbf{1}_v)$ is satisfiable and since the number of connected components does not increase after applying σ, the result of the substitution is also satisfiable.

Analogously define $\mathrm{T}(G', c' + \mathbf{1}_u) := \mathrm{T}(G, c + \mathbf{1}_u)|_\sigma$. Notice that G' and c' are the same as above.

3. Construction of ρ_1 and ρ_2. Consider arbitrary satisfying assignment of $\mathrm{T}(G', c' + \mathbf{1}_v)$ and let ρ_1 be its restriction to $E(p)$. We also define a partial assignment ρ_2 with the same support such that for all $e \in E(p)$, $\rho_2(x_e) = 1 - \rho_1(x_e)$.

4. Satisfiability of the conjunction. Let us define $\psi_1 := \varphi_1|_{\sigma \cup \rho_1}$ and $\psi_2 := \varphi_2|_{\sigma \cup \rho_2}$. We claim that the conjunction of $\psi_1 \wedge \psi_2$ is satisfiable.

The formula φ_1 is a subformula of $\mathrm{T}(G, c)$, so if we remove from φ_1 all clauses of the parity condition of v, then it will be a subformula of $\mathrm{T}(G, c + \mathbf{1}_v)$. By the construction of σ, C_v is the only clause from the parity condition of v in $\mathrm{T}(G, c)$ that is not satisfied by σ. But C_v is not included in φ_1, hence $\varphi_1|_\sigma$ does not contain clauses of the parity condition of v. Hence, $\varphi_1|_\sigma$ is a subformula of $\mathrm{T}(G, c + \mathbf{1}_v)|_\sigma = \mathrm{T}(G', c' + \mathbf{1}_v)$. Thus, ψ_1 is a subformula of $\mathrm{T}(G', c' + \mathbf{1}_v)|_{\rho_1}$.

Analogously, ψ_2 is a subformula of $\mathrm{T}(G', c' + \mathbf{1}_u)|_{\rho_2}$.

We claim that Tseitin formulas $\mathrm{T}(G', c' + \mathbf{1}_v)|_{\rho_1}$ and $\mathrm{T}(G', c' + \mathbf{1}_u)|_{\rho_2}$ coincide. To prove it, it is sufficient to verify that for each vertex from $V(p)$, its charges in both formulas are equal.

Consider a vertex $w \in V(p)$. If $w \in V(p) \setminus \{v, u\}$, then $(c' + \mathbf{1}_v)(w) = (c' + \mathbf{1}_u)(w)$. Assignments ρ_1 and ρ_2 change the charge of w in the same way. Indeed, let e_1 and e_2 be edges of p incident to w, then $\rho_2(x_{e_1}) + \rho_2(x_{e_2}) = (1 - \rho_1(x_{e_1})) + (1 - \rho_1(x_{e_2})) = \rho_1(x_{e_1}) + \rho_1(x_{e_2}) \pmod 2$.

If $w = v$, then $(c' + \mathbf{1}_v)(v) = 1 - c'(v)$ and $\rho_1(e_v) = 1 - \rho_2(e_v)$, then the charge of v in the formula $\mathrm{T}(G', c' + \mathbf{1}_v)|_{\rho_1}$ equals $(1 - c'(v)) + (1 - \rho_2(e_v)) = c'(v) + \rho_2(e_v) \pmod 2$, and that is the charge of v in the formula $\mathrm{T}(G', c' + \mathbf{1}_u)|_{\rho_2}$. The case $w = u$ is analogous.

Thus, we have that $\psi_1 \wedge \psi_2$ is a subformula of the satisfiable formula $\mathrm{T}(G', c' + \mathbf{1}_v)|_{\rho_1}$, so $\psi_1 \wedge \psi_2$ is satisfiable.

5. Construction of β. We define a partial assignment β on the edges of G that are not in $E(v) \cup E(u) \cup E(p)$ and incident to vertices of $V(p)$. Let β be such that $\mathrm{T}(G', c' + \mathbf{1}_v)|_{\rho_1 \cup \beta}$ is satisfiable. Let $\alpha_i := \sigma \cup \rho_i \cup \beta$ for $i \in \{1, 2\}$. Notice that $\mathrm{T}(G, c + \mathbf{1}_v)|_{\alpha_1} = \mathrm{T}(G_0, c_0)$ is a satisfiable formula for some c_0.

6. Final step. On step 4 we showed that $\varphi_1|_{\sigma \cup \rho_1} \wedge \varphi_2|_{\sigma \cup \rho_2} \subseteq \mathrm{T}(G, c + \mathbf{1}_v)|_{\sigma \cup \rho_1}$. Hence, $\varphi_1|_{\alpha_1} \wedge \varphi_2|_{\alpha_2} \subseteq \mathrm{T}(G, c + \mathbf{1}_v)|_{\alpha_1}$ Let us show that $\varphi_1|_{\alpha_1} \wedge \varphi_2|_{\alpha_2} = \mathrm{T}(G, c + \mathbf{1}_v)|_{\alpha_1}$.

Consider some $w \in V \setminus V(p)$, let $C_w|_{\alpha_1}$ be a clause from the parity condition of the vertex w in the formula $\mathrm{T}(G, c + \mathbf{1}_v)|_{\alpha_1}$. Notice that α_1 and α_2 assign the same values to edges incident to w, hence $C_w|_{\alpha_1} = C_w|_{\alpha_2}$.

$C_w|_{\alpha_1} \in \mathrm{T}(G, c + \mathbf{1}_v)|_{\alpha_1}$, then $C_w \in \mathrm{T}(G, c + \mathbf{1}_v)$, hence $C_w \in \mathrm{T}(G, c)$ (since $w \neq v$). The formula $\varphi_1 \wedge \varphi_2 \subseteq \mathrm{T}(G, c)$ is unsatisfiable and G is 2-connected, hence, by Lemma 2.18 every clause of $\mathrm{T}(G, c)$ appears in at least one of the formulas φ_1 or φ_2. If $C_w \in \varphi_1$, then $C_w|_{\alpha_1} \in \varphi_1|_{\alpha_1}$. If $C_w \in \varphi_2$, then $C_w|_{\alpha_2} \in \varphi_2|_{\alpha_2}$. ◀

◀

▶ Remark 3.7. Notice that in the proof of Theorem 3.1 we use not very much specific about OBDDs. Namely, we use only Lemmas 2.1, 2.2, 2.4 and for a lower bound on the size of OBDD we use Theorem 2.20.

