Electronic Colloquium on Computational Complexity, Report No. 187 (2023)

Lower bounds for regular resolution over parities

Klim Efremenko*[1], Michal Garlik†[2], and Dmitry Itsykson‡[1,3]

[1]Ben-Gurion University of the Negev, Israel
[2]Imperial College London, UK
[3]On leave from Steklov Institute of Mathematics at St. Petersburg

November 27, 2023

Abstract

The proof system resolution over parities (Res(⊕)) operates with disjunctions of linear equations (linear clauses) over $\mathbb{F}_2$; it extends the resolution proof system by incorporating linear algebra over $\mathbb{F}_2$. Over the years, several exponential lower bounds on the size of tree-like Res(⊕) refutations have been established. However, proving a superpolynomial lower bound on the size of dag-like Res(⊕) refutations remains a highly challenging open question.

We prove an exponential lower bound for regular Res(⊕). Regular Res(⊕) is a subsystem of dag-like Res(⊕) that naturally extends regular resolution. This is the first known superpolynomial lower bound for a fragment of dag-like Res(⊕) which is exponentially stronger than tree-like Res(⊕). In the regular regime, 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.

Namely, we show that the size of any regular Res(⊕) refutation of the binary pigeonhole principle BPHP_n^{n+1} is at least $2^{\Omega(\sqrt[3]{n}/\log n)}$. A corollary of our result is an exponential lower bound on the size of a strongly read-once linear branching program solving a search problem. This resolves an open question raised by Gryaznov, Pudlak, and Talebanfard [24].

As a byproduct of our technique, we prove that the size of any tree-like Res(⊕) refutation of the weak binary pigeonhole principle BPHP_n^m is at least $2^{\Omega(n)}$ using Prover-Delayer games. We also give a direct proof of a width lower bound: we show that any dag-like Res(⊕) refutation of BPHP_n^m contains a linear clause C with $\Omega(n)$ linearly independent equations.

Contents

*e-mail: klimefrem@gmail.com. Supported by the Israel Science Foundation (ISF) through grant No.1456/18 and European Research Council Grant No. 949707.

†e-mail: michal.garlik@gmail.com. Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 101002742). Part of this work was carried out while Michal was employed at the Euler International Mathematical Institute and supported by a grant from the Ministry of Science and Higher Education of the Russian Federation (agreement No. 075-15-2019-1620 dated 08/11/2019 and 075-15-2022-289 dated 06/04/2022).

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

ISSN 1433-8092

1 Introduction

Propositional proof complexity studies proof systems for the language of unsatisfiable CNF formulas (UNSAT). Complexity classes NP and coNP are different if and only if no proof system has polynomial size proofs for all formulas from UNSAT [14]. The main direction in proof complexity is to prove superpolynomial lower bounds on proof sizes for particular proof systems; this direction is also known as Cook's program for separating NP and coNP.

Resolution is one of the most studied and simplest propositional proof systems. A resolution refutation of a CNF formula φ is a sequence of 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 by the resolution rule from C_j and C_k, where $j, k < i$. The resolution rule allows us to derive the clause $C \vee D$ from clauses $C \vee x$ and $D \vee \neg x$, where x is a variable. Resolution is highly connected with contemporary SAT-solvers. The first practical SAT solvers were based on splitting (so-called DPLL algorithms due to Davis, Putnam, Loveland, and Logeman [16, 15]). Protocols of DPLL algorithms running on unsatisfiable formulas can be viewed as tree-like Resolution refutations. Current fastest SAT-solvers are based on the CDCL (Conflict-Driven Clause Learning) approach; the execution of such algorithms on unsatisfiable formulas actually contains a Resolution refutation [9]. Thus, formulas that require very large resolution proofs are also hard instances for DPLL and CDCL solvers.

Nowadays we know many formulas that require exponential size resolution refutations, however, for Frege systems (which include the standard propositional proof systems from logic textbooks) we don't know any superpolynomial lower bounds, and there is even a lack of good candidates for hard formulas. A derivation in a Frege system is a sequence of Boolean formulas (and a sequence of Boolean circuits in Extended Frege). The question of proving lower bounds for Frege systems is usually compared to proving lower bounds on the size of Boolean formulas or circuits for explicit Boolean functions. In general, both questions seem to be intractable by currently known techniques. However, some progress has been made on restricted versions of both questions. An exponential lower bound on the size of constant depth circuits computing parity was proved in the 1980s [18, 1]. Later, using a similar technique combined with a forcing argument, Ajtai proved a superpolynomial lower bound for bounded depth Frege systems [2]. Razborov and Smolenski in 1987 proved a lower bound for constant depth circuits built up from $\neg, \vee, \wedge$ and MOD_p gates [40, 39]. However, the analogous question of proving a lower bound for bounded depth Frege operating with formulas using $\neg, \vee, \wedge$ and MOD_p gates (denoted $\mathrm{AC}^0[p]$-Frege) is open for all values of $p > 1$.

In this paper, we study a subsystem of $\mathrm{AC}^0[2]$-Frege called resolution over parities, or Res($\oplus$) [30, 29]. The proof lines in this proof system are disjunctions of linear equations over $\mathbb{F}_2$, called *linear clauses.* Every linear clause $\bigvee_{i\in I}(f_i = a_i)$ is the negation of the linear system $\bigwedge_{i\in I}(f_i = a_i + 1)$. An ordinary clause (a disjunction of literals) is a special case of a linear clause since the literal $\neg x$ is equivalent to $x = 0$ and the literal x is equivalent to $x = 1$. A Res($\oplus$) refutation of an unsatisfiable CNF formula φ is a sequence of linear clauses $C_1, C_2, \dots, C_s$ 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 to derive D from C if C semantically implies D, i.e., any assignment satisfying C also satisfies D. For resolution refutations, the weakening rule is not necessary and can be eliminated without increasing the size of the refutation. However, in Res($\oplus$) the weakening rule is important since this is the only way to get equations depending on more than one variable.

Roughly speaking, Res($\oplus$) is a combination of resolution and linear algebra over $\mathbb{F}_2$. Unsatisfiable linear systems over $\mathbb{F}_2$ encoded as a CNF are easy for tree-like Res($\oplus$) [30], however, it is known that unsatisfiable linear systems over $\mathbb{F}_2$ based on expander graphs are hard for resolution [42].

Proving superpolynomial lower bounds on the size of Res($\oplus$) refutations is a frontier open problem on the way to obtaining lower bounds for $\mathrm{AC}^0[2]$-Frege. This question seems to be very challenging. There have been a few attempts to attack lower bounds for Res($\oplus$), below we summarize the main results achieved so far in this direction.

1.1 Review of previous results

1.1.1 Tree-like lower bounds

A Res($\oplus$) refutation is said to be tree-like if every linear clause appearing in the refutation is used at most once as a premise of a rule. There are many techniques for proving lower bounds for tree-like Res($\oplus$). We give a brief overview of the most important ones.

Prover-Delayer games. Initially, Prover-Delayer games were defined by Impagliazzo and Pudlak [37] for proving lower bounds for tree-like Resolution. Istykson and Sokolov [30] extended this game to tree-like Res($\oplus$). For a given unsatisfiable CNF formula φ, two players, Prover and Delayer, play the following game: the game starts with an empty board; in each step, Prover chooses a linear form f and Delayer either chooses $\alpha \in \{0, 1\}$ or allows Prover to choose α herself. In the latter case, Delayer earns a coin. They write the equation $f = \alpha$ on the board. The game stops when the linear system on the board contradicts a clause of φ. It is known that if Delayer can earn at least t coins with any behavior of Prover, then the size of any tree-like Res($\oplus$) refutation of φ is at least 2^t [30].

Using Prover-Delayer games, one can prove exponential lower bounds on the size of tree-like Res($\oplus$) refutations of the (unary) pigeonhole principle [29, 30] and various ordering principles [23]. These games were also used for proving exponential lower bounds on the running time of drunken DPLL($\oplus$) algorithms on satisfiable formulas [26]. Lower bound proofs using Prover-Delayer games are very explicit. However, such proofs are not known to work for CNFs consisting of clauses of small (e.g. constant) width.

Randomized communication complexity. Every unsatisfiable CNF formula φ defines a search problem $\mathrm{Search}(\varphi)$: given an assignment of variables, find a clause of φ falsified by this assignment.

Itsykson and Sokolov [29, 30] noticed that any size-S tree-like $\mathrm{Res}(\oplus)$ refutation of a formula φ can be transformed into a randomized communication protocol for $\mathrm{Search}(\varphi)$ of cost $O(\log S)$, where variables of φ are distributed between two communicating parties. Examples of formulas φ with large randomized communication complexity of $\mathrm{Search}(\varphi)$ can be found in [10, 25, 22]; all of them have a lifted structure: an essential formula (expressing some standard combinatorial principle) is lifted (i.e. composed) with some gadget. Itsykson and Ryazanov used slightly different communication complexity arguments to prove an exponential lower bound on the size of tree-like $\mathrm{Res}(\oplus)$ refutations of the perfect matching principle for graphs with an even number of vertices (while for graphs with an odd number of vertices perfect matching has short tree-like $\mathrm{Res}(\oplus)$ refutations [30]). Itsykson and Riazanov [30] also proved a lower bound on the randomized communication complexity of $\mathrm{Search}(\mathrm{BPHP}_n^{n+1})$, where BPHP_n^{n+1} is the binary pigeonhole principle with $n+1$ pigeons and n holes; Göös and Jain [21], using another approach, proved a lower bound on the randomized communication complexity for the search problem based on the slightly weak binary pigeonhole principle $\mathrm{Search}(\mathrm{BPHP}_n^{2n})$.

Reduction from polynomial calculus degree. Garlik and Kołodziejczyk [19] noted that any tree-like $\mathrm{Res}(\oplus)$ refutation of a k-CNF formula φ of size S can be converted to a dag-like $\mathrm{Res}(\oplus)$ refutation of φ of width $\log S + k + O(1)$, where the width of a refutation is the maximum number of linear equations that appear in a linear clause in the refutation. It is easy to see that a $\mathrm{Res}(\oplus)$ refutation of width w may be converted to a polynomial calculus (over $\mathbb{F}_2$) refutation of degree $w + O(1)$. This gives another method of proving lower bounds for tree-like $\mathrm{Res}(\oplus)$ via polynomial calculus degree lower bounds. For example, degree lower bounds for random 3-CNFs [6] and the functional graph pigeonhole principle [35] imply exponential lower bounds on the size of tree-like $\mathrm{Res}(\oplus)$ proofs of these formulas.

Lifting from resolution depth. Recently Chattopadhyay, Mande, Sanyal, and Sherif [12] developed a lifting technique from resolution depth to tree-like $\mathrm{Res}(\oplus)$ refutation size using stifling gadgets. Namely, if φ requires resolution depth d and g is a k-stifling gadget, then $\varphi \circ g$ requires a tree-like $\mathrm{Res}(\oplus)$ refutation of size at least 2^{kd}.

1.1.2 What is known for dag-like $\mathrm{Res}(\oplus)$?

Itsykson and Sokolov [30] considered systems $\mathrm{Res}(\oplus; \leqslant k)$ which are subsystems of $\mathrm{Res}(\oplus)$ operating with linear clauses that contain at most k equalities depending on more than one variable. Exponential lower bounds for $\mathrm{Res}(\oplus; \leqslant n^\delta)$ (where $\delta < 1$ is a constant and n is the number of variables) can be obtained by monotone interpolation. Exponential lower bounds for $\mathrm{Res}(\oplus; \leqslant \epsilon n)$ (where $\epsilon < 1$ is a constant) can be obtained by a simulation in Polynomial Calculus Resolution with a moderately exponential blowup [30].

Lauria [33] considered a system $\mathrm{Res}(\oplus_k)$, which is a subsystem of $\mathrm{Res}(\oplus)$ in which each equation in each linear clause uses at most k variables. This system is weaker than $\mathrm{Res}(k)$, hence lower bounds follow from lower bounds for $\mathrm{Res}(k)$.

Krajíček [32] presented a randomized feasible interpolation that is based on randomized communication complexity of evaluating a proofline. Krajíček reduced the question of a lower bound on $\mathrm{Res}(\oplus)$ to lower bounds for monotone CLO circuits (circuits with local oracles) that separate two disjoint NP sets. However, lower bounds for monotone CLO circuits are still unknown.

Khaniki [31] proved a superlinear lower bound for the dag-like version of $\mathrm{Res}(\oplus)$ (however, the proof system considered uses a different set of rules, and it is not clear whether the lower bound would remain non-trivial for the rules we use).

Several works have investigated proof systems Res(lin_R) operating with disjunctions of linear equations over a ring R for various rings. Raz and Tzameret studied the proof system Res($lin_{\mathbb{Z}}$) over integers [38]. Part and Tzameret considered many other fields and rings [36]. The field $\mathbb{F}_2$ differs from other rings in that over $\mathbb{F}_2$ the negation of an equality can also be represented as an equality; nevertheless Res($lin_{\mathbb{Z}}$) polynomially simulates Res($\oplus$) [30]. Part and Tzameret proved that the binary value principle $1+x_1+2x_2+\cdots+2^{n-1}x_n = 0$ requires exponential size refutations in dag-like Res($lin_{\mathbb{Q}}$). Alekseev [7] proved that the binary value principle is hard even for a stronger proof system called extended polynomial calculus over $\mathbb{Q}$. However, the binary value principle is not a CNF formula.

1.1.3 Read-once linear branching programs

It is known that every tree-like resolution refutation of a formula φ can be viewed as a decision tree for the problem Search(φ) and vice versa. The same equivalence exists between tree-like Res($\oplus$) refutations of φ and parity decision trees solving Search(φ) [30]. In the dag-like case, it is known that *regular* resolution refutations of φ are equivalent to read-once branching programs computing Search(φ). Recall that regular resolution is a subsystem of resolution. For every clause C we additionally have the list of variables V_C that were resolved in the resolution rules in the derivation of C. In the regular regime, it is allowed to resolve clauses C and D on a variable x only if $x \notin V_C \cup V_D$.

