Electronic Colloquium on Computational Complexity, Report No. 128 (2024)

Lifting to regular resolution over parities via games

Yaroslav Alekseev*[1] and Dmitry Itsykson†[2,3]

[1]Technion – Israel Institute of Technology, Haifa, Israel
[2]Ben-Gurion University of the Negev, Beer Sheva, Israel
[3]On leave from Steklov Institute of Mathematics at St. Petersburg

August 27, 2024

Abstract

The propositional proof system resolution over parities (Res($\oplus$)) combines resolution and the linear algebra over $\mathbb{F}_2$. It is a challenging open question to prove a superpolynomial lower bound on the proof size in this system. For many years, superpolynomial lower bounds were known only in tree-like cases. Recently, Efremenko, Garlik, and Itsykson [12] proved an exponential lower bound for regular Res($\oplus$) that is strictly stronger than tree-like Res($\oplus$). Bhattacharya, Chattopadhyay, and Dvorak [7] have recently shown that regular Res($\oplus$) can not polynomially simulate resolution. The proof is based on add-hock lifting; the authors of [7] posed an open research direction to develop a universal lifting technique for regular resolution over parities.

We develop a lifting technique to regular Res($\oplus$) refutation complexity that can be applied to various formulas. Our approach does not explicitly transform proofs but lifts strategies in games characterizing corresponding proof complexity measures. To lift strategies of resolution games to strategies of Res($\oplus$) games, we use the notion of closure introduced by Efremenko, Garlik, and Itsykson [12]; it turns out that this notion can be perfectly combined with a stifling gadget lifting. At first, we demonstrate our approach by giving an elementary and beautiful lifting theorem showing that if a CNF formula ϕ requires resolution width w, then ϕ lifted by 1-stifling gadget (e.g., Maj_3) requires Res($\oplus$)-rank at least w.

Then we prove the lifting theorems for formulas that have good enough Delayer's strategies in the advanced Prover-Delayer games; such formula lifted by a 2-stifling gadget (e.g., Maj_5) requires regular Res($\oplus$) refutations of exponential size. Using this result, we show that lifted Tseitin formulas are hard for regular Res($\oplus$) (thus resolving the open question raised by [7]). We also show that for every formula with a large resolution depth, we can apply mixing and constant-size lifting such that the final formula will require exponential size regular Res($\oplus$) refutations. Finally, we give an alternative and improved separation between resolution and regular Res($\oplus$): we construct an n-variable formula that has resolution refutation of size poly(n) but requires regular Res($\oplus$) refutation of size $2^{\Omega(\sqrt{n})}$.

1 Introduction

Propositional proof systems are used to certify that given Boolean formulas are unsatisfiable. Cook and Rekhow [10] noticed that NP $\neq$ coNP implied that for every propositional proof system, there

*e-mail: tolstreg@gmail.com
†e-mail: dmitrits@gmail.com. Supported by European Research Council Grant No. 949707.

ISSN 1433-8092

is a family of hard formulas that require superpolynomial proof sizes. However, currently, we cannot prove superpolynomial proof-size lower bounds for many particular proof systems.

One of the long-standing open questions in proof complexity is proving superpolynomial lower bounds on the size of derivations in Frege proof systems. Proof lines in Frege systems are Boolean formulas; each particular Frege system is defined by a sound and implicationally complete set of rules. Currently, we only know how to prove Frege lower bounds in bounded-depth cases where proof lines are restricted to be bounded-depth formulas over $\neg, \vee$ and $\wedge$ (see [1], for example). The techniques used to prove those lower bounds are quite similar to techniques used in bounded-depth circuits lower bounds. So, it was conjectured that techniques used by Razborov and Smolenski [24, 25] to prove a lower bound for constant depth circuits built up from $\neg$, $\vee$, $\wedge$, and Mod_p gates can be extended to bounded-depth Frege operating with formulas using $\neg$, $\vee$, $\wedge$ and Mod_p gates (denoted $\mathrm{AC}^0[\mathrm{p}]$-Frege). However, proving lower bounds for $\mathrm{AC}^0[\mathrm{p}]$-Frege is still open for all values of $p > 1$.

The weakest subsystem of $\mathrm{AC}^0[2]$-Frege for which we still do not know superpolynomial lower bounds is resolution over parities (Res($\oplus$)). The proof lines in this proof system are disjunctions of linear equations over $\mathbb{F}_2$, called linear clauses. A Res($\oplus$) refutation of an unsatisfiable CNF formula φ is a sequence of linear clauses $C_1, C_2, \ldots, C_s$ such that (1) C_s is the empty clause (i.e. identically false); (2) for every i, C_i is either a clause of φ or is obtained from C_j and C_k with $j, k < i$ by the resolution rule, or is obtained from C_j with $j < i$ by the weakening rule. The resolution rule allows to derive the clause $C \vee D$ from clauses $C \vee (f = 0)$ and $D \vee (f = 1)$, where f is a linear form. The weakening rule allows us to derive D from C if C semantically implies D. Recently, this proof system received a lot of attention from different researchers. Here are some of them.

Tree-like lower bounds. There are plenty of works establishing tree-like Res($\oplus$) lower bounds for particular formulas using different techniques: Prover-Delayer games [19, 20, 15, 16], reductions from communication complexity [19, 20, 18, 21], reductions from polynomial calculus degree [13].

Chattopadhyay et al. [8] proved that resolution depth can be lifted by stifling gadgets to tree-like Res($\oplus$) size. Independently, Beame and Koroth [6] got similar results. Resolution depth of an usatisfiable CNF formula φ equals the query complexity of finding a clause of φ that is falsified by the given assignment. Proving lower bound on resolution depth is relatively easy; so this lifting theorem gives us a relatively easy way to prove tree-like Res($\oplus$) lower bounds for many formulas.

Regular Res($\oplus$) lower bounds for Binary Pigeonhole Principle. Recently, Efremenko, Garlik and Itsykson [12] proved the first exponential lower bounds on the size of regular (bottom-regular) Res($\oplus$) refutations. Regular Res($\oplus$) is a fragment of Res($\oplus$) in which resolving linear clauses C_1 and C_2 on a linear form f is permitted only if, for both $i \in \{1, 2\}$, the linear form f does not lie within the linear span of all linear forms that were used in resolution rules during the derivation of C_i. Regular Res($\oplus$) is known to be strictly stronger than tree-like Res($\oplus$).

A formula that was shown by [12] to be hard for regular Res($\oplus$) is the Binary Pigeonhole Principle (BPHP). The key technical tool for proving this lower bound is the notion of *closure*; given a $\mathbb{F}_2$-linear system in variables of BPHP, the closure is roughly speaking the set of pigeons on which this linear system is actually talking about.

Regular Res($\oplus$) and resolution separation. Bhattacharya, Chattopadhyay, and Dvorák [7] have recently showen that specific CNF formulas require an exponential size refutation in regu-

lar Res($\oplus$) but admits polynomial size refutation in resolution. This result heavily utilizes the techniques from [12] and the techniques from lifting literature. However, unlike [8], this result is formula-specific and does not immediately provide a complexity measure that can be lifted to regular Res($\oplus$) size. The possibility of general lifting was left as an open question.

1.1 Our contributions

Our work was inspired by the paper of Bhattacharya, Chattopadhyay, and Dvorák [7]. The main goal is to design a general lifting for regular Res($\oplus$). We want to find a relatively simple proof-complexity measure of the formula, from which one may lift lower bounds to the size of regular Res($\oplus$) refutations. In particular, we want to use our lifting theorem to resolve open questions from [7] about lifted Tseitin and random $O(1)$-CNF formulas. Also, we want to use this lifting to simplify and improve the existing separation between regular Res($\oplus$) and Resolution.

1.1.1 Demonstration of our approach: lifting from resolution width to Res($\oplus$) rank.

At first, we demonstrate our approach with a beautiful example of lifting theorem from resolution width to Res($\oplus$) rank.

Consider a CNF formula $\Phi(y_1, y_2, \ldots, y_m)$. Let $g : \{0,1\}^\ell \to \{0,1\}$ be a gadget. Consider a lifted formula $\Phi \circ g$ that is obtained from Φ by applying the substitutions $y_i := g(x_{i,1}, x_{i,2}, \ldots, x_{i,\ell})$ for all $i \in [m]$ and then converting the resulting formula in CNF. Let $X = \{x_{i,j} \mid i \in [m], j \in [\ell]\}$ be the set of variables of $\Phi \circ g$; we refer to variables of X as lifted variables and to variables $Y = \{y_1, y_2, \ldots, y_m\}$ as unlifted variables.

A gadget g is called k-stifling if for every $a \in \{0,1\}$ and every $\ell - k$ variables of g, we can fix them such that regardless of the value of the rest k variables, the value of the gadget will be fixed to a. It is easy to see that the majority function $Maj_{2k+1} : \{0,1\}^{2k+1} \to \{0,1\}$ is k-stifling for every k.

We show that the resolution width is lifted to the rank of Res($\oplus$) refutations by a 1-stifling gadget.

Theorem 1.1 (Theorem 3.1)**.** *Let $g\colon \{0,1\}^\ell \to \{0,1\}$ be a 1-stifling gadget. Consider a k-CNF formula Φ such that $\Phi \circ g$ has a* Res($\oplus$) *refutation of rank W (i.e., for every linear clause in the refutation the dimension of the set of linear forms in this linear clause is at most W), where $W \geq k$. Then Φ has a Resolution refutation of width at most W (i.e., every clause in the refutation contains at most W variables).*

Below, we highlight the main ideas used in the proof of Theorem 1.1 and our further lifting results.

Lifting of strategies in games. Standard approach in proving lifting theorems from a complexity measure μ_1 in weak proof system Π_1 to a complexity measure μ_2 in a strong proof system Π_2 is an explicit transformation of a Π_2-proof π_2 of a lifted formula $\phi \circ g$ to a Π_1-proof π_1 of the unlifted formula ϕ such that $\mu(\pi_2)$ can be estimated from the above by $\mu(\pi_1)$. Instead of doing this, we use games characterizing lower bounds on μ_1 and μ_2 and show how a strategy proving lower bound on μ_1 can be transformed to a strategy proving lower bound on μ_2 for lifted formulas. The first very easy example of such an approach was proposed by Urquhart [27] who showed that resolution depth can be lifted to tree-like resolution size by $\oplus_2$ gadget by the transformation of Adversary's

strategy in Prover-Adversary games characterizing resolution depth of a formula ϕ to a strategy of Delayer in Prover-Delayer games characterizing tree-like resolution size of the formula $\phi \circ \oplus_2$. We give a much more non-trivial application of this simple idea.

In the proof of Theorem 1.1 we use *Spoiler-Duplicator* games introduced by Atserias and Dalmau [5] characterizing resolution width and the similar games characterizing Res($\oplus$) rank (width) [16] (see Section 3 for definitions of these games). We assume that Φ does not have a resolution refutation of width at most W, then there is a strategy of Duplicator in the $(W+1)$-pebble game. We convert this strategy to a strategy of Duplicator in the $(W+1)$-pebble Res($\oplus$)-game for formula $\Phi \circ g$ getting that $\Phi \circ g$ does not have a Res($\oplus$) refutation of rank W.

To describe the transformation of these strategies, we present a new look at the notion of closure defined in [12] for the binary pigeonhole principle.

A new look at closure. Consider a $\mathbb{F}_2$-matrix A whose columns correspond to lifted variables. We say that a matrix A is *safe*[1] if the linear span of the columns of A has a basis consisting of columns of A such that for every unlifted variable y_i, there is at most one basis element, corresponding to a column $x_{i,j}$ for some j. To solve a linear system with a safe matrix A, we can choose the values of all non-basis variables arbitrarily, and then the values of basis variables will be determined uniquely to satisfy the system. Thus, if a gadget g is 1-stifling and a matrix A is safe, then any satisfiable linear system $Ax = b$ for every assignment of unlifted variables σ, $Ax = b$ has a solution that is consistent with σ.

Now assume that a $\mathbb{F}_2$-matrix A with columns corresponding to lifted variables is not necessarily safe. *The closure* of A is the inclusion minimal set of unlifted variables $I \subseteq Y$ such that if we remove from A all columns corresponding to I, we get a safe matrix. A linear system $Ax = b$ can restrict values of unlifted variables only from the closure of A; all other unlifted variables in the solution of $Ax = b$ can be chosen arbitrarily. Roughly speaking, the closure of the matrix is the set of unlifted variables that this matrix is speaking about. It is known that the closure of A is unique, and its size does not exceed A's rank. So, the closure gives us a mapping from linear systems in lifted variables to sets of unlifted variables. We use this mapping directly to transform winning strategies in different Spoiler-Duplicator games (see Section 3 for details).

1.1.2 Lifting to regular resolution size

We define several games on Boolean formulas; we will lift strategies in these games to regular Res($\oplus$) proofs. We start with the first game that is convenient for this lifting. Later, we define simpler games and lift their strategies to strategies in the first game.

Advanced games of Prover and Delayer. Let an unsatisfiable CNF formula formulas Φ be represented in the form of $\bigwedge_{v \in V} \phi_v$, where each ϕ_v is a CNF formula, in which each clause consists of the same set of variables.

A partial assignment ρ is called *q-correct* for Φ if for every set $U \subseteq V$ such that $|\mathrm{Vars}(\bigwedge_{v \in U} \phi_v)| < |\mathrm{Vars}(\Phi)| - q$, ρ can be extended to an assignment satisfying $\bigwedge_{v \in U} \phi_v$, where $\mathrm{Vars}(\phi)$ denotes the set of variables that appear in ϕ.

[1] Here, we give slightly simplified notions of safe matrices and closure of the matrices. In Section 2.4, we define the notion of a safe set of linear forms over lifted variables; there is the following correspondence: the set of linear forms is safe if its coefficient matrix is safe. In Section 2.5, we define the closure of the set of linear forms over lifted variables, and again, the closure of the set of linear forms equals the closure of its coefficient matrix.

Let us define an advanced (Φ, q)-game of Prover and Delayer. On every move, Prover chooses a variable x, and Delayer has two options:

- Delayer can earn a *white* coin and reports $*$. Then Prover chooses the Boolean value of x.
- Delayer can earn a *white* coin and pay a *black* coin to choose the Boolean value of x by himself.

