Electronic Colloquium on Computational Complexity, Report No. 203 (2024)

Direct Sums for Parity Decision Trees

Tyler Besselman
NYU Shangai

Mika Göös
EPFL

Siyao Guo
NYU Shangai

Gilbert Maystre
EPFL

Weiqiang Yuan
EPFL

December 9, 2024

Abstract

Direct sum theorems state that the cost of solving k instances of a problem is at least $\Omega(k)$ times the cost of solving a single instance. We prove the first such results in the randomised parity decision tree model. We show that a direct sum theorem holds whenever (1) the lower bound for parity decision trees is proved using the *discrepancy method*; or (2) the lower bound is proved relative to a *product distribution*.

Contents

ISSN 1433-8092

1 Introduction

One of the most basic questions that can be asked for any model of computation is:

How does the cost of computing k independent instances scale with k?

A *direct sum* theorem states that if the cost of solving a single copy is C, then solving k copies has cost at least $\Omega(k \cdot C)$, which matches the trivial algorithm that solves the k copies separately. Direct sums have been studied exhaustively for randomised query complexity R^{dt}, randomised communication complexity R^{cc}, and other concrete models of computation; see Section 1.3 for prior work. In this work, we initiate the study of direct sum problems for randomised parity decision tree complexity R^{pt}, a computational model sandwiched between the widely-studied R^{dt} and R^{cc}.

Parity decision trees. Parity decision trees generalise the usual notion of decision trees by allowing *parity queries*. To compute a function $f\colon \{0,1\}^n \to \{0,1\}$ on input $x \in \{0,1\}^n$, a deterministic parity decision tree T performs queries of the form *"what is $\langle a, x\rangle$?"* where $a \in \{0,1\}^n$ and $\langle a, x\rangle \coloneqq \sum_i a_i x_i \bmod 2$. Once enough queries have been made, T outputs $f(x)$. Parity decision trees are more powerful than ordinary decision trees: We have $\mathsf{D}^{\mathsf{pt}}(f) \le \mathsf{D}^{\mathsf{dt}}(f)$ where $\mathsf{D}^{\mathsf{dt}}(f)$ (resp. $\mathsf{D}^{\mathsf{pt}}(f)$) denotes the (parity) decision tree complexity of f, defined as the least depth of a deterministic (parity) decision tree computing f. On the other hand, the n-bit XOR function is an example where $\mathsf{D}^{\mathsf{dt}}(\mathrm{XOR}) = n$ while $\mathsf{D}^{\mathsf{pt}}(\mathrm{XOR}) = 1$. We define a *randomised* parity decision tree $\mathcal{T}$ as a distribution over deterministic parity trees $T \sim \mathcal{T}$. Then $\mathsf{R}^{\mathsf{pt}}_{\varepsilon}(f)$ is defined as the worst-case depth (over both input and randomness of the tree) of the best randomised parity tree $\mathcal{T}$ computing f with error ε, that is, $\Pr[\mathcal{T}(x) \ne f(x)] \le \varepsilon$ for all x. As usual, we let $\mathsf{R}^{\mathsf{pt}} \coloneqq \mathsf{R}^{\mathsf{pt}}_{1/3}$. To simplify notation, we drop the superscript pt and write $\mathsf{D} = \mathsf{D}^{\mathsf{pt}}$ and $\mathsf{R} = \mathsf{R}^{\mathsf{pt}}$ for short.

Our main research question is now formulated as follows. Let $f^k\colon (\{0,1\}^n)^k \to \{0,1\}^k$ denote the *direct sum* function that takes k instances $x \coloneqq (x^1, \dots, x^k)$ and returns the value of f on each of them, $f^k(x) \coloneqq (f(x^1), \dots, f(x^k))$. We study the following question.

Question 1. *Do we have $\mathsf{R}(f^k) \ge \Omega(k) \cdot \mathsf{R}(f)$ for every function f?*

We show two (incomparable) main results: We answer Question 1 affirmatively when the randomised parity decision tree lower bound is proved using the *discrepancy method* (Section 1.1), or when the lower bound is proved relative to a *product distribution* (Section 1.2).

1.1 First result: Direct sum for discrepancy

Discrepancy is one of the oldest-known methods for proving randomised communication lower bounds [Yao83, BFS86]. Let us tailor its definition to the setting of randomised parity trees. Thinking of $\{0,1\}^n$ as the vector space $\mathbb{Z}_2^n$, consider some affine subspace $S \subseteq \{0,1\}^n$ and a probability distribution μ over the inputs $\{0,1\}^n$. The discrepancy of S measures how biased f is on S. Namely, let $C_S^b \coloneqq \Pr_{\boldsymbol{x}\sim\mu}[f(\boldsymbol{x}) = b \wedge \boldsymbol{x} \in S]$. The difference $\Delta_S \coloneqq |C_S^0 - C_S^1|$ is called the bias of S under μ. We define $\mathsf{bias}(f)$ as the minimum over μ of the the maximum difference Δ_S an affine subspace can attain. Finally, the *discrepancy bound* $\mathsf{disc}(f)$ is defined as $\log(1/\mathsf{bias}(f))$. As in communication complexity, it is not hard to see that $\mathsf{R}(f) \ge \Omega(\mathsf{disc}(f))$; see Section 3 for details.

Theorem 1. *We have $\mathsf{R}(f^k) \ge \Omega(k) \cdot \mathsf{disc}(f)$ for any function f.*

In particular, if we have a function f whose randomised parity decision tree complexity is equal to its discrepancy, $\mathsf{R}(f) = \Theta(\mathsf{disc}(f))$, then Theorem 1 shows $\mathsf{R}(f^k) \geq \Omega(k) \cdot \mathsf{R}(f)$ answering Question 1 for that function. To prove Theorem 1, we first establish a particularly simple characterisation of $\mathsf{disc}(f)$ that relies on affine spaces defined by a single constraint. We then prove a perfect direct sum (and even an XOR lemma) for discrepancy using Fourier analysis.

1.2 Second result: Direct sum for product distributions

The standard approach for proving randomised lower bounds is to use Yao's principle [Yao77], which states that $\mathsf{R}(f) = \max_\mu \mathsf{D}_{1/3}(f, \mu)$. Here $\mathsf{D}_\varepsilon(f, \mu)$ is the *distributional* ε-error complexity of f defined as the least depth of a (deterministic) parity tree T such that $\Pr_{\boldsymbol{x}\sim\mu}[T(\boldsymbol{x}) \neq f(\boldsymbol{x})] \leq \varepsilon$. We say that a distribution μ over $\{0,1\}^n$ is *product* if it can be written as the product of n independent Bernoulli distributions. We define the best lower bound provable using a product distribution as

$$\mathsf{D}^\times_\varepsilon(f) \coloneqq \max_{\mu \text{ product}} \mathsf{D}_\varepsilon(f, \mu) \qquad \text{and} \qquad \mathsf{D}^\times \coloneqq \mathsf{D}^\times_{1/3}.$$

Our second result answers Question 1 affirmatively (modulo logarithmic factors) whenever the randomised parity decision tree lower bound is proved relative to a product distribution.

Theorem 2. *We have $\mathsf{R}(f^k) \geq \Omega(k/\log n) \cdot \mathsf{D}^\times(f)$ for any n-bit function f.*

We show moreover that the $O(\log n)$-factor loss in Theorem 2 can be avoided when μ is the uniform distribution (or more generally any *bounded-bias* distribution). One should compare this to the state-of-the-art in communication complexity, where the quantitatively best distributional direct sum results are also for product distributions and suffer logarithmic-factor losses [JRS03, BBCR13].

To prove Theorem 2, we introduce a new complexity measure tailored for product distributions, which we call *skew complexity* $\mathsf{S}(f)$ and which we define precisely in Section 4. We prove that this new measure admits a perfect direct sum theorem, $\mathsf{S}(f^k) = \Omega(k) \cdot \mathsf{S}(f)$, and that it characterises the measure $\mathsf{D}^\times$ up to an $O(\log n)$ factor. (We also show that the logarithmic loss is necessary for our approach: there is a function f such that $\mathsf{S}(f) = O(1)$, even though $\mathsf{D}^\times(f) = \Theta(\log n)$.) We give a more in-depth technical overview in Section 2.

Comparison of main results. We also show that our two main results (Theorems 1 and 2) are incomparable: For some functions f, our first result gives a much stronger lower bound for f^k than the second result—and vice versa. See Section 7 for the proof.

Lemma 3. *The complexity measures* disc *and* $\mathsf{D}^\times$ *are incomparable:*

1. *There is an n-bit function f such that $\mathsf{disc}(f) = O(\log n)$ while $\mathsf{D}^\times(f) = \Theta(n)$.*
2. *There is an n-bit function f such that $\mathsf{disc}(f) = \Theta(n)$ while $\mathsf{D}^\times(f) = O(1)$.*

1.3 Related work

Parity decision trees. Even though the direct sum problem for parity decision trees has not been studied before, the model has been studied extensively. Parity decision trees were first defined by Kushilevitz and Mansour [KM93] in the context of learning theory. Several prior works have studied their basic combinatorial properties [ZS10, OWZ+14] as well as Fourier-analytic properties [GTW21, GSTW23], often with connections to the log-rank conjecture [TWXZ13, STV17, San19, CHZZ24, HHLO24, MS24]; see also the survey [KLMY21]. There are various lifting theorems involving parity decision trees: lifting from D^{pt} to D^{cc} [HHL18], from D^{dt} to D^{pt} [CMSS23, BK23, AFS24], and from R^{dt} to R^{pt} [SP25, BI24]. These lifting theorems have played a central role in proving lower bounds for proof systems that can reason using parities [IS20, EGI24, FHR+24, BCD24, CD24, AI24].

Decision trees. In the decision tree model with classical queries, a deterministic direct sum theorem, $\mathsf{D}^{\mathsf{dt}}(f^k) = k \cdot \mathsf{D}^{\mathsf{dt}}(f)$, and even the stronger *composition theorem*, $\mathsf{D}^{\mathsf{dt}}(g \circ f^k) = \mathsf{D}^{\mathsf{dt}}(g) \cdot \mathsf{D}^{\mathsf{dt}}(f)$, are easy to show by combining adversary strategies [Sav02]. In the randomised case, an optimal direct sum result, $\mathsf{R}^{\mathsf{dt}}(f^k) \geq \Omega(k) \cdot \mathsf{R}^{\mathsf{dt}}(f)$, is known [KvdW07, JKS10, Dru12]. Whether a composition theorem holds for randomised query complexity, $\mathsf{R}^{\mathsf{dt}}(g \circ f^k) \geq \Omega(\mathsf{R}^{\mathsf{dt}}(g) \cdot \mathsf{R}^{\mathsf{dt}}(f))$ (for total g and f), is a major open problem [BK18, BB20, BBGM22, BB23, San24]. In the randomised setting, it is possible that the direct sum problem f^k requires strictly more than $\Theta(k) \cdot \mathsf{R}^{\mathsf{dt}}(f)$ queries: if one wants to succeed in computing all k copies with probability $\geq 2/3$, then a naive application of the union bound would require each copy to have error $\ll 1/k$. Results stating that one sometimes has $\mathsf{R}^{\mathsf{dt}}(f^k) \geq \omega(k) \cdot \mathsf{R}^{\mathsf{dt}}(f)$ are called "strong" direct sum theorems [BB19, BKST24] and they sometimes hold even for composed functions [BGKW20, BKLS23, GM21].

Communication complexity. The direct sum question for deterministic communication complexity was posed in [FKNN95] and it remains a notoriously difficult open problem [IR24a]. By contrast, in the randomised setting, the direct sum problem is characterised by *information complexity* [BR14], which has inspired a line of works too numerous to cite here; see [IR24b, §1.1] for an up-to-date overview. One of the key findings is that a direct sum for communication protocols is *false* in full generality in the distributional setting [GKR16, RS18]. We leave open the intriguing possibility that the information complexity approach can be adapted to parity decision trees. Historically, one of the first direct sum theorems proved for randomised communication was for the discrepancy bound [Sha03, LSv08] (analogously to our Theorem 1). Here, discrepancy is known to be equivalent to to the γ_2-norm [LS08]. We also mention that a near-optimal direct sum theorem holds for product distributions [BBCR13] (analogously to our Theorem 2).

1.4 Open question: Deterministic direct sum

The main question left open by our work is Question 1, namely, whether $\mathsf{R} = \mathsf{R}^{\mathsf{pt}}$ admits a direct sum theorem for all functions f. However, we would also like to highlight the analogous question in the deterministic case $\mathsf{D} = \mathsf{D}^{\mathsf{pt}}$. As discussed above, this is a long-standing open problem in the case of deterministic communication complexity D^{cc}. The best results so far are:

1. $\mathsf{D}^{\mathsf{cc}}(f^k) \geq \tilde{\Omega}(k) \cdot \mathsf{D}^{\mathsf{cc}}(f)^{1/2}$ as proved in [FKNN95].
2. $\mathsf{D}^{\mathsf{cc}}(f^k) \geq \tilde{\Omega}(k) \cdot \mathsf{D}^{\mathsf{cc}}(f)/\log \operatorname{rank}(f)$ as proved in [IR24a].

We observe in Appendix A.1 that both approaches have analogues in the parity setting.

Theorem 4. *For any function f and $k \geq 1$,*

1. $\mathsf{D}(f^k) \geq k \cdot \mathsf{D}(f)^{1/2}$,
2. $\mathsf{D}(f^k) \geq k \cdot \mathsf{D}(f)/\log \operatorname{spar}(f)$.

We leave it as an open question whether a perfect direct sum theorem holds for deterministic parity decision trees. We think one should attack this problem before addressing the (presumably much harder) problem for deterministic communication complexity.

2 Technical overview

We focus here on our second main result in Theorem 2 stating that $\mathsf{R}(f^k) \geq \Omega(k/\log n) \cdot \mathsf{D}^{\times}(f)$ and which is technically the much more involved theorem. Our main technical result is the following direct sum result for distributional complexity. Here $\mu^k \coloneqq \mu \times \cdots \times \mu$ (k times).

Theorem 5. *For any $f\colon \{0,1\}^n \to \{0,1\}$, product distribution μ over $\{0,1\}^n$, and $k \geq 1$,*

$$\mathsf{D}_\varepsilon(f^k, \mu^k) \geq \Omega\left(\frac{k\delta}{\log n}\right) \cdot \mathsf{D}_{\varepsilon+\delta}(f, \mu) \qquad \forall \varepsilon, \delta \geq 0.$$

Theorem 2 follows by taking $\varepsilon = \delta = 1/6$. Indeed, let μ be the distribution achieving the maximum for $\mathsf{D}^{\times}$. Using the easy direction of the minimax principle:

$$\mathsf{R}(f^k) \geq \mathsf{D}_{1/6}(f^k, \mu^k) \geq \Omega(k/\log n) \cdot \mathsf{D}_{1/3}(f, \mu) \geq \Omega(k/\log n) \cdot \mathsf{D}^{\times}(f).$$

Warm-up: Uniform distribution. We showcase the basic proof technique by sketching the proof in the simple case where μ is the uniform distribution. Fix an n-bit function f and let $\mathcal{U}$ be the uniform distribution over $\{0,1\}^n$. In the uniform (and more generally in the *bounded-bias*) case, we are actually able to avoid the $\log n$ factor loss and obtain, for all $k \geq 1$,

$$\mathsf{D}_\varepsilon(f^k, \mathcal{U}^k) \geq \Omega(k\delta) \cdot \mathsf{D}_{\varepsilon+\delta}(f, \mathcal{U}) \qquad \forall \delta \geq 0. \tag{1}$$

Fix a decision tree T of depth d computing k copies of f with error at most ε when $\boldsymbol{x} \sim \mathcal{U}^k$. We show how to extract a tree T^* that computes a single copy $\boldsymbol{y} \sim \mathcal{U}$ with error at most $\varepsilon + \delta$ and depth $\leq O(d/k\delta)$. Leaves of T correspond to affine subspaces of $(\{0,1\}^n)^k$ of codimension $\leq d$. More generally, one can associate with any node v of T the set $C_v = \{w_1, \ldots, w_{d(v)}\}$ of linear constraints that led to the node ($d(v)$ is the depth of the node v; the root is at level 0) and the vector $b \in \{0,1\}^k$ of desired values. The set of inputs S_v that reach node v is then given by $S_v \coloneqq \{x \in (\{0,1\}^n)^k : \langle w_j, x\rangle = b_j,\ \forall j \in [d(v)]\}$.

Of relevance here are the *pure constraints* one can extract from C_v. A pure constraint for copy $i \in [k]$ is some $w \in (\{0,1\}^n)^k$ such that $w^j \neq 0^n$ if and only if $j = i$. To be more precise, the number of pure queries that can be extracted for query i at node v is defined with:

$$\mathrm{pure}_i(C_v) \coloneqq \dim(\mathrm{span}(C_v) \cap W_i) \quad \text{where} \quad W_i \coloneqq \left\{w \in (\{0,1\}^n)^k : w^j = 0^n,\ \forall j \neq i\right\}.$$

We describe next two illustrative example when there are $k = 2$ copies.

1. Node v corresponds to constraints "$x_1^1 + x_1^2 = 0$" and "$x_1^2 = 1$". Then, $\mathrm{pure}_1(C_v) = 1$ as it is possible to extract the pure parity constraint $x_1^1 = 1$ by adding the two constraints. In the same vein, $\mathrm{pure}_2(C_v) = 1$.

2. Node v corresponds to constraints "$x_1^1 + x_1^2 = 0$ and "$x_1^2 + x_2^2 = 1$". Then, $\mathrm{pure}_1(C_v) = 0$ as it not possible to extract a pure constraint for the first copy.

Observation 1. *For any node v, we have $d(v) \geq \sum_{i\in[k]} \mathrm{pure}_i(C_v)$.*

As the second example highlights, it is possible for the inequality to be strict. This is a notable difference with classical decision trees: for any subcube $C \in (\{0, 1, *\}^n)^k$, the sum of fixed bits of each copy is the total number of fixed bits in C.

Where to plant y? The overarching idea of our result is that under the uniform distribution, *queries that increase the pure rank for a copy are the only one that bring usable information.* It is thus enough to find a copy with low expected pure rank in T and plant the real instance y there. To make this precise, taking the expectation over leaves of T when $\boldsymbol{x} \sim \mathcal{U}$ with Observation 1 implies the existence of some copy $i \in [k]$ with low expected pure rank:

$$\mathbb{E}_{\boldsymbol{x}\sim\mathcal{U}^k}[\text{pure}_i(C_{\ell(\boldsymbol{x})})] \leq O(d/k).$$

Let us fix this advantageous copy to be $i = 1$. On input $y \in \{0,1\}^n$ we run the tree T with y planted as x^1 and delay actual querying of bits of y as much as possible. Suppose that the process has reached node v with constraint set C_v and there is a new parity query w to be answered. If $w \in \text{span}(C_v)$, the answer to that query can be found (an optimised tree would not do such a query). If $w \notin \text{span}(C_v)$, we say that w is *critical* for C_v if it would increase the pure rank for the first copy $\text{pure}_1(C_v \cup \{w\}) > \text{pure}_1(C_v)$. If w is critical, there is no way to avoid making a parity query to the real input y and our algorithm does it. If w is not critical however, it is enough to answer with an uniform bit (that is, move to a random child of v in T) without querying y at all.

To see this, further split $w = w^1 w^{-1}$, where $w^1 \in \{0,1\}^k$ is the constraint for the first copy and $w^{-1} \in (\{0,1\}^n)^{k-1}$ is the constraint for the rest of the copies. If w has $\text{pure}_1(C_v \cup \{w\}) = \text{pure}_1(C_v)$ and $w \notin \text{span}(C_v)$, it must be that $0^n w^{-1} \notin \text{span}(C_v)$. Since $\boldsymbol{x}^{-1}$ is drawn from the uniform distribution we thus have for any fixed y consistent with S_v:

$$\Pr_{\boldsymbol{x}^{-1}}\left[\langle w, y\boldsymbol{x}^{-1}\rangle = 0 \,|\, (y, \boldsymbol{x}^{-1}) \in S_v\right] = \Pr_{\boldsymbol{x}^{-1}}\left[\langle w^{-1}, \boldsymbol{x}^{-1}\rangle = \langle w^1, y\rangle \,|\, (y, \boldsymbol{x}^{-1}) \in S_v\right] = \frac{1}{2}. \tag{2}$$

Correctness and efficiency. Let us call the above randomised tree solving one copy as $\mathcal{T}$. Correctness can be argued by showing that the distribution of leaves attained in the process for $\boldsymbol{y} \sim \mathcal{U}$ is the same as the distribution of leaves attained by $\boldsymbol{x} \sim \mathcal{U}^k$ in T. On the other hand, $\mathcal{T}$ has expected depth $O(d/k)$ as a real query to $\boldsymbol{y}$ is only ever made $\text{pure}_i(C_\ell)$ times for each leaf ℓ. In conclusion, $\mathcal{T}$ has the following guarantees:

1. $\Pr_{\boldsymbol{y}\sim\mathcal{U},\boldsymbol{T}\sim\mathcal{T}}[\boldsymbol{T}(\boldsymbol{y}) \neq f(\boldsymbol{y})] \leq \varepsilon$.