In the recent work [13], de Colnet and Mengel introduce so-called str-DNNF(∧, r) refutations that are defined similarly to OBDD(∧, reordering) refutations, but use structured DNNFs instead of OBDDs. OBDD is a partial case of str-DNNF, an order of variables in OBDD corresponds to a vtree (variable tree) in str-DNNF. “r” stands for the restructuring and it is the extension of the reordering rule.

We claim that Theorem 3.1 also holds for str-DNNF(∧, r) refutations: there are lemmas analogous to Lemma 2.1, Lemma 2.2, Lemma 2.4 (see [26, Theorem 1], [13, Lemma 3]); and for a lower bound on the size of DNNF, one should use Theorem 4.1 (proved in Section 4) instead of Theorem 2.20.

▶ **Corollary 3.8.** *Let G be a graph and $\mathrm{T}(G, c)$ be an unsatisfiable Tseitin formula, $H_1, H_2, \dots, H_k$ be all unsatisfiable connected components of G. Then any* OBDD(∧, reordering) *refutation of $\mathrm{T}(G, c)$ has a size of at least $2^{\Omega(t)}$, where $t = \min_{i\in[k]} \mathrm{tw}(H_i)$.*

Proof. See Appendix B for the proof. ◀

3.2 Almost Automatability

▶ **Theorem 3.9.** *Let $\mathrm{T}(G, c)$ be unsatisfiable Tseitin formula based on graph $G = (V, E)$, and there is an* OBDD(∧, reordering) *refutation for it of size S. Then one can construct tree-like* OBDD(∧) *in time $S^{\mathcal{O}(\log |V|)}\mathrm{poly}(|\mathrm{T}(G, c)|)$.*

We use the following lemma.

▶ **Lemma 3.10** ([18, Corollary 6.3])**.** *Let $\mathrm{T}(G, c)$ be an unsatisfiable Tseitin formula based on a graph $G = (V, E)$ and P be a path decomposition of G. Given $\mathrm{T}(G, c)$ and P, one can construct a tree-like* OBDD(∧) *refutation of size $\mathcal{O}(|E||V|2^{w(P)} + |\mathrm{T}(G, c)|^2)$ in time that is polynomial of sizes of the input and the output.*

Proof of Theorem 3.9. Assume that G is connected. By Theorem 2.9, one can obtain a tree decomposition of width $\mathcal{O}(\mathrm{tw}(G))$ in time $2^{\mathcal{O}(\mathrm{tw}(G))}|V|$. Using Lemma 2.5, we construct a path decomposition of G of width $\mathcal{O}(\mathrm{tw}(G) \log |V|)$. Using Lemma 3.10, we build a tree-like OBDD(∧) refutation in time at most $\mathrm{poly}\left(|\mathrm{T}(G, c)|, 2^{\mathrm{tw}(G) \log |V|}\right)$ using this path decomposition.

Now consider the case when G is not necessarily connected and its unsatisfiable connected components are $\{H_1, \dots, H_k\}$. For each component H_i, we compute an approximation t_i of its treewidth using Theorem 2.9: $\mathrm{tw}(H_i) \leq t_i \leq \alpha\,\mathrm{tw}(H_i)$ for some constant $\alpha \geq 1$. We make it in $\sum_{i=1}^{k} 2^{\mathcal{O}(\mathrm{tw}(H_i))} \leq 2^{\mathcal{O}(\mathrm{tw}(G))}|V|$ time. Then we choose H_i with the smallest t_i, let it be H_a. We construct a tree-like OBDD($\wedge$, reordering) refutation of $\mathrm{T}(H_a, c)$ as above in time $\mathrm{poly}\left(|\mathrm{T}(H_a, c)|, 2^{t_a \log |V(H_a)|}\right)$. By Corollary 3.8, $S \geq 2^{\Omega(t)}$, where $t = \min_{i \in [k]} \mathrm{tw}(H_i)$, so $S \geq 2^{\Omega(t_a)}$.

Thus, the running time is at most $S^{\mathcal{O}(\log |V|)}\mathrm{poly}(|\mathrm{T}(G, c)|)$. ◀

4 Bounds on DNNF and Regular Resolution

The main result of this section is the following theorem:

▶ **Theorem 4.1.** *Let* $\mathrm{T}(G, c)$ *be satisfiable and* D *be a DNNF computing* $\mathrm{T}(G, c)$*. Then* $|D| \geq 2^{\Omega(\mathrm{tw}(G))}$*.*

The following reduction from DNNF to regular resolution theorem was proved by de Colnet and Mengel [12].

▶ **Theorem 4.2** ([12, Theorem 8])**.** *Let* $\mathrm{T}(G, c)$ *be an unsatisfiable Tseitin formula where* G *is connected and let* S *be the size of its smallest resolution refutation. Then for every satisfiable Tseitin formula* $\mathrm{T}(G, c')$ *there exists a DNNF of size* $\mathcal{O}(S \times |V(G)|)$ *computing it.*

Theorem 4.2 and Theorem 4.1 imply the following theorem.

▶ **Theorem 4.3.** *Let* $G = (V, E)$ *be a connected graph and* $\mathrm{T}(G, c)$ *be an unsatisfiable formula. Then any regular resolution refutation of* $\mathrm{T}(G, c)$ *has a size of at least* $2^{\Omega(\mathrm{tw}(G))}$*.*

Proof. See Appendix C.1 for the proof. ◀

In Appendix C.2, we also prove a matching upper bound.

▶ **Theorem 4.4.** *Let* $G = (V, E)$ *be a graph and* $\mathrm{T}(G, c)$ *be a satisfiable Tseitin formula. Then there exists a DNNF of size at most* $2^{\mathcal{O}(\mathrm{tw}(G))} \cdot |E|$ *computing* $\mathrm{T}(G, c)$*.*

Proof sketch. We consider a nice tree decomposition T with "introduce edge" nodes. We construct a DNNF D such that for every node $t \in T$ with bag X_t and for every charge function $f \colon X_t \to \{0, 1\}$ there exists a node $d_{t,f} \in D$ that computes $\mathrm{T}(G_t, c_t)$, where G_t is a subgraph of G corresponding to the subtree of t, and c_t acts on X_t as f and on the other vertices as c. We make it by bottom-up induction on the T. The root of T gives a node in D that computes $\mathrm{T}(G, c)$. ◀

4.1 Rectangle game

De Colnet and Mengel [12] proposed a game to prove DNNF lower bounds. For simplicity, we describe it only in a special case when the computed function is a Tseitin formula.