Linear branching programs extend branching programs by allowing them to query $\mathbb{F}_2$-linear forms instead of just variables. Gryaznov, Pudlak, and Talebanfard [24] introduced two versions of the read-once property for linear branching programs: weak and strong. Gryaznov, Pudlak, and Talebanfard [24] gave an explicit construction of a Boolean function that requires strongly read-once linear branching programs of exponential size in the average case; recently Eshan Chattopadhyay and Liao [13] and Li and Jong [34] have improved the lower bound.

As in the classical case, a weakly (and thus strongly) read-once linear branching program for Search(φ) can be converted to a Res($\oplus$) refutation of φ of the same size [24]. Gryaznov, Pudlak, and Talebanfard [24] raised the question of proving a lower bound for weakly and strongly read-once branching programs computing Search(φ). In this paper, we resolve the question for strongly read-once linear branching programs by giving an exponential size lower bound.

1.2 Our contributions

The main objective of this paper is to take a significant stride towards moving the frontier of currently known lower bounds for the tree-like Res($\oplus$) much closer to the dag-like Res($\oplus$). To achieve this, we consider a natural barrier, a specific fragment within dag-like Res($\oplus$) that possesses additional structural properties. This specific fragment naturally extends regular resolution, which is an important subsystem of resolution. Despite regular resolution being known as weaker than the resolution itself [20, 5], for major combinatorial principles it is either known (e.g. the pigeonhole principle or ordering principle) or widely believed (e.g. Tseitin formulas) that their shortest regular proofs are at most polynomially longer than their shortest resolution proofs. Despite the extensive study of resolution, several key questions remain open within the general resolution and find solutions only within the regular regime. For instance, Strong Exponential Time Hypothesis (SETH) for proof size [11], optimal average-case lower bound $n^{\Omega(k)}$ on proof size of formulas encoding that a random Erdős–Rényi graph does not contain a k-clique [8], and the exact derivation complexity of Tseitin formulas in terms of the treewidth of the underlying graph [27, 17] are known only for regular resolution.

We introduce the notion of *regular* Res($\oplus$) refutation. Analogously to ordinary resolution, with each linear clause C forming a proof line of a Res($\oplus$) derivation we remember the set F_C of all linear forms that were used in the resolution rules in the derivation of the clause C. In regular Res($\oplus$) refutations, it is allowed to resolve linear clauses C and D on a linear form f only if $f \notin \langle F_C \rangle \cup \langle F_D \rangle$, where $\langle \ldots \rangle$ denotes span.

It is easy to see that regular resolution is a subsystem of regular Res($\oplus$). It is known that tree-like Res($\oplus$) and regular resolution do not polynomially simulate each other and, moreover, they can be exponentially separated from each other (see Sections 3.2 and 3.5 of [30]). Regular Res($\oplus$) simulates tree-like Res($\oplus$)

(see Lemma 2.5). Thus regular Res(⊕) is exponentially stronger than both regular resolution and tree-like Res(⊕).

We prove an exponential lower bound on the size of regular Res(⊕) refutations. As a hard formula we use the binary pigeonhole principle $\mathrm{BPHP}^m_{2^\ell}$ which encodes in CNF that there are m pairwise distinct strings from $\{0,1\}^\ell$. We usually say that there are 2^ℓ holes and m pigeons and the ith string is the binary number of the hole where the ith pigeon sits. This formula is unsatisfiable if and only if $m > 2^\ell$.

Our main result is the following theorem:

Theorem (Theorem 8.1)**.** Any regular Res(⊕) refutation of $\mathrm{BPHP}^{2^\ell+1}_{2^\ell}$ has size at least $2^{\Omega\left(\frac{\sqrt[3]{n}}{\log n}\right)}$, where $n = 2^\ell$.

Corollary (Corollary 8.2)**.** The size of any strongly read-once linear branching program solving $\mathrm{Search}(\mathrm{BPHP}^{2^\ell+1}_{2^\ell})$ is at least $2^{\Omega\left(\frac{\sqrt[3]{n}}{\log n}\right)}$, where $n = 2^\ell$.

This corollary answers the question raised by Gryaznov, Pudlak and Talebanfard [24].

We develop tools to prove the theorem and use them to obtain additional results. Theorem 5.5 shows that any Res(⊕) refutation of BPHP^m_n contains a linear clause such that $\mathrm{rk}(\neg C) \geq n/4$. The attractive feature of this result is that it is proven very directly: we demonstrate a procedure for finding such a wide clause in every refutation. All previously known lower bounds on width/rank were based on polynomial calculus degree lower bounds, hence they were very indirect. Further, we show that the size of any tree-like Res(⊕) refutation of BPHP^m_n is at least $2^{n/4}$ (see Theorem 5.8). This proof is mostly interesting because of its explicitness. It is done by means of a Prover-Delayer game for tree-like Res(⊕), and this is the first example of such a game for formulas that have only narrow clauses.

1.3 Technique

In this section, we give a high-level overview of our technique and proof strategy for the lower bounds.

Proof lines in a Res(⊕) refutation are linear clauses, so we can view them as negations of systems of linear equations. Let $X = \{x_{i,j} \mid i \in [m], j \in [\ell]\}$ denote the set of variables of BPHP^m_n, where $n = 2^\ell$. The meaning of the variables is that the string $x_{i,1}, \ldots, x_{i,\ell}$ encodes in binary the hole in which the ith pigeon sits. We study satisfiable systems of linear equations over $\mathbb{F}_2$ in variables from X.

1.3.1 Safe and dangerous sets of linear forms

We say that a set of linear forms $f_1, f_2, \ldots, f_k$ with variables from X is *dangerous* if it is linearly independent and mentions less than k pigeons. The set of linear forms F is *safe* if its span $\langle F \rangle$ does not contain any dangerous sets. In Section 3 we prove the following equivalence.

Theorem (Theorem 3.1)**.** A set of linearly independent forms $f_1, f_2, \ldots, f_k$ is safe if and only if their coefficient matrix contains k linearly independent columns corresponding to k distinct pigeons.

The proof of this theorem uses an extension of Hall's matching theorem for matroids [43].

It follows from the theorem that to solve a linear system that has a safe set of linear forms $f_1, f_2, \ldots, f_k$ as left-hand sides, we can assign values to all variables except the chosen k (which correspond to distinct pigeons) arbitrarily, and the values of the remaining k variables will be uniquely determined. Thus any linear system based on a safe set of forms actually restricts at most one bit for each pigeon.

1.3.2 Closure

For every set of linear forms F we define the notion of its *closure* as an inclusion minimal set of pigeons such that if we set all variables mentioning these pigeons to zero, then the set F becomes safe. In Section 4 we study this notion and prove the following properties of the closure:

- (Uniqueness) For every F its closure is unique. We denote it by $\mathrm{Cl}(F)$;

- (Monotonicity) If $F \subseteq F'$, then $\mathrm{Cl}(F) \subseteq \mathrm{Cl}(F')$;
- (Span invariance) $\mathrm{Cl}(F) = \mathrm{Cl}(\langle F \rangle)$;
- (Size bound) $|\mathrm{Cl}(F)| \le \dim\langle F \rangle$.

The definition of the closure is similar in spirit to the concept of the closure operator for expanders that was originally defined in [6, 3] under different names and further called closure i.e. in [4, 28, 41]; however, we don't know any formal connections between these two notions. Informally speaking, the closure of a set of linear forms is a set of pigeons that may be highly restricted by a system of linear equations having this set of forms.

1.3.3 Locally consistent linear systems

We say that a linear system is *locally consistent* if it has a solution that sends the pigeons from the closure of the set of linear forms of the system to holes injectively. We notice that the empty system is locally consistent and that the negation of the clauses of BPHP^m_n are not locally consistent. In order to analyze a $\mathrm{Res}(\oplus)$ refutation of BPHP^m_n we only focus on those its clauses whose negations are locally consistent. To demonstrate that this notion is indeed useful we first present a rank lower bound and then a tree-like lower bound.

Rank lower bound. We are going to show that any $\mathrm{Res}(\oplus)$ refutation of BPHP^m_n contains a linear clause C such that the rank of the linear system $\neg C$ is at least $\frac{n}{4}$. In Section 5 we give some properties of local consistency, here we sketch two of them that we need for the rank lower bound.

- If a linear clause C is a weakening of a linear clause D and $\neg C$ is locally consistent, then $\neg D$ is also locally consistent.

  Proof. If C is a weakening of D, then by contraposition $\neg C$ semantically implies $\neg D$, hence $L(D) \subseteq \langle L(C) \rangle$, where $L(A)$ denotes the set of linears form in a linear clause A. Thus, by the closure properties, $\mathrm{Cl}(L(D)) \subseteq \mathrm{Cl}(\langle L(C) \rangle) = \mathrm{Cl}(L(C))$. Any solution of $\neg C$ is also a solution of $\neg D$ and if there exists a solution of $\neg C$ that is injective on $\mathrm{Cl}(L(C))$, then it also is injective on $\mathrm{Cl}(L(D))$. □

- Let a linear clause C be obtained by the resolution rule applied to linear clauses D and E. Assume that $\neg C$ is locally consistent and $\mathrm{rk}(\neg C) < \frac{n}{4}$. Then at least one of $\neg D$ and $\neg E$ is locally consistent.

  Proof sketch. Using the previous property it is sufficient to show that for every linear form f for some $\alpha \in \{0, 1\}$, $\neg C \wedge (f = \alpha)$ is locally consistent. In other words, it is sufficient to show that $\neg C$ has a solution that is injective on $\mathrm{Cl}(L(C) \cup \{f\})$. If $\mathrm{Cl}(L(C) \cup \{f\}) = \mathrm{Cl}(L(C))$, then we have nothing to do since the solution we need exists by the local consistency of $\neg C$. If $\mathrm{Cl}(L(C) \cup \{f\}) \ne \mathrm{Cl}(L(C))$, then we take the system $\neg C$ and fix the values of all variables mentioning the pigeons from $\mathrm{Cl}(L(C))$ according to an assignment guaranteed by the local consistency of $\neg C$. The resulting system has its set of forms safe, hence we may substitute almost arbitrarily values to variables (we do not control at most one variable of each pigeon). Since $\mathrm{rk}(\neg C) < \frac{n}{4}$, it follows that $|\mathrm{Cl}(L(C) \cup f)| \le \mathrm{rk}(\neg C) + 1 \le n/4$, so we have enough freedom to construct a solution such that all pigeons in $\mathrm{Cl}(L(C) \cup \{f\})$ will be in different holes. □

These properties give a simple way to find in a $\mathrm{Res}(\oplus)$ refutation of BPHP^m_n a linear clause C such that $\neg C$ has a large rank. We start a path in the empty clause and make steps from a linear clause to one of its premises, maintaining the property that the negation of the current linear clause is locally consistent. This path cannot reach a clause of the initial formula since its negation is not locally consistent. Thus, our path can only end in a linear clause such that $\mathrm{rk}(\neg C) \ge \frac{n}{4}$.

Tree-like lower bound. The lower bound on the size of tree-like Res($\oplus$) refutations of BPHP^m_n is proved by constructing a strategy for Delayer in Prover-Delayer games. The strategy is as follows: Delayer tries to keep the current linear system locally consistent. As we have already seen, if Φ is locally consistent and $\mathrm{rk}(\Phi) < n/4$, then for every linear form f there is $\alpha \in \{0, 1\}$ such that $\Phi \wedge (f = \alpha)$ is also locally consistent. Consider the following strategy of Delayer: if Φ is the linear system on the board and f is a query such that for some $\alpha \in \{0, 1\}$, $\Phi \wedge (f = \alpha)$ is not locally consistent, then Delayer answers $\alpha + 1$; otherwise Delayer allows Prover to choose a value (i.e. Delayer earns a coin). In Section 5.2 we show that such a strategy guarantees that before the system on the board becomes not locally consistent, Delayer earns at least $n/4$ coins. Thus the size of any Res($\oplus$) tree-like refutation of BPHP^m_n is at least $2^{n/4}$.

1.3.4 Regular lower bound

The proof of the lower bound on the size of regular refutations of BPHP^{n+1}_n consists of two main steps:

1. We consider the following random walk in a refutation graph of BPHP^{n+1}_n. We choose a random full assignment σ of BPHP^{n+1}_n. We start a path in the empty clause and make $\sqrt[3]{n}$ steps, each time we go from a clause to its premise that is falsified by σ; we may stop earlier than in $\sqrt[3]{n}$ steps if we come to a clause of BPHP^{n+1}_n. We claim that with probability at least $\frac{1}{2}$ this path will end in a clause C such that $\neg C$ is locally consistent. The idea is that in $\sqrt[3]{n}$ steps we come to a linear clause C with $\mathrm{rk}(\neg C) \leq \sqrt[3]{n}$, hence $\mathrm{Cl}(L(C)) \leq \sqrt[3]{n}$ and, thus, by birthday paradox, with high probability a random assignment assigns different holes for pigeons from $\mathrm{Cl}(L(C))$. See Section 6 for details.

2. We consider any clause C from a regular Res($\oplus$) refutation of $\mathrm{BPHP}^{2^\ell+1}_{2^\ell}$ such that $\neg C$ is locally consistent. Let t be the distance in the refutation graph between C and the empty clause. We claim that the rank of $\neg C$ is at least $\Omega(\frac{t}{\ell})$. The proof is given in Section 5.1. This is the only place in the whole proof where we use regularity.

So if we take a random walk of length $\sqrt[3]{n}$ in a regular refutation, with probability at least $\frac{1}{2}$ we will reach a linear clause C such that the rank of $\neg C$ is at least $\Omega\left(\frac{\sqrt[3]{n}}{\log n}\right)$. By the construction of our random walk, the assignment σ refutes C, hence it satisfies $\neg C$. For a particular clause C such that the rank of $\neg C$ equals t, a random assignment refutes C with probability exactly 2^{-t}. Hence the refutation contains at least $2^{\Omega\left(\frac{\sqrt[3]{n}}{\log n}\right)}$ linear clauses.

2 Preliminaries

2.1 Linear algebra

For a set of vectors U from a vector space V we denote by $\langle U \rangle$ 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 system is a conjunction of linear equations.

