Electronic Colloquium on Computational Complexity, Report No. 118 (2025)

Lower bounds for the Bit Pigeonhole Principle in Bounded-Depth Resolution over Parities

Farzan Byramji * Russell Impagliazzo †

Abstract

We prove that for the bit pigeonhole principle with any number of pigeons and n holes, any depth D proof in resolution over parities must have size $\exp(\Omega(n^3/D^2))$. Our proof uses the random walk with restarts approach of Alekseev and Itsykson [STOC '25], along with ideas from recent simulation theorems for randomized parity decision trees.

1 Introduction

The Resolution proof system is perhaps the most well-studied proof system in propositional proof complexity. Lines in Resolution are clauses (disjunctions of literals). The resolution rule allows one to derive from clauses $A \vee x$ and $B \vee \neg x$, the clause $A \vee B$ (where A and B are arbitrary clauses). A resolution refutation of a CNF formula ϕ is a sequence of clauses deriving the empty clause (which is clearly unsatisfiable) from the clauses of ϕ thereby proving that ϕ cannot have a satisfying assignment. The first superpolynomial lower bound for Resolution was proved by Haken [Hak85] for the unary CNF encoding of the pigeonhole principle. Many more lower bounds for Resolution proofs of several other natural formulas have been proved in the following decades.

Going beyond reasoning with clauses, it is natural to consider proof systems where the lines are more powerful circuits like AC^0-circuits or $AC^0[p]$-circuits. A superpolynomial lower bound for the pigeonhole principle in AC^0-Frege was first shown by Ajtai [Ajt94] which was later strengthened to an exponential lower bound [BIKP+92]. On the other hand, proving any superpolynomial lower bound for $AC^0[p]$-Frege is a longstanding challenge. A natural subsystem of $AC^0[2]$-Frege is the proof system Resolution over parities Res($\oplus$), introduced by Itsykson and Sokolov [IS20] a decade ago as a step towards $AC^0[2]$-Frege. Res($\oplus$) extends the power of Resolution to allow linear algebra over $\mathbb{F}_2$. The lines in Res($\oplus$) are linear clauses, which are disjunctions of linear equations over $\mathbb{F}_2$. In Res($\oplus$), a resolution step can derive from given linear clauses A and B a linear clause C such that every $x \in \{0,1\}^n$ satisfying both A and B also satisfies C. In particular, for any linear form v, we can derive $A \vee B$ from $A \vee (v = 0)$ and $B \vee (v = 1)$.

Proving lower bounds for this seemingly simple strengthening of Resolution has turned out to also be quite challenging in general, though there has been progress on restricted subsystems of Res($\oplus$). Several works like [IS20, IR21, GOR24, CMSS23, BK23] have shown lower bounds for tree-like Res($\oplus$). In the last couple of years, there has been progress on restricted DAG-like subsystems of Res($\oplus$) starting with the work of Efremenko, Garlik and Itsykson [EGI24] who showed an exponential lower bound for bottom-regular Res($\oplus$) proofs of the bit pigeonhole principle. Bhattacharya, Chattopadhyay and Dvořák [BCD24] and later Alekseev and Itsykson [AI25] gave such exponential lower bounds for formulas that have polynomial size proofs in ordinary Resolution.

*University of California, San Diego. fbyramji@ucsd.edu. Supported by NSF Award AF: Medium 2212136.
†University of California, San Diego. rimpagliazzo@ucsd.edu. Supported by NSF Award AF: Medium 2212136.

ISSN 1433-8092

Going beyond bottom-regular Res($\oplus$), [AI25] gave an exponential lower bound for Res($\oplus$) proofs whose depth is at most $O(N \log\log N)$ where N is the number of variables of the formula. The depth limit was further pushed to $O(N \log N)$ by Efremenko and Itsykson [EI25]. These lower bounds were proved for the Tseitin formula lifted with 2-stifling gadgets. No other examples were known where such strong lower bounds could be obtained for depth-restricted Res($\oplus$). These works also raised the question of proving lower bounds for depth $\Omega(N^{1+\epsilon})$ for some $\epsilon > 0$.

Our main result is such an exponential lower bound for Res($\oplus$) proofs of the bit pigeonhole principle with n holes when the depth is $O(n^{1.5-\epsilon})$ for any $\epsilon > 0$. Independently, Bhattacharya and Chattopadhyay [BC25] have proved an exponential lower bound for any depth $n^{2-\epsilon}$ Res($\oplus$) proof of the Tseitin formula on an expander lifted with logarithmic size inner product gadget.

Let BPHP_n^m denote the bit pigeonhole principle formula with m pigeons and n holes.

Theorem 1.1. Suppose there exists a Res($\oplus$) proof of BPHP_n^m whose size is S and depth is D. Then $D\sqrt{\log S} = \Omega(n^{1.5})$.

When $m = O(n)$, the number of variables N is $O(n \log n)$. So Theorem 1.1 implies that for any $\epsilon > 0$, any depth $O(N^{1.5-\epsilon})$ proof of $\mathrm{BPHP}_n^{O(n)}$ must have size $\exp(\tilde{\Omega}(N^{2\epsilon}))$.

Another natural setting is when the depth is bounded by the number of variables $m \log n$ without restricting m to be linear in n. Theorem 1.1 implies that any such Res($\oplus$) proof of BPHP_n^m must have size at least $\exp\left(\Omega\left(\frac{n^3}{m^2 \log^2 n}\right)\right)$. In particular, we get an exponential lower bound if $m = n^{1.5-\Omega(1)}$ improving upon [EGI24] who gave an exponential lower bound for bottom-regular proofs when $m = n + 1$.

The pigeonhole principle tautology and its variations have played a central role in proof complexity, especially in connections between propositional proofs and bounded arithmetic. The first superpolynomial lower bounds for resolution [Hak85] were for the pigeonhole principle, and concerted effort has been devoted to extending these lower bounds to stronger proof systems [Ajt94, BIKP+92]. On the other hand, Buss gave a polynomial upper bound for PHP in Frege [Bus87], and Paris, Wilkie and Woods [PWW88] gave a quasipolynomial upper bound for the weak pigeonhole principle in constant depth Frege, which they used to show that the existence of arbitrarily large primes was provable in $I\Delta_0$. (See also the simpler proof of Maciel, Pitassi and Woods [MPW02].) The dual weak pigeonhole principle was used by Jeřábek [Jeř04] to define a bounded arithmetic theory that captures probabilistic polynomial time reasoning. The bit pigeonhole principle has recently been studied in various proof systems including Resolution and DNF-resolution [AMO15, DGGM24], cutting planes and its strengthenings [HP17, IR21, BW25, dRV25], Sherali Adams and Sum of Squares [DGGM24]. Because of the variety of surprising proofs of these tautologies that have been found, which often make fine distinctions between variations, and because of their use in formalizing counting arguments, precisely characterizing the proofs of the pigeonhole principle and its variants is an on-going theme of work in proof complexity.

1.1 Proof overview

Our proof follows the random walk with restarts strategy of Alekseev and Itsykson [AI25] with some simplifying modifications. To illustrate their approach, we now sketch a proof that every size S Resolution proof of the bit pigeonhole principle with n holes must have depth $\Omega(n^{1.5}/\log S)$. Our strategy for the Res($\oplus$) bound will be very similar.