Let X be a set of propositional variables. (X_1, X_2) is called a *variable partition* if $X_1 \sqcup X_2 = X$ and X_1, X_2 are not empty. If X is a set of variables of a Tseitin formula $\mathrm{T}(G, c)$ based on a graph $G = (V, E)$, then every variable partition (X_1, X_2) naturally corresponds to an edge partition (E_1, E_2).

A (combinatorial) *rectangle* for a variable partition (X_1, X_2) of a variable set X is defined to be a set of full assignments of form $R = A \times B$ where $A \subseteq \{0, 1\}^{X_1}$ and $B \subseteq \{0, 1\}^{X_2}$. A rectangle R *respects* a Boolean function $f \colon \{0, 1\}^X \to \{0, 1\}$ if $R \subseteq \mathrm{sat}(f)$, i.e. R consists only of satisfying assignments of f.

We define the *adversarial multi-partition rectangle cover game* for a satisfiable Tseitin formula $\mathrm{T}(G, c)$ with a set of variables X to be played as follows: two players, Charlotte and Adam, construct in several rounds a set $\mathcal{R}$ of combinatorial rectangles that respect $\mathrm{T}(G, c)$ and cover the set $\mathrm{sat}(\mathrm{T}(G, c))$.

The game starts with $\mathcal{R} = \emptyset$ and consists of several rounds. On each round Charlotte chooses an input $a \in \mathrm{sat}(\mathrm{T}(G, c))$ and a branch decomposition T of G. Then Adam chooses an edge e of T and let (E_1, E_2) be the e-separation. Then Charlotte chooses a rectangle R for the corresponding partition of variables of $\mathrm{T}(G, c)$ that respects $\mathrm{T}(G, c)$ and covers a, and adds R to $\mathcal{R}$. This completes the round.

The game is over when $\mathrm{sat}(\mathrm{T}(G, c))$ is covered by $\mathcal{R}$. The adversarial multi-partition rectangle complexity of $\mathrm{T}(G, c)$, denoted by $\mathrm{aR}(\mathrm{T}(G, c))$ is the minimum number of rounds in which Charlotte can finish the game, whatever the choices of Adam are.

▶ **Theorem 4.5** ([12, Theorem 16]). *Let D be a complete DNNF computing a satisfiable Tseitin formula $\mathrm{T}(G, c)$. Then $|D| \geq \mathrm{aR}(\mathrm{T}(G, c))$.*

In the next subsection, we prove the following lemma:

▶ **Lemma 4.6.** *Let $\mathrm{T}(G, c)$ be a satisfiable Tseitin formula where G is a 3-connected graph. Then $\mathrm{aR}(\mathrm{T}(G, c)) \geq 2^{\Omega(\mathrm{bw}(G))}$.*

Let us prove Theorem 4.1 using this lemma.

Proof of Theorem 4.1. Let S be the minimum size of a DNNF computing $\mathrm{T}(G, c)$. If $\mathrm{tw}(G) \leq 2$, then the statement is trivial. Otherwise, by Theorem 2.10, there is a 3-connected graph G' such that G' is a topological minor of G and $\mathrm{tw}(G') = \mathrm{tw}(G)$. By Lemma 2.19, there is a DNNF of size at most S computing a satisfiable $\mathrm{T}(G', c')$. By Lemma 2.3, there is a *complete* DNNF of size $S' \leq S^{\mathcal{O}(1)}$ computing $\mathrm{T}(G', c')$. By Theorem 4.5, $S' \geq \mathrm{aR}(\mathrm{T}(G, c))$. By Lemma 4.6, $\mathrm{aR}(\mathrm{T}(G, c)) \geq 2^{\Omega(\mathrm{bw}(G))}$. Note that $\mathrm{bw}(G) = \Theta(\mathrm{tw}(G))$ by Theorem 2.11.

Thus, $S' \geq \mathrm{aR}(\mathrm{T}(G, c)) \geq 2^{\Omega(\mathrm{tw}(G))}$; hence, $S \geq 2^{\Omega(\mathrm{tw}(G))}$. ◀

4.2 Proof of Lemma 4.6

Our goal is to prove the inequality $\mathrm{aR}(\mathrm{T}(G, c)) \geq 2^{\Omega(\mathrm{tw}(G))}$ for a 3-connected graph G (i.e. to prove Lemma 4.6). The plan of the proof is the following. We will describe a strategy for Adam. The goal of Adam is to play such that every rectangle R chosen by Charlotte has a small size, so a large number of such rectangles is required to cover $\mathrm{sat}(\mathrm{T}(G, c))$. We show that there exists a formula $\mathrm{T}(G', c')$ such that R is a subset of $\mathrm{sat}(\mathrm{T}(G', c'))$ and that Adam can play in such a way that the number of satisfying assignments of $\mathrm{T}(G', c')$ is small, hence $|R|$ is also small.

Let $G = (V, E)$ be a graph with edges colored in two colors: $E = E_1 \sqcup E_2$; edges in E_1 are colored with the first color and edges in E_2 are colored with the second one. We call a vertex $v \in V$ *bicolored*, if there are edges of both colors that are incident to it; we call a set $A \subseteq V$ *bicolored* if all vertices in it are bicolored (it is not necessary that all bicolored vertices are in A).