We say that a linear equation $f = a$ is implied by a linear system Φ (or $f = a$ is a corollary of Φ) if any solution of Φ satisfies $f = a$.

Lemma 2.1 (folklore)**.** A linear equation $f = a$ is implied by a satisfiable linear system Φ if and only if $f = a$ can be obtained as a linear combination of equations from Φ.

2.2 Resolution over parities

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. $L\left(\bigvee_{i=1}^{t}(f_i = a_i)\right) = \{f_1, f_2, \ldots, f_t\}$.

Let φ be an unsatisfiable CNF formula. A refutation of φ in the proof system Res($\oplus$) [30] 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 to derive from linear clauses $C \vee (f = a)$ and $D \vee (f = a + 1)$ the linear clause $C \vee D$.
- *Weakening rule* allows 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.

Actually, we will use another graph $\tilde{G}_\Pi$ that is 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 edges of this path. The label of the sink 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 $(\sigma, C) \in$ Search(ϕ).

Consider a Res($\oplus$) refutation graph G_Π of a CNF formula φ. We now show that the graph G_Π can be relabeled such that it turns 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 $f_{v_1} = a$.

Remark 2.2. Assume that a linear clause C_1 is the result of the resolution rule applied to a weakening of a linear clause C_2 and a weakening of a linear clause C_3. Notice that the resolved linear form is not necessarily uniquely determined. Consider, for example, $C_1 = (x+y = 0)$, $C_2 = (x = 0)$, $C_3 = (y = 0)$ and consider two derivations: (1) $(x + y = 0 \vee y = 1)$ is a weakening of C_2, and C_1 can be obtained from $(x + y = 0 \vee y = 1)$ and C_3 by resolving on y; (2) $(x + y = 0 \vee x = 1)$ is a weakening of C_3, and C_1 can be obtained from $(x + y = 0 \vee x = 1)$ and C_2 by resolving on x.

To avoid ambiguity in the construction of a linear branching program associated with a Res($\oplus$) refutation graph, we will assume that every Res($\oplus$) refutation graph also keeps a record of the resolved linear forms at all its nodes.

Lemma 2.3. 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 of its nodes labeled with linear clauses C_u and C_v such that there is a path p connecting 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 $\neg C_v$.

Proof. We prove the lemma by induction on the length of p. In the base of induction, p has zero length and $u = v$; the statement is trivial. Induction step. Let w be the predecessor of v on the path p and let w be labeled with C_w. Assume that C_w is the result of the resolution rule applied to $D_1 \vee (f_w = a)$ and $D_2 \vee (f_w = a + 1)$, where $D_1 \vee (f_w = a)$ is the weakening of C_v. Let Ψ_p be the linear system written on the part of the path p from u to w. By the construction, the edge (w, v) is labeled with $f_w = a + 1$, hence $\Phi_p = \Psi_p \wedge (f_w = a + 1)$. By the inductive hypothesis, $\neg C_u \wedge \Psi_p$ implies $\neg C_w = \neg D_1 \wedge \neg D_2$. Then $\neg C_u \wedge \Phi_p = \neg C_u \wedge \Psi_p \wedge (f_w = a+1)$ implies $\neg D_1 \wedge (f_w = a+1)$, and $\neg D_1 \wedge (f_w = a+1)$ implies $\neg C_v$, since C_v semantically implies $D_1 \vee (f_w = a)$. Thus $\neg C_u \wedge \Phi_p$ implies $\neg C_v$ and the inductive step is proved. □

Lemma 2.3 implies that every Res($\oplus$) refutation graph of a formula φ may be also considered as a linear branching program solving the search problem Search(φ).

A Res($\oplus$) refutation is called *tree-like* if any non-sink node of the refutation graph has at most one incoming edge. A *parity decision tree* is a linear branching program such that any non-sink node has at most one incoming edge. So the last observation implies that a tree-like Res($\oplus$) refutation of φ can be thought of as a parity decision tree for Search(φ).

2.4 Regular refutations

For a node v of a linear branching program, we denote by Pre(v) the linear span of all linear forms f such that f is a query at a node $u \neq v$ on a path from the source to v. 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 linear branching program is *weakly read-once* if for all non-sink nodes v, $f_v \notin \text{Pre}(v)$, where f_v is a query at a node v [24]. A linear branching program is *strongly read-once* if for all nodes v, $\text{Pre}(v) \cap \text{Post}(v) = \{0\}$ [24].

A Res($\oplus$) refutation is called *top-regular* if the associated linear branching program is weakly read-once.

We have already shown that any Res($\oplus$) refutation graph of φ can be considered as a linear branching program for Search(φ). Gryaznov, Pudlák and Talebanfard [24] showed that any weakly read-once linear branching program for Search(φ) can be viewed as a top-regular Res($\oplus$) refutation of φ. We found the proof of this statement in [24] slightly confusing, so we include a proof here for clarity.

Lemma 2.4 ([24])**.** For any weakly read-once linear branching program solving $\mathrm{Search}(\varphi)$, there is a labeling of its non-sink nodes with linear clauses that makes it a refutation graph of a top-regular $\mathrm{Res}(\oplus)$ refutation of φ. Moreover, for every node, its query coincides with the linear form resolved at this node.

Proof. Let P be a weakly read-once linear branching program solving $\mathrm{Search}(\varphi)$. Note that for any path from the root of P, the conjunction of the linear equations labeling the edges of the path is satisfiable, since by the weakly read-once property each next equation is linearly independent of the previous ones. We will construct the required refutation by putting linear clauses in nodes of P starting from the sinks, which are already labeled with clauses of φ. We maintain the following invariant: for every path from the source to a node labeled with a linear clause C, the conjunction of the linear equations labeling the edges of the path in P implies $\neg C$. The invariant holds for the sinks by the definition of a linear branching program. Consider a node v with query f and let both of the direct successors of v be already labeled with linear clauses C_0 and C_1 respectively; assume that the edge labeled with $f = \alpha$ goes from v to a node v_α labeled with C_α, for $\alpha \in \{0, 1\}$. Consider a path p from the source to v and let Φ_p be the linear system corresponding to this path in P.

According to the invariant, $\Phi_p \wedge (f = \alpha)$ implies $\neg C_\alpha$. Let $\neg C_\alpha = \bigwedge_{i \in I^{(\alpha)}} (g_i = a_i)$. For every $i \in I^{(\alpha)}$, either Φ_p implies $g_i = a_i$ or $\Phi_p \wedge (f = \alpha)$ implies $g_i = a_i$ but Φ_p does not imply $g_i = a_i$. In the second case, by Lemma 2.1, $g_i = a_i$ is a linear combination of the equations $\Phi_p \wedge (f = a)$, but $g_i = a_i$ is not a linear combination of the equations Φ_p, hence $g_i + f = a_i + \alpha$ is a linear combination of the equations Φ_p, and therefore Φ_p implies $g_i + f = a_i + \alpha$. Let $I_1^{(\alpha)} = \{i \in I^{(\alpha)} \mid g_i = a_i \text{ is implied by } \Phi_p\}$ and $I_2^{(\alpha)} = I^{(\alpha)} \setminus I_1^{(\alpha)}$. Note that the partition $I_1^{(\alpha)}$ and $I_2^{(\alpha)}$ does not depend on p. Indeed, assume that for two different paths p_1 and p_2 from the source to v for some $i \in I^{(\alpha)}$, Φ_{p_1} implies $g_i = a_i$ and Φ_{p_2} implies $g_i + f = a_i + \alpha$. Then $f \in \langle F_1 \cup F_2 \rangle$, where F_1 and F_2 are the sets of the linear forms from left-hand sides of Φ_{p_1} and Φ_{p_2}. The latter implies $f \in \mathrm{Pre}(v)$, which contradicts the weakly read-once property.

Consider the clause $C'_\alpha = \neg(\bigwedge_{i \in I_1^{(\alpha)}} (g_i = a_i) \wedge \bigwedge_{i \in I_2^{(\alpha)}} (g_i + f = a_i + \alpha) \wedge (f = \alpha))$. It is easy to see that C'_α can be obtained from C_α by the weakening rule. We have seen that for every $i \in I_2^{(\alpha)}$, Φ_p implies $g_i + f = a_i + \alpha$. Consider the clause $C = \neg\left(\bigwedge_{i \in I_1^{(0)} \cup I_1^{(1)}} (g_i = a_i) \wedge \bigwedge_{\alpha \in \{0,1\}} \left(\bigwedge_{i \in I_2^{(\alpha)}} (g_i + f = a_i + \alpha)\right)\right)$. Note that C is obtained from C'_0 and C'_1 by the resolution rule. By the construction Φ_p implies $\neg C$. By the remark above the same holds for every path p from the source to v. So we may put C to the node v. □

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

Lemma 2.5 ([24])**.** Given a tree-like $\mathrm{Res}(\oplus)$ refutation of φ, one can construct a tree-like $\mathrm{Res}(\oplus)$ refutation of φ of no larger size that is top-regular and bottom-regular.

Proof sketch. Consider the parity decision tree associated with a given tree-like refutation. First, we will ensure that $f_v \notin \mathrm{Pre}(v)$ for all non-sink v. Since the underlying graph of the refutation is a tree, for every node v there is a unique path from the source to v. Note that if for some v, $f_v \in \mathrm{Pre}(v)$, then the value of f_v is determined by the path from the source to v. Therefore, the desired property is achieved by applying the following while loop to the parity decision tree. While there is some node v with $f_v \in \mathrm{Pre}(v)$, delete the query in v and merge v with the child corresponding to the correct value of f_v.

In the rest of the proof, we will repeatedly use the following transformation of the parity decision tree: consider two nodes a and b such that there is a path from a to b and change the query at b to $f_a + f_b$ (it is easy to recompute the labels of edges leaving b since the value of f_a is already known). Note that such transformations cannot violate the property $f_v \notin \mathrm{Pre}(v)$, which we have already achieved.

Apply the following while loop to the parity decision tree obtained so far. While there is a non-sink node v and an edge (v, w) such that $f_v \in \mathrm{Post}(w)$, take such a v with the minimum possible distance from the source (the root of the tree). Extend the set $\{f_v\}$ to some basis of the set of all linear forms over the variables of φ. For all vertices $u \neq v$ reachable from v, if f_u has the f_v-coordinate in the basis equal to 1, change the query at u to $f_v + f_u$.

Note that one loop iteration does not change $\mathrm{Post}(v)$ and hence it does not affect the predecessors of v, but at the end of the iteration $f_v \notin \mathrm{Post}(w)$ for both immediate successors w of v.

Finally, we translate the resulting parity decision tree back to tree-like $\mathrm{Res}(\oplus)$ refutation by Lemma 2.4. □

Lemma 2.6. Suppose that ϕ is an unsatisfiable CNF formula in n variables, and Π is a regular $\mathrm{Res}(\oplus)$ 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 $\mathrm{Post}(v)$ is at most $n - d$.

Proof. Consider a path from the source to v of length d: $u_0, u_1, \ldots, u_d = v$. It is clear that $\mathrm{Post}(u_d) \subseteq \mathrm{Post}(u_{d-1}) \subseteq \mathrm{Post}(u_{d-2}) \subseteq \ldots \mathrm{Post}(u_0)$. The regularity implies that $n \geq \dim \mathrm{Post}(u_0) \geq 1 + \dim \mathrm{Post}(u_1) \geq \cdots \geq d + \dim \mathrm{Post}(v)$. □

Lemma 2.6 gives the only consequence of regularity that we need for our proof of the lower bound on the size of regular $\mathrm{Res}(\oplus)$ refutations.

2.5 Binary pigeonhole principle

The binary pigeonhole principle $\mathrm{BPHP}^m_{2^\ell}$ states that m pigeons can be placed in 2^ℓ holes such that every pigeon sits in a hole and no two pigeons sit in the same hole. The address of each hole can be represented as an ℓ-bit binary string, and so this principle can be expressed as the statement that there are m pairwise different ℓ-bit binary strings $s_1, s_2, \ldots, s_m$, where s_i is the binary number of the hole in which the ith pigeon sits. $\mathrm{BPHP}^m_{2^\ell}$ has $m\ell$ variables corresponding to the bits of s_i for $i \in [m]$; namely, for every $i \in [m]$ and $j \in [\ell]$, the variable $x_{i,j}$ denotes the jth bit of s_i. Then $\mathrm{BPHP}^m_{2^\ell}$ is $\bigwedge_{i \neq k \in [m]} s_i \neq s_k$, where the predicate $s_i \neq s_k$ is encoded as a 2ℓ-CNF formula with 2^ℓ many clauses as follows: $\bigwedge_{\alpha \in \{0,1\}^\ell} (s_i \neq \alpha \vee s_k \neq \alpha)$, where $s_i \neq \alpha \vee s_k \neq \alpha$ is the following clause with with 2ℓ literals:

$$(x_{i,1} = \alpha_1 + 1) \vee (x_{i,2} = \alpha_2 + 1) \vee \cdots \vee (x_{i,\ell} = \alpha_\ell + 1) \vee (x_{k,1} = \alpha_1 + 1) \vee (x_{k,2} = \alpha_2 + 1) \vee \cdots \vee (x_{k,\ell} = \alpha_\ell + 1),$$

where $x = 1$ denotes x, $x = 0$ denotes $\neg x$, and $\alpha_1, \alpha_2, \ldots, \alpha_l$ are the bits of α.

We usually denote the number of holes by $n = 2^\ell$. If $m > n$, then the formula BPHP^m_n is unsatisfiable.

Let X be the set of variables of the formula $\mathrm{BPHP}^m_{2^\ell}$. Every Boolean assignment σ with domain X naturally corresponds to a mapping $\tilde{\sigma} : [m] \to [n]$ as follows: $\tilde{\sigma}(i) = 1 + \sum_{j=1}^{\ell} 2^{j-1} \sigma(x_{i,j})$.

3 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 $i \in [m]$. Since X is the set of variables of the binary pigeonhole principle $\mathrm{BPHP}^m_{2^\ell}$, when we start working with this principle later, we will use that blocks correspond to pigeons and the variables of a block encode a hole.