We wish to find a sequence of clauses $C_0 = \bot, C_1, \ldots, C_k$ in the proof such that for each $i \in [k]$, there is a path of length $d := \sqrt{n}$ between C_{i-1} and C_i, and $k = \Omega(n/\log S)$. To ensure that k is large, we will want the width to only increase by $O(\log S)$ when going from clause C_{i-1} to C_i.

We will maintain the property that for each clause C_i, whose width is $O(i \log S)$, there is a partial assignment ρ_i fixing $O(i \log S)$ blocks of the input to distinct holes which falsifies C_i.

Suppose we have found a clause C_i which is at depth $i\sqrt{n}$ from the root. We find the next clause C_{i+1} in the following way. Let $U_i \subseteq \{0,1\}^l$ denote the collection of holes which have been assigned to some pigeon by ρ_i and let $\mathrm{Fix}_i \subseteq [m]$ denote the blocks that are fixed by ρ_i. Suppose $|U_i| \leq n/3$. Consider a random assignment $x \in (\{0,1\}^l)^m$ which is picked uniformly from all assignments consistent with ρ and for which all of the pigeons in $[m] \setminus \mathrm{Fix}_i$ are sent to holes in $\{0,1\}^l \setminus U_i$. Note that the only clauses of BPHP^m_n which can be falsified by such an assignment are those mentioning two pigeons from $[m] \setminus \mathrm{Fix}_i$.

Consider the depth d decision tree T obtained by unraveling the DAG rooted at C_i and only considering nodes up to distance d. Since each query affects at most one block, by the standard birthday problem calculation, with probability at least $1 - d^2/2(n - |U_i|) \geq 1/4$, the decision tree has not found a collision under the above distribution. Since the DAG only has S nodes, there is some clause C for which the corresponding leaves in the decision tree are reached with probability at least $1/(4S)$ without having found a collision. If C mentions r pigeons outside Fix_i, then the probability that a random assignment picked above falsifies C is at most $(n/2(n - |U_i|))^r \leq (3/4)^r$. Together these imply that $r \leq O(\log S)$. So we can take $C_{i+1} := C$ and there is some extension ρ_{i+1} of ρ_i fixing only $O(\log S)$ additional blocks to distinct holes in $\{0,1\}^l \setminus U_i$. This argument can be repeated as long as $|U_{i+1}| \leq n/3$ implying that we can take $k = \Omega(n/\log S)$ as desired.

The overall strategy for the $\mathrm{Res}(\oplus)$ lower bound is similar. Our random walk analysis can be seen as an extension of the ideas of Efremenko, Garlik and Itsykson [EGI24] but phrased in terms of simulating PDTs as in the alternative proof of the random walk analysis for BPHP provided by Byramji and Impagliazzo [BI24]. At a high level, their argument for the uniform distribution is as follows. To give a lower bound against PDTs, their analysis tries to mimic the argument for ordinary decision trees. To do this, they rely on the idea of localizing parities used for PDT lifting theorems [CMSS23, BK23] combined with the observation that if some variable $x_{i,j}$ in a parity P is uniformly distributed and is independent of all other variables $x_{i',j'}$, $i' \neq i$ or $j' \neq j$, occurring in P, then this parity is also a uniform bit which is independent of all variables other than $x_{i,j}$. This gives the ability to condition on the value of this parity while still not having revealed any information about the other variables.

Whenever we localize a parity to a variable $x_{i,j}$, we also condition on the values of all $x_{i,h}, h \in [l] \setminus \{j\}$. In this way, each parity query affects just one block. In the overall analysis, we essentially only rely on the revealed bits ignoring the parity constraints. Specifically in the case of BPHP, for a block where one of the variables is marked ($x_{i,j}$ is fixed in terms of other variables) and the rest are fixed to bits, we think of that pigeon as being sent to two holes obtained by considering both possible values of fixing the marked variable. This does not change the probability of a collision significantly from sending each pigeon to just one random hole.

Our proof here is an extension of the above idea to product distributions where each block is uniform over a large fraction, say 2/3, of inputs $\{0,1\}^l$, by using ideas from recent simulation theorems for randomized parity decision trees [PS25, BI24, BGGMY25]. Since each parity is fairly balanced with respect to this distribution, we can still show that to simulate any depth d PDT on such a distribution we only need to reveal $O(d)$ many blocks except with probability $\exp(-\Omega(d))$.

To reason about $\mathrm{Res}(\oplus)$ proofs, instead of maintaining partial assignments fixing blocks to only bits, we will consider affine restrictions which fix some blocks to linear functions of all unfixed blocks. Still, our affine restrictions will be very similar to the above partial assignments, in that for each fixed block, all but at most one of the variables are fixed to bits. As described above, this lets us think of that pigeon being sent to one of at most two holes. To find such an affine restriction which falsifies a linear clause $\neg\Phi$ reached, we rely on properties of closure and safe

systems, introduced by Efremenko, Garlik and Itsykson [EGI24], along with some other ideas used in their BPHP$_n^{n+1}$ lower bound for regular Res($\oplus$).

To get the bound $D\sqrt{\log S} = \Omega(n^{1.5})$ instead of just $D \log S = \Omega(n^{1.5})$, we rely on the insight of Alekseev and Itsykson [AI25] that this random walk with restarts approach works even when we have small success probability as long as it is at least polynomial in $1/S$. This allows us to choose the length of the walk to be $\sqrt{n \log S}$ instead of just $\sqrt{n}$. (Here we can assume $S \leq 2^{\epsilon n}$ for some suitable constant ϵ since otherwise the statement follows from the general depth lower bound $\Omega(n)$ [EGI24].)

2 Preliminaries

For a search problem $\mathcal{R} \subseteq \{0,1\}^N \times \mathcal{O}$, for any $x \in \{0,1\}^N$, $\mathcal{R}(x)$ denotes $\{o \in \mathcal{O} \mid (x,o) \in \mathcal{R}\}$. For a string $y \in \{0,1\}^l$ and $j \in [l]$, we use $y^{\oplus j}$ to denote the string which is the same as y but with the j^{th} bit flipped ($(y^{\oplus j})_j \neq y_j$).

We will mostly consider inputs x from $(\{0,1\}^l)^m$ viewed as m blocks of l bits. It will be convenient to allow parities to also contain a constant term, that is a parity is a function on $(\{0,1\}^l)^m$ of the form $b + \sum_{i \in [m], j \in [l]} c_{i,j} x_{i,j}$ where $b \in \mathbb{F}_2$ and $c_{i,j} \in \mathbb{F}_2$ for all $i \in [m], j \in [l]$. For $B \subseteq [m]$, we use $\mathcal{L}_B$ to denote the collection of all parities P whose support lies in B, $\text{supp}(P) \subseteq B$.

A linear form is a homogeneous polynomial of degree 1. If v is a linear form and $b \in \mathbb{F}_2$, $v = b$ is a linear equation. A linear system is a collection of linear equations. For any satisfiable linear system, we will assume that it is represented by a collection of linearly independent equations. We will not distinguish between different linear systems defining the same affine subspace.