Let us construct a new graph $\mathrm{Split}(G, E_1, E_2, A) = (V', E')$: we split each node v in A into two fresh nodes and direct each edge e incident to v to one of the copies depending on the color of e. More formally, let

$$V' = V \setminus A \cup \{v^i \mid i \in \{1, 2\} \text{ and } v \in A\};$$

$$E' = \{(f_i(v), f_i(u)) \mid i \in \{1, 2\} \text{ and } (v, u) \in E_i\}, \text{ where } f_i(v) = \begin{cases} v^i, & v \in A \\ v, & v \notin A \end{cases}.$$

▶ **Lemma 4.7** (Generalization of Lemma 18 and Lemma 21 from [12]). *Let $G = (V, E)$ be a graph, $E = E_1 \sqcup E_2$ be a coloring of the edges in two colors, and $A \subseteq V$ be a bicolored set.*

Let $\mathrm{T}(G, c)$ *be a satisfiable Tseitin formula,* $R \subseteq \mathrm{sat}(\mathrm{T}(G, c))$ *be a rectangle w.r.t. to the partition* (E_1, E_2)*. Then for a graph* $G' = \mathrm{Split}(G, E_1, E_2, A)$ *and a charge function* c' *such that* $\mathrm{T}(G', c')$ *is satisfiable, the following holds:* $|R| \le |\mathrm{sat}(\mathrm{T}(G', c'))| = 2^{|E|-(|V|+|A|)+\#G'}$.

Proof. See Appendix C.1 for the proof. ◀

In [12] this statement is proven for the case when A is an independent set, but this restriction actually is not used in the proof. In appendix, we prove this lemma for arbitrary bicolored A explicitly.

Lemma 4.7 yields an upper bound for the size of Charlotte's rectangle if the set of bicolored vertices is large enough. However, we have a summand $\#G'$ in the exponent. In [12] the authors make sure that $\#G' = 1$ restricting A to be a specific independent set. We weaken this condition and simply make sure that $\#G'$ is not too large, which makes it possible for us to pick a larger set A (and we do not require that A is an independent set).

Proof of Lemma 4.6. To prove the lower bound on aR, we show a winning strategy for Adam. Let a be an assignment and T be a branch decomposition picked by Charlotte. By definition of the branchwidth, there exists a cut of T that yields a partition of the edges $E(G) = E_1 \sqcup E_2$ such that there are at least $\mathrm{bw}(G)$ bicolored vertices. Adam chooses such a cut, let B be the set of bicolored vertices. Then Charlotte picks a rectangle R respecting the partitions (E_1, E_2). We will show that $|R| \le |\mathrm{sat}(\mathrm{T}(G, c))|\, 2^{-\Omega(\mathrm{bw}(G))}$, hence, there are at least $2^{\Omega(\mathrm{bw}(G))}$ rounds. We denote $N = |\mathrm{sat}(\mathrm{T}(G, c))| = 2^{|E|-|V|+1}$. Our goal is to show that $|R| \le 2^{-\Omega(\mathrm{bw}(G))} N$.

For a graph F, we denote by $\deg_F(v)$ the number of different neighbors of v except for v itself (it differs from the usual degree of v since we count all parallel edges only once and do not count self-loops at all).

Let H be a subgraph of G induced by vertices B. We consider the following set of low-degree vertices: $B_{\le 2} = \{v \in B \mid \deg_H(v) \le 2\}$. We consider two cases depending on whether $B_{\le 2}$ is large or not.

First case: $B_{\le 2}$ is large. Assume that $|B_{\le 2}| \ge |B|/100$. $B_{\le 2}$ contains an independent (in H and consequently in G) set I of size at least $|B_{\le 2}|/(2 + 1)$. Observe that I is bicolored as a subset of a bicolored set B.

▶ **Lemma 4.8** ([12, Lemma 22]). *Let* $G = (V, E)$ *be a 3-connected graph,* $E = E_1 \sqcup E_2$ *be a coloring of the edges in two colors,* $A \subseteq V$ *be an independent set in* G *and bicolored.*

Then there exists $S \subseteq A$ *such that* $|S| \ge |A|/3$ *and* $\mathrm{Split}(G, E_1, E_2, S)$ *is connected.*

Applying this lemma to I, we get a set S of size at least $|I|/3$ such that $\mathrm{Split}(G, E_1, E_2, S)$ is connected. Application of Lemma 4.7 to S yields the inequality $|R| \le 2^{|E|-(|V|+|S|)+1} = 2^{-|S|}N$. $|S| \ge |I|/3 \ge |B_{\le 2}|/9 \ge |B|/900 \ge \mathrm{bw}(G)/900$ which completes the proof in the first case.

Second case: $B_{\le 2}$ is small. Now assume that $|B_{\le 2}| < |B|/100$. Let $G' = \mathrm{Split}(G, E_1, E_2, B)$, $B'_i = \{v^i \mid v \in B\}$ for $i \in \{1, 2\}$ and $B' = B'_1 \sqcup B'_2$ be the set of copies of vertices from B in G'. Let H' be a subgraph of G' induced by vertices of B'. Observe that $\deg_H(v) = \deg_{H'}(v^1) + \deg_{H'}(v^2)$ for every $v \in B$.

Note that if we add an edge (v^1, v^2) for each $v \in B$ in graph G', then it becomes connected since G is connected. Hence, each connected component C of G' intersects B'. We call an intersection of C and B' as the *imprint* of C on B'. We are going to bound the number of connected components in G' by estimating the sizes of these imprints.

Let $v \in B'$ and $v \in C$, where C is a connected component of G'. Denote by $h(v) = |C \cap B'|$ the size of the imprint of the component C. Let $w(v) = 1/h(v)$ be a weight of a vertex $v \in B'$. Notice that the sum of weights $\sum_{v \in B} \big(w(v^1) + w(v^2)\big)$ equals the number of connected components in G'.

Fix a node $v \in B$ which has been split into v^1 and v^2. W.l.o.g. we assume that $h(v^1) \geq h(v^2)$. Let us consider the following cases:

$h(v^2) = 1$. Observe that $\deg_{G'}(v^2) \geq 1$ (otherwise v is not incident to any edge from E_2). Thus the connected component of v^2 in G' contains some vertices except v^2; let us denote the set of these vertices as X. The imprint of this component, by the assumption, contains a single vertex. Then, if v is removed from G the vertices of X become not reachable from the rest of the vertices of G. It is easy to see that there are vertices in G being not in $X \cup \{v\}$: consider the neighbors of v that do not belong to X, which exist since v is bicolored. Therefore, G is not 2-connected, which is a contradiction, so this case is impossible.