The point of departure for our lower bound results is to consider a system of linear equations in the variables X and to formalize the notion of many independent linear consequences of this system talking about variables from a small number of blocks. It turns out that this notion is already determined by the set of linear forms forming the left-hand side of the system.

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 $\mathrm{supp}(f)$. 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 $\mathrm{supp}(F)$. We say that a linearly independent set of linear forms F is *dangerous* if $|F| > |\mathrm{supp}(F)|$. We say that a set of linear forms F is *safe* if $\langle F \rangle$ does not contain a dangerous set.

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 3.1. Let $f_1, f_2, \dots, f_k$ be linearly independent linear forms and let M be their coefficient matrix. Then the following conditions are equivalent.

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

(2) For every set $T \subseteq [m]$, the dimension of the span of the set of columns of M corresponding to the variables with support in T is at least $|T| - (m - k)$.

(3) 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.

The main ingredient of the proof of this theorem is treated in the next subsection.

3.1 Extended Hall's Theorem

The following extension of the well-known Hall's matching theorem was proved by Welsh in 1971.

Theorem 3.2 ([43])**.** Let L be some vector space and $V_1, V_2, \dots, V_n$ be sets of vectors from L such that for every $A \subseteq [n]$ the dimension of $\langle \cup_{i\in A} V_i \rangle$ is at least $|A|$. Then for every $i \in [n]$ there exists $v_i \in V_i$ such that $v_1, v_2, \dots, v_n$ are linearly independent.

We need a slightly more general statement but it has virtually the same proof.

Theorem 3.3. Suppose that L is a vector space, $V_1, V_2, \dots, V_n$ are sets of vectors from L and $k \in [n]$ is such that for every $A \subseteq [n]$ the dimension of $\langle \cup_{i\in A} V_i \rangle$ is at least $|A| - k$. Then there exist distinct indices $i_1, i_2, \dots, i_{n-k} \in [n]$ and for every $j \in [n-k]$ there exists $v_j \in V_{i_j}$ such that $v_1, v_2, \dots, v_{n-k}$ are linearly independent.

Proof. We argue by contradiction. Suppose that the theorem is false. Consider a counterexample with the minimum value of $\sum_{i=1}^{n} |V_i|$.

First, suppose that $|V_i| \le 1$ for every i. Since $\dim\langle \cup_{i\in[n]} V_i \rangle \ge n-k$, there exist $n-k$ linearly independent vectors $v_1, v_2, \dots, v_{n-k}$ in $\cup_{i\in[n]} V_i$; no two of them are from the same V_i since the size of each V_i is at most one. So the conclusion of the theorem is satisfied, and the counterexample cannot have $|V_i| \le 1$ for all i.

Thus there exists $j \in [n]$ such that $|V_j| \ge 2$. Let us choose two different elements from V_j and denote them by a and b. Since we have chosen the counterexample with the minimum total number of elements, the sets $V_1, V_2, \dots, V_j \setminus a, \dots, V_n$ and $V_1, V_2, \dots, V_j \setminus b, \dots, V_n$ do not satisfy the hypothesis of the theorem.

Thus, there exist $J_1 \subseteq [n] \setminus \{j\}$ and $J_2 \subseteq [n] \setminus \{j\}$ such that $\dim\langle \cup_{i\in J_1} V_i \cup (V_j \setminus \{a\}) \rangle \le |J_1| - k$ and $\dim\langle \cup_{i\in J_2} V_i \cup (V_j \setminus \{b\}) \rangle \le |J_2| - k$.

From these two inequalities we get

$$|J_1| + |J_2| \ge \dim\langle \cup_{i\in J_1} V_i \cup (V_j \setminus \{a\}) \rangle + \dim\langle \cup_{i\in J_2} V_i \cup (V_j \setminus \{b\}) \rangle + 2k \ge$$
$$\dim\langle \cup_{i\in J_1\cap J_2} V_i \rangle + \dim\langle \cup_{i\in J_1\cup J_2} V_i \cup V_j \rangle + 2k \ge |J_1 \cap J_2| + |J_1 \cup J_2| + 1 = |J_1| + |J_2| + 1,$$

a contradiction. In the second inequality, we use that for every two subspaces X and Y of the same linear space,

$$\dim(X) + \dim(Y) = \dim\langle X \cup Y \rangle + \dim(X \cap Y).$$

□

3.2 Proof of Theorem 3.1

Proof of Theorem 3.1. Let us prove the equivalence of the first two conditions. Consider an arbitrary set of blocks $T \subseteq [m]$. Consider a submatrix M_T of M that contains only the columns indexed by variables with support in T. Consider the vector space $V_T \subseteq \{0,1\}^k$ consisting of all vectors that have zero inner product with every column of M_T. The dimension of V_T equals $k - \mathrm{rk}(M_T)$. Consider the space $H_T = \langle \sum \alpha_i f_i \mid \alpha \in V_T \rangle$. Notice that $H_T = \{g \in \langle f_1, f_2, \dots, f_k \rangle \mid \mathrm{supp}(g) \subseteq [m] \setminus T\}$. Since $f_1, f_2, \dots, f_k$ are linearly independent, $\dim H_T = \dim V_T = k - \mathrm{rk}(M_T)$.

The set $f_1, f_2, \dots, f_k$ is safe if and only if for every $T \subseteq [m]$, $\dim H_T \le m - |T|$ which is equivalent to $\mathrm{rk}(M_T) \ge k - (m - |T|)$. Thus, items (1) and (2) are equivalent.

Now assume (2) and let us prove (3). Consider vector spaces $V_1, V_2, \dots, V_m$, where V_i consists of columns of M corresponding to the variables with support $\{i\}$. By Theorem 3.3 applied to $V_1, V_2, \dots, V_m$ and $(m-k)$, there exist distinct numbers $i_1, i_2, \dots, i_k$ and vectors $v_j \in V_{i_j}$ for $j \in [k]$ such that $v_1, v_2, \dots, v_k$ are linearly independent. Note that $v_1, v_2, \dots, v_k$ are columns of M corresponding to different blocks. Thus the third condition holds.

Finally, assume that the third condition holds and there are k chosen columns of M corresponding to different blocks. Let $T \subseteq [m]$. At most $m - |T|$ of the chosen columns have their corresponding block in $[m] \setminus T$, hence there are at least $k - m + |T|$ of the chosen columns with their corresponding block in T. Therefore, the dimension of the span of the set of columns of M corresponding to variables with support in T is at least $|T| - (m - k)$. I.e., the second condition holds. □

4 Closure of a set of linear forms

Let $S \subseteq [m]$ be a set of blocks; for a linear form f we denote by $f[\setminus S]$ a linear form obtained from f by substituting 0 for all variables with support in S. In other words, $f[\setminus S]$ is the projection of f to the linear space of all forms with support in $[m] \setminus S$. Being a projection, $[\setminus S]$ is a linear operator for every $S \subseteq [m]$.

For a set of linear forms F we will use the notation $F[\setminus S] = \{f[\setminus S] \mid f \in F\}$.

A set of linear forms F is *minimally dangerous* if it is dangerous and $\langle F \rangle$ does not contain a dangerous set with strictly smaller support than the support of F.

Assume that a set of linear forms F is not safe. We would like to remove some blocks of variables from F to obtain a safe set. Consider the following algorithm:

Algorithm 4.1. Input: a set of linear forms F.

```
1. $S \leftarrow \emptyset$;
2. While $\langle F[\setminus S] \rangle$ contains dangerous sets:
    • Find a minimally dangerous set in $\langle F[\setminus S] \rangle$. Let $T$ be its support.
    • $S \leftarrow S \cup T$.
3. Return $S$.
```

We will show in Corollary 4.6 that the output of Algorithm 4.1 does not depend on the choice of minimally dangerous sets in step 2 and we will call the output of the algorithm the closure of F. We start with a formal definition of the closure and a bit later we prove that Algorithm 4.1 indeed computes it.

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

Note that $F[\setminus [m]] = \{0\}$ if $F \ne \emptyset$, hence $F[\setminus [m]]$ is safe, and therefore a closure of F exists. Our goal in this section is to prove the main properties of the closure.

1. (Uniqueness) For any F its closure is unique and we will denote it by $\mathrm{Cl}(F)$.
    - This property is proved in Subsection 4.1 (see Lemma 4.4).
2. (Monotonicity) If $F_1 \subseteq F_2$, then $\mathrm{Cl}(F_1) \subseteq \mathrm{Cl}(F_2)$.

    Proof. $F_1[\setminus \mathrm{Cl}(F_2)] \subseteq F_2[\setminus \mathrm{Cl}(F_2)]$, hence $F_1[\setminus \mathrm{Cl}(F_2)]$ is safe. Consider an inclusion minimal set $S \subseteq \mathrm{Cl}(F_2)$ such that $F_1[\setminus S]$ is safe. Then S is a closure of F_1 and, by the uniqueness, $\mathrm{Cl}(F_1) = S \subseteq \mathrm{Cl}(F_2)$. □

3. (Span invariance) $\mathrm{Cl}(F) = \mathrm{Cl}(\langle F \rangle)$.

   Proof. Since $[\backslash S]$ is a linear operator, $\langle F \rangle[\backslash S] = \langle F[\backslash S] \rangle$. Hence for every S, the set $F[\backslash S]$ is safe iff $\langle F \rangle[\backslash S]$ is safe, and so $\mathrm{Cl}(F) = \mathrm{Cl}(\langle F \rangle)$. □

4. (Size bound) $|\mathrm{Cl}(F)| + \dim\langle F[\backslash \mathrm{Cl}(F)]\rangle \le \dim\langle F \rangle$ and hence $|\mathrm{Cl}(F)| \le \dim\langle F \rangle$.

   - This property is proved in Subsection 4.2 (see Lemma 4.7).

5. (Increment) Let F be a set of linear forms and f be a linear form such that $\mathrm{Cl}(F \cup \{f\}) \neq \mathrm{Cl}(F)$. Then

$$\dim\langle F[\backslash \mathrm{Cl}(F)]\rangle - \dim\langle (F \cup \{f\})[\backslash \mathrm{Cl}(F \cup \{f\})]\rangle = |\mathrm{Cl}(F \cup \{f\})| - |\mathrm{Cl}(F)|.$$

   - This property is proved in Subsection 4.3 (see Lemma 4.9).

4.1 Uniqueness of closure

Lemma 4.2. Let F be a set of linear forms and T be a subset of $[m]$. Then

$$\dim\langle F \rangle = \dim\langle F[\backslash T]\rangle + \dim\{f \in \langle F \rangle \mid \mathrm{supp}(F) \subseteq T\}.$$

Proof. Let $f_1, f_2, \ldots, f_k$ be basis of $\{f \in \langle F \rangle \mid \mathrm{supp}(F) \subseteq T\}$ and let us extend it to a basis of $\langle F \rangle$: $f_1, f_2, \ldots, f_k, g_1, g_2, \ldots, g_\ell$. We will prove that $g_1[\backslash T], g_2[\backslash T], \ldots, g_\ell[\backslash T]$ is a basis of $\langle F[\backslash T]\rangle$, from which the lemma follows.

Consider an arbitrary element of $\langle F[\backslash T]\rangle$; it has the form $h[\backslash T]$ for some $h \in \langle F \rangle$ by the linearity of $[\backslash T]$. Let us write h as a linear combination of the basis elements: $h = \sum_{i=1}^{k} \alpha_i f_i + \sum_{i=1}^{m} \beta_i g_i$. Then $h[\backslash T] = \sum_{i=1}^{m} \beta_i g_i[\backslash T]$. Thus, $g_1[\backslash T], g_2[\backslash T], \ldots, g_\ell[\backslash T]$ generate $\langle F[\backslash T]\rangle$ and it remains to show they are linearly independent. Suppose that $\sum_{i=1}^{\ell} \gamma_i g_i[\backslash T] = 0$ for some scalars γ_i. Then $\mathrm{supp}(\sum_{i=1}^{\ell} \gamma_i g_i) \subseteq T$ and hence $\sum_{i=1}^{\ell} \gamma_i g_i \in \langle f_1, f_2, \ldots, f_k \rangle$, because $f_1, f_2, \ldots, f_k$ is a basis of $\{f \in \langle F \rangle \mid \mathrm{supp}(F) \subseteq T\}$. Since $f_1, f_2, \ldots, f_k, g_1, g_2, \ldots, g_\ell$ are linearly independent, all γ_i's have to be zero. □

Lemma 4.3. Let H be a minimally dangerous set and S be a strict subset of $\mathrm{supp}(H)$. Then $H[\backslash S]$ is not safe.

Proof. Because H is dangerous, $\dim\langle H \rangle = |H| > |\mathrm{supp}(H)|$. Since H is minimally dangerous, $|S| \ge \dim\{h \in \langle H \rangle \mid \mathrm{supp}(H) \subseteq S\}$. By Lemma 4.2, $\dim\langle H[\backslash S]\rangle = \dim\langle H \rangle - \dim\{h \in \langle H \rangle \mid \mathrm{supp}(H) \subseteq S\} > |\mathrm{supp}(H)| - |S|$. Hence a basis of $H[\backslash S]$ is dangerous. □

Lemma 4.4 (Uniqueness)**.** For any F its closure is unique.

Proof. Let S_1 and S_2 be two different closures of F. Then $S_1 \cap S_2$ is not a closure. Hence $\langle F[\backslash(S_1 \cap S_2)]\rangle$ contains a dangerous set and hence it contains a minimally dangerous set H. Since $\mathrm{supp}(H) \subseteq [m] \setminus (S_1 \cap S_2)$, either S_1 or S_2 does not contain $\mathrm{supp}(H)$. W.l.o.g. assume that S_1 does not contain $\mathrm{supp}(H)$. Then by Lemma 4.3, the set $H[\backslash S_1] = H[\backslash(S_1 \cap \mathrm{supp}(H))]$ is not safe. Since $H \subseteq \langle F[\backslash(S_1 \cap S_2)]\rangle$, we have $H[\backslash S_1] \subseteq \langle F[\backslash S_1]\rangle$. This is a contradiction since S_1 is a closure of F and so $\langle F[\backslash S_1]\rangle$ (and hence all its subsets) has to be safe. □

4.2 Closure size bound

Lemma 4.5. Let $S \subseteq \mathrm{Cl}(F)$ and let $\langle F[\backslash S]\rangle$ contain a minimally dangerous set H. Then $\mathrm{supp}(H) \subseteq \mathrm{Cl}(F)$.