2. $\mathbb{E}_{\boldsymbol{y}\sim\mathcal{U},\boldsymbol{T}\sim\mathcal{T}}[\#\text{queries}(\boldsymbol{T}, \boldsymbol{y})] \leq d/k$.

Using Markov inequality, it is possible to derandomise $\mathcal{T}$ to get a deterministic parity tree T^* solving f with a worst-case guarantee instead of an average-case one. This step introduces a parameter δ controlling a trade-off between cost and error and yields the desired result (1).

2.1 Beyond uniform: The skew measure

Observe that (2) can fail badly for non-uniform μ. As an illustrative example suppose that two random bits $\boldsymbol{a}, \boldsymbol{b}$ are generated with $\boldsymbol{a} \sim \mathsf{Ber}(1/2)$ and $\boldsymbol{b} \sim \mathsf{Ber}(1/8)$. The constraint $\boldsymbol{a} \oplus \boldsymbol{b} = 1$ is not pure from the point of view of $\boldsymbol{a}$. However, since $\boldsymbol{b}$ is skewed towards being 0, the realisation of the constraint gives information about $\boldsymbol{a}$: $\Pr[\boldsymbol{a} = 0 \,|\, \boldsymbol{a} + \boldsymbol{b} = 1] = 1/8 \ll 1/2$. Thus, it seems one needs to query $\boldsymbol{a}$ to answer to the query $\boldsymbol{a} + \boldsymbol{b}$ even though the query is not critical for $\boldsymbol{a}$!

To circumvent this, we introduce the *skew* measure. This new measure is built around the observation that each bit of an input $\boldsymbol{x} \sim \mu$ can be sampled independently in two steps. Indeed, the following process is equivalent to $\mathsf{Ber}(1/8)$:

1. Let $\boldsymbol{\rho} \in \{0, \star\}$ be '0' with probability $3/4$ and $\star$ with probability $1/4$.

2. If $\boldsymbol{\rho} = 0$, return '0', else return a sample $\mathsf{Ber}(1/2)$.

Note that if we are "lucky" and $\boldsymbol{\rho} = \star$, we are back in the uniform case and (2) holds again. If not, we have somehow pre-emptively fixed the return bit to value 0. The skew measure explicitly splits product distributions into a *random partial fixing* $\boldsymbol{\rho}$ followed by a uniform distribution over unfixed bits of $\boldsymbol{\rho}$. A tree computing in this model gets help from $\boldsymbol{\rho}$ because $\boldsymbol{\rho}$ reduces the complexity of the function. When those bits are unfixed, it is on the other hand easier to analyse the behaviour of the tree as it is the uniform case again.

In Sections 5 and 6, we show a perfect direct sum for the skew measure and that perhaps surprisingly, this new measure is only a $\log n$-factor away from $\mathsf{D}^\times$.

3 Direct sum for disc

The goal of this section is to prove Theorem 1, restated here for convenience.

Theorem 1. *We have* $\mathsf{R}(f^k) \geq \Omega(k) \cdot \mathsf{disc}(f)$ *for any function* f*.*

Let us start by defining discrepancy formally. We denote by $\mathcal{S}_n$ the set of all affine subspaces of $\{0,1\}^n$ and $\mathcal{O}_n \subseteq \mathcal{S}_n$ the set of affine subspaces of codimension 1. Note that all spaces $S \in \mathcal{O}_n$ can be written as $S = \{x \in \{0,1\}^n : \langle a, x\rangle = b\}$ for some $a \in \{0,1\}^n$ and $b \in \{0,1\}$.

Definition 6. *Let* $f\colon \{0,1\}^n \to \{0,1\}$ *be a boolean function and* μ *be a distribution over* $\{0,1\}^n$*. The (parity) discrepancy of* f *with respect to* μ *is defined as:*

$$\mathsf{disc}(f,\mu) \coloneqq -\log \max_{S\in\mathcal{S}^n} \mathsf{bias}(f,\mu,S) \quad \text{where} \quad \mathsf{bias}(f,\mu,S) \coloneqq \left|\sum\nolimits_{x\in S}(-1)^{f(x)}\mu(x)\right|.$$

The (parity) discrepancy of f *is* $\mathsf{disc}(f) \coloneqq \max_\mu \mathsf{disc}(f,\mu)$ *where* μ *ranges over all distributions.*

Observe that $\mathsf{disc}(f) \geq 1$ for all non-constant f and by standard arguments, $\mathsf{R}(f) \geq \mathsf{disc}(f)$ (see Lemma 39). Using the latter, the only thing left to get Theorem 1 is to prove a direct sum result for discrepancy. We do this in a very strong way by actually establishing an XOR lemma for disc. Let $f^{\oplus k}$ denote the function that takes k instance and aggregates their result under f using XOR, so that $f^{\oplus k}(x^1,\dots,x^k) \coloneqq f(x^1) \oplus \cdots \oplus f(x^k)$.

Lemma 7. *For any function* f*, distribution* μ *and* $k \geq 1$*,*

$$k \cdot \mathsf{disc}(f,\mu) \geq \mathsf{disc}(f^{\oplus k},\mu^k) \geq k \cdot \big(\mathsf{disc}(f,\mu) - 1\big).$$

This result is the strongest possible. Indeed, we cannot omit the "-1" on the right because of the counterexample $f \coloneqq$ XOR: we have $\mathsf{disc}(f^{\oplus k},\mu^k) \leq 1$ for any distribution μ. In Appendix A.2 we revisit this XOR lemma and show that it also holds in the distribution-free setting, with $\mathsf{disc}(f^{\oplus k}) \approx k \cdot \mathsf{disc}(f)$. As a final comment, we note that it is easier to work with $f^{\oplus k}$ instead of f^k in the discrepancy setting, as it is somewhat tedious to define discrepancy for multi-valued functions. Before formally proving Lemma 7, we show how it is used to prove the main result Theorem 1.

Proof of Theorem 1. Any decision tree computing f^k can be converted to a decision tree computing $f^{\oplus k}$. This is achieved by replacing the label $y \in \{0,1\}^k$ of each leaf by its parity $\langle y, 1^k\rangle$. This operation does not increase the error probability or cost and so, using the easy direction of Yao's principle:

$$\begin{aligned}
\mathsf{R}(f^k) &\geq \max_{\mu} \mathsf{D}(f^k, \mu^k, 1/3) && \text{(Lemma 38)}\\
&\geq \max_{\mu} \mathsf{D}(f^{\oplus k}, \mu^k, 1/3)\\
&\geq \max_{\mu} \mathsf{disc}(f^{\oplus k}, \mu^k) - \log_2(3) && \text{(Lemma 39)}\\
&\geq k \cdot \max_{\mu}(\mathsf{disc}(f, \mu) - 1) - \log_2(3) && \text{(Lemma 7)}\\
&\geq k \cdot (\mathsf{disc}(f) - 1) - \log_2(3).
\end{aligned}$$

If $\mathsf{disc}(f) \geq 10$, then the string of inequalities yields $k \cdot (\mathsf{disc}(f) - 1) - \log_2(3) \geq k \cdot \mathsf{disc}(f)/10$. If f is constant, the claim is vacuously true. Finally, we show that for any non-constant f, $\mathsf{R}(f^k) \geq k - \log(3/2)$ which completes the claim. Indeed, if $\mathsf{disc}(f) \leq 10$, then $k - \log(3/2) \geq k \cdot \mathsf{disc}(f)/100$.

To this end, let f be a non-constant function and μ a distribution over $\{0,1\}^n$ which is balanced over 0-inputs and 1-inputs, i.e. $\mu(f^{-1}(0)) = \mu(f^{-1}(1)) = 1/2$. Let T be the best deterministic parity decision tree for $\mathsf{D}_{1/3}(f, \mu)$ and suppose toward contradiction that it has strictly less than $L \coloneqq 2^k \cdot (2/3)$ leaves. Let $G \subseteq \{0,1\}^n$ be the set of solutions which appear as a label on a leaf of T. We have $|G| < L$ and since μ is balanced, any solution $y \in \{0,1\}^k$ is equally likely so that:

$$\Pr_{\boldsymbol{x} \sim \mu^k}[T(\boldsymbol{x}) = f^k(\boldsymbol{x})] \leq \Pr_{\boldsymbol{x} \sim \mu^k}[f^k(\boldsymbol{x}) \in G] \leq |G| \cdot 2^{-k} < 2/3.$$

Thus, T errs with probability $> 1/3$: a contradiction. □

We now proceed to prove Lemma 7 in three steps.

3.1 Step 1: Characterisation of discrepancy

Much like discrepancy for communication protocols can be characterised by the γ_2-norm of the communication matrix [Sha03, LS08], we show that the parity discrepancy of f on μ is characterised by the L_∞-norm of the Fourier transform of a related function F_μ. This characterisation has two purposes. First, proving an XOR lemma requires exploring all the possible ways for the k copies to sum to 1. This kind of convolution operation is greatly simplified in the Fourier domain, where it simply corresponds to standard multiplication. Second, the characterisation is also quite convenient to prove lower bounds on $\mathsf{disc}(f, \mu)$ (which we do in Sections 7 and 8): it shows that maximum bias is (almost) attained for affine spaces of codimension 1 already.

The function F_μ. We relate a real-valued boolean function $F\colon \{0,1\}^n \to \mathbb{R}$ with its Fourier transform $\widehat{F} : \{0,1\}^n \to \mathbb{R}$ using the usual basis:

$$\forall z \in \{0,1\}^n, \quad \widehat{F}(z) \coloneqq \sum\nolimits_{x \in \{0,1\}^n} F(x) \cdot (-1)^{\langle x,z \rangle} \cdot 2^{-n}; \qquad \text{[Fourier transform]}$$

$$\forall x \in \{0,1\}^n, \quad F(x) \coloneqq \sum\nolimits_{z \in \{0,1\}^n} \widehat{F}(z) \cdot (-1)^{\langle z,x \rangle}. \qquad \text{[Inverse Fourier transform]}$$

See also [O'D14] for more background on Fourier analysis. We use $\|\widehat{F}\|_\infty$ to denote the maximum absolute value of a Fourier coefficient of F. To analyze $\mathsf{disc}(f, \mu)$, we introduce an associated function $F_\mu\colon \{0,1\}^n \to \mathbb{R}$ defined by $F_\mu(x) \coloneqq (-1)^{f(x)} \cdot \mu(x) \cdot 2^n$ and prove the following characterisation.

Lemma 8. *For every function $f\colon \{0,1\}^n \to \{0,1\}$ and distribution μ over $\{0,1\}^n$:*

$$\max_{S\in\mathcal{O}_n} \mathsf{bias}(f,\mu,S) \le \max_{S\in\mathcal{S}_n} \mathsf{bias}(f,\mu,S) \le \|\widehat{F_\mu}\|_\infty \le 2\cdot \max_{S\in\mathcal{O}_n} \mathsf{bias}(f,\mu,S).$$

Proof. The first inequality holds immediately because $\mathcal{O}_n \subseteq \mathcal{S}_n$. For the second, fix a maximizing $S \in \mathcal{S}^n$. Suppose that $\mathrm{codim}(S) = d$ and fix its constraints $a_j \in \{0,1\}^n$ and $b_j \in \{0,1\}$ for $j \in [d]$ so that $S = \{x \in \{0,1\}^n : \langle a_j, x\rangle = b_j \; \forall j \in [d]\}$. Observe that the vectors $\{a_j\}_{j\in[d]}$ are linearly independent. Let $\Phi := \sum_{x\in S}(-1)^{f(x)}\mu(x)$ so that $\mathsf{bias}(f,\mu,S) = |\Phi|$ and observe that

$$\Phi = 2^{-n}\cdot \sum_{x\in S} F_\mu(x) = 2^{-n}\cdot \sum_{x\in S}\sum_{z\in\{0,1\}^n} \widehat{F}_\mu(z)(-1)^{\langle z,x\rangle} = 2^{-n}\cdot \sum_{z\in\{0,1\}^n} \widehat{F}_\mu(z)\sum_{x\in S}(-1)^{\langle z,x\rangle}.$$

We focus on analysing terms $T_z := \sum_{x\in S}(-1)^{\langle z,x\rangle}$. Let $V := \mathrm{span}\{a_1, \ldots, a_d\}$ and observe that whenever $z \in V$, $|T_z| = |S|$. Indeed, if $\beta_1, \ldots, \beta_d \in \{0,1\}$ is a linear combination of z in V:

$$T_z = \sum_{x\in S}(-1)^{\langle z,x\rangle} = \sum_{x\in S}\prod_{j\in[d]}(-1)^{\beta_j\langle a_j,x\rangle} = \sum_{x\in S}(-1)^{\sum_j \beta_j b_j} = |S|\cdot(-1)^{\sum_j \beta_j b_j}.$$

On the other hand, $T_z = 0$ for all $z \notin V$. Indeed, Letting $S^b = S \cap \{x \in \{0,1\}^n : \langle x,z\rangle = b\}$ we have $T_z = |S^0| - |S^1|$. Because $z \notin V$, the constraint $\langle x,z\rangle = b$ splits S in half and thus $|S^0| = |S^1| = |S|/2$. Factoring in those observation, we get:

$$|\Phi| = 2^{-n}\cdot\left|\sum\nolimits_{z\in\{0,1\}^n} \widehat{F_\mu}(z)\cdot T_z\right| \le 2^{-n}\cdot|S|\cdot\sum\nolimits_{z\in V}\left|\widehat{F_\mu}(z)\right| \le 2^{-n}\cdot|S|\cdot|V|\cdot\|\widehat{F_\mu}\|_\infty.$$

Recall that S has codimension d and as such $|S| = 2^{n-d}$ and $|V| = 2^d$, implying the desired inequality $\mathsf{bias}(f,\mu,S) \le \|\widehat{F_\mu}\|_\infty$. We now prove the third inequality of the lemma. Fix any maximum Fourier coefficient $y^\star \in \{0,1\}^n$ and observe:

$$\|\widehat{F_\mu}\|_\infty = |\widehat{F_\mu}(y^\star)| = \left|\sum\nolimits_{x\in\{0,1\}^n} F_\mu(x)\cdot(-1)^{\langle x,y^\star\rangle}\cdot 2^{-n}\right| \le 2\cdot\max_{b\in\{0,1\}}\left|\sum\nolimits_{x:\langle x,y\rangle=b} F_\mu(x)\cdot 2^{-n}\right|.$$

Fix the maximizing argument to $b^\star$ and define $S^\star := \{x \in \{0,1\}^n : \langle x,y^\star\rangle = b^\star\}$. Note that $S^\star \in \mathcal{O}_n$ and as such:

$$\|\widehat{F_\mu}\|_\infty \le 2\cdot\left|\sum\nolimits_{x\in S^\star}(-1)^{f(x)}\mu(x)\right| \le 2\cdot\max_{S\in\mathcal{O}_n}\mathsf{bias}(f,\mu,S).$$ □

3.2 Step 2: Direct sum for the maximum Fourier coefficient

The outer-product of functions $F, G : \{0,1\}^n \to \mathbb{R}$ is defined as the function $F \otimes G : \{0,1\}^{2n} \to \mathbb{R}$ with $(F\otimes G)(x^1,x^2) := F(x^1)\cdot G(x^2)$. Next is a direct sum result for its max Fourier coefficient.

Claim 9. *For any $F, G : \{0,1\}^n \to \mathbb{R}$, $\|\widehat{F\otimes G}\|_\infty = \|\widehat{F}\|_\infty \cdot \|\widehat{G}\|_\infty$.*

Proof. Let $H = F \otimes G$; for any $z^1, z^2 \in \{0,1\}^n$, the definition of Fourier transform implies

$$\begin{aligned}\widehat{H}(z^1,z^2) &= 2^{-2n}\cdot\sum\nolimits_{x^1,x^2\in\{0,1\}^n} H(x^1,x^2)\cdot(-1)^{\langle x^1x^2,z^1z^2\rangle}\\ &= 2^{-2n}\cdot\sum\nolimits_{x^1,x^2\in\{0,1\}^n} F(x^1)\cdot G(x^2)\cdot(-1)^{\langle x^1,z^1\rangle}\cdot(-1)^{\langle x^2,z^2\rangle}\\ &= \widehat{F}(z^1)\cdot\widehat{G}(z^2).\end{aligned}$$

From this, the equivalence is immediate:

$$\|H\|_\infty = \max_{z^1,z^2}|\widehat{H}(z^1,z^2)| = \max_{z^1,z^2}|\widehat{F}(z^1)|\cdot|\widehat{G}(z^2)| = \|F\|_\infty\cdot\|G\|_\infty.$$ □

3.3 Step 3: Conclusion

We tie together Lemma 8 and Claim 9 and prove Lemma 7.

Proof of Lemma 7. Let $H\colon (\{0,1\}^n)^k \to \mathbb{R}$ be the function associated with $f^{\oplus k}$ and μ^k in Lemma 8. It is possible to express H as the k-fold outer-product of F_μ: $H = F_\mu \otimes \cdots \otimes F_\mu$. Indeed, for $x \in (\{0,1\}^n)^k$, we have:

$$H(x) = 2^{-kn} \cdot (-1)^{f^{\oplus k}(x)} \mu^k(x) = \prod\nolimits_{i\in[k]} 2^{-n}(-1)^{f(x^i)}\mu(x^i) = \prod\nolimits_{i\in[k]} F_\mu(x^i).$$

Thus, using the characterisation of Lemma 8 and Claim 9 k times:

$$\max_{S\in\mathcal{S}_{kn}} \mathsf{bias}(f^{\oplus k}, \mu^k, S) \le \|\widehat{H}\|_\infty = \left(\|\widehat{F_\mu}\|_\infty\right)^k \le 2^k \cdot \left(\max_{S\in\mathcal{S}_n} \mathsf{bias}(f,\mu,S)\right)^k.$$

The XOR-lemma $\mathsf{disc}(f^{\oplus k}, \mu^k) \ge k\cdot(\mathsf{disc}(f,\mu)-1)$ follows directly. We now show the other direction, $\mathsf{disc}(f^{\oplus k}, \mu^k) \le k \cdot \mathsf{disc}(f,\mu)$. To do so, fix some $S \in \mathcal{S}_n$ maximizing $\mathsf{bias}(f,\mu,S)$ and define $T \in \mathcal{S}_{kn}$ which is concatenation of k copies of S. Formally:

$$T = \left\{x \in (\{0,1\}^n)^k : x^i \in S \quad \forall i \in [k]\right\}.$$

Now, it is easy to check that $\mathsf{bias}(f^{\oplus k}, \mu^k, T) = \mathsf{bias}(f,\mu,S)^k$ and the claim follows. □

4 Direct sum for $\mathsf{D}^\times$ part I: proof organisation

The goal of this section is to prepare the ground for a proof of our main technical contribution: a direct sum for parity trees in the distributional setting (restated below).

Theorem 5. *For any $f\colon \{0,1\}^n \to \{0,1\}$, product distribution μ over $\{0,1\}^n$, and $k \ge 1$,*

$$\mathsf{D}_\varepsilon(f^k, \mu^k) \ge \Omega\left(\frac{k\delta}{\log n}\right) \cdot \mathsf{D}_{\varepsilon+\delta}(f,\mu) \qquad \forall \varepsilon, \delta \ge 0.$$

The plan is as follows. We first give two precise strengthenings of Theorem 5, we then introduce the skew measure and finally present the overall proof structure.

4.1 Two strengthenings of Theorem 5

For technical convenience, we study distributional complexity for *randomised* trees. For a deterministic parity tree T we let $q(T,x)$ be the number of queries made by T on input x. If $\mathcal{T}$ is a randomised tree and μ is a distribution, we define $\overline{q}(\mathcal{T},\mu)$ and $\mathrm{err}_f(\mathcal{T},\mu)$ in the natural way with:

$$\overline{q}(\mathcal{T},\mu) \coloneqq \mathbb{E}_{\substack{\boldsymbol{T}\sim\mathcal{T}\\ \boldsymbol{x}\sim\mu}}[q(\boldsymbol{T},\boldsymbol{x})] \quad \text{and} \quad \mathrm{err}_f(\mathcal{T},\mu) \coloneqq \Pr_{\substack{\boldsymbol{T}\sim\mathcal{T}\\ \boldsymbol{x}\sim\mu}}[\boldsymbol{T}(\boldsymbol{x}) \ne f(\boldsymbol{x})].$$

Finally, we define $\overline{\mathsf{D}}_\varepsilon(f,\mu) = \min_{\mathcal{T}}\{\overline{q}(\mathcal{T},\mu) : \mathrm{err}_f(\mathcal{T},\mu) \le \varepsilon\}$. It is clear that $\overline{\mathsf{D}}_\varepsilon(f,\mu) \le \mathsf{D}_\varepsilon(f,\mu)$ but a converse result is more complicated, as the derandomisation can increase both the error and the depth simultaneously.

Claim 10. *For any $f\colon \{0,1\}^n \to \{0,1\}$, μ over $\{0,1\}^n$ and $\varepsilon, \delta \ge 0$, $\mathsf{D}_{\varepsilon+\delta}(f,\mu) \le \overline{\mathsf{D}}_\varepsilon(f,\mu)/\delta$.*

We delay a proof of this folklore fact to Appendix A.3. We also refer readers to [JKS10] which proves the analogue for ordinary decision trees. With this tool in hand, we can reduce Theorem 5 to Theorem 11 (see below). Indeed, Theorem 5 is immediately true in the regime $\delta \le 2/n$. For $\delta \ge 2/n$, setting $\gamma := 1/n$ in Theorem 11 and derandomising with parameter $\delta' := \delta - 1/n$ in Claim 10 is enough.