The game ends when the current partial assignment is not q-correct for the formula Φ.

Delayer's strategy is called *linearly described* if Delayer can see only the set of requested variables and he does not know the values of the variables chosen by Prover and even by himself before. When he chooses a value by himself, he chooses a $\mathbb{F}_2$-affine function that is applied to the values of previous variables, and the result of this function is used as a value chosen by Delayer.

Theorem 1.2 (Theorem 4.6)**.** *Let Φ be an unsatisfiable CNF formula. Assume that Delayer has a linearly described strategy in the (Φ, q)-game that guarantees him to earn t white coins while paying at most c black coins. Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 2-stifling gadget. Then the size of any regular* Res($\oplus$) *refutation of $\Phi \circ g$ is at least $2^{t-q\ell-c(\ell-1)}$.*

The general plan of the proof of this theorem is the same as was used in [12] and then in [7]. We consider a random walk in the refutation graph corresponding to a random assignment σ: the walk starts in the empty clause, and on each step, we go from the clause to its premise falsified by σ and show that with non-negligible probability, the walk goes through a linear clause with many linearly independent linear forms. But such a clause can be falsified by σ with a very small probability. Hence, the proof should contain many such linear clauses. The difference between plans in [12] and [7] is that the first paper uses uniform distribution on assignments while [7] uses non-uniform. As in [12], we use the uniform distribution, but the probability that we reach a clause with many independent linear forms in our case is exponentially small (but not too small), while in [7] and [12] the success probability is a constant. While there is a similarity between our proof and the proof from [12], our result is more general, so the proof requires more careful analysis and more profound arguments.

This theorem allows us to answer an open question from [7] and prove the lower bound for lifted Tseitin formulas:

Theorem 1.3 (Informal restatement of Corollary 5.8)**.** *Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 2-stifling gadget and G be is a good enough constant-degree expander. Then the size of any regular* Res($\oplus$) *refutation of $\mathrm{T}(G,c) \circ g$ is at least $2^{\Omega(n)}$, where $\mathrm{T}(G,c)$ is an unsatisfiable Tseitin formula based on a graph G.*

Lifting from resolution depth to regular Res($\oplus$) **size.** A disadvantage of Theorem 1.2 is that advanced (Φ, q)-games of Prover and Delayer are a bit complicated. We aim to show that strategies from much simpler games can be lifted to strategies in the advanced Prover-Delayer games. We will do simplifications in two steps, and finally, we will be able to start the lifting from strategies in very simple games characterizing resolution depth (see Section 6.2 for the definition of these games).

In the first step, we define the simplified (Φ, q)-games of Prover and Adversary. In this game, there are two players: Prover and Adversary. On every move, Prover chooses a variable x of the formula Φ, and Adversary chooses the 0/1 value of this variable. The game ends when the current partial assignment is not q-correct. For every move, Adversary earns a coin.

Lemma 1.4 (Lemma 6.1)**.** *Assume that there is a strategy of Adversary in the simplified (Φ, q)-game that allows him to earn at least t coins. Let $\oplus_r : \{0,1\}^r \to \{0,1\}$ be the parity function. Then for the advanced $(\Phi \circ \oplus_r, qr)$-game, there is a linearly described strategy of Delayer that guarantees him to earn tr white coins while paying at most t black coins.*

Lemma 1.4 is proved using the same idea that Urquhart used to lift resolution depth to tree-like resolution size [27].

The second step is not a classical lifting since we do not use composition with a gadget but use some modification of the formula that we call *mixing*. Mixing does not change formulas semantically. Alekhnovich et. el. [2] defined the following transformation of CNF formulas (also known as Alehhnovich's trick): every clause C is substituted by $C \vee x$ and $C \vee \neg x$, where x is a random variable. Such transformation helps to separate regular resolution from general resolution. By mixing, we mean similar transformation, but we add to every clause several (constant or logarithmic number of random variables); i.e., we change clause C to $\bigwedge_{\alpha \in \{0,1\}^k} C \vee x_1^{\alpha_1} \vee \ldots \vee x_k^{\alpha_k}$, where x^0 denotes $\neg x$ and x^1 denotes x. Mixing is used to lift games characterizing resolution depth for ϕ to simplified $(\mathrm{mix}(\phi), q)$-games of Prover and Adversary. Namely, mixing helps to achieve q-correctness.

Putting it all together, we get the following theorem.

Theorem 1.5 (Informal restatement of Theorem 6.8)**.** *Let φ be an unsatisfiable k-CNF formula in n variables such that $d_R(\varphi) \geq \alpha n$. Let $\mathrm{mix}(\varphi)$ denotes the mixing of φ, where every clause of φ is mixed with $O(\frac{1}{\alpha^2})$ variables. Then any regular $\mathrm{Res}(\oplus)$ refutation of $\mathrm{mix}(\varphi) \circ g$ has size at least $2^{\alpha n/4 - 1}$, where $g = \oplus_5 \circ Maj_5$ is the composition of parity and majority gadgets.*

Plan of the proof of Theorem 1.5 is the following:

1. We lift a strategy of Adversary in the depth-characterizing game for formula ϕ allowing to earn him αn coins to a strategy of Adversary in the simplified $(\mathrm{mix}(\varphi), \epsilon n)$-game allowing him to earn $\alpha n/2$ coins.

2. We lift the later strategy to the linearly described strategy of Delayer in the advanced $(\mathrm{mix}(\varphi) \circ \oplus_5, \epsilon n)$-game by Lemma 1.4.

3. Get lower bound on size of regular $\mathrm{Res}(\oplus)$ refutation of $\mathrm{mix}(\varphi) \circ \oplus_5 \circ Maj_5$ refutations by Theorem 1.2 using that Maj_5 is a 2-stifling gadget.

By Theorem 1.3, expander-based Tseitin formulas lifted by a constant-size gadget are $O(1)$-CNF formulas with n variables of size $O(n)$ which require regular $\mathrm{Res}(\oplus)$ refutation of size $2^{\Omega(n)}$. This is the best possible lower bound up to a constant in the exponent, and such tight lower bounds for regular $\mathrm{Res}(\oplus)$ were not known before. Theorem 1.5 allows us to construct many formulas given the same tight lower bounds. To do this, we can apply Theorem 1.5 to $O(1)$-CNF formulas with n variables, $O(n)$ clauses and with resolution depth $\Omega(n)$. It is well-known that resolution depth is at least resolution width (see, for example, [27]). Hence, we can, for example, apply Theorem 1.5 to random $O(1)$-CNF formulas with n variables and $O(n)$ clauses that are known to have resolution width $\Omega(n)$ [3].

1.1.3 Improved and simplified separation of regular $\mathrm{Res}(\oplus)$ and resolution

Using ideas from Theorem 1.5, we give an alternative proof of separation between regular $\mathrm{Res}(\oplus)$ and Resolution. Namely, we give the family of formulas with M variable that has $\mathrm{poly}(M)$ size

resolution refutation but requires regular Res($\oplus$) refutations of size $2^{\Omega(M^{1/2})}$. This improves the lower bound $2^{\Omega(M^{1/12})}$ obtained in [7].

Theorem 1.6 (Informal restatement of Corollary 7.3 and Theorem 7.5)**.** *Let G_n be a graph of $n \times n$ grid. Let* $\mathrm{mix}(\mathrm{Peb}(G_n))$ *denote some specific mixing of the pebbling formula based on G_n, where each clause is mixed with $O(\log n)$ variables. Then there exists a constant k and a gadget $g : \{0,1\}^k \to \{0,1\}$ such that* $\mathrm{mix}(\mathrm{Peb}(G_n)) \circ g$ *requires regular* Res($\oplus$) *refutations of size at least $2^{n/4}$ but has resolution refutations of size* $\mathrm{poly}(n)$.

The lower bound is proved very similar to Theorem 1.5. To get a good lower bound, we use the specific strategy of resolution depth games for $\mathrm{Peb}(G_n)$ and define a specific mixing operation that mixes every clause with $O(\log n)$ variables. The upper bound in Theorem 1.6 is very straightforward. By the construction of mixing, $\mathrm{Peb}(G_n)$ can be derived from $\mathrm{mix}(\mathrm{Peb}(G_n))$ in $\mathrm{poly}(n)$ resolution steps. Thus, $\mathrm{mix}(\mathrm{Peb}(G_n))$ has resolution refutation of size $\mathrm{poly}(n)$ and width $O(\log n)$, hence for a constant-size gadget g, $\mathrm{mix}(\mathrm{Peb}(G_n)) \circ g$ has resolution refutation of size $\mathrm{poly}(n)$.

Organisation of the paper. In Section 2, we give the basic definitions and facts, including the definitions of closure and lifting. In Section 3, we demonstrate our lifting technique by proving resolution width to Res($\oplus$) rank lifting theorem. In Section 4, we prove our main lifting theorem from strategies in advanced Prover-Delayer games to the size of regular Res($\oplus$) refutations. Section 5 applies this lifting theorem to Tseitin formulas. In Section 6, we prove the lifting from resolution depth to regular Res($\oplus$) size. In Section 7, we prove the improved separation between regular Res($\oplus$) and Resolution. In Section 8, we formulate open questions.

2 Preliminaries

2.1 Basic notations

For a propositional formula ϕ we denote by $\mathrm{Vars}(\phi)$ the set of all variables mentioned in ϕ.

For a set of vectors U from a vector space V we denote by $<U>$ the linear span of U.

In this paper, all scalars are from the field $\mathbb{F}_2$. Let X be a set of variables that take values in $\mathbb{F}_2$. A linear form in variables from X is a homogeneous linear polynomial over $\mathbb{F}_2$ in variables from X or, in other words, a polynomial $\sum_i^n x_i a_i$, where $x_i \in X$ is a variable and $a_i \in \mathbb{F}_2$ for all $i \in [n]$. A linear equation is an equality $f = a$, where f is a linear form and $a \in \mathbb{F}_2$.

A *linear clause* is a disjunction of linear equations: $\bigvee_{i=1}^t (f_i = a_i)$. Notice that over $\mathbb{F}_2$ a linear clause $\bigvee_{i=1}^t (f_i = a_i)$ may be represented as the negation of a linear system: $\neg \bigwedge_{i=1}^t (f_i = a_i + 1)$.

For a linear clause C we denote by $L(C)$ the set of linear forms that appear in C; i.e. $L\left(\bigvee_{i=1}^t (f_i = a_i)\right) = \{f_1, f_2, \ldots, f_t\}$. The same notation we use for linear systems: if Ψ is a $\mathbb{F}_2$-linear system, $L(\Psi)$ denotes the set of all linear forms from Ψ.

2.2 Resolution over parities

Let φ be an unsatisfiable CNF formula. A refutation of φ in the proof system Res($\oplus$) [20] is a sequence of linear clauses $C_1, C_2, \ldots, C_s$ such that C_s is the empty clause (i.e., identically false) and for every $i \in [s]$ the clause C_i is either a clause of φ or is obtained from previous clauses by one of the following inference rules:

- *Resolution rule* allows us to derive from linear clauses $C \vee (f = a)$ and $D \vee (f = a + 1)$ the linear clause $C \vee D$.

- *Weakening rule* allows us to derive from a linear clause C an arbitrary linear clause D in the variables of φ that semantically follows from C (i.e., any assignment satisfying C also satisfies D).

A resolution refutation of a formula φ is a special case of a Res($\oplus$) refutation, where all linear clauses are ordinary clauses.

Any Res($\oplus$) refutation Π of a CNF formula φ can be represented as a directed acyclic graph G_Π with one source. Each node of G_Π is labeled with a linear clause, the source is labeled with the empty clause, sinks are labeled with clauses of ϕ and every node except sinks has one or two outgoing edges such that (1) if a node labeled with C_1 has two outgoing edges to nodes labeled with C_2 and C_3, then C_1 is the result of the resolution rule applied to C_2 and C_3 and (2) if a node labeled with C_1 has only one outgoing edge to a node labeled with C_2, then C_1 is the result of the weakening rule applied to C_2.

We will use another graph $\tilde{G}_\Pi$ obtained from G_Π by contractions of all edges corresponding to weakening rules. For every node u of $\tilde{G}_\Pi$:

- Let u be the result of merging the nodes $v_1, v_2, \ldots, v_k$ $(k > 1)$ forming a path in G_π such that each of the edges $(v_1, v_2), \ldots, (v_{k-1}, v_k)$ of the path corresponds to an application of the weakening rule. Assume that the nodes $v_1, v_2, \ldots, v_k$ are labeled with $C_1, C_2, \ldots, C_k$, respectively;

- We label u with C_k, the strongest of the clauses.

We call the resulting graph $\tilde{G}_\Pi$ *the refutation graph.* It has the following properties:

- $\tilde{G}_\Pi$ is a directed acyclic graph with one source, and each of its sinks is labeled with a clause of φ;

- every node of $\tilde{G}_\Pi$ except sinks has two outgoing edges, and if a node labeled with C_1 has two outgoing edges to nodes labeled with C_2 and C_3, then C_1 is the result of the resolution rule applied to a weakening of C_2 and a weakening of C_3.

By the *size* of a Res($\oplus$) refutation Π, we mean the number of vertices in its refutation graph $\tilde{G}_\Pi$.

2.3 Res($\oplus$) Refutations as Linear Branching Programs

Let X be a set of variables. A *linear branching program* is a directed acyclic graph with one source; every node except sinks has two outgoing edges; for every non-sink node v there is a linear form f_v in variables from X that is called a *query* at the node v; one edge leaving v is labeled $f_v = 0$ and the other edge is labeled $f_v = 1$. Each sink of the graph is labeled with an element from a set A (the set of answers). Every linear branching program computes a function from $\{0,1\}^X \to A$: a full assignment of variables from X determines the unique path from the source to a sink such that this assignment satisfies all equations labeling the path's edges. The sink label is the result of the function.

For every unsatisfiable CNF formula φ we define a relation Search(φ) that consists of all pairs of (σ, C), where σ is an assignment of the variables of φ and C is a clause of φ falsified by σ. We may think of Search(φ) as a search problem where, given an assignment σ, we have to find C such that $(\sigma, C) \in \text{Search}(\phi)$.