Proof. Assume that $\mathrm{supp}(H) \not\subseteq \mathrm{Cl}(F)$, then $(\mathrm{Cl}(F) \cap \mathrm{supp}(H)) \subsetneq \mathrm{supp}(H)$. By Lemma 4.3, $H[\backslash(\mathrm{Cl}(F) \cap \mathrm{supp}(H))] = H[\backslash \mathrm{Cl}(F)]$ is not safe. Since $H \subseteq \langle F[\backslash S]\rangle$, we have $H[\backslash \mathrm{Cl}(F)] \subseteq \langle F[\backslash \mathrm{Cl}(F)]\rangle$ and this is a contradiction, since $\langle F[\backslash \mathrm{Cl}(F)]\rangle$ and all its subsets have to be safe by the definition of the closure. □

Corollary 4.6. Algorithm 4.1 computes $\mathrm{Cl}(F)$.

Proof. Each iteration of the loop increases S. Since $S \subseteq [m]$, the algorithm stops in a finite number of steps. Let $S' \subseteq [m]$ be the output of the algorithm.

Let us prove by induction that $S \subseteq \mathrm{Cl}(F)$ at every moment during the execution of Algorithm 4.1. Initially, $S := \emptyset$, so the assertion holds. The induction step follows by Lemma 4.5.

It follows that $S' \subseteq \mathrm{Cl}(F)$. We also know that $F[\backslash S']$ is safe. Thus, $S' = \mathrm{Cl}(F)$. □

Lemma 4.7 (Size bound)**.** $|\mathrm{Cl}(F)| + \dim\langle F[\backslash \mathrm{Cl}(F)]\rangle \leq \dim\langle F\rangle$, and hence $|\mathrm{Cl}(F)| \leq \dim\langle F\rangle$.

Proof. We prove by induction that during the execution of Algorithm 4.1 the following inequality holds: $|S| + \dim\langle F[\backslash S]\rangle \leq \dim\langle F\rangle$. Since the algorithm outputs $S = \mathrm{Cl}(F)$, we get the required inequality.

At the start of the algorithm, the inequality holds. Let us show that it holds after each step. Suppose the algorithm has found in $\langle F[\backslash S]\rangle$ a minimally dangerous set H with support T. As $H \subseteq \{f \in \langle F[\backslash S]\rangle \mid \mathrm{supp}(f) \subseteq T\}$, we have $\dim\{f \in \langle F[\backslash S]\rangle \mid \mathrm{supp}(f) \subseteq T\} \geq \dim\langle H\rangle > |T|$.

By Lemma 4.2, $\dim\{f \in \langle F[\backslash S]\rangle \mid \mathrm{supp}(f) \subseteq T\} = \dim\langle F[\backslash S]\rangle - \dim\langle F[\backslash(S \cup T)]\rangle$. Therefore, $\dim\langle F[\backslash(S \cup T)]\rangle < \dim\langle F[\backslash S]\rangle - |T|$.

Finally, $|S \cup T| + \dim\langle F[\backslash(S \cup T)]\rangle < |S| + |T| + \dim\langle F[\backslash S]\rangle - |T| = |S| + \dim\langle F[\backslash S]]\rangle \leq \dim\langle F\rangle$. In the last inequality, we use the inductive hypothesis. □

4.3 Closure increment

Note that it is possible for F to be safe and for $F[\backslash T]$ not to be safe. For example, if $F = \{x_{1,1} + x_{3,1}, x_{1,2} + x_{2,3}\}$ and $T = \{2, 3\}$, then $F[\backslash T] = \{x_{1,1}, x_{1,2}\}$ is dangerous.

Lemma 4.8. Let F be safe and $f_1, f_2, \ldots, f_k$ be linearly independent elements of $\langle F\rangle$ such that $\mathrm{supp}(f_1, f_2, \ldots, f_k) = T$ and $|T| = k$. Then the set $F[\backslash T]$ is safe.

Proof. We argue by contradiction. Let $g_1, g_2, \ldots, g_s$ be a linearly independent set from $\langle F[\backslash T]\rangle$ with support S and $|S| \leq s - 1$. Let $g'_1, g'_2, \ldots, g'_s$ be elements of $\langle F\rangle$ such that $g'_i[\backslash T] = g_i$.

Then $\mathrm{supp}(\{f_1, f_2, \ldots, f_k, g'_1, \ldots, g'_s\}) \subseteq S \cup T$ and the size of $S \cup T$ is at most $s + k - 1$. To get a contradiction we verify that all these forms are linearly independent. Indeed, assume that $\sum_{i=1}^{k} \alpha_i f_i + \sum_{j=1}^{s} \beta_i g'_i = 0$. By applying $[\backslash T]$ operator to this equation we get $\sum_{j=1}^{k} \beta_i g_i = 0$, hence $\beta_i = 0$ for $i \in [s]$. Since $f_1, f_2, \ldots, f_k$ are linearly independent, we get that $\alpha_i = 0$ for $i \in [k]$. □

Lemma 4.9 (Increment)**.** Let F be a set of linear forms and f be a linear form such that $\mathrm{Cl}(F \cup \{f\}) \neq \mathrm{Cl}(F)$. Then

$$\dim\langle F[\backslash \mathrm{Cl}(F)]\rangle - \dim\langle (F \cup \{f\})[\backslash \mathrm{Cl}(F \cup \{f\})]\rangle = |\mathrm{Cl}(F \cup \{f\})| - |\mathrm{Cl}(F)|.$$

Proof. Since $\mathrm{Cl}(F \cup \{f\})$ is strictly greater than $\mathrm{Cl}(F)$, the set $(F \cup f)[\backslash \mathrm{Cl}(F)]$ is not safe. Consider an arbitrary minimally dangerous set of linear forms $h_1, h_2, \ldots, h_k$ in $\langle (F \cup f)[\backslash \mathrm{Cl}(F)]\rangle$. For every $i \in [k]$, either $h_i \in \langle F[\backslash \mathrm{Cl}(F)]\rangle$ or $h_i \in f[\backslash \mathrm{Cl}(F)] + \langle F[\backslash \mathrm{Cl}(F)]\rangle$. We can assume that $h_1, h_2, \ldots, h_k$ have been chosen such that $I := \{i \in [k] \mid h_i \in f[\backslash \mathrm{Cl}(F)] + \langle F[\backslash \mathrm{Cl}(F)]\rangle\}$ has the minimum cardinality. We know $|I| \geq 1$, otherwise $\mathrm{Cl}(F)$ is not the correct closure. Moreover, it is easy to see that $|I| = 1$. Indeed, if $i_1 \neq i_2 \in I$, then we can replace the form h_{i_1} in $h_1, h_2, \ldots, h_k$ with $h_{i_1} + h_{i_2}$; this alters neither the linear independence nor the support, but $h_{i_1} + h_{i_2} \in \langle F[\backslash \mathrm{Cl}(F)]\rangle$, a contradiction with the minimality of $|I|$. W.l.o.g assume that $h_i \in \langle F[\backslash \mathrm{Cl}(F)]\rangle$ for $i \in [k-1]$ and $h_k \in f[\backslash \mathrm{Cl}(F)] + \langle F[\backslash \mathrm{Cl}(F)]\rangle$.

Let $T = \mathrm{supp}(h_1, h_2, \ldots, h_k)$, then $T \subseteq [m] \backslash \mathrm{Cl}(F)$. Note that T has size exactly $k-1$, since if the support of the set $h_1, h_2, \ldots, h_k$ were smaller, then the set $h_1, h_2, \ldots, h_{k-1}$ would be dangerous and in $\langle F[\backslash \mathrm{Cl}(F)]\rangle$.

Claim 4.10. $f[\backslash(\mathrm{Cl}(F) \cup T)] \in \langle F[\backslash(\mathrm{Cl}(F) \cup T)]\rangle$.

Proof. Let us apply the linear operator $[\backslash T]$ to the statement $f[\backslash \mathrm{Cl}(F)] + h_k \in \langle F[\backslash \mathrm{Cl}(F)]\rangle$. Since $h_k[\backslash T] = 0$, we get $f[\backslash(\mathrm{Cl}(F) \cup T)] \in \langle F[\backslash(\mathrm{Cl}(F) \cup T)]\rangle$. □

Claim 4.11. $T = \mathrm{Cl}(F \cup \{f\}) \setminus \mathrm{Cl}(F)$.

Proof. By monotonicity, $\mathrm{Cl}(F) \subseteq \mathrm{Cl}(F \cup \{f\})$. Since $h_1, h_2, \ldots, h_k$ is minimally dangerous, it follows by Lemma 4.5 that $T \subseteq \mathrm{Cl}(F \cup \{f\})$. The set $h_1, h_2, \ldots, h_{k-1}$ is safe, hence $|\mathrm{supp}(\{h_1, h_2, \ldots, h_{k-1}\})| = k-1$, and so $\mathrm{supp}(\{h_1, h_2, \ldots, h_{k-1}\}) = T$. Consequently, Lemma 4.8 applied to $F[\setminus \mathrm{Cl}(F)]$ and $h_1, h_2, \ldots, h_{k-1}$ shows that $F[\setminus(\mathrm{Cl}(F) \cup T)]$ is safe. By Claim 4.10, $\langle F[\setminus(\mathrm{Cl}(F) \cup T)]\rangle = \langle (F \cup \{f\})[\setminus(\mathrm{Cl}(F) \cup T)]\rangle$, hence $(F \cup \{f\})[\setminus(\mathrm{Cl}(F) \cup T)]$ is also safe. Thus, $\mathrm{Cl}(F \cup \{f\}) = \mathrm{Cl}(F) \cup T$. As $T \subseteq [m] \setminus \mathrm{Cl}(F)$, the claim follows. □

Consider the space $\{g \in \langle F[\setminus \mathrm{Cl}(F)]\rangle \mid \mathrm{supp}(g) \subseteq T\}$; by the definition of closure its dimension is at most $|T|$, but as it contains all $h_1, h_2, \ldots, h_{k-1}$, the dimension is exactly $|T|$.

By Lemma 4.2, $\dim\langle F[\setminus \mathrm{Cl}(F)]\rangle - \dim\langle F[\setminus(\mathrm{Cl}(F) \cup T)]\rangle = \dim\{g \in \langle F[\setminus \mathrm{Cl}(F)]\rangle \mid \mathrm{supp}(g) \subseteq T\} = |T|$. By Claim 4.10, $\langle F[\setminus(\mathrm{Cl}(F) \cup T)]\rangle = \langle (F \cup \{f\})[\setminus(\mathrm{Cl}(F) \cup T)]\rangle = \langle (F \cup \{f\})[\setminus \mathrm{Cl}(F \cup \{f\})]\rangle$. Thus,

$$\dim\langle F[\setminus \mathrm{Cl}(F)]\rangle - \dim\langle (F \cup \{f\})\,[\setminus \mathrm{Cl}(F \cup \{f\})]\rangle = |T| = |\mathrm{Cl}(F \cup \{f\})| - |\mathrm{Cl}(F)|.$$

□

5 Locally consistent linear systems

In this section we take advantage of the fact that X is not just a set of variables indexed by two parameters, but that X is the set of variables of the formula BPHP^m_n, where $n = 2^\ell$. We will exploit the semantics of these variables, described in Section 2.5.

Let Φ be a linear system with variables from X and F be a set of linear forms from the left-hand sides of these equations. An assignment $\sigma : X \to \{0,1\}$ is called a *locally injective solution* of Φ if σ satisfies Φ and $\tilde{\sigma}$ is injective on $\mathrm{Cl}(F)$, where $\tilde{\sigma}$ is defined in Section 2.5. We say that Φ is *locally consistent* if it has a locally injective solution.

Consider some examples of locally consistent linear systems:

1. An empty linear system (i.e. the negation of the empty clause) is locally consistent.

2. The negation of any clause of BPHP^m_n is not locally consistent. Indeed, the system looks like $\bigwedge_{j=1}^{\ell}(x_{i_1,j} = a_j) \wedge \bigwedge_{j=1}^{\ell}(x_{i_2,j} = a_j)$. It is easy to see that $\mathrm{Cl}(x_{i_1,1}, x_{i_1,2}, \ldots, x_{i_1,\ell}, x_{i_2,1}, x_{i_2,2}, \ldots, x_{i_2,\ell}) = \{i_1, i_2\}$. So there is no locally injective solution.

Proposition 5.1. Let Φ and Ψ be linear systems and suppose that every equation in Φ is implied by Ψ. If Ψ is locally consistent, then Φ is also locally consistent.

Proof. Let F and G be the sets of linear forms of systems Φ and Ψ respectively. Then by Lemma 2.1, $F \subseteq \langle G\rangle$, hence by the properties of closure $\mathrm{Cl}(F) \subseteq \mathrm{Cl}(G)$. Thus a locally injective solution of Ψ is also a locally injective solution of Φ. □

Corollary 5.2. If a linear clause C is a weakening of a linear clause D and $\neg C$ is locally consistent, then $\neg D$ is also locally consistent.

The next two lemmas tell us how to keep local consistency when a new equation is added to a system.

Lemma 5.3. Let Φ be a locally consistent linear system with the set of linear forms F and let f be a linear form. Suppose that $\mathrm{Cl}(F \cup \{f\}) = \mathrm{Cl}(F)$ and $f[\setminus \mathrm{Cl}(F)] \notin \langle F[\setminus \mathrm{Cl}(F)]\rangle$. Then for every $a \in \{0,1\}$ the system $\Phi \wedge (f = a)$ is locally consistent.