Theorem 11. *For any $f\colon \{0,1\}^n \to \{0,1\}$, product distribution μ, and $k \ge 1$,*

$$\overline{\mathsf{D}}_\varepsilon(f^k, \mu^k) \ge \Omega\big(k/\log(n/\gamma)\big) \cdot \overline{\mathsf{D}}_{\varepsilon+\gamma}(f, \mu) \quad \forall \gamma \in (0, 1/n).$$

Definition 12. *We say that a product distribution μ over $\{0,1\}^n$ is λ-bounded for some $\lambda \in (0,1]$ if $\Pr_{\boldsymbol{x}\sim\mu}[\boldsymbol{x}_i = 1] \in [\lambda/2, 1-\lambda/2]$ for every $i \in [n]$.*

In the next sections, we also show the following qualitative improvement over Theorem 11 for bounded distributions.

Theorem 13. *For any $f\colon \{0,1\}^n \to \{0,1\}$, λ-bounded distribution μ and $k \ge 1$,*

$$\overline{\mathsf{D}}_\varepsilon(f^k, \mu^k) \ge \Omega\left(k\lambda\right) \cdot \overline{\mathsf{D}}_\varepsilon(f, \mu).$$

Let us highlight the difference between Theorem 11 and Theorem 13: the latter is free from both the $\log n$ factor and the extra error γ. This theorem is especially interesting when the hard distribution for the function at hand (e.g. MAJ) is close to the uniform one.

4.2 The Skew measure

For the rest of this paper, we let $\mathcal{U}$ be the uniform distribution. Let μ be a distribution over $\{0,1\}^n$ and $S \subseteq \{0,1\}^n$. We use $\mu(S) := \sum_{s\in S} \mu(s)$ to denote the mass of S with respect to μ. When $\mu(S) > 0$, we let μ_S be the distribution of μ conditioned on S. Let $\rho \in \{0, \star\}^n$ be a partial assignment corresponding to the sub-cube $C_\rho = \{x \in \{0,1\}^n : \rho_i = 0 \implies x_i = 0 \quad \forall i \in [n]\}$. We use μ_ρ to denote μ_{C_ρ}.

4.2.1 Random partial fixings

Let μ be a product distribution over $\{0,1\}^n$. We say that μ is 0-*biased* if $\Pr_{\boldsymbol{x}\sim\mu}[\boldsymbol{x}_i = 0] \ge 1/2$ for every $i \in [n]$. For the rest of the paper, we will assume without loss of generality that any encountered input distribution is 0-biased. Indeed, should μ not be 0-biased, we can apply the following iterative transformation. Let $f_0 := f$ and $\mu_0 := \mu$. For every $i \in [n]$, if $\Pr_{\boldsymbol{x}\sim\mu}[\boldsymbol{x}_i = 1] \le 1/2$ – the coordinate is already biased in the right direction – we simply leave $f_i := f_{i-1}$ and $\mu_i := \mu_{i-1}$. Otherwise, let:

$$f_i(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) := f_{i-1}(x_1, \ldots, x_{i-1}, 1 - x_i, x_{i+1}, \ldots, x_n);$$
$$\mu_i(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) := \mu_{i-1}(x_1, \ldots, x_{i-1}, 1 - x_i, x_{i+1}, \ldots, x_n).$$

Observe that μ_n is 0-biased and $\overline{\mathsf{D}}_\varepsilon(f_n^k, \mu_n^k) = \overline{\mathsf{D}}_\varepsilon(f^k, \mu^k)$ for every $\varepsilon \ge 0$ and $k \ge 1$. Now that we are certain that μ is 0-biased, let $\delta_i := 2\Pr_{\boldsymbol{x}\sim\mu}[\boldsymbol{x}_i = 1] \in [0,1]$. We define next the *random partial fixing* distribution with respect to μ. The intuition comes from the observation that each bit of μ can be written as a convex combination of the fixed bit '0' and a uniform bit.

Definition 14 (Random Partial Fixing)**.** *The random partial fixing with respect to μ, denoted $\mathcal{R}_\mu$, is a distribution of partial assignments $\boldsymbol{\rho} \in \{0, \star\}^n$ sampled as follows: For each $i \in [n]$, we set independently*

$$\boldsymbol{\rho}_i = \begin{cases} 0 & \text{w.p. } 1 - \delta_i \\ \star & \text{w.p. } \delta_i \end{cases}.$$

Observe that the following alternative two-step process is equivalent to sampling an input directly from μ. First, sample $\boldsymbol{\rho} \sim \mathcal{R}_\mu$ and then sample and return $\boldsymbol{x} \sim \mathcal{U}_{\boldsymbol{\rho}}$.

4.2.2 The new measure

Given a parity decision tree T and a partial assignment ρ over the input string, let T_ρ denote the pruned T by

1. fixing all the variables in the support of ρ,
2. removing redundant queries (those can be written as a linear combination of previous queries).

For randomised parity decision tree $\mathcal{T}$, we define $\mathcal{T}_\rho$ as the distribution of $\boldsymbol{T}_\rho$, where $\boldsymbol{T} \sim \mathcal{T}$.

Definition 15. *For every randomised parity decision tree $\mathcal{T}$ and product distribution μ, define the skew average cost $\overline{sq}(\mathcal{T}, \mu) := \mathbb{E}_{\boldsymbol{\rho} \sim \mathcal{R}_\mu}[\overline{q}(\mathcal{T}_{\boldsymbol{\rho}}, \mathcal{U}_{\boldsymbol{\rho}})]$. Let $f\colon \{0,1\}^n \to \{0,1\}$ be a function. For $\varepsilon \geq 0$, we define the skew measure $\mathsf{S}_\varepsilon(f)$ with:*

$$\mathsf{S}_\varepsilon(f, \mu) := \min_{\mathcal{T}} \{\overline{sq}(\mathcal{T}, \mu) \mid \mathrm{err}_f(\mathcal{T}, \mu) \leq \varepsilon\}.$$

Claim 16. *For any $f\colon \{0,1\}^n \to \{0,1\}$, product distribution μ, and $\varepsilon \geq 0$, $\overline{\mathsf{D}}_\varepsilon(f, \mu) \geq \mathsf{S}_\varepsilon(f, \mu)$. Furthermore, equality holds if $\mu = \mathcal{U}$.*

Proof. The claim is immediate as $\overline{sq}(\mathcal{T}, \mu) \leq \overline{q}(\mathcal{T}, \mu)$ for every randomised parity tree $\mathcal{T}$ and product distribution μ. □

4.3 Proof plan

The proofs of Theorems 11 and 13 are carried out in two steps. First, we prove a perfect direct sum for the skew measure in Section 5.

Theorem 17. *We have $\mathsf{S}_\varepsilon(f^k, \mu^k) \geq k \cdot \mathsf{S}_\varepsilon(f, \mu)$ for any function f, product μ and $\varepsilon \geq 0$.*

As a second step, we demonstrate in Section 6 that $\overline{\mathsf{D}}_\varepsilon(f, \mu) \approx \mathsf{S}_\varepsilon(f, \mu)$. We first prove a lossless conversion for product distribution which are *constant-bounded*. We then extend this to general product distributions for which we lose a $\log(n)$-factor. Let us recall here that the $\log n$ loss for general (unbounded) product distribution is inherent to the skew measure. Indeed, we show in Section 8 the existence of some f and μ for which $\overline{\mathsf{D}}_{1/3}(f, \mu) = \Theta(\log n)$ but $\mathsf{S}_0(f, \mu) = \Theta(1)$.

Theorem 18. *For any $f\colon \{0,1\}^n \to \{0,1\}$, product distribution μ, $\gamma \in (0, 1/n)$, we have*

$$\overline{\mathsf{D}}_{\varepsilon+\gamma}(f, \mu) \leq O\big(\log(n/\gamma)\big) \cdot \mathsf{S}_\varepsilon(f, \mu) \quad \forall \varepsilon \geq 0.$$

Theorem 19. *For any $f\colon \{0,1\}^n \to \{0,1\}$ and λ-bounded product distribution μ, we have*

$$\overline{\mathsf{D}}_\varepsilon(f, \mu) \leq O(1/\lambda) \cdot \mathsf{S}_\varepsilon(f, \mu) \quad \forall \varepsilon \geq 0.$$

Combining the results above it is now straightforward to conclude and prove Theorems 11 and 13. For instance, the proof of the former goes as follows.

Proof of Theorem 11.

$$\begin{aligned} \overline{\mathsf{D}}_\varepsilon(f^k, \mu^k) &\geq \mathsf{S}_\varepsilon(f^k, \mu^k) && \text{(Claim 16)} \\ &\geq k \cdot \mathsf{S}_\varepsilon(f, \mu) && \text{(Theorem 17)} \\ &\geq \Omega\big(k/\log(n/\gamma)\big) \cdot \overline{\mathsf{D}}_{\varepsilon+\gamma}(f, \mu). && \text{(Theorem 18)} \end{aligned}$$

□

4.4 Some notation

Let us finish this section by defining some notations which will be useful for the rest of the paper. Let T be a parity decision tree on input $\{0,1\}^n$. We define $\mathcal{N}(T)$ as the set of nodes of T and $\mathcal{L}(T)$ as the set of leaves of T. For each node $v \in \mathcal{N}(T)$, we define the following: (items marked with $*$ are only defined for non-leaf nodes)

- path(v): the set of nodes on the root-to-v path (including the root, excluding v)
- $d(v) \coloneqq |\text{path}(v)|$: the depth of v
- $*$ $Q^v \in \{0,1\}^n$: the query made at node v
- $*$ child(v, b) the child of v corresponding to the query outcome $\langle x, Q^v \rangle = b$, where $b \in \{0,1\}$
- $Q^{\prec v}$: an $n \times d(v)$ boolean matrix with column vectors $\{Q^u\}_{u \in \text{path}(v)}$
- $*$ $Q^{\preceq v} \coloneqq [Q^{\prec v}\ Q^v]$ of dimension $n \times (d(v)+1)$.
- $b^{\prec v} \in \{0,1\}^{d(v)}$: the labels on the root-to-v path

For every boolean matrix $A \in \{0,1\}^{n \times m}$, we use rank($A$) to denote the rank of A (understood as a matrix over $\mathbb{F}_2$) and let col(A) $\subseteq \{0,1\}^n$ be the column space of A. For every $S \subseteq [n]$, let $A_S \in \{0,1\}^{|S| \times m}$ stand for the sub-matrix of A consisting of row with indices in S. For every $x, y \in \{0,1\}^n$ and $S \subseteq [n]$, we denote $\langle x_S, y_S \rangle = \sum_{i \in S} x_i y_i$ by $\langle x, y \rangle_S$.

Let μ and ν be two distributions over S. We use $d_{\text{TV}}(\mu, \nu) \coloneqq \sup_{S' \subseteq S} |\mu(S') - \nu(S')|$ to denote the total variation distance between μ and ν and write $\mu \equiv \nu$ if $d_{\text{TV}}(\mu, \nu) = 0$.

5 Direct sum for $\mathsf{D}^\times$ part II: direct sum for S

In this section, we prove a perfect direct sum for S (restated below). A direct consequence of this fact is a perfect direct sum for distributional parity query complexity under the uniform distribution.

Theorem 17. *We have $\mathsf{S}_\varepsilon(f^k, \mu^k) \geq k \cdot \mathsf{S}_\varepsilon(f, \mu)$ for any function f, product μ and $\varepsilon \geq 0$.*

Corollary 20. *We have $\overline{\mathsf{D}}_\varepsilon(f^k, \mathcal{U}^k) \geq k \cdot \overline{\mathsf{D}}_\varepsilon(f, \mathcal{U})$ for any function f and $\varepsilon \geq 0$.*

Proof. Combine Claim 16 with Theorem 17. ☐

To prove Theorem 17, our overall strategy is to take a tree achieving $\mathsf{S}_\varepsilon(f^k, \mu^k)$ and extract a tree computing a single copy of f under μ to within error ε while having cost bounded by $\mathsf{S}_\varepsilon(f^k, \mu^k)/k$. To do so, we employ the extraction strategy hinted at in Section 2. The extractor works as long as the input distributions are uniform, which is the case after the random partial fixing step of S.

5.1 Extracting a single instance under uniform distributions

Let T be a deterministic parity tree taking inputs $x \in \mathcal{X} \coloneqq \{0,1\}^{m_1} \times \cdots \times \{0,1\}^{m_k}$ and returning labels in $\{0,1\}^k$. We assume without loss of generality that the queries along any root-to-leaf path are linearly independent. Let $L(\ell) \in \{0,1\}^k$ be the label associated with the leaf $\ell \in \mathcal{L}(T)$. For $i \in [k]$, we define the linear subspace $W_i \subseteq \mathcal{X}$ of query vectors that are zero everywhere except for copy i:

$$W_i \coloneqq \{w \in \mathcal{X} : w^j = 0^{m_j} \iff j \neq i\}.$$

We say a node $v \in \mathcal{N}(T)$ is *critical* with respect to i if $\text{col}(Q^{\prec v}) \cap W_i \neq \text{col}(Q^{\preceq v}) \cap W_i$ and denote the set of critical indices at node v with $I_v \coloneqq \{i \in [k] : v \text{ is critical w.r.t. } i\}$. Finally, we let $d_i(v) \coloneqq \sum_{u \in \text{path}(v)} \mathbb{1}[i \in I^u]$ be the relative depth of v with respect to instance i and highlight that $d_i(v) = \dim(\text{col}(Q^{\prec v}) \cap W_i)$. The algorithm $\mathsf{Ext}_i(T)$ which extracts a tree for the i-th instance out

Algorithm 1 $\mathsf{Ext}_i(T)$

Input: $y \in \{0,1\}^{m_i}$
Output: $a \in \{0,1\}$

1: Initialize $v \leftarrow$ root of $\boldsymbol{T}$
2: **while** v is not a leaf **do**
3:     **if** $i \in I^v$ **then**
4:         Let w be any vector in $(\mathrm{col}(Q^{\preceq v}) \setminus \mathrm{col}(Q^{\prec v})) \cap W_i$, query $\langle y, w\rangle$
5:         Compute $b^v \coloneqq \langle y, Q^v\rangle$ from $b^{\prec v}$ and $\langle y, w\rangle$
6:         Move $v \leftarrow \mathrm{child}(v, b^v)$
7:     **else**
8:         Sample $\boldsymbol{\xi} \sim \mathsf{Ber}(1/2)$
9:         Move $v \leftarrow \mathrm{child}(v, \boldsymbol{\xi})$
10:     **end if**
11: **end while**
12: **return** $L_i(v)$

of T is described in Algorithm 1. Observe that it is indeed possible to compute the value of $\langle y, Q^v\rangle$ from $b^{\prec v}$ and $\langle y, w\rangle$ on line 5: Since $w \notin \mathrm{col}(Q^{\prec v})$, we have $\mathrm{rank}([Q^{\prec v} \ \ w]) = \mathrm{rank}(Q^{\prec v}) + 1$. On the other hand, as $w \in \mathrm{col}(Q^{\preceq v})$, we have $\mathrm{rank}([Q^{\preceq v} \ \ w]) = \mathrm{rank}(Q^{\preceq v}) = \mathrm{rank}(Q^{\prec v}) + 1$. Thus $Q^v \in \mathrm{col}([Q^{\prec v} \ \ w])$, which means that Q^v can be written as a linear combination of the columns of $[Q^{\prec v} \ \ w]$: $Q^v = Q^{u_1} + \cdots + Q^{u_t} + w$ where $u_1, \ldots, u_t$ are some ancestors of v. This in turn implies that $\langle y, Q^v\rangle = \sum_{i\in[t]} \langle y, Q^{u_i}\rangle + \langle y, w\rangle$.

We stress that although T is a deterministic tree, $\mathsf{Ext}_i(T)$ is a randomized decision tree with internal randomness inherited from the bits $\boldsymbol{\xi}$. Our main technical claim is that for any fixed $y \in \{0,1\}^{m_i}$, the algorithm $\mathsf{Ext}_i(T)$ perfectly simulates a run of T when the input is on a random input $\boldsymbol{x} = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{i-1}, y, \boldsymbol{x}^{i+1}, \ldots, \boldsymbol{x}^k)$ and $\boldsymbol{x}^j \sim \mathcal{U}(\{0,1\}^{m_j})$. In a nutshell, the randomness of the other $k-1$ instances can be substituted with the internal randomness $\boldsymbol{\xi}$. To make this precise, we let $X_v = \{x \in \mathcal{X} : x^T Q^{\prec v} = b^{\prec v}\}$ be the set of inputs leading to the node $v \in \mathcal{N}(T)$.

Claim 21. *For any* $y \in \{0,1\}^{m_i}$, $\Pr_{\boldsymbol{\xi}}[\mathsf{Ext}_i(T)$ reaches node v in its execution on $y] = \Pr_{\boldsymbol{x}}[\boldsymbol{x} \in X^v]$.

Proof. Let us fix $i \coloneqq 1$ and $d \coloneqq d(v)$ for simplicity. We establish and alternative description of X^v that puts pure constraints on instance 1 first. Pick $t \coloneqq d_1(v)$ independent vectors $Q_1, \ldots, Q_t \in \mathrm{col}(Q^{\prec v}) \cap W_1$ and extend them arbitrarily to a basis $\{Q_j\}_{j\in[d]}$ of $Q^{\prec v}$. As each vector of this basis can be expressed as a linear combination of $\{Q^u\}_{u\in\mathrm{path}(v)}$, it is possible to apply those linear combination to $b^{\prec v}$ and obtain values $\{b_j\}_{j\in[d]}$ such that $X_v = \{x \in \mathcal{X} \mid \forall j \in [d] : \langle x, Q_j\rangle = b_j\}$. The set $Y^v \subseteq \{0,1\}^{m_1}$ of inputs that can reach node v in a run of $\mathsf{Ext}_1(T)$ thus corresponds to

$$Y^v \coloneqq \left\{y \in \{0,1\}^{m_1} \mid \forall j \in [t] : \langle y, Q_j^1\rangle = b_j\right\}.$$

If $y \notin Y^v$, the statement follows directly as both probabilities are zero. However, if $y \in Y^v$,

$$\Pr_{\boldsymbol{\xi}}[\mathsf{Ext}_1(T) \text{ reaches node } v \text{ in its execution}] = 2^{-d+t}.$$

This is so, because a node v can only be reached by having the "right" $d-t$ coin tosses of $\boldsymbol{\xi}$ (provided that $y \in Y^v$). Thus, it remains to show that $\Pr_{\boldsymbol{x}}[\boldsymbol{x} \in X^v] = 2^{-d+t}$ if $y \in Y^v$.

Let $m = \sum_{i\in[k]} m_i$ and $S = \{m_1 + 1, \ldots, m\}$ be the indices of the bits of every copies but the first one. Fix the $m \times (d-t)$ boolean matrix $A = [Q_{t+1} \cdots Q_d]$ and observe that $\mathrm{rank}(A) = d - t$ by construction. We show that $\mathrm{rank}(A_S) = d - t$ too. If $\mathrm{rank}(A_S) < \mathrm{rank}(A)$, we can find a non-empty set $J \subseteq \{t+1, \ldots, d\}$ such that $\sum_{j \in J} (Q_j)_S = 0$. This implies that $Q' := \sum_{j \in J} Q_j \in W_i \cap \mathrm{col}(Q^{\prec v})$. But Q' is linearly independent of $\{Q_1, \ldots, Q_t\}$ – this contradicts $\dim(\mathrm{col}(Q^{\prec v}) \cap W_i) = t$. Therefore, if $y \in Y^v$, we use this observation to conclude:

$$\begin{aligned}
\Pr_{\boldsymbol{x}}[\boldsymbol{x} \in X^v] &= \Pr_{\boldsymbol{x}}[\forall j \in [d] : \langle \boldsymbol{x}, Q_j \rangle = b_j] \\
&= \Pr_{\boldsymbol{x}}[\boldsymbol{x}^T A = (b_j)_{t+1 \le j \le d}] \\
&= \Pr_{\boldsymbol{z} := (\boldsymbol{x}^2, \ldots, \boldsymbol{x}^k)} \left[\boldsymbol{z}^T A_S = (b_j + \langle y, Q_j^1 \rangle)_{t+1 \le j \le d}\right] \\
&= 2^{-\mathrm{rank}(A_S)} \\
&= 2^{-d+t}.
\end{aligned}$$

□

5.2 Proof of Theorem 17

We are now ready to show Theorem 17. Let $\mathcal{T}$ be a randomised parity decision tree which witnesses $C := \mathsf{S}_\varepsilon(f^k, \mu^k)$. For each $i \in [k]$, define the randomized decision tree $\mathcal{T}_i \colon \{0,1\}^n \to \{0,1\}$ with:

1. Sample $\boldsymbol{T} \sim \mathcal{T}$.
2. Sample $\boldsymbol{\rho}^1, \ldots, \boldsymbol{\rho}^{i-1}, \boldsymbol{\rho}^{i+1}, \ldots, \boldsymbol{\rho}^k \sim \mathcal{R}_\mu$.
3. Let $\widetilde{\boldsymbol{\rho}} := (\boldsymbol{\rho}^1, \ldots, \boldsymbol{\rho}^{i-1}, \star^n, \boldsymbol{\rho}^{i+1}, \ldots, \boldsymbol{\rho}^k)$.
4. Return $\mathsf{Ext}_i(T_{\widetilde{\boldsymbol{\rho}}})$.