Consider a Res($\oplus$) refutation graph G_Π of a CNF formula φ. We now show that the graph G_Π can be relabeled to turn into a linear branching program with the set of answers equal to the set of clauses of φ. Sinks of G_Π are already labeled with clauses of φ. For every non-sink node v of G_π, there is a linear form f_v that is used in the resolution rule at the node v; f_v will be a query at the node v of the linear branching program. Consider an arbitrary node v_1 of G_Π with outgoing edges to nodes v_2 and v_3 and let us define labels of the edges (v_1, v_2) and (v_1, v_3). Let v_1, v_2 and v_3 be labeled with linear clauses C_1, C_2 and C_3, respectively. Let C_1 be the result of the resolution rule applied to $D_2 \vee (f_{v_1} = a)$ and $D_3 \vee (f_{v_1} = a + 1)$, where $D_2 \vee (f_{v_1} = a)$ is a weakening of C_2 and $D_3 \vee (f_{v_1} = a + 1)$ is a weakening of C_3. We label the edge (v_1, v_2) with the linear equation $f_{v_1} = a + 1$ and the edge (v_1, v_3) with $f_{v_1} = a$.

Lemma 2.1 ([12]). *Consider a* Res($\oplus$) *refutation graph with its edges labeled as in the linear branching program associated with it. Let u and v be two nodes labeled with linear clauses C_u and C_v such that a path p connects u to v. Let Φ_p be the conjunction of the equations labeling the edges of p. Then $\Phi_p \wedge \neg C_u$ implies $\neg C_v$. In particular, for any path from the source of a* Res($\oplus$) *refutation graph to a node v labeled with C_v, the system of linear equations written on the edges of this path implies $\neg C_v$.*

Lemma 2.1 implies that every Res($\oplus$) refutation graph of a formula φ may also be considered a linear branching program solving the search problem Search(φ).

For a node v of a linear branching program, we denote by Post(v) the linear span of all linear forms f such that f is a query at a node on a path from v to a sink.

A Res($\oplus$) refutation is called *bottom-regular*, or just *regular*, if for every edge (v, w) in the associated linear branching program $f_v \notin \text{Post}(w)$, where f_v is the query at v.

Lemma 2.2 ([12]). *Suppose that ϕ is an unsatisfiable CNF formula in n variables, and Π is a regular* Res($\oplus$) *refutation of ϕ. Let G_Π be the refutation graph associated with Π. Then, for every node v in G_Π such that there is a path from the source to v of length d, the dimension of* Post(v) *is at most $n - d$.*

2.4 Safe and dangerous sets of linear forms

We consider the set of propositional variables $X = \{x_{i,j} \mid i \in [m], j \in [\ell]\}$. The variables from X are divided into m blocks by the value of the first index. The variables $x_{i,1}, x_{i,2}, \ldots, x_{i,\ell}$ form the ith block, for $i \in [m]$.

Consider sets of linear forms using variables from X over the field $\mathbb{F}_2$. The *support* of a linear form $f = x_{i_1,j_1} + x_{i_2,j_2} + \cdots + x_{i_k,j_k}$ is the set $\{i_1, i_2, \ldots, i_k\}$ of blocks of variables that appear in f with non-zero coefficients. We denote the support by supp(f). The support of a set of linear forms F is the union of the supports of all linear forms in this set. We denote it by supp(F). We say that a linearly independent set of linear forms F is *dangerous* if $|F| > |\text{supp}(F)|$. We say that a set of linear forms F is *safe* if $<F>$ does not contain a dangerous set. If F is linearly dependent but $<F>$ contains a dangerous set, instead of saying that F is dangerous, we say it is not safe.

Every linear form corresponds to a vector of its coefficients indexed by the variables from the set X. Given a list of linear forms $f_1, f_2, \ldots, f_k$, one may consider their coefficient matrix of size $k \times |X|$ in which the i-th row coincides with the coefficient vector of f_i.

Theorem 2.3 ([12]). *Let $f_1, f_2, \ldots, f_k$ be linearly independent linear forms and let M be their coefficient matrix. Then, the following conditions are equivalent.*

(1) The set of linear forms $f_1, f_2, \ldots, f_k$ is safe.

(2) One can choose k blocks and one variable from each of these blocks such that the columns of M corresponding to the k chosen variables are linearly independent.

2.5 Closure

Let $S \subseteq [m]$ be a set of blocks; for a linear form f we denote by $f[\backslash S]$ a linear form obtained from f by substituting 0 for all variables with support in S.For a set of linear forms F we will use the notation $F[\backslash S] = \{f[\backslash S] \mid f \in F\}$.

A *closure* of a set of linear forms F is any inclusion-wise minimal set $S \subseteq [m]$ such that $F[\backslash S]$ is safe.

Lemma 2.4 (Uniqueness [12]). *For any F, its closure is unique.*

We denote the closure of F by $\mathrm{Cl}(F)$.

Lemma 2.5 (Monotonicity [12]). *If $F_1 \subseteq F_2$, then $\mathrm{Cl}(F_1) \subseteq \mathrm{Cl}(F_2)$.*

Lemma 2.6 (Span invariance [12]). $\mathrm{Cl}(F) = \mathrm{Cl}(<F>)$.

Lemma 2.7 (Size bound [12]). $|\mathrm{Cl}(F)| + \dim <F[\setminus \mathrm{Cl}(F)]> \leq \dim <F>$, *and hence* $|\mathrm{Cl}(F)| \leq \dim <F>$.

2.6 Lifting of formulas via gadget

For every CNF formula Φ with variables $Y = \{y_1, y_2, \ldots, y_m\}$ and every Boolean function $g : \{0,1\}^\ell \to \{0,1\}$ we define a CNF formula $\Phi \circ g$ with variables $X = \{x_{i,j} \mid i \in [m], j \in [\ell]\}$ representing $\Phi(g(x_{1,1}, x_{1,2}, \ldots, x_{1,\ell}), g(x_{2,1}, x_{2,2}, \ldots, x_{2,\ell}), \ldots, g(x_{m,1}, x_{m,2}, \ldots, x_{m,\ell}))$ (i.e. we substitute to every variable of Φ the function g applied to ℓ fresh variables). Let $\Phi = \bigwedge_{i \in I} C_i$, where C_i is a clause for all $i \in I$. For every $i \in [m]$ we denote by $y_i \circ g$ the cannonical CNF formula representing $g(x_{i,1}, x_{i,2}, \ldots, x_{i,\ell})$ which has ℓ variables in every clause and by $(\neg y_i) \circ g$ the cannonical CNF formula representing $\neg g(x_{i,1}, x_{i,2}, \ldots, x_{i,\ell})$ which has ℓ variables in every clause. Let $C_i = l_{i,1} \vee l_{i,2} \vee \cdots \vee l_{i,n_i}$, where l_i is a literal. Then we denote by $C_i \circ g$ a CNF formula that represents $l_{i,1} \circ g \vee l_{i,2} \circ g \vee \cdots \vee l_{i,n_i} \circ g$ as follows: $C_i \circ g$ consists of all clauses of the form $D_1 \vee D_2 \vee \cdots \vee D_{n_i}$, where D_j is a clause of $l_{i,j} \circ g$ for all $j \in [m_i]$. And $\Phi \circ g := \bigwedge_{i \in I} C_i \circ g$.

Lemma 2.8. *If clause C contains variables $\{y_i\}_{i \in I}$, then every clause $C \circ g$ contains variables $\{x_{i,j} \mid i \in I, j \in [\ell]\}$.*

Proof. The definition straightforwardly implies the lemma. □

We refer to $\Phi \circ g$ as a formula Φ *lifted with a gadget* g. We refer to the set $Y = \{y_1, y_2, \dots, y_m\}$ as a set of *unlifted* variables and to the set $X = \{x_{i,j} \mid i \in [m], j \in [\ell]\}$ as a set of *lifted* variables.

Sometimes, we will identify subsets of $[m]$ with corresponding subsets of Y. It is especially convenient to use such correspondence for the notions of support and closure. So, we will assume that the support and the closure of the set of linear forms over lifted variables is the set of unlifted variables.

A partial assignment ρ to the set of variables X is called block-respectful if, for every i, ρ either assigns values to all variables with support i or does not assign values to any of them.

Suppose that ρ is a block-respectful partial assignment. Then we define by $\hat{\rho}$ the partial assignment on the set of variables Y such that $\hat{\rho}(y_i) = g(\rho(x_{i,1}, x_{i,2}, \dots, x_{i,\ell}))$ (here we assume that if the right-hand side is undefined, then the left-hand side is also undefined).

Let $k < \ell$. A gadget (i.e. Boolean function) $g : \{0,1\}^\ell \to \{0,1\}$ is called *k-stifling* [8] if for every $A \subset [\ell]$ of size k for every $c \in \{0,1\}$ there exists $a \in \{0,1\}^\ell$ such that for every $b \in \{0,1\}^\ell$ if a and b agree on set of indices $[\ell] \setminus A$, then $g(b) = c$.

Lemma 2.9. *Let Ψ be a satisfiable linear system in the lifted variables X and $L(\Psi)$ be safe. Let $g : \{0,1\}^\ell \to \{0,1\}$ be an 1-stifling gadget. Then for any full assignment σ to the unlifted variables Y there exists a full assignment τ to the lifted variables X such that τ satisfies Ψ and $\hat{\tau} = \sigma$.*

Proof. W.l.o.g. assume that all equations of Ψ are linearly independent.

By Theorem 2.3, the span of columns of the coefficient matrix of Ψ contains a basis that contains at most one column for every block. Let $Z \subseteq X$ be the set of variables corresponding to this basis. Using a 1-stifling property of g we can construct the full assignment τ_0 to variables $X \setminus Z$ such that for every assignment ρ extending τ_0 to the set of variables X, for every $i \in [m]$, $g(\rho(x_{i,1}), \rho(x_{i,2}), \dots, \rho(x_{i,\ell})) = \sigma(y_i)$. Since Z corresponds to the basis of the span of the columns of the linear system Ψ, τ_0 can be extended to an assignment τ that satisfies Ψ. By the construction, $\hat{\tau} = \sigma$. □

We can prove the following immediate corollary of this lemma. Informally, the next lemma states that any linear system over the lifted variables restricts unlifted variables only from the closure.

Lemma 2.10. *Let Ψ be a satisfiable linear system in the lifted variables X. Let $g : \{0,1\}^\ell \to \{0,1\}$ be an 1-stifling gadget. Suppose*

- *σ is a full assignment to lifted variables X satisfying Ψ.*
- *π is a full assignment to unlifted variables Y such that $\pi|_{\mathrm{Cl}(L(\Psi))} = \hat{\sigma}|_{\mathrm{Cl}(L(\Psi))}$.*

Then there exists a full assignment τ to the lifted variables X such that τ satisfies Ψ and

$$\hat{\tau} = \pi.$$

Proof. Let T be the set of all lifted variables with support in $\mathrm{Cl}(L(\Psi))$. Let σ_0 be the restriction of σ to T. The linear system $(\Psi)|_{\sigma_0}$ is satisfiable, and its set of linear forms is safe by the definition of closure. By Lemma 2.9, there exists an assignment γ to the lifted variables $\mathrm{Vars}(\Psi) \setminus T$ that satisfies $(\Psi)|_{\sigma_0}$ and such that $\widehat{(\sigma_0 \cup \gamma)} = \pi$. Thus, we can take $\tau = \sigma_0 \cup \gamma$. □

3 Lifting resolution width to Res(⊕) rank

The *width* of a resolution refutation is the maximal number of literals in any clause of the refutation.

Similarly, we can define the rank of Res(⊕) refutation. The *rank* of a Res(⊕) refutation is the maximal rank of the negation of any linear clause in the refutation.

The goal of this section is to relate these two measures by a lifting theorem:

Theorem 3.1. *Let $g\colon \{0,1\}^\ell \to \{0,1\}$ be a 1-stifling gadget. Consider a k-CNF formula Φ such that $\Phi \circ g$ has a* Res(⊕) *refutation of rank W, where $W \geq k$. Then Φ should have a resolution refutation of width at most W.*

To prove this theorem, we will need to consider Spoiler-Duplicator games.

Spoiler-Duplicator game for resolution. Firstly defined by Atserias and Dalmau [5], the k-pebble game on an unsatisfiable CNF formula Φ proceeds as follows: starting with the empty assignment, on every turn Spoiler has two options:

- If the size of the current assignment is less than k, then Spoiler can ask Duplicator about the value of some variable x from Φ. Then Duplicator chooses the value of x.
- Spoiler can erase one of the variables from the domain of the assignment.

Spoiler wins if the current assignment contradicts one of the clauses of Φ. Duplicator wins if he can answer Spoiler's responses such that Spoiler does not win.

One can formalize this game by defining a Duplicator's winning strategy as follows:

Let Φ be an unsatisfiable CNF formula. We say that the Duplicator wins the k-pebble game on Φ if there is a non-empty family $\mathcal{H}$ of partial truth assignments that do not falsify any clause from Φ such that:

- If $f \in \mathcal{H}$, then $|f| \leq k$.
- If $f \in \mathcal{H}$ and $g \subseteq f$, then $g \in \mathcal{H}$.
- If $f \in \mathcal{H}$, $|f| < k$ and x is a variable, then there is value $a \in \{0,1\}$ such that $f \cup \{x := a\} \in \mathcal{H}$.

The following equivalence establishes the connection between the resolution width and Spoiler-Duplicator games:

Lemma 3.2 ([5])**.** *Let Φ be a k-CNF formula, and $W \geq k$ be an integer number. Then Φ does not have a resolution refutation of width W if and only if a Duplicator's winning strategy exists in $(W+1)$-pebble game on Φ.*

Spoiler-Duplicator game for Res(⊕)**.** Similarly to the games of Atserias and Dalmau, one can define Spoiler-Duplicator k-pebble Res(⊕)-games [16]. This game on an unsatisfiable CNF formula Φ proceeds as follows: starting with the empty system of $\mathbb{F}_2$-linear equations, on every turn, Spoiler has two options:

- If the rank of the current system is less than k, then Spoiler can ask Duplicator about the value of some *linear form* ℓ over the variables from Φ. Then Duplicator can choose the value $a \in \{0,1\}$ of ℓ and add $\ell = a$ to the current linear system.

- Spoiler can change the linear system to any other linear system, which is implied by the current system.