Proof. Let us fix $a \in \{0,1\}$. Let ρ be the restriction of a locally injective solution of Φ to the variables with support $\mathrm{Cl}(F)$. The system $\Phi|_\rho$ is satisfiable and since $f[\setminus \mathrm{Cl}(F)] \notin \langle F[\setminus \mathrm{Cl}(F)]\rangle$, the system $(\Phi \wedge (f = a))|_\rho$ is also satisfiable. Therefore there is a solution of $\Phi \wedge (f = a)$ that coincides with ρ on the variables with support in $\mathrm{Cl}(F)$. Thus, $\Phi \wedge (f = a)$ is locally consistent. □

Lemma 5.4. Let Φ be a locally consistent linear system with a set of forms F, and let f be a linear form. Suppose that $\mathrm{Cl}(F \cup \{f\}) \neq \mathrm{Cl}(F)$ and $|\mathrm{Cl}(F \cup \{f\})| \leq 2^{\ell-2}$. Then there exists $a \in \{0,1\}$ such that the system $\Phi \wedge (f = a)$ is locally consistent.

Proof. It is sufficient to show that there exists a solution τ of Φ such that $\tilde{\tau}$ is injective on $\mathrm{Cl}(F \cup \{f\})$. Then, setting a to the value of f at τ, the lemma follows.

Let σ be a solution of Φ such that $\tilde{\sigma}$ is injective on $\mathrm{Cl}(F)$. Let ρ be the restriction of σ to the variables with support in $\mathrm{Cl}(F)$. It follows that the system $\Phi|_\rho$ is satisfiable. To satisfy $\Phi|_\rho$ it is sufficient to satisfy a maximal linearly independent set of its equations. Note that the set of linear forms of $\Phi|_\rho$ is exactly $F[\setminus \mathrm{Cl}(F)]$, hence it is safe. Let M be the matrix of the right-hand side of some maximal linearly independent part of $\Phi|_\rho$ and let s be the number of rows of M. By Theorem 3.1 the matrix M contains s linearly independent columns such that the set Z of variables corresponding to these columns does not contain two variables mentioning the same pigeon. So in order to satisfy $\Phi|_\rho$ we can arbitrarily fix all variables except those in Z, and then the values of the variables in Z are uniquely determined.

Our goal is to show that there exists a solution γ of Φ extending ρ such that $\tilde{\gamma}$ is injective on $\mathrm{Cl}(F \cup \{f\})$. We will construct an assignment τ of the variables X such that τ extends ρ and for every assignment τ' that differs from τ only on a subset of the variables from Z, $\widetilde{\tau'}$ is injective on $\mathrm{Cl}(F \cup \{f\})$. By the above remark, there exists a solution γ of Φ that differs from τ only on a subset of the variables from Z; this γ satisfies all the requirements.

We define τ such that it coincides with ρ on the variables with support in $\mathrm{Cl}(F)$. We define τ arbitrarily on the variables with support in $[m] \setminus \mathrm{Cl}(F \cup \{f\})$. Let $\mathrm{Cl}(F \cup \{f\}) \setminus \mathrm{Cl}(F) = \{s_1, s_2, \ldots, s_k\}$. By induction on i we define τ on the variables with support in $\{s_1, s_2, \ldots, s_i\}$ such that for every assignment τ' that differs from τ only on a subset of the variables from Z, $\widetilde{\tau'}$ is injective on $\mathrm{Cl}(F) \cup \{s_1, s_2, \ldots, s_i\}$. The base of induction is $i = 0$ and there is nothing to prove. For the induction step from $i-1$ to i, we identify the holes for s_i that can lead to a collision with $\mathrm{Cl}(F) \cup \{s_1, s_2, \ldots, s_{i-1}\}$ under a change of values to Z-variables:

- There are $|\mathrm{Cl}(F)|$ holes occupied by ρ. A hole for the pigeon s_i should be different from these holes even if the values of some variables from Z are flipped. Since there is at most one Z-variable corresponding to s_i, ρ forbids at most $2|\mathrm{Cl}(F)|$ holes to pigeon s_i;

- There are $i-1$ holes occupied by pigeons $s_1, \ldots, s_{i-1}$. A hole for the pigeon s_i should be different from these holes even if the values of some variables from Z are flipped. Since there is at most one variable from Z for each pigeon in $\{s_1, \ldots, s_{i-1}\} \cup \{s_i\}$, the pigeons $s_1, \ldots, s_{i-1}$ prohibit at most $4(i-1)$ holes to pigeon s_i.

So there are at most $2|\mathrm{Cl}(F)| + 4(i-1) \leq 2|\mathrm{Cl}(F)| + 4(k-1) \leq 4(\mathrm{Cl}(F \cup \{f\}) - 1) < 2^\ell$ forbidden holes, hence there is at least one non-forbidden hole that we can use for pigeon s_i. □

5.1 Rank lower bound

Theorem 5.5. Any $\mathrm{Res}(\oplus)$ refutation of $\mathrm{BPHP}^m_{2^\ell}$ contains a clause C such that the rank of $\neg C$ is at least $2^{\ell-2}$.

Proof. Given a $\mathrm{Res}(\oplus)$ refutation of $\mathrm{BPHP}^m_{2^\ell}$, we will construct a path in the refutation graph of $\mathrm{BPHP}^m_{2^\ell}$. We start the path at the source and we continue the path as long as it is possible to satisfy the following invariant: for every clause C on our path, $\neg C$ is locally consistent. The empty clause at the source satisfies the invariant.

Consider a linear clause C in the refutation graph of $\mathrm{BPHP}^m_{2^\ell}$. Let linear clauses D and E be direct successors of C. We will show that if $\neg C$ is locally consistent and $\mathrm{rk}(\neg C) < 2^{\ell-2}$, then $\neg E$ or $\neg D$ is also locally consistent. Since the negations of the clauses of $\mathrm{BPHP}^m_{2^\ell}$ are not locally consistent, the constructed path can finish only in a clause C such that $|\mathrm{rk}(\neg C)| \geq 2^{\ell-2}$.

By the definition of the refutation graph, there are linear clauses D' and E' and a linear equation $f = a$ such that $D' \vee (f = a)$ is a weakening of D and $E' \vee (f = 1-a)$ is a weakening of E and $C = D' \vee E'$. Hence, the clause $C \vee (f = a)$ is a weakening of $D' \vee (f = a)$ and $C \vee (f = 1-a)$ is a weakening of $E' \vee (f = 1-a)$. So by Corollary 5.2 it is sufficient to prove that at least one of the two systems $\neg C \wedge (f = 1)$ and $\neg C \wedge (f = 0)$ is locally consistent.

There are two cases. In the first case, $\mathrm{Cl}(L(C) \cup \{f\}) = \mathrm{Cl}(L(C))$. (Recall that $L(C)$ denotes the set of linear forms that appear in C.) Let σ be a locally injective solution of $\neg C$. Then σ is necessarily a locally injective solution of either $\neg C \wedge (f = 0)$ or $\neg C \wedge (f = 1)$.

In the second case, $\mathrm{Cl}(L(C)) \subsetneq \mathrm{Cl}(L(C) \cup \{f\})$. Note that $\mathrm{rk}(\neg C \wedge (f = 0))$ is at most $2^{\ell-2}$, hence, by Lemma 4.7, $|\,\mathrm{Cl}(L(C) \cup \{f\})| \le 2^{\ell-2}$. By Lemma 5.4, $\neg C \wedge (f = 1)$ or $\neg C \wedge (f = 0)$ is locally consistent. □

5.2 Tree-like lower bound

We consider a Prover-Delayer game with an unsatisfiable CNF φ. There are two players: Prover and Delayer. They have a board on which they write linear equations in the variables of φ. The game starts with an empty board.

The game consists of a sequence of moves, each of which has the following form. Prover writes a linear form f on the board. Delayer responds in one of two ways. Either Delayer chooses $\alpha \in \{0, 1\}$, completing Prover's form f on the board to the equation $f = \alpha$, and the move is complete; or Delayer asks Prover to choose α herself, for which Delayer earns a coin, Prover chooses α and completes the form f on the board to the equation $f = \alpha$, and the move is complete. The game ends when the system of equations on the board contradicts a clause of φ. Delayer's goal in this game is to earn as many coins as possible. The following is not difficult to prove.

Lemma 5.6 ([30])**.** If for an unsatisfiable formula φ there is a strategy for Delayer that guarantees him to earn at least t coins, then the size of any tree-like Res($\oplus$) refutation of φ is at least 2^t.

In our strategy for $\mathrm{BPHP}^m_{2^\ell}$, Delayer will try to keep the linear system on the board locally consistent. The following lemma shows that as long as this is the case, the game cannot end.

Lemma 5.7. Let $\ell > 1$. If Φ is locally consistent, then for any clause C of $\mathrm{BPHP}^m_{2^\ell}$, Φ does not contradict C (i.e. there is a solution of Φ that satisfies C).

Proof. Consider some clause C of the formula $\mathrm{BPHP}^m_{2^\ell}$; it says that either pigeon i is not in hole a, or pigeon j is not in hole a.

Let F be the set of linear forms of Φ. If both i and j are in $\mathrm{Cl}(F)$, then a locally injective solution of Φ satisfies C, hence Φ does not contradict C.

Now assume that $i \notin \mathrm{Cl}(F)$ or $j \notin \mathrm{Cl}(F)$. W.l.o.g. assume that $i \notin \mathrm{Cl}(F)$. Suppose that Φ contradicts C; in particular, this means that the system Φ semantically implies the equations $x_{i,1} = a_1, x_{i,2} = a_2, \dots, x_{i,\ell} = a_\ell$. By Lemma 2.1, these equations are linear combinations of the equations from Φ, hence $x_{i,1}, x_{i,2}, \dots, x_{i,l} \in \langle F \rangle$. Since $i \notin \mathrm{Cl}(F)$, $x_{i,1}, x_{i,2}, \dots, x_{i,\ell} \in \langle F[\setminus \mathrm{Cl}(F)] \rangle$. But the set $\{x_{i,1}, \dots, x_{i,\ell}\}$ is dangerous whenever $\ell > 1$. So we get a contradiction with the definition of the closure. □

Theorem 5.8. The size of any tree-like Res($\oplus$) refutation of $\mathrm{BPHP}^m_{2^\ell}$ is at least $2^{2^{\ell-2}}$.

Proof. If $\ell = 1$, the statement is trivial; so we assume that $\ell > 1$. Let us describe a strategy for Delayer. Let F be a dynamic variable that equals the set of linear forms written on the board. Delayer will keep a locally consistent system on the board as long as possible. Note that the board is empty at the start and the empty system is locally consistent. By Lemma 5.7, the game cannot be over while the system on the board is locally consistent.

The strategy of Delayer is the following:

- If $\mathrm{Cl}(F) = \mathrm{Cl}(F \cup \{f\})$ and $f[\setminus \mathrm{Cl}(F)]$ is in $\langle F[\setminus \mathrm{Cl}(F)] \rangle$, then Delayer chooses a value $\alpha \in \{0, 1\}$ such that $f = \alpha$ is satisfied by some locally injective solution of the system on the board. Trivially, the resulting system is locally consistent.

- If $\mathrm{Cl}(F) = \mathrm{Cl}(F \cup \{f\})$ and $f[\setminus \mathrm{Cl}(F)]$ is not in $\langle F[\setminus \mathrm{Cl}(F)]\rangle$, then Delayer earns a coin by letting Prover choose α. By Lemma 5.3 the invariant will hold for the new system on the board.
- In the last case, $\mathrm{Cl}(F \cup \{f\}) \setminus \mathrm{Cl}(F) = T$ for some $T \neq \emptyset$. If $|\mathrm{Cl}(F \cup \{f\})| \leq 2^{\ell-2}$, then by Lemma 5.4 there exists $a \in \{0,1\}$ such the answer a leads to a locally consistent system on the board, and Delayer uses this answer. The only case where Delayer can't maintain local consistency is when $|\mathrm{Cl}(F \cup \{f\})| > 2^{\ell-2}$. In this case, he answers arbitrarily and then "gives up".

We claim that at any time before Delayer gives up, the quantity $|\mathrm{Cl}(F)| + \dim\langle F[\setminus \mathrm{Cl}(F)]\rangle$ records the number of coins he has earned so far. We prove this by induction on the number of moves made. The base corresponds to the start of the game and the statement is trivial.

If $\mathrm{Cl}(F) = \mathrm{Cl}(F \cup \{f\})$ and $f[\setminus \mathrm{Cl}(F)]$ is in $\langle F[\setminus \mathrm{Cl}(F)]\rangle$, then Delayer does not earn a coin, $\dim\langle F[\setminus \mathrm{Cl}(F)]\rangle$ and $\mathrm{Cl}(F)$ are not changed.

If $\mathrm{Cl}(F) = \mathrm{Cl}(F \cup \{f\})$ and $f[\setminus \mathrm{Cl}(F)]$ is not in $\langle F[\setminus \mathrm{Cl}(F)]\rangle$, then Delayer earns a coin, $\dim\langle F[\setminus \mathrm{Cl}(F)]\rangle$ increases by one, and $\mathrm{Cl}(F)$ does not change.

If $T = \mathrm{Cl}(F \cup \{f\}) \setminus \mathrm{Cl}(F) \neq \emptyset$, then Delayer does not earn a coin, $\mathrm{Cl}(F)$ increases by $|T|$ and, by Lemma 4.9, $\dim\langle F[\setminus \mathrm{Cl}(F)]\rangle$ decreases by $|T|$. This finishes the inductive step.

Thus, at the time Delayer gives up, he has earned at least $2^{\ell-2}$ coins. Lemma 5.6 completes the proof of the theorem. □

6 Random path in a refutation graph

We consider a random process on a linear branching program associated with a refutation graph of the formula $\mathrm{BPHP}^m_{2^\ell}$. We take a uniformly random full assignment $\boldsymbol{\sigma}$ to the variables X and take t steps starting from the source. At each step, we go along the edge labeled with an equation satisfied by $\boldsymbol{\sigma}$; if we come to a sink earlier than in t steps, we just remain there. Our nearest goal is to prove that with significant probability in t steps we reach a vertex labeled with a clause C such that $\neg C$ is locally consistent. Moreover, we will show that with significant probability $\boldsymbol{\sigma}$ is a locally injective solution of $\neg C$.

We will prove the following lemma in Subsection 6.1.

