Weak Parity

Scott Aaronson* Andris Ambainis† Kaspars Balodis‡ Mohammad Bavarian§

Abstract

We study the query complexity of WEAK PARITY: the problem of computing the parity of an n-bit input string, where one only has to succeed on a $1/2+\varepsilon$ fraction of input strings, but must do so with high probability on those inputs where one does succeed. It is well-known that n randomized queries and $n/2$ quantum queries are needed to compute parity on *all* inputs. But surprisingly, we give a randomized algorithm for WEAK PARITY that makes only $O(n/\log^{0.246}(1/\varepsilon))$ queries, as well as a quantum algorithm that makes $O(n/\sqrt{\log(1/\varepsilon)})$ queries. We also prove a lower bound of $\Omega\left(n/\log\left(1/\varepsilon\right)\right)$ in both cases, as well as lower bounds of $\Omega(\log n)$ in the randomized case and $\Omega(\sqrt{\log n})$ in the quantum case for any $\varepsilon > 0$. We show that improving our lower bounds is intimately related to two longstanding open problems about Boolean functions: the Sensitivity Conjecture, and the relationships between query complexity and polynomial degree.

1 Introduction

Given a Boolean input $X = (x_1, \ldots, x_n) \in \{0,1\}^n$, the PARITY problem is to compute

$$\textsc{Par}\left(X\right) := x_1 \oplus \cdots \oplus x_n\,. \tag{1}$$

This is one of the most fundamental and well-studied problems in computer science.

Since $\textsc{Par}(X)$ is sensitive to all n bits at every input X, any classical algorithm for PARITY requires examining all n bits. As a result, PARITY is often considered a "maximally hard problem" for query or decision-tree complexity. In the quantum case, one can get a *slight* improvement to $\lceil n/2 \rceil$ queries, by applying the Deutsch-Jozsa algorithm [10] to successive pairs of coordinates ((x_1, x_2), (x_3, x_4), etc.) and then XORing the results. However, that factor-of-two improvement is known to be the best possible by quantum algorithms [12, 5].[1]

So we might wonder: can we learn *anything* about a string's parity by making a sublinear number of queries? One natural goal would be to compute the parity, not for all inputs, but merely for as many inputs as possible. This motivates the following problem, which will be the focus of this paper.

*MIT. aaronson@csail.mit.edu. Supported by the National Science Foundation under Grant No. 0844626 and an Alan T. Waterman Award.

†University of Latvia. andris.ambainis@lu.lv. Supported by the European Commission under the project QALGO (Grant No. 600700) and the ERC Advanced Grant MQC (Grant No. 320731).

‡University of Latvia. kbalodis@gmail.com. Supported by the European Social Fund within the project "Support for Doctoral Studies at University of Latvia."

§MIT. bavarian@mit.edu. Partially supported by NSF STC Award 0939370.

[1] Moreover, this holds even for *unbounded-error* quantum algorithms, which only need to guess $\textsc{Par}(X)$ with *some* probability greater than $1/2$, but must do so for every X.

Problem 1 (WEAK PARITY or $\text{WEAKPAR}_{n,\varepsilon}$) *Given $\varepsilon > 0$, design an algorithm that queries a Boolean input $X \in \{0,1\}^n$ as few times as possible, and whose acceptance probability $p(X)$ satisfies*

$$\Pr_{X \in \{0,1\}^n} \left[|p(X) - \text{PAR}(X)| \leq \frac{1}{3} \right] \geq \frac{1}{2} + \varepsilon. \tag{2}$$

Equivalently, the algorithm should satisfy $|A| \geq (1/2 + \varepsilon) 2^n$, where $A \subseteq \{0,1\}^n$ is the set of all inputs X such that $|p(X) - \text{PAR}(X)| \leq 1/3$.

We will sometimes refer to the above as "bounded-error" WEAK PARITY. In the "zero-error" variant, we instead want to satisfy the stronger condition

$$\Pr_{X \in \{0,1\}^n} [p(X) = \text{PAR}(X)] \geq \frac{1}{2} + \varepsilon. \tag{3}$$

To build intuition, let's start with some elementary remarks about WEAK PARITY.

(i) Of course it's trivial to guess $\text{PAR}(X)$ on a $1/2$ fraction of inputs X, for example by always outputting 0. (On the other hand, being *wrong* on a $1/2 + \varepsilon$ fraction of X's is just as hard as being right on that fraction.)

(ii) As usual, the constant $1/3$ in equation (2) is arbitrary; we can replace it by any other constant in $(0, 1/2)$ using amplification.

(iii) There is no requirement that the acceptance probability $p(X)$ approximate a total Boolean function. In other words, if $X \notin A$ then $p(X)$ can be anything in $[0,1]$.

(iv) It is not hard to see that WEAK PARITY is completely uninteresting for deterministic classical algorithms. Indeed, any such algorithm that makes fewer than n queries correctly guesses $\text{PAR}(X)$ on exactly half of the inputs.