$h(v^2) = 2$ and $h(v^1) = 2$. Observe that for every node $u \in B'$ the inequality $h(u) \geq \deg_{H'}(u) + 1$ holds, so $\deg_{H'}(v^1) + \deg_{H'}(v^2) + 2 \leq h(v^1) + h(v^2) = 4$, i.e. $\deg_H(v) = \deg_{H'}(v^1) + \deg_{H'}(v^2) \leq 4 - 2 = 2$. Then we have $v \in B_{\leq 2}$. There are at most $|B|/100$ such nodes v. The weight in this case equals $w(v^1) + w(v^2) = 1/2 + 1/2 = 1$.

$h(v^2) \geq 2$ and $h(v^1) \geq 3$. Then $w(v^1) + w(v^2) \leq 1/2 + 1/3 = 5/6$.

Now the number of connected components of G' can be estimated as follows:

$$\#G' = \sum_{v \in B} \big(w(v^1) + w(v^2)\big) \leq 1 \times |B|/100 + (5/6) \times |B| < 0.9|B|.$$

Applying Lemma 4.7 to B we get that

$$|R| \leq 2^{|E|-(|V|+|B|)+0.9|B|} = 2^{-0.1|B|-1} \cdot 2^{|E|-|V|+1} \leq 2^{-0.1\,\mathrm{bw}(G)-1} N = 2^{-\Omega(\mathrm{bw}(G))} N. \quad ◀$$

The statement. The content of this statement had to be truncated due to de facto introduced censorship in Russia. Nevertheless, the authors express their condolences to all the victims of the events taking place in Ukraine.

References

1 Michael Alekhnovich and Alexander A. Razborov. Satisfiability, Branch-Width and Tseitin tautologies. *Computational Complexity*, 20(4):649–678, 2011. doi:10.1007/s00037-011-0033-1.

2 Albert Atserias and Moritz Müller. Automating Resolution is NP-Hard. *J. ACM*, 67(5):31:1–31:17, 2020. doi:10.1145/3409472.

3 Eli Ben-Sasson. Hard examples for the bounded depth Frege proof system. *Computational Complexity*, 11(3-4):109–136, 2002. doi:10.1007/s00037-002-0172-5.

4 Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. A $c^k n$ 5-Approximation Algorithm for Treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016. doi:10.1137/130947374.

5 Randal E. Bryant. Symbolic manipulation of Boolean functions using a graphical representation. In Hillel Ofek and Lawrence A O'Neill, editors, *Proceedings of the 22nd ACM/IEEE conference on Design automation, DAC 1985, Las Vegas, Nevada, USA, 1985.*, pages 688–694. ACM, 1985. doi:10.1145/317825.317964.

6 Sam Buss, Dmitry Itsykson, Alexander Knop, Artur Riazanov, and Dmitry Sokolov. Lower Bounds on OBDD Proofs with Several Orders. *ACM Trans. Comput. Log.*, 22(4):26:1–26:30, 2021. `doi:10.1145/3468855`.

7 Sam Buss, Dmitry Itsykson, Alexander Knop, and Dmitry Sokolov. Reordering Rule Makes OBDD Proof Systems Stronger. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPIcs*, pages 16:1–16:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. `doi:10.4230/LIPIcs.CCC.2018.16`.

8 Julia Chuzhoy and Zihan Tan. Towards Tight(er) Bounds for the Excluded Grid Theorem. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1445–1464, 2019. `doi:10.1137/1.9781611975482.88`.

9 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. `doi:10.1007/978-3-319-21275-3`.

10 Stefan S. Dantchev and Søren Riis. Tree Resolution Proofs of the Weak Pigeon-Hole Principle. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity, Chicago, Illinois, USA, June 18-21, 2001*, pages 69–75, 2001. `doi:10.1109/CCC.2001.933873`.

11 Adnan Darwiche and Pierre Marquis. A Knowledge Compilation Map. *J. Artif. Intell. Res.*, 17:229–264, 2002. `doi:10.1613/jair.989`.

12 Alexis de Colnet and Stefan Mengel. Characterizing Tseitin-Formulas with Short Regular Resolution Refutations. In Chu-Min Li and Felip Manyà, editors, *Theory and Applications of Satisfiability Testing - SAT 2021 - 24th International Conference, Barcelona, Spain, July 5-9, 2021, Proceedings*, volume 12831 of *Lecture Notes in Computer Science*, pages 116–133. Springer, 2021. `doi:10.1007/978-3-030-80223-3_9`.

13 Alexis de Colnet and Stefan Mengel. Lower Bounds on Intermediate Results in Bottom-Up Knowledge Compilation. *CoRR*, abs/2112.12430, 2021. `arXiv:2112.12430`.

14 Susanna F. de Rezende, Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, and Dmitry Sokolov. Automating algebraic proof systems is NP-hard. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 209–222. ACM, 2021. `doi:10.1145/3406325.3451080`.

15 Nicola Galesi, Dmitry Itsykson, Artur Riazanov, and Anastasia Sofronova. Bounded-Depth Frege Complexity of Tseitin Formulas for All Graphs. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPIcs*, pages 49:1–49:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. `doi:10.4230/LIPIcs.MFCS.2019.49`.

16 Michal Garlík. Failure of Feasible Disjunction Property for k-DNF Resolution and NP-hardness of Automating It. *Electron. Colloquium Comput. Complex.*, page 37, 2020. URL: `https://eccc.weizmann.ac.il/report/2020/037`.

17 Ludmila Glinskih and Dmitry Itsykson. Satisfiable Tseitin Formulas Are Hard for Nondeterministic Read-Once Branching Programs. In Kim G. Larsen, Hans L. Bodlaender, and Jean-François Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*, volume 83 of *LIPIcs*, pages 26:1–26:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. `doi:10.4230/LIPIcs.MFCS.2017.26`.

18 Ludmila Glinskih and Dmitry Itsykson. On Tseitin Formulas, Read-Once Branching Programs and Treewidth. *Theory Comput. Syst.*, 65(3):613–633, 2021. `doi:10.1007/s00224-020-10007-8`.

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

20 Daniel J. Harvey and David R. Wood. The treewidth of line graphs. *J. Comb. Theory, Ser. B*, 132:157–179, 2018. doi:10.1016/j.jctb.2018.03.007.

21 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. doi:10.1109/FOCS.2017.18.

22 Dmitry Itsykson, Alexander Knop, Andrei E. Romashchenko, and Dmitry Sokolov. On OBDD-based Algorithms and Proof Systems that Dynamically Change the order of Variables. *J. Symb. Log.*, 85(2):632–670, 2020. doi:10.1017/jsl.2019.53.