We show in Lemma 22 that $\mathrm{err}_f(\mathcal{T}_i, \mu) \le \varepsilon$ simultaneously for all $i \in [k]$. On the other hand, we show in Lemma 23 that $\sum_{i \in [k]} \overline{sq}(\mathcal{T}_i, \mu) \le C$. By an averaging argument, this shows the existence of a copy $i^\star \in [k]$ with cost $\le C/k$ and therefore $\mathsf{S}_\varepsilon(f, \mu) \le C/k$. The remainder of this section is devoted to proving both claims.

Lemma 22. *For every $i \in [k]$, $\mathrm{err}_f(\mathcal{T}_i, \mu) \le \mathrm{err}_{f^k}(\mathcal{T}, \mu^k)$.*

Proof. It is enough to prove the statement assuming $\mathcal{T}$ is a deterministic parity tree T and $i = 1$. Let $\mathcal{R}$ be the distribution of $\widetilde{\boldsymbol{\rho}}$ in the step 3 of generating $\mathcal{T}_1$. Fix some $\rho \in \mathrm{supp}(\mathcal{R})$ and note that $\rho^1 = \star^n$. We also define $\mathcal{U}^{-1} := \mathcal{U}_{\rho^2} \times \cdots \times \mathcal{U}_{\rho^k}$. Using Claim 21 on a leaf $\ell \in \mathcal{L}(T_\rho)$ yields:

$$\begin{aligned}
\Pr_{\boldsymbol{y}, \boldsymbol{\xi}}[\mathsf{Ext}_1(T_\rho) \text{ reaches } \ell \text{ on } \boldsymbol{y} \wedge L_1(\ell) \ne f(\boldsymbol{y})] &= \mathop{\mathbb{E}}_{\boldsymbol{y}}\left[\Pr_{\boldsymbol{\xi}}[\mathsf{Ext}_1(T_\rho) \text{ reaches } \ell \text{ on } \boldsymbol{y}] \cdot \mathbb{1}\left[L_1(\ell) \ne f(\boldsymbol{y})\right]\right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{y}}\left[\Pr_{\boldsymbol{x}^{-1} \sim \mathcal{U}^{-1}}[(\boldsymbol{y}, \boldsymbol{x}^{-1}) \in X^\ell] \cdot \mathbb{1}\left[L_1(\ell) \ne f(\boldsymbol{y})\right]\right] \\
&= \Pr_{\boldsymbol{x} \sim \mu \times \mathcal{U}^{-1}}[\boldsymbol{x} \in X^\ell \wedge L_1(\ell) \ne f(\boldsymbol{x}^1)].
\end{aligned}$$

Thus:

$$\begin{aligned}
\mathrm{err}_f(\mathcal{T}_1, \mu) &= \mathop{\mathbb{E}}_{\widetilde{\boldsymbol{\rho}} \sim \mathcal{R}}\left[\Pr_{\boldsymbol{y} \sim \mu, \boldsymbol{\xi}}[\mathsf{Ext}_1(T_{\widetilde{\boldsymbol{\rho}}})(\boldsymbol{y}) \ne f(\boldsymbol{y})]\right] \\
&= \mathop{\mathbb{E}}_{\widetilde{\boldsymbol{\rho}}}\left[\sum\nolimits_{\ell \in \mathcal{L}(T_{\widetilde{\boldsymbol{\rho}}})} \Pr_{\boldsymbol{x} \sim \mu \times \mathcal{U}^{-1}}\left[\boldsymbol{x} \in X^\ell \wedge L_1(\ell) \ne f(\boldsymbol{x}^1)\right]\right] \\
&\le \mathop{\mathbb{E}}_{\widetilde{\boldsymbol{\rho}}}\left[\sum\nolimits_{\ell \in \mathcal{L}(T_{\widetilde{\boldsymbol{\rho}}})} \Pr_{\boldsymbol{x} \sim \mu \times \mathcal{U}^{-1}}\left[\boldsymbol{x} \in X^\ell \wedge L(\ell) \ne f(\boldsymbol{x})\right]\right] \\
&= \mathop{\mathbb{E}}_{\widetilde{\boldsymbol{\rho}}}\left[\mathrm{err}_{f^k}(T_{\widetilde{\boldsymbol{\rho}}}, \mu \times \mathcal{U}^{-1})\right].
\end{aligned}$$

Observe now that for any $x \in \text{supp}(\mu \times \mathcal{U}^{-1})$, we have $T_{\widetilde{\boldsymbol{\rho}}}(x) = T(x)$. Using the definition of $\mathcal{R}_\mu$ thus yields:

$$\text{err}_f(\mathcal{T}_1, \mu) \le \mathop{\mathbb{E}}_{\widetilde{\boldsymbol{\rho}}\sim\mathcal{R}} \left[\text{err}_{f^k}(T_{\boldsymbol{\rho}^{-1}}, \mu \times \mathcal{U}^{-1})\right] = \text{err}_{f^k}(T, \mu^k). \qquad \square$$

Lemma 23. $\sum_{i\in[k]} \overline{sq}(\mathcal{T}_i, \mu) \le \overline{sq}(\mathcal{T}, \mu^k)$.

Proof. It is sufficient to prove this for the case where $\mathcal{T}$ is a deterministic tree T. We have:

$$\begin{aligned}
\sum_{i\in[k]} \overline{sq}(\mathcal{T}_i, \mu) &= \sum_{i\in[k]} \mathop{\mathbb{E}}_{\boldsymbol{\rho}^i\sim\mathcal{R}_\mu} \left[\overline{q}((\mathcal{T}_i)_{\boldsymbol{\rho}^i}, \mathcal{U}_{\boldsymbol{\rho}^i})\right] \\
&= \sum_{i\in[k]} \mathop{\mathbb{E}}_{\substack{\boldsymbol{\rho}^i\sim\mathcal{R}_\mu \\ \widetilde{\boldsymbol{\rho}}\sim\mathcal{R}}} \left[\overline{q}\left(\left(\mathsf{Ext}_i(T_{\widetilde{\boldsymbol{\rho}}})\right)_{\boldsymbol{\rho}^i}, \mathcal{U}_{\boldsymbol{\rho}^i}\right)\right] \\
&= \sum_{i\in[k]} \mathop{\mathbb{E}}_{\boldsymbol{\rho}\sim\mathcal{R}_\mu^k} \left[\overline{q}\left(\mathsf{Ext}_i(T_{\boldsymbol{\rho}}), \mathcal{U}_{\boldsymbol{\rho}^i}\right)\right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{\rho}\sim\mathcal{R}_\mu^k} \left[\sum_{i\in[k]} \overline{q}\left(\mathsf{Ext}_i(T_{\boldsymbol{\rho}}), \mathcal{U}_{\boldsymbol{\rho}^i}\right)\right].
\end{aligned}$$

where the third equality is due to the fact that the operations of applying Ext and fixing variables are commutable. Let $\rho \in (\{0,\star\}^n)^k$ be a partial fixing and $\ell \in \mathcal{L}(T_\rho)$. The probability that node ℓ is visited during the process $\mathsf{Ext}_i(T_\rho)$ when the input is $\boldsymbol{x}^i \sim \mathcal{U}_{\rho^i}$ is $2^{-d(\ell)}$. Observe that $\mathsf{Ext}_i(T_\rho)$ only makes $d_i(\ell)$ queries to $\boldsymbol{x}^1$ to reach ℓ. As such, we have:

$$\begin{aligned}
\sum_{i\in[k]} \overline{q}\left(\mathsf{Ext}_i(T_{\boldsymbol{\rho}}), \mathcal{U}_{\boldsymbol{\rho}^i}\right) &= \sum_{i\in[k]} \sum_{\ell\in\mathcal{L}(T')} 2^{-d(\ell)} d_i(\ell) \\
&\le \sum_{\ell\in\mathcal{L}(T')} 2^{-d(\ell)} d(\ell) \\
&= \overline{q}(T_\rho, \mathcal{U}_\rho).
\end{aligned}$$

The inequality is due to the fact that $\sum_{i\in[k]} d_i(v) \le d(v)$. This is because $\dim(W_i \cap W_j) = 0$ for each $i \ne j$ and so

$$\sum_{i\in[k]} d_i(v) = \sum_{i\in[k]} \dim(\text{col}(Q^{\prec v}) \cap W_i) \le \dim(\text{col}(Q^{\prec v})) = d(v).$$

To conclude, we have

$$\sum_{i\in[k]} \overline{sq}(\mathcal{T}_i, \mu) = \mathop{\mathbb{E}}_{\boldsymbol{\rho}\sim\mathcal{R}_\mu^k} \left[\sum_{i\in[k]} \overline{q}\left(\mathsf{Ext}_i(T_{\boldsymbol{\rho}}), \mathcal{U}_{\boldsymbol{\rho}^i}\right)\right] \le \mathop{\mathbb{E}}_{\boldsymbol{\rho}\sim\mathcal{R}_\mu^k} \left[\overline{q}(T_{\boldsymbol{\rho}}, \mathcal{U}_{\boldsymbol{\rho}})\right] = \overline{sq}(T, \mu^k). \qquad \square$$

6 Direct sum for $\mathsf{D}^\times$ part III: from S to $\mathsf{D}^\times$

In this section, we show how to convert parity tree of the S_ε model to the more common $\overline{\mathsf{D}}_\varepsilon$ model and prove Theorems 18 and 19. Let us fix for this section a boolean function $f\colon \{0,1\}^n \to \{0,1\}$ together with some 0-biased product distribution μ over $\{0,1\}^n$. Let T be a deterministic parity tree trying to solve f against μ. We begin by establishing an alternative view of the quantity $\overline{sq}(T,\mu)$. For any fixed $x \in \{0,1\}^n$, define the product distribution R_μ^x over $\{0,\star\}^n$ with:

$$\Pr_{\boldsymbol{\rho}\sim\mathcal{R}_\mu^x}[\boldsymbol{\rho}_i = \star] = \begin{cases} \delta_i/(2-\delta_i) & \text{if } x_i = 0 \\ 1 & \text{if } x_i = 1 \end{cases} \qquad \text{where} \quad \delta_i := 2 \cdot \Pr_{\boldsymbol{x}\sim\mu}[\boldsymbol{x}_i = 1] \in [0,1]. \tag{3}$$

Sampling $\boldsymbol{\rho} \sim R_\mu$, $\boldsymbol{x} \sim \mathcal{U}_{\boldsymbol{\rho}}$ and completing $\boldsymbol{x}_j = 0$ for all $\boldsymbol{\rho} = 0$ is equivalent to first sampling $\boldsymbol{x} \sim \mu$ and then some $\boldsymbol{\rho} \sim R_\mu^{\boldsymbol{x}}$. One can therefore see the process of $\overline{sq}(T, \mu)$ as follows:

1. Sample $\boldsymbol{x} \sim \mu$, $\boldsymbol{\rho} \sim R_\mu^{\boldsymbol{x}}$.
2. Run T on $\boldsymbol{x}$.
3. Every time T attempts to make a query, check if $\boldsymbol{\rho}$ simplifies the query: $\boldsymbol{\rho}_i = 0 \implies \boldsymbol{x}_i = 0$.

We describe this alternative view in detail in Algorithm 2. With this new interpretation, we can recast the quantity $\overline{sq}(T, \mu)$ with

$$\overline{sq}(T, \mu) = \mathbb{E}_{\boldsymbol{x} \sim \mu, \boldsymbol{\rho} \sim R_\mu^{\boldsymbol{x}}}[\text{Number of times line 4 is executed in Algorithm 2}]. \tag{4}$$

The idea to convert S_ε algorithms to $\overline{\mathsf{D}}_\varepsilon$ ones is to simulate the process of Algorithm 2 by maintaining an incomplete but consistent view $p \in \{0, \star, ?\}^n$ of ρ. Initially, $p = ?^n$ – i.e. nothing is known about ρ – and we gradually update p based on the queries we get. For instance, if $x_i = 1$, then (3) asserts $\rho_i = \star$. This scheme helps to relate the cost of the converted $\overline{\mathsf{D}}_\varepsilon$ algorithm with $\overline{sq}(T, \mu)$. The description of the converted algorithm is given in Algorithm 3.

Definition 24. *Let $p \in \{0, \star, ?\}^n$ be a fixing. The following are subsets of indices:*

$$S_\star^p = \{j \in [n] : p_j = \star\} \quad S_0^p = \{j \in [n] : p_j = 0\} \quad S_?^p = \{j \in [n] : p_j = ?\} \quad S_{\neq 0}^p = S_\star^p \cup S_?^p$$

We also write $S(p, \star)$ to mean $S_\star^p$ and likewise for other sets.

Let $P^v \subseteq \{0, \star, ?\}^n$ be the set of all possible p that could be at the start of an iteration of Algorithm 3 at node v. We now prove an invariant of Algorithm 3 and then its correctness.

Lemma 25. *For any state $v \in \mathcal{N}(T)$ and $p \in P^v$ that Algorithm 3 could be in at the start of a while iteration (line 3), it holds that:*

$$\operatorname{rank}\Big(Q_{S(p,\neq 0)}^{\prec v}\Big) = \operatorname{rank}\Big(Q_{S(p,\star)}^{\prec v}\Big) = |S(p, \star)|.$$

Proof. We prove the claim by induction on T. The statement is true when v is the root because both $Q^{\prec v}$ and $S_\star^p$ are empty. Let us now assume that the statement is true for some v and $p \in P^v$ and prove that the invariant caries over to the next iteration regardless of the query outcomes and the randomness $\boldsymbol{\eta}$ of the process. If p' is the updated value of p at line 19, this amounts to showing that $\operatorname{rank}(Q_{S(p',\neq 0)}^{\preceq v}) = \operatorname{rank}(Q_{S(p',\star)}^{\preceq v}) = |S(p', \star)|$. We consider three cases.

Case $D^{v,p} = \emptyset$: Then, there is no update for p and $p' = p$. Since $\operatorname{rank}(Q_{S(p,\star)}^{\preceq v}) = \operatorname{rank}(Q_{S(p,\star)+j}^{\preceq v})$ for all $j \in S(p, \neq 0)$, we have $\operatorname{rank}(Q_{S(p,\neq 0)}^{\preceq v}) = \operatorname{rank}(Q_{S(p,\star)}^{\preceq v}) = |S(p, \star)|$, as desired.

Case $D^{v,p} \neq \emptyset$ and $p'_j = 0$ for all $j \in D^{v,p}$: Then, $S_\star^{p'} = S_\star^p$ and $S(p', \neq 0) = S(p, \neq 0) \setminus D^{v,p}$. By definition of $D^{v,p}$, we still have $\operatorname{rank}(Q_{S(p,\star)}^{\preceq v}) = \operatorname{rank}(Q_{S(p,\star)+j}^{\preceq v})$ for all $j \in S(p', \neq 0)$, so $\operatorname{rank}(Q_{S(p',\neq 0)}^{\preceq v}) = \operatorname{rank}(Q_{S(p',\star)}^{\preceq v}) = |S(p', \star)|$.

Case $D^{v,p} \neq \emptyset$ and $p'_j = \star$ for some $j \in D^{v,p}$: Then $S_\star^{p'} = S_\star^p + j$ and it must hold that $\operatorname{rank}(Q_{S(p',\star)}^{\preceq v}) = |S(p', \star)|$. On the other hand,

$$\operatorname{rank}\Big(Q_{S(p',\neq 0)}^{\preceq v}\Big) \leq \operatorname{rank}\Big(Q_{S(p,\neq 0)}^{\prec v}\Big) + 1 = |S(p, \star)| + 1 = |S(p', \star)|.$$

Where the inequality follows from the fact that $S(p', \neq 0) \subseteq S(p, \neq 0)$. Finally, this implies $\operatorname{rank}(Q_{S(p',\neq 0)}^{\preceq v}) = \operatorname{rank}(Q_{S(p',\star)}^{\preceq v}) = |S(p, \star)|$. ∎

Algorithm 2 an alternative view of $\overline{sq}(T, \mu)$

Input: $x \in \{0,1\}^n$, $\rho \in \{0,\star\}^n$
Output: $a \in \{0,1\}$

$v \leftarrow$ root of T
while v is not a leaf **do**
    if $\mathrm{rank}(Q^{\preceq v}_{S(\rho,\star)}) = \mathrm{rank}(Q^{\prec v}_{S(\rho,\star)}) + 1$ **then**
        Query $b^v \leftarrow \langle x, Q^v \rangle$
    else
        Infer $b^v \leftarrow \langle x, Q^v \rangle$ from the fact that $(Q^{\prec v})^T x = b^{\prec v}$ and $x_j = 0$ for all $\rho_j = 0$
    end if
    Move $v \leftarrow \mathrm{child}(v, b^v)$.
end while
return $L(v)$

Algorithm 3 converts an algorithm T for S_ε to $\overline{\mathsf{D}}_\varepsilon$

Input: $x \in \{0,1\}^n$
Output: $a \in \{0,1\}$

$v \leftarrow$ root of T
$p \leftarrow ?^n$
while v is not a leaf **do**
    $D^{v,p} \leftarrow \{j \in [n] : p_j = ? \text{ and } \mathrm{rank}(Q^{\preceq v}_{S(p,\star)+j}) = \mathrm{rank}(Q^{\preceq v}_{S(p,\star)}) + 1\}$
    if $D^{v,p} = \emptyset$ **then**
        Infer $b^v \leftarrow \langle x, Q^v \rangle$ from the fact that $(Q^{\prec v})^T x = b^{\prec v}$ and $x_j = 0$ for all $p_j = 0$
    else
        for $j \in D^{v,p}$ **do**
            Query x_j
            Sample $\boldsymbol{\eta} \sim \mathsf{Ber}(\delta_j/(2-\delta_j))$
            if $x_j = 1$ or $\boldsymbol{\eta} = 1$ **then**
                $p_j \leftarrow \star$
                break
            end if
            $p_j \leftarrow 0$
        end for
        Query $b^v \leftarrow \langle x, Q^v \rangle$
    end if
    Move $v \leftarrow \mathrm{child}(v, b^v)$
end while
return $L(v)$

Lemma 26. *For any* $x \in \{0,1\}^n$, $\Pr_{\boldsymbol{\eta}}[\text{Algorithm 3 outputs } 1] = \mathbb{1}[T(x) = 1]$.

Proof. It is not hard to see that if Algorithm 3 gets the correct value of $\langle x, Q^v \rangle$ at each iteration of the while loop, it perfectly simulates T. Thus, it suffices to show that whenever $D^{v,p} = \emptyset$, the algorithm can compute the value of $\langle x, Q^v \rangle$ from the previous query outcomes. Lemma 25 and its proof implies that if $D^{v,p} = \emptyset$, then $\operatorname{rank}(Q^{\prec v}_{S(p,\neq 0)}) = \operatorname{rank}(Q^{\preceq v}_{S(p,\neq 0)}) = |S(p,\star)|$. Thus $Q^v_{S(p,\neq 0)}$ can be written as a linear combination of column vectors of $Q^{\prec v}_{S(p,\neq 0)}$. Namely, $Q^v_{S(p,\neq 0)} = \sum_{j\in[t]} Q^{v_j}_{S(p,\neq 0)}$, where $v_1, \dots, v_t$ are some ancestors of v. On the other hand, we know that $x_j = p_j = 0$ for all $j \in S^p_0$. Consequently, we have

$$\langle x, Q^v \rangle = \langle x, Q^v \rangle_{S(p,\neq 0)} = \sum\nolimits_{j\in[t]} \langle x, Q^{v_j} \rangle_{S(p,\neq 0)} = \sum\nolimits_{j\in[t]} b^{v_j}.$$

Thus, Algorithm 3 follows the same path of vertices as T, irrespective of the randomness $\boldsymbol{\eta}$. Consequently, its outputs corresponds to the one of T. □

We now turn our attention to the efficiency of Algorithm 3. We shall start with the special case of μ being a constant-bounded distribution. In this particular case, we obtain a lossless conversion. We then turn our attention to general product distributions, for which Algorithm 3 suffers a $\log(n)$ factor. This loss factor is inherent to reducing S_ε to $\overline{\mathsf{D}}$ as Section 8 shows.

6.1 Conversion for constant-bounded distribution

We now prove a strong efficiency result for Algorithm 3 in the special case where μ is λ-bounded (see Definition 12). A proof of our goal (Theorem 19) then follows easily.

Lemma 27. *We have* $\overline{q}(\text{Algorithm 3 on } T, \mu) \le (2/\lambda) \cdot \overline{sq}(T, \mu)$.

Before proving this, we need an alternative view of the randomness used in the for-loop of Algorithm 3 (line 8 to 16). At the start of the process, a random partial fixing $\boldsymbol{\rho} \sim \mathcal{R}^x_\mu$ is generated. The algorithm is then deterministic: whenever some x_j is queried in the for-loop, this is replaced by a query to $\boldsymbol{\rho}_j$. The algorithm updates p_j with $\boldsymbol{\rho}_j$ and exits the loop if $\boldsymbol{\rho}_j = \star$. This process is given in detail in Algorithm 4. Note that as R^x_μ is a product distribution, one can actually implement Algorithm 4 without querying all of x at the start. Indeed, it is enough to query x_j whenever one needs the value of $\boldsymbol{\rho}_j$, similarly to Algorithm 3. This implies that both processes are equivalent.