Spoiler wins if the current system contradicts one of the clauses of Φ. Duplicator wins if he can answer Spoiler's responses such that Spoiler does not win.

Similarly to [16], we formally define a Duplicator's strategy in Res($\oplus$)-games: We say that the Duplicator wins the Res($\oplus$) existential k-pebble game on Φ if there is a non-empty family $\mathcal{H}$ of linear systems over $\mathbb{F}_2$ such that:

- For every $F \in \mathcal{H}$ and every clause C in Φ, there exists a solution of F that satisfies C.
- If $F \in \mathcal{H}$, then $\mathrm{rk}(F) \leq k$.
- If $F \in \mathcal{H}$ and F semantically implies G, then $G \in \mathcal{H}$.
- If $F \in \mathcal{H}$ and $\mathrm{rk}(F) \leq k-1$ and f is a linear form, then there is $a \in \mathbb{F}_2$ such that $F \wedge \{f = a\} \in \mathcal{H}$.

Similarly to Lemma 3.2, one can prove the following lemma:

Lemma 3.3 (cf. [16])**.** *If for an unsatisfiable CNF formula Φ Duplicator has a winning strategy $\mathcal{H}$ in a $(W+1)$-pebble* Res($\oplus$)*-game, then Φ does not have a* Res($\oplus$) *refutation of rank at most W.*

Proof. Suppose there is a Res($\oplus$) refutation Π for Φ of rank at most W and a Duplicator's winning strategy $\mathcal{H}$ in the $(W+1)$-pebble Res($\oplus$)-game. Consider a linear branching program associated with Π. Consider the following strategy of Spoiler: it starts at the source labeled with the empty clause, and the current linear system is also empty. Every his move, Spoiler asks for a value of the linear form corresponding to the current node (this move is legal since the rank of the current linear system is at most W) and moves to the node corresponding to the answer of Duplicator and then changes the current linear system to the negation of the clause in the new node (this move is legal by Lemma 2.1). Properties of the Duplicator's strategy imply that this process will never stop since the current linear system does not contradict clauses of Φ. This is a contradiction since there are no infinite paths in the directed acyclic graph. □

Now, using those two lemmas, we can prove the main result of this section.

Proof of Theorem 3.1. Suppose there is no resolution refutation of Φ of width W. Consider a $(W+1)$-winning strategy $\mathcal{H}$ for the resolution Spoiler-Duplicator games for Φ that exists by Lemma 3.2. Consider the following family of linear systems $\tilde{\mathcal{H}}$: it will consist of all $\mathbb{F}_2$-linear systems F over the variables from $\Phi \circ g$ for which the following holds:

- $\mathrm{rk}(F) \leq W+1$.
- There exist $h \in \mathcal{H}$ and a solution σ of F such that $\hat{\sigma}$ coincides with h on $\mathrm{Cl}(L(F))$.

We show that $\tilde{\mathcal{H}}$ satisfies all the properties of $(W+1)$-winning Res($\oplus$)-strategy:

- By definition, for every $F \in \tilde{\mathcal{H}}$, $\mathrm{rk}(F) \leq W+1$.

- Let us show that for every $F \in \tilde{\mathcal{H}}$ and every clause C' from $\Phi \circ g$, there exists a solution of F that satisfies C'. Indeed, consider a clause C from Φ such that C' is a clause from $C \circ g$. By the definition of $\tilde{\mathcal{H}}$, there exists $h \in \mathcal{H}$ such that there is a solution σ of F such that $\hat{\sigma}$ coincides with h on $\mathrm{Cl}(L(F))$. Since $h \in \mathcal{H}$, there is a full assignment π of unlifted variables, which is consistent with h and satisfies C. By Lemma 2.10, there exists an assignment τ of the lifted variables that satisfies F and such that $\hat{\tau} = \pi$. Hence, τ satisfies $C \circ g$ and, thus, τ satisfies C'.

- Let us show that if G is a linear system satisfying and for some $F \in \tilde{\mathcal{H}}$, F semantically implies G, then $G \in \tilde{\mathcal{H}}$. Indeed, since F semantically implies G, $L(G) \subseteq <L(F)>$. Then by Lemmas 2.5 and 2.6, $\mathrm{Cl}(L(G)) \subseteq \mathrm{Cl}(L(F))$. Clear that $\mathrm{rk}(G) \leq \mathrm{rk}(F) \leq W + 1$. Since, $F \in \tilde{\mathcal{H}}$ there exist $h \in \mathcal{H}$ and there is a solution σ of F such that $\hat{\sigma}$ coincides with h on $\mathrm{Cl}(L(F))$. Notice that σ is also a solution of G and $\hat{\sigma}$ coincides with h on $\mathrm{Cl}(L(G))$. Hence G is in $\tilde{\mathcal{H}}$.

- Finally, we need to show that for any $F \in \tilde{\mathcal{H}}$ with $\mathrm{rk}(F) < W + 1$ and for every linear form f, there exists a constant $a \in \mathbb{F}_2$ such that $F \wedge \{f = a\} \in \tilde{\mathcal{H}}$.

  There exist $h \in \mathcal{H}$ and a solution σ of F such that $\hat{\sigma}$ coincides with h on $\mathrm{Cl}(L(F))$. W.l.o.g., assume that the domain of h is precisely $\mathrm{Cl}(L(F))$. By Lemma 2.7, $|(\mathrm{Cl}(L(F) \cup \{f\}))| \leq \mathrm{rk}(F) + 1 \leq W + 1$. By the properties of $\mathcal{H}$ there is $g \in \mathcal{H}$ such that $h \subseteq g$ and g is defined on $\mathrm{Cl}(L(F) \cup \{f\})$; indeed, we can extend h for all variables from $\mathrm{Cl}(L(F) \cup \{f\})) \setminus \mathrm{Cl}(L(F))$ one by one. Again, using Lemma 2.10, we can find a solution τ of F such that $\hat{\tau}$ coincides with g on $\mathrm{Cl}(L(F) \cup \{f\}))$. Let a be a value of linear form f on the solution τ. Then τ clearly satisfies $F \wedge \{f = a\}$. On the other hand, $\hat{\tau}$ coincides with g on $\mathrm{Cl}(L(F) \cup \{f\}))$. Thus $F \wedge \{f = a\} \in \tilde{\mathcal{H}}$.

Since $\tilde{\mathcal{H}}$ is $(W + 1)$-winning Res($\oplus$)-strategy, it is impossible to construct a rank-W Res($\oplus$) refutation for $\Phi \circ g$ by Lemma 3.3.

□

4 Lifting from strategies in advanced Prover-Delayer games to regular resolution over parities

In this section, our goal is to show that if a formula Φ has some good properties and $g : \{0,1\}^\ell \to \{0,1\}$ is a 2-stifling gadget, then $\Phi \circ g$ requires large Res($\oplus$) proofs (for precise statement see Theorem 4.6). The high-level proof plan is as follows:

1. We consider a random full assignment σ of the variables of $\Phi \circ g$ and make several steps in a branching program associated with a regular Res($\oplus$) refutation of $\Phi \circ g$ from the source according to σ. Let C be a clause at the end of the path.

2. We show that with probability p the linear system $\neg C$ has rank at least r.

3. By the construction σ satisfies $\neg C$. Random assignment satisfies a linear system with rank at least r with probability at least 2^{-r}. Hence, the refutation must contain at least $p2^r$ clauses.

The main technical part is the realization of the step 2 of the above plan.

- In Subsection 4.1, we define the notion of a q-corect partial assignment of an unlifted formula Φ. In Theorem 4.1, we show that if a clause C is on a large enough distance from the source of the branching program and $\neg C$ has a solution π such that $\hat{\pi}|_{\mathrm{Cl}(L(C))}$ is q-correct, then $\neg C$ has a large rank.

- In Subsection 4.2, we define the advanced Prover-Delayer games for the formula Φ. In Theorem 4.2 we consider a random walk described in the 1st step of the above plan and show that if Delayer has a good strategy in the game, then with high enough probability $\hat{\sigma}|_{\mathrm{Cl}(L(C))}$ is q-correct.

- In Subsection 4.3 we prove the main result (Theorem 4.6).

4.1 Rank lower bound and q-correct partial assignments

Let Φ be an unsatisfiable CNF formula that can be represented in the form of $\bigwedge_{v\in V}\phi_v$, where ϕ_v is a CNF formula, in which each clause consists of the same set of variables. In the simple case, each ϕ_v contains just one clause.

A partial assignment ρ is called *q-correct* for Φ if for every set $U \subseteq V$ such that $|\mathrm{Vars}(\bigwedge_{v\in U}\phi_v)| < |\mathrm{Vars}(\Phi)| - q$, ρ can be extended to an assignment satisfying $\bigwedge_{v\in U}\phi_v$.

Theorem 4.1. *Let $g\colon \{0,1\}^\ell \to \{0,1\}$ be a 1-stifling gadget. Consider a regular* Res($\oplus$) *refutation of the lifted formula $\Phi\circ g$ and its linear branching program. Let C be a linear clause such that there is a path of length t from the source of the linear branching program to C. Suppose that $\neg C$ has a solution σ such that $\hat{\sigma}|_{\mathrm{Cl}(L(C))}$ is q-correct (here $\hat{\sigma}|_{\mathrm{Cl}(L(C))}$ is the restriction of $\hat{\sigma}$ to $\mathrm{Cl}(L(C))$ which is identified with the set of unlifted variables). Then $\mathrm{rk}(\neg C) \geq t - \ell q$.*

Proof. Consider the linear branching program associated with the Res($\oplus$) refutation of $\Phi\circ g$. Let W consist of all sinks u of the linear branching program such that there is a path from C to u and the conjunction of linear equations labeling the edges of this path is consistent with the linear system $\neg C$ (i.e., the conjunction of the linear system on the path and $\neg C$ is satisfiable). Let A be the set of labels of the nodes from W; A consists of clauses of $\Phi\circ g$. It is easy to see that A semantically implies C. Indeed, consider an assignment σ of the lifted variables that falsifies C. We start a path in the linear branching program from C to a sink such that σ satisfies all equalities along the edges. Let the path end in a sink w labeled with a clause D. By Lemma 2.1, σ falsifies D.

Let $U := \{v \in V \mid \exists C \in A, C \text{ is a clause of } \phi_v \circ g\}$.

Assume that $|\mathrm{Vars}(\bigwedge_{v\in U}\phi_v)| < |\mathrm{Vars}(\Phi)| - q$, then since $\hat{\sigma}|_{\mathrm{Cl}(L(C))}$ is q-correct, there exists τ extending $\hat{\sigma}|_{\mathrm{Cl}(L(C))}$ such that τ satisfies $\bigwedge_{v\in U}\phi_v$. By Lemma 2.10, there exists a full assignment π to lifted variables such that $\hat{\pi} = \tau$ and π satisfies $\neg C$. Then π satisfies $\bigwedge_{v\in U}\phi_v\circ g$, hence, π satisfies all clauses from A. Since C is a semantic implication of A, π satisfies C, this is a contradiction since π satisfies $\neg C$.

Hence, $|\mathrm{Vars}(\bigwedge_{v\in U}\phi_v)| \geq |\mathrm{Vars}(\Phi)| - q$. Since for all $v \in V$, all clauses from ϕ_v have the same set of variables, Lemma 2.8 implies that $|\mathrm{Vars}(A)| = |\mathrm{Vars}(\bigwedge_{v\in U}\phi_v \circ g)|$. Again by Lemma 2.8, $|\mathrm{Vars}(A)| \geq \ell(|\mathrm{Vars}(\Phi)| - q) \geq |\mathrm{Vars}(\Phi\circ g)| - q\ell$.

Consider $W = <L(C) \cup \mathrm{Post}(C)>$. Using regularity, by Lemma 2.2, we get that

$$\dim(\mathrm{Post}(C)) \leq |\mathrm{Vars}(\Phi\circ g)| - t,$$

and thus

$$\dim(W) \le \dim <L(C)> + \dim(\mathrm{Post}(C)) \le \mathrm{rk}(\neg C) + |\mathrm{Vars}(\Phi \circ g)| - t.$$

On the other hand, for every clause $D \in A$, there is a path from C to D such that $\neg C$ is consistent with the system of all equations labeling the path's edges. By Lemma 2.1, all variables that appear in D are linear combinations of $L(C)$ and the linear forms of the equations at the edges of this path from C to D.

Hence,

$$\dim(W) \ge |\mathrm{Vars}(A)| \ge |\mathrm{Vars}(\Phi \circ g)| - q\ell.$$

Combining those two inequalities together, we get

$$\mathrm{rk}(\neg C) \ge t - q\ell.$$

□

4.2 Random walk and Prover-Delayer games

Let Φ be an unsatisfiable CNF formula that can be represented in the form of $\bigwedge_{v \in V} \phi_v$, where ϕ_v is a CNF formula, in which each clause consists of the same set of variables.

We define a game associated with Φ and a natural number q.

An advanced (Φ, q)-game of Prover and Delayer. In this game, there are two players: Prover and Delayer. On every move, Prover chooses a variable $x \in \mathrm{Vars}(\Phi)$, and Delayer has two options:

- Delayer can earn a *white* coin and reports $*$. Then Prover chooses the Boolean value of x.
- Delayer can earn a *white* coin and pay a *black* coin to choose the Boolean value of x by himself.

The game ends when the current partial assignment is not q-correct for the formula Φ.

A strategy of Delayer is a function $f\colon H \times \mathrm{Vars}(\Phi) \to \{0, 1, *\}$, where H is the set of all possible sequences of pairs of queries asked by Prover with answers (so it is the sequences of the form $(x_{i_1} := \alpha_1, x_{i_2} := \alpha_1, \ldots, x_{i_k} := \alpha_k)$). The strategy is utilized in a natural way: on every step, given the sequence of previous queries with answers Q and a last queried variable x, Delayer answers $f(Q)$.

A linearly described strategy in an advanced game. A linearly described strategy is a special case of a strategy of Delayer. A linearly described strategy has the following form: given a sequence of queries $(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$ (without the answers for those queries) and a last queried variable x it outputs either $*$ or a $\mathbb{F}_2$-linear function $h(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$.