23 Dmitry Itsykson, Artur Riazanov, Danil Sagunov, and Petr Smirnov. Near-Optimal Lower Bounds on Regular Resolution Refutations of Tseitin Formulas for All Constant-Degree Graphs. *Comput. Complex.*, 30(2):13, 2021. doi:10.1007/s00037-021-00213-2.

24 Ephraim Korach and Nir Solel. Tree-Width, Path-Width, and Cutwidth. *Discret. Appl. Math.*, 43(1):97–101, 1993. doi:10.1016/0166-218X(93)90171-J.

25 Jan Krajíček. *Proof Complexity*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2019. doi:10.1017/9781108242066.

26 Knot Pipatsrisawat and Adnan Darwiche. New Compilation Languages Based on Structured Decomposability. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 517–522. AAAI Press, 2008. URL: http://www.aaai.org/Library/AAAI/2008/aaai08-082.php.

27 Neil Robertson and Paul D. Seymour. Graph Minors. II. Algorithmic Aspects of Tree-Width. *J. Algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4.

28 Neil Robertson and Paul D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *J. Comb. Theory, Ser. B*, 52(2):153–190, 1991. doi:10.1016/0095-8956(91)90061-N.

29 Neil Robertson, Paul D. Seymour, and Robin Thomas. Quickly Excluding a Planar Graph. *J. Comb. Theory, Ser. B*, 62(2):323–348, 1994. doi:10.1006/jctb.1994.1073.

30 G.S. Tseitin. On the complexity of derivation in the propositional calculus. In *Studies in Constructive Mathematics and Mathematical Logic Part II*. A. O. Slisenko, editor, 1968.

31 Alasdair Urquhart. Hard examples for resolution. *J. ACM*, 34(1):209–219, 1987. doi:10.1145/7531.8928.

32 Ingo Wegener. *Branching Programs and Binary Decision Diagrams*. SIAM, 2000. URL: http://ls2-www.cs.uni-dortmund.de/monographs/bdd/.

A Proofs for Section 2 (Preliminaries)

▶ **Lemma 2.13** (Folklore). *Let $G = (V, E)$ be a graph, $\mathrm{T}(G, c)$ be satisfiable, σ be a full assignment for the set of variables of $\mathrm{T}(G, c)$. Then the number of parity conditions falsified by σ is even.*

Proof. Let $A \subseteq V$ be a set of vertices with falsified parity conditions. Consider the sum $S = \sum_{v\in V}\sum_{e\in E(v)} \sigma(x_e) \equiv \sum_{v\in A}(1 - c(v)) + \sum_{v\notin A} c(v) \equiv |A| + \sum_{v\in V} c(v) \equiv |A| \pmod 2$, where the last congruence holds since $\mathrm{T}(G, c)$ is satisfiable. On the other hand, $S \equiv 0 \pmod 2$ since for each edge e the summand $\sigma(x_e)$ appears in the sum twice. ◀

B Proofs for Section 3 ($\mathrm{OBDD}(\wedge, \mathrm{reordering})$)

▶ **Lemma 3.3.** *Let G be an almost 3-connected graph. Let u and v be two vertices that do not belong to the same long edge. Then the graph $G \setminus \{v, u\}$ is connected.*

Proof. Let G is a subdivision of a 3-connected graph H. Since u and v do not belong to the same long edge, every interior vertex is connected in $G \setminus \{v, u\}$ with some non-deleted main vertex. So it is sufficient to prove that all non-deleted main vertices are in the same connected component in $G \setminus \{v, u\}$.

The graph $G \setminus \{v, u\}$ is obtained from G by deletion of two vertices. We may look on the deletion of an interior vertex from a long edge in G as on the deletion of the corresponding edge in H. We also look on the deletion of a main vertex in G as on the deletion of the corresponding vertex in H. Since H is 3-connected, then it remains to be connected after removing of either two vertices, or two edges, or one vertex and one edge. Hence, all non-deleted main vertices of $G \setminus \{v, u\}$ belong to one connected component. ◀

In order to handle not only connected graphs, we need to show that $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ satisfies the *strong feasible disjunction property* [25].

▶ **Lemma B.1.** *Let $\varphi(\vec{x})$ and $\psi(\vec{y})$ be two CNF formulas with disjoint sets of variables. If there is an $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ refutation of $\varphi \wedge \psi$ of size S, then at least one of φ or ψ has an $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ refutation of size at most S.*

Proof. Consider the smallest $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ refutation of $\varphi \wedge \psi$; its size is at most S. The last OBDD in this refutation is $\varphi' \wedge \psi'$, where φ' is a subformula φ and ψ' is a subformula of ψ. Since variables of φ' and ψ' are disjoint, then at least one of φ' and ψ' is unsatisfiable.

W.l.o.g. assume that φ' is unsatisfiable. All previous OBDDs in the refutation are satisfiable, for each OBDD D in the refutation we substitute the values of variables $\vec{y}$ that satisfies all clauses of ψ included in D. The result of this substitution is equivalent to the part of D that contains only clauses from φ.

By Lemma 2.1 such substitution does not increase the size of D. After all such substitutions we obtain a correct $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ refutation of φ of size at most S. ◀

▶ **Corollary 3.8.** *Let G be a graph and $\mathrm{T}(G, c)$ be an unsatisfiable Tseitin formula, $H_1, H_2, \ldots, H_k$ be all unsatisfiable connected components of G. Then any $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ refutation of $\mathrm{T}(G, c)$ has a size of at least $2^{\Omega(t)}$, where $t = \min_{i \in [k]} \mathrm{tw}(H_i)$.*

Proof. Let S be the size of an $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ refutation of $\mathrm{T}(G, c)$. Let $H_{k+1}, \ldots, H_{k+m}$ be all satisfiable connected components of G, where $m \geq 0$. Notice that the sets of variables of $\mathrm{T}(H_i, c)$ are disjoint for different $i \in [k + m]$. By $k + m - 1$ applications of Lemma B.1 we get that for some $i \in [k]$ there exists an $\mathrm{OBDD}(\wedge, \mathrm{reordering})$ refutation of $\mathrm{T}(H_i, c)$ of size S. By Theorem 3.1, S is at least $2^{\Omega(\mathrm{tw}(H_i))}$. ◀

C Proofs for Section 4 (Bounds on DNNF and Regular Resolution)

C.1 Lower Bound

▶ **Theorem 4.3.** *Let $G = (V, E)$ be a connected graph and $\mathrm{T}(G, c)$ be an unsatisfiable formula. Then any regular resolution refutation of $\mathrm{T}(G, c)$ has a size of at least $2^{\Omega(\mathrm{tw}(G))}$.*

Proof. Consider a regular resolution refutation of $\mathrm{T}(G, c)$, let S be its size.