Suppose one runs Algorithm 4 on $\boldsymbol{x} \sim \mu$ and $\boldsymbol{\rho} \sim R^{\boldsymbol{x}}_\mu$. Fix some state (v,p) the algorithm could be in at the start of the while loop (line 5). We let $\mathcal{X}^{v,p}$ be the distribution of $\boldsymbol{x}$ conditioned on reaching state (v,p). Furthermore, for a fixed $x \in \{0,1\}^n$ and (v,p) reachable with x we let $\mathcal{R}^{v,p,x}$ be the marginal distribution of $\boldsymbol{\rho}$ conditioned on reaching state (v,p) and $\boldsymbol{x} = x$. We now develop explicit formulations for those distributions.

Explicit definition of $\mathcal{X}^{v,p}$: Let $\widehat{\mathcal{X}}^{v,p}$ be the distribution over $\{0,1\}^n$ defined as follows:
1. For all $j \in S^p_0$, fix $\boldsymbol{x}_j = 0$.
2. For all $j \in S^p_?$, sample $\boldsymbol{x}_j \sim \mathsf{Ber}(\delta_j/2)$.
3. Determine $\{\boldsymbol{x}_j : j \in S^p_\star\}$ by solving $\left\{\langle x, Q^u \rangle_{S(p,\star)} = \langle x, Q^u \rangle_{S(p,\neq\star)} + b^u\right\}_{u\in \mathrm{path}(v)}$

Explicit definition of $\mathcal{R}^{v,p,x}$: Let $\widehat{\mathcal{R}}^{p,x}$ be the product distribution over $\{0,\star\}^n$ defined as follows:
1. For all $j \in S^p_?$ such that $x_j = 0$, let $\boldsymbol{\rho}_j = *$ with probability $\delta_j/(2-\delta_j)$ and $\boldsymbol{\rho}_j = 0$ else.
2. For all $j \in S^p_?$ such that $x_j = 1$, fix $\boldsymbol{\rho}_j = \star$.
3. For all $j \in S(p, \neq ?)$, fix $\boldsymbol{\rho}_j = p_j$.

Algorithm 4 an alternative view of Algorithm 3 where the randomness is fixed at the start

Input: $x \in \{0,1\}^n$
Output: $a \in \{0,1\}$

1: $v \leftarrow$ root of T
2: $p \leftarrow ?^n$
3: Sample $\boldsymbol{\rho} \sim R_\mu^x$
4: **while** v is not a leaf **do**
5: $\quad D^{v,p} \leftarrow \{j \in [n] : p_j = ? \wedge \operatorname{rank}(Q^{\preceq v}_{S(p,\star)+j}) = \operatorname{rank}(Q^{\preceq v}_{S(p,\star)} + 1\}$
6: $\quad$ **if** $D^{v,p} = \emptyset$ **then**
7: $\quad\quad$ Infer $b^v \leftarrow \langle x, Q^v \rangle$ from the fact that $(Q^{\prec v})^T x = b^{\prec v}$ and $x_j = 0$ for all $p_j = 0$
8: $\quad$ **else**
9: $\quad\quad$ **for** $j \in D^{v,p}$ **do**
10: $\quad\quad\quad$ $p_j \leftarrow \boldsymbol{\rho}_j$
11: $\quad\quad\quad$ **if** $p_j = \star$ **then**
12: $\quad\quad\quad\quad$ **break**
13: $\quad\quad\quad$ **end if**
14: $\quad\quad$ **end for**
15: $\quad\quad$ Query $b^v \leftarrow \langle x, Q^v \rangle$
16: $\quad$ **end if**
17: $\quad$ Move $v \leftarrow \operatorname{child}(v, b^v)$.
18: **end while**
19: **return** $L(v)$

Claim 28. *For every reachable state (v,p) and $x \in \operatorname{supp}(\mathcal{X}^{v,p})$ in Algorithm 4, we have*

1. $\mathcal{R}^{v,p,x} \equiv \widehat{\mathcal{R}}^{p,x}$;
2. $\mathcal{X}^{v,p} \equiv \widehat{\mathcal{X}}^{v,p}$.

We delay the proof of this technical lemma to Appendix A.4. We can now prove the efficiency of our algorithm for λ-bounded distributions.

Proof of Lemma 27. To relate Algorithm 2 with Algorithm 4, it is helpful to insert the book-keeping of p in Algorithm 2 (lines 5 to 16, without 10) in between lines 2 and 16 of Algorithm 2. This doesn't change the number of queries or guarantees of Algorithm 2 but now both processes share the same state space over (v,p). For $x \in \{0,1\}^n$ and $\rho \in \{0,\star\}^n$, define $A(x,\rho)$ and $B(x,\rho)$ as the number of queries each process makes:

$A(x,\rho) \coloneqq$ number of times line 4 is executed in Algorithm 2 on input (x,ρ);

$B(x,\rho) \coloneqq$ number of times lines 10 and 15 are executed in Algorithm 4 on input (x,ρ).

Using (4), it is thus enough to prove that $\mathbb{E}_{\boldsymbol{x},\boldsymbol{\rho}}[A(\boldsymbol{x},\boldsymbol{\rho})] \geq \Omega(\lambda) \cdot \mathbb{E}_{\boldsymbol{x},\boldsymbol{\rho}}[B(\boldsymbol{x},\boldsymbol{\rho})]$ when $\boldsymbol{x} \sim \mu$ and $\boldsymbol{\rho} \sim R_\mu^{\boldsymbol{x}}$. We have:

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{\rho}}[A(\boldsymbol{x},\boldsymbol{\rho})] = \sum_{(v,p)} \Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\text{state } (v,p) \text{ is reached}] \cdot \Pr_{\substack{\boldsymbol{x}\sim\mathcal{X}^{v,p} \\ \boldsymbol{\rho}\sim\mathcal{R}^{v,p,\boldsymbol{x}}}}\Big[\operatorname{rank}(Q^{\preceq v}_{S(\boldsymbol{\rho},\neq 0)}) = \operatorname{rank}(Q^{\prec v}_{S(\boldsymbol{\rho},\neq 0)}) + 1\Big].$$

As both algorithms follow the same path in the state space, this expectation can be computed with respect to the code of Algorithm 4. Fix some state (v,p) and observe that if there exists some $j \in D^{v,p}$ such that $\boldsymbol{\rho}_j = \star$, then by Lemma 25,

$$\operatorname{rank}(Q^{\preceq v}_{S(\boldsymbol{\rho},\neq)}) = \operatorname{rank}(Q^{\preceq v}_{S(p,\star)+j}) = \operatorname{rank}(Q^{\preceq v}_{S(p,\star)}) + 1 = \operatorname{rank}(Q^{\prec v}_{S(\boldsymbol{\rho},\neq 0)}) + 1.$$

Therefore, for $\boldsymbol{x} \sim \mathcal{X}^{v,p}$ and $\boldsymbol{\rho} \sim \mathcal{R}^{v,p,\boldsymbol{x}}$, we have

$$\begin{aligned}\Pr_{\boldsymbol{x},\boldsymbol{\rho}}\left[\operatorname{rank}(Q^{\preceq v}_{S(\boldsymbol{\rho},\neq 0)}) = \operatorname{rank}(Q^{\prec v}_{S(\boldsymbol{\rho},\neq 0)}) + 1\right] &\geq \Pr_{\boldsymbol{x},\boldsymbol{\rho}}\left[\exists j \in D^{v,p} : \boldsymbol{\rho}_j = \star\right] \\ &= 1 - \Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\forall j \in D^{v,p} : \boldsymbol{\rho}_j = \boldsymbol{x}_j = 0].\end{aligned}$$

The last equality is due to the fact that for all $j \in D^{v,p}$, if $\boldsymbol{\rho}_j = 0$ then $\boldsymbol{x}_j = 0$. Let $D := D^{v,p}$. We can now substitute $\widehat{\mathcal{X}}^{v,p}$ for $\mathcal{X}^{v,p}$ and $\widehat{\mathcal{R}}^{p,\boldsymbol{x}}$ for $\mathcal{R}^{v,p,\boldsymbol{x}}$ using Claim 28:

$$\begin{aligned}\Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\forall j \in D\colon \boldsymbol{\rho}_j = \star \wedge \boldsymbol{x}_j = 0] &= \Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\forall j \in D\colon \boldsymbol{x}_j = 0] \cdot \Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\forall j \in D\colon \boldsymbol{\rho}_j = \star \mid \forall j \in D\colon \boldsymbol{x}_j = 0] \\ &= \prod_{j\in D}(1-\delta_j/2) \cdot \prod_{j \in D} \frac{2-2\delta_j}{2-\delta_j} \\ &= \prod_{j\in D}(1-\delta_j) \\ &\leq (1-\lambda)^{|D|}.\end{aligned}$$

Thus, if $\boldsymbol{x} \sim \mu$ and $\boldsymbol{\rho} \sim R^{\boldsymbol{x}}_{\mu}$, we have

$$\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{\rho}}[A(\boldsymbol{x},\boldsymbol{\rho})] \geq \sum_{(v,p)} \Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\text{state } (v,p) \text{ is reached}] \cdot \left(1 - (1-\lambda)^{|D^{v,p}|}\right).$$

We now bound the expected number of queries made by $\mathcal{T}$. When $D^{v,p} = \emptyset$, $\mathcal{T}$ skips making a query at v. On the other hand, when $D^{v,p} \neq \emptyset$, the algorithm goes over $j \in D^{v,p}$ and stops making queries as soon as it hits some $\rho_j = \star$. This probability is independent for each $j \in D^{v,p}$ and can be computed explicitly using Claim 28. For $\boldsymbol{x} \sim \mathcal{X}^{v,p}$ and $\boldsymbol{\rho} \sim \mathcal{R}^{v,p,x}$:

$$\Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\boldsymbol{\rho}_j = *] = \Pr_{\boldsymbol{x}}[\boldsymbol{x}_j = 0] \cdot \Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\rho_j = \star \mid \boldsymbol{x}_j = 0] + \Pr_{\boldsymbol{x}}[\boldsymbol{x}_j = 1] \cdot \Pr_{\boldsymbol{x}}[\boldsymbol{\rho}_j = \star \mid \boldsymbol{x}_j = 1] = \delta_j \geq \lambda.$$

Therefore, if $\boldsymbol{x} \sim \mu$ and $\boldsymbol{\rho} \sim R^{\boldsymbol{x}}_{\mu}$,

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{\rho}}[B(\boldsymbol{x},\boldsymbol{\rho})] &\leq \sum_{(v,p)} \Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\text{state } (v,p) \text{ is reached}] \cdot \left(\mathbb{1}\left[D^{v,p} \neq \emptyset\right] + \sum_{j=0}^{|D^{v,p}|-1} (1-\lambda)^j\right) \\ &\leq \sum_{(v,p)} \Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\text{state } (v,p) \text{ is reached}] \cdot \left(\mathbb{1}\left[D^{v,p} \neq \emptyset\right] + \left(1-(1-\lambda)^{|D^{v,p}|}\right)/\lambda\right) \\ &\leq \sum_{(v,p)} \Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\text{state } (v,p) \text{ is reached}] \cdot (2/\lambda) \cdot \left(1-(1-\lambda)^{|D^{v,p}|}\right).\end{aligned}$$

□

With this in hand, we can now prove Theorem 19, which we restate below for convenience.

Theorem 19. *For any $f\colon \{0,1\}^n \to \{0,1\}$ and λ-bounded product distribution μ, we have*

$$\overline{\mathsf{D}}_{\varepsilon}(f,\mu) \leq O(1/\lambda) \cdot \mathsf{S}_{\varepsilon}(f,\mu) \quad \forall \varepsilon \geq 0.$$

Proof. Let $\mathcal{T}$ be a randomised parity tree such that $\overline{sq}(\mathcal{T},\mu) = \mathsf{S}_{\varepsilon}(f,\mu)$ and $\operatorname{err}_f(\mathcal{T},\mu) \leq \varepsilon$. Define $\mathcal{T}'$ to be the randomised algorithm obtained by sampling $\boldsymbol{T} \sim \mathcal{T}$ and returning Algorithm 3 applied to $\boldsymbol{T}$. Using Lemma 26, we immediately obtain that $\operatorname{err}(\mathcal{T}',\mu) \leq \varepsilon$. On the other hand:

$$\overline{q}(\mathcal{T}',\mu) = \mathop{\mathbb{E}}_{\boldsymbol{T}}\left[\overline{q}(\text{Algorithm 3 on } \boldsymbol{T},\mu)\right] \leq (2/\lambda) \cdot \mathop{\mathbb{E}}_{\boldsymbol{T}}[\overline{sq}(\boldsymbol{T},\mu)] = (2/\lambda) \cdot \mathsf{S}_{\varepsilon}(f,\mu).$$

Thus, $\overline{\mathsf{D}}_{\varepsilon}(f,\mu) \leq O(1/\lambda) \cdot \mathsf{S}_{\varepsilon}(f,\mu)$, as desired. □

6.2 Conversion for general product distribution

Algorithm 3 is not efficient for arbitrary product distribution since queries can be very biased so that $\prod_{j \in D^{v,p}}(1-\delta_j) = 1 - o(1)$. In such cases, we cannot even afford to pay one query as the corresponding expected increment for $\overline{sq}$ is $o(1)$.

To overcome this obstacle, we introduce the following idea. Run the algorithm as if every query x_j returned 0, i.e. assuming $x_j = \boldsymbol{\rho}_j = 0$ for all $j \in S(p, ?)$ (this is likely to happen for very biased distributions). This generates a list of indices for which we assume $x_j = 0$. Upon reaching a leaf, we check efficiently whether one of those x_j is actually 1. If no such j exists, we're done – at the cost of *no real* queries! On the other hand, if a 1 is found, we backtrack to this state and restart the procedure. Since we've found $x_j = 1$, it must be that $\rho_j = \star$ and the S_ε algorithm has to pay one query there.

The process $\mathsf{BuildList}$ that "runs assuming $x_j = 0$" and produces a list of indices to check is described in Section 6.2. Then, the updated algorithm for converting an S_ε algorithm to a $\overline{\mathsf{D}}_\varepsilon$ one is formulated in Algorithm 5.

How to run line 4? This problem can be formulated as follows. Let $\mathrm{FFO}_n : \{0,1\}^n \to [n] \cup \perp$ be the search problem that asks for the index of the first (running from left to right) '1' in x or $\perp$ if $x = 0^n$. Even though a simple adversary argument shows that one cannot perfectly compute FFO_n by making $< n$ parity queries, a folklore result [FPRU90, Nis93, HR24], proves that there is a randomised protocol making $O(\log n)$ queries that computes FFO_n with some small error.

Lemma 29. *For any $\alpha > 0$, $\mathsf{R}_\alpha(\mathrm{FFO}_n) \le O\big(\log n + \log(1/\alpha)\big)$.*

Proof. This folklore fact is discussed for the parity context in Appendix A.3. ☐

We let $\mathcal{T}'_\gamma$ be the parity tree obtained by running Algorithm 5 with error parameter $\alpha \coloneqq \gamma/n$ on line 4. Given two indices $i, j \in J$, we say $i \prec_J j$ if i appears strictly earlier than j in J, and $i \preceq_J j$ if $i \prec_J j$ or $i = j$. Fix any $x \in \mathrm{supp}(\mathcal{X}^{v,p})$. Let i^* denote the first index i in J such that $x_i = 1$ and suppose that i^* is added to J when $u = u^*$. Observe that if such i^* exists, $x_j = 0$ for all $j \prec_J i^*$. As a consequence, we know that u^* must be reached. Moreover, we can immediately get the values of $\boldsymbol{\rho}_j$ by flipping biased coins for all $j \preceq_J i^*$. Therefore, given i^*, one can perfectly simulate Algorithm 3 by going over J and updating p, until finding the first index $j^* \preceq_J i^*$ such that $\boldsymbol{\rho}_{j^*} = \star$. We are now ready to prove the correctness and efficiency of $\mathcal{T}'_\gamma$.

Lemma 30. *For any fixed $x \in \{0,1\}^n$ $\Pr[\mathcal{T}'_\gamma(x) = 1] \in \mathbb{1}[T(x) = 1] \pm \gamma$.*

Proof. The randomness of $\mathcal{T}'_\gamma$ stems from $\boldsymbol{\eta}$ and the randomness involved in running the FFO algorithm at line 4. To analyse the latter, observe that line 4 is called at most n times and each call fails with probability at most $\alpha = \gamma/n$, hence:

$$d_{\mathrm{TV}}(\mathcal{T}'_0(x), \mathcal{T}'_\gamma(x)) \le \Pr[\text{at least one oracle call at line 4 gives a wrong index}] \le n \cdot (\gamma/n) = \gamma.$$

If no call fails the discussion above implies that Algorithm 5 behaves identically to the earlier Algorithm 3. Hence, correctness of the former (Lemma 26) implies $\Pr[\mathcal{T}'_0(x) = 1] = \mathbb{1}[T(x) = 1]$. ☐

Lemma 31. *We have $\overline{q}(\mathcal{T}'_\gamma, \mu) \le O(\log(n/\gamma)) \cdot \overline{sq}(T, \mu) + \gamma \cdot n$.*

Proof. We first prove that the expected number of iterations of the outer while-loop is low assuming that the algorithm always gets the correct index i^* at line 4. Similar to what we did in Section 6.1, we view the randomness used in the for-loop (line 6 to 15) in Algorithm 5 as a pre-generated partial

Algorithm 5 converts an algorithm for S_ε to $\overline{\mathsf{D}}_\varepsilon$ for general product distributions

Input: $x \in \{0,1\}^n$

Output: $a \in \{0,1\}$

Initialize $v \leftarrow$ root of T, $p \leftarrow ?^n$
while v is not a leaf **do**
 $(J, \ell) \leftarrow \mathsf{BuildList}(v, p)$
 Find the first element $i^* \in J$ with $x_{i^*} = 1$ or set $i^* = \perp$ if none exists
 FOUND $\leftarrow 0$
 for $j \in J$ **do**
  Sample $\boldsymbol{\eta} \sim \mathsf{Ber}(\delta_j/(2-\delta_j))$
  if $j = i^*$ or $\boldsymbol{\eta} = 1$ **then**
   $p_j \leftarrow \star$
   $u \leftarrow w_j$
   FOUND $\leftarrow 1$
   break
  end if
  $p_j \leftarrow 0$
 end for
 if FOUND $= 1$ **then**
  Query $\langle x, Q^u \rangle$ and set b^u as the outcome
  Move $v \leftarrow \mathrm{child}(u, b^u)$
 else
  Update $v \leftarrow \ell$
 end if
end while
return $L(v)$

Algorithm 6 the subroutine $\mathsf{BuildList}$

Input: $v \in \mathcal{N}(T)$, $p \in \{0, \star, ?\}^n$

Output: a list of indices J and a leaf ℓ

Initialize $J \leftarrow []$, $u \leftarrow v$, $p' \leftarrow p$
while u is not a leaf **do**
 $D^{v,p'} \leftarrow \{j \in [n] \colon p'_j = ? \wedge \mathrm{rank}(Q^{\preceq u}_{S(p',\star)+j}) = \mathrm{rank}(Q^{\preceq u}_{S(p',\star)}) + 1\}$
 for $j \in D^{v,p}$ **do**   ▷ in arbitrary order
  $p'_j \leftarrow 0$
  $w_j \leftarrow u$
  $J \leftarrow [J, j]$
 end for
 Infer $b^u \leftarrow \langle x, Q^u \rangle$ assuming $(Q^{\prec v})^T x = b^{\prec v}$ and $x_j = 0$ for all $j \in S(p', 0)$
 Move $u \leftarrow \mathrm{child}(u, b^u)$
end while
return (J, u)

assignment $\boldsymbol{\rho} \sim \mathcal{R}_\mu^x$. Note that the bits of $\boldsymbol{\rho}$ are independent. If i^* is the first index in J with $x_{i^*} = 1$, we know that $x_{i^*} = 1$ and $x_j = 0$ for all $j \prec_J i^*$. At the same time, $\boldsymbol{\rho}_j$ for all $j \preceq_J i$ are revealed to the algorithm one-by-one. As soon as some $\boldsymbol{\rho}_j = \star$ is found, the algorithm quits the loop.