Delayer utilizes this strategy in the following way. On every step given a sequence of queries with answers $(x_{i_1} := \alpha_1, x_{i_2} := \alpha_1, \ldots, x_{i_k} := \alpha_k)$ and a last queried variable x:

- If strategy outputs $*$ on $(x_{i_1}, x_{i_2}, \ldots, x_{i_k}, x)$, then Delayer answers $*$.
- If strategy outputs function $h(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$, then Delayer answers $h(\alpha_1, \alpha_2, \ldots, \alpha_k)$. In that case, we say that the value of x is *forced*.

Theorem 4.2. *Assume that in the advanced (Φ, q)-game, Delayer has a linearly described strategy that guarantees him to earn t white coins while paying at most c black coins, where $t+q < |\mathrm{Vars}(\Phi)|$. Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 2-stifling gadget. Consider a $\mathrm{Res}(\oplus)$ refutation of $\Phi \circ g$ and the linear branching program associated with it. Consider a random full assignment σ for $\mathrm{Vars}(\Phi \circ g)$ and make t steps in the linear branching program from the source according to σ (if we reach a sink earlier than in t steps, we stay there); let we stop in a node labeled with a linear clause C. Then $\hat{\sigma}|_{\mathrm{Cl}(L(C))}$ is q-correct for Φ with probability at least $2^{-c(\ell-1)}$.*

We start with an informal proof plan of Theorem 4.2.

1. Let Ψ_i denote the linear system corresponding to the first i edges of the path in a linear branching program defined by σ. By the properties of linear branching program $\Psi_t \models \neg C$ and, hence, $L(C) \subseteq <L(\Psi_t)>$ and, thus by Lemmas 2.5 and 2.6, $\mathrm{Cl}(L(C)) \subseteq \mathrm{Cl}(L(\Psi_t))$. Thus, $\Pr[\hat{\sigma}|_{\mathrm{Cl}(L(C))}$ is q-correct for $\Phi] \geq \Pr[\hat{\sigma}|_{\mathrm{Cl}(L(\Psi_t))}$ is q-correct for $\Phi]$. Let us denote the latter probability by P^*. We will prove that $P^* \geq 2^{-c(\ell-1)}$.

2. To estimate P^* w.l.o.g. we may assume that the linear branching program is a tree (i.e., parity decision tree). We can convert a linear branching program to a tree in a standard way by repeating the nodes at the same distance from the source.

3. We consider the following metaphor: we assume that all full Boolean assignments of $\Phi \circ g$ are grains of sand. Initially, we put all the sand in the root of the parity decision tree. Each round, every grain of sand in an interior node moves to a child of the current node corresponding to the equation on the edge going to this child. A grain of sand (i.e., full assignment) τ disappears from the node v (and does not move to children in the tree) if $\hat{\tau}|_{\mathrm{Cl}(L(F_v))}$ is not consistent with Delayer's strategy, where F_v is the system of equations written on the path from the root to v. It is easy to see that P^* equals the fraction of sand still in the tree in t steps.

4. The following lemma allows us to estimate the fraction of sand still in the tree in t steps.

Lemma 4.3. *Consider a binary tree with root r and a set of leaves L. We associate every node v except leaves with a number $p_v \neq 0$. For every node v of the tree, there is a number n_v such that if u and w are children of v, then $n_v p_v = (n_u + n_w)$. Let for every leaf l the unique path from the root to l be denoted $\pi_l = (s_1 = r, s_2, \ldots, s_t = l)$; let us denote $p(\pi_l) = \prod_{i=1}^{t-1} p_{s_i}$. Then $n_r = \sum_{l \in L} n_l \frac{1}{p(\pi_l)}$.*

Proof. Induction on the number of leaves in the tree. □

In Lemma 4.3, n_v is the amount of sand that was in the node v. And p_v is the fraction of sand that does not disappear when the sand moves from v to its children.

5. $P^* = \frac{1}{n_r} \sum_{l \in L} n_l$, where L consists of the nodes on distance t and the leaves on distance at most t. A system of linear equations on the path to a leaf of the tree contradicts a clause of $\Phi \circ g$. Hence, there is no sand in the leaves, and we can assume that L does not contain leaves. To estimate the probability P^*, it is sufficient to bound from below $p(\pi_l)$ for some node $l \in L$. If we denote by $s_1 = r, s_2, \ldots, s_t = l$ the path from the root to l, then $p(\pi_l) = \prod_{i=1}^{t-1} p_{s_i}$. The following lemma estimates p_{s_i} regarding Delayer's strategy.

Lemma 4.4. *Let h be a linear description of Delayer's strategy in the advanced (Φ, q)-game. Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 2-stifling gadget. Let F be a system of linear equations in the lifted variables (i.e. variables of $\Phi \circ g$). Let f be a linear form. Consider some order θ on unlifted variables such that variables from $\mathrm{Cl}(L(F))$ preceded variables from $\mathrm{Cl}(L(F) \cup \{f\}) \setminus \mathrm{Cl}(L(F))$ that preceded all other variables. Let T be the set of solutions τ of F such that $\hat{\tau}|_{\mathrm{Cl}(L(F))}$ is consistent with the strategy*

h if variables appear according θ. Let T' be the set of solutions τ of F such that $\hat{\tau}|_{\mathrm{Cl}(L(F)\cup\{f\})}$ is consistent with the strategy h if variables appear according θ. Then $|T'| \geq |T|2^{-(\ell-1)n}$, where n is the number of variables from $\mathrm{Cl}(L(F)\cup\{f\}) \setminus \mathrm{Cl}(L(F))$ such that Delayer recognizes them as forced according the strategy h if variables appear in the order θ.

We will prove Lemma 4.4 in Subsection 4.4.

Lemma 4.4 and properties of Delayer's strategy imply that $\pi(l) \geq 2^{(\ell-1)c}$ for every $l \in L$. Thus by Lemma 4.3, $P^* \geq 2^{(\ell-1)c}$.

Proof of Theorem 4.2. We denote by Ψ_i the linear system corresponding to the first i edges of the path in the linear branching program defined by σ. By Lemma 2.1, the linear system Ψ_t semantically implies $\neg C$, then $L(C) \in <L(\Psi_t)>$, then by Lemmas 2.5 and 2.6, $\mathrm{Cl}(L(C)) \subseteq \mathrm{Cl}(L(\Psi_t))$. So it is sufficient to prove that with probability at least $2^{-c(\ell-1)}$, $\hat{\sigma}|_{\mathrm{Cl}(L(\Psi_t))}$ is q-correct for Φ.

We convert the linear branching program to a parity decision tree in the standard way by making several copies of nodes and edges with the same labels. Since the end of the path corresponding σ has the same label in the tree and the linear branching program, we can continue reasoning assuming that we walk in the parity decision tree. Let H be the subtree of the parity decision tree that contains vertices on the distance at most t from the root; let r be the root of H.

Let v be a vertex of H. We denote by F_v the system linear equations written on the path from the root to v.

For every v of H we construct an order θ_v of the set of variables $\mathrm{Cl}(L(F_v))$. We construct them by induction from the root to leaves. θ_r is some order on the empty set. If u and w are children of v, then $\theta_u = \theta_w$ and equals to an order that extends θ_v such that all elements of $\mathrm{Cl}(L(F_v))$ preceded to all elements of $\mathrm{Cl}(L(F_u)) \setminus \mathrm{Cl}(L(F_v))$.

For every vertex v of the tree, we define a set T_v consisting of the set of full assignments τ satisfying F_v such that $\hat{\tau}|_{\mathrm{Cl}(L(F_v))}$ is consistent with Delayer's strategy when variables appear in the order θ_v.

Recall that $|\mathrm{Vars}(\Phi \circ g)| = m\ell$. Then, for the root, T_r is the set of all full assignments, thus $|T_r| = 2^{m\ell}$. By Lemma 2.7, for every vertex v of H, $|\mathrm{Cl}(L(F_v))| \leq \dim <L(F_v)> \leq t$, hence for every $\sigma \in T_v$, $\hat{\sigma}|_{\mathrm{Cl}(L(F_v))}$ is q-correct by the properties of Delayer's strategy.

Claim 4.5. *If $T_v \neq \emptyset$, then F_v does not contradict any clause of $\Phi \circ g$.*

Proof. Assume for the sake of contradiction that F_v contradicts a clause of $\Phi \circ g$. Let it be a clause of $C \circ g$, where C is a clause of Φ. Consider some $\sigma \in T_v$, by the remark above, $\hat{\sigma}|_{\mathrm{Cl}(L(F_v))}$ is q-correct. We claim that $\hat{\sigma}|_{\mathrm{Cl}(L(F_v))}$ does not falsify C. Indeed, if $\hat{\sigma}|_{\mathrm{Cl}(L(F_v))}$ falsifies C, then $|\mathrm{Vars}(C)| \leq |\mathrm{Cl}(L(F_v))| \leq t$, but $t+q < |\mathrm{Vars}(\Phi)|$ and this contradicts to q-correctness of $\hat{\sigma}|_{\mathrm{Cl}(L(F_v))}$. Hence, $\hat{\sigma}|_{\mathrm{Cl}(L(F_v))}$ can be extended to a full assignment τ to unlifted variables that satisfies C. By Lemma 2.10, there is a solution π of F_v such that $\hat{\pi} = \tau$. Since τ satisfies C, π satisfies $C \circ g$; a contradiction. □

If a and b are distinct leaves of H, then linear systems F_a and F_b contradict each other. Hence, $T_a \cap T_b = \emptyset$. Thus, to prove the theorem it is sufficient to show that $\sum_{a\in\mathcal{L}} |T_a| \geq 2^{-(l-1)c} \cdot 2^{m\ell}$, where $\mathcal{L}$ denotes the set of leaves of H.

For every vertex v in H with children u and w, we define p_v such that $p_v := \frac{|T_w|+|T_u|}{|T_v|}$ if $|T_v| \neq 0$ and $p_v := 1$, otherwise. Notice that if $|T_v| = 0$, then $|T_u| = |T_w| = 0$, hence the equality $p_v|T_v| = |T_u| + |T_v|$ is always satisfied. Since u and w are children of v, there exists a

linear form f and $\alpha \in \{0,1\}$ such that $F_u = F \wedge (f = \alpha)$ and $F_v = F \wedge (f = 1-\alpha)$. Hence, $\mathrm{Cl}(L(F_u)) = \mathrm{Cl}(L(F_w)) = \mathrm{Cl}(L(F_v) \cup \{f\})$. It is easy to see that $T_u \cup T_v$ is the set of assignments τ satisfying F_v such that $\hat{\tau}|_{\mathrm{Cl}(F_v \cup \{f\})}$ is consistent with Delayer's strategy if variables appear in the order $\theta_u = \theta_w$. By Lemma 4.4 applied to the order θ_u, we get that $p_v \geq 2^{-(\ell-1)k}$, where k is the number of forced variables in $\mathrm{Cl}(L(F_u)) \setminus \mathrm{Cl}(L(F_v))$ according to Delayer's strategy if variables appear in the order θ_u.

Let a be a leaf of H. Consider a path from the root of H to a: $u_1 = r, u_2, \ldots, u_s = a$. As we noticed above, $p_{u_i} \geq 2^{-(\ell-1)k_i}$, where k_i is the number of forced variables in $\mathrm{Cl}(F_{u_{i+1}}) \setminus \mathrm{Cl}(F_{u_i})$ according to Delayer's strategy if variables appear in the order θ_a. By the properties of the strategy, Delayer should spend at most c black coins if he earns at most t. Since $|\mathrm{Cl}(L(F_a))| \leq t$, $\prod_{i=1}^{s-1} p_{u_i} \geq 2^{-(\ell-1)c}$.

Recall that $\mathcal{L}$ is the set of leaves of H. By Lemma 4.3 applied to H, p_v and $n_v := |T_v|$, $\sum_{a \in \mathcal{L}} |T_a| \geq |T_r| 2^{-(\ell-1)c} = 2^{m\ell} 2^{-(\ell-1)c}$. □

4.3 Lower bound for regular Res(⊕)

Theorem 4.6. *Let Φ be an unsatisfiable CNF formula. Assume that in the (Φ, q)-game Delayer has a linearly described strategy that guarantees him to earn t white coins while paying at most c black coins. Let $g : \{0,1\}^\ell \to \{0,1\}$ be 2-stifling gadget. Then the size of any regular* Res(⊕) *refutation of $\Phi \circ g$ is at least $2^{t-q\ell-c(\ell-1)}$.*

Proof. Consider a regular Res(⊕) refutation of $\Phi \circ g$ and the linear branching program associated with it. Consider a random full assignment σ of variables $\Phi \circ g$ and make t steps according to σ in the linear branching program. By Theorem 4.2 with probability $2^{-c(\ell-1)}$ we reach a node labeled with a linear clause C such that the partial assignment $\hat{\sigma}|_{\mathrm{Cl}(L(C))}$ is q-correct. Then by Theorem 4.1, $\mathrm{rk}(\neg C) \geq t - q\ell$. Hence, a random assignment satisfies $\neg C$ with probability at most $2^{-t+q\ell}$. Thus, the refutation consists of at least $2^{t-q\ell-c(\ell-1)}$ linear clauses. □

4.4 Proof of Lemma 4.4

Lemma 4.7. *Let $g_1, g_2, \ldots, g_n$ be 2-stifling gadgets from $\{0,1\}^\ell \to \{0,1\}$. Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be arbitrary Boolean functions from $\{0,1\}^{(\ell-1)n} \to \{0,1\}$ and $\beta_1, \beta_2, \ldots, \beta_n$ be affine functions from $\{0,1\}^{(\ell-1)n} \to \{0,1\}$. Then there exist $r_1, r_2, \ldots, r_n \in \{0,1\}^{\ell-1}$ such that for every $i \in [n]$, $g_i(r_i, \alpha_i(r_1, r_2, \ldots, r_n)) = \beta_i(r_1, r_2, \ldots, r_n)$.*

Proof of Lemma 4.7. We have to prove that the system of n equations has a solution. We introduce variables $r_{i,1}, r_{i,2}, \ldots, r_{i,\ell}$ for bits of r_i. We prove the lemma by induction on the number of equations n.

Assume that there is $i \in [n]$ such that β_i is a constant. Then we can satisfy the ith equation by choosing an appropriate value of r_i by using a 2-stifling property of g. Thus we fix values of variables corresponding to r_i and remove the ith equation. The remaining equations we can satisfy by the induction hypothesis.