2.1 Safe collections and closure

We make use of the notions of closure and safe collections of linear forms, introduced by [EGI24].

A set of linear forms V, is said to be safe if there is no subset $W \subseteq \text{span}(V)$ such that the linear forms in V are linearly independent and the support of W (the blocks in $[m]$ whose variables appear in W) has size less than $|W|$.

For $S \subseteq [m]$, $V[\backslash S]$ denotes the linear forms obtained from V by setting all variables in blocks in S to 0. The closure of V, denoted $\text{Cl}(V)$, is the minimal set S such that $V[\backslash S]$ is safe. Abusing notation, for a linear system Φ, we will use $\text{Cl}(\Phi)$ to denote the closure of the linear forms of Φ. Similarly we say that Φ is safe if the associated set of linear forms is safe. The fact that safety and closure do not depend on the choice of basis [EGI24] allows us to freely apply invertible operations to the rows of a linear system or a collection of linear forms.

We now recall some of the properties that we will use.

Lemma 2.1 ([EGI24])**.** Let V be a collection of k independent linear forms and M the corresponding coefficient matrix. V is safe if and only if we can pick k variables, no two from the same block, such that the corresponding columns in M are linearly independent.

Lemma 2.2 ([EGI24])**.** If F is a collection of linear forms, then $|\text{Cl}(F)| + \dim\langle F[\backslash \text{Cl}(F)]\rangle \leq \dim(F)$.

2.2 Affine DAGs and affine restrictions

We will use affine DAGs to reason about Res($\oplus$) proofs in a top-down way as in prior work. These are essentially the same as Res($\oplus$) refutation graphs [EGI24, AI25]. We prefer the term affine DAGs so that we can discuss search problems that are not necessarily false clause search problems.

An affine DAG for a search problem $\mathcal{R} \subseteq \{0,1\}^N \times \mathcal{O}$ is a directed acyclic graph with one source where every internal node has outdegree 2 and it is labeled in the following way:

- At each non-sink node v, we have a linear system Φ_v on $\{0,1\}^N$ and a parity query P_v on $\{0,1\}^N$.
- At each sink w, we have a linear system Φ_w on $\{0,1\}^N$ and an output label $o_w \in \mathcal{O}$.

The linear system at the source node must be the empty system (which is satisfied by all $x \in \{0,1\}^N$). For every internal node v, one of the outgoing edges is labeled $P_v = 0$ and the other is labeled $P_v = 1$. We have the consistency requirement that if the edge (v, w) is labeled by $P_v = b$, then the system Φ_w is implied by the system $\Phi_v \wedge \{P_v = b\}$. For the DAG to correctly solve $\mathcal{R}$, we require that for every sink w, every $x \in \{0,1\}^N$ satisfying Φ_w must also satisfy $(x, o_w) \in \mathcal{R}$.

Semantically, each node in an affine DAG corresponds to an affine subspace of $\mathbb{F}_2^N$ and for an internal node v, if its immediate successors are w_1 and w_2, then the affine subspace corresponding to node v is contained in the union of the affine subspaces at nodes w_1 and w_2. For considering affine restrictions below, it will be convenient to allow inconsistent systems in an affine DAG. Similarly, an empty set will be considered an affine subspace.

Every Res($\oplus$) refutation of a CNF formula ϕ gives an affine DAG solving the search problem $\mathcal{R}_\phi$ associated with ϕ [EGI24] whose size and depth are no larger than the size and depth, respectively, of the Res($\oplus$) refutation. With this in mind, we will only discuss affine DAGs from now on.

The following lemma is essentially Lemma 2.3 in [EGI24].

Lemma 2.3 ([EGI24])**.** Consider nodes u and v in an affine DAG with systems Φ_u and Φ_v respectively such that there is some path p from u to v. Let Ψ be the system consisting of all equations labeling the edges of p. Then $\Phi_u \wedge \Psi$ implies Φ_v.

We now consider inputs in $(\{0,1\}^l)^m$. We will make use of certain block-respecting affine restrictions in our proof. Since we do not use any other kind of affine restrictions, we simply call them affine restrictions. Recall that $\mathcal{L}_B$ denotes the collection of all linear functions supported on B. Let $A \subseteq [m]$. We say that $\rho : \{x_{i,j} \mid i \in A, j \in [l]\} \to \mathcal{L}_{[m]\setminus A}$ is an affine restriction fixing A.

For a parity P on $(\{0,1\}^l)^m$ and an affine restriction ρ fixing A, we use $P|_\rho$ to denote the parity obtained by substituting for all $x_{i,j}, i \in A, j \in [l]$ according to ρ. This results in a parity in $\mathcal{L}_{[m]\setminus A}$. Similarly, we use $\Phi|_\rho$ to denote the linear system obtained by substituting in Φ all $x_{i,j}, i \in A, j \in [l]$ according to ρ. This substitution could make the system inconsistent in which case we simply represent the resulting system by $1 = 0$. Otherwise, we tacitly assume that all equations in $\Phi|_\rho$ are linearly independent by removing any equations which are implied by others. Again the exact choice of which redundant equations to remove is not important for what follows.

Let us give an equivalent way of describing $\Phi|_\rho$ when it is satisfiable. We use Ψ_ρ to denote the collection of equations defining the affine restriction ρ. (This is essentially just a change in how we view ρ.) The collection of all equations implied by $\Phi|_\rho$ is the same as the collection of equations implied by $\Phi \wedge \Psi_\rho$ whose support does not contain any variable from A.

If $V \subseteq (\mathbb{F}_2^l)^m$ is the affine subspace defined by Φ and W is the set of inputs consistent with ρ (or equivalently, satisfy Ψ_ρ), the affine subspace defined by $\Phi|_\rho$ is obtained by considering the projection of $V \cap W$ onto the blocks not fixed by ρ.

For a given affine DAG $\mathcal{D}$ and affine restriction ρ fixing A, we use $\mathcal{D}|_\rho$ to denote the affine DAG obtained by applying ρ to each parity and linear system appearing in $\mathcal{D}$. Observe that the consistency condition still holds for the DAG obtained after applying the restriction. This is perhaps easiest to see from the semantic view above of how a restriction affects an affine subspace. (Note that an inconsistent system implies every linear system.) $\mathcal{D}|_\rho$ does not mention any variable from A.

If $\mathcal{D}$ solves a search problem $\mathcal{R} \subseteq (\{0,1\}^l)^m \times \mathcal{O}$, then $\mathcal{D}|_\rho$ solves $\mathcal{R}|_\rho$ where $\mathcal{R}|_\rho \subseteq (\{0,1\}^l)^{[m]\setminus A} \times \mathcal{O}$ is defined by $\mathcal{R}|_\rho(y) = \mathcal{R}(x)$ where x is the unique extension of y according to ρ. In other words, x is the unique solution of Ψ_ρ which agrees with y on the blocks outside A.

