Electronic Colloquium on Computational Complexity, Report No. 22 (2024)

Exponential Separation Between Powers of Regular and General Resolution Over Parities

Sreejata Kishor Bhattacharya* Arkadev Chattopadhyay† Pavel Dvořák‡

Abstract

Proving super-polynomial lower bounds on the size of proofs of unsatisfiability of Boolean formulas using resolution over parities, is an outstanding problem that has received a lot of attention after its introduction by Raz and Tzamaret [11]. Very recently, Efremenko, Garlík and Itsykson [5] proved the first exponential lower bounds on the size of ResLin proofs that were additionally restricted to be *bottom-regular*. We show that there are formulas for which such regular ResLin proofs of unsatisfiability continue to have exponential size even though there exists short proofs of their unsatisfiability in ordinary, non-regular resolution. This is the first super-polynomial separation between the power of general ResLin and and that of regular ResLin for any natural notion of regularity.

Our argument, while building upon the work of Efremenko et al, uses additional ideas from the literature on *lifting theorems* .

1 Introduction

One of the most basic and well studied proof systems in propositional proof complexity is resolution. Its weakness by now is reasonably well understood after years of research. Yet, this understanding is quite fragile as natural and simple strengthenings of resolution quickly pose challenges that remain outstanding. One such system is resolution over parities, introduced by Raz and Tzameret [11], which has been abbreviated as ResLin. More precisely, a *linear* clause is a disjunction of affine equations, generalizing the notion of ordinary clauses. If A and B are two such linear disjunctions and ℓ is a linear form, then the inference rule of ResLin derives the linear clause $A \vee B$ from clauses $A \vee (\ell = 1)$ and $B \vee (\ell = 0)$. To appreciate the power of this system, let's recall that a linear clause, unlike an ordinary clause, can be expressed using many different bases. Indeed, no super-polynomial lower bounds on the size of general proofs in this system is currently known for any explicit unsatisfiable Boolean formula.

Progress was first made in the work of Itsykson and Sokolov [9] when they proved exponential lower bounds on the size of tree-like ResLin proofs for central tautologies including the Pigeonhole Principle and Tseitin formulas over expanding graphs. Proving such lower bounds was systematized, only recently, in the independent works of Chattopadhyay, Mande, Sanyal and Sherif [4] and that of Beame and Koroth [2]. These works could be used to conclude that tree-like ResLin proofs are exponentially weaker than general ResLin proofs.

In the world of ordinary resolution, it is known that there is an intermediate proof system, known as *regular* resolution, whose power strictly lies in between tree-like and general proofs. In the graph of a regular resolution proof, no derivation path from an axiom clause to the final empty clause resolves a variable of the formula more than once. In the dual view of searching for a falsified clause, this corresponds precisely to read-once branching programs, where no source to sink path queries a variable more than once. Taking cue from this, Gryaznov, Pudlák and Talebanfard [8] introduced models of read-once linear branching programs (ROLBP), to capture notions of regularity in ResLin. They identified two such notions that extend the notion of regularity in ordinary resolution, or the read-once property of branching programs. Consider a node v of an ROLBP. Let $Pre(v)$ denote the vector space spanned by all the linear queries that appear in some path from the source node to v. Similarly, let $Post(v)$ denote the space spanned by all linear queries that lie in some path from v to a sink node. In the most restrictive notion, called strongly regular proofs or strongly read-once linear branching programs, $Pre(v)$

*TIFR-Mumbai, email: sreejata.bhattacharya@tifr.res.in

†TIFR - Mumbai, email: arkadev.c@tifr.res.in. Partially funded by the Department of Atomic Energy and a Google India Research Award.

‡TIFR-Mumbai & Charles University, Prague, email: koblich@iuuk.mff.cuni.cz. Work done entirely at TIFR. Supported by Czech Science Foundation GAČR grant #22-14872O.

ISSN 1433-8092

and $Post(v)$ have no non-trivial intersection for every v. In a more relaxed notion, called weak regularity or weakly ROLBP, the linear query made at node v is not contained in th $Pre(v)$. Gryaznov et al [8]. were able to prove an exponential lower bound on the size of strongly ROLBP for computing a *function*, using the notion of directional affine dispersers. However, their argument is not known to work for search problems for even strongly ROLBP.

There is another natural notion of weak regularity that complements the notion of weak regularity defined in [8]. In this notion, we forbid the linear query made at a node v to lie in an affine space $Post(u)$, for each u that is a child of v. We will call this notion bottom-regular and the former top-regular. Both top and bottom regularity are generalizations of strong regularity. Very recently, Efremenko, Garlik and Itsykson [5] proved the first exponential lower bounds on the size of bottom-regular ResLin proofs. The tautology they used was the Binary Pigeonhole Principle (BPHP). It is plausible that the BPHP remains hard for general ResLin proofs. One way to prove such a bound would be to show that every general ResLin proof could be converted to a bottom-regular ResLin proof with a (quasi-)polynomial blow up.

Our main result is a strong refutation of that possibility. We show that bottom-regular ResLin proofs require exponential size blow-up to simulate non-regular proofs even in the ordinary resolution system. The formulas we use are twists of certain formulas used by Alekhnovich, Johannsen, Pitassi and Urquhart [1] for providing separation between regular and general resolution. Alekhnovich et al provided two formulas for proving such a separation. In the second one, the starting point is a pebbling formula on pyramid graphs. It turns out that they are easy for regular resolution. To get around that, they consider *stone formula* for pyramid graphs, that they prove is hard for regular resolution while remaining easy for general resolution. We further obfuscate such stone formulas using another idea of Alekhnovich et al that appears in the construction of their first formula. Finally, we *lift* these formulas by logarithmic size Inner-product (IP) gadgets.

This lifted formula presents fresh difficulties to be overcome to carry out the implicit technique of Efremenko et al. We overcome them by exploitng two properties of the Inner-product. First, we exploit the property that IP has low discrepancy, invoking a result of Chattopadhyay, Filmus, Koroth, Meir and Pitassi [3]. Second, we use the fact that IP has the stifling property, inspired by the recent work of Chattopadhyay, Mande, Sanyal and Sherif [4].

Our unsatisfiable formula that yields a separation of resolution and bottom-regular ResLin is a stone formula for a pyramid graph G_n of n levels lifted by an inner-product gadget $\mathsf{IP} : \{0,1\}^b \to \{0,1\}$. We denote this formula as $\mathsf{SP}_n \circ \mathsf{IP}$ and it is defined over $M := 2N^2 \cdot b$ variables, where $b = \Theta(\log n)$ and $N = n(n+1)/2$ is the number of vertices of the pyramid graph G_n. For exact definition of $\mathsf{SP}_n \circ \mathsf{IP}$, see Section 2.

Theorem 1. *The formula $\mathsf{SP}_n \circ \mathsf{IP}$ admits a resolution refutation of length that is polynomial in M but any bottom-regular ResLin refutation of it must have length at least $2^{\Omega(M^{1/24}/\log M)}$.*

Comparision with Efremenko et al. [5]: Our work builds upon the very recent work of Efremenko, Garlík and Itsykson, who proved an exponential lower bound for the regular linear resolution complexity of the formula Binary-PHP$_{n,n+1}$. A crucial property of this formula that they use is that if we sample an assignment to the variables from the uniform distribution, with high probability one needs to make at least $n^{\Omega(1)}$ bit-queries to locate a falsified clause. Later, they use the following simple property of uniform distribution: let A be an affine subspace of co-dimension r. Then, the probability mass of A under uniform distribution is very small (inverse-exponential in r). Call this property (*).

Our goal is to show an exponential lower bound on the regular linear resolution complexity of a formula that has a small resolution refutation. A candidate formula would be a CNF which exhibits exponential separation between resolution and regular resolution. Some such formulas are $MGT_{n,\rho}$ and *stone formulas* (both defined in Alekhnovich et al [1]). However, all such formulas have constant width – and therefore, a uniformly random assignment falsifies a constant fraction of clauses. It follows that for both these formulas there is a query algorithm making only constantly many queries which finds a falsified clause with high probability under the uniform distribution. Thus, directly adapting the argument of Efremenko et al. would not work for these formulas.

Our main observation is that property (*) continues to hold for a much larger class of distributions than the uniform distribution when the base formula is *lifted* with an appropriate gadget. More precisely, if we take any distribution μ on the assignments of the base formula and consider its *uniform lift* μ', property (*) holds for μ'. We are now free to choose *any* distribution on the assignments of the base formula for which locating a falsifying axiom requires many queries on average (this is just a sketch;

we actually need something slightly stronger). This gives us enough freedom to construct appropriate distributions. Some more ingredients are required to make this idea work; we explain them in the subsequent sections.

2 A Formula Just Hard For Regular ResLin

Let us first recall the stone formula that was used by Alekhnovich et al. [1] for separating the powers of regular and general resolution. Let $G = (V, E)$ be a directed acyclic graph with exactly one root $r \in V$ and each vertex has out-degree 0 or 2. For $|V| = N$, we will use the following set of variables.

Vertex variables: For all $v \in V, 1 \leq j \leq N : P_{v,j}$.
Semantic interpretation: $P_{v,j}$ is set to 1 if stone j is placed on vertex v.

Stone variables: For all $1 \leq j \leq N : R_j$
Semantic interpretation: R_j is set to 1 if stone j is colored red, otherwise it is set to 0.

Auxiliary variables: For all $v \in V, 1 \leq j \leq N-1 : Z_{v,j}$
Semantic interpretation: These are auxiliary variables used to encode the fact that at least one stone is placed on vertex v and still our hard set contains only constant-width clauses.

Let $\mathcal{V}$ denote the set of variables and $\rho : [N]^3 \to \mathcal{V}$ be an arbitrary mapping that we call an *obfuscation map*. Let S be a set of all sinks of G. We define $\text{Stone}(G, \rho)$ to be the formula comprising the following set of clauses.

Root clauses: For all $1 \leq j \leq N$: $\neg P_{r,j} \vee \neg R_j$
Semantic interpretation: All stones placed on the root r of G must be coloured blue.

Sink clauses: For all $1 \leq j \leq N, s \in S$: $\neg P_{s,j} \vee R_j$
Semantic interpretation: Each stone placed on a sink of G must be coloured red.

Induction clauses: For all $v \in V(G)$ with out-neighbors u, w and for each $i, j, k \in [N]$:

$$\neg P_{u,i} \vee \neg R_i \vee \neg P_{w,j} \vee \neg R_j \vee \neg P_{v,k} \vee R_k \vee \rho(i,j,k)$$

$$\neg P_{u,i} \vee \neg R_i \vee \neg P_{w,j} \vee \neg R_j \vee \neg P_{v,k} \vee R_k \vee \neg\rho(i,j,k)$$

Semantic interpretation: after resolving out the variable $\rho(i, j, k)$, the clause says that if the stones placed on u and w are colored red, the stone placed at v must also be colored red, i.e., the implication

$$(P_{u,i} \wedge R_i \wedge P_{w,j} \wedge R_j \wedge P_{v,k}) \to R_k.$$

Stone-placement clauses: For all $v \in V(G)$:

$$\begin{aligned}
&P_{v,1} \vee \neg Z_{v,1} \\
&Z_{v,1} \vee P_{v,2} \vee \neg Z_{v,2} \\
&\cdots \\
&Z_{v,N-2} \vee P_{v,N-1} \vee \neg Z_{v,N-1} \\
&Z_{v,N-1} \vee P_{v,N}
\end{aligned}$$

Semantic interpretation: Together, the clauses are equivalent to

$$P_{v,1} \vee P_{v,2} \vee \cdots \vee P_{v,N}$$

i.e., at least one stone is placed on the vertex v.

Suppose we place red and blue stones on vertices of the directed acyclic graph G such that there is a blue stone on the root r and red stones on the sinks of G. Then, there has to be a non-sink vertex v with a blue stone such that the two out-neighbors u, w need to have red stones placed on them – this would falsify one of the induction clause for the vertices v, u, and w. Thus, the set of clauses $\text{Stone}(G, \rho)$ is unsatisfiable.

We instantiate Stone(G, ρ) for a pyramid graph. We set $G_n = (V, E)$ to be the pyramid graph of n levels of vertices where the i-th level contains i vertices. Thus, $V = \{(i, j) \mid 1 \le i \le n, 1 \le j \le i\}$, where the level of a vertex (i, j) is its first coordinate i. We have $|V| = N = \frac{1}{2}n(n+1)$. We define the edge set $E = \big\{\big((i, j), (i+1, j)\big), \big((i, j), (i+1, j+1)\big) \mid 1 \le i < n, 1 \le j \le i\big\}$. Thus, from each vertex (i, j) that is not in the last level there are two edges to the 'nearest' two vertices in the next level, i.e., $(i+1, j)$ and $(i+1, j+1)$. The set of sinks of G_n consists of the vertices of the last level, i.e., the vertices (n, j). The root r of G_n is $(1, 1)$, the only vertex in the first level. We define the set of clauses $\mathsf{SP}_{n,\rho}$ (Stone formula for a Pyramid graph) to be Stone(G_n, ρ). We denote the number of variables of $\mathsf{SP}_{n,\rho}$ by m, i.e. $|\mathcal{V}| = m = N^2 + N + N(N-1) = 2N^2$.