Assume that there exists $i \in [n]$ such that $\beta_i(r_1, \ldots, r_n)$ is dependent on $r_{i,j}$ for some $j \in [\ell]$. Then the equation $\beta_i = 0$ expresses $r_{i,j}$ from the other variables. We change all occurrences of $r_{i,j}$ to this expression. Now on the left-hand side, we have g_i applied to $r_{i,k}$ for $k \in [\ell-1] \setminus \{j\}$ and two more complicated positions. After this, we fix values of $r_{i,k}$ for $k \in [\ell-1] \setminus \{j\}$ such that the value of the gadget equals zero regardless of the values of the two remaining positions. This is possible

since g_i is 2-stifling. Thus we eliminate variables r_i and delete the ith equation. The remaining equations can be satisfied by the induction hypothesis.

Let us consider a directed graph with vertices $[m]$. We say that there is an edge from i to j if β_i depends on a variable corresponding to r_j. The graph contains a directed cycle since every vertex has an outgoing edge.

Consider the minimal directed cycle: β_{i_1} depends on the variable r_{i_2,j_2}, β_{i_2} depends on r_{i_3,j_3} etc., β_{i_k} depends on r_{i_1,j_1}. Notice that variables $r_{i_1,j_1}, r_{i_2,j_2}, \ldots, r_{i_l,j_k}$ have exactly one occurrence in $\beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_k}$ since otherwise the cycle can be decreased. Consider the linear system $\beta_{i_1} = 0, \wedge \cdots \wedge \beta_{i_l} = 0$. From this system we can express variables $r_{i_1,j_1}, r_{i_2,j_2}, \ldots, r_{i_l,j_k}$ in other variables using these equations. We substitute these expressions instead of variables. After this substitution the right-hand sides of equations with numbers $i_1, i_2, \ldots, i_k$ will be fixed to 0. For every $j \in [k]$ the left-hand side of i_jth equation contains $\ell - 2$ positions that contain only variables, so we can fix the values of all variables to satisfy the i_jth equation using the 2-stifling property. Thus we eliminate all variables corresponding r_i and delete ith equation for $i \in \{i_1, \ldots, i_k\}$. The remaining equations can be satisfied by the induction hypothesis. □

Proof of Lemma 4.4. If $\mathrm{Cl}(L(F)) = \mathrm{Cl}(L(F) \cup \{f\})$, then the lemma is trivial. So we assume that $\mathrm{Cl}(L(F)) \subsetneq \mathrm{Cl}(L(F) \cup \{f\})$. Let ρ be a solution of F restricted to variables with support in $\mathrm{Cl}(F)$ such that $\hat{\rho}$ is consistent with the strategy h if variables appear in the order θ. Let $T_\rho = \{\tau \in T \mid \tau \text{ is consistent with } \rho\}$. Let $T'_\rho = \{\tau \in T' \mid \tau \text{ is consistent with } \rho\}$. It is sufficient to show that for any ρ, $|T'_\rho| \geq |T_\rho| 2^{-(\ell-1)n}$.

The linear system $F|_\rho$ is satisfiable, and the set of its linear forms is safe by the definition of closure. Hence, by Theorem 2.3, one can choose a basis of the span of the columns of the matrix of Φ_ρ such that there is at most one basis element in every block. Let Z denote the set of variables corresponding to this basis. Every solution of $F|_\rho$ defines an element of T_ρ. Hence, the size of T_ρ equals the number of solutions of $F|_\rho$. The set of solutions of $F|_\rho$ can be constructed as follows: choose arbitrary values of non-Z variables, and then the values of Z-variables are uniquely determined. Let D denote the number of non-Z variables in $F|_\rho$. Then $|T_\rho| = 2^D$.

Let K be the set of unlifted variables from $\mathrm{Cl}(L(F) \cup \{f\}) \setminus \mathrm{Cl}(L(F))$ that are forced in the strategy h where variables appear in the order θ. We will show that if we arbitrarily fix:

- values of non-Z variables such that their support is not in K and
- values of the $x_{i,\ell}$ if $y_i \in K$ and Z does not contain variables with support i,

then we can extend this assignment to an element of T'_ρ. The number of unfixed variables out of Z is exactly $(\ell - 1)n$. Hence we will get the desired inequality $|T'_\rho| \geq 2^{D-(\ell-1)n} = |T_\rho| 2^{-n(\ell-1)}$.

For all blocks that are not in K, we have fixed values of variables out of Z. For every $y_i \in K$ we have to choose values of unfixed variables from $x_{i,1}, x_{i,2}, \ldots, x_{i,\ell}$ such that when we determine values of Z variables, the value of the gadget applied to $x_{i,1}, x_{i,2}, \ldots, x_{i,\ell}$ will be consistent with the strategy h.

We know that $|K| = n$; w.l.o.g. $K = \{y_1, y_2, \ldots, y_n\}$. According to the strategy h values of forced variables are computed by linear functions from the values of the previous variables. W.l.o.g. we assume that the correct values of these variables depend only on unforced variables. Unfortunately, it is possible that when we have fixed all lifted variables except one in the block with support not in K, the value of the gadget is not determined and depends on the value of the last variable. Then, the gadget's value is linearly dependent on the unfixed variable from Z. In

this case, the required value of some variable from K may depend on these unfixed Z-variables, but in this case, the dependence is linear. For every $y_i \in K$ we denote unfixed variables among $x_{i,1}, x_{i,2}, \dots, x_{i,\ell}$ by vector of variables r_i; for all $i \in [n]$, r_i consists of exactly $\ell - 1$ variables. Notice that the values of Z-variables are chosen to satisfy the system $\Phi|_\rho$. Hence, the values of every Z-variable can be computed from $r_1, r_2, \dots, r_\ell$ by an affine function. Thus, we have to satisfy the following system of equations:

$$\begin{aligned}
&g_1(r_1, \alpha_1(r_1, r_2, \dots, r_n)) = \beta_1(r_1, r_2, \dots, r_n)\\
&g_2(r_2, \alpha_2(r_1, r_2, \dots, r_n)) = \beta_2(r_1, r_2, \dots, r_n)\\
&\dots\\
&g_n(r_n, \alpha(r_1, r_2, \dots, r_n)) = \beta_n(r_1, r_2, \dots, r_n),
\end{aligned}$$

where for $i \in [n]$, g_i is a function obtained from g by a variable permutation, α_i and β_i are affine functions; α_i corresponds to either expression of a Z-variable (if there are Z-th variables in ith block) or constant (if there are no Z variables in ith block and we have fixed $x_{i,\ell}$); β_i corresponds the linear dependence between the value of the variable y_i according the strategy h and values of several Z-variables.

This system of equations has a solution by Lemma 4.7. □

5 Lifted Tseitin formulas are hard for regular Res($\oplus$)

In this section, we give an example of the application of Theorem 4.6, namely, we show that lifted Tseitin formulas based on spectral expanders are hard for regular Res($\oplus$).

5.1 Tseitin formulas based on expanders

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 $P_v := \left(\sum_{e \ni v} x_e \equiv c(v) \bmod 2\right)$, where $e \ni v$ means that an edge e is incident to the vertex v. The Tseitin formula $\mathrm{T}(G, c)$ is the conjunction of vertices' parity conditions: $\bigwedge_{v \in V} P_v$. Tseitin formulas are represented in CNF as follows: we represent P_v in CNF in a canonical way for all $v \in V$.

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; thus, we implicitly use the corresponding restriction on the set of vertices.

Lemma 5.1 (Folklore, see e.g. [26])**.** *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.*

Lemma 5.2 (Folklore)**.** *The result of the substitution $x_e := b$ 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.*

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

It is well known that for all $1 > \alpha > 0$ and all large enough constants d there exist natural number n_0 and a family $\{G_n\}_{n=n_0}^{\infty}$ of (n, d, α)-expanders. There are explicit constructions such that G_n can be constructed in $\mathrm{poly}(n)$ time [22]. Also, it is known that a random d-regular graph is an expander with high probability.

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

Lemma 5.3 (Cheeger inequality [9])**.** *Let $G(V, E)$ be an (n, d, α)-expander. Then for all $A \subseteq V$ such that $|A| \leq \frac{n}{2}$ the following inequality holds: $|E(A, V \setminus A)| \geq \frac{1-\alpha}{2} d|A|$.*

Corollary 5.4. *Every (n, d, α)-expander with $0 < \alpha < 1$ is connected.*

Proof. If G is not connected, then we will get a contradiction with Lemma 5.3 if we choose A to be the smallest connected component. □

Lemma 5.5 (Expander mixing lemma [4])**.** *Let $G(V, E)$ be (n, d, α)-expander, $A, B \subseteq V$. Then $\left||E(A, B)| - \frac{d|A||B|}{n}\right| \leq \alpha d\sqrt{|A||B|}$.*

Lemma 5.6 (Lemma 11 from [14])**.** *Every graph that can be obtained by deleting $l \leq \frac{n}{4}$ edges from an algebraic (n, d, α)-expander G contains at most $\frac{2l}{d(1-\alpha)} + 1$ connected components.*

5.2 Lower bound

Theorem 5.7. *Let $\mathrm{T}(G, c)$ be an unsatisfiable Tseitin formula based on a spectral (n, d, α)-expander, where $\alpha < 1/2$ and $d \geq 4$. Let $\beta \leq \frac{1}{4}$ and $t = \beta n$ be a natural number. Then in the $(\mathrm{T}(G, c), \epsilon t)$-game, Delayer has a linearly described strategy that guarantees him to earn t white coins while paying at most $\frac{2}{d(1-\alpha)} t$ black coins, where $\epsilon = \frac{\alpha}{1-\alpha} + \frac{2\beta}{d} \cdot \frac{1}{(1-\alpha)^2}$.*

Proof. Delayer strategy is to maintain the following invariant for the current substitution ρ: the largest connected component of the formula $\mathrm{T}(G, c)|_\rho$ is unsatisfiable, and all other components are satisfiable.

Notice that during the first t moves, the size of the largest connected component of $\mathrm{T}(G, c)|_\rho$ has size at most $n/2$. Indeed, consider the first moment where the largest connected component becomes at most $n/2$. Its size at this moment is at least $n/4$. By Lemma 5.3, by this moment we have to remove at least $\frac{dn}{4} \cdot \frac{1-\alpha}{2}$ edges but it is more than $n/4$ if $\alpha < 1/2$ and $d \geq 4$.

Let A be the set of vertices that do not belong to the maximal connected component of $\mathrm{T}(G, c)|_\rho$. Since every move in the game corresponds to removing the edge in the graph, during the first t steps we have removed at most t edges. Hence, by Lemma 5.5, $t \geq E(A, V \setminus A) \geq d|A|\frac{1-\alpha}{2}$. Thus, $|A| \leq \frac{2t}{d(1-\alpha)}$.

Notice that $|E(A, A)|$ equals two times the number of edges with both ends inside A. Using Lemma 5.5, we can estimate the number of edges inside A as follows: $\frac{1}{2}|E(A, A)| \leq \frac{1}{2}\left(\alpha d|A| + \frac{d|A|^2}{n}\right) = \frac{1}{2}|A|d(\alpha + |A|/n) \leq \frac{t}{1-\alpha}\left(\alpha + \frac{2}{d(1-\alpha)\frac{t}{n}}\right) = \beta n\left(\frac{\alpha}{1-\alpha} + \frac{2\beta}{d} \cdot \frac{1}{(1-\alpha)^2}\right) = \epsilon t$.

To maintain the invariant, Delayer looks at the current graph, and if the request corresponds to a non-bridge edge, Delayer responds $*$. Otherwise, Delayer chooses the answer to satisfy the invariant.

Whenever we remove a bridge, the number of connected components increases by one. Lemma 5.6 implies that during t moves, the number of times when the requested edge is a bridge is at most $\frac{2}{d(1-\alpha)}t$. Hence, hence after t moves, Delayer earns t white coins and pays at most $\frac{2}{d(1-\alpha)}t$ black coins.

Let us explain why the strategy is linearly described. Forced variables correspond to bridges. Consider the moment with the current substitution ρ when Prover asks the value of an edge e, which is a bridge. Let U denote the smaller connected component connecting by e. Delayer takes the value of a bridge such that U is satisfiable. By Lemma 5.1, it means that the sum of charges of all vertices from U should be even. We know the initial sum of charges; if an edge connects two vertices from U, then the substitution of a value to this edge does not reflect on the parity of the sum of charges. So, we must compute the parity of ones substituted to edges that connect U with $V \setminus U$. So, the value of the bridge is the result of an affine function applied to the values of previous variables. □

Corollary 5.8. *Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 2-stifling gadget and G be an (n,d,α)-expander, where $d \geq 6\ell$ and $\alpha \leq \frac{1}{6\ell}$. Then, the size of any regular* Res($\oplus$) *refutation of* $\mathrm{T}(G,c) \circ g$ *is at least* $2^{\Omega(n)}$.

Proof. Let $\beta = \frac{\lfloor n/4 \rfloor}{n}$, clear that $\beta = 1/4$. By Theorem 5.7 and Theorem 4.6, size of any regular Res($\oplus$) refutation of $\mathrm{T}(G,c) \circ g$ is at least $2^{t(1-\epsilon\ell)-t\left(\frac{2}{d(1-\alpha)}\right)(\ell-1)}$, where $\epsilon = \frac{\alpha}{1-\alpha} + \frac{2\beta}{d} \cdot \frac{1}{(1-\alpha)^2}$. It is sufficient to have $\epsilon + \frac{2}{d(1-\alpha)} = \frac{\alpha}{1-\alpha} + \frac{2\beta}{d} \cdot \frac{1}{(1-\alpha)^2} + \frac{2}{d(1-\alpha)} < 1/\ell$.