2.3 Bit pigeonhole principle and collision-finding

The bit pigeonhole principle BPHP_n^m on m pigeons and $n = 2^l$ holes ($m > n$) encodes the unsatisfiable statement that m pigeons can be placed into n holes such that no two pigeons are in the same hole. For each pigeon $i \in [m]$, we have variables $x_{i,j}, j \in [l]$ encoding the hole it flies to. In the associated false clause search problem, given such an assignment $x \in (\{0,1\}^l)^n$, the goal is to find distinct $i, k \in [m]$ and $z \in \{0,1\}^l$ such that $x_i = x_k = z$.

We will consider the closely related search problem of collision-finding $\mathrm{Coll}_n^m \subseteq (\{0,1\}^l)^m \times \binom{[m]}{2}$ where we only need to find distinct i, k such that $x_i = x_k$. It is easy to see that any affine DAG solving the false clause search problem associated with BPHP_n^m also solves the collision-finding problem once we change the output labels suitably.

For our proof, we will need to consider a promise version of collision finding where each pigeon is promised to only fly into a collection of available/allowed holes $A \subseteq \{0,1\}^l$. We use $\mathrm{Coll}_A^m \subseteq (\{0,1\}^l)^m \times \binom{[m]}{2}$ to denote this problem. Formally, $\mathrm{Coll}_A^m = \{(x, \{i,k\}) \mid x_i = x_k, i \neq k\} \cup \{(x, \{i_1, i_2\}) \mid x_k \notin A \text{ for some } k \in [m], i_1 \neq i_2\}$. The second set is to simply allow all outputs for any input violating the promise. If $m > |A|$, the problem Coll_A^m is total.

3 PDT lower bound for collision-finding

Let $n := 2^l = |\{0,1\}^l|$. We use $U \subseteq \{0,1\}^l$ to denote a collection of used/forbidden holes. Let $A = \{0,1\}^l \setminus U$ be the set of available holes. Let μ be the uniform distribution on A^m. Let T be a deterministic parity decision tree on $(\{0,1\}^l)^m$ of depth d.

We describe a procedure, Algorithm 1, which will let us give a lower bound on the probability that T has not solved Coll_A^m when run on the distribution μ. Since the procedure is similar to recent simulations for randomized parity decision trees [PS25, BI24, BGGMY25], we call it a simulation.

Lemma 3.1. The simulation maintains the following invariants in the beginning of each iteration of the while loop:

1. The collection of equations L uniquely determines $x_i, i \in [m]\setminus F$ as linear functions of $x_i, i \in F$. Moreover, for each assignment to all blocks in F, if we assign values to x_i ($i \in [m] \setminus F$) according to L, we have $x_i \in A$ for each $i \in [m] \setminus F$.

2. L implies all the parity constraints on the path from the root to the current node.

Proof. The proof is by induction on the number of iterations, *iter*. At the beginning of the first iteration, $iter = 0$ and all the statements are seen to be trivially true.

So suppose the statement holds at the beginning of the iteration when $iter = i$, $i \geq 0$. We want to show that all the conditions also hold at the end of the iteration (at which point we have $iter = i + 1$).

In the case that the condition in Line 10 is satisfied, we do not change L and the parity constraint labeling the edge traversed in this iteration is implied by the condition in Line 10. So all the invariants are satisfied at the end of the iteration.

Next let us consider the case where the else clause on Line 12 is executed. The case $y^{\oplus j} \notin A$ is clear since in this case we explicitly set $x_{i,j} = y_j$ ensuring that $x_i = y \in A$. We stay at the same node in this case.

Algorithm 1: PDT simulator

Input: $U \subseteq (\{0,1\}^l)^m$, PDT T

$A = \{0,1\}^l \setminus U$; `// Available holes`

$C \leftarrow []$; `// List of holes used during simulation`

$F \leftarrow [m]$; `// Free blocks`

$L \leftarrow \emptyset$; `// Collection of equations`

$v \leftarrow$ root of T;

$iter \leftarrow 0$;

while v is not a leaf **do**

  $P' \leftarrow$ query at v ;

  $P \leftarrow P'$ after substituting according to L;

  `//` P `now depends only on blocks in` F `and is equivalent to` P' `under` L

  if P is a constant, $b \in \mathbb{F}_2$ **then**

    Update v according to b;

  else

    $(i,j) \leftarrow \min\{(i,j) \in F \times [l] \mid x_{i,j} \text{ appears in } P\}$;

    Pick y uniformly at random from A;

    $L \leftarrow L \cup \{x_{i,h} = y_h \mid h \in [l] \setminus \{j\}\}$;

    if $y^{\oplus j} \notin A$ **then**

      $L \leftarrow L \cup \{x_{i,j} = y_j\}$;

      Append $\{y\}$ to C;

    else

      Pick $b \in \mathbb{F}_2$ uniformly at random;

      $L \leftarrow L \cup \{P = b\}$;

      Append $\{y, y^{\oplus j}\}$ to C;

      Update v according to b;

    $F \leftarrow F \setminus \{i\}$;

  $iter \leftarrow iter + 1$;

if there exist $i \neq k$ such that $C_i \cap C_k \neq \emptyset$ **then**

  return FAIL ; `// There is a potential collision`

else

  return v, C, L, F

In the other case $y^{\oplus j} \in A$, we know that no matter what bit we set $x_{i,j}$ to we will have $x_i \in A$. In this case, we fix $x_{i,j}$ as determined by the constraint $P = b$. Since we ensured that under L, the constraint $P = b$ is equivalent to the original parity constraint at the edge just crossed, we have preserved the invariant that each parity constraint on the path from the root to the current node is implied by L.

In all these cases, to see the first point, note that L has an upper triangular structure if we rearrange the columns so that all the fixed blocks appear before the free blocks. Moreover, as explained above, the fixed bits in these blocks ensure that for any assignment to the free blocks, each fixed block belongs to A. □

Lemma 3.2. Let $W(v)$ be the event that node v of T is visited by Algorithm 1. Let $V(v)$ be the event that for a random $x \sim \mu$, running T on x reaches v. Then for every $v \in T$, we have $\Pr[W(v)] = \Pr[V(v)]$.

Proof. Suppose at the beginning of an iteration, we have a linear system L which fixes some subset $S := [m] \setminus F$ of blocks as linear functions of the blocks F such that conditioned on $x \sim \mu$ satisfying L, the distribution on x restricted to the free blocks F is uniform on A^F. We will show that the simulation adds equations to L with the correct probability and the resulting system L also satisfies the above condition. The claim then follows by induction and Lemma 3.1.

In the case that the parity we are trying to simulate is already determined by L, it is clear that the simulation goes to the correct child with probability 1.

Consider the case where the parity query is not determined by L. Let P denote the equivalent parity query under L which only depends on the blocks in F. Fix $i \in F, j \in [l]$ occurring in P. We condition on all $x_{i,h}, h \neq j$ (Lines 15-16). If these uniquely determine $x_{i,j}$ also according to A, then we fix $x_{i,j}$ to the unique possible value. This happens in the if clause at Line 16. Since block i is independent of the other blocks, the blocks in $F \setminus \{i\}$ continue to be distributed according to the uniform distribution on $A^{F\setminus\{i\}}$.