(v) Even a randomized or quantum algorithm must be "uncorrelated" with $\text{PAR}(X)$, if it always makes $T < n$ queries (in the randomized case) or $T < n/2$ queries (in the quantum case). In other words, we must have

$$\sum_{X \in \{0,1\}^n} \left(p(X) - \frac{1}{2} \right) \left(\text{PAR}(X) - \frac{1}{2} \right) = 0, \tag{4}$$

where $p(X)$ is the algorithm's acceptance probability. The reason is just Fourier analysis: if we switch domains from $\{0,1\}$ to $\{1,-1\}$, then $\text{PAR}(X) = x_1 \cdots x_n$. But for a randomized algorithm, $p(X)$ is a multilinear polynomial in $x_1, \ldots, x_n$ of degree at most $T < n$, while for a quantum algorithm, Beals et al. [5] showed that $p(X)$ is a multilinear polynomial of degree at most $2T < n$. And any such polynomial has correlation 0 with the degree-n monomial $x_1 \cdots x_n$.

(vi) Crucially, however, equation (4) does *not* rule out sublinear randomized or quantum algorithms for WEAK PARITY (which exist for all $\varepsilon = o(1)$, as we will see!). The reason is a bit reminiscent of the famous *hat puzzle*:[2] suppose, for example, that an algorithm output $\text{PAR}(X)$ with probability exactly $2/3$ on a $3/4$ fraction of inputs X, and with probability 0 on the remaining $1/4$ fraction of inputs. Such an algorithm would succeed at WEAK PARITY for $\varepsilon = 1/4$, despite maintaining an overall correlation of 0 with $\text{PAR}(X)$.

[2] In that puzzle, n players are each assigned a red hat or a blue hat uniformly at random, and can see the colors of every hat except their own. At least one player must guess the color of her own hat, and every guess must be correct. Surprisingly, even though each player has only a $1/2$ probability of being correct, it is possible for the players to win this game with probability $\sim 1 - 1/n$, by "conspiring" so that the cases where they are wrong coincide with each other. See http://en.wikipedia.org/wiki/Hat_puzzle

(vii) The correlation argument does establish that, for the *zero-error* variant of WEAK PARITY, any randomized algorithm must make at least n queries, and any quantum algorithm must make at least $n/2$ queries, with *some* nonzero probability.[3] Even then, however, an algorithm that makes an *expected* sublinear number of queries on each input X is not ruled out (and as we will see, such algorithms exist).

The regime of WEAK PARITY that interests us the most is where ε is very small—the extreme case being $\varepsilon = 1/2^n$. We want to know: *are there fast randomized or quantum algorithms to guess the parity of X on slightly more than half the inputs?*

Despite an immense amount of work on query complexity, so far as we know the above question was never asked before. Here we initiate its study, both by proving upper and lower bounds, and by relating this innocent-looking question to longstanding open problems in combinatorics, including the Sensitivity Conjecture. Even though WEAK PARITY might look at first like a curiosity, we will find that the task of understanding its query complexity is tightly linked to *general* questions about query complexity, and these links help to motivate its study. Conversely, WEAK PARITY illustrates how an old pastime in complexity theory—namely, understanding the largest possible gaps between query complexity measures for *arbitrary* Boolean functions—can actually have implications for the query complexities of *specific* problems.

2 Our Results

First, in Section 4, we prove an upper bound of $O(n/\log^{0.246}(1/\varepsilon))$ on the zero-error randomized query complexity of WEAK PARITY, and an upper bound of $O(n/\sqrt{\log 1/\varepsilon})$ on its bounded-error quantum query complexity. (For zero-error quantum query complexity, we get the slightly worse bound $O\left(n \cdot \frac{\left(\log\log \frac{1}{\varepsilon}\right)^2}{\sqrt{\log 1/\varepsilon}}\right)$.)

Our quantum algorithm is based on Grover's algorithm, while our randomized algorithm is based on the well-known $O\left(n^{0.754}\right)$ randomized algorithm for the complete binary AND/OR tree. For the zero-error quantum algorithm, we use a recent zero-error quantum algorithm for the complete binary AND/OR tree due to Ambainis et al. [3].

Then, in Section 5, we prove a not-quite-matching lower bound of $\Omega\left(n/\log\left(1/\varepsilon\right)\right)$ queries, by using random self-reducibility to reduce ordinary PARITY to WEAK PARITY. This lower bound is the same for randomized and quantum, and for zero-error and bounded-error.

The gap between our upper and lower bounds might seem tiny. But notice that the gap steadily worsens for smaller ε, reaching $O(n^{0.754})$ or $O(\sqrt{n})$ or $O(\sqrt{n}\log^2 n)$ versus the trivial $\Omega\left(1\right)$ when $\varepsilon = 1/2^n$. This leads us to ask whether we can prove a nontrivial lower bound that works for *all* $\varepsilon > 0$. Equivalently, can we rule out an $O\left(1\right)$-query randomized or quantum algorithm that computes PARITY on a subset $A \subseteq \{0,1\}^n$ of size $2^{n-1}+1$?