In order to prove our lower bound against regular ResLin, it turns out to be convenient working with a formula that is obtained by *lifting* $\mathsf{SP}_{n,\rho}$. Let $g : \{0,1\}^b \to \{0,1\}$ be a Boolean function, called gadget. For $a \in \{0,1\}$, we denote the set of all pre-images of a by $g^{-1}(a)$, i.e., $g^{-1}(a) \coloneqq \big\{x \in \{0,1\}^b \mid g(x) = a\big\}$.

Let $C = [X_1 = c_1] \vee \cdots \vee [X_k = c_k]$ be a clause over variables $X_1, \ldots, X_k$ and $c_1, \ldots, c_k \in \{0,1\}$, where $[X_i = c_i]$ denotes a literal, i.e. X_i if $c_i = 1$, and $\neg X_i$ otherwise. To lift the clause C we introduce b variables $Y_1^i, \ldots, Y_b^i$ for each variable X_i of C. Then, we replace C by a set of clauses that are equivalent to the formula C, where we replace each literal $[X_i = c_i]$ by a subformula with a semantic interpretation that $(Y_1^i, \ldots, Y_b^i)$ equals to some $(d_1^i, \ldots, d_b^i) \in g^{-1}(c_i)$. For a technical reason, we also need that in each clause of $C \circ g$ all variables Y_j^i's are present. We define the lift as

$$C \circ g \coloneqq \left\{ \bigvee_{1 \le i \le k, 1 \le j \le b} [Y_j^i = 1 - d_j^i] \;\middle|\; d^i \in g^{-1}(1 - c_1), \ldots, d_k \in g^{-1}(1 - c_k) \right\},$$

Observation 1. *An assignment $(\alpha_1, \ldots, \alpha_k)$ falsifies clause C if and only if for every assignment $(\beta^1, \ldots, \beta^k)$, with $\beta^i \in g^{-1}(\alpha_i)$ falsifies a clause in $C \circ g$. Moreover, each such assignment $(\beta^1, \ldots, \beta^k)$ falsifies exactly one clause in $C \circ g$.*

Proof. Let $\beta^i \in g^{-1}(\alpha_i)$. Suppose the assignment $(\beta^1, \ldots, \beta^k)$ does not satisfy a clause $\bigvee[Y_j^i = 1 - d_j^i]$ for $d^i \in g^{-1}(1 - c_i)$. Thus, $\beta_j^i = d_j^i$ for all $i \in [k], j \in [b]$ and consequently $\alpha_i = 1 - c_i$ for all $i \in [k]$. Therefore, the assignment $(\alpha_1, \ldots, \alpha_k)$ does not satisfy the clause $C = \bigvee[X_i = c_i]$.

On the other hand, suppose the assignment $(\alpha_1, \ldots, \alpha_k)$ does not satisfy the clause C, i.e., $\alpha_i = 1 - c_i$ for all $i \in [k]$. Then for all $i \in [k]$, pick $d^i = (\beta_1^i, \ldots, \beta_b^i) \in g^{-1}(1 - c_i)$. The assignment $(\beta^1, \ldots, \beta^k)$ falsify the clause $\bigvee[Y_j^i = 1 - d_j^i]$. It is clear that for any other $(d'^1, \ldots, d'^k) \neq (\beta^1, \ldots, \beta^k)$ the assignment $(\beta^1, \ldots, \beta^k)$ satisfies the clause $\bigvee[Y_j^i = 1 - d_j'^i]$.

□

For a set of clauses Φ (a CNF formula), we define its lift as $\Phi \circ g \coloneqq \cup_{C \in \Phi}(C \circ g)$. We have the following corollary of Observation 1.

Corollary 1. *The set of clauses Φ is unsatisfiable if and only if the set of clauses $\Phi \circ g$ is unsatisfiable.*

We remark that if the base set Φ contains only clauses of width at most k, then $\Phi \circ g$ contains clauses of width at most kb and $|\Phi \circ g| \le 2^{bk} \cdot |\Phi|$. In particular, a constant-width, poly-size unsatisfiable formula, when lifted by an $O(\log n)$ size gadget, yields an $O(\log n)$-width, poly-size unsatisfiable formula. Our hard problem will be the stone formula lifted by an inner product gadget $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$, where $\mathsf{IP} : \{0,1\}^b \to \{0,1\}$ is the inner product function for $b = O(\log n)$.

3 Resolution Proof Systems and Branching Programs

A proof in a propositional proof system starts from a set of clauses S, called axioms, that is purportedly unsatisfiable. It generates a proof by deriving the empty clause from the axioms, using inference rules. The main inference rule in the standard resolution, called the resolution rule, derives a clause $A \vee B$ from clauses $A \vee x$ and $B \vee \neg x$ (i.e., we resolve the variable x). If we can derive the empty clause from the original set S then it proves the set S is unsatisfiable. We will need the following basic and well known fact that states resolution is complete without being too inefficient:

Lemma 1. *Let C be any clause, and F be any CNF formula over n Boolean variables and of polynomial size, that semantically implies C. Then, C can be derived from F by a resolution proof of size at most $2^{O(n)}$.*

Linear resolution, aka ResLin and introduced by Raz and Tzamaret [11], is a generalization of standard resolution, using linear clauses (disjunction of linear equations over $\mathbb{F}_2$) to express lines of a proof. It consists of two rules:

Resolution Rule: From linear clauses $A \vee (f = 0)$ and $B \vee (f = 1)$ derive a linear clause $A \vee B$.

Weakening Rule: From a linear clause A derive a linear clause B that is semantically implied by A (i.e., any assignment satisfying A also satisfies B).

The length of a resolution (or ResLin) refutation of a formula Φ is the number of applications of the rules above in order to refute the formula Φ. The width of a resolution (or ResLin) refutation is the maximum width of any (linear) clause that is used in the resolution proof.

It is known that a resolution proof and a linear resolution proof, for an unsatisfiable set of clauses S, correspond to a branching program and a linear branching program, respectively, for a search problem *Search*(S) (see for example Garg et al [6], who credit it to earlier work of Razborov [12] that was simplified by Pudlák [10] and Sokolov[13]) that is defined as follows. For a given assignment α of the n variables of S, one needs to find a clause in S that is unsatisfied by α (at least one exists as the set S is unsatisfiable). A linear branching program computing a search problem $\mathsf{P} \subseteq \mathbb{F}_2^n \times O$ is defined as follows.

- There is a directed acyclic graph $\mathcal{P}$ of one source and some sinks. Each non-sink node has out-degree at most two. For an inner node v the two out-neighbors u and w (i.e., there are edges (v, u) and (v, w) in $\mathcal{P}$) are called children of v.
- Each node v of $\mathcal{P}$ is labeled by an affine space $A_v \subseteq \mathbb{F}_2^n$.
- The source is labeled by $\mathbb{F}_2^n$.
- Let v be a node of out-degree 2 and u and w be children of v. Then, $A_u = A_v^0$ and $A_w = A_v^1$, where $A_v^c = \{x \in A_v \mid \langle f_v, x\rangle = c\}$ for a linear query $f_v = \mathbb{F}_2^n$ and $c \in \{0, 1\}$. We call such v a *query node.*
- Let v be a node of out-degree 1 and u be the child of v. Then, $A_v \subseteq A_u$. We call such v a forget node. then $A_v \subseteq A_u \cup A_w$.
- Each sink v of $\mathcal{P}$ has an assigned output $o_v \in O$ such that A_v is o_v-monochromatic according to P, i.e., $\alpha \in A_v \implies (\alpha, o_v) \in \mathsf{P}$.

A standard/ordinary branching program is defined analogously but its nodes are labeled by cubes instead of affine spaces. Consequently, variables instead of arbitrary linear functions are queried at its query nodes.

The correspondence between a branching program computing *Search*(S) and a (linear) resolution proof refuting S is roughly the following. We can represent the resolution proof as a directed acyclic graph where nodes are labeled by (linear) clauses. The sources are labeled by clauses of S and there is exactly one sink that is labeled by an empty clause. Each node that is not a source has at most two parents and it corresponds to an application of the (linear) resolution rule (if the node has 2 parents), or the weakening rule (if the nodes has 1 parent). To get a (linear) branching program for *Search*(S) we just flip the direction of the edges in the resolution graph and negate the clauses that are used for node labeling. Thus, each node is labeled by a cube or an affine space, the query nodes correspond to applications of the resolution rule, and the forget nodes correspond to applications of the weakening rule. It is clear the size of a branching program $\mathcal{P}$ (number of nodes of $\mathcal{P}$) is exactly the same as length of the corresponding resolution refutation.

Regular resolution is a subsystem of the resolution system, such that in any path of the resolution proof graph each variable can be resolved at most once. A read-once branching program corresponds to a regular resolution proof, i.e., on each directed path from the source to a sink each variable is queried at most once. There is interest in two generalizations of regular resolution to linear regular resolution – top-regular linear resolution [8] and bottom-regular linear resolution [5] (in both papers called as regular linear resolution). We will define both of them by their corresponding linear branching programs.

Definition 1 ([8])**.** *Let v be a node of a linear branching program with an underlying directed acyclic graph $\mathcal{P}$. Let Pre(v) be the space spanned by all linear functions queried on any path from the source of $\mathcal{P}$ to v. Let Post(v) be the space spanned by all linear functions queried on any path from v to any sink of $\mathcal{P}$.*

A linear branching program is *top-read-once*[1] [8] if for each query node v, we have $f_v \notin Pre(v)$. A linear branching program is *bottom-read-once* [5] if for each edge (v, u) such that v is a query node holds that $f_v \notin Post(u)$. A linear resolution proof is *top-regular*, or *bottom-regular* if the corresponding branching program is top-read-once, or bottom-read-once, respectively. We use both notion of branching program and resolution to state and prove our result, whichever is more suitable for the presentation. Our separation is only for bottom-regular ResLin, i.e., for bottom-read-once linear branching program, which we abbreviate to BROLBP.

Lemma 2 (Lemma 2.6 [5] stated for branching programs)**.** *Let $\mathcal{P}$ be a BROLBP computing a search problem $\mathsf{P} \subseteq \{0,1\}^n \times O$. Let v be a node of $\mathcal{P}$ such that there is a path of length t from the source of $\mathcal{P}$ to v. Then,* $\dim(Post(v)) \leq n - t$.

Even more restrictive subsystems, are tree-like resolution, and tree-like ResLin. These subsystems correspond to decision trees and parity decision trees. A parity decision tree (PDT) is a branching program such that its underlying graph is a tree, and a decision tree (DT) is a restriction where we query only bits of the input, instead of linear functions. It is clear that tree-like resolution is a subsystem of regular resolution. Analagously, the tree-like ResLin is a subsystem of both bottom-regular and top-regular ResLin.

4 Linear Algebraic Facts

In this section, we will describe the notions of linear algebra that we will need in our arguments. Let $M \in \mathbb{F}_2^{m \times t}$ be a matrix. We denote the row space of M by $\mathcal{R}(M)$. For a vector $c \in \{0,1\}^t$, an affine space $\mathcal{S}(M, c)$ is a space of solution of the linear system $(M|c)$, i.e., $\mathcal{S}(M, c) = \{\alpha \in \{0,1\}^m \mid M \cdot \alpha = c\}$.

We interpret a vector $\alpha \in \mathbb{F}_2^m$ also as a 0/1-assignment of m variables. A partial assignment α' of m variables is a string in $\{0, 1, *\}^m$. A variable X is *assigned* if $\alpha_X \in \{0,1\}$. For a total assignment $\alpha \in \{0,1\}^m$ and $T \subseteq [m]$ we define a restriction $\alpha_{|T}$ of α to T to be a partial assignment arising from α by unassigning the variables that are not in T, i.e., for each $i \in [m]$

$$(\alpha_{|T})_i = \begin{cases} \alpha_i & \text{if } i \in T \\ * & \text{otherwise.} \end{cases}$$

The only property we need from our gadget in this section is stifling introduced by Chattopadhyay et al. [4]. Thus, we state the results in this section for a general stifled gadget $g : \{0,1\}^b \to \{0,1\}$.