In the case that $x_{i,j}$ is not uniquely determined by $x_{i,h}, h \notin [l]$, $x_{i,j}$ is a uniform random bit. Since the parity P (which only depends on blocks in F) contains $x_{i,j}$, this parity is independent of $x_{i',h'}$ for $i' \in F \setminus \{i\}$ and is uniformly distributed. The independence implies that conditioned on $P = b$ for any b, the distribution on all blocks in $F \setminus \{i\}$ remains the uniform distribution on $A^{F\setminus\{i\}}$. □

Lemma 3.3. Let d denote the depth of the PDT T. If $|A| \geq 2n/3$ and $35 \leq d \leq |A|/300$, then the probability that Algorithm 1 does not return FAIL is at least $\exp(-64d^2/|A|)/2$.

Proof. We need to lower bound the probability that there is no collision in C. We will give a lower bound on the probability that there is no collision in C and $iter \leq 4d$. It suffices to give an upper bound on $\Pr[iter > 4d]$ and a lower bound on the probability that there is no collision in C in the first $4d$ iterations.

To bound the probability that $iter > 4d$, we will use that in each of the first $4d$ iterations (if we have not already reached a leaf), after conditioning on the history, the probability that Line 10 or 19 is executed is at least $1/2$. Remember that whenever either of these lines is executed, we move down the PDT. Since the depth of the PDT is d, the only way we could not have reached a leaf after $4d$ iterations is if these lines were executed fewer than d times during the first $4d$ iterations.

Define for each $i \in [4d]$, the following random variables X_i and Y_i where X_i encodes the changes to the state of the simulation (L, F, v) that occur during the i^{th} iteration and Y_i is a binary random variable which is 1 if the simulation ended before the i^{th} iteration, or Line 10 or 19 is executed and satisfied in the i^{th} iteration. Note that Y_i is determined by $X_1, \ldots, X_i$.

Conditioned on $X_1, X_2, \ldots, X_{i-1}$ ($i \leq 4d$), we know (deterministically) that one of the following cases happens.

1. There is no i^{th} iteration
2. The if clause following Line 10 is executed in the i^{th} iteration
3. The else clause Line 12 is executed in the i^{th} iteration

In the first two cases, we have $Y_i = 1$ by definition.

So let us verify in the third case that $\Pr[Y_i = 1 \mid X_1, X_2, \ldots, X_{i-1}] \geq 1/2$. Note that together $X_1, X_2, \ldots, X_{i-1}$ determine F, L in the beginning of the i^{th} iteration. The probability that $y^{\oplus j} \notin A$ is at most $(n-|A|)/|A|$ since each such y with $y^{\oplus j} \notin A$ corresponds to a unique used hole in $[n] \setminus A$. Since we assumed $|A| \geq 2n/3$, we get that this probability is at most $1/2$. Thus we have $y^{\oplus j} \in A$ with probability at least $1/2$ in which case Line 19 is executed.

This implies $\mathbb{E}[\sum_{i=1}^{4d} Y_i] \geq 2d$. We now use a variant of the Chernoff bound which applies to such dependent Y_i's (see, for instance, [MU17, Lemma 17.3]) to conclude

$$\Pr[\sum_{i=1}^{4d} Y_i < d] \leq \exp(-d/4).$$

We next give a lower bound on the probability that no collision is created in C in the first $4d$ iterations. To do this, we will give a lower bound on the conditional probability that in a given iteration i, conditioned on the events in previous iterations, the sampled y is such that the new set $\{y\}$ or $\{y, y^{\oplus j}\}$ appended to C does not cause a collision with one of the previous sets in C.

If in the i^{th} iteration, we are in the trivial case where P is already determined by L, then the probability of there being no new collision is 1.

So consider the case where some y is sampled uniformly from A. In each iteration, we append at most one set to C and each such set contains at most 2 elements. This means that the union of all sets in C has size at most $2(i-1)$ at the beginning of the i^{th} iteration. So the total number of bad $y \in A$ sampling which can cause a collision with a previous string in C is at most $2 \cdot 2(i-1) = 4(i-1)$ where the additional factor 2 comes from also considering collisions caused by $y^{\oplus j}$. Thus the probability that the sampled y is such that the appended $\{y\}$ or $\{y, y^{\oplus j}\}$ causes a collision is at most $4(i-1)/|A|$.

Since we have shown that with probability at least $1 - 4(i-1)/|A|$, no new collision is created in C in the i^{th} iteration after conditioning on all the events in previous iterations, this is also a lower bound on the probability that no new collision is created if we condition on any subset of allowed events in the previous iterations. In particular, conditioned on there being no collision in C at the beginning of the i^{th} iteration, the probability that there is no collision in C at the end of the i^{th} iteration is at least $1 - 4(i-1)/|A|$.

Combining these, we get that the overall probability that there is no collision in C in the first $4d$ iterations is at least

$$\prod_{i=0}^{4d-1} \left(1 - \frac{4i}{|A|}\right) \geq \prod_{i=0}^{4d-1} \exp\left(-\frac{8i}{|A|}\right) \geq \exp(-64d^2/|A|)$$

For the first inequality, we used $1 - x \geq \exp(-2x)$ which holds for $x \leq 0.75$.

By the union bound,

$$\begin{aligned}&\Pr[\text{no collision in } C \text{ and } iter \le 4d]\\ &\ge \Pr[\text{no collision in } C \text{ in first } 4d \text{ iterations}] - \Pr[iter \ge 4d]\\ &\ge \exp(-64d^2/|A|) - \exp(-d/4) \ge \exp(-64d^2/|A|)/2\end{aligned}$$

where the last inequality used the assumed bounds on d. □

Lemma 3.4. Let v, C, L, F be returned by a successful run of Algorithm 1. Suppose $|A| \ge 3$. For every pair $i, k \in [m]$, $i \ne k$, there exists $x \in A^m$ such that $x_i \ne x_k$ and x reaches the leaf v.

Proof. By Lemma 3.1, L implies all the parity constraints on the path from the root to v. So it is sufficient to find $x \in A^m$ satisfying L for which $x_i \ne x_j$. We consider cases depending on whether i, k belong to F.

1. If $\{i, k\} \subseteq F$, set x_i and x_k to distinct values from A, set all other blocks in F to arbitrary strings in A and fix blocks outside F according to L. By Lemma 3.1, such a string x is unique and lies in A^m.

2. If exactly one of $\{i, k\}$ lies in F, say k, then we consider the possible strings that x_i can be according to L. Since $i \notin F$, L fixes at least $l - 1$ bits of x_i. So there are at most two such strings and there is some other string in A since $|A| \ge 3$. Set x_k to such a string. Set all other blocks in F to arbitrary strings in A and fix blocks outside F according to L.

3. If both i, k lie outside F, we set blocks in F to arbitrary strings in A and fix blocks outside F according to L. Since this was a successful run, we must have $x_i \ne x_k$ as all the sets in C are disjoint and x_i, x_k belong to sets at different indices in C. □

4 Size-depth lower bound for affine DAGs solving collision-finding

The following simple lemma will let us upper bound the rank of a linear system in terms of the probability that it is satisfied under the distribution μ.