Lemma 6.1. Suppose that Φ is a system of linear equations in the variables X and denote by F the set of linear forms from the left-hand side of Φ. Let f be a linear form, $a \in \{0,1\}$, and assume that $|\mathrm{Cl}(F \cup \{f\})| \leq t$. Consider a random full assignment $\boldsymbol{\sigma}$. Then

$$\Pr[\tilde{\boldsymbol{\sigma}} \text{ is injective on } \mathrm{Cl}(F) \text{ but not injective on } \mathrm{Cl}(F \cup \{f\}) \mid \boldsymbol{\sigma} \text{ satisfies } \Phi \wedge (f = a)] \leq \frac{6t^2}{n}.$$

Lemma 6.2. Let $\boldsymbol{\sigma}$ be a uniformly random full assignment of variables of $\mathrm{BPHP}^m_{2^\ell}$. Consider the refutation graph of a $\mathrm{Res}(\oplus)$ refutation of $\mathrm{BPHP}^m_{2^\ell}$ and the linear branching program associated with it. Consider a path of length t in the linear branching program corresponding to $\boldsymbol{\sigma}$; if the path reaches a sink, it stops there. Assume that the path ends in a node labeled with a clause $\boldsymbol{C}$. Then, the probability that $\tilde{\boldsymbol{\sigma}}$ is not injective on $\mathrm{Cl}(L(\boldsymbol{C}))$ is at most $\frac{6t^3}{n}$.

Proof. Let $\boldsymbol{\Phi}_i$ denote the conjunction of the linear equations labeling the edges of the path of length i in the linear branching program corresponding to $\boldsymbol{\sigma}$ (if the path reaches a sink earlier than in i steps, it stops there). Denote by $\boldsymbol{F}_i$ the set of linear forms of $\boldsymbol{\Phi}_i$.

$$\begin{aligned}
&\Pr[\boldsymbol{\sigma} \text{ is not injective on } \mathrm{Cl}(\boldsymbol{F}_t)] \\
&\qquad = \Pr[\exists j \in [0; t-1] \text{ such that } \boldsymbol{\sigma} \text{ is not injective on } \mathrm{Cl}(\boldsymbol{F}_{j+1}) \text{ but injective on } \mathrm{Cl}(\boldsymbol{F}_j)] \\
&\qquad \le \sum_{j=0}^{t-1} \Pr[\boldsymbol{\sigma} \text{ is not injective on } \mathrm{Cl}(\boldsymbol{F}_{j+1}) \text{ but injective on } \mathrm{Cl}(\boldsymbol{F}_j)] \\
&\qquad \le t \cdot \max_{0\le j\le t-1} \Pr[\boldsymbol{\sigma} \text{ is not injective on } \mathrm{Cl}(\boldsymbol{F}_{j+1}) \text{ but injective on } \mathrm{Cl}(\boldsymbol{F}_j)].
\end{aligned}$$

Let P_i denote the set of paths from the source of the linear branching program of length i (or shorter if they end in a sink earlier). For a path π from the source, we denote by $\Phi^{(\pi)}$ the system of equations corresponding to π (i.e. the conjunction of the equations labeling the edges of π).

$$\begin{aligned}
&\Pr[\boldsymbol{\sigma} \text{ is not injective on } \mathrm{Cl}(\boldsymbol{F}_{j+1}) \text{ but injective on } \mathrm{Cl}(\boldsymbol{F}_j)] \\
&\qquad = \sum_{\pi\in P_{j+1}} \Pr[\boldsymbol{\sigma} \text{ is not injective on } \mathrm{Cl}(\boldsymbol{F}_{j+1}) \text{ but injective on } \mathrm{Cl}(\boldsymbol{F}_j) \mid \boldsymbol{\sigma} \text{ satisfies } \Phi^{(\pi)}] \\
&\qquad\qquad \cdot \Pr[\boldsymbol{\sigma} \text{ satisfies } \Phi^{(\pi)}] \overset{\text{(Lemma 6.1)}}{\le} \frac{6t^2}{n} \sum_{\pi\in P_{j+1}} \Pr[\boldsymbol{\sigma} \text{ satisfies } \Phi^{\pi}] = \frac{6t^2}{n}.
\end{aligned}$$

In the inequality we used Lemma 6.1 on $\boldsymbol{F}_j$; the hypothesis of the lemma is satisfied thanks to Lemma 4.7: $\mathrm{Cl}(\boldsymbol{F}_{j+1}) \le \dim\langle \boldsymbol{F}_{j+1}\rangle \le j+1 \le t$.

Note that $\boldsymbol{\sigma}$ satisfies $\boldsymbol{\Phi}_t$, hence by Lemmas 2.3 and 2.1, $L(\boldsymbol{C}) \subseteq \langle \boldsymbol{F}_t\rangle$, and therefore $\mathrm{Cl}(L(\boldsymbol{C})) \subseteq \mathrm{Cl}(\boldsymbol{F}_t)$. Thus

$$\Pr[\boldsymbol{\sigma} \text{ is not injective on } \mathrm{Cl}(L(\boldsymbol{C}))] \le \Pr[\boldsymbol{\sigma} \text{ is not injective on } \mathrm{Cl}(\boldsymbol{F}_t)] \le \frac{6t^3}{n}.$$

□

6.1 Proof of Lemma 6.1

Proof of Lemma 6.1. If $6t \ge n$, then the statement of the lemma is trivial; so we assume that $6t < n$.

If $\mathrm{Cl}(F \cup \{f\}) = \mathrm{Cl}(F)$, then the probability we have to estimate is zero. We therefore assume that $\mathrm{Cl}(F\cup\{f\}) \setminus \mathrm{Cl}(F) = T \neq \emptyset$.

Consider a partial assignment ρ such that $\tilde{\rho}$ injectively maps its domain $\mathrm{Cl}(F)$ to $[2^\ell]$ and ρ can be extended to a solution of $\Phi \wedge (f = a)$. We are going to estimate the probability conditioned on ρ:

$$\begin{aligned}
&\Pr[\tilde{\boldsymbol{\sigma}} \text{ is not injective on } \mathrm{Cl}(F\cup\{f\}) \mid \boldsymbol{\sigma} \text{ satisifies } \Phi \wedge (f=a) \text{ and } \boldsymbol{\sigma} \text{ extends } \rho] \\
&\qquad = 1 - \Pr[\tilde{\boldsymbol{\sigma}} \text{ is injective on } \mathrm{Cl}(F\cup\{f\}) \mid \boldsymbol{\sigma} \text{ satisfies } \Phi \wedge (f=a) \text{ and } \boldsymbol{\sigma} \text{ extends } \rho].
\end{aligned}$$

Notice that

$$\begin{aligned}
&\Pr[\tilde{\boldsymbol{\sigma}} \text{ is injective on } \mathrm{Cl}(F\cup\{f\}) \mid \boldsymbol{\sigma} \text{ satisifes } \Phi \wedge (f=a) \text{ and } \boldsymbol{\sigma} \text{ extends } \rho] \\
&\qquad = \Pr[\tilde{\boldsymbol{\sigma}} \text{ is injective on } T \text{ and } \tilde{\boldsymbol{\sigma}}(T) \cap \tilde{\rho}(\mathrm{Cl}(F)) = \emptyset \mid \boldsymbol{\sigma} \text{ satisfies } (\Phi \wedge (f=a))\,|_\rho].
\end{aligned}$$

The system $\Phi|_\rho$ is satisfiable, so to solve $\Phi|_\rho$ it is sufficient to consider a maximal linearly independent set of its equations. The set of linear forms from the left-hand side of the equations of $\Phi|_\rho$ is $F[\setminus \mathrm{Cl}(F)]$, hence it is safe. Let M be the matrix of some maximal linearly independent part of $\Phi|_\rho$ and let M contain

k rows. By Theorem 3.1 the matrix M contains k linearly independent columns corresponding to a set of variables Z such that Z does not contain two variables mentioning the same pigeon. However, we have the system $(\Phi \wedge (f = a))|_\rho$ and to satisfy it we have to satisfy both $\Phi|_\rho$ and $(f = a)|_\rho$.

Let us define a matrix M' corresponding to a maximal linear independent part of the system $(\Phi\wedge(f = a))|_\rho$ as follows. Note that $f[\setminus \mathrm{Cl}(F)] \notin \langle F[\setminus \mathrm{Cl}(F)]\rangle$, otherwise $\langle (F \cup \{f\})[\setminus \mathrm{Cl}(F)]\rangle = \langle F[\setminus \mathrm{Cl}(F)]\rangle$ is safe and therefore $\mathrm{Cl}(F) = \mathrm{Cl}(F \cup \{f\})$. Hence $(f = a)|_\rho$ does not follow from $\Phi|_\rho$. Let M' be obtained from M by adding to M one row corresponding to the equality $(f = a)|_\rho$. The rank of M' equals $k + 1$, so in M' we can take the k linearly independent columns corresponding to the variables in Z and add to this set a $(k+1)$th column from M' such that all these columns are linearly independent. Let Z' be the set of variables corresponding to these $k + 1$ columns. While no two different variables in Z mention the same pigeon, the variable corresponding to the $(k + 1)$th column may mention the same pigeon as some variable in Z. To satisfy $(\Phi \wedge (f = a))|_\rho$ we can assign values to all variables outside Z' arbitrarily and then the values of the variables from Z' are uniquely determined.

The probability we are interested in is over a random solution of the system $(\Phi \wedge (f = a))|_\rho$. Let us assume that we first randomly assign values to all variables with support in $[m] \setminus \mathrm{Cl}(F)$ (including values of the variables in Z', which are not under our control) and then possibly change the values of the variables in Z' to satisfy $(\Phi \wedge (f = a))|_\rho$. We stress that we will change at most one bit for $k - 1$ pigeons and at most two bits for one pigeon (or, if no two variables in Z' share a pigeon, we will change at most one bit for $k + 1$ pigeons).

By the assumptions of the lemma, $|\,\mathrm{Cl}(F)| \le |\,\mathrm{Cl}(F \cup \{f\})| \le t$, hence $|T| \le t$. Let us order the set T such that the pigeon with two variables from Z' (if there is such a pigeon in T) comes first. Considering the pigeons from T in this order, for the first pigeon in T we have at most $4t$ forbidden holes: at most t holes are occupied by ρ and we might need to flip two bits corresponding to the two variables in Z' mentioning this pigeon. For the second pigeon, we have at most $2(t + 4)$ forbidden holes: t holes are occupied by ρ and at most four holes are reserved for the first pigeon; the factor 2 corresponds to one bit of the second pigeon that may be flipped in case a variable in Z' mentions the second pigeon. For the third pigeon we have at most $2(t + 6)$ forbidden holes, etc.

Thus, the probability that all pigeons from T in a random solution of $(\Phi \wedge (f = a))|_\rho$ are in different holes that are not used by ρ is at least

$$\frac{n-4t}{n}\prod_{k=2}^{t}\frac{n-2(t+4+2(k-2))}{n} \ge \left(\frac{n-6t}{n}\right)^t \ge 1-\frac{6t^2}{n}.$$

In the last inequality, we use Bernoulli's inequality $(1+x)^n \ge 1+nx$, which holds for every integer $n \ge 1$ and real number $x \ge -1$.

So we have proved an upper bound on the probability conditioned on ρ:

$$\Pr[\tilde{\boldsymbol{\sigma}} \text{ is not injective on } \mathrm{Cl}(F \cup \{f\}) \text{ but injective on } \mathrm{Cl}(F) \mid \boldsymbol{\sigma} \text{ satisfies } \Phi \wedge (f = a) \text{ and } \boldsymbol{\sigma} \text{ extends } \rho] \le \frac{6t^2}{n}.$$

Since the condition event we need is partitioned according to ρ into the condition events as in the last inequality (including ρ's with non-injective $\tilde{\rho}$, for which the estimated probability is 0), the lemma follows. □

7 Rank lower bound for regular $\mathrm{Res}(\oplus)$ refutations

In this section we assume that X is the set of variables of $\mathrm{BPHP}^{2^\ell+1}_{2^\ell}$. That is, the number of pigeons exceeds the number of holes exactly by one.

Lemma 7.1. Let a linear system Φ in variables from X be locally consistent and contain t linearly independent equations, where $t < 2^{\ell-2}$. Let $K \subseteq [n+1]$ and $|K| \geq t+1$. Then there is a solution σ of Φ such that $\tilde{\sigma}$ is injective on $[n+1] \setminus K$.

Proof. Since Φ is locally consistent, there exists a locally injective assignment, and let ρ be its restriction to the variables with support in $\operatorname{Cl}(F)$. Note that $\tilde{\rho}$ is injective on its domain and $\Phi|_\rho$ is satisfiable.

The set of linear forms $F[\setminus \operatorname{Cl}(F)]$ is safe by the definition of the closure. Let $h = \dim\langle F[\setminus \operatorname{Cl}(F)]\rangle$. Consider h linearly independent forms from $F[\setminus \operatorname{Cl}(F)]$: $f_1, f_2, \dots f_h$. Note that $\Phi|_\rho$ contains equations with linear forms $f_1, f_2, \dots f_h$ and all other equations of $\Phi|_\rho$ follow from these equations. Let M be the coefficient matrix of $f_1, f_2, \dots, f_h$. By the definition of $\operatorname{Cl}(F)$, the set $f_1, f_2, \dots, f_h$ is safe, so by Theorem 3.1, there are h linear independent columns in M that correspond to h different pigeons. Let us denote by Z the set of variables corresponding to these h columns. To satisfy the system $\Phi|_\rho$, one can arbitrarily choose values for all variables except those in Z, and then the values of the variables in Z are uniquely determined. We are going to describe a solution σ of Φ such that $\tilde{\sigma}$ is injective on $[n+1] \setminus K$ and σ extends ρ.