Definition 2. *A Boolean function $g : \{0,1\}^b \to \{0,1\}$ is* stifled[2] *if the following holds*

$$\forall i \in [b] \text{ and } a \in \{0,1\}\ \exists \delta \in \{0,1\}^b$$
$$\text{such that for all } \gamma \in \{0,1\}^b \text{ with } \gamma_{|[b]\setminus\{i\}} = \delta_{|[b]\setminus\{i\}} \text{ holds that } g(\gamma) = a.$$

We call δ from the previous definition a *stifling assignment*. The utility of stifling is the following. An adversary can pick any variable $i \in [b]$ of g. For arbitrary $a \in \{0,1\}$ we pick a partial assignment $\delta_a \in \{0, 1, *\}^m$ that assign a value to all variables except the i-th variable. Now, no matter how the adversary choses the value for the i-th variable to get a total assignment $\gamma_a \in \{0,1\}^b$ from δ_a, the value $g(\gamma_a)$ will be always a.

The entries of vectors of $\mathbb{F}_2^{mb}$ are naturally divided into m blocks, each having b co-ordinates/bits, i.e., for $j \in [m]$, the j-th block contains the coordinates $(j-1)b+1, \ldots, jb$. For a vector $u \in \mathbb{F}_2^{mb}$ and a block $j \in [m]$ a vector $u^j \in \mathbb{F}_2^b$ is a vector corresponding to the block j of u, i.e., $u^j = (u_{(j-1)b+1}, \ldots, u_{jb})$. For the block structure, we adapt a notion of closure introduced by Efremenko et al. [5].

Let $R = (u_1, \ldots, u_t), u_i \in \mathbb{F}_2^{mb}$ be a tuple of row vectors that forms a matrix $M \in \mathbb{F}_2^{mb \times t}$ in echelon form. I.e., there are coordinates $a_1 < a_2 < \cdots < a_t \in [mb]$ such that for all $i \in [t]$: $(u_i)_{a_i} = 1$, and $(u_i)_{a_j} = 0$ for all $j \neq i$. Thus, the matrix M restricted to the columns $a_1, \ldots, a_t$ is the identity matrix $I_t \in \mathbb{F}_2^{t \times t}$. The entry a_i is called *pivot* of u_i. The tuple R is *safe* if each pivot a_i is in different block $j \in [m]$.

We say a vector $u \in \mathbb{F}_2^{mb}$ *touches* a block $j \in [m]$ if the vector u^j is non-zero. A set of vectors $R \subseteq \mathbb{F}_2^{mb}$ touches a block j if at least one of the vectors in R touches j. A subspace U of $\mathbb{F}_2^{mb}$ is *spread* if any k linearly independent vectors of U touches at least k blocks, for each $1 \leq k \leq m$. Efremenko et al. [5] showed the following result.

[1] Gryaznov et al. [8] used just the name weakly read once for such programs.

[2] 1-stifled called by Chattopadhyay et al. [4]

Theorem 2 (Theorem 3.1, Efremenko et al. [5]). *Let U be a spread subspace of $\mathbb{F}_2^{mb}$, and dimension of U is at most m. Then, U has a safe basis.*

Let U be a subspace of $\mathbb{F}_2^{mb}$ and $T \subseteq [m]$ be a set of blocks. The subspace U_T of U is the linear space of all vectors u that do not touch any block outside T, i.e., $U_T = \{u \in U \mid \forall j \notin T : u^j = (0, \ldots, 0)\}$. The subspace $U_{\downarrow T}$ of $\mathbb{F}_2^{mb}$ is the projection of U onto T, i.e., all co-ordinates lying outside T are replaced by zeroes. The following is a basic fact.

Observation 2. *Let U be any subspace of $\mathbb{F}_2^{mb}$ and $T \subseteq [m]$ be a set of blocks. Then,*

$$dim(U) = dim(U_T) + dim\big(U_{\downarrow \bar{T}}\big).$$

Proof. Let U' be a suitable subspace of U such that $U = U_T \oplus U'$. It is simple to see that $U_{\downarrow \bar{T}} = (U')_{\downarrow \bar{T}}$. This is because every vector $u \in U$ can be written as $x+u'$, with $x \in U_T$ and $u' \in U'$. But, as $x_{\downarrow \bar{T}} = 0$, we have $u_{\downarrow \bar{T}} = u'_{\downarrow \bar{T}}$. Hence, we conclude that $\dim\big(U_{\downarrow \bar{T}}\big) = \dim\big((U')_{\downarrow \bar{T}}\big) \leq \dim\big(U'\big)$. To establish our result, we will simply show that $\dim(U') \leq \dim\big((U')_{\downarrow \bar{T}}\big)$. This follows if we show that whenever $u'_1, \ldots, u'_r \in U'$ are linearly independent vectors, so are $\big(u'_1\big)_{\downarrow \bar{T}}, \ldots, \big(u'_r\big)_{\downarrow \bar{T}}$. If that is not the case then there exists a vector $x \in U_T$ such that $x, u'_1, \ldots, u'_r$ are not linearly independent, contradicting our assumption. □

We say a set of blocks $T \subseteq [m]$ is an *obstruction* of a space U if $U_{\downarrow \bar{T}}$ is spread, where $\bar{T}$ is complement of T, i.e., $\bar{T} = [m] \setminus T$. An obstruction $T \subseteq [m]$ of a space U is minimal if any proper subset $T' \subset T$ is not an obstruction of U, i.e., $U_{\downarrow \bar{T}'}$ is not spread. Efremenko et al. [5] showed the following properties of minimal obstructions.

Lemma 3. *Let U be a subspace of $\mathbb{F}_2^{mb}$. Then, a minimal obstruction $T \subseteq [m]$ of U is unique and $|T| \leq \dim(U)$.*

Exploiting Lemma 3, we define a closure $\mathrm{Cl}(A)$ of an affine space $A = \mathcal{S}(M, c) \subseteq \mathbb{F}_2^{mb}$ to be the minimal obstruction $T \subseteq [m]$ of the row space of M. A set $\mathrm{VarCl}(A) \subseteq [mb]$ of closure variables is a set of variables that appear in the blocks of the closure $\mathrm{Cl}(A) \subseteq [m]$. Further, Efremenko et al. [5] proved the following relationship between the closures of two affine spaces when one contains the other.

Lemma 4. *Let $A \subseteq A'$ be two affine spaces of $\mathbb{F}_2^{mb}$. Then, $Cl(A') \subseteq Cl(A)$.*

Let $\beta \in \{0, 1, *\}^{mb}$ be a partial assignment of variables of $\mathsf{SP}_{n,\rho} \circ g$. We say the partial assignment β is *block-respecting* if for each block $j \in [m]$ either all variables of the j-th block are assigned, i.e., $\beta_i^j \in \{0, 1\}$ for all $i \in [b]$, or none of them are assigned, i.e., $\beta_i^j = *$ for all $i \in [b]$. A block-respecting assignment $\beta \in \{0, 1, *\}^{mb}$ gives us naturally a partial assignment $\overrightarrow{g}(\beta) \in \{0, 1, *\}^m$ by applying the gadget g to the assigned blocks. Formally, for each $j \in [m]$ we have

$$\overrightarrow{g}(\beta)_j = \begin{cases} g(\beta_1^j, \ldots, \beta_b^j) & \text{if for all } i \in [b] : \beta_i^j \text{ are assigned,} \\ * & \text{otherwise.} \end{cases}$$

Let $A \subseteq \mathbb{F}_2^{mb}$ be an affine space and $\beta \in A$. A *closure assignment* $\beta_{|\mathrm{VarCl}(A)} \in \{0, 1, *\}^{mb}$ is a restriction of β to the set of variables in the closure of A (i.e., to $\mathrm{VarCl}(A)$). Note that any closure assignment is block-respecting.

Lemma 5. *Let $A = \mathcal{S}(M, c) \subseteq \mathbb{F}^{mb}$ be an affine space and let $g : \{0, 1\}^b \to \{0, 1\}$ be a stifled gadget. Let $\beta' \in \{0, 1, *\}^{mb}$ be a closure assignment and $\alpha' := \overrightarrow{g}(\beta') \in \{0, 1, *\}^m$. Then, for any extension of α' to a total assignment $\alpha \in \{0, 1\}^m$, there exists $\beta \in A$ such that $\overrightarrow{g}(\beta) = \alpha$.*

Proof. First, we modify M to an appropriate form M'. We assume, wlog, the rows of M are linearly independent. Let $U = \mathcal{R}(M)$ and $T \subseteq [m]$ be the closure of A. Let $(u_1, \ldots, u_d)$ be an arbitrary basis of U_T and let M_1 be the matrix whose rows are the vectors $u_1, \ldots, u_d$. Note that by definition of the closure, the matrix M_1 contains only 0 on blocks that are not in T. The first block of M' consists of the matrix M_1.

Further, let $(w_1, \ldots, w_{d'})$ be a safe basis of $U_{\downarrow \bar{T}}$. Such a basis exists by the definition of closure and Theorem 2. Let $a_1, \ldots, a_{d'}$ be pivots of $w_1, \ldots, w_{d'}$. By definition of safe basis, each of these pivots are in a distinct block. Moreover, none of these pivots are in the blocks of T.

Recall that $U_{\downarrow \bar{T}}$ arises from U by zeroing-out the blocks in T. Thus, there is a linear mapping $L : U \to U_{\downarrow \bar{T}}$ defined as

$$L(u)^j = \begin{cases} u^j & \text{if } j \notin T, \\ (0, \ldots, 0) & \text{otherwise.} \end{cases}$$

Now, let w'_i be an arbitrary pre-image of w_i according to L, i.e., $L(w'_i) = w_i$. Since $(w_1, \ldots, w_{d'})$ are linearly independent, the vectors $(w'_1, \ldots, w'_{d'})$ are linearly independent as well. Let M_2 be a matrix which rows are the vectors $w'_1, \ldots, w'_{d'}$. Note that the columns of M_2 indexed by the pivots $a_1, \ldots, a_{d'}$ still form the identity matrix. The second block of M' is the matrix M_2, which finishes the construction of M' that looks as follows.

$$M' = \begin{array}{c|c|c||l} B_1 & 0 & 0 & = M_1 \\ B_2 & I_{d'} & B_3 & = M_2 \\ \hline \text{Closure } T & \text{Pivots of } M_2 & & \end{array}$$

By Observation 2, $\dim(U) = \dim(U_T) + \dim(U_{\downarrow \bar{T}})$. Further, $d = \dim(U_T)$, and $d' = \dim(U_{\downarrow \bar{T}})$. The rows of M_1 and M_2 are linearly independent. Moreover, no row of M_2 can be generated by rows of M_1 as pivots $(a_1, \ldots, a_{d'})$ of the matrix M_2 corresponds to the columns where the matrix M_1 has only 0 entries. Thus, $\operatorname{rank}(M') = \dim(U) = \operatorname{rank}(M)$. Further, any row of M' is a vector of $U = \mathcal{R}(M)$ and therefore $\mathcal{R}(M) = \mathcal{R}(M')$. Thus, there is a vector c' of appropriate length such that $A = \mathcal{S}(M', c')$.

Now, let $\alpha \in \{0,1\}^m$ be an extension of α'. For α, we need to find a pre-image β of according to g such that $\beta \in A$. We pick β to be an extension of β'. Let c_1 and c_2 be parts of c' that corresponds to M_1 and M_2, respectively, i.e. $A = \{\beta \in \mathbb{F}_2^{mb} \mid M_1 \cdot \beta = c_1, M_2 \cdot \beta = c_2\}$. Since β' is a closure assignment, then for the extension β holds that $M_1 \cdot \beta = c_1$ as M_1 has only 0 entries outside the closure T. Thus, it remains to set blocks of β that are outside the closure T.

Let $S \subseteq [m]$ be a set of blocks of pivots of M_2, i.e., the blocks that contains coordinates $a_1, \ldots, a_{d'}$. The blocks in the closure T are already set by β'. We set the blocks of β not in T and S arbitrarily according to α. I.e., for $j \in [m] \setminus (S \cup T)$, we set β^j to be an arbitrary vector $\gamma \in \{0,1\}^b$ such that $g(\gamma) = \alpha_j$.

For the blocks in S we use the stifling property of g. Let $j \in S$ and the j-th block contains the pivot a_ℓ (recall that each block in S contains exactly one pivot from $a_1, \ldots, a_{d'}$). Say, the pivot a_ℓ is the i-th variable in the block j. Since g is stifled, there is a stifling assignment $\delta_a \in \{0,1\}^b$ for the i-th variable and $a = \alpha_j$. We set all entries of β^j, except the i-th entry, according to δ_a, i.e., $\beta^j_{|[b]\setminus\{i\}} = (\delta_a)_{|[b]\setminus\{i\}}$. Now, no matter how we set β^j_i it will always holds that $g(\beta^j) = a = \alpha_j$ because g is stifled. Note that the columns a_ℓ of the matrix M' contains 1 only in the row corresponding to the ℓ-th equation E_ℓ of the system $(M_2|c_2)$. Moreover, the equation E_ℓ has only 0 on the entries corresponding to the other pivots $a_1, \ldots, a_{d'}$ different from a_ℓ. Thus, we can set β^j_i to satisfy the equation E_ℓ and this equation will be satisfied no matter how we set the entries corresponding to other pivots as the equation E_ℓ does not depend on them. We can repeat this procedure for all blocks in S to produce the sought assignment $\beta \in A$ such that $\overrightarrow{g}(\beta) = \alpha$.