Lemma 4.1. Let Ψ be a linear system on $(\{0,1\}^l)^m$ whose rank is r. Let $A \subseteq \{0,1\}^l$ be such that $|A| \ge 2n/3$. Let μ be the uniform distribution on A^m. Then

$$\Pr_{x \sim \mu}[x \text{ satisfies } \Psi] \le \left(\frac{3}{4}\right)^r.$$

Proof. Fix a collection of r linearly independent columns of the coefficient matrix of Ψ. Let B the blocks containing the corresponding variables. We have $b := |B| \le r$. Condition on all blocks outside B. For any possible assignment to all blocks outside B, the number of solutions for the resulting system Ψ' whose rank is r and depends on b blocks (bl variables) is $\frac{n^b}{2^r}$. In particular, the number of such solutions from A^b is at most $\frac{n^b}{2^r}$. By assumption, $|A|^b \ge (2n/3)^b$. So the conditional probability is at most $\frac{n^b}{2^r}/(2n/3)^b = 1.5^b/2^r \le (3/4)^r$. Since this holds for all possible assignments to the blocks outside B, we have the bound $\Pr_{x\sim\mu}[x \text{ satisfies } \Psi] \le \left(\frac{3}{4}\right)^r$ as desired. □

The next lemma will be used between random walks to obtain an affine restriction which fixes few blocks to distinct strings and implies the linear system at the node reached at the end of the random walk.

Lemma 4.2. Let v, C, L, F be returned by a successful run of Algorithm 1 when run on T and $U \subseteq \{0,1\}^l$. Let Ψ be the linear system labeling the node in the affine DAG corresponding to v. Let r be the rank of Ψ. If $|U| + 2r \leq n/2$, then there exists an affine restriction ρ fixing blocks $[m] \setminus F'$ and satisfying the following conditions:

1. The number of blocks fixed by ρ, say s, is at most r.
2. Ψ is implied by Ψ_ρ (the linear system equivalent to the affine restriction ρ).
3. There exists a set $U' \subseteq \{0,1\}^l \setminus U$ such that $|U'| \leq 2s$, $|U'| \geq s$ and for any assignment to F', if we set the blocks $[m] \setminus F'$ according to ρ, all the strings assigned to $[m] \setminus F'$ lie in U' and are distinct.

Proof. Consider the blocks in $\mathrm{Cl}(\Psi) \cap F$. Let $\hat{U}$ be the set of all possible holes used by the blocks in $\mathrm{Cl}(\Psi) \setminus F$ during this run of the simulation. Since we assign at most two holes to each fixed block during the simulation, $|\hat{U}| \leq 2|\mathrm{Cl}(\Psi) \setminus F|$. Assign distinct strings from $\{0,1\}^l \setminus (U \cup \hat{U})$ to the blocks in $\mathrm{Cl}(\Psi) \cap F$. This can be done if $|\mathrm{Cl}(\Psi) \cap F| \leq n - (|U| + 2|\mathrm{Cl}(\Psi) \setminus F|)$. This holds since $|U| + 2|\mathrm{Cl}(\Psi) \setminus F| + |\mathrm{Cl}(\Psi) \cap F| \leq |U| + 2(|\mathrm{Cl}(\Psi) \setminus F| + |\mathrm{Cl}(\Psi) \cap F|) \leq |U| + 2r \leq n/2$ where the second inequality used Lemma 2.2.

Now assign all blocks in $F \setminus \mathrm{Cl}(\Psi)$ arbitrarily and extend to a full assignment x by fixing all blocks in $[m] \setminus F$ according to L. By Lemma 3.1, there is a unique such extension which also ensures that all fixed blocks lie in $\{0,1\}^l \setminus U$. By Lemma 3.1, we have that L implies all parity constraints on some path p from the root to the node labeled by Ψ. By Lemma 2.3, the system of parity constraints on path p implies Ψ. Combining these, we get that L implies Ψ and in particular, the full assignment x defined above satisfies Ψ.

This means that the partial assignment to blocks in $\mathrm{Cl}(\Psi)$ which agrees with x satisfies all equations in $\mathrm{span}(\Psi)$ whose support is contained in $\mathrm{Cl}(\Psi)$. Let σ_1 denote this (bit-fixing) restriction which sets blocks in $\mathrm{Cl}(\Psi)$ according to x. Since we are considering a successful run of the algorithm, the blocks in $\mathrm{Cl}(\Psi) \setminus F$ are assigned distinct strings from $\hat{U}$ which does not contain any strings from U. Moreover, we ensured above that each block in $\mathrm{Cl}(\Psi) \cap F$ is not assigned a string from $U \cup \hat{U}$. So σ_1 fixes blocks in $\mathrm{Cl}(\Psi)$ to distinct strings in $\{0,1\}^l \setminus U$.

Now consider $\Psi' := \Psi|_{\sigma_1}$. Since σ_1 fixes blocks in $\mathrm{Cl}(\Psi)$ to bits, Ψ' is safe. By Lemma 2.1, there exist $\dim(\Psi')$ variables lying in distinct blocks such that the corresponding columns in Ψ' are linearly independent. Let X denote the set of these variables and B the set of blocks containing X. We will now define a restriction σ_2 which fixes all variables in blocks B except those in X to bits. We wish to ensure that irrespective of how the bits in X are set, σ_2 guarantees that together with σ_1 all blocks in $\mathrm{Cl}(\Psi)$ and B are assigned distinct strings from $\{0,1\}^l \setminus U$.

We do this inductively, considering all blocks in B in any order and for each block, picking an assignment to its $l-1$ variables that do not belong to X in such a way that the two possible strings do not agree with any of the previously assigned strings. This can be done as long as $n/2 > |U| + |\mathrm{Cl}(\Psi)| + 2(\dim(\Psi') - 1)$. To see this, note that the left side denotes the number of assignments to $l-1$ bits. The right hand side denotes the maximum number of forbidden assignments: there are $|U|$ forbidden holes from before, $|\mathrm{Cl}(\Psi)|$ holes assigned to blocks in the closure by σ_1 and 2 forbidden assignments for each block in B which we have already fixed. This condition is satisfied since we have $|U| + |\mathrm{Cl}(\Psi)| + 2(\dim(\Psi') - 1) < |U| + 2(|\mathrm{Cl}(\Psi)| + \dim(\Psi')) \leq |U| + 2r \leq n/2$ where the second to last inequality used Lemma 2.2. So such a restriction σ_2 exists.

Finally consider the system $\Psi'|_{\sigma_2}$. Since the restriction σ_2 only fixed variables in blocks B outside X to bits, X is still a collection of $\dim(\Psi')$ variables whose corresponding columns are linearly independent. So we can solve for each variable in X to express it as a linear function

depending only on variables in blocks outside $\mathrm{Cl}(\Psi) \cup B$. This gives us the desired affine restriction once we combine with the restrictions σ_1 and σ_2.

Let us verify that this affine restriction satisfies the desired properties. The number of blocks fixed is $s = |\mathrm{Cl}(\Psi)| + \dim(\Psi[\setminus \mathrm{Cl}(\Psi)]) \leq r$ by Lemma 2.2.