It is easy to verify that this is true since $d \geq 6\ell$, $\beta \leq \frac{1}{4}$, and $\alpha \leq \frac{1}{6\ell}$:

$$\frac{\alpha}{1-\alpha} + \frac{2\beta}{d} \cdot \frac{1}{(1-\alpha)^2} + \frac{2}{d(1-\alpha)} \leq \frac{1}{6\ell} \cdot \frac{6}{5} + \frac{1}{6\ell} \cdot \frac{1}{2} \cdot \left(\frac{6}{5}\right)^2 + \frac{1}{3\ell} \cdot \frac{6}{5} \leq \frac{1}{5\ell} + \frac{1}{6\ell} + \frac{2}{5\ell} < 1/\ell.$$

□

6 Lifting from resolution depth

In this section, we show how to construct formulas that require large regular Res($\oplus$) refutations based on formulas requiring large resolution depth. In Subsection 6.1, we define simplified games that are very similar to games characterizing resolution depth. We show that the strategy in these simplified games can be converted into the strategy in advanced games for the formula lifted by the parity gadget. In Subsection 6.3, we define a mixing transformation of formulas that does not change the formula semantically but allows us to convert strategies in the games characterizing depth to the simplified games.

6.1 Simplified games

Let Φ be an unsatisfiable CNF formula that can be represented in the form of $\bigwedge_{v \in V} \phi_v$, where ϕ_v is a CNF formula, in which each clause consists of the same set of variables.

We define one more game associated with Φ and natural number q.

A simplified (Φ, q)-game of Prover and Adversary. In this game there are two players: Prover and Adversary. On every move, Prover chooses a variable x of the formula Φ, and Adversary chooses the 0/1 value of this variable. The game ends when the current partial assignment is not q-correct. For every his move Adversary earns a coin.

A *strategy* for the Adversary is a function $f\colon H \times \mathrm{Vars}(\Phi) \to \{0,1\}$, where H is the set of all possible sequences of k queries asked by Delayer in the previous rounds (so it is the sequences of the form $(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$). The strategy is utilized naturally: on every step, given a previous sequence of queries Q of Delayer and a last queried variable x, Adversary answers $f(Q, x)$. Notice that in this definition f *does not depend* on the previous answers of Delayer since Delayer can compute these answers by itself if necessary.

If $g : \{0,1\}^\ell \to \{0,1\}$ is some gadget, then $\Phi \circ g$ can be represented as $\bigwedge_{v \in V}(\phi_v \circ g)$. By Lemma 2.8, all clauses of $\phi_v \circ g$ use the same set of variables.

Lemma 6.1. *Assume that there is a strategy of Adversary in the simplified (Φ, q)-game that allows him to earn at least t coins. Let $\oplus_r : \{0,1\}^r \to \{0,1\}$ be the parity function. Then for the advanced $(\Phi \circ \oplus_r, qr)$-game, there is a linearly described strategy of Delayer that guarantees him to earn tr white coins while paying at most t black coins.*

Proof. Let Φ depend on variables $y_1, y_2, \ldots, y_m$. Then $\Phi \circ \oplus_r$ depends on variables $\{x_{i,j} \mid r \in [m], j \in [r]\}$.

Let us present a linear description f of Delayer's strategy. Let L be an ordered list of already asked variables and $x_{i,j}$ be a new variable.

- If $|L \cap \{x_{i,1}, x_{i,2}, \ldots, x_{i,r}\}| < r - 1$, then $f(L, x_{i,j}) = *$.
- Otherwise, let $L' = \{y_j \mid \{x_{j,1}, x_{j,2}, \ldots, x_{j,r}\} \subseteq L\}$. Let for $y_i \in L$, n_i be the maximal number in the list L of a variable with support i. We introduce an order in L': y_k is less than y_j if $n_k < n_j$. Consider the strategy of Adversary in the simplified game that guarantees him to earn t coins. Assume that Prover chooses variables according to the introduced order in L' and then asks y_i. Let $\alpha \in \{0,1\}$ be the value of y_i according to this strategy. In this case $f(L, x_{i,j}) := \sum_{k \in [r] \setminus j} x_{i,k} + \alpha$.

Let us show that this strategy guarantees Delayer to earn tr white coins. Consider any subset $U \subseteq V$ such that $|\mathrm{Vars}(\bigvee_{u \in U} \phi_u \circ \oplus_r)| < |\mathrm{Vars}(\Phi \circ \oplus_r)| - qr$. By Lemma 2.8, $|\mathrm{Vars}(\bigvee_{u \in U} \phi_U)| < |\mathrm{Vars}(\Phi)| - q$. Assume that Delayer uses the described strategy. Consider some state of the game after at most tr moves. Let $M = \{k \in [m] \mid x_{k,j}$ were requested for all j $\in [r]\}$; $|M| \leq t$. There are at most t such k such that $x_{k,i}$ were asked. For $m \in [k]$, values of $\sum_{i=1}^r x_{k_i}$ are fixed according to the value of y_k to the strategy of Adversary in the simplified game. Hence the current partial assignment in the variables $\{y_k \mid k \in M\}$ can be extended to satisfy $\bigvee_{u \in U} \phi_U$. So we can extend partial assignment in the lifted variables to satisfy $\bigvee_{u \in U}(\phi_u \circ \oplus_r)$.

It is easy to see that for every moment when Delayer pays a black coin, there are at least $(r-1)$ moments when he does not pay. Hence the number of paid black coins is at most t. □

6.2 Resolution depth

Simplified games are similar to games characterizing resolution depth [27].

The depth of a resolution proof is the length of the shortest path between an empty clause and a clause of the refuted formula. The resolution depth of an unsatisfiable CNF formula φ denoted by $d_R(\varphi)$ is the minimal possible depth overall resolution refutations of φ.

The resolution depth of an unsatisfiable CNF formula φ can be characterized by the following game of Prover and Adversary: on every move, Prover asks the value of the variable of φ and Adversary answers and earns a coin. The game ends whenever the current assignment falsifies the clause of φ.

Lemma 6.2 ([27]). *$d_R(\varphi) \geq t$ iff Adversary has a strategy that guarantees him to earn at least t coins.*

The main difference between simplified (Φ, q)-games and games characterizing depth is the condition of the end of the game. In the next subsection, we define the mixing operation for formulas to map formulas with large resolution depth to formulas with good Adversary strategies in simplified games.

6.3 Mixed formulas

6.3.1 Mixers

A bipartite graph $G(X, Y, E)$ is an (n, D, α, ϵ)-mixer if

- $|X| = n$, $|Y| = n$;
- Degrees of all vertices from X are at most D;
- For every $A \subseteq X$ and $B \subseteq Y$ if $|A| \geq \alpha n$ and $|B| \geq \epsilon n$, then there is at least one egde between A and B.

Lemma 6.3. *For every integer n, real $\alpha \in (0; 1)$ and $\epsilon \in (0, 1)$ there exists an (n, D, α, ϵ)-mixer, where $D = O(\frac{1}{\epsilon\alpha})$.*

Proof. We construct a graph with vertices X and Y such that $|X| = n, |Y| = n$ by the random process.

1. Initially, the set of edges E is empty;
2. For every $v \in X$ repeat $D = \lceil K\frac{1}{\alpha\epsilon} \rceil$ times (the value of K will be chosen later):
    3. Choose $u \in Y$ at random;
    4. Add (v, u) to E;

For every $A \subseteq X$ and $B \subseteq Y$ such that $|A| \geq \alpha n$, $|B| \geq \epsilon n$, the probability that there are no edges between A and B is at most $(1 - \epsilon)^{D|A|} \leq (1 - \epsilon)^{Kn/\epsilon} < e^{-Kn}$.

The number of pairs A and B is at most 2^{2n}. Hence by the union bound the probability that there exist such $A \subseteq X$ and $B \subseteq Y$ such that $|A| \geq \alpha n$, $|B| \geq \epsilon n$ and there are no edges between A and B is at most $e^{-Kn}2^{2n}$ that is less than 1 for $K \geq 2$. Hence with positive probability, the constructed graph is an (n, D, α, ϵ)-mixer with $D = O(\frac{1}{\alpha\epsilon})$. □

The explicit constructions of mixers can also be obtained from spectral expanders using the expander mixing lemma (Lemma 5.5).

6.3.2 Mixed formulas

Let C be a clause and Z be a set of propositional variables with no occurrences in C. Let $\mathrm{Clauses}(Z)$ be the set of all $2^{|Z|}$ different clauses, each containing all variables from Z. We denote by $\mathrm{pad}(C, Z)$ the CNF formula $\bigwedge_{D \in \mathrm{Clauses}(Z)} (C \vee D)$.

Lemma 6.4. *There is a resolution derivation of C from $\mathrm{pad}(C, Z)$ of length $2^{|Z|}$.*

Proof. The proof is straightforward by induction on the number of variables in Z. □

Lemma 6.5. *C is semantically equivalent to $\mathrm{pad}(C, Z)$.*

Proof. $\mathrm{pad}(C, Z) = \bigwedge_{D \in \mathrm{Clauses}(Z)} (C \vee D)$ and it is semantically equivalent to $C \vee \bigwedge_{D \in \mathrm{Clauses}(Z)} D$ and the later is semantically equivalent to C since $\bigwedge_{D \in \mathrm{Clauses}(Z)} D$ is identically false. □

Let $\varphi = \bigwedge_{v \in V} C_v$ be a CNF formula from n variables (for every $v \in V$, C_v is a clause) and $G(X, Y, E)$ be a bipartite graph with $|X| = |Y| = n$ and with degrees of all vertices from X at most D. We define a CNF formula $\mathrm{mix}_G(\varphi)$ as follows:

- Let π_1 be a bijection from $\mathrm{Vars}(\varphi) \to X$ and π_2 be a bijection from $Y \to \mathrm{Vars}(\varphi)$.
- $\mathrm{mix}_G(\varphi) = \bigwedge_{v \in V} \psi_v$, where $\psi_v = \mathrm{pad}(C_v, \pi_2(\Gamma(\pi_1(\mathrm{Vars}(C_v)))) \setminus \mathrm{Vars}(C_v))$, where for $A \subseteq X$, $\Gamma(A)$ is the set of neighbors of the set A in the graph G.

Notice that if φ is a k-CNF formula, then $\mathrm{mix}_G(\varphi)$ is a kD-CNF formula. By Lemma 6.5, $\mathrm{mix}_G(\varphi)$ is semantically equivalent to φ.

Lemma 6.6. *If G is (n, D, α, ϵ)-mixer, then if $\mathrm{Vars}(\bigwedge_{v \in V} C_v) \geq \alpha n$, then $\mathrm{Vars}(\bigwedge_{v \in V} \psi_v) \geq (1 - \epsilon)n$.*

Proof. The proof is straightforward. □

6.3.3 Lifting from resolution depth

Lemma 6.7. *Let a CNF formula φ with n variables have a resolution depth at least d; let G be $(n, D, \frac{d}{2n}, \epsilon)$-mixer. Then, in the simplified $(\mathrm{mix}_G(\varphi), \epsilon n)$-game, Adversary has a strategy that guarantees him to earn at least $\lfloor d/2 \rfloor$ coins.*

Proof. Adversary will use his strategy in the game characterizing the resolution depth of the formula ϕ, given by Lemma 6.2. Consider the game after $\lfloor d/2 \rfloor$ moves. Let ρ be the current substitution.

Let $\mathrm{mix}_G(\varphi) = \bigwedge_{v \in V} \psi_v$. Assume that for some $U \subseteq V$, ρ can not be extended to satisfy $\bigwedge_{v \in U} \psi_v$. By Lemma 6.5, $\bigwedge_{v \in U} \psi_v$ is semantically equivalent to $\bigwedge_{v \in U} C_v$. Since ρ is the first part of the strategy in the game characterizing depth, $\mathrm{Vars}(\bigwedge_{v \in U} C_v) \geq \frac{d}{2}$ (otherwise, Prover can just query all the variables from $\mathrm{Vars}(\bigwedge_{v \in U} C_v)$ and end the game in less than $\frac{d}{2}$ steps). Hence, by Lemma 6.6, $\mathrm{Vars}(\bigwedge_{v \in U} \psi_v) \geq (1 - \epsilon)n$. □

Theorem 6.8. *Let φ_n be the family of unsatisfiable k-CNF formulas in n variables such that $d_R(\varphi_n) \geq \alpha n$. Let G be a (n, D, α, ϵ)-mixer, where $\epsilon = \alpha/100$, $D = O(\frac{1}{\alpha^2})$, that exists by Lemma 6.3. Then any regular $\mathrm{Res}(\oplus)$ refutation of $\mathrm{mix}_G(\varphi_n) \circ \oplus_5 \circ Maj_5$ has size at least $2^{\alpha n/4 - 1}$.*

Notice that if in the conditions of Theorem 6.8, k and α are constants, then $\mathrm{mix}_G(\varphi_n)\circ\oplus_5\circ Maj_5$ is $O(k)$-CNF formula.

Proof. By Lemma 6.7, in the simplified $(\mathrm{mix}_G(\varphi_n), \epsilon n)$-game, there is a strategy of Adversary that guarantees him to earn at least $\lfloor \alpha n/2 \rfloor$ coins.

By Lemma 6.1, in the advanced $(\mathrm{mix}_G(\varphi_n) \circ \oplus_5, 5\epsilon n)$ game there is a strategy of Delayer that guarantees him to earn at least $5\lfloor \alpha n/2 \rfloor$ white coins while paying at most $\lfloor \alpha n/2 \rfloor$ black coins.

$Maj_5 : \{0,1\}^5 \to \{0,1\}$ is a 2-stifling gadget; hence, by Theorem 4.6, the size of any regular Res($\oplus$) refutation of $\mathrm{mix}_G(\varphi_n) \circ \oplus_5 \circ Maj_5$ is at least $2^{5\lfloor \alpha n/2 \rfloor - 25\epsilon n - 4\lfloor \alpha n/2 \rfloor} \geq 2^{\alpha n/4 - 1}$. □

7 Regular Res($\oplus$) does not simulate resolution

In this section, we give an alternative and improved separation between regular Res($\oplus$) and Resolution firstly proved by Bhattacharya, Chattopadhyay, and Dvorak [7].

One of the possible ways to do it using our technique is to apply the combination of mixing and lifting from the previous section to pebbling formulas $\mathrm{Peb}(G_n)$ that have $O(n)$ variables and resolution depth $\Omega(n/\log n)$ [27, 23]. The problem is that we need $(n, D, O(1/\log n), O(1/\log n))$ mixers, and for them, $D = O(\log^2 n)$ and the resulting formula will have superpolynomial size. This will imply some separation but not very good. Instead, we will consider Pebbling formulas on the well-structured grid graphs. For such formulas, we can require a much weaker mixing property that allows us to decrease the degree of mixers to $O(\log n)$.