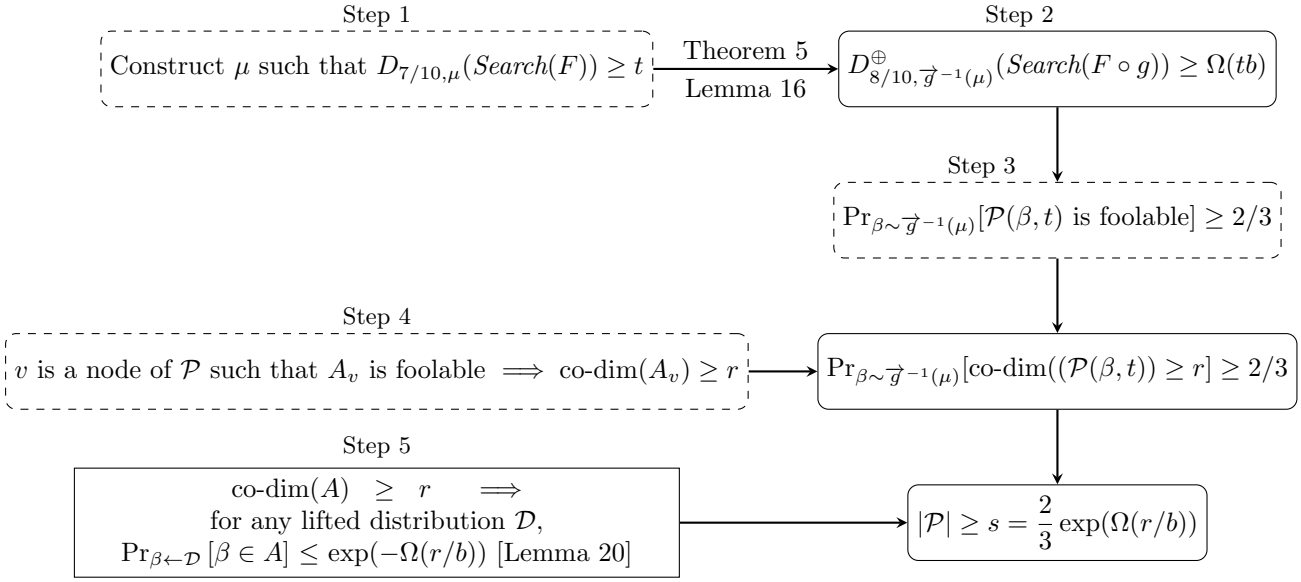
□

5 Proof Outline

In this section, we provide an outline the proof of our main result, Theorem 1. The proof consists of two parts. The first part shows that the formula $\text{Stone}(G, \rho) \circ g$ has a polynomial length resolution proof for any directed acyclic graph G on N vertices and out-degree 2, any obfuscation map $\rho : [N]^3 \to \mathcal{V}$, and any gadget $g : \{0,1\}^b \to \{0,1\}$, where b is logarithmic in N (recall that the number of variables m of the formula $\text{Stone}(G, \rho)$ is $2N^2$). This part of the proof is an adaptation of an analogous proof for the stone formula given by Alekhnovich et al. [1].

The second part establishes that there is a graph G and an obfuscation map $\rho : [N]^3 \to \mathcal{V}$ such that any bottom-regular ResLin proof of $\text{Stone}(G, \rho) \circ \mathsf{IP}$ has exponential length in m, where IP is the inner product function on $b = \Theta(\log m)$ bits. The proof of this part is involved and non-trivial. We outline the main steps in the figure below, immediately followed by a high-level description of each step depicted.

Outline of the Lower Bound Proof Our argument is an adaptation of the method presented in Efremenko et al [5] with addition of some new ingredients. In the picture below, depicting the method, the solid boxes refer to parts that are quite general and not specific to a formula, while the dashed boxes contain modules that are more specific to $\mathsf{SP}_n \circ \mathsf{IP}$ and similar formulas. Let $\mathcal{P}(\beta, t)$ be the node that $\mathcal{P}$ arrives at after making t linear queries on β.



Some Details: Let $\mathcal{P}$ be a bottom-read-once branching program computing $Search(\text{Stone}(G,\rho) \circ \mathsf{IP})$ corresponding to a bottom-regular ResLin proof of $\text{Stone}(G,\rho) \circ \mathsf{IP}$ where G is a pyramid graph of n levels and ρ is a carefully chosen obfuscation map.

The proof consists of several steps.

1. We design a distribution μ over the assignment of variables of the base formula F over m variables, typically supported over *critical assignments*, i.e. those which result in the falsification of exactly one clause. This module requires one to show that $Search(F)$ is average-case hard for deterministic decision trees of small height, wrt μ. In particular, our Lemma 15 proves that the problem $Search(\text{Stone}(G,\rho))$ is average-case hard for deterministic decision trees of height at most $O(n^{1/6})$. As the μ exhibited is formula specific, the box corresponding to this module is dashed.

2. In this step, we prove that the search problem associated with the lifted formula $F \circ g$ remains average case hard for *parity decision trees* wrt a *lifted distribution* as long as the gadget g has small rectangular discrepancy. More precisely, let $\overrightarrow{g}^{-1}(\mu)$ denote the lifted distribution generated by the following sampling: sample an input $z \in \{0,1\}^m$ according to μ. Then, sample at random an input $\beta \in \{0,1\}^{mb}$, conditioned on $\overrightarrow{g}(\beta) = z$. Using Theorem 5, implicit in the proof of the main result of Chattopadhyay, Filmus, Koroth, Meir and Pitassi [3], we conclude that $Search(F \circ g)$ is average-case hard for deterministic parity decision trees of small height, under the lifted distribution $\overrightarrow{g}^{-1}(\mu)$. This step is generic and works for any gadget of size $c \cdot \log(m)$, that has sufficiently small rectangular discrepancy under the uniform distribution over $\{0,1\}^b$. The gadget we use here is IP.

3. We then want to define a notion of progress the branching program $\mathcal{P}$ has made on arriving at a node v. To do so, consider the affine space A_v that labels the node. A_v may have nearly fixed/exposed the values of some of the blocks of input. These dangerous blocks are precisely $\text{Cl}(A_v)$ as defined in Section 4. They form the minimum obstruction set. Intuitively, the danger is $\mathcal{P}$ may have nearly found out a falsified clause of $F \circ g$ on reaching v if that clause was made up entirely of variables from blocks in $\text{Cl}(A_v)$. However, in this step we observe that the average-case hardness of the Search problem for PDTs proved in the previous step precludes this from happening with appreciable probability, when the input is sampled according to the lifted distribution $\overrightarrow{g}^{-1}(\mu)$. To formalize this idea, we need to concretely say when A_v is (not) dangerous. So far, we have not been able to lay out a general notion of danger, but notions specific to individual formulas have been defined. For $\text{Stone}(G,\rho)$, this notion is captured by Definition 4 of *foolable* spaces, provided in Section 7.3. Theorem 6 shows that w.h.p., $\mathcal{P}$ reaches a foolable space on walking for $n^{1/6}$ steps, querying an input sampled according to $\overrightarrow{g}^{-1}(\mu)$.

4. In this step, we show that when the affine space A_v is not dangerous, i.e. it is foolable or consistent, the appropriate notion depending on the formula at hand, A_v has large co-dimension. All steps

until now held for general branching programs (or equivalently proof DAGs). This step is the only one where the bottom-read-once property is exploited. For Stone(G, ρ), this is achieved in Section 7.4, at the end, by Lemma 19.

5. In this step, we prove a general result about lifted distributions. For any affine space A of co-dim$(A) = r$ and any distribution μ on $\{0,1\}^m$, we prove that β sampled by $\overrightarrow{g}^{-1}(\mu)$ is in A only with probability $2^{-\Omega(r/b)}$, as long as the gadget g is balanced and stifling. In other words, lifted distributions, even though their support is quite sparse in the ambient space, are pseudo-random for the rank measure. This property, though simple to prove, turns out to be extremely useful, especially for formulas like the stone formulas that are barely hard.

At this stage we are ready to put together the above steps in the following way. Let R be a set of nodes w of $\mathcal{P}$ such that there is a path from the root of $\mathcal{P}$ to w of length t, and co-dim$(A_w) \geq t$. Setting $t := n^{1/6}$ we have the following.

$$\begin{aligned}
\frac{7}{10} &\leq \Pr_{\beta \sim \overrightarrow{\mathsf{IP}}^{-1}(\mu)} \left[\text{co-dim}(A_v) \geq t \text{ for } v = \mathcal{P}(\beta, t)\right] && \text{(by Step 3 and 4)} \\
&\leq \sum_{w \in R} \Pr_{\beta \sim \overrightarrow{\mathsf{IP}}^{-1}(\mu)} \left[\mathcal{P}(\beta, t) = w\right] && \text{(by union bound)} \\
&\leq |R| \cdot 2^{\Omega(-t/b)} && \text{(by Step 5)}
\end{aligned}$$

By rearranging, we get a lower bound $|R| \geq 2^{\Omega(n^{1/6}/\log n)}$ that yields a lower bound $2^{\Omega(M^{1/24}/\log M)}$ for the size of $\mathcal{P}$ as the number of variables of *Search*(Stone$(G, \rho) \circ \mathsf{IP}$) is $M = n^4$.

6 Upper Bound

In this section, we show the upper bound part of Theorem 1.

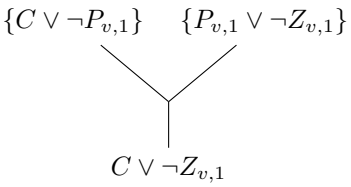
Theorem 3. *Let $G = (V, E)$ be an directed acyclic graph with N vertices, exactly one root r, and each non-sink vertex has out-degre 2. Let $\rho : [N]^3 \to \mathcal{V}$ be any obfuscation map, and $g : \{0,1\}^b \to \{0,1\}$ be a Boolean function for $b \leq O(\log N)$. Then, the formula Stone$(G, \rho) \circ g$ admits a resolution refutation of length polynomial in N.*

The proof of Theorem 3 is an adaptation of the proof given by Alekhnevich et al. [1] for lifted formulas. We remark that Alekhnevich et al. [1] presented a resolution refutation for the stone formulas of constant width. This allow us to adapt the refutation for the lifted formula. For the rest of the subsection, we fix a graph G, an obfuscation map ρ, and the gadget g given by the assumption of Theorem 3. First, we prove several auxiliary lemmas about the formula Stone$(G, \rho) \circ g$.

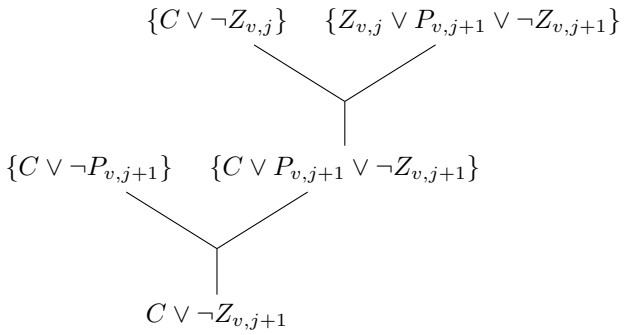
Lemma 6. *Let C be a clause with width w. Suppose we have derived the clauses $C \vee \neg P_{v,j}$ for a fixed $v \in V$ and all $1 \leq j \leq N$. Then, we can derive C in N steps in width $\leq w + 2$.*

Proof. We derive the clause C in N steps. We will subsequently derive $C \vee \neg Z_{v,j+1}$ from $C \vee \neg Z_{v,j}$

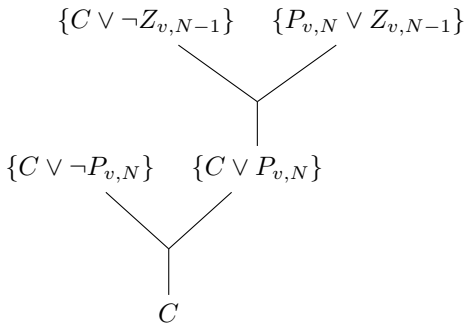
Base step: Deriving $C \vee \neg Z_{v,1}$.



Step j: For $j \in [N-2]$, deriving $C \vee \neg Z_{v,j+1}$ from $C \vee \neg Z_{v,j}$.