By Theorem 4.1 there exists a constant $\alpha > 0$ such that any DNNF computing $\mathrm{T}(G, c)$ has a size of at least $2^{\alpha\, \mathrm{tw}(G)}$. By Theorem 4.2, S is at least $2^{\alpha\, \mathrm{tw}(G)}/|V| = 2^{\alpha\, \mathrm{tw}(G) - \log|V|}$. If $\alpha\, \mathrm{tw}(G) - \log|V| \geq \alpha\, \mathrm{tw}(G)/2$, then $S \geq 2^{\alpha\, \mathrm{tw}(G)/2} = 2^{\Omega(\mathrm{tw}(G))}$.

Otherwise, $\alpha\, \mathrm{tw}(G) - \log|V| < \alpha\, \mathrm{tw}(G)/2$, i.e. $\log|V| > \alpha\, \mathrm{tw}(G)/2$. A resolution refutation must use at least one clause of each vertex $v \in V$ (otherwise it is also a refutation of satisfiable $\mathrm{T}(G, c + \mathbf{1}_v)$), so its size $S \geq |V| = 2^{\log|V|} > 2^{\alpha\, \mathrm{tw}(G)/2} = 2^{\Omega(\mathrm{tw}(G))}$. ◀

▶ **Lemma 4.7** (Generalization of Lemma 18 and Lemma 21 from [12]). *Let $G = (V, E)$ be a graph, $E = E_1 \sqcup E_2$ be a coloring of the edges in two colors, and $A \subseteq V$ be a bicolored set.*

Let $\mathrm{T}(G, c)$ be a satisfiable Tseitin formula, $R \subseteq \mathrm{sat}(\mathrm{T}(G, c))$ be a rectangle w.r.t. to the partition (E_1, E_2). Then for a graph $G' = \mathrm{Split}(G, E_1, E_2, A)$ and a charge function c' such that $\mathrm{T}(G', c')$ is satisfiable, the following holds: $|R| \leq |\mathrm{sat}(\mathrm{T}(G', c'))| = 2^{|E|-(|V|+|A|)+\#G'}$.

Proof. Let $\sigma = (\sigma_1, \sigma_2) \in R$, where σ_i assigns values to $\{x_e \mid e \in E_i\}$ for $i \in \{1, 2\}$.

Let $v \in A$ be bicolored. We denote $c_i^\sigma(v) = \sum_{e \in E(v) \cap E_i} \sigma_i(x_e) \bmod 2$. Observe that $c_1^\sigma(v) + c_2^\sigma(v) = c(v)$, since σ is a satisfying assignment of $\mathrm{T}(G, c)$. Since R is a rectangle, for any $\sigma' = (\sigma_1', \sigma_2') \in R$ the condition $c_i^{\sigma'}(v) = c_i^\sigma(v)$ holds for $i \in \{1, 2\}$. Thus, c_i^σ does not depend on σ, so we define $c_i(v) = c_i^\sigma(v)$ for each $i \in \{1, 2\}$ and $v \in A$.

Now let us consider the graph $G' = \mathrm{Split}(G, E_1, E_2, A)$. We define the charging function c' as $c'(v^i) = c_i(v)$ for $v \in A$ and $i \in \{1, 2\}$, and as $c'(v) = c(v)$ for $v \notin A$.

Let $\tau \in R$, define a full assignment τ' of variables of $\mathrm{T}(G', c')$ in the following way: $\tau'(x_{(f_i(v), f_i(u))}) = \tau_i(x_{(v,u)})$ for $(v, u) \in E_i$. Observe that τ' is a satisfying assignment of $\mathrm{T}(G', c')$ by the construction of c'. Thus, for each $\tau \in R$, there is a corresponding $\tau' \in \mathrm{sat}(\mathrm{T}(G', c'))$, and the correspondence function is injective. Hence, $|R| \leq \mathrm{sat}(\mathrm{T}(G', c'))$.

Finally, by Lemma 2.16, $|\mathrm{sat}(\mathrm{T}(G', c'))| = 2^{|E|-(|V|+|A|)+\#G'}$. ◀

C.2 Upper Bound

In this subsection, we prove an upper bound on the size of the smallest DNNF that computes a satisfiable Tseitin formula.

Let $T = (V_T, E_T)$ be a tree decomposition of $G = (V, E)$, and $\{X_t\}_{t \in V_T}$ be its bags. T is called an *extended version of nice tree decomposition* (ENDT) [9, Section 7.3.2], if the following conditions hold:

1. There is a distinguished node $r \in V_T$ such that $X_r = \emptyset$. We call r the root of T and assume that T is rooted.
2. Every $t \in V_T$ has one of the following types:
   a. t is a leaf node: t has no children and $X_t = \emptyset$;
   b. t introduces vertex $v \in V$: t has exactly one child s and $X_t = X_s \sqcup \{v\}$;
   c. t forgets vertex $v \in V$: t has exactly one child s and $X_s = X_t \sqcup \{v\}$;
   d. t introduces edge $(v, u) \in E$: t has exactly one child s, the vertices $v, u \in X_s$, and $X_t = X_s$;
   e. t is a join node: t has exactly two children s_1 and s_2, $X_t = X_{s_1} = X_{s_2}$.
3. Every edge $e \in E$ is introduced exactly once in T.

Since ENTD is also a plain tree decomposition, each vertex $v \in V$ is forgotten exactly once in ENTD.