Let $H_n(V_n, E_n)$ be a directed grid graph with the set of vertices $V_n = [n] \times [n]$. The edges are oriented to the left and the bottom or formally

- for $i > 1, j > 1$ the vertex (i, j) has two outgoing edges to $(i-1, j)$ and $(i, j-1)$;
- for $j > 1$, the vertex $(1, j)$ has one outgoing edge to $(1, j-1)$; and for $i > 1$ the vertex $(i, 1)$ has one outgoing edge to $(i-1, 1)$.

We define the Pebbling formula $\mathrm{Peb}(H_n)$ as follows. The set of variables is $\{x_v \mid v \in V_n\}$. The formula $\mathrm{Peb}(H_n)$ is defined to be $\neg x_{1,1} \wedge \bigwedge_{v \in V_n} C_v$, where $C_v = x_v \vee \bigvee_{u \in V_n : (u,v) \in E_n} \neg x_u$.

A bipartite graph $G(X, Y, E)$ is called (n^2, D, ϵ)-grid mixer if

- $|X| = |Y| = n^2$; let σ be a bijection from $V_n \to X$;
- degree of all vertices from X are at most D;
- for all $A \subseteq [n]$ and $B \subseteq [n]$ such that $|A| \geq n/2$ and $|B| \geq n/2$ and all $C \subseteq Y$ such that $|C| \geq \epsilon n$, there is at least one edge between $\sigma(A \times B)$ and C.

Lemma 7.1. *For every integer n and real $\epsilon \in (0,1)$ there exists an (n^2, D, ϵ)-grid mixer, where $D = O(\log n/\epsilon)$.*

Proof. We construct a graph with vertices X and Y such that $|X| = n^2, |Y| = n^2$ by the random process. Let σ be a bijection from $[n] \times [n] \to X$.

1. Initially, the set of edges E is empty;

2. For every $v \in X$ repeat $D = \lceil K \log n \rceil$ times (the value of K will be chosen later):

    3. Choose $u \in Y$ at random;
    4. Add (v, u) to E;

For every $A \subseteq [n]$ and $B \subseteq [n]$ such that $|A| \geq n/2$, $|B| \geq n/2$ and every $C \subseteq Y$ such that $|C| \geq \epsilon n$, the probability that there are no edges between $\sigma(A \times B)$ and C is at most $(1 - \frac{\epsilon}{n})^{Dn^2/4} \leq (1 - \frac{\epsilon}{n})^{\frac{n}{\epsilon} \cdot \epsilon K n \log n/4} < e^{-K\epsilon n \log n/4}$.

The number of pairs A and B is at most 2^{2n}, and the number of different C is at most $\binom{n^2}{\epsilon n} \leq 2^{2\epsilon n \log n}$. Hence by the union bound the probability that there exist such $A \subseteq [n]$ and $B \subseteq [n]$ and $C \subseteq Y$ such that $|A| \geq n/2$, $|B| \geq n/2$, $|C| \geq \epsilon n$ and there are no edges between $\sigma(A \times B)$ and C is at most $e^{-Kn\epsilon \log n/4} 2^{2n + 2n \log n\epsilon}$ that is less than 1 for $K \geq \frac{16}{\epsilon}$. Hence with positive probability, the constructed graph is an (n, D, α, ϵ)-mixer with $D = O\left(\frac{\log n}{\epsilon}\right)$. □

Let $G_{n,\epsilon}$ be an (n^2, D, ϵ)-grid mixer with $D = O\left(\frac{\log(n)}{\epsilon}\right)$. Consider the Pebbling formula $\mathrm{Peb}(H_n) = \neg x_{1,1} \wedge \bigwedge\limits_{v \in V_n} C_v$ and let $\Phi_{n,\epsilon} := \neg x_{1,1} \wedge \mathrm{mix}_G \left(\bigwedge\limits_{v \in V_n} C_v \right)$.

Theorem 7.2. *In the simplified $(\Phi_{n,\epsilon}, \epsilon n)$-game there is a strategy of Adversary that guarantees him to earn at least $n/4$ coins.*

Proof. For every $i \in [n]$ we define the ith *cross* as the set of vertices $\{(a, b) \in V_n \mid a = i \text{ or } b = i\}$.

The top-right part of the ith cross is the set of vertices $\{(a, i), (i, a) \mid a > i\}$.

Let us describe the Adversary's strategy. Adversary has two variables i and p, where i takes values from $[n]$ and p denotes a path in H_n from (i, i) to $(1, 1)$. Initially $i = 1$ and p consists of the only vertex $(1, 1)$. During the game, Adversary maintains the following invariant:

- For every $i' < i$ there were requests to variables from i'th cross.
- There were no requests to variables from the top-right part of the ith cross.
- If a variable x_v was requested, then x_v was assigned to 0 if v belongs the path p and to 1 otherwise.

The strategy of Adversary is as follows. Let Prover ask the value of x_v

- Adversary responds 0 if v belongs p and 1, otherwise;
- If v belongs to top-right part of the i-th cross, then
    - Let j be the number of the minimal cross such that there were no requests to its variables. If there are no such crosses, Adversary gives up. Notice that the invariant guarantees $j > i$.
    - Since v is the first request to the top-right part of the ith cross, one of the following paths does not contain any requests $(j, j), (j-1, j), \ldots, (i, j), (i, j-1), \ldots, (i, i)$ or $(j, j), (j, j-1), \ldots, (j, i), (j-1, i), \ldots, (i, i)$; let p' denotes this path.
    - $i := j$; the new value of p is p' prolonged by the previous value of p.

Let t be an integer number and $t \le n/4$. Consider the moment after t rounds of the game where Adversary follows the described strategy. Notice that every two crosses have exactly two common vertices, hence there are crosses without requests, and Adversary does not give up. Let k be the number of requests made to the first $i-1$ crosses during the first t rounds. Since every two crosses have two common vertices and all crosses with numbers lesser than i contain requests, $k \ge \frac{i-1}{2}$.

Let $A = \{l \mid i \le l \le n$ and there are no requests to variables $x_{l,j}$ for $j \in [n]\}$ and $B = \{l \mid i \le l \le n$ and there are no requests to variables $x_{j,l}$ for $j \in [n]\}$.

$|A| \ge (n-(i-1)) - (t-k) \ge (n-2k) - (t-k) \ge n - 2t \ge n/2$. Analogously, $|B| \ge n/2$.

Let us verify that the current partial assignment is ϵn-correct.

Let $\Phi_{n,\epsilon} = \neg x_{1,1} \wedge \bigwedge\limits_{v \in V_n} \psi_v$, where ψ_v is the result of pad applied to C_v.

Consider some $U \subseteq V_n$. If $A \times B \subseteq U$, then $\mathrm{Vars}(\bigwedge_{v\in U} \psi_v) \ge n^2 - \epsilon n$ by the property of the greed-mixer G. Assume that there is u such that $u \in A \times B$ and $u \notin U$. We will show that the current assignment can be extended to satisfy $\neg x_{1,1} \wedge \bigwedge\limits_{v \in V_n \setminus \{u\}} C_v$. Since C_v and ψ_v are semantically equivalent, we will get that the current assignment can be extended to satisfy $\neg x_{1,1} \wedge \bigwedge\limits_{v \in U} \psi_v$.

Let $u \in A \times B$. We claim that the formula $\neg x_{1,1} \wedge \bigwedge\limits_{v \in V_n \setminus \{u\}} C_v$ can be satisfied by extending the current assignment from the game. Indeed, there is the following path p' from u to (i,i): at first, we decrease the first coordinate to level i and then decrease the second coordinate to level i. Consider an assignment that assigns 0 to vertices of p and p' and 1 to all other variables. By the construction of the strategy, this assignment agrees with the current assignment. It is easy to see that C_u is the only unsatisfied condition from $\mathrm{Peb}(H_n)$. Thus, the current assignment is (ϵn)-correct. □

Corollary 7.3. *The size of any regular* $\mathrm{Res}(\oplus)$ *refutation of* $\Phi_{n,\epsilon} \circ \oplus_r \circ \mathrm{Maj}_\ell$ *is at least* $2^{n/4}$*, where* $\ell = 5, r = 6$ *and* $\epsilon = 1/120$.

Proof. By Theorem 7.2 and Lemma 6.1, in the advanced $(\Phi_{n,\epsilon} \circ \oplus_r, \epsilon rn)$ game there is a strategy of Delayer that guarantees him to earn at least $rn/4$ white coins while paying at most $n/4$ black coins.

Since Maj_l is a 2-stifling gadget for $l \ge 5$, by Theorem 4.6, the size of any regular $\mathrm{Res}(\oplus)$ refutation of $\Phi_n \circ \oplus_r \circ \mathrm{Maj}_\ell$ is at least $2^{\frac{n}{4}r - \epsilon r \ell n - \frac{n}{4}(\ell-1)}$. Let us choose $\ell = 5, r = 6$ and $\epsilon = \frac{1}{4r\ell} = 1/120$, then we get the size lower bound $2^{n/4}$. □

Lemma 7.4 ([20])**.** *Let* $g : \{0,1\}^k \to \{0,1\}$ *be a gadget. If a CNF formula* ϕ *has a resolution refutation of size* S *and width* w*, then the formula* $\phi \circ g$ *has a resolution refutation of size* $S2^{O(wk)}$.

Theorem 7.5. *For every* ϵ, r *and* ℓ*, the formula* $\Phi_{n,\epsilon} \circ \oplus_r \circ \mathrm{Maj}_\ell$ *has a resolution refutation of size* $\mathrm{poly}(n)2^{O(1/\epsilon)}$ *(or just* $\mathrm{poly}(n)$ *if* ϵ *is a constant).*

Proof. It is well known that $\mathrm{Peb}(H_n)$ has a resolution proof of size $O(n^2)$ and width $O(1)$. Then by Lemma 6.4, $\Phi_{n,\epsilon}$ has a resolution refutation of size $\mathrm{poly}(n)2^{O(1/\epsilon)}$ and width $O(\log n)$. Two gadgets may be considered as one gadget from $r\ell$ variables. Then by Lemma 7.4, $\Phi_{n,\epsilon} \circ \oplus_r \circ \mathrm{Maj}_\ell$ has a resolution refutation of size $\mathrm{poly}(n)2^{O(1/\epsilon)}$. □

8 Further research

We consider it interesting to address the following questions connected with our research:

1. Lifting theorem from [8] can also be applied in the Boolean case for parity decision trees. It would be interesting to devise a lifting technique to, for example, strongly read-once branching programs [17] in the Boolean case as well.

2. Lifting was used to prove NP-hardness of automating of algebraic proof systems [11]. Can we prove that regular Res($\oplus$) is NP-hard to automate using the lifting approach from the paper?

3. It is also interesting to prove super polynomial lower bounds in a stronger proof system. We think the next natural step is to consider *weakly regular* Res($\oplus$), where for each path, all resolved linear forms are linearly independent.

Acknowledgments. The authors thank Klim Efremenko, Michal Garlik, Yuval Filmus, and Alexander Knop for fruitful discussions.

References

[1] Miklós Ajtai. The complexity of the pigeonhole principle. *Combinatorica*, 14:417–433, 1988.

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

[3] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 190–199. IEEE Computer Society, 2001.

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

[5] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, 2008. Computational Complexity 2003.

[6] Paul Beame and Sajin Koroth. On Disperser/Lifting Properties of the Index and Inner-Product Functions. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:17, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[7] Sreejata Kishor Bhattacharya, Arkadev Chattopadhyay, and Pavel Dvorák. Exponential separation between powers of regular and general resolution over parities. In Rahul Santhanam, editor, *39th Computational Complexity Conference, CCC 2024, July 22-25, 2024, Ann Arbor, MI, USA*, volume 300 of *LIPIcs*, pages 23:1–23:32. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[8] Arkadev Chattopadhyay, Nikhil S. Mande, Swagato Sanyal, and Suhail Sherif. Lifting to parity decision trees via stifling. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 33:1–33:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[9] J Cheeger. A lower bound for the smallest eigenvalue of the laplacian. *Problems Anal.*, page 195, 1970.

[10] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, March 1979.

[11] 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.

[12] Klim Efremenko, Michal Garlík, and Dmitry Itsykson. Lower bounds for regular resolution over parities. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 640–651. ACM, 2024.

[13] Michal Garlík and Leszek Aleksander Kolodziejczyk. Some subsystems of constant-depth frege with parity. *ACM Trans. Comput. Log.*, 19(4):29:1–29:34, 2018.

[14] 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.

[15] Svyatoslav Gryaznov. Notes on resolution over linear equations. In René van Bevern and Gregory Kucherov, editors, *Computer Science - Theory and Applications - 14th International Computer Science Symposium in Russia, CSR 2019, Novosibirsk, Russia, July 1-5, 2019, Proceedings*, volume 11532 of *Lecture Notes in Computer Science*, pages 168–179. Springer, 2019.

[16] Svyatoslav Gryaznov, Sergei Ovcharov, and Artur Riazanov. Resolution over linear equations: Combinatorial games for tree-like size and space. *ACM Trans. Comput. Theory*, jul 2024. Just Accepted.

[17] Svyatoslav Gryaznov, Pavel Pudlák, and Navid Talebanfard. Linear branching programs and directional affine extractors. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 4:1–4:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[18] Dmitry Itsykson and Artur Riazanov. Proof complexity of natural formulas via communication arguments. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 3:1–3:34. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[19] Dmitry Itsykson and Dmitry Sokolov. Lower bounds for splittings by linear combinations. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, volume 8635 of *Lecture Notes in Computer Science*, pages 372–383. Springer, 2014.

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

[21] Jan Krajíček. Randomized feasible interpolation and monotone circuits with a local oracle. *J. Math. Log.*, 18(2):1850012:1–1850012:27, 2018.

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

[23] W.J. Paul, R.E. Tarjan, and J.R. Celoni. Space bounds for a game on graphs. *Math. Systems Theory*, 10:239–251, 1976.

[24] Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR*, 41:333–338, 1987.

[25] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, 1987.

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

[27] Alasdair Urquhart. A near-optimal separation of regular and general resolution. *SIAM J. Comput.*, 40(1):107–121, 2011.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il