Final step: Deriving C.



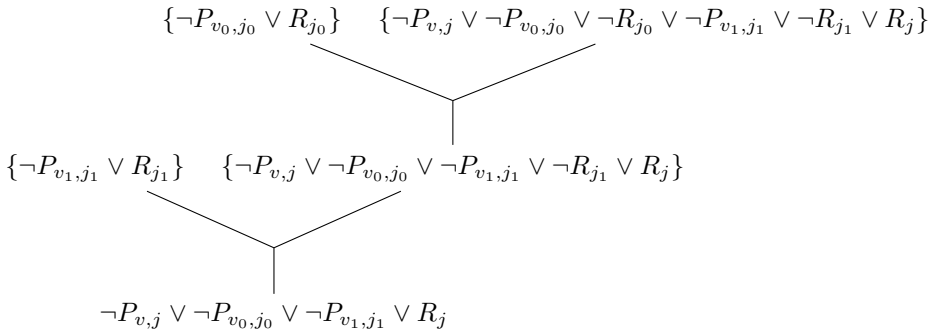
□

For a vertex v, we define the set of clauses $S(v) = \{\neg P_{v,j} \vee R_j | 1 \leq j \leq N\}$.

Lemma 7. *Let v be a vertex in G with children v_0, v_1. We can derive $S(v)$ from $S(v_0)$, and $S(v_1)$ in constant width and length $O(N^3)$.*

Proof. We derive $S(v)$ in several steps.

1. For every $j, j_0, j_1 \in [N]$, we perform the following sequence of operations:



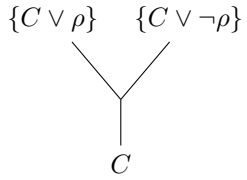
2. For each fixed j_0, j, we apply Lemma 6 to the clause $C \coloneqq \neg P_{v,j} \vee \neg P_{v_0,j_0} \vee R_j$ and we derive $\neg P_{v,j} \vee \neg P_{v_0,j_0} \vee R_j$.
3. For each fixed j, we apply Lemma 6 to the clause $C \coloneqq \neg P_{v,j} \vee R_j$ and we derive $\neg P_{v,j} \vee R_j$.

□

Lemma 8. *The formula Stone(G, ρ) has a resolution refutation of width $O(1)$ and size polynomial in N.*

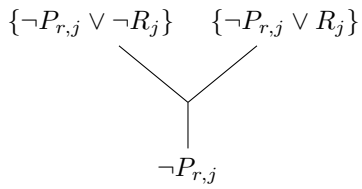
Proof. The refutation proceeds in the following steps.

Elimination of the ρ's : For every induction clause C, we resolve the appended ρ-variable.



Derivation of $S(r)$: For each sink s of G, the clauses $S(s)$ are present in the axioms of Stone(G, ρ). By Lemma 7, we subsequently derive the set $S(r)$ for the root r of G.

Empty clause derivation: For each $1 \leq j \leq N$, we derive $\neg P_{r,j}$.



Now by applying Lemma 6 for C being an empty clause $\perp$, we derive $\perp$, that concludes the proof.

□

Now, from constant-width polynomial-length refutation of Stone(G, ρ) we derive a polynomial-length refutation of the lifted formula Stone$(G, \rho) \circ g$.

Lemma 9. *Let $g : \{0,1\}^b \to \{0,1\}$ be a Boolean function and Φ be a CNF unsatisfiable formula over n variables containing only constant width clauses. Suppose Φ has a resolution refutation of length ℓ and constant width. Then, $\Phi \circ g$ contains clauses of width $O(b)$ and admits a resolution refutation of size $\ell \cdot 2^{O(b)}$.*

Proof. By construction, if C is a clause of width k, then $|C \circ g| \leq 2^{bk}$. If k is constant, this is $2^{O(b)}$. We show that, for every derivation step $(A \vee x), (B \vee \neg x) \to (A \vee B)$ in a proof for F, we can derive all clauses of $(A \vee B) \circ g$ from the clauses of $(A \vee x) \circ g$ and $(B \vee \neg x) \circ g$ in polynomial size, assuming each of A, B has constant width. This follows from the fact that $(A \vee x) \circ g$ and $(B \vee \neg x) \circ g$ semantically imply $(A \vee B) \circ g$: an assignment $(x_{i,1}, \ldots, x_{i,b})_{i \in [M]}$ satisfies formula $C \circ g$ if and only if the assignment $(g(x_{i,1}, \ldots, x_{i,b}))_{i \in [n]}$ satisfies clause C. And since clauses $A \vee x, B \vee \neg x$ semantically imply $A \vee B$, it follows that the formulas $(A \vee x) \circ g$ and $(B \vee \neg x) \circ g$ semantically imply the formula $(A \vee B) \circ g$.

As both A and B are constant-width clauses, each of the formulae $(A \vee x) \circ g$ and $(B \vee \neg x) \circ g$ are defined on at most $O(b)$ variables. By Lemma 1, we can derive each clause in $(A \vee B) \circ g$ from $(A \vee x) \circ g$ and $(B \vee \neg x) \circ g$ in at most $2^{O(b)}$ resolution steps.

Using this fact, we can mimic the resolution refutation of Φ. For each intermediate clause C derived in the resolution refutation for Φ, we can derive all clauses in $C \circ g$. In the end, we derive $\perp \circ g = \{\perp\}$, i.e., the empty clause. Assuming the width of the resolution refutation for Φ is bounded by some constant, the total length of our simulation is at most $\ell \cdot 2^{O(b)}$. □

Now, Theorem 3 is a corollary of Lemma 8, and 9.

7 Lower Bound

In this subsection, we prove the lower bound part of Theorem 1 following the outline given in Section 5.

Theorem 4. *There is an obfuscation map $\rho : [N]^3 \to \mathcal{V}$ such that any bottom-regular ResLin refutation of $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$ must have length at least $2^{\Omega(n^{1/6}/\log n)}$.*

For the rest of this section, we fix $G = (V, E)$ to be the pyramid graph of n levels, and $N = n(n+1)/2$ vertices.

7.1 The Stone Formula is Average-Case Hard for Decision Trees

We identify a distribution μ such that for any obfuscation map $\rho : [N]^3 \to \mathcal{V}$, the search problem *Search*($\mathsf{SP}_{n,\rho}$) is hard on average w.r.t. μ for deterministic decision trees of sufficiently small height, say $n^{1/6}$. First, we fix an arbitrary bijection $f : [N] \to V$ of stones and vertices of the pyramid, the assignments of variables of $\mathsf{SP}_{n,\rho}$ in the support of μ will correspond to placing the stone i on the vertex $f(i)$. The distribution μ sample the assignments as follows.

1. We assign stone i to vertex $f(i)$. Thus for each $v \in V$, $i \in [N]$, and $j \in [N-1]$, we set:

$$P_{v,i} = \begin{cases} 1 & \text{if } f(i) = v \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{v,j} = \begin{cases} 0 & \text{if } j < f^{-1}(v) \\ 1 & \text{otherwise} \end{cases}$$

   Thus, all stone-placement clauses are satisfied.

2. We sample $n-2$ independent uniform bits $B_2, \ldots, B_{n-1} \in \{0,1\}$

3. Let $X_1 = 1$, and for $2 \le j \le n-1$, let $X_j = X_{j-1} + B_j$. We color the vertices (j, X_j) blue for $1 \le j \le n-1$ and other vertices red, i.e., for each stone $i \in [N]$, we set:

$$R_i = \begin{cases} 0 & \text{if } j \le n-1 \text{ and } (j, X_j) = f(i) \\ 1 & \text{otherwise} \end{cases}$$

Let $\alpha \in \mathrm{Supp}(\mu)$. The assignment α corresponds to the following stone placement. It places a different stone on each vertex. There is a path P from the root $r = (1,1)$ to a vertex v in the level $n-1$ given by the random variables $X_1, \ldots, X_{n-1}$, i.e. the vertices of the path are $\{(1, X_1), \ldots, (n-1, X_{n-1})\}$. The stones on the vertices of P are colored blue, all other stones are colored red. We call the path P as the *blue path induced by* α and the vertex v as the *end* of P. Note that the only clause falsified by the assignment α is one of the induction clause for the vertex v and its children u and w. Formally for the stones $i = f^{-1}(u), j = f^{-1}(w)$, and $k = f^{-1}(v)$, the assignment α falsifies exactly one of the following two induction clauses of the vertex v (depending how α sets the value of $\rho(i,j,k)$):

$$D_1(v) := \neg P_{u,i} \vee \neg R_i \vee \neg P_{w,j} \vee \neg R_j \vee \neg P_{v,k} \vee R_k \vee \rho(i,j,k)$$

$$D_0(v) := \neg P_{u,i} \vee \neg R_i \vee \neg P_{w,j} \vee \neg R_j \vee \neg P_{v,k} \vee R_k \vee \neg\rho(i,j,k)$$

The idea of the lower bound for decision trees is that even if the tree query colors of several vertices, it can not identify the end of the path P. Therefore, it can not output the right falsified clause.

First, we prove few lemmas about the random walk $Y_1, \ldots, Y_k$ on a number line starting on $q \in \mathbb{N}$ defined as $Y_1 = q$ and for $i > 2$

$$Y_i = \begin{cases} Y_{i-1} + 1 & \text{with probability } \frac{1}{2} \\ Y_{i-1} & \text{with probability } \frac{1}{2} \end{cases}$$

Note that the random variables $X_1, \ldots, X_{n-1}$ used in the construction of μ are distributed as $Y_1, \ldots, Y_{n-1}$ for $Y_1 = 1$.

Lemma 10. *For any $p \in \{q, \ldots, q+k-1\}$ and $t \in \{2, \ldots, k\}$, we have $\Pr[Y_t = p] \le O(1/\sqrt{t})$.*

Proof. Note that $Y_t = q + \sum_{i=2}^{t} B_i$, where each B_i is an independent uniform random bit. Thus, $Y_t = p = p' + q$ for $p' \in \{0, \ldots, k-1\}$ if and only if $\sum_{i=2}^{t} B_i = p'$.

$$\Pr[Y_t = p] = \Pr\left[\sum_{i=2}^{t} B_i = p'\right] = \binom{t-1}{p'} \cdot 2^{-t+1} \le \binom{t-1}{\lfloor \frac{t-1}{2} \rfloor} \cdot 2^{-t+1} = O\left(\frac{1}{\sqrt{t}}\right)$$

□

Lemma 11. *Let $\mathcal{D}$ be a distribution and let E be an event such that $\Pr[\neg E] \le \varepsilon$. Then, $d_{TV}(\mathcal{D}, \mathcal{D}|E) \le \varepsilon$.*

Proof. We shall construct a coupling (X, Y) of $(\mathcal{D}, \mathcal{D}|E)$ such that $\Pr[X \neq Y] \leq \varepsilon$. This will prove the lemma. To construct the coupling, sample $x \sim \mathcal{D}$. If $x \in E$, output $(X, Y) = (x, x)$. Otherwise, sample $y \sim \mathcal{D}|E$ independently and output $(X, Y) = (x, y)$. It is easy to see that this is a coupling of $(\mathcal{D}, \mathcal{D}|E)$ and $\Pr[X \neq Y] \leq \varepsilon$. □

Now, we show that the walk $Y_1, \ldots, Y_k$ reaches a fixed point with small probability even if the walk avoids a sufficiently small set of forbidden points.

Lemma 12. *Let $S \subseteq \mathbb{N} \times \mathbb{N}$ be a set of forbidden points, $|S| \leq t$, with $k > 2t$. Assume there exists a walk that avoids S – i.e., $Y_i \neq j$ for all $(i, j) \in S$. Let $\mathcal{D}$ be the conditional distribution of the walk $Y_1, \ldots, Y_k$ conditioned on the event that the random walk avoids S. Then, for any $z \in \{q, \ldots, q + k - 1\}$ it holds that*

$$\Pr_{\mathcal{D}}[Y_k = z] \leq O\left(\frac{t^{3/2}}{k^{1/2}}\right).$$

Proof. Let $\mathcal{U}$ be the distribution of the random walk $Y_1, \ldots, Y_k$ not conditioned on anything. There exists an interval $I = [L, R] \subseteq [k]$ with $|I| \geq (k - t)/(k + 1) \geq k/4t$ such that no point of S has the first coordinate in I – i.e., for all $(i, j) \in S : i < L$ or $i > R$.