To see that Ψ_ρ implies Ψ, first note that all equations supported on the closure of Ψ are satisfied by the restriction σ_1 above and ρ is an extension of σ_1. Next we need to check that $\Psi|_{\sigma_1}$ is implied by ρ. This was ensured since $\Psi|_{\sigma_1}$ is implied by σ_2 and $\Psi'|_{\sigma_2}$ above which are contained in ρ.

For the third point, we obtain U' by considering all possible strings that the blocks in $\mathrm{Cl}(\Psi) \cup B$ are assigned by $\sigma_1 \cup \sigma_2$. Note that the blocks in $\mathrm{Cl}(\Psi)$ are fixed completely to distinct strings outside U. Each block in B is assigned two strings since only one variable is left undetermined by σ_2. So the set of all strings assigned to these blocks has size $|\mathrm{Cl}(\Psi)| + 2\dim(\Psi') \leq 2|\mathrm{Cl}(\Psi)| + 2\dim(\Psi') = 2s$. Similarly, $|U'| = |\mathrm{Cl}(\Psi)| + 2\dim(\Psi') \geq |\mathrm{Cl}(\Psi)| + \dim(\Psi') = s$. We also ensured above that no matter how the bits in X are assigned, all the blocks in $\mathrm{Cl}(\Psi) \cup B$ receive distinct strings. □

Next we prove our main lemma which performs a random walk to find a node far from the root where the DAG has not made much progress.

Lemma 4.3. Let $A \subseteq \{0,1\}^l$ with $|A| \geq 2n/3$ and $m > |A|$. Suppose there is an affine DAG C solving Coll_A^m whose depth is at most D and size is at most S. Suppose $\ln S \leq n/10^6$. Set $d := \lfloor \sqrt{n \ln S} \rfloor$. There exist $m' \geq m - O(\log S)$ and $A' \subseteq A$ with $|A'| \geq |A| - O(\log S)$ such that $m' > |A'|$ and there exists an affine DAG C' solving $\mathrm{Coll}_{A'}^{m'}$ whose depth is at most $D - d$ and size is at most S.

Proof. Let T be the depth d PDT obtained by starting at the root of the DAG C, repeating nodes at depth at most d if required and removing any nodes beyond depth d. Run Algorithm 1 on T and $U := \{0,1\}^l \setminus A$. By Lemma 3.3, it succeeds with probability p at least $S^{-O(1)}$. The assumptions on d for Lemma 3.3 are satisfied as explained next. The lower bound on d follows from n being large. Combining the assumptions $\log S \leq n/10^6$ and $|A| \geq 2n/3$ implies the required upper bound on d. Since the affine DAG has size at most S, there must be a node w in the DAG such that the leaves in T corresponding to w are successfully reached with probability at least $p/S \geq S^{-O(1)}$. By Lemma 3.2, this is also a lower bound on the probability that when x is picked uniformly at random from A^m and we follow the path taken by x in the DAG C starting at the root, the path reaches the node w.

Let Φ denote the linear system at w in the DAG C. By Lemma 4.1 and the $S^{-O(1)}$ lower bound on the probability of Φ being satisfied by a random $x \in A^m$, we get that the rank r of Φ is at most $O(\log S)$. Now fix a successful run of the simulation which ends at a leaf v in T corresponding to w. Apply Lemma 4.2 to this successful run and the linear system Φ to obtain ρ, s and $U' \subseteq A$. Here ρ is an affine restriction fixing s blocks as linear functions of the other $m - s$ blocks.

Set $m' = m - s$ and $A' = A \setminus U'$. Since $|U'| \geq s$, we have $m' = m - s > |A| - s \geq |A| - |U'| = |A'|$. Consider $C|_\rho$ which solves $\mathrm{Coll}_A^m|_\rho$. Note that since Ψ_ρ implies Φ (Lemma 4.2), $\Phi|_\rho$ is the empty system. So the node w in $C|_\rho$ is now labeled by the empty system and we can consider the affine DAG C' consisting only of nodes reachable from w. C' still solves $\mathrm{Coll}_A^m|_\rho$.

We claim that C' solves $\mathrm{Coll}_{A'}^{m'}$ after some minor modifications. We first modify C' so that it solves $\mathrm{Coll}_{A'}^{m'}$ when the input blocks are indexed using F' instead of $[m']$. Fix any distinct i and k in F'. For any sink node in C' whose output label is not contained in F', we replace it by $\{i, k\}$.

We now verify that for every sink in C', the output label is correct for every input satisfying the linear system at the sink. Note that we only need to check this for inputs in $(A')^{F'}$ since for other inputs, all outputs are considered correct. We consider cases on the original output label i', k' (before changing it above):

1. If $\{i', k'\} \subseteq F'$, then the label stays unchanged. Consider any $x \in (A')^{F'}$ satisfying the linear system labeling this node. Then since $A' \subseteq A$, x's extension y according to ρ satisfies $y \in A^m$ and we must have $y_{i'} = y_{k'}$ since C' solves $\mathrm{Coll}_A^m|_\rho$. This implies $x_{i'} = x_{k'}$ since $y_{i'} = x_{i'}$ and $y_{k'} = x_{k'}$. So the output label is correct for all x at this node.

2. If $|\{i', k'\} \cap F'| = 1$, the label is changed to $\{i, k\}$. Suppose $i' \in F'$ (the case $k' \in F'$ is analogous). Consider any $x \in (A')^{F'}$ satisfying the linear system at this node. Its extension y according to ρ lies in A^m. So we have $y_{i'} = y_{k'}$. However, $y_{k'} \in U'$ by Lemma 4.2 and $y_{i'} = x_{i'} \in A'$ which is a contradiction since $A' = A \setminus U'$. So there is no $x \in (A')^{F'}$ which satisfies the linear system at this node.

3. If $\{i', k'\} \cap F' = \emptyset$, the label is changed to $\{i, k\}$. By Lemma 4.2, for every $x \in (\{0,1\}^l)^{F'}$, the extension y according to ρ satisfies $y_{i'} \neq y_{k'}$ since i', k' lie outside F'. This means that the linear system at this node must be inconsistent after applying the restriction ρ.

So C' correctly solves $\mathrm{Coll}_{A'}^{m'}$ on the index set F'. Finally we relabel according to any bijection between F' and $[m']$ to obtain a DAG solving $\mathrm{Coll}_{A'}^{m'}$.

It is clear that the DAG C' has size at most S. If the DAG C' has depth more than $D - d$, then the original DAG C must have depth more than D. To see this, take a successful run of the algorithm ending at a leaf v in T where v corresponds to w in C. Observe that v must be at depth d since otherwise w would be a sink in C which is not possible by Lemma 3.4. So there is a path p in C from the source to w of length d corresponding to the root to v path in T. If there is a path in C' of length more than $D - d$, there is a corresponding path in C which we can combine with p to obtain a path of length more than D, which would be a contradiction. □

We now make repeated use of the above lemma to prove our result.

Theorem 4.4. Suppose there exists an affine DAG C on $(\{0,1\}^l)^n$ of depth D and size S which solves Coll_n^m, $m > n$. Then D and S satisfy $D\sqrt{\log S} \geq \Omega(n^{1.5})$.