For each $x \in \{0,1\}^n$ and $\rho \in \mathrm{supp}(\mathcal{R}_\mu^x)$, consider running $\mathcal{T}'_\gamma$ on input x using randomness ρ. Define $K(x,\rho)$ as the number of iterations of the outer while loop when $\mathcal{T}'_\gamma$ always gets the correct i^* on line 4. Let p^* denote the final state of p. Since in each iteration except for the last one, we update some p_j as $\star$, we have $K(x,\rho) \le |S(p^*,\star)| + 1$. By Lemma 25, we further have

$$K(x,\rho) \le \mathrm{rank}\left(Q_{S(p^*,\star)}^{\prec\ell(x)}\right) + 1 = \mathrm{rank}\left(Q_{S(p^*,\neq 0)}^{\prec\ell(x)}\right) + 1,$$

where $\ell(x) \in \mathcal{L}(T)$ is the unique leaf at which T terminates given x. Since for all $p_j \neq ?$, $p_j = \rho_j$, we have $S_\star^{p^*} \subseteq S_\star^{\rho} \subseteq S_{\neq 0}^{p^*}$, hence $K(x,\rho) \le \mathrm{rank}(Q_{S(\rho,\star)}^{\prec\ell(x)}) + 1$. On the other hand, by definition we have

$$\overline{sq}(T,\mu) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim\mu \\ \boldsymbol{\rho}\sim\mathcal{R}_\mu^{\boldsymbol{x}}}}\left[\mathrm{rank}\left(Q_{S(\boldsymbol{\rho},\star)}^{\prec\ell(\boldsymbol{x})}\right)\right] \quad \Longrightarrow \quad \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim\mu \\ \boldsymbol{\rho}\sim\mathcal{R}_\mu^{\boldsymbol{x}}}}[K(\boldsymbol{x},\boldsymbol{\rho})] \le \overline{sq}(T,\mu) + 1.$$

Lemma 29 asserts that line line 4 can be implemented to error γ/n using $O(\log(n/\gamma))$ parity queries. Since all those calls are completed successfully with probability $\ge \gamma$, we finally have:

$$\overline{q}(\mathcal{T}'_\gamma,\mu) \le (1-\gamma)\cdot \mathbb{E}_{\substack{\boldsymbol{x}\sim\mu \\ \boldsymbol{\rho}\sim\mathcal{R}_\mu^{x}}}[K(\boldsymbol{x},\boldsymbol{\rho})]\cdot O(\log(n/\gamma)) + \gamma\cdot n \le O(\log(n/\gamma))\cdot \overline{sq}(T,\mu) + \gamma\cdot n. \qquad \square$$

Theorem 18. *For any $f\colon \{0,1\}^n \to \{0,1\}$, product distribution μ, $\gamma \in (0,1/n)$, we have*

$$\overline{\mathsf{D}}_{\varepsilon+\gamma}(f,\mu) \le O\big(\log(n/\gamma)\big)\cdot \mathsf{S}_\varepsilon(f,\mu) \quad \forall \varepsilon \ge 0.$$

Proof. Let $\mathcal{T}$ be a randomised parity tree such that $\overline{sq}(\mathcal{T},\mu) = \mathsf{S}_\varepsilon(f,\mu)$ and $\mathrm{err}_f(\mathcal{T},\mu) \le \varepsilon$. Define $\mathcal{T}^*$ to be the randomised algorithm obtained by sampling $\boldsymbol{T} \sim \mathcal{T}$ and returning the corresponding $\mathcal{T}'_\gamma$. Using Lemma 30, we immediately obtain that $\mathrm{err}(\mathcal{T}^*,\mu) \le \varepsilon + \gamma$. By Lemma 31 and the range of parameters allowed for γ, we get

$$\overline{q}(\mathcal{T}^*,\mu) = \mathbb{E}_{\boldsymbol{T}}\left[\overline{q}(\mathcal{T}'_\gamma,\mu)\right] \le O(\log(n/\gamma))\cdot \mathbb{E}_{\boldsymbol{T}}[\overline{sq}(T,\mu)] = O(\log(n)/\gamma)\cdot \overline{sq}(\mathcal{T},\mu). \qquad \square$$

7 Separations I: disc vs. $\mathsf{D}^\times$

In this section we prove Lemma 3, restated here for convenience.

Lemma 3. *The complexity measures* disc *and* $\mathsf{D}^\times$ *are incomparable:*

1. *There is an n-bit function f such that* $\mathsf{disc}(f) = O(\log n)$ *while* $\mathsf{D}^\times(f) = \Theta(n)$.
2. *There is an n-bit function f such that* $\mathsf{disc}(f) = \Theta(n)$ *while* $\mathsf{D}^\times(f) = O(1)$.

Proof. For the first item, we can consider the n-bit majority function $f \coloneqq \mathrm{MAJ}_n$. It follows from [BGPW15, Theorem 1.2] that $\mathsf{D}^\times(\mathrm{MAJ}_n) \ge \Omega(n)$ where the hard distribution is uniform. By contrast, it is not hard to see that $\mathsf{disc}(\mathrm{MAJ}_n) \le O(\log n)$ (if we query x_i for a random $i \in [n]$, it will have bias $\ge \Omega(1/\sqrt{n})$ toward predicting $\mathrm{MAJ}_n(x)$). We prove the second item by a probabilistic argument. Consider a random function $\boldsymbol{f}$, which is set with $\boldsymbol{f}(x) \sim \mathsf{Ber}(2^{-0.9n})$ independently for each $x \in \{0,1\}^n$. In Claim 32, we show that $\mathsf{disc}(\boldsymbol{f}) = \Theta(n)$ and in Claim 33 that $\mathsf{D}^\times(\boldsymbol{f}) = O(1)$ with high probability. $\square$

Claim 32. *With probability* $1-2^{-2^{\Omega(n)}}$, $\mathsf{disc}(\boldsymbol{f}) \geq 0.01n$.

Proof. For each non-constant function $f : \{0,1\}^n \to \{0,1\}$, we define the "hard" distribution μ_f as

$$\mu_f(x) := \begin{cases} 1/(2|f^{-1}(0)|) & \text{if } f(x) = 0 \\ 1/(2|f^{-1}(1)|) & \text{if } f(x) = 1 \end{cases}.$$

To prove the claim, it suffices to show $\Pr_{\boldsymbol{f}}[\mathsf{disc}(\boldsymbol{f}, \mu_{\boldsymbol{f}}) \geq 0.01n] \geq 1 - 2^{-2^{\Omega(n)}}$. Using Lemma 8, this can be further simplified to prove:

$$\Pr_{\boldsymbol{f}}\left[\max_{S \in \mathcal{O}_n} \mathsf{bias}(\boldsymbol{f}, \mu_{\boldsymbol{f}}, S) \leq 2^{-0.01n-1}\right] \geq 1 - 2^{-2^{\Omega(n)}}.$$

To that end, fix any $S \in \mathcal{O}^n$, note that $|S| = |\{0,1\} \setminus S| = 2^{n-1}$ and observe that by a Chernoff bound,

$$\begin{aligned} \Pr_{\boldsymbol{f}}\left[|\mu(\boldsymbol{f}^{-1}(1) \cap S) - 1/4| \geq 2^{-0.02n}\right] &\leq \Pr_{\boldsymbol{f}}\left[|\boldsymbol{f}^{-1}(1)| < 2^{0.1n-1}\right] \\ &\quad + \Pr_{\boldsymbol{f}}\left[||\boldsymbol{f}^{-1}(1) \cap S| - 2^{0.1n-1}| > 2^{0.07n}\right] \\ &\quad + \Pr_{\boldsymbol{f}}\left[||\boldsymbol{f}^{-1}(1) \setminus S| - 2^{0.1n-1}| > 2^{0.07n}\right] \\ &\leq 3e^{-2^{0.03n}}. \end{aligned}$$

Using a similar argument, we can also show $\Pr_{\boldsymbol{f}}[|\mu(\boldsymbol{f}^{-1}(0) \cap S) - 1/4| \geq 2^{-0.02n}] \leq 3e^{-2^{0.03n}}$. By definition, $\mathsf{bias}(\boldsymbol{f}, \mu_{\boldsymbol{f}}, S) = |\mu(\boldsymbol{f}^{-1}(0) \cap S) - |\mu(\boldsymbol{f}^{-1}(1) \cap S)|$, we thus have $\Pr[\mathsf{bias}(\boldsymbol{f}, \mu_{\boldsymbol{f}}, S) \geq 2^{-0.01n-1}] \leq 6e^{-2^{0.03n}}$. Finally, observe that $|\mathcal{O}_n| \leq 2^n$ and so using a union bound,

$$\begin{aligned} \Pr_{\boldsymbol{f}}[\mathsf{disc}(\boldsymbol{f}) \geq 0.01n] &\geq \Pr_{\boldsymbol{f}}[\max_{S \in \mathcal{O}_n} \mathsf{bias}(\boldsymbol{f}, \mu_{\boldsymbol{f}}, S) \leq 2^{-0.01n-1}] \\ &\geq 1 - 2^n \Pr[\mathsf{bias}(\boldsymbol{f}, \mu_{\boldsymbol{f}}, S) \geq 2^{-0.01n-1}] \\ &\geq 1 - 2^{-2^{\Omega(n)}}. \end{aligned}$$

□

Claim 33. *With probability* $1 - 2^{-\Omega(n)}$, $\mathsf{D}^{\times}(f) \leq 20000$.

Proof. Let $\mathcal{D}^{\times} := \{\mathsf{Ber}(p_1, \ldots, p_n) \mid p_1 \ldots, p_n \in [0, 1/2]\}$ denote the set of 0-biased product distributions, where $\mathsf{Ber}(p_1, \ldots, p_n) := \mathsf{Ber}(p_1) \times \cdots \times \mathsf{Ber}(p_n)$. As observed in Section 4, it suffices to show $\Pr_{\boldsymbol{f}}[\max_{\mu \in \mathcal{D}^{\times}} \mathsf{D}_{1/3}(\boldsymbol{f}, \mu) \leq 20000] \geq 1 - 2^{-\Omega(n)}$.

As a first attempt, one might want to prove that $\mathsf{D}_{1/3}(f, \mu) = O(1)$ with sufficiently high probability for any fixed μ and then apply union bound over all $\mu \in \mathcal{D}^{\times}$. However, this cannot be done directly since $\mathcal{D}^{\times}$ is infinite. Luckily, we can circumvent this barrier by discretizing $\mathcal{D}^{\times}$. Let us define $\mathcal{D}^{\times}_{\mathbb{Z}} := \{\mathsf{Ber}(a_1/10n, \ldots, a_n/10n) \mid a_1, \ldots, a_n \in \{0, \ldots, 5n\}\}$. For every $\mu = \mathsf{Ber}(p_1, \ldots, p_n) \in \mathcal{D}^{\times}_{\mathbb{Z}}$ and $f : \{0,1\}^{\to}\{0,1\}$, consider the following two cases:

- If $\sum_i p_i \geq 10$, then $M := \max_{x \in \{0,1\}^n} \mu(x) \leq e^{-\sum_i p_i} \leq 1000 \sum_i p_i$. Observe that

$$\Pr_{\boldsymbol{f}}\left[\sum_{x \in \{0,1\}} f(x)\mu(x) \geq 1/5\right] \leq 2^M \cdot (2^{-0.9n})^{M/5} \leq 2^{-150 \sum_i p_i n},$$

  thus $\Pr_{\boldsymbol{f}}[\mathsf{D}_{1/4}(\boldsymbol{f}, \mu) = 0] \geq 1 - 2^{-150 \sum_i p_i n}$.

- Otherwise, we devise the following protocol: Sort $\mu(x_1) \geq \cdots \geq \mu(x_{2^n})$. Pick the top 1000 inputs $X = \{x_1, \ldots, x_{1000}\}$, then we check if our input x is in X. If yes, we output $f(x)$, otherwise we output 0. Formally, we define the function $g : \{0,1\}^n \to \{0,1\}$ where

$$g(x) \coloneqq \begin{cases} f(x) & \text{if } x \in X \\ 0 & \text{if } x \notin X \end{cases}.$$

Since testing whether $x = x_i$ can be done with m queries with success probability $1 - 2^{-m}$, by choosing $m = 20$ and running the testing for every $i \in [1000]$, one can show $\mathcal{R}_{0.01}(g) \leq 20000$. It remains to prove that $\Pr_{\boldsymbol{f}}[\boldsymbol{f}(x) = \boldsymbol{g}(x)] \geq 4/5$ with high probability. Observe that for each $x \notin X$, $\mu(x) \leq 1/1000$. Therefore:

$$\Pr_{\boldsymbol{f}}\left[\sum\nolimits_{x \notin X} [\mu(x)\boldsymbol{f}(x)] \leq 1/5\right] \geq 1 - 2^{1000} \cdot (2^{-0.9n})^{200} \geq 1 - 2^{-150n}.$$

For those $\boldsymbol{f}$, we have $\Pr_{\boldsymbol{f}}[\boldsymbol{f}(x) = \boldsymbol{g}(x)] \geq 4/5$, which implies that $\mathsf{D}_{0.22}(\boldsymbol{f}, \mu) \leq 20000$.

By union bound over $\mu \in \mathcal{D}^{\times}_{\mathbb{Z}}$, we can deduce that

$$\begin{aligned}
\Pr_{\boldsymbol{f}}\left[\max_{\mu \in \mathcal{D}^{\times}_{\mathbb{Z}}} \mathsf{D}_{0.22}(\boldsymbol{f}, \mu) > 20000\right] &\leq \sum\nolimits_{\mu \in \mathcal{D}^{\times}_{\mathbb{Z}}} \Pr_{\boldsymbol{f}}[\mathsf{D}_{0.22}(\boldsymbol{f}, \mu) > 20000] \\
&\leq \sum\nolimits_{a_1=0}^{5n} \cdots \sum\nolimits_{a_n=0}^{5n} \mathbb{1}\left[\sum\nolimits_i a_i \geq 100n\right] \cdot e^{-150\sum_i a_i} \\
&\quad + \sum\nolimits_{a_1=0}^{5n} \cdots \sum\nolimits_{a_n=0}^{5n} \mathbb{1}\left[\sum\nolimits_i a_i < 100n\right] \cdot 2^{-150n} \\
&\leq \sum\nolimits_{a_1=0}^{5n} \cdots \sum\nolimits_{a_n=0}^{5n} e^{-100(a_1+5)} \cdots e^{-100(a_n+5)} + 2^{101n} \cdot 2^{-150n} \\
&\leq \left(\sum\nolimits_{a_1=0}^{5n} e^{-100(a_1+5)}\right)^n + 2^{-49n} \\
&\leq 2^{-\Omega(n)}.
\end{aligned}$$

Consider now any product distribution $\mu = \mathsf{Ber}(p_1, \ldots, p_n) \in \mathcal{D}^{\times}$, define its rounded version $\lceil \mu \rceil$:

$$\lceil \mu \rceil \coloneqq \left(\mathsf{Ber}\left(\frac{\lceil 10n \cdot p_1 \rceil}{10n}\right), \ldots, \mathsf{Ber}\left(\frac{\lceil 10n \cdot p_n \rceil}{10n}\right)\right) \in \mathcal{D}^{\times}_{\mathbb{Z}}.$$

Observe that $d_{\mathrm{TV}}(\mu, \lceil \mu \rceil) \leq 1 - (1 - 1/10n)^n \leq 1 - 1/e^{-1/10} < 0.1$, thus we have $\mathrm{err}_f(T, \lceil \mu \rceil) \leq \mathrm{err}_f(T, \mu) + 0.1$ for any parity tree T and $f : \{0,1\}^n \to \{0,1\}$. Together with the string of inequalities developed above, we conclude that with probability at least $1 - 2^{-\Omega(n)}$,

$$\max\nolimits_{\mu \in \mathcal{D}^{\times}} \mathsf{D}_{1/3}(\boldsymbol{f}, \mu) \leq \max\nolimits_{\mu \in \mathcal{D}^{\times}_{\mathbb{Z}}} \mathsf{D}_{1/3-0.1}(\boldsymbol{f}, \mu) \leq \max\nolimits_{\mu \in \mathcal{D}^{\times}_{\mathbb{Z}}} \mathsf{D}_{0.22}(\boldsymbol{f}, \mu) \leq 20000. \qquad \square$$

8 Separations II: S vs. $\mathsf{D}^{\times}$

The goal of this section is to provide the following example of a function.

Theorem 34. *There exists a function $f\colon \{0,1\}^n \to \{0,1\}$ and a product distribution μ such that $\mathsf{D}^{\times}(f) = \Theta(\mathsf{disc}(f, \mu)) = \Theta(\log n)$ and $\mathsf{S}_0(f, \mu) = \Theta(1)$.*

Recall that by Theorem 18, this is the largest possible gap between S and $\mathsf{D}^{\times}$. To prove the separation, we use the function $\mathrm{FPE}\colon \{0,1\}^{2n} \to \{0,1\}$ which takes two inputs $x, y \in \{0,1\}^n$ and returns the value y_i associated with the location i of the first '1' in x. More precisely, we let $\mathrm{FO}(x) \in [n]$ be the location (from left to right) of the first '1' in x and $\mathrm{FO}(x) = 1$ if $x = 0^n$ and let $\mathrm{FPE}(x, y) = y_{\mathrm{FO}(x)}$. We choose as hard distribution the product distribution $\mu \coloneqq \mathcal{X} \times \mathcal{Y}$ where for each $i \in [n]$:

$$\mathcal{X}_i \sim \mathsf{Ber}(1/\sqrt{n}) \quad \text{and} \quad \mathcal{Y}_i \sim \mathsf{Ber}(1/2).$$

Let us note that the choice of $1/\sqrt{n}$ in the distribution $\mathcal{X}$ is arbitrary: any $p = n^a$ for constant $a \in (-1, 0)$ is enough to guarantee that $x \neq 0^n$ with high probability and get the $\Omega(\log n)$ lower bound.

Proof of Theorem 34. We first prove that $\mathsf{S}_0(\mathrm{FPE}, \mu) = \Theta(1)$. Consider the following simple brute-force query algorithm T that computes f: Query the bits of x one-by-one from left to right, until finding the first index i such that $x_i = 1$. Then query y_i and return y_i if such i exists. Otherwise ($x = 0^n$), simply return 1.

Observe that $\mathrm{err}_{\mathrm{FPE}}(T, \mu) = 0$. Thus we only need to show $\overline{sq}(f, \mu) = \Theta(1)$. Let $X_i \coloneqq \{x \mid x_i = 1, x_j = 0, \forall j < i\}$ denote the set of $x \in \{0,1\}^n$ for which $\mathrm{FO}(x) = i$. Note that $\{0,1\}^n = X_1 \sqcup \cdots \sqcup X_n \sqcup \{0^n\}$ forms a partition of $\{0,1\}^n$. By the definition of μ, we have $\mu(X_i) = (1 - 1/\sqrt{n})^{i-1}/\sqrt{n}$. For all $x \in X_i$, T queries the same set of variables $\{x_1, \ldots, x_{i-1}, x_i, y_i\}$ on x. Moreover, sample $\boldsymbol{\rho} \sim \mathcal{R}^x_\mu$ and for each $1 \leq j < i$, since $x_j = 0$, we have that $\Pr[\boldsymbol{\rho}_j = \star] = 1/(\sqrt{n} - 1)$. Therefore,

$$h(x) \coloneqq \mathbb{E}_{\boldsymbol{\rho} \sim \mathcal{R}^x_\mu}[q(T_{\boldsymbol{\rho}}, x)] \leq \frac{i-1}{\sqrt{n} - 1} + 2.$$

We conclude that

$$\begin{aligned}
\overline{sq}(T, \mu) &= \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu}[h(\boldsymbol{x})] \\
&\leq \sum\nolimits_{i=1}^{n} \mu(X_i) \cdot \mathbb{E}_{\boldsymbol{x} \sim \mu_{X_i}}[h(\boldsymbol{x})] + (n+1) \cdot \mu(0^n) \\
&\leq \frac{1}{n - \sqrt{n}} \cdot \sum\nolimits_{i=1}^{n} (i-1)(1 - 1/\sqrt{n})^{i-1} + n \cdot (1 - 1/\sqrt{n})^n + 2 \\
&< \frac{2}{n} \cdot \sum\nolimits_{i=0}^{\infty} i(1 - 1/\sqrt{n})^i + 3 \\
&= \Theta(1).
\end{aligned}$$

Let us now turn our attention to $\mathsf{disc}(\mathrm{FPE}, \mu)$. The lower-bound $\mathsf{disc}(\mathrm{FPE}, \mu) \geq \Omega(\log n)$ is covered in Claim 35. The upper bound $\mathsf{disc}(\mathrm{FPE}, \mu) \leq O(\log n)$ is a direct consequence of $\mathsf{bias}(\mathrm{FPE}, \mu, S) \geq n^{-1/2}$ for $S = \{(x, y) \in \{0,1\}^n : y_1 = 1\}$. More interestingly, one can actually show the stronger statement $\mathsf{D}_{1/3}(f, \mu) \leq O(\log n)$. Indeed, $\boldsymbol{x} \sim \mathcal{X}$ has exactly one '1' in the first $\lceil \sqrt{n} \rceil$ bits with probability $\geq e^{-1.01} \geq 1/3$ for n large enough. In that case, a simple binary search amongst the first $\lceil \sqrt{n} \rceil$ bits of x using parity queries is enough to find that location and return corresponding bit of y. ∎

Claim 35. $\mathsf{disc}(\mathrm{FPE}, \mu) \geq \Omega(\log n)$

Proof. Using the characterisation of the bias with codimension-1 subspace Lemma 8, it is enough to show:

$$\max\nolimits_{S \in \mathcal{O}^n} \mathsf{bias}(\mathrm{FPE}, \mu, S) \leq n^{-1/3}.$$