We show that for all p such that $\Pr_{\mathcal{D}}[Y_L = p] > 0$, we have $\Pr_{\mathcal{D}}[Y_k = z | X_L = p] \leq O\left(t^{3/2}/k\right)$. Note that as we condition on $Y_L = p$, the conditioning on avoiding the points $(i, j) \in S$ with $i < L$ has no effect as $Y_1, \ldots, Y_k$ is a Markov chain. By Lemma 10 we have

$$\Pr_{\mathcal{U}}[Y_k = z | Y_L = p] \leq O\left(\sqrt{\frac{t}{k}}\right).$$

However, we need to bound the probability over the distribution $\mathcal{D}$. By Lemma 10, for all $(i, j) \in S$ with $i > R$ we have

$$\Pr_{\mathcal{U}}[Y_i = z | Y_L = p] \leq O\left(\sqrt{\frac{t}{k}}\right).$$

By union bound,

$$\Pr_{\mathcal{U}}[\exists (i, j) \in S \text{ such that } Y_i = j | Y_L = p] \leq O\left(\frac{t^{3/2}}{k^{1/2}}\right).$$

Let E be the event that $Y_1, \ldots, Y_k$ avoids S conditioned on that $Y_L = p$. We have that $\Pr_{\mathcal{U}}[\neg E] \leq O\left(\frac{t^{3/2}}{k^{1/2}}\right)$. Note that $\mathcal{D} = \mathcal{U}|E$. Thus by Lemma 11 applied on the distribution $\mathcal{D}|Y_L = p$, we have that for any event F

$$\Pr_{\mathcal{D}}[F | Y_L = p] - \Pr_{\mathcal{U}}[F | Y_L = p] \leq O\left(\frac{t^{3/2}}{k^{1/2}}\right)$$

This implies for an event F to be $Y_k = z$ that

$$\Pr_{\mathcal{D}}[Y_k = z | X_L = p] \leq \Pr_{\mathcal{U}}[Y_k = z | Y_L = p] + O\left(\frac{t^{3/2}}{k^{1/2}}\right) \leq O\left(\frac{t^{3/2}}{k^{1/2}}\right).$$

Since this holds for all p such that $\Pr_{\mathcal{D}}[Y_L = p] > 0$, we conclude that

$$\Pr_{\mathcal{D}}[Y_k = z] \leq O\left(\frac{t^{3/2}}{k^{1/2}}\right).$$

□

Now, we show that any deterministic decision tree of small height solving the *Search*($\mathsf{SP}_{n,\rho}$) problem makes on average significant errors when inputs to the problem are sampled according to the distribution μ described before. We say a decision tree queries the color of a vertex v of G if it queries the variable $R_{f^{-1}(v)}$. Recall that the stone $f^{-1}(v)$ is placed on the vertex v by assignments in $\mathrm{Supp}(\mu)$.

Note that there is a simple, even non-adaptive, decision tree of height $O(\sqrt{n})$ that makes few errors. It simply queries the colours of $O(\sqrt{n})$ nodes of the pyramid graph, centered around the $(n-1)/2$-th node at level $n - 1$. With very high probability, there is a blue node among the queried ones which uniquely identifies the falsified induction clause. Nevertheless, we will show that all decision trees of height at most $n^{1/6}$, will make errors with large probability to identify a falsified clause.

Consider a decision tree $\mathcal{T}$ for *Search*($\mathsf{SP}_{n,\rho}$). We transform $\mathcal{T}$ into a decision tree $\mathcal{T}'$ in a canonical form:

- Initially, $\mathcal{T}'$ always queries the color of the root r of G.
- Suppose $\mathcal{T}$ outputs an induction clause $D_0(v)$ or $D_1(v)$ for a vertex v of G. Then, $\mathcal{T}'$ queries the color of the vertex v first. If the color of v is red (i.e., $R_{f^{-1}(v)} = 1$), then $\mathcal{T}'$ outputs an error symbol. Note that in that case both clauses $D_0(v)$ and $D_1(v)$ are satisfied. Otherwise it outputs the same induction clause that $\mathcal{T}$ outputs.
- If $\mathcal{T}$ outputs any other clause, then $\mathcal{T}'$ outputs an error symbol.

We remark this modification increases the height of the tree by at most two. A decision tree $\mathcal{T}'$ in a canonical form can output either an induction clauses from $\{D_0(v), D_1(v) \mid v \in V(G)\}$ or an error symbol. Moreover, the probability of $\mathcal{T}$ making an error is precisely the probability of reaching a leaf node of $\mathcal{T}'$ labeled with an error symbol.

Note that the assignment to $P_{v,i}$ and $Z_{v,j}$ are fixed by any assignment in $\mathrm{Supp}(\mu)$, i.e., for each vertex the stone placed on it is determined. Thus, we can assume WLOG the decision tree only queries the variables R_j.

Note that for each cube $C \subseteq \{0,1\}^m$ there is a corresponding partial assignment $\alpha_C \in \{0,1,*\}^m$ such that the cube C is exactly the set of total extension of α_C, i.e., $C = \{\alpha \in \{0,1\}^m \mid \alpha \text{ extends } \alpha_C\}$. We say a cube $C \subseteq \{0,1\}^m$ fixes a vertex $v \in V(G)$ to red (or blue) if the corresponding partial assignment α_C assigns a value 1 (or 0) to the variable $R_{f^{-1}(v)}$.

Now, fix a decision tree $\mathcal{T}$ for *Search*($\mathsf{SP}_{n,\rho}$) in a canonical form and let h be the height of $\mathcal{T}$. We say a cube $C \in \{0,1\}^m$ is *useful* if it fixes some vertex in a level $\ell > \frac{n}{2h}$ to blue and for all levels ℓ' such that $\ell - \frac{n}{2h} \le \ell' < \ell$, the cube C does not fix any vertex to blue in the level ℓ'. A node p of $\mathcal{T}$ is *useful* if the cube C_p associated with p is useful. Clearly, the root of $\mathcal{T}$ is not useful.

Lemma 13. *If a leaf p of $\mathcal{T}$ does not outputs an error symbol, then the leaf p is useful.*

Proof. Let p outputs one of the induction clauses $D_0(v)$ or $D_1(v)$ for $v \in V(G)$ as falsified. Since $\mathcal{T}$ is in the canonical form, the cube C_p fixes the vertex v to blue. Recall that the vertex v is in the level $n-1$. Let $1 = \ell_1 < \cdots < \ell_r = n-1$ be the levels where C_p fixes some vertices to blue. Since $r \le h$, there must exist an i such that $\ell_{i+1} - \ell_i \ge (n-1)/h \ge (n-2)/h$. Thus, there is no vertex fixed to blue on levels $\ell_i + 1, \ldots, \ell_{i+1} - 1$, i.e., on all $\ell_{i+1} - \ell_i - 1 \ge n/(2h)$ levels above the level ℓ_{i+1}. We conclude a node w fixed to blue on the level ℓ_{i+1} makes the leaf useful. □

Lemma 14. $\Pr_{\alpha\sim\mu}[\textit{Computation path of } \alpha \textit{ reaches a useful node}] \le O(h^3/\sqrt{n})$.

Proof. Let $\mathcal{T}(\alpha, k)$ denote the node of $\mathcal{T}$ reached by α after k queries. For each $1 \le k \le h$, we upper bound the probability that the computation path of α reaches a useful node for the first time at step k. Then, we shall use union bound on k. Formally, we bound the probability as follows.

$$\begin{aligned}
&\Pr_{\alpha\sim\mu}[\text{computation path of } \alpha \text{ reaches a useful node}] \\
&\quad= \Pr_{\alpha\sim\mu}\big[\exists k \in [h] : \mathcal{T}(\alpha,k) \text{ is useful and } \mathcal{T}(\alpha,k-1) \text{ is not useful}\big] \\
&\quad\le \sum_{k=1}^{h} \Pr_{\alpha\sim\mu}\big[\mathcal{T}(\alpha,k) \text{ is useful and } \mathcal{T}(\alpha,k-1) \text{ is not useful}\big] \\
&\quad\le h \cdot \max_{k\in[h]} \Pr_{\alpha\sim\mu}\big[\mathcal{T}(\alpha,k) \text{ is useful} \mid \mathcal{T}(\alpha,k-1) \text{ is not useful}\big]
\end{aligned}$$

We bound the last probability for any $k \in [h]$. Let $p = \mathcal{T}(\alpha, k-1)$. We assume the node p is not useful. Let $(i,j) \in V(G)$ be the lowest vertex that is fixed by C_p to blue. Suppose that in the the next step $\mathcal{T}$ queries a color of the vertex (i',j'). If the next node has to be useful, we need $i' > i + n/(2h)$ and the color of (i',j') has to be blue. The probability that the response to the query is blue is

$$\Pr_{\alpha\sim\mu|C_p}[\text{The blue path induced by } \alpha \text{ visits } (i',j')]$$

Now, consider the random walk $X_1, \ldots, X_{n-1}$ that determine the blue path P induced by $\alpha \sim \mu$. Recall that vertices of P are $\{(1,X_1), \ldots, (n-1, X_{n-1})\}$. The cube C_p fixes colors of some vertices of G, let B (or R) be a set of vertices fixed to blue (or red) by C_p. Thus, the conditioning on the cube C_p restricts the random walk $X_1, \ldots, X_{n-1}$ that it must visit the blue points in B and must not visit the red points in R. Formally, for any $(q,y) \in B$ it holds that $X_q = y$ and for any $(q',y') \in R$ it holds that $X_{q'} \neq y'$.

We know there is at least one walk that avoids R (the walk given by α).
Since $|R| \leq h$, we have the following bound.

$$\begin{aligned}
&\Pr_{\alpha\sim\mu|C_p}\left[\text{The blue path induced by } \alpha \text{ visits } (i',j')\right] \\
&\quad = \Pr_{\mu|C_p}\left[X_{i'} = j'\right] = O\left(\frac{h^{3/2}}{\sqrt{n/h}}\right) \qquad \text{(applying appropriate time shift, by Lemma 12)} \\
&\quad = O\left(\frac{h^2}{\sqrt{n}}\right)
\end{aligned}$$

By union bound over $k \in [h]$, we have the final bound, the probability that the computation path ever reaches a useful node is at most $O(h^3/\sqrt{n})$.

□

We end this subsection with a proof that the formula $\mathsf{SP}_{n,\rho}$ is average-case hard for decision trees.

Lemma 15. *For any $\varepsilon > 0$, there exists $\gamma > 0$ such that every deterministic decision tree of height at most $\gamma \cdot n^{1/6}$ for Search($\mathsf{SP}_{n,\rho}$) makes error with probability $\geq 1 - \varepsilon$ w.r.t. the distribution μ.*

Proof. If the decision tree answers correctly, by Lemma 13 it must reach a useful node at some point. By Lemma 14, the probability of this ever happening is at most $c \cdot \frac{h^3}{\sqrt{n}}$ for some constant $c > 0$. Thus, we can pick $\gamma > 0$ small enough such that for $h = \gamma \cdot n^{1/6}$, it holds that $c \cdot \frac{h^3}{\sqrt{n}} < \varepsilon$. □

7.2 Lifting the Average-Case Hardness to Parity Decision Trees

We lift the distribution μ to a distribution μ' of variables of $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$ as follows:

1. Sample an assignment α according to μ.
2. Sample a uniformly random assignment from $\overrightarrow{\mathsf{IP}}^{-1}(\alpha)$.

We remark that an assignment β sampled by μ' falsifies exactly one clause of $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$, in particular one clause that arises by a lifting clause C of $\mathsf{SP}_{n,\rho}$ where C is the unique clause falsified by the assignment $\overrightarrow{\mathsf{IP}}(\beta)$.

In this section, we prove *Search*($\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$) is average-case hard for parity decision trees of small height, under the lifted distribution. To do so, we shall use a result of Chattopadhyay et al. [3], that built upon the earlier work of Göös, Pitassi and Watson [7].

We will need to consider randomized decision trees that output Boolean strings in $\{0,1\}^t$, rather than 0/1. For a given deterministic 2-party communication protocol Π, let $\Pi(x,y)$ denote the transcript generated by Π on input (x,y).

Theorem 5 (Implicit in [3])**.** *Assume $b \geq 50\log(m)$. Let Π be any deterministic 2-party communication protocol of cost c, where Alice and Bob each get inputs from $\{0,1\}^{mb}$. Then, there exists a randomized decision tree $\mathcal{T}$ of cost $O(c/b)$ such that the following holds for every $x, y \in \{0,1\}^m$:*

$$d_{TV}\big(\mathcal{T}(x,y), \Pi(\overrightarrow{\mathsf{IP}}^{-1}(x), \overrightarrow{\mathsf{IP}}^{-1}(y))\big) \leq 1/10.$$

The above theorem says that a randomized decision tree is able to simulate by probing only a few bits of its input (x,y) the transcript of a deterministic communication protocol when it is given a random input from $\big(\overrightarrow{\mathsf{IP}}^{-1}(x), \overrightarrow{\mathsf{IP}}^{-1}(y)\big)$. Its relevance for us is due to the following simple observation.