Proof. We may assume that $S \leq e^{n/10^6}$. If this is not the case, then the $\Omega(n)$ lower bound on the depth of any affine DAG for Coll_n^m [EGI24] implies the desired bound.

Set $A = \{0,1\}^l$. We repeatedly invoke Lemma 4.3 with A, m and C, updating A, m and C according to A', m', C' guaranteed by the statement. We can do this as long as $|A| \geq 2n/3$. The other conditions required by the lemma continue to hold by the conclusion of the lemma statement. In each iteration, $|A|$ decreases by at most $O(\log S)$. So we can use Lemma 4.2 at least $(n/3)/O(\log S) = \Omega(n/\log S)$ many times. This finally gives a DAG of depth at most $D - \Omega(nd/\log S)$ where $d = \lfloor \sqrt{n \ln S} \rfloor$. Since depth must be nonnegative, we have $D \geq \Omega(nd/\log S) = \Omega(n^{1.5}/\sqrt{\log S})$ as desired. □

References

[AI25] Yaroslav Alekseev and Dmitry Itsykson. "Lifting to Bounded-Depth and Regular Resolutions over Parities via Games". In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*. STOC '25. Prague, Czechia: Association for Computing Machinery, 2025, pp. 584–595. ISBN: 9798400715105. DOI: 10.1145/3717823.3718150. URL: https://doi.org/10.1145/3717823.3718150.

[Ajt94] Miklós Ajtai. "The complexity of the pigeonhole principle". In: *Combinatorica* 14.4 (1994), pp. 417–433.

[AMO15] Albert Atserias, Moritz Müller, and Sergi Oliva. "Lower bounds for DNF-refutations of a relativized weak pigeonhole principle". In: *The Journal of Symbolic Logic* 80.2 (2015), pp. 450–476.

[BC25] Sreejata Kishor Bhattacharya and Arkadev Chattopadhyay. *Exponential Lower Bounds on the Size of ResLin Proofs of Nearly Quadratic Depth*. Tech. rep. TR25-106. Electronic Colloquium on Computational Complexity (ECCC), 2025. URL: https://eccc.weizmann.ac.il/report/2025/106/.

[BCD24] Sreejata Kishor Bhattacharya, Arkadev Chattopadhyay, and Pavel Dvořák. "Exponential Separation Between Powers of Regular and General Resolution over Parities". In: *39th Computational Complexity Conference (CCC 2024)*. Ed. by Rahul Santhanam. Vol. 300. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 23:1–23:32. ISBN: 978-3-95977-331-7. DOI: 10.4230/LIPIcs.CCC.2024.23.

[BGGMY25] Tyler Besselman, Mika Göös, Siyao Guo, Gilbert Maystre, and Weiqiang Yuan. "Direct Sums for Parity Decision Trees". In: *40th Computational Complexity Conference (CCC 2025)*. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025.

[BI24] Farzan Byramji and Russell Impagliazzo. *Lifting to randomized parity decision trees*. Tech. rep. TR24-202. To appear in RANDOM 2025. Electronic Colloquium on Computational Complexity (ECCC), 2024. URL: https://eccc.weizmann.ac.il/report/2024/202/.

[BIKP+92] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, Pavel Pudlák, and Alan Woods. "Exponential lower bounds for the pigeonhole principle". In: *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*. 1992, pp. 200–220.

[BK23] Paul Beame and Sajin Koroth. "On Disperser/Lifting Properties of the Index and Inner-Product Functions". In: *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Vol. 251. Leibniz International Proceedings in Informatics (LIPIcs). 2023, 14:1–14:17. ISBN: 978-3-95977-263-1. DOI: 10.4230/LIPIcs.ITCS.2023.14.

[Bus87] Samuel R Buss. "Polynomial size proofs of the propositional pigeonhole principle". In: *The Journal of Symbolic Logic* 52.4 (1987), pp. 916–927.

[BW25] Paul Beame and Michael Whitmeyer. "Multiparty communication complexity of collision-finding and cutting planes proofs of concise pigeonhole principles". In: *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2025, pp. 21–1.

[CMSS23] Arkadev Chattopadhyay, Nikhil S Mande, Swagato Sanyal, and Suhail Sherif. "Lifting to Parity Decision Trees via Stifling". In: *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik. 2023.

[DGGM24] Stefan Dantchev, Nicola Galesi, Abdul Ghani, and Barnaby Martin. "Proof complexity and the binary encoding of combinatorial principles". In: *SIAM Journal on Computing* 53.3 (2024), pp. 764–802.

[dRV25] Susanna F de Rezende and Marc Vinyals. “Lifting with colourful sunflowers”. In: *40th Computational Complexity Conference (CCC 2025)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2025, pp. 36–1.

[EGI24] Klim Efremenko, Michal Garlík, and Dmitry Itsykson. “Lower Bounds for Regular Resolution over Parities”. In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 2024, pp. 640–651.

[EI25] Klim Efremenko and Dmitry Itsykson. “Amortized Closure and Its Applications in Lifting for Resolution over Parities”. In: *40th Computational Complexity Conference (CCC 2025)*. 2025.

[GOR24] Svyatoslav Gryaznov, Sergei Ovcharov, and Artur Riazanov. “Resolution Over Linear Equations: Combinatorial Games for Tree-like Size and Space”. In: *ACM Transactions on Computation Theory* 16.3 (2024), pp. 1–15.

[Hak85] Armin Haken. “The intractability of resolution”. In: *Theoretical computer science* 39 (1985), pp. 297–308.

[HP17] Pavel Hrubeš and Pavel Pudlák. “Random formulas, monotone circuits, and interpolation”. In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2017, pp. 121–131.

[IR21] Dmitry Itsykson and Artur Riazanov. “Proof complexity of natural formulas via communication arguments”. In: *36th Computational Complexity Conference (CCC 2021)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2021, pp. 3–1.

[IS20] Dmitry Itsykson and Dmitry Sokolov. “Resolution over linear equations modulo two”. In: *Annals of Pure and Applied Logic* 171.1 (2020), p. 102722.

[Jeř04] Emil Jeřábek. “Dual weak pigeonhole principle, Boolean complexity, and derandomization”. In: *Annals of Pure and Applied Logic* 129.1-3 (2004), pp. 1–37.

[MPW02] Alexis Maciel, Toniann Pitassi, and Alan R Woods. “A New Proof of the Weak Pigeonhole Principle”. In: *Journal of Computer and System Sciences* 64.4 (2002), pp. 843–872.

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[PS25] Vladimir Podolskii and Alexander Shekhovtsov. “Randomized Lifting to Semi-Structured Communication Complexity via Linear Diversity”. In: *16th Innovations in Theoretical Computer Science Conference (ITCS)*. LIPIcs. Schloss Dagstuhl, 2025.

[PWW88] Jeff B Paris, Alex J Wilkie, and Alan R. Woods. “Provability of the pigeonhole principle and the existence of infinitely many primes”. In: *The Journal of Symbolic Logic* 53.4 (1988), pp. 1235–1244.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il