▶ **Lemma C.1** ([9, Section 7.3.2]). *Let G be a graph without parallel edges and self-loops, and T be its tree decomposition T of width k. Then one can construct an ENTD of width at most k and size at most $\mathcal{O}(k|V(G)|)$ in $\mathrm{poly}(|V(G)|, |V(T)|)$ time.*

▶ **Theorem 4.4.** *Let $G = (V, E)$ be a graph and $\mathrm{T}(G, c)$ be a satisfiable Tseitin formula. Then there exists a DNNF of size at most $2^{\mathcal{O}(\mathrm{tw}(G))} \cdot |E|$ computing $\mathrm{T}(G, c)$.*

Proof. Since $\mathrm{T}(G, c)$ is satisfiable, all isolated vertices have zero charges, so we can delete them from the graph. So we can assume that $V = \mathcal{O}(E)$. Also, we can assume that there are no self-loops since they do not change a Tseitin formula.

Let $G' = (V, E')$ be a graph obtained from G as follows: for each $v, u \in V$, if there are several parallel edges between v and u, we delete all of them except one. By Lemma C.1, there exists an ENTD T' for graph G' with $w(T') = \mathrm{tw}(G')$ and $|T'| = \mathcal{O}(\mathrm{tw}(G')|V|)$. Note that $\mathrm{tw}(G') = \mathrm{tw}(G)$.

Now we construct ENTD for G from T' in the following way: if $t \in T'$ introduces edge $e = (v, u)$, we replace t with a path of nodes that introduce all parallel edges between v and u in G. The width of T is $\mathrm{tw}(G') = \mathrm{tw}(G)$ and its size is at most $|T'| + |E| \leq \mathcal{O}(\mathrm{tw}(G')|V| + |E|) = \mathcal{O}(\mathrm{tw}(G)|V| + |E|)$.

For each $t \in T$, we denote by $G_t = (V_t, E_t)$ a subgraph of G such that V_t and E_t are the sets of vertices and edges that are introduced in the subtree of t. Notice that $G_r = G$.

We claim that there exists a DNNF D of size $2^{\mathcal{O}(\mathrm{tw}(G))}|T|$ such that for every node $t \in T$ for every charge function $f : X_t \to \{0,1\}$ there exists a node $d_{t,f} \in D$ that computes $\varphi_{t,f} := \mathrm{T}(G_t, f \sqcup c|_{V_t \setminus X_t})$. Moreover, the subcircuit of a node $d_{t,f}$ uses only variables corresponding to the edges of G_t. Then the subcircuit of $d_{r,f_\emptyset}$ is the required DNNF computing $\mathrm{T}(G, c)$, where $f_\emptyset$ is the function with an empty domain.

We consider the nodes of T in such an order that the distance to the root of T does not increase. For each considered $t \in$T, we add to D at most $2^{\alpha\,\mathrm{tw}(G)}$ nodes for some constant α. Then for each charge function $f : X_t \to \{0,1\}$ we select a node $d_{t,f}$ of D such that $d_{t,f}$ computes $\varphi_{t,f}$. Every new $\wedge$-node in D will be decomposable, so D stays DNNF. Initially, D is an empty DNNF.

Assume that we consider a node t of T and a charge function $f : X_t \to \{0,1\}$. There are several cases.

t is a leaf node. $V_t = \emptyset$, thus f is a function with an empty domain and $\varphi_{t,f}$ is identically true, so add to D a gate labeled with constant 1 and let this gate be $d_{t,f}$.

t introduces vertex v. Let s be the child of t.

If $f(v) = 1$, then the formula $\varphi_{t,f}$ is unsatisfiable since v is isolated in G_t. We add to D a gate labeled with constant 0 and let this gate be $d_{t,f}$.

If $f(v) = 0$, then $\varphi_{t,f} = \varphi_{s,f|_{X_s}}$, since adding zero-charged isolated vertex does not change Tseitin formula. We do not add any new nodes in D and define $d_{t,f} := d_{s,f|_{X_s}}$.

t forgets vertex u. Let s be the child of t. $G_t = G_s$, so $\varphi_{t,f} = \varphi_{s,f \sqcup c|_{\{u\}}}$. We do not add any new nodes in D and define $d_{t,f} := d_{s,f \sqcup c|_{\{u\}}}$.

t introduces edge $e = (v, u)$. Let s be the child of t. By Lemma 2.14, the result of the substitution $x_e := b$ to $\varphi_{t,f}$ for $b \in \{0,1\}$ is $\varphi_{s,f+b\cdot\mathbf{1}_v+b\cdot\mathbf{1}_u}$. We add to D at most five gates and build a subcircuit $d_{t,f} := (\neg x_e \wedge d_{s,f}) \vee (x_e \wedge d_{s,f+\mathbf{1}_v+\mathbf{1}_u})$. Notice that $d_{t,f}$ computes $(\neg x_e \wedge \varphi_{s,f}) \vee (x_e \wedge \varphi_{s,f+\mathbf{1}_v+\mathbf{1}_u})$, hence, it computes $\varphi_{t,f}$. Observe that new $\wedge$-gates are decomposable since e is not in G_s.

t is a join node. Let s_1, s_2 be the children of t. $V_{s_1} \cap V_{s_2} = X_t$ since every $v \in V$ is forgotten exactly once in T. Since each edge is introduced in T only once, E_{s_1} and E_{s_2} are disjoint. We add to D $2^{|X_t|+1}$ nodes and build a subcircuit $d_{t,f} := \bigvee_{g : X_t \to \{0,1\}} d_{s_1,g} \wedge d_{s_2,f+g}$. All new $\wedge$-gates are decomposable since E_{s_1} and E_{s_2} are disjoint. Given that for all $f : X_t \to \{0,1\}$, $d_{s_i,f}$ computes $\varphi_{s_i,f}$, it is easy to see that $d_{t,f}$ computes $\varphi_{t,f}$.

Now we estimate the size of constructed DNNF D. For each node $t \in T$ and each function $f : X_t \to \{0,1\}$ we add at most $\max(2^{|X_t|+1}, 5) \leq 2^{|X_t|+3}$ nodes to the DNNF, so in total we have at most $|T| \cdot 2^{2w(T)+3}$ nodes. Since $w(T) = \mathrm{tw}(G)$ and $|T| \leq \mathcal{O}(\mathrm{tw}(G)|V| + |E|)$, the size of the resulting DNNF is $2^{\mathcal{O}(\mathrm{tw}(G))}(\mathrm{tw}(G)|V| + |E|) = 2^{\mathcal{O}(\mathrm{tw}(G))}|E|$. ◀