Observation 3. *Every deterministic parity decision tree of height h can be simulated exactly by a deterministic 2-party communication protocol of cost at most $2h$.*

Now, we can lift our avarage-case hardness to parity decision trees.

Lemma 16. *There exists a constant $c > 0$ such that for every obfuscation map ρ and every parity decision tree $\mathcal{T}$ of height at most $c \cdot n^{1/6}\log n$ purporting to solve Search($\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$), the following is true:*

$$\Pr_{\beta\sim\overrightarrow{\mathsf{IP}}^{-1}(\mu)}\left[\mathcal{T}(\beta) \textit{is falsified on } \beta\right] \leq \frac{2}{5}.$$

Proof. Assume $\mathcal{T}$ makes an error with probability $< 3/5$. Then our main idea is that we would be able to construct an ordinary decision tree for $Search(\mathsf{SP}_{n,\rho})$ of depth $O(n^{1/6})$ which makes error with probability $< 7/10$ under distribution μ. This contradicts Lemma 15.

Using Observation 3, we get a deterministic 2-party protocol of cost at most $2 \cdot \text{depth}(\mathcal{T})$ that makes error less than $3/5$ for solving $Search\big(\mathsf{SP}_{n,\rho} \circ \mathsf{IP}\big)$. Theorem 5 then yields a randomized decision tree $\mathcal{T}'$ with the following properties On input $z \sim \mu$,

- $\mathcal{T}'$ makes at most $O\left(\text{cost}(\Pi)/\log n\right)$ queries to z, i.e., at most $O(n^{1/6})$.
- If $\mathcal{D}_1$ denotes the actual distribution of the transcript of Π when its run on input sampled uniformly at random from $\overrightarrow{\mathsf{IP}}^{-1}(z)$ and $\mathcal{D}_2$ denotes the distribution of the transcript of Π simulated by $\mathcal{T}'$,

$$||\mathcal{D}_1 - \mathcal{D}_2|| \leq \frac{1}{10}$$

We now modify $\mathcal{T}'$ to output a clause as follows. A transcript of Π leads it to output a clause of $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$. The modified $\mathcal{T}'$ outputs the unique corresponding un-lifted clause of $\mathsf{SP}_{n,\rho}$. The probability of error is at most $\Pr[\Pi \text{ errs }] + 1/10 < 7/10$. This gives a randomized decision tree; by fixing the coins we can replace it by a deterministic decision tree, contradicting Lemma 15. □

7.3 Foolable Nodes Are Frequent

Let $\mathcal{P}$ be a bottom-read-once linear branching program for $Search\big(\mathsf{SP}_{n,\rho} \circ \mathsf{IP}\big)$ that corresponds to a bottom-regular ResLin proof of the unsatisfiability of $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$. Our goal is to show size of $\mathcal{P}$ is large. To do so, we establish that the affine spaces associated with many nodes of $\mathcal{P}$ have a certain property that allows to fool them. In particular, let β be an assignment sampled by μ'. We will prove that with high probability after making $t = O(n^{1/6})$ according to β, we will end in a node v of $\mathcal{P}$ such that the associated affine space A_v does not have much information about β. We will show that this implies the affine space A_v contains many useful assignments that allows us to prove the co-dimension A_v is large. Now, we define the sought property formally.

Definition 3. *Let $\alpha \in Supp(\mu)$ and P be the blue path induced by α with the end v. Let u and w be the two children of v. We say a restriction $\alpha_{|T}$ for $T \subseteq [m]$ is* fooling *if $\alpha_{|T}$ does not assign value to any variable mentioning v, u, or w, i.e. the variables $P_{x,i}$ and $Z_{x,j}$ for $x \in \{v, u, w\}, i \in [N]$, and $j \in [N-1]$.*

Definition 4. *An affine space $A \subseteq \mathbb{F}^{mb}$ is* foolable *if there is a total assignment $\beta \in A$ such that the partial assignment $\overrightarrow{\mathsf{IP}}\big(\beta_{|\,VarCl(A)}\big) \in \{0, 1, *\}^m$, is fooling.*

We remark that we implicitly suppose that the partial assignment $\overrightarrow{\mathsf{IP}}\big(\beta_{|\text{VarCl}(A)}\big)$ is a restriction of some total assignment $\alpha \in \text{Supp}(\mu)$ as we defined fooling assignments only for restrictions of assignments from $\text{Supp}(\mu)$.

We call a node v of $\mathcal{P}$ *foolable*, if the associated affine space A_v is foolable. Recall that $\mathcal{P}(\beta, t)$ is the node that $\mathcal{P}$ arrives at after making t linear queries on β. It turns out the node $\mathcal{P}(\beta, t)$ is foolable with high probability if t is sufficiently small, as stated in the following theorem.

Theorem 6. *Let $\mathcal{P}$ be any bottom-read-once linear branching program corresponding to a bottom-regular ResLin proof of $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$. There exists a constant $c > 0$, such that if $t < c \cdot n^{1/6}$, then*

$$\Pr_{\beta \sim \overrightarrow{IP}^{-1}(\mu)} \Big[\mathcal{P}(\beta, t) \textit{ is foolable}\Big] > \frac{3}{5}.$$

Proof. Let v denote the random node $\mathcal{P}(\beta, t)$. We will show that whp $\overrightarrow{\mathsf{IP}}\big(\beta_{|\,VarCl(A_v)}\big)$ is fooling. In order to so, we will first construct a PDT $\mathcal{T}$ of depth $O(n^{1/6} \log n)$ from $\mathcal{P}$ in the following manner: on input β, it will simulate the path traced out in $\mathcal{P}$ for t steps by making precisely those linear queries that would have been issued in $\mathcal{P}$. At the end of it, $\mathcal{T}$ does the following: Let A be the affine space corresponding to the queries issued and responses received so far. For every vertex $k = (i, j)$ in the pyramid graph G_n such that one of its variables (either $P_{k,\ell}$ or $Z_{k,\ell'}$ for some $\ell \in [N]$, or $\ell' \in [N]$) is in $\text{Cl}(A)$, query the b coordinates from the blocks of the following set of variables:

- $\big\{R_{f^{-1}(w)} | w \in \{(i-1, j-1), (i-1, j), (i, j-1), (i, j), (i, j+1), (i+1, j), (i+1, j+1)\} \cap V(G)\big\}$

If the partial assignment consistent with β that is recovered by $\mathcal{T}$ falsifies any clause of $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$ then, output that clause. Otherwise, if $\mathcal{P}$ outputs any falsified clause then output the same. If neither of the previous case occurs, output error.

Clearly, the depth of $\mathcal{T}$ is $O(n^{1/6} \log n)$. Thus, by Lemma 16, the probability that it outputs a falsified clause is at most $2/5$. Let A_v denote the affine space at $\mathcal{P}(\beta, t)$. Note that $A \subseteq A_v \implies \mathrm{Cl}(A_v) \subseteq \mathrm{Cl}(A)$ (Lemma 4). It is straight-forward to verify that if $\overrightarrow{\mathsf{IP}}\big(\beta_{|VarCl(A_v)}\big)$ is not fooling, then $\mathcal{T}$ successfully outputs a clause falsified by β: suppose v is the endpoint of the path sampled by μ with children u, w. If one of the variables belonging to u, v or w is in $\mathrm{Cl}(A)$, in the final step the PDT queries the stones placed on u, v, w and detects that an induction clause at u is falsified. The result now follows from Lemma 16. □

7.4 Foolability Implies Large Rank

In this subsection, we prove there is an obfuscation map $\rho : [N]^3 \to \mathcal{V}$ such that for a foolable node v of a bottom-read-once linear branching program $\mathcal{P}$ computing $\mathit{Search}(\mathsf{SP}_{n,\rho} \circ \mathsf{IP})$, the associated affine space A_v must have large co-dimension.

First, we prove an auxiliary lemma. Let $\alpha \in \{0, 1, *\}^m$ be a partial assignment for variables of $\mathsf{SP}_{n,\rho}$. We say a stone j is *marked* by α if α assigned a value to a variable that mentions the stone j, i.e. $R_j, P_{v,j}$ for any vertex $v \in V$, $P_{f(j),k}$ for any stone $k \in [N]$, or $Z_{f(j),\ell}$ for any $\ell \in [N-1]$. Let $Q(\alpha) \subseteq [N]$ be a set of stones marked by α. We can use the non-marked stone to extend a fooling restriction $\alpha' \in \{0, 1, *\}^m$ of an assignment $\alpha \in \mathrm{Supp}(\mu)$ to a total assignment α' where we change the unique unsatisfied clause as stated in the following lemma.

Lemma 17. *Let ρ be any obfuscation map and $\alpha \in \mathit{Supp}(\mu)$, with v being the vertex at which the blue path of α ends. Further, let $\alpha' := \alpha_{|T} \in \{0, 1, *\}^m$ be a fooling restriction of α for $T \subseteq [m]$ such that $|Q(\alpha')| \leq N - 3$. Then, for any $i < j < k \in [N] \setminus Q(\alpha')$, the partial assignment α' can be extended to a total assignment $\alpha_1 \in \{0, 1\}^m$ that satisfies all clauses of $\mathsf{SP}_{n,\rho}$ except exactly one of the following two induction clauses:*

$$C_1 := \neg P_{u,i} \vee \neg R_i \vee \neg P_{w,j} \vee \neg R_j \vee \neg P_{v,k} \vee R_k \vee \rho(i, j, k), \text{ or}$$
$$C_2 := \neg P_{u,i} \vee \neg R_i \vee \neg P_{w,j} \vee \neg R_j \vee \neg P_{v,k} \vee R_k \vee \neg\rho(i, j, k),$$

where u and w are the out-neighbors of v.

Proof. We will extend α' almost to α with the following changes. The red stones i and j will be placed on the vertices u and w, and the blue stone k will be placed on the vertex v. The stone placed on the vertices $f(i)$, $f(j)$, and $f(k)$ will be some different stones of appropriate colors.

Since α' is fooling, the variables mentioning v and its out-neighbors u and w are not assigned by α'. Thus, α' does not falsify any clauses of $\mathsf{SP}_{n,\rho}$. Moreover, the stones i, j, k are not marked by α', i.e., the variables mentioning these stones are not assigned by α' as well. Thus, we place the stone i on u, the stones j on w, and the stones k on v. We set the color of the stone i and j to red, and the color of the stone k to blue. Formally, we set the variables as follows, $P_{u,i} = 1, P_{w,j} = 1$, and $P_{v,k} = 1$. For $x \in \{u, w, v\}$ and $\ell \notin \{i, j, k\}$, we set variables $P_{x,\ell}$ to 0. We set $R_i = R_j = 1$, and $R_k = 0$. We set the auxiliary variables $Z_{x,\ell}$ for $x \in \{u, w, v\}$ and $\ell \in [N-1]$ appropriately so that no stone-placement clause for u, w, and v is falsified. This is a proper extension of α' (we assign a value to the variables that are not assigned by α'). Moreover, the only unsatisfied clause by our extension is one of the two induction clauses C_1 or C_2.

It remains to finish our extension to get a total assignment α_1 that does not falsify any other clause. For vertices other than u, w, v and $f(i), f(j), f(k)$, we place stones on them according to α, i.e., we place a stone ℓ to the vertex $f(\ell)$ and color it blue if $f(\ell)$ is on the path P and red otherwise. Since α' is a restriction of α, we can extend α' to represent the placement and coloring of stones described above.

Since the stones i, j, and k are not marked by α', the variables $P_{x,\ell}$ and $Z_{x,\ell'}$ for $x \in \{f(i), f(j), f(k)\}$, any stone $\ell \in [N]$, and any index $\ell' \in [N-1]$ are still not assigned. Thus, we can place any stone on the vertices $f(i), f(j)$, and $f(k)$. For a vertex $x \in \{f(i), f(j), f(k)\}$, we place any blue stone if it is on the path P and otherwise, any red stone.

The stones that were placed on the vertices u, w, and v by α are not placed on any vertex by α_1. Thus, if their color is not given by the restriction α', we can set their color arbitrarily.

□

We remark that a stone may be placed in α_1 on more than one vertex. But this does not falsify any clause of $\mathsf{SP}_{n,\rho}$ because the stone-placement clauses merely enforce that there is a stone placed on each vertex of G.

For our hard formula, we use an appropriate obfuscation map ρ given by the following lemma.