In Section 6, we show that we *can* (barely) rule out such an algorithm. In 1988, Chung et al. [9] showed that any induced subgraph of the Boolean hypercube $\{0,1\}^n$, of size at least $2^{n-1}+1$, must have at least one vertex of degree $\Omega\left(\log n\right)$. As a consequence, we deduce that for all $\varepsilon > 0$, any bounded-error randomized algorithm for WEAK PARITY must make $\Omega(\log n)$ queries, and any bounded-error quantum algorithm must make $\Omega(\sqrt{\log n})$ queries. For the $\Omega(\log n)$ randomized lower bound, we also include a self-contained proof due to Andy Drucker.

It has been conjectured that Chung et al.'s $\Omega\left(\log n\right)$ degree lower bound can be improved to $n^{\Omega(1)}$. Previously, however, Gotsman and Linial [13] showed that such an improvement would imply the notorious *Sensitivity Conjecture* in the study of Boolean functions. In Section 6, we observe that an $n^{\Omega(1)}$ lower

[3] For the bounded-error variant of WEAK PARITY, the argument also establishes that if $\varepsilon > 1/4$, then any randomized algorithm must make n queries, and any quantum algorithm must make $n/2$ queries.

bound for Chung et al.'s problem would *also* yield an $n^{\Omega(1)}$ lower bound on the bounded-error randomized and quantum query complexities of WEAK PARITY, for all $\varepsilon > 0$. Thus, while we do not have a direct reduction between WEAK PARITY and the Sensitivity Conjecture in either direction, it seems plausible that a breakthrough on one problem would lead to a breakthrough on the other.

Next, in Section 7, we connect WEAK PARITY to another longstanding open problem in the study of Boolean functions—and in this case, we give a direct reduction. Namely, suppose we could prove a lower bound of $\Omega\left(n/\log^{1-c}(1/\varepsilon)\right)$ on the bounded-error randomized query complexity of WEAK PARITY. We show that this would imply that $\mathrm{R}_2(f) = \Omega(\deg(f)^c)$ for all total Boolean functions $f : \{0,1\}^n \rightarrow \{0,1\}$, where $\mathrm{R}_2(f)$ is the bounded-error randomized query complexity of f, and $\deg(f)$ is its exact degree as a real polynomial. Similar statements hold for other kinds of query complexity (e.g., the bounded-error quantum query complexity $\mathrm{Q}_2(f)$, and the zero-error randomized query complexity $\mathrm{R}_0(f)$).

Nisan [16] showed that $\mathrm{R}_2(f) = \Omega(\deg(f)^{1/3})$ for all total Boolean functions f, while Beals et al. [5] showed that $\mathrm{Q}_2(f) = \Omega(\deg(f)^{1/6})$ for all f.[4] Meanwhile, the largest known separations are $\mathrm{R}_2(f) = O(\deg(f)^{0.753\ldots})$ if f is the complete binary AND/OR tree (see Section 3 for a definition), and $\mathrm{Q}_2(f) = O(\sqrt{\deg(f)})$ if f is the OR function. However, even improving on the 3^{rd}- and 6^{th}-power relations remains open. Our result says that, if there existed Boolean functions f with larger separations than are currently known, then we *could* improve our algorithms for WEAK PARITY. And conversely, any randomized lower bound for WEAK PARITY better than $\Omega(n/\log^{2/3}(1/\varepsilon))$, or any quantum lower bound better than $\Omega(n/\log^{5/6}(1/\varepsilon))$, would improve the known relations between degree and query complexity for *all* Boolean functions.

Lastly, in Section 8, we briefly consider the weak query complexities of functions other than PARITY. We show that, for *every* Boolean function f, it is possible to agree with $f(X)$ on $2^{n-1}+1$ inputs X using a bounded-error quantum algorithm that makes $O(\sqrt{n})$ queries, or a zero-error randomized algorithm that makes $O(n^{0.754})$ queries, or a zero-error quantum algorithm that makes $O(\sqrt{n}\log^2 n)$ queries.

3 Preliminaries

We assume some familiarity with classical and quantum query complexity; see Buhrman and de Wolf [8] for an excellent introduction. This section reviews the most relevant definitions and facts.

3.1 Classical Query Complexity

Given a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, the *deterministic query complexity* $\mathrm{D}(f)$ is the minimum number of queries made by any deterministic, classical algorithm that computes $f(X)$ for every input $X \in \{0,1\}^n$. (Here and throughout, a query returns x_i given i, and the "number of queries" means the number maximized over all $X \in \{0,1\}^n$.)

Also, the *zero-error randomized query complexity* $\mathrm{R}_0(f)$ is the minimum number of queries made by any randomized algorithm that computes $f(X)$ with success probability at least $2/3$ for every X—and that, whenever it fails to compute $f(X)$, instead outputs "don't know." The *bounded-error randomized query complexity* $\mathrm{R}_2(f)$ is the minimum number of queries made by a randomized algorithm that computes $f(X)$ with success probability at least $2/3$ for every X, and that can behave arbitrarily (for example, by outputting the wrong answer) when it fails. We have the following relations for every f:

$$n \geq \mathrm{D}(f) \geq \mathrm{R}_0(f) \geq \mathrm{R}_2(f). \tag{5}$$

[4] More precisely, they showed respectively that $\mathrm{R}_2(f) = \Omega(\mathrm{D}(f)^{1/3})$ and $\mathrm{Q}_2(f) = \Omega(\mathrm{D}(f)^{1/6})$ for all f. However, the stated results follow by combining those results with the elementary fact $\deg(f) \leq \mathrm{D}(f)$.