Fix an affine space $S^\star$ of codimension 1 that maximize the above expression, i.e. some $\alpha, \beta \in \{0,1\}^n$ and $\gamma \in \{0,1\}$ such that $S^\star = \{(x,y) \in \{0,1\}^{2n} : \alpha \cdot x + \beta \cdot y = \gamma\}$. To simplify notation, we assume in what follows that $\gamma = 0$ but the proof is similar for the case $\gamma = 1$. Let us partition S in two sets:

$$S^0 \coloneqq \{(x,y) \in \{0,1\}^{2n} : \alpha \cdot x = 0 \text{ and } \beta \cdot y = 0\};$$
$$S^1 \coloneqq \{(x,y) \in \{0,1\}^{2n} : \alpha \cdot x = 1 \text{ and } \beta \cdot y = 1\}.$$

We have:

$$\max_{S \in \mathcal{O}^n} \mathsf{bias}(\mathrm{FPE}, \mu, S) = \mathsf{bias}(\mathrm{FPE}, \mu, S^\star) \leq \mathsf{bias}(\mathrm{FPE}, \mu, S^0) + \mathsf{bias}(\mathrm{FPE}, \mu, S^1).$$

Let us suppose without loss of generality that $\mathsf{bias}(\mathrm{FPE}, \mu, S^0) \geq \mathsf{bias}(\mathrm{FPE}, \mu, S^1)$ so that it is enough to show $\mathsf{bias}(\mathrm{FPE}, \mu, S^0) \leq 2n^{-1/2}$. Note that if $\Pr_{\boldsymbol{x},\boldsymbol{y} \sim \mu}[(\boldsymbol{x},\boldsymbol{y}) \in S^0] = 0$, we're done. If not, we can re-express the bias in the language of probability:

$$\begin{aligned}
\mathsf{bias}(\mathrm{FPE}, \mu, S^0) &= \left| \sum\nolimits_{(x,y) \in S^0} (-1)^{\mathrm{FPE}(x)} \mu(x) \right| \\
&= \left| \sum\nolimits_{b \in \{0,1\}} (-1)^b \cdot \Pr_{\boldsymbol{x},\boldsymbol{y}} \left[\mathrm{FPE}(x) = b \wedge (\boldsymbol{x},\boldsymbol{y}) \in S^0 \right] \right| \\
&= \Pr_{\boldsymbol{x},\boldsymbol{y}} \left[(\boldsymbol{x},\boldsymbol{y}) \in S^0 \right] \cdot \left| \sum\nolimits_{b \in \{0,1\}} (-1)^b \cdot \Pr_{\boldsymbol{x},\boldsymbol{y}} \left[\mathrm{FPE}(\boldsymbol{x}) = b \,|\, (\boldsymbol{x},\boldsymbol{y}) \in S^0 \right] \right|.
\end{aligned}$$

Let us denote the quantity within the absolute value by Φ. Observe that S^0 can be conveniently decomposed as $S^0 = S^X \times S^Y$ where $S^X \coloneqq \{x \in \{0,1\}^n : \alpha \cdot x = 0\}$ and $S^Y \coloneqq \{y \in \{0,1\}^n : \beta \cdot y = 0\}$. With this, we have:

$$\begin{aligned}
\Phi &= \sum\nolimits_{b \in \{0,1\}} (-1)^b \cdot \Pr_{\boldsymbol{x},\boldsymbol{y}} \left[\mathrm{FPE}(\boldsymbol{x},\boldsymbol{y}) = b \,|\, (\boldsymbol{x},\boldsymbol{y}) \in S^0 \right] \\
&= \sum_{i \in [n]} \sum_{b \in \{0,1\}} (-1)^b \cdot \Pr_{\boldsymbol{x},\boldsymbol{y}} \left[\mathrm{FO}(x) = i \,|\, (\boldsymbol{x},\boldsymbol{y}) \in S^0 \right] \Pr_{\boldsymbol{x},\boldsymbol{y}} \left[\mathrm{FPE}(\boldsymbol{x},\boldsymbol{y}) = b \,|\, (\boldsymbol{x},\boldsymbol{y}) \in S^0 \wedge \mathrm{FO}(\boldsymbol{x}) = i \right] \\
&= \sum_{i \in [n]} \Pr_{\boldsymbol{x} \sim \mathcal{X}} \left[\mathrm{FO}(\boldsymbol{x}) = i \,|\, \boldsymbol{x} \in S^X \right] \cdot \sum_{b \in \{0,1\}} (-1)^b \cdot \underbrace{\Pr_{\boldsymbol{y} \sim \mathcal{Y}} \left[\boldsymbol{y}_i = b \,|\, \boldsymbol{y} \in S^Y \right]}_{\coloneqq p_i^b}.
\end{aligned}$$

Recall that S^Y is a codimension-1 space and $\mathcal{Y}$ is the uniform distribution over $\{0,1\}^n$. Thus, if $|\alpha|$ (the number of non-zero entries in α) is zero or ≥ 2, it must be that $p_i^b = 1/2$ for all $i \in [n]$ and $b \in \{0,1\}$. In that case, the claim is proven because $\Phi = 0$ and so $\mathsf{bias}(\mathrm{FPE}, \mu, S^0) = 0$. We can thus assume that $|\alpha| = 1$ and fix $i^* \in [n]$ to be the unique coordinate such that $\alpha_{i^*} = 1$. Now, observe that $p_i^b = 1/2$ for all $i \neq i^*$ and $b \in \{0,1\}^n$, $p_{i^*}^0 = 1$ and $p_{i^*}^1 = 0$ so that:

$$\Phi = \sum_{i \in [n]} \Pr_{\boldsymbol{x} \sim \mathcal{X}} \left[\mathrm{FO}(\boldsymbol{x}) = i \,|\, \boldsymbol{x} \in S^X \right] \cdot (p_i^0 - p_i^1) = \Pr_{\boldsymbol{x} \sim \mathcal{X}} \left[\mathrm{FO}(\boldsymbol{x}) = i^* \,|\, \boldsymbol{x} \in S^X \right].$$

Finally, we use the fact that the event $\mathrm{FO}(\boldsymbol{x}) = i^*$ with $\boldsymbol{x} \sim \mathcal{X}$ is unlikely to happen if S^X has large mass under $\mathcal{X}$. In any case, the probability is maximized for $i^* = 1$ and hence:

$$\begin{aligned}
\mathsf{bias}(\mathrm{FPE}, \mu, S^0) &= \Pr_{\boldsymbol{x} \sim \mathcal{X}} \left[\boldsymbol{x} \in S^X \right] \cdot \Pr_{\boldsymbol{y} \sim \mathcal{Y}} \left[\boldsymbol{y} \in S^Y \right] \cdot |\Phi| \\
&\leq \Pr_{\boldsymbol{x}} \left[\mathrm{FO}(\boldsymbol{x}) = i^* \wedge \boldsymbol{x} \in S^X \right] \\
&\leq \Pr_{\boldsymbol{x}} \left[\mathrm{FO}(\boldsymbol{x}) = 1 \right].
\end{aligned}$$

The event $\mathrm{FO}(\boldsymbol{x}) = 1$ can happen because $\boldsymbol{x}_1 = 1$ or $\boldsymbol{x} = 0^n$, thus we bound the bias with

$$\Pr_{\boldsymbol{x} \sim \mathcal{X}} \left[\mathrm{FO}(\boldsymbol{x}) = 1 \right] \leq \Pr_{\boldsymbol{x}} \left[\boldsymbol{x}_1 = 1 \right] + \Pr_{\boldsymbol{x}} \left[\boldsymbol{x} = 0^n \right] \leq n^{-1/2} + e^{-\sqrt{n}} \leq 2n^{-1/2}. \qquad \square$$

A Appendix

A.1 Direct sums for D

In this appendix, we prove that the best known direct sum results in the context of deterministic communication complexity can be obtained in the parity decision tree setting. We restate our theorem for convenience below.

Theorem 4. *For any function f and $k \geq 1$,*

1. $\mathsf{D}(f^k) \geq k \cdot \mathsf{D}(f)^{1/2}$,
2. $\mathsf{D}(f^k) \geq k \cdot \mathsf{D}(f)/\log \operatorname{spar}(f)$.

Let us first introduce a couple of definitions. Fix a function $f\colon \{0,1\}^n \to \{0,1\}$. A parity certificate for $f(x)$ is an affine space $S \subseteq \{0,1\}^n$ such that $x \in S$ and for any $x' \in S$, $f(x) = f(x')$. Similarly to the classical case, the parity certificate complexity $\mathsf{C}(f)$ is the smallest codimension of a space that certifies the value $f(x)$ – where the hardest possible $x \in \{0,1\}^n$ is taken. We also define $\operatorname{spar}(f) := \|\hat{f}\|_0 = |\{z \mid \hat{f}(z) \neq 0\}|$ for the number of non-zero Fourier coefficients of f. To prove Theorem 4, it is enough to prove a direct sum for parity certificate complexity and employ the following two results:

1. $\mathsf{C}(f) \geq \mathsf{D}(f)^{1/2}$ [ZS10]
2. $\mathsf{C}(f) \geq \mathsf{D}(f)/\log \operatorname{spar}(f)$ [TWXZ13]

Lemma 36. *For any $f\colon \{0,1\}^n \to \{0,1\}$ and $k \geq 1$, $\mathsf{C}(f^k) \geq k \cdot \mathsf{C}(f)$.*

Proof of Lemma 36. Fix an input $x \in \{0,1\}^n$ attaining $d := \mathsf{C}(f)$ and suppose towards contradiction that $\mathsf{C}(f^k) < dk$. This implies in particular that there exists an affine space $S \subseteq (\{0,1\}^n)^k$ described by $m < dk$ equations $Q^T x = b$ (where $Q \in \{0,1\}^{n \times m}, b \in \{0,1\}^m$) that certifies the value of the input $y \in (\{0,1\}^n)^k$ which is composed of k copies of x. Define d_i for $i \in [k]$ with:

$$d_i := \dim(\operatorname{col}(Q) \cap W_i) \quad W_i := \{w \in (\{0,1\}^n)^k : w^j = 0^n \iff j \neq i\}.$$

Observe that $\sum_{i \in [k]} d_i \leq m < dk$ and as such there must be some i^* with $d_{i^*} < k$. Fix for simplicity $i^* = 1$. Using Gaussian elimination, one can re-express $S = S_1 \cap S_2$ where

1. the constraints in S_1 are exclusively on bits of the first copy and
2. any constraint in S_2 has at least one bit of a copy other than the first.

Since S_1 is about the first copy only, it can be identified with a single-copy affine space $S^* \subseteq \{0,1\}^n$ where $\operatorname{codim}(S^*) = d_1 < k$ in a natural way. Observe that $x \in S^*$ as $y \in S$. Because the codimension of S^* is strictly less than k, there must be some $x' \in S^*$ with $f(x) \neq f(x')$. Note that fixing $x^1 := x'$ leaves the system of linear constraints S_2 feasible and as such there exists $x^2, \ldots, x^k \in \{0,1\}^n$ such that $y' := (x', x^2, \ldots, x^k) \in S$: a contradiction since $f(y) \neq f(y')$. □

A.2 Direct sum for distribution-free discrepancy

Theorem 37. *For every function $f\colon \{0,1\}^n \to \{0,1\}$ and $k \geq 1$,*

$$k \cdot \mathsf{disc}(f) + 1 \geq \mathsf{disc}(f^{\oplus k}) \geq k \cdot \big(\mathsf{disc}(f) - 1\big).$$

Proof. The lower bound is a simple consequence of Lemma 7 by fixing μ to be a distribution such that $\mathsf{disc}(f) = \mathsf{disc}(f, \mu)$ and observing that $\mathsf{disc}(f^{\oplus k}) \geq \mathsf{disc}(f^{\oplus k}, \mu^k)$. The other direction is more interesting as it says that the hardest distribution for $f^{\oplus k}$ is basically k products of the hardest distribution for a single copy f. Let $\|\widehat{f}\|_\infty^* := \min_\mu \|\widehat{F_\mu}\|_\infty$ where μ ranges over all distributions. Using Lemma 8, we obtain the following relation between $\mathsf{disc}(f)$ and $\|\widehat{f}\|_\infty^*$:

$$-\log \|\widehat{f}\|_\infty^* + 1 \geq \mathsf{disc}(f) \geq -\log \|\widehat{f}\|_\infty^*.$$

Therefore, to prove the upper-bound, it is enough to show a perfect direct product for $\|\widehat{f}\|_\infty^*$ and apply it k time. To this end, fix some other function $g : \{0,1\}^n \to \{0,1\}$ and let us show that

$$\|\widehat{f \oplus g}\|_\infty^* \geq \|\widehat{f}\|_\infty^* \cdot \|\widehat{g}\|_\infty^*.$$

Where we recall that $f \oplus g : \{0,1\}^{2n} \to \{0,1\}$. We can write $\|\widehat{f}\|_\infty^*$ as the value of the following linear program where the variables describe a distribution μ:

$$\begin{aligned} \text{min.} \quad & c \\ \text{s.t.} \quad & \Big| \textstyle\sum_{x \in \{0,1\}^n} (-1)^{f(x)} \cdot \mu_x \cdot (-1)^{\langle x,z \rangle} \Big| \leq c \qquad && \forall z \in \{0,1\}^n \\ & \textstyle\sum_{x \in \{0,1\}^n} \mu_x = 1 \\ & \mu_x \geq 0 && \forall x \in \{0,1\}^n \end{aligned} \tag{5}$$

The dual of (5) is:

$$\begin{aligned} \text{max.} \quad & d \\ \text{s.t.} \quad & \textstyle\sum_{z \in \{0,1\}^n} (-1)^{f(x)} \cdot \beta_z \cdot (-1)^{\langle x,z \rangle} \geq d \quad && \forall x \in \{0,1\}^n \\ & \textstyle\sum_{z \in \{0,1\}^n} |\beta_z| = 1 \end{aligned} \tag{6}$$

Let (β^f, d^f) and (β^g, d^g) be the optimal feasible solutions to (6) with respect to f and g. By the strong duality theorem, it holds that $\|\widehat{f}\|_\infty^* = d^f$ and $\|\widehat{g}\|_\infty^* = d^g$. We now extract a feasible solution for (6) with respect to the function $f \oplus g$. Let $\beta \in \{0,1\}^{2n}$ be defined with $\beta_{(z_1,z_2)} = \beta^f_{z_1} \cdot \beta^g_{z_2}$ and observe that $(\beta, d^f \cdot d^g)$ is a feasible solution for the dual of $\|\widehat{f \oplus g}\|_\infty^*$. By applying the strong duality theorem again, we have $\|\widehat{f \oplus g}\|_\infty^* \geq d^f \cdot d^g = \|\widehat{f}\|_\infty^* \cdot \|\widehat{g}\|_\infty^*$, as desired. ☐

A.3 Some facts about parity decision trees

Yao's minimax principle is a powerful technique to analyse randomised algorithms – we adapt here the statement to parity trees, but the proof is exactly the same as the original one [Yao77].

Lemma 38. *For any $f\colon \{0,1\}^n \to \{0,1\}$ and distribution μ over $\{0,1\}^n$, $\mathsf{R}_\varepsilon(f) \geq \mathsf{D}_\varepsilon(f, \mu)$.*

The following is a folklore fact relating randomised parity tree complexity and discrepancy [Yao83, BFS86] which we re-prove in the parity context.

Lemma 39. $\mathsf{D}_\varepsilon(f, \mu) \geq \mathsf{disc}(f, \mu) + \log(1 - 2\varepsilon)$ *for any $\varepsilon \in [0, 1/2)$.*

Proof. Fix a parity decision tree T of depth $d \coloneqq \mathsf{D}_\varepsilon(f,\mu)$ which makes error $\mathrm{err}_f(T,\mu) \le \varepsilon$, note that

$$\begin{aligned}1-2\varepsilon &\le \Pr_{\boldsymbol{x}\sim\mu}[T(\boldsymbol{x}) = f(\boldsymbol{x})] - \Pr_{\boldsymbol{x}\sim\mu}[T(\boldsymbol{x}) \neq f(\boldsymbol{x})] \\ &= \sum\nolimits_{S\in\mathcal{L}} \Pr_{\boldsymbol{x}\sim\mu}[T(\boldsymbol{x}) = f(\boldsymbol{x}) \wedge \boldsymbol{x}\in S] - \Pr_{\boldsymbol{x}\sim\mu}[T(\boldsymbol{x}) \neq f(\boldsymbol{x}) \wedge \boldsymbol{x}\in S].\end{aligned}$$

As $|\mathcal{L}(T)| \le 2^d$, there exists some $S \in \mathcal{L}(T)$ – an affine subspace – with large correlation:

$$\mathsf{bias}(f,\mu,S) = \left|\Pr_{\boldsymbol{x}\sim\mu}[T(\boldsymbol{x}) = f(\boldsymbol{x}) \wedge \boldsymbol{x}\in S] - \Pr_{\boldsymbol{x}\sim\mu}[T(\boldsymbol{x}) \neq f(\boldsymbol{x}) \wedge \boldsymbol{x}\in S]\right| \ge \frac{1-2\varepsilon}{2^d}.$$ □

Lemma 29. *For any $\alpha > 0$, $\mathsf{R}_\alpha(\mathrm{FFO}_n) \le O\big(\log n + \log(1/\alpha)\big)$.*

Proof. Let $\mathrm{NOR}_n : \{0,1\}^n \to \{0,1\}$ be the function that checks whether the input is 0^n and rejects otherwise. Observe that one iteration of the sumcheck protocol can be performed in one parity query. More precisely for any $x \in \{0,1\}^n$, if $\boldsymbol{s} \sim U(\{0,1\}^n)$ then:

$$\Pr_{\boldsymbol{s}}[\langle x, \boldsymbol{s}\rangle = 1] = \begin{cases} 1/2 & \text{if } x \neq 0^n \\ 0 & \text{if } x = 0^n \end{cases}.$$

Performing two random checks independently shows that $\mathsf{R}(\mathrm{NOR}_n, 1/4) \le O(1)$. It is a folklore result that a (classical) randomised decision tree can solve FFO_n with probability ε using $O(\log n + \log(1/\varepsilon))$ oracle NOR-queries even if the oracle fails with probability 1/3 [FPRU90, Nis93]. We highlight that this is an improvement over the naive method that boosts the noisy NOR queries and yields complexity $O(\log(n)^2 \log(1/\varepsilon))$. Recent work [HR24, §3] revisits this trick in depth for communication complexity and their result can be re-interpreted in the context of parity decision trees as follows:

$$\forall f\colon\ \mathsf{R}(f,\varepsilon) \le O\big(\mathsf{D}^{\mathrm{NOR}}(f) + \log(1/\varepsilon)\big).$$

Thus, plugging in $f = \mathrm{FFO}$ and noting that $\mathsf{D}^{\mathrm{NOR}}(\mathrm{FFO}_n) \le \log n$ (with binary search), we get the desired result. □

Claim 10. *For any $f\colon \{0,1\}^n \to \{0,1\}$, μ over $\{0,1\}^n$ and $\varepsilon, \delta \ge 0$, $\mathsf{D}_{\varepsilon+\delta}(f,\mu) \le \overline{\mathsf{D}}_\varepsilon(f,\mu)/\delta$.*

Proof. Let $\mathcal{T}$ be a randomised PDT satisfying that $d \coloneqq \overline{q}(\mathcal{T},\mu) = \overline{\mathsf{D}}_\varepsilon(f,\mu)$ and $\mathrm{err}_f(\mathcal{T},\mu) \le \varepsilon$. To prove the lemma, it suffices to construct a deterministic parity tree T of depth $T \le d/\gamma$ with $\mathrm{err}_f(T,\mu) \le \varepsilon + \gamma$. Sample $\boldsymbol{T} \sim \mathcal{T}$. We construct a new tree $\boldsymbol{T}'$ by pruning $\boldsymbol{T}$ as follows: We remove all the nodes of $\boldsymbol{T}$ of depth greater than d/δ. If any node of depth d/δ becomes a leaf, we label it with an arbitrary bit. Note that $\boldsymbol{T}'$ has depth $\le d/\delta$. Finally, let $\mathcal{T}'$ denote the distribution over $\boldsymbol{T}'$ inherited from $\mathcal{T}$.

We observe that for each $x \in \{0,1\}^n$, both $\boldsymbol{T}(x) = f(x)$ and $\boldsymbol{T}'(x) \neq f(x)$ happen only if $q(\boldsymbol{T}, x) > d/\gamma$. Moreover, by Markov's inequality,

$$\Pr_{\substack{\boldsymbol{T}\sim\mathcal{T} \\ \boldsymbol{x}\sim\mu}}[q(\boldsymbol{T},\boldsymbol{x}) > d/\gamma] \le \frac{\overline{q}(\boldsymbol{T},\mu)}{d/\gamma} = \gamma.$$

Therefore, $\mathrm{err}_f(\mathcal{T}',\mu) \le \mathrm{err}_f(\mathcal{T},\mu) + \gamma \le \varepsilon + \gamma$. By an averaging argument, there exists some $T \in \mathrm{supp}(\mathcal{T}')$ of depth $\le d/\delta$ that computes f with error $\mathrm{err}_f(T,\mu) \le \varepsilon + \gamma$, as desired. □

A.4 Ommited proofs of Section 6

In this appendix, we prove Claim 28, an alternative description for the distributions of Section 6.1. Let $p^1, p^2 \in \{0, \star, ?\}^n$. We write $p^1 \sim p^2$ if p^1 and p^2 are consistent over their non$-?$ entries. That is, $p^1 \sim p^2$ if for all $j \in [n]$, if $p^1_j \neq ?$ and $p^2_j \neq ?$, then $p^1_j = p^2_j$. Claim 28 follows from Claims 40 and 41.