Lemma 18 (Lemma 3.9, Alekhnovich et al.[3] [1])**.** *For N sufficiently large, there exists a mapping $\rho : [N]^3 \to \mathcal{V}$ such that for every Q and $X \in \mathcal{V}$, where $Q \subseteq [N]$ of size at most $\frac{N}{400}$ there exist $i < j < k \in [N] \setminus Q$ such that $\rho(i,j,k) = X$.*

Lemma 19. *There exists an obfuscation map $\rho : [N]^3 \to \mathcal{V}$ such that the following holds. Let p be a foolable node in a bottom-read-once branching program $\mathcal{P}$ computing Search($\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$). If there is a path of length t from the source to p, then co-dim$(A_p) \geq \min\{\frac{N}{800}, t\}$.*

Proof. Fix $\rho : [N]^3 \to \mathcal{V}$ to be a map with the property guaranteed by Lemma 18. Let $A_p = \mathcal{S}(M, c)$ and suppose $\mathrm{rank}(M) \leq \frac{N}{800}$. Let S be all sinks of $\mathcal{P}$ reachable from p. First, we show that for each variable Y of $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$ there is a sink in S outputting a clause D such that Y is in D. Let X be a variable of $\mathsf{SP}_{n,\rho}$ such that Y is in the block of X.

Let $\alpha' := \overrightarrow{\mathsf{IP}}(\beta_{|\mathrm{VarCl}(A_p)})$ be a fooling partial assignment given by the assumption that A_p is foolable. Since $\mathrm{rank}(M) \leq \frac{N}{800}$ and consequently, $|\mathrm{Cl}(A_p)| \leq \frac{N}{800}$, the assignment α' assigns a value to at most $\frac{N}{800}$ variables of $\mathsf{SP}_{n,\rho}$. Assigning a value to a variable $P_{v',j}$ for a vertex $v' \in V(G)$ causes marking two stones, in particular, stones $f^{-1}(v')$ and j. Assigning a value to a variable R_j (or $Z_{v',\ell}$) for a stone j (or a vertex $v' \in V(G)$) causes marking the stone j (or $f^{-1}(v')$, respectively). Thus, the partial assignment α' marks at most $\frac{N}{400}$ stones.

Now, we apply the property of chosen ρ given by Lemma 18 for the variable X and the set $Q := Q(\alpha')$. Thus, we have stones $i < j < k \in [N] \setminus Q(\alpha')$ such that $\rho(i,j,k) = X$. By Lemma 17, we extend α' to a total assignment $\alpha_1 \in \{0,1\}^m$ such that the only falsified clause of $\mathsf{SP}_{n,\rho}$ by α_1 is one of the following two:

$$C_1 := \neg P_{u,i} \vee \neg R_i \vee \neg P_{w,j} \vee \neg R_j \vee \neg P_{v,k} \vee R_k \vee X, \text{or}$$
$$C_2 := \neg P_{u,i} \vee \neg R_i \vee \neg P_{w,j} \vee \neg R_j \vee \neg P_{v,k} \vee R_k \vee \neg X.$$

Without loss of generality, suppose α_1 falsifies only the clause C_1.

Since α' is an image of a closure assignment and α_1 extends α', there is an assignment $\beta_1 \in A_p$ such that $\overrightarrow{\mathsf{IP}}(\beta_1) = \alpha_1$ by Lemma 5. By Observation 1, the assignment β_1 falsifies exactly one clause D of $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$. The clause D arises as one of the clauses in the family obtained from lifting clause C_1. Further, the clause D contains all variables from blocks corresponding to the variables of C_1. Thus, the variable Y is in D as the variable Y is in the block of the variable X. Since $\beta_1 \in A_p$, and D is the only clause falsified by β_1, there has to be a sink in S that is labeled by D.

Now, let U be a space spanned by rows of matrices defining the spaces of sinks in S. Formally, let $s \in S$ is labeled by an affine space $A_s = \mathcal{S}(M_s, c_s)$ (that corresponds to a clause of $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$). Then, $U = \mathrm{Span}\big(\{\mathcal{R}(M_s) \mid s \in S\}\big)$. Since each variable of $\mathsf{SP}_{n,\rho} \circ \mathsf{IP}$ is mentioned at some sink in S, the space U has full dimension, i.e., $\dim(U) = mb$.

Let $W = \mathrm{Span}\big(\mathcal{R}(M) \cup \mathit{Post}(p)\big)$. By Lemma 2, we have

$$\dim(W) \leq \mathrm{rank}(M) + \dim(\mathit{Post}(p)) \leq \mathrm{rank}(M) + mb - t.$$

On the other hand, $U \subseteq W$ and thus, $\dim(W) \geq \dim(U) = mb$. Therefore by putting both inequalities together, we get $\mathrm{rank}(M) \geq t$.

□

7.5 Lifted Distributions Fool Rank

In the earlier two subsections, we have established the following two facts: (i)in any ROLBP $\mathcal{P}$ corresponding to a bottom-regular ResLin proof of our lifted stone formula, when inputs β are sampled according to distributions that are IP lifts, the node $\mathcal{P}(\beta, t)$ is with large probability a foolable node; (ii) in such a ROLBP, the constraint matrix for the affine space associated with a foolable node has large

[3] The set of clauses used by Alekhnovich et al. is over slightly smaller set of variables. Thus their statement of the lemma has larger bound for Q ($|Q| \leq \frac{N}{100}$). However, the number of variables in our case is still quadratic in N (in particular $2N^2$) as in the result of Alekhnovich et al. Thus, proof of both lemmas is basically the same – by counting argument.

rank. To prove that $\mathcal{P}$ has large size, it is sufficient to argue that each large rank constraint system is satisfiable with small probability. Of course, such a statement is well known to be true if we sample inputs from the uniform distribution in $\mathbb{F}_2^{bm}$. However, our distribution is not so at all. In particular, it has quite sparse support. Still, it turns out that any lifted distribution is pseudo-random with respect to the rank measure if the gadget satisfies the following property: consider a gadget $g : \{0,1\}^b \to \{0,1\}$. For any $S \subset [b]$, and $o \in \{0,1\}$ an assignment α to the bits outside of S is called o-stifling if g gets fixed to o by α, i.e., the induced subfunction $g_{|\overline{S}\leftarrow\alpha}$ gets fixed to the constant function that always evaluates to o, no matter how the bits in S are set. We say g is ϵ-balanced, k-stifling if for any subset $S \subseteq [b]$ of size at most k, and for any $o \in \{0,1\}$, the following is true: when we sample $x \in \{0,1\}^b$ uniformly at random from $g^{-1}(o)$, then the projection of x on co-ordinates outside of S are o-stifling with probability at least ϵ.

Claim 1. *Inner-product defined on $2b \geq 8$ bits is $7/18$-balanced and 1-stifling gadget.*

Proof. By simple manipulation,

$$\Pr_{(x,y)\sim\{0,1\}^{2b}}\big[\mathsf{IP}(x,y) = 0 \wedge x_1 = 0\big] = \mathbb{E}\left[\left(\frac{1+(-1)^{\sum_{i=1}^b x_i y_1}}{2}\right)\left(\frac{1+(-1)^{x_1}}{2}\right)\right].$$

The RHS becomes,

$$\frac{1}{4} + \frac{1}{4}\mathbb{E}\big[(-1)^{x_1}\big] + \frac{1}{2}\mathbb{E}\big[(-1)^{\sum_{i=1}^b x_i y_i}\big]$$

The second sum is 0, and the third is at most $\frac{1}{2^{b+1}}$. Overall, this gives that

$$\Pr_{(x,y)\sim\{0,1\}^{2b}}\big[\mathsf{IP}(x,y) = 0 \wedge x_1 = 0\big] \geq \frac{1}{4} - \frac{1}{2^{b+1}}.$$

Further, by a similar method,

$$\Pr_{(x,y)\sim\{0,1\}^{2b}}\big[\mathsf{IP}(x,y) = 0\big] \leq \frac{1}{2} + \frac{1}{2^b}.$$

Thus,

$$\Pr_{(x,y)\sim\{0,1\}^{2b}}\big[x_1 = 0 \,|\, \mathsf{IP}(x,y) = 0\big] \geq \frac{1/4 - 1/2^{b+1}}{1/2 + 1/2^b}.$$

Noting that any assignment that sets $x_1 = 0$ stifles y_1. Hence, y_1 is stifled with probability at least $7/18$, if $b \geq 4$, by the projection of a random 0 (and similarly 1) assignment to IP. Completely analogously, any bit is stifled with the same probability. □

Remark 7.1. *It is worth noting that several gadgets, including Inner-Product, Indexing, Majority, even on sufficiently large but constant number of bits, are stifling and balanced.*

Now we state the utility of balanced, stifling gadgets.

Lemma 20. *Let g be any ϵ-balanced, 1-stifling gadget and $z \in \mathbb{F}_2^m$ be any fixed vector. Then, for every matrix $M \in \mathbb{F}_2^{r\times bm}$ of full rank r, and vector $\gamma \in \mathbb{F}_2^r$ the following holds:*

$$\Pr_{\beta\sim\overrightarrow{g}^{-1}(z)}\Big[M\beta = \gamma\Big] = 2^{-\Omega_\epsilon(r/b)}.$$

Proof. After Gaussian elimination, turning M into row-echelon form, there are at least r/b different blocks in which pivots of rows appear. Let us call each such block a pivot block. The distribution $\overrightarrow{g}^{-1}(z)$ samples independently at random from $g^{-1}(z_i)$ for each of the i-th block. It will be convenient to think that we sample, one after the other, independently from blocks in this way, starting from the rightmost. Consider the situation when we arrive at a pivot block having sampled all blocks to its right. Let the bit of z corresponding to this pivot block be $o \in \{0,1\}$. Consider any one row that has a pivot in that block. Let the equation corresponding to this row be denoted by ℓ. By the property of g, with probability at least ϵ the random assignment from $g^{-1}(o)$ will stifle the pivot of ℓ. Conditioned on that event, the stifled bit will be set to each of $0, 1$ with probability exactly $1/2$ as each possible setting gives rise to a distinct assignment in $g^{-1}(o)$. Hence, the probability that ℓ is satisfied by the sampled assignment to this block is at most $(1 - \epsilon/2)$. Thus, continuing this way, the probability that all the equations are satisfied is at most $\big(1 - \epsilon/2\big)^{r/b}$, yielding the desired result. □

7.6 Putting Everything Together

Now, we are ready to finish the proof of our lower bound, i.e., Theorem 4.

Proof of Theorem 4. Let $\mathcal{P}$ be the ROLBP derived from a bottom-regular ResLin proof of Stone$(G_n, \rho) \circ$ IP. Let $t = \lfloor c \cdot n^{1/6} \rfloor$ for an appropriately chosen small constant $c > 0$. Combining Theorem 6 and Lemma 19, we get

$$\Pr_{\beta \sim \overrightarrow{\mathsf{IP}}^{-1}(\mu)} \left[A_{\mathcal{P}(\beta,t)} \text{ has co-dimension } \geq t \right] \geq \frac{3}{5}. \tag{7.1}$$

On the other hand, for any node v of $\mathcal{P}$ which has co-dim$(A_v) \geq t$, Lemma 20 yields,

$$\Pr_{\beta \sim \overrightarrow{\mathsf{IP}}^{-1}(\mu)} \left[\mathcal{P}(\beta, t) \text{ is } v \right] \leq 2^{-\Omega\left(\frac{t}{\log n}\right)}. \tag{7.2}$$

If s is the total number of nodes of $\mathcal{P}$, combining (7.1) and (7.2), we get immediately

$$s \cdot 2^{-\Omega\left(t/\log(n)\right)} \geq \frac{3}{5}.$$

Substituting the value of t in the above, the result immediately follows.

□

References

[1] Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart. An exponential separation between regular and general resolution. *Theory of Computing*, 3(5):81–102, 2007. Preliminary version in STOC, 2002.

[2] Paul Beame and Sajin Koroth. On disperser/lifting properties of the index and inner-product functions. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[3] Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting using low-discrepancy gadgets. *SIAM J. Comput.*, 50(1):171–210, 2021. Preliminary version in ICALP, 2019.

[4] 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.

[5] Klim Efremenko, Michal Garlík, and Dmitry Itsykson. Lower bounds for regular resolution over parities. *Electron. Colloquium Comput. Complex.*, TR23-187, 2023.

[6] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Theory Comput.*, 16:1–30, 2020. Preliminary version in STOC 2018.

[7] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. *SIAM J. Comput.*, 49(4), 2020. Preliminary version in FOCS 2017.

[8] 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.

[9] 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.

[10] Pavel Pudlák. On extracting computations from propositional proofs (a survey). In Kamal Lodaya and Meena Mahajan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*, volume 8 of *LIPIcs*, pages 30–41. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010.

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

[12] Alexander A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded-arithmetic. *Izvestiya. Math.*, 59(1):205–227, 1995.

[13] Dmitry Sokolov. Dag-like communication and its applications. In Pascal Weil, editor, *Computer Science - Theory and Applications - 12th International Computer Science Symposium in Russia, CSR 2017, Kazan, Russia, June 8-12, 2017, Proceedings*, volume 10304 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2017.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il