We could also have defined $\mathrm{R}_0(f)$ as the minimum *expected* number of queries made by a randomized algorithm that computes $f(X)$ with certainty for every input X (where the expectation is over the internal randomness of the algorithm, and must be bounded for every X). We will sometimes use this interpretation, which changes the value of $\mathrm{R}_0(f)$ by at most a constant factor.

We will use the following well-known result:

Theorem 2 *$\mathrm{D}(f) \leq \mathrm{R}_0(f)^2$ and $\mathrm{D}(f) = O(\mathrm{R}_2(f)^3)$ for all total Boolean functions f.*[5]

We will write $\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon})$ to denote the minimum number of queries made by any randomized algorithm that, for at least a $1/2+\varepsilon$ fraction of inputs $X \in \{0,1\}^n$, outputs $\textsc{Par}(X)$ with probability at least $2/3$. We will also write $\mathrm{R}_0(\textsc{WeakPar}_{n,\varepsilon})$ to denote the minimum number of queries made by any randomized algorithm that satisfies the following two properties, for at least a $1/2+\varepsilon$ fraction of inputs X:

- The algorithm outputs $\textsc{Par}(X)$ with probability at least $2/3$.
- If the algorithm does *not* output $\textsc{Par}(X)$, then it outputs "don't know."

In both the R_2 and R_0 cases, for the remaining inputs X (i.e., those on which the algorithm fails), the algorithm's output behavior can be arbitrary, but the upper bound on query complexity must hold for *all* inputs $X \in \{0,1\}^n$.

Note that we could also define $\mathrm{R}'_0(\textsc{WeakPar}_{n,\varepsilon})$ as the minimum *expected* number of queries made by any randomized algorithm that, for at least a $1/2+\varepsilon$ fraction of inputs X, outputs $\textsc{Par}(X)$ with probability 1. In this case, the expected number of queries needs to be bounded only for those X's on which the algorithm succeeds. For completeness, let us verify the following.

Proposition 3 *$\mathrm{R}_0(\textsc{WeakPar}_{n,\varepsilon})$ and $\mathrm{R}'_0(\textsc{WeakPar}_{n,\varepsilon})$ are equal up to constant factors.*

Proof. Let A be a randomized algorithm that realizes $\mathrm{R}_0(\textsc{WeakPar}_{n,\varepsilon}) \leq T$. Then we can simply run A repeatedly, until it outputs either 0 or 1. This will yield an algorithm that, for at least a $1/2+\varepsilon$ fraction of inputs $X \in \{0,1\}^n$, outputs $\textsc{Par}(X)$ with certainty after $O(T)$ queries in expectation. (The algorithm might not halt for the remaining X's, but that's okay.)

Conversely, let A' be a randomized algorithm that realizes $\mathrm{R}'_0(\textsc{WeakPar}_{n,\varepsilon}) \leq T$. Then we can run A' until it's either halted or made $3T$ queries, and can output "don't know" in the latter case. By Markov's inequality, this will yield an algorithm that, for at least a $1/2+\varepsilon$ fraction of inputs X, outputs $\textsc{Par}(X)$ with probability at least $2/3$, and otherwise outputs "don't know." Furthermore, the number of queries will be bounded by $3T$ for every X. ■

3.2 Quantum Query Complexity

The *zero-error quantum query complexity* $\mathrm{Q}_0(f)$ is the minimum number of queries made by any quantum algorithm that computes $f(X)$ with success probability at least $2/3$, for every input X—and that, whenever it fails to compute $f(X)$, instead outputs "don't know." Here a query maps each computational basis state of the form $|i,b,z\rangle$ to a basis state of the form $|i,b \oplus x_i, z\rangle$, where z is a "workspace register" whose dimension can be arbitrary. The final output (0, 1, or "don't know") is obtained by measuring a designated part of z. The *bounded-error randomized query complexity* $\mathrm{Q}_2(f)$ is the minimum number of queries made

[5] The $\mathrm{D}(f) \leq \mathrm{R}_0(f)^2$ part follows from the folklore result that $\mathrm{D}(f) \leq \mathrm{C}(f)^2$, where $\mathrm{C}(f)$ is the so-called *certificate complexity*, together with the fact that $\mathrm{R}_0(f) \geq \mathrm{C}(f)$. The $\mathrm{D}(f) = O(\mathrm{R}_2(f)^3)$ part was proved by Nisan [16]. It also follows from the result of Beals et al. [5] that $\mathrm{D}(f) \leq \mathrm{bs}(f)^3$, where $\mathrm{bs}(f)$ is the *block sensitivity* (see Section 3.4), together with the fact that $\mathrm{R}_2(f) = \Omega(\mathrm{bs}(f))$.

by a quantum algorithm that computes $f(X)$ with success probability at least $2/3$ for every X, and whose output can be arbitrary when it fails. We have the following relations for every f:

$$\mathrm{R}_0(f) \geq \mathrm{Q}_0(f) \geq \mathrm{Q}_2(f), \quad \mathrm{R}_2(f) \geq \mathrm{Q}_2(f). \tag{6}$$

Like in the randomized case, we can also interpret $\mathrm{Q}_0(f)$ as the minimum *expected* number of queries made by a quantum algorithm that computes $f(X)$ with certainty for every input X, if we generalize the quantum query model to allow intermediate measurements. Doing so changes $\mathrm{Q}_0(f)$ by at most a constant factor.

We will use the following results of Beals et al. [5] and Midrijanis [15] respectively:

Theorem 4 (Beals et al. [5]) $\mathrm{D}(f) = O(\mathrm{Q}_2(f)^6)$ *for all total Boolean* f.

Theorem 5 (Midrijanis [15]) $\mathrm{D}(f) = O(\mathrm{Q}_0(f)^3)$ *for all total Boolean* f.[6]

Just like in the randomized case, we will write $\mathrm{Q}_2(\textsc{WeakPar}_{n,\varepsilon})$ for the minimum number of queries made by any quantum algorithm that, for at least a $1/2+\varepsilon$ fraction of inputs X, outputs $\textsc{Par}(X)$ with probability at least $2/3$; and will write $\mathrm{Q}_0(\textsc{WeakPar}_{n,\varepsilon})$ for the minimum number of queries made by any quantum algorithm that satisfies the following two properties, for at least a $1/2+\varepsilon$ fraction of X's:

- The algorithm outputs $\textsc{Par}(X)$ with probability at least $2/3$.
- If the algorithm does *not* output $\textsc{Par}(X)$, then it outputs "don't know."

Once again, if we generalize the quantum query model to allow intermediate measurements, then we can also define $\mathrm{Q}_0(\textsc{WeakPar}_{n,\varepsilon})$ as the minimum *expected* number of queries made by any quantum algorithm that, for at least a $1/2+\varepsilon$ fraction of X's, outputs $\textsc{Par}(X)$ with probability 1 (with the expected number of queries bounded only for those X's on which the algorithm succeeds). Doing so changes $\mathrm{Q}_0(\textsc{WeakPar}_{n,\varepsilon})$ by at most a constant factor, for the same reasons as in Proposition 3.

3.3 Degree

Given a Boolean function f, the *degree* $\deg(f)$ is the degree of the (unique) real multilinear polynomial $p : \mathbb{R}^n \to \mathbb{R}$ that satisfies $p(X) = f(X)$ for all $X \in \{0,1\}^n$. Degree has a known combinatorial characterization that will be useful to us:[7]

Proposition 6 (folklore) *Given a d-dimensional subcube S in $\{0,1\}^n$, let S_0, S_1 be the subsets of S with even and odd Hamming weight respectively (thus $|S_0| = |S_1| = 2^{d-1}$). Also, given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, call f "parity-correlated" on S if*

$$|\{X \in S_0 : f(X) = 1\}| \neq |\{X \in S_1 : f(X) = 1\}|. \tag{7}$$

Then $\deg(f)$ equals the maximum dimension of a subcube on which f is parity-correlated.

It is not hard to see that $\deg(f) \leq \mathrm{D}(f)$ for all Boolean functions f. Combined with Theorems 2 and 4, this implies that $\mathrm{R}_2(f) = \Omega(\deg(f)^{1/3})$ and $\mathrm{Q}_2(f) = \Omega(\deg(f)^{1/6})$, as stated in Section 2.

[6] This improved the result of Buhrman et al. [7] that $\mathrm{D}(f) = O(\mathrm{Q}_0(f)^4)$, as well as the result of Aaronson [2] that $\mathrm{R}_0(f) = O(\mathrm{Q}_0(f)^3 \log n)$.

[7] A proof of this folklore fact can be found in many places; one example is Aaronson [1].

3.4 Sensitivity and Block Sensitivity

Given an input $X \in \{0,1\}^n$ and a subset $B \subseteq [n]$, let X^B denote X with all the bits in B flipped. Then for a Boolean function f, the *sensitivity* $\mathrm{s}^X(f)$ is the number of indices $i \in [n]$ such that $f\left(X^{\{i\}}\right) \neq f(X)$, while the *block sensitivity* $\mathrm{bs}^X(f)$ is the maximum number of pairwise-disjoint "blocks" $B_1, \ldots, B_k \subseteq [n]$ that can be found such that $f\left(X^{B_j}\right) \neq f(X)$ for all $j \in [k]$. We then define

$$\mathrm{s}(f) := \max_{X \in \{0,1\}^n} \mathrm{s}^X(f), \qquad \mathrm{bs}(f) := \max_{X \in \{0,1\}^n} \mathrm{bs}^X(f). \tag{8}$$

Clearly $\mathrm{s}(f) \leq \mathrm{bs}(f)$. The famous *Sensitivity Conjecture* (see Hatami et al. [14] for a survey) asserts that the gap between $\mathrm{s}(f)$ and $\mathrm{bs}(f)$ is never more than polynomial:[8]

Conjecture 7 (Sensitivity Conjecture) *There exists a polynomial p such that $\mathrm{bs}(f) \leq p(\mathrm{s}(f))$ for all f.*

Nisan and Szegedy [17] showed that $\mathrm{bs}(f) \leq 2\deg(f)^2$ (recently improved by Tal [22] to $\mathrm{bs}(f) \leq \deg(f)^2$), while Beals et al. [5] showed that $\deg(f) \leq \mathrm{bs}(f)^3$.[9] Thus, degree and block sensitivity are polynomially related. This implies that Conjecture 7 is equivalent to the conjecture that sensitivity is polynomially related to degree.

3.5 AND/OR Tree

A particular Boolean function of interest to us will be the *complete binary AND/OR tree*. Assume $n = 2^d$; then this function is defined recursively as follows:

$$\mathrm{T}_0(x) := x, \tag{9}$$

$$\mathrm{T}_d(x_1, \ldots, x_n) := \begin{cases} \mathrm{T}_{d-1}\left(x_1, \ldots, x_{n/2}\right) \mathrm{AND}\, \mathrm{T}_{d-1}\left(x_{n/2+1}, \ldots, x_n\right) & \text{if } d > 0 \text{ is odd,} \\ \mathrm{T}_{d-1}\left(x_1, \ldots, x_{n/2}\right) \mathrm{OR}\, \mathrm{T}_{d-1}\left(x_{n/2+1}, \ldots, x_n\right) & \text{if } d > 0 \text{ is even.} \end{cases} \tag{10}$$

It is not hard to see that

$$\mathrm{D}(\mathrm{T}_d) = \deg(\mathrm{T}_d) = 2^d = n. \tag{11}$$

By contrast, Saks and Wigderson [19] proved the following.

Theorem 8 (Saks-Wigderson [19]) $\mathrm{R}_0(\mathrm{T}_d) = O\left(\left(\frac{1+\sqrt{33}}{4}\right)^d\right) = O(n^{0.753\ldots})$.

Saks and Wigderson [19] also proved a matching lower bound of $\mathrm{R}_0(\mathrm{T}_d) = \Omega(n^{0.753\ldots})$, while Santha [20] proved that $\mathrm{R}_2(\mathrm{T}_d) = \Omega(n^{0.753\ldots})$ even for bounded-error algorithms. Note that T_d gives the largest known gap between $\mathrm{D}(f)$ and $\mathrm{R}_2(f)$ for any total Boolean function f.

Recently, building on the breakthrough quantum walk algorithm for game-tree evaluation [11] (see also [4]), Ambainis et al. [3] proved the following.

Theorem 9 (Ambainis et al. [3]) $\mathrm{Q}_0(\mathrm{T}_d) = O(\sqrt{n}\log^2 n)$.

By comparison, it is not hard to show (by reduction from PARITY) that $\mathrm{Q}_2(\mathrm{T}_d) = \Omega(\sqrt{n})$. Once again, Theorem 9 gives the largest known gap between $\mathrm{D}(f)$ and $\mathrm{Q}_0(f)$ for any total f.[10]

Finally, the following fact will be useful to us.

[8] Rubinstein [18] showed that $\mathrm{bs}(f)$ *can* be quadratically larger than $\mathrm{s}(f)$.

[9] This follows immediately from their result that $\mathrm{D}(f) \leq \mathrm{bs}(f)^3$, which improved on the bound $\mathrm{D}(f) \leq \mathrm{bs}(f)^4$ due to Nisan [16], and which they then combined with the result $\mathrm{Q}_2(f) = \Omega(\sqrt{\mathrm{bs}(f)})$ to prove Theorem 4, that $\mathrm{D}(f) = O(\mathrm{Q}_2(f)^6)$.

[10] It improves slightly on an earlier result of Buhrman et al. [7], who showed that for every $\varepsilon > 0$, there exists an f such that $\mathrm{Q}_0(f) = O(\mathrm{D}(f)^{1/2+\varepsilon})$. For Q_2, we can do slightly better ($\mathrm{Q}_2(f) = O(\sqrt{\mathrm{D}(f)})$) by just taking f to be the OR function.

Proposition 10 *Let $n = 2^d$. The number of inputs $X \in \{0,1\}^n$ such that $\mathrm{T}_d(X) = \textsc{Par}(X)$ is exactly $2^{n-1}+1$ if d is even, and exactly $2^{n-1}-1$ if d is odd.*

Proof. This is most easily proved by switching to the Fourier representation. Let

$$\mathrm{T}_d^*(x_1,\ldots,x_n) := 1 - 2\,\mathrm{T}_d\left(\frac{1-x_1}{2},\ldots,\frac{1-x_n}{2}\right), \tag{12}$$

$$\textsc{Par}^*(X) := x_1 \cdots x_n. \tag{13}$$

Then the problem reduces to computing the correlation

$$C_d := \sum_{X\in\{0,1\}^n} \mathrm{T}_d^*(X)\,\textsc{Par}^*(X), \tag{14}$$

since

$$|\{X \in \{0,1\}^n : \mathrm{T}_d(X) = \textsc{Par}(X)\}| = 2^{n-1} + \frac{C_d}{2}. \tag{15}$$

Since every two distinct monomials have correlation 0, we in turn have

$$C_d = 2^n \alpha_d = -2\beta_d, \tag{16}$$

where α_d and β_d are the coefficients in front of the monomial $x_1 \cdots x_n$ in the polynomials $\mathrm{T}_d^*(x_1,\ldots,x_n)$ and $\mathrm{T}_d(x_1,\ldots,x_n)$ respectively. It is not hard to see, by inspection of the polynomial T_d and induction on d, that $\beta_d = -1$ if d is even (i.e., if the root node has the form $\mathrm{OR}(x,y) = x+y-xy$), and that $\beta_d = 1$ if d is odd (i.e., if the root node has the form $\mathrm{AND}(x,y) = xy$). Hence $C_d = 2$ if d is even, and $C_d = -2$ if d is odd. This completes the proof. ■

4 Algorithms for WEAK PARITY

We now prove our first result: that there exist nontrivial randomized and quantum algorithms for WEAK PARITY. For simplicity, we first consider the special case $\varepsilon = 2^{-n}$; later we will generalize to arbitrary ε.

Lemma 11 *We have*

$$\mathrm{Q}_2(\textsc{WeakPar}_{n,2^{-n}}) = O(\sqrt{n}), \tag{17}$$

$$\mathrm{R}_0(\textsc{WeakPar}_{n,2^{-n}}) = O(n^{0.754}), \tag{18}$$

$$\mathrm{Q}_0(\textsc{WeakPar}_{n,2^{-n}}) = O(\sqrt{n}\log^2 n). \tag{19}$$

Proof. For Q_2, observe that the OR function, $\mathrm{OR}(X)$, agrees with the parity of X on $2^{n-1}+1$ inputs $X \in \{0,1\}^n$: namely, all the inputs of odd Hamming weight, plus the input 0^n. Thus, simply computing $\mathrm{OR}(X)$ gives us an algorithm for $\textsc{WeakPar}_{n,\varepsilon}$ with $\varepsilon = 2^{-n}$. And of course, OR can be computed with bounded error in $O(\sqrt{n})$ quantum queries, using Grover's algorithm.

For R_0, assume for simplicity that n has the form 2^d; this will not affect the asymptotics. By Proposition 10, if d is even then the AND/OR tree $\mathrm{T}_d(X)$ agrees with $\textsc{Par}(X)$ on $2^{n-1}+1$ inputs X, while if d is odd then $1-\mathrm{T}_d(X)$ does. Either way, simply computing $\mathrm{T}_d(X)$ gives us an algorithm for $\textsc{WeakPar}_{n,2^{-n}}$. Furthermore, by Theorem 8, there is a zero-error randomized algorithm for $\mathrm{T}_d(X)$ that makes $O(n^{0.754})$ queries.

For Q_0, we also compute either $\mathrm{T}_d(X)$ or $1-\mathrm{T}_d(X)$ as our guess for $\textsc{Par}(X)$, except now we use the zero-error quantum algorithm of Theorem 9, which makes $O(\sqrt{n}\log^2 n)$ queries. ■

Next, we give a general strategy for converting a WEAK PARITY algorithm for small ε into an algorithm that works for larger ε, with the query complexity gradually increasing as ε does.

Lemma 12 *For all positive integers k, we have*

$$\mathrm{R}_2(\textsc{WeakPar}_{kn,\varepsilon}) \leq k \cdot \mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon}). \tag{20}$$

So in particular, suppose $\mathrm{R}_2(\textsc{WeakPar}_{n,1/f(n)}) \leq T(n)$. Then for all N and $\varepsilon > 0$,

$$\mathrm{R}_2(\textsc{WeakPar}_{N,\varepsilon}) \leq \frac{N \cdot T\left(f^{-1}(1/\varepsilon)\right)}{f^{-1}(1/\varepsilon)}. \tag{21}$$

Exactly the same holds if we replace R_2 by R_0, Q_2, or Q_0 throughout.

Proof. Let A be a randomized algorithm for $\textsc{WeakPar}_{n,\varepsilon}$, and let X be an input to $\textsc{WeakPar}$ of size kn. Then our strategy is to group the bits of X into n blocks $Y_1, \ldots, Y_n$ of k bits each, then run A on the input

$$\textsc{Par}(Y_1), \ldots, \textsc{Par}(Y_n), \tag{22}$$

and output whatever A outputs. If A made $T(n)$ queries originally, then this strategy can be implemented using $k \cdot T(n)$ queries: namely, k queries to the underlying input X every time A queries a bit $\textsc{Par}(Y_i)$. Furthermore, let $p(Z)$ be A's success probability on input $Z \in \{0,1\}^n$. Then the strategy succeeds whenever

$$\left|p\left(\textsc{Par}(Y_1), \ldots, \textsc{Par}(Y_n)\right) - \left(\textsc{Par}(Y_1) \oplus \cdots \oplus \textsc{Par}(Y_n)\right)\right| \leq \frac{1}{3}, \tag{23}$$

and by assumption, this occurs for at least a $1/2 + \varepsilon$ fraction of Z's.

The inequality (21) is just a rewriting of (20), if we make the substitutions $\varepsilon := 1/f(n)$ and $n := f^{-1}(1/\varepsilon)$ to get

$$\mathrm{R}_2(\textsc{WeakPar}_{f^{-1}(1/\varepsilon),\varepsilon}) \leq T\left(f^{-1}(1/\varepsilon)\right), \tag{24}$$

followed by $k := N/f^{-1}(1/\varepsilon)$. Finally, since we never used that A was classical or bounded-error, everything in the proof still works if we replace R_2 by R_0, Q_2, or Q_0 throughout. ■

Combining Lemmas 11 and 12 now easily gives us our upper bounds:

Theorem 13 *For all n and $\varepsilon \in [2^{-n}, 1/2]$, we have*

$$\mathrm{Q}_2(\textsc{WeakPar}_{n,\varepsilon}) = O\left(\frac{n}{\sqrt{\log 1/\varepsilon}}\right), \tag{25}$$

$$\mathrm{R}_0(\textsc{WeakPar}_{n,\varepsilon}) = O\left(\frac{n}{\log^{0.246} 1/\varepsilon}\right), \tag{26}$$

$$\mathrm{Q}_0(\textsc{WeakPar}_{n,\varepsilon}) = O\left(n \cdot \frac{(\log\log 1/\varepsilon)^2}{\sqrt{\log 1/\varepsilon}}\right). \tag{27}$$

We do not know any upper bound on $\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon})$ better than our upper bound on $\mathrm{R}_0(\textsc{WeakPar}_{n,\varepsilon})$.

As a final note, all of our algorithms actually satisfy a stronger property than the definition of Weak Parity requires. Namely, the algorithms all compute a total Boolean function $f(X)$ that agrees with $\textsc{Par}(X)$ on a $1/2 + \varepsilon$ fraction of inputs. This means, for example, that we can obtain a randomized algorithm that outputs $\textsc{Par}(X)$ with probability 1 on a $1/2 + \varepsilon$ fraction of inputs $X \in \{0,1\}^n$, and that halts after $O(n/\log^{0.246}(1/\varepsilon))$ queries in expectation on *every* input X (not just those inputs for which the algorithm succeeds). We can similarly obtain a quantum algorithm with expected query complexity $O\left(n \cdot \frac{(\log\log 1/\varepsilon)^2}{\sqrt{\log 1/\varepsilon}}\right)$ and the same success condition.

5 Lower Bound via Random Self-Reducibility

Our next result is a *lower* bound on the bounded-error randomized and quantum query complexities of WEAK PARITY. The lower bound matches our upper bounds in its dependence on n, though not in its dependence on ε.

Theorem 14 $\mathrm{Q}_2(\text{WEAKPAR}_{n,\varepsilon}) = \Omega\left(n/\log\left(1/\varepsilon\right)\right)$ *for all* $0 < \varepsilon < \frac{1}{2}$.

Proof. Let C be a quantum algorithm for $\text{WEAKPAR}_{n,\varepsilon}$ that never makes more than T queries. Using C, we will produce a new quantum algorithm C', which makes $O\left(T \log \frac{1}{\varepsilon}\right)$ queries, and which guesses $\text{PAR}(X)$ on *every* input $X \in \{0,1\}^n$ with probability stricter greater than $1/2$. But it is well-known that any quantum algorithm of the latter kind must make at least $n/2$ queries: in other words, that PARITY has unbounded-error quantum query complexity $n/2$ (this follows from the polynomial method [5]). Putting the two facts together, we conclude that

$$T = \Omega\left(\frac{n}{\log 1/\varepsilon}\right). \tag{28}$$

To produce C', the first step is simply to amplify C. Thus, let C^* be an algorithm that outputs the majority answer among $d \log 1/\varepsilon$ invocations of C. Then by a Chernoff bound, provided the constant d is sufficiently large,

$$\Pr_{X \in \{0,1\}^n}\left[\left|\Pr\left[C^*\left(X\right) \text{ accepts}\right] - \text{PAR}\left(X\right)\right| \leq \varepsilon\right] \geq \frac{1}{2} + \varepsilon. \tag{29}$$

Next, C' chooses a string $Y \in \{0,1\}^n$ uniformly at random and sets $Z := X \oplus Y$. It then runs C^* to obtain a guess b about $\text{PAR}(Z)$. Finally, C' outputs $\text{PAR}(Y) \oplus b$ as its guess for $\text{PAR}(X)$.

Clearly C' has the same quantum query complexity as C^*: it is easy to simulate a query to a bit z_i of Z, by querying the corresponding bit x_i of X and then XORing with y_i. Furthermore, notice that Z is uniformly random, regardless of X, and that if $b = \text{PAR}\left(Z\right)$ then $\text{PAR}(Y