Pigeons from $\operatorname{Cl}(F)$ are already in distinct holes, and we assume that their holes are reserved. First, we will reserve holes for the h pigeons that have a corresponding variable in Z. For each such pigeon, we will reserve two holes that differ only in the ith bit, where i is the second index of the variable in Z mentioning the pigeon. We do it inductively. Assume that we have already reserved pairs of holes for the first j pigeons. Let us treat the $(j+1)$th pigeon. Let $i \in [\ell]$ be the second index of the variable in Z mentioning this pigeon. There are $2^{\ell-1}$ pairs of strings of length ℓ such that the strings in each pair differ exactly in the ith bit. Some elements of some pairs may have been reserved earlier; we will call such pairs touched. The number of touched pairs is at most $|\operatorname{Cl}(F)| + 2j < |\operatorname{Cl}(F)| + 2h \leq 2t < 2^{\ell-1}$; in the second inequality we used that by Lemma 4.7, $|\operatorname{Cl}(F)| + h \leq t$. So we may reserve an untouched pair for the $(j+1)$th pigeon.

So far we have reserved at most $2h + |\operatorname{Cl}(F)|$ holes for $h + |\operatorname{Cl}(F)|$ pigeons, but possibly some of them are in K. Let k_1 denote the number of such pigeons. Then we cancel the reservation for these k_1 pigeons, so now there are at most $2h + |\operatorname{Cl}(F)| - k_1$ reserved holes, hence there are at least $n - 2h - |\operatorname{Cl}(F)| + k_1$ holes that are not reserved. And there are $(n+1) - h - |\operatorname{Cl}(F)| - (|K| - k_1) = n - 2h - |\operatorname{Cl}(F)| + k_1 + (1 + h - |K|)$ pigeons that are neither in K nor in $\operatorname{Cl}(F) \cup \operatorname{supp}(Z)$. Since $|K| \geq t+1 \geq h+1$, we now send these pigeons to the unreserved holes injectively. All pigeons in $\operatorname{Cl}(F)$ are sent to holes by $\tilde{\rho}$ (which respects the reserved holes for all pigeons in $\operatorname{Cl}(F) \setminus K$). Send each pigeon from $\operatorname{supp}(Z) \setminus K$ to any hole from its reserved pair, and send all pigeons unsent so far to arbitrary holes. Now, adjust the values of the variables Z to obtain a solution of Φ; it necessarily has the required properties. □

Lemma 7.2. Let v be a node of a regular $\operatorname{Res}(\oplus)$ refutation graph of $\mathrm{BPHP}_{2^\ell}^{2^\ell+1}$. Suppose that v is labeled with a linear clause C and the linear system $\neg C$ is locally consistent. Suppose further that there is a path from the source to v of length t. Then the rank of $\neg C$ is at least $\min\{2^{\ell-2}, \frac{t}{\ell+1}\}$.

Proof. Consider the linear branching program associated with the $\operatorname{Res}(\oplus)$ refutation graph. Let U consist of all sinks u such that there is a path from v 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 U; A consists of clauses of $\mathrm{BPHP}_{2^\ell}^{2^\ell+1}$. It is easy to see that A semantically implies C. Indeed, consider an assignment σ of the variables X that falsifies C. We start a path in the linear branching program from v 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 2.3, σ falsifies D.

Suppose the clauses from A mention exactly $(n+1) - r$ pigeons, where $n = 2^\ell$. Let K be the set of pigeons that are not mentioned in A. We have $|K| = r$.

Claim 7.3. If $\neg C$ is locally consistent, then $\operatorname{rk}(\neg C) \geq \min\{2^{\ell-2}, r\}$.

Proof. Assume that the rank of $\neg C$ is less than $2^{\ell-2}$ and less than r. Then by Lemma 7.1, there exists a full assignment σ that satisfies $\neg C$, and all pigeons from $[n+1] \setminus K$ are in different holes. Hence, σ satisfies all clauses from A. Since C semantically follows from A, then σ satisfies C. But σ also satisfies $\neg C$; this is a contradiction. □

Claim 7.4. Assume that $\dim(\mathrm{Post}(v)) \leq (n+1)\ell - h$[1]. Then $\mathrm{rk}(\neg C) \geq h - r\ell$.

Proof. Recall that $L(C)$ denotes all linear forms mentioned in C. Consider the linear space $V = \langle L(C) \cup \mathrm{Post}(v)\rangle$. On the one hand, $\dim V \leq \mathrm{rk}(\neg C) + \dim(\mathrm{Post}(v)) \leq \mathrm{rk}(\neg C) + (n+1)\ell - h$. On the other hand, for every clause $D \in A$ there is a path from v to D such that $\neg C$ is consistent with the system of all equations labeling the edges of the path. By Lemmas 2.3 and 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 v to D. Every clause of $\mathrm{BPHP}_{2^\ell}^{2^\ell+1}$ contains ℓ variables for each of the two pigeons mentioned in it. Hence, $\dim V \geq (n+1-r)\ell$. From the upper and the lower bound on $\dim V$ we conclude that $\mathrm{rk}(\neg C) \geq h - r\ell$. □

Now assume that $\neg C$ is locally consistent and that $\mathrm{rk}(\neg C) < 2^{\ell-2}$. By Lemma 2.6, $\dim(\mathrm{Post}(v)) \leq (n+1)\ell - t$. Hence, by Claim 7.4, $\mathrm{rk}(\neg C) \geq t - r\ell$. By Claim 7.3, $\mathrm{rk}(\neg C) \geq r$. Then $(\ell+1)\mathrm{rk}(\neg C) \geq t$, hence $\mathrm{rk}(\neg C) \geq \frac{t}{\ell+1}$. □

8 Regular Res(⊕) size lower bound

Theorem 8.1. Any regular Res(⊕) refutation of $\mathrm{BPHP}_{2^\ell}^{2^\ell+1}$ has size at least $2^{\Omega\left(\frac{\sqrt[3]{n}}{\log n}\right)}$, where $n = 2^\ell$.

Proof. Let $t := \lfloor \sqrt[3]{n}/3 \rfloor$. Consider a regular Res(⊕) refutation graph of $\mathrm{BPHP}_{2^\ell}^{2^\ell+1}$ and the linear branching program associated with it. Take a random assignment $\boldsymbol{\sigma}$ of the variables of $\mathrm{BPHP}_{2^\ell}^{2^\ell+1}$ and make t steps in the linear branching program starting from the source and proceeding along the edges labeled with equations satisfied by $\boldsymbol{\sigma}$ (if this path reaches a sink earlier than in t steps, we finish the path there). Lemma 6.2 implies that with probability at least $1 - \frac{6t^3}{n} \geq \frac{1}{2}$ we finish in a node with a linear clause C such that $\neg C$ is locally consistent. Note that C is not at a sink, since the sinks contain clauses of $\mathrm{BPHP}_{2^\ell}^{2^\ell+1}$ and the corresponding linear systems are not locally consistent. By Lemma 7.2, the rank of $\neg C$ is at least $\Omega\left(\frac{\sqrt[3]{n}}{\log n}\right)$. By Lemma 2.3, $\boldsymbol{\sigma}$ satisfies $\neg C$. But for any linear clause D from the refutation, if the rank of $\neg D$ is s, then the probability that a random assignment satisfies $\neg D$ is exactly 2^{-s}. Thus the size of the refutation is at least $2^{\Omega\left(\frac{\sqrt[3]{n}}{\log n}\right)}$. □

Corollary 8.2. Any strongly read-once linear branching program solving $\mathrm{Search}(\mathrm{BPHP}_n^{n+1})$ has size at least $2^{\Omega\left(\frac{\sqrt[3]{n}}{\log n}\right)}$.

Proof. Notice that any strongly read-once branching program has the following property: for any of its non-sink node v, $f_v \notin \mathrm{Pre}(v)$ and for each edge (v,w), $f_v \notin \mathrm{Post}(w)$, where f_v is the query at v. Indeed, $f_v \in \mathrm{Post}(v)$ by the definition of $\mathrm{Post}(v)$, hence $\mathrm{Pre}(v) \cap \mathrm{Post}(v) = \{0\}$ implies that $f_v \notin \mathrm{Pre}(v)$. And if $f_v \in \mathrm{Post}(w)$, then $f_v \in \mathrm{Pre}(w) \cap \mathrm{Post}(w)$, contradicting the strongly read-once property.

Consider a strongly read-once branching program solving $\mathrm{Search}(\mathrm{BPHP}_n^{n+1})$. By Lemma 2.4, we may label all its non-sink nodes with linear clauses to get a refutation graph of BPHP_n^{n+1}. According to the observations above, this refutation is regular, and so by Theorem 8.1 its size is at least $2^{\Omega\left(\frac{\sqrt[3]{n}}{\log n}\right)}$. Since the size of the strongly read-once branching program equals the size of the refutation, the proof of the corollary is complete. □

References

[1] M. Ajtai. $\sum_1^1$-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983.

[2] Miklós Ajtai. The complexity of the pigeonhole principle. *Combinatorica*, 14(4):417–433, 1994.

[1] Recall that $(n+1)\ell$ is the total number of variables in $\mathrm{BPHP}_{2^\ell}^{2^\ell+1}$.

[3] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Pseudorandom generators in propositional proof complexity. *SIAM J. Comput.*, 34(1):67–88, 2004.

[4] Michael Alekhnovich, Edward A. Hirsch, and Dmitry Itsykson. Exponential lower bounds for the running time of DPLL algorithms on satisfiable formulas. *J. Autom. Reason.*, 35(1-3):51–72, 2005.

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

[6] 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.

[7] Yaroslav Alekseev. A Lower Bound for Polynomial Calculus with Extension Rule. In Valentine Kabanets, editor, *36th Computational Complexity Conference (CCC 2021)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[8] Albert Atserias, Ilario Bonacina, Susanna F. de Rezende, Massimo Lauria, Jakob Nordström, and Alexander A. Razborov. Clique is hard on average for regular resolution. *J. ACM*, 68(4):23:1–23:26, 2021.

[9] Paul Beame, Henry A. Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. *J. Artif. Intell. Res. (JAIR)*, 22:319–351, 2004.

[10] Paul Beame, Toniann Pitassi, and Nathan Segerlind. Lower bounds for lov[a-acute]sz–schrijver systems and beyond follow from multiparty communication complexity. *SIAM J. Comput.*, 37(3):845–869, 2007.

[11] Christopher Beck and Russell Impagliazzo. Strong ETH holds for regular resolution. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 487–494. ACM, 2013.

[12] 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.

[13] Eshan Chattopadhyay and Jyun-Jie Liao. Hardness against linear branching programs and more. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference, CCC 2023, July 17-20, 2023, Warwick, UK*, volume 264 of *LIPIcs*, pages 9:1–9:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[14] 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.

[15] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5:394–397, 1962.

[16] M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7:201–215, 1960.

[17] Alexis de Colnet and Stefan Mengel. Characterizing tseitin-formulas with short regular resolution refutations. *J. Artif. Intell. Res.*, 76:265–286, 2023.

[18] Merrick Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *22nd Annual Symposium on Foundations of Computer Science (sfcs 1981)*, pages 260 – 270, 1981.

[19] 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.

[20] Andreas Goerdt. Regular resolution versus unrestricted resolution. *SIAM J. Comput.*, 22(4):661–683, 1993.

[21] Mika Göös and Siddhartha Jain. Communication complexity of collision. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, September 19-21, 2022, University of Illinois, Urbana-Champaign, USA (Virtual Conference)*, volume 245 of *LIPIcs*, pages 19:1–19:9. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[22] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM J. Comput.*, 47(5):1778–1806, 2018.

[23] 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.

[24] 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.

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

[26] Dmitry Itsykson and Alexander Knop. Hard satisfiable formulas for splittings by linear combinations. In Serge Gaspers and Toby Walsh, editors, *Theory and Applications of Satisfiability Testing - SAT 2017 - 20th International Conference, Melbourne, VIC, Australia, August 28 - September 1, 2017, Proceedings*, volume 10491 of *Lecture Notes in Computer Science*, pages 53–61. Springer, 2017.

[27] Dmitry Itsykson, Artur Riazanov, Danil Sagunov, and Petr Smirnov. Near-optimal lower bounds on regular resolution refutations of tseitin formulas for all constant-degree graphs. *Comput. Complex.*, 30(2):13, 2021.

[28] Dmitry Itsykson and Dmitry Sokolov. Lower bounds for myopic DPLL algorithms with a cut heuristic. In Takao Asano, Shin-Ichi Nakano, Yoshio Okamoto, and Osamu Watanabe, editors, *Algorithms and Computation - 22nd International Symposium, ISAAC 2011, Yokohama, Japan, December 5-8, 2011. Proceedings*, volume 7074 of *Lecture Notes in Computer Science*, pages 464–473. Springer, 2011.

[29] 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.

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

[31] Erfan Khaniki. On proof complexity of resolution over polynomial calculus. *ACM Trans. Comput. Log.*, 23(3):16:1–16:24, 2022.

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

[33] Massimo Lauria. A note about k-DNF resolution. *Inf. Process. Lett.*, 137:33–39, 2018.

[34] Xin Li and Yan Zhong. Explicit directional affine extractors and improved hardness for linear branching programs. *Electron. Colloquium Comput. Complex.*, TR23-058, 2023.

[35] Mladen Miksa and Jakob Nordström. A generalized method for proving polynomial calculus degree lower bounds. In David Zuckerman, editor, *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, volume 33 of *LIPIcs*, pages 467–487. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

[36] Fedor Part and Iddo Tzameret. Resolution with counting: Dag-like lower bounds and different moduli. *Comput. Complex.*, 30(1):2, 2021.

[37] Pavel Pudlák and Russell Impagliazzo. A lower bound for DLL algorithms for k-sat (preliminary version). In David B. Shmoys, editor, *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, January 9-11, 2000, San Francisco, CA, USA*, pages 128–136. ACM/SIAM, 2000.

[38] Ran Raz and Iddo Tzameret. Resolution over linear equations and multilinear proofs. *Ann. Pure Appl. Log.*, 155(3):194–224, 2008.

[39] A. A. Razborov. “lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mat. Zametki*, 41:598–607, 1987.

[40] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 77–82, 1987.

[41] Dmitry Sokolov. (semi)algebraic proofs over ±1 variables. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 78–90. ACM, 2020.

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

[43] D.J.A. Welsh. Generalized versions of Hall’s theorem. *Journal of Combinatorial Theory, Series B*, 10(2):95–101, 1971.