Claim 40. *For every reachable state (v, p), consistent $x \in \{0,1\}^n$ and $\rho \in \{0, \star\}^n$, $\mathcal{R}^{v,p,x} \equiv \widehat{\mathcal{R}}^{p,x}$.*

Proof. Upon inspection of $\widehat{\mathcal{R}}^{v,p}$, it is enough to prove that for all $x \in \{0,1\}^n$:

$$\Pr_{\boldsymbol{\rho} \sim \mathcal{R}^{v,p,x}}[\boldsymbol{\rho} = \rho] = \prod_{j \in S^p_{\neq ?}} \mathbb{1}[\rho_j = p_j] \times \prod\nolimits_{\substack{j \in S^p_? \\ x_j = 1}} \mathbb{1}[\rho_j = \star] \times \prod\nolimits_{\substack{j \in S^p_? \\ x_j = 0}} \begin{cases} \delta_j/(2-\delta_j) & \text{if } \rho_j = \star \\ 1 - \delta_j/(2-\delta_j) & \text{if } \rho_j = 0 \end{cases}.$$

Fix $x \in \{0,1\}^n$. We prove this by induction on the state space (v, p) consistent with x. The entry-point of the state space is $(\mathrm{root}(T), ?^n)$. In this case, the statement holds by definition. Suppose now that the statement is true for state (v, p). Depending on the value of ρ, there are several next state (v', p') possible. Observe however that the next vertex of T to be visited does not depend on ρ, as it is fixed to be $v' := \mathrm{child}(v, \langle x, Q^v \rangle)$. For any fixed $\rho \in \{0, \star\}^n$, we have:

$$\begin{aligned} \Pr\nolimits_{\boldsymbol{\rho} \sim \mathcal{R}^{v',p',x}}[\boldsymbol{\rho} = \rho] &= \Pr\nolimits_{\boldsymbol{x} \sim \mu, \boldsymbol{\rho} \sim R^{\boldsymbol{x}}_\mu}[\boldsymbol{\rho} = \rho \mid (v', p') \text{ is reached and } \boldsymbol{x} = x] \\ &= \frac{\Pr_{\boldsymbol{x}, \boldsymbol{\rho}}[\boldsymbol{\rho} = \rho \text{ and } (v', p') \text{ is reached and } \boldsymbol{x} = x]}{\Pr_{\boldsymbol{x}, \boldsymbol{\rho}}[(v', p') \text{ is reached and } \boldsymbol{x} = x]}. \end{aligned}$$

Note that there can be only one state from which (v', p') can be reached, namely (v, p). Indeed, suppose that there is another state $(v, p^\star)$ from which (v', p') can be reached. Then (v, p) and (v, p^*) have a common ancestor (u, q). Since the paths diverged after (u, q), it must be that $p \nsim p^*$ and thus $p^* \nsim p'$: a contradiction. Thus, we have the following equivalence:

$$(v', p') \text{ is reached} \quad \Longleftrightarrow \quad (v, p) \text{ is reached and } \rho \sim p'.$$

Therefore, we have:

$$\Pr\nolimits_{\boldsymbol{\rho} \sim \mathcal{R}^{v',p',x}}[\boldsymbol{\rho} = \rho] = \frac{\Pr_{\boldsymbol{\rho} \sim \mathcal{R}^{v,p,x}}[\boldsymbol{\rho} = \rho] \cdot \mathbb{1}[\rho \sim p']}{\Pr_{\boldsymbol{\rho} \sim \mathcal{R}^{v,p,x}}[\boldsymbol{\rho} \sim p']}. \tag{7}$$

We can now use the inductive hypothesis on (v, p). Since $\rho \sim p'$ implies $\rho \sim p$, the numerator of (7) simplifies to:

$$\prod\nolimits_{j \in S^{p'}_{\neq ?}} \mathbb{1}[\rho_j = p_j] \times \prod\nolimits_{\substack{j \in S^p_? \\ x_j = 1}} \mathbb{1}[\rho_j = \star] \times \prod\nolimits_{\substack{j \in S^p_? \\ x_j = 0}} \begin{cases} \delta_j/(2-\delta_j) & \text{if } \rho_j = \star \\ 1 - \delta_j/(2-\delta_j) & \text{if } \rho_j = 0 \end{cases}.$$

Let $\Delta = S^p_? \setminus S^{p'}_?$ and observe that the denominator of (7) is equal to:

$$\prod\nolimits_{\substack{j \in \Delta \\ x_j = 1}} \mathbb{1}[\rho_j = \star] \times \prod\nolimits_{\substack{j \in \Delta \\ x_j = 0}} \begin{cases} \delta_j/(2-\delta_j) & \text{if } \rho_j = \star \\ 1 - \delta_j/(2-\delta_j) & \text{if } \rho_j = 0 \end{cases}.$$

□

Claim 41. *For every reachable state* (v,p) *and* $x \in \{0,1\}^n$, $\mathcal{X}^{v,p} \equiv \widehat{\mathcal{X}}^{v,p}$.

Proof. Fix some (v,p) and $x \in \{0,1\}^n$. Upon inspection of $\widehat{X}^{v,p}$, it is enough to prove that

$$\Pr_{\boldsymbol{x}\sim\mathcal{X}^{v,p}}[\boldsymbol{x}=x] = M(x,v,p)\cdot\prod\nolimits_{j\in S^p_?} 1-\delta_j/2 - x_j\cdot(1-\delta_j),$$

where $M(x,v,p)$ is an indicator set to 1 if and only if for all $j\in[n]$, $p_j=0$ implies $x_j=0$ and $\langle x, Q^u\rangle = b^u$ for all $u\in\text{path}(v)$. By Baye's rule we have:

$$\begin{aligned}\Pr_{\boldsymbol{x}\sim\mathcal{X}^{v,p}}[\boldsymbol{x}=x] &= \Pr_{\substack{\boldsymbol{x}\sim\mu\\ \boldsymbol{\rho}\sim\mathcal{R}^{\boldsymbol{x}}_\mu}}[\boldsymbol{x}=x \mid (v,p)\text{ is reached on }(\boldsymbol{x},\boldsymbol{\rho})]\\ &= \frac{p(x)}{\sum_{x'\in\{0,1\}^n} p(x')} \text{ where } p(x) \coloneqq \Pr_{\boldsymbol{x},\boldsymbol{\rho}}[\boldsymbol{x}=x]\cdot\Pr_{\boldsymbol{x},\boldsymbol{\rho}}[(v,p)\text{ is reached on }(\boldsymbol{x},\boldsymbol{\rho})\mid\boldsymbol{x}=x].\end{aligned}$$

To analyse $p(x)$, we have:

$$\Pr_{\substack{\boldsymbol{x}\sim\mu\\ \boldsymbol{\rho}\sim\mathcal{R}^{\boldsymbol{x}}_\mu}}[\boldsymbol{x}=x] = \Pr_{\boldsymbol{x}\sim\mu}[\boldsymbol{x}=x] = \prod\nolimits_{j\in[n]}\Pr_{\boldsymbol{x}\sim\mu}[\boldsymbol{x}_j=x_j] = \prod\nolimits_{j\in[n]} 1-(\delta_j/2)-x_j\cdot(1-\delta_j)$$

On the other hand, the second component of $p(x)$ is clearly zero if $M(x,v,p)=0$. For instance, v cannot be reached if x does not satisfy all equations on the path to v. Thus, we have:

$$\begin{aligned}\Pr_{\substack{\boldsymbol{x}\sim\mu\\ \boldsymbol{\rho}\sim\mathcal{R}^{\boldsymbol{x}}_\mu}}[(v,p)\text{ is reached on }(\boldsymbol{x},\boldsymbol{\rho})\mid\boldsymbol{x}=x] &= \Pr_{\boldsymbol{\rho}\sim\mathcal{R}^x_\mu}[(v,p)\text{ is reached on }(x,\boldsymbol{\rho})]\\ &= M(x,v,p)\cdot\Pr_{\boldsymbol{\rho}\sim\mathcal{R}^x_\mu}[\boldsymbol{\rho}\sim p]\\ &= M(x,v,p)\cdot\prod\nolimits_{j\in S^p_0}\frac{2-2\delta_j}{2-\delta_j}\cdot\prod\nolimits_{j\in S^p_\star}\left(\frac{\delta_j}{2-\delta_j}\right)^{1-x_j}.\end{aligned}$$

Combining those two observations, we get:

$$p(x) = M(v,p,x)\cdot\prod\nolimits_{j\in S^p_?}\big(1-\delta_j/2-x_j\cdot(1-\delta_j)\big)\cdot\prod\nolimits_{j\in S^p_0}(1-\delta_j)\cdot\prod\nolimits_{j\in S^p_\star}\delta_j/2.$$

Observe that the last two products do not involve x at all and can thus be cancelled in the initial expression:

$$\Pr_{\boldsymbol{x}\sim\mathcal{X}^{v,p}}[\boldsymbol{x}=x] = \frac{p'(x)}{\sum_{x'}p'(x)} \text{ where } p'(x) = M(x,v,p)\cdot\prod\nolimits_{j\in S^p_?}\big(1-\delta_j/2-x_j\cdot(1-\delta_j)\big).$$

Finally, observe that $M(x,v,p)$ fixes the value of all the bits of x except for $S^p_?$. Thus, the summation in the denominator equals 1 and the claim follows. ☐

Acknowledgements

M.G., G.M., and W.Y. are supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00026. T.B., and S.G. are supported by the National Natural Science Foundation of China Grant No.62102260, NYTP Grant No.20121201, and NYU Shanghai Boost Fund.

References

[AFS24] Yaroslav Alekseev, Yuval Filmus, and Alexander Smal. Lifting Dichotomies. In *39th Computational Complexity Conference (CCC 2024)*, volume 300 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.CCC.2024.9.

[AI24] Yaroslav Alekseev and Dmitry Itsykson. Lifting to bounded-depth and regular resolutions over parities via games. Technical Report TR24-128, ECCC, 2024. URL: https://eccc.weizmann.ac.il/report/2024/128/.

[BB19] Eric Blais and Joshua Brody. Optimal separation and strong direct sum for randomized query complexity. In *Proceedings of the 34th Computational Complexity Conference*, CCC '19. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. doi:10.4230/LIPIcs.CCC.2019.29.

[BB20] Shalev Ben-David and Eric Blais. A tight composition theorem for the randomized query complexity of partial functions: Extended abstract. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 240–246, 2020. doi:10.1109/FOCS46700.2020.00031.

[BB23] Shalev Ben-David and Eric Blais. A new minimax theorem for randomized algorithms. *J. ACM*, 70(6), 2023. doi:10.1145/3626514.

[BBCR13] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. *SIAM Journal on Computing*, 42(3):1327–1363, 2013. doi:10.1137/100811969.

[BBGM22] Shalev Ben-David, Eric Blais, Mika Göös, and Gilbert Maystre. Randomised Composition and Small-Bias Minimax . In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 624–635. IEEE Computer Society, 2022. doi:10.1109/FOCS54457.2022.00065.

[BCD24] Sreejata Kishor Bhattacharya, Arkadev Chattopadhyay, and Pavel Dvořák. Exponential Separation Between Powers of Regular and General Resolution over Parities. In *39th Computational Complexity Conference (CCC 2024)*, volume 300 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:32. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.CCC.2024.23.

[BFS86] Laszlo Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 337–347, 1986. doi:10.1109/SFCS.1986.15.

[BGKW20] Shalev Ben-David, Mika Göös, Robin Kothari, and Thomas Watson. When Is Amplification Necessary for Composition in Randomized Query Complexity? In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*, volume 176 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.28.

[BGPW15] Mark Braverman, Ankit Garg, Denis Pankratov, and Omri Weinstein. Information lower bounds via self-reducibility. *Theory of Computing Systems*, 59(2):377–396, 2015. doi:10.1007/s00224-015-9655-z.

[BI24] Farzan Byramji and Russell Impagliazzo. Lifting to randomized parity decision trees. Technical Report TR24-202, ECCC, 2024. URL: https://eccc.weizmann.ac.il/report/2024/202/.

[BK18] Shalev Ben-David and Robin Kothari. Randomized query complexity of sabotaged and composed functions. *Theory of Computing*, 14(5):1–27, 2018. doi:10.4086/toc.2018.v014a005.

[BK23] Paul Beame and Sajin Koroth. On Disperser/Lifting Properties of the Index and Inner-Product Functions. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.14.

[BKLS23] Joshua Brody, Jae Tak Kim, Peem Lerdputtipongporn, and Hariharan Srinivasulu. A strong XOR lemma for randomized query complexity. *Theory of Computing*, 19(11):1–14, 2023. doi:10.4086/toc.2023.v019a011.

[BKST24] Guy Blanc, Caleb Koch, Carmen Strassle, and Li-Yang Tan. A Strong Direct Sum Theorem for Distributional Query Complexity. In *39th Computational Complexity Conference (CCC 2024)*, volume 300 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:30. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.CCC.2024.16.

[BR14] Mark Braverman and Anup Rao. Information equals amortized communication. *IEEE Transactions on Information Theory*, 60(10):6058–6069, 2014. doi:10.1109/TIT.2014.2347282.

[CD24] Arkadev Chattopadhyay and Pavel Dvorak. Super-critical trade-offs in resolution over parities via lifting. Technical Report TR24-132, ECCC, 2024. URL: https://eccc.weizmann.ac.il/report/2024/132/.

[CHZZ24] Tsun Ming Cheung, Hamed Hatami, Rosie Zhao, and Itai Zilberstein. Boolean functions with small approximate spectral norm. *Discrete Analysis*, 2024. doi:10.19086/da.122971.

[CMSS23] Arkadev Chattopadhyay, Nikhil Mande, Swagato Sanyal, and Suhail Sherif. Lifting to Parity Decision Trees via Stifling. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.33.

[Dru12] Andrew Drucker. Improved direct product theorems for randomized query complexity. *Comput. Complex.*, 21(2):197–244, 2012. doi:10.1007/s00037-012-0043-7.

[EGI24] Klim Efremenko, Michal Garlík, and Dmitry Itsykson. Lower bounds for regular resolution over parities. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, volume 41 of *STOC '24*, pages 640–651. ACM, 2024. doi:10.1145/3618260.3649652.

[FHR+24] Yuval Filmus, Edward Hirsch, Artur Riazanov, Alexander Smal, and Marc Vinyals. Proving Unsatisfiability with Hitting Formulas. In *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 48:1–48:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ITCS.2024.48.

[FKNN95] Tomás Feder, Eyal Kushilevitz, Moni Naor, and Noam Nisan. Amortized communication complexity. *SIAM Journal on Computing*, 24(4):736–750, 1995. doi:10.1137/S0097539792235864.

[FPRU90] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Computing with unreliable information. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 128–137. Association for Computing Machinery, 1990. doi:10.1145/100216.100230.

[GKR16] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication for boolean functions. *J. ACM*, 63(5), 2016. doi:10.1145/2907939.

[GM21] Mika Göös and Gilbert Maystre. A majority lemma for randomised query complexity. In *Proceedings of the 36th Computational Complexity Conference*, CCC '21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2021. doi:10.4230/LIPIcs.CCC.2021.18.

[GSTW23] Uma Girish, Makrand Sinha, Avishay Tal, and Kewen Wu. Fourier growth of communication protocols for XOR functions. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 721–732, 2023. doi:10.1109/FOCS57990.2023.00047.

[GTW21] Uma Girish, Avishay Tal, and Kewen Wu. Fourier growth of parity decision trees. In *Proceedings of the 36th Computational Complexity Conference*, CCC '21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2021. doi:10.4230/LIPIcs.CCC.2021.39.

[HHL18] Hamed Hatami, Kaave Hosseini, and Shachar Lovett. Structure of protocols for XOR functions. *SIAM Journal on Computing*, 47(1):208–217, 2018. doi:10.1137/17M1136869.

[HHLO24] Hamed Hatami, Kaave Hosseini, Shachar Lovett, and Anthony Ostuni. Refuting Approaches to the Log-Rank Conjecture for XOR Functions. In *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 82:1–82:11. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ICALP.2024.82.

[HR24] Nathaniel Harms and Artur Riazanov. Better Boosting of Communication Oracles, or Not. In Siddharth Barman and Sławomir Lasota, editors, *44th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2024)*, volume 323 of *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.FSTTCS.2024.25.

[IR24a] Siddharth Iyer and Anup Rao. An XOR lemma for deterministic communication complexity, 2024. URL: https://arxiv.org/abs/2407.01802.

[IR24b] Siddharth Iyer and Anup Rao. XOR lemmas for communication via marginal information. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 652—658. Association for Computing Machinery, 2024. doi:10.1145/3618260.3649726.

[IS20] Dmitry Itsykson and Dmitry Sokolov. Resolution over linear equations modulo two. *Annals of Pure and Applied Logic*, 171(1):102722, 2020. doi:10.1016/j.apal.2019.102722.

[JKS10] Rahul Jain, Hartmut Klauck, and Miklos Santha. Optimal direct sum results for deterministic and randomized decision tree complexity. *Information Processing Letters*, 110(20):893–897, 2010. doi:10.1016/j.ipl.2010.07.020.

[JRS03] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A direct sum theorem in communication complexity via message compression. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming*, pages 300–315. Springer Berlin Heidelberg, 2003. doi:10.1007/3-540-45061-0_26.

[KLMY21] A. Knop, S. Lovett, S. McGuire, and W. Yuan. Guest column: Models of computation between decision trees and communication. *SIGACT News*, 52(2):46–70, 2021. doi:10.1145/3471469.3471479.

[KM93] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993. doi:10.1137/0222080.

[KvdW07] Hartmut Klauck, Robert Špalek, and Ronald de Wolf. Quantum and classical strong direct product theorems and optimal time-space tradeoffs. *SIAM Journal on Computing*, 36(5):1472–1493, 2007. doi:10.1137/05063235X.

[LS08] Nati Linial and Adi Shraibman. Learning complexity vs. communication complexity. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 53–63, 2008. doi:10.1109/CCC.2008.28.

[LSv08] Troy Lee, Adi Shraibman, and Robert Špalek. A direct product theorem for discrepancy. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 71–80, 2008. doi:10.1109/CCC.2008.25.

[MS24] Nikhil Mande and Swagato Sanyal. On parity decision trees for fourier-sparse boolean functions. *ACM Trans. Comput. Theory*, 16(2), 2024. doi:10.1145/3647629.

[Nis93] Noam Nisan. The communication complexity of threshold gates. *Proc. of Combinatorics, Paul Erdős is Eighty*, 1993.

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. doi:10.1017/CBO9781139814782.

[OWZ$^+$14] Ryan ODonnell, John Wright, Yu Zhao, Xiaorui Sun, and Li-Yang Tan. A Composition Theorem for Parity Kill Number . In *2014 IEEE Conference on Computational Complexity (CCC)*, pages 144–154. IEEE Computer Society, 2014. doi:10.1109/CCC.2014.22.

[RS18] Anup Rao and Makrand Sinha. Simplified separation of information and communication. *Theory of Computing*, 14(20):1–29, 2018. doi:10.4086/toc.2018.v014a020.

[San19] Swagato Sanyal. Fourier sparsity and dimension. *Theory of Computing*, 15(11):1–13, 2019. doi:10.4086/toc.2019.v015a011.

[San24] Swagato Sanyal. Randomized query composition and product distributions. In *41st International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 289 of *LIPIcs*, pages 56:1–56:19. Schloss Dagstuhl, 2024. doi:10.4230/LIPIcs.STACS.2024.56.

[Sav02] Petr Savický. On determinism versus unambiquous nondeterminism for decision trees. Technical Report TR02-009, Electronic Colloquium on Computational Complexity (ECCC), 2002. URL: http://eccc.hpi-web.de/report/2002/009/.

[Sha03] Ronen Shaltiel. Towards proving strong direct product theorems. *computational complexity*, 12(1):1–22, 2003. doi:10.1007/s00037-003-0175-x.

[SP25] Alexander Shekhovtsov and Vladimir Podolskii. Randomized lifting to semi-structured communication complexity via linear diversity. In *16th Innovations in Theoretical Computer Science Conference (ITCS)*, LIPIcs. Schloss Dagstuhl, 2025.

[STV17] Amir Shpilka, Avishay Tal, and Ben Volk. On the structure of boolean functions with small spectral norm. *computational complexity*, 26(1):229–273, 2017. doi:doi.org/10.1007/s00037-015-0110-y.

[TWXZ13] Hing Yin Tsang, Chung Hoi Wong, Ning Xie, and Shengyu Zhang. Fourier sparsity, spectral norm, and the log-rank conjecture. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 658–667, 2013. doi:10.1109/FOCS.2013.76.

[Yao77] Andrew Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, SFCS '77, page 222–227. IEEE Computer Society, 1977. doi:10.1109/SFCS.1977.24.

[Yao83] Andrew Yao. Lower bounds by probabilistic arguments. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science*, SFCS '83, page 420–428. IEEE Computer Society, 1983. doi:10.1109/SFCS.1983.30.

[ZS10] Zhiqiang Zhang and Yaoyun Shi. On the parity complexity measures of boolean functions. *Theoretical Computer Science*, 411(26-28):2612–2618, 2010. doi:10.1016/j.tcs.2010.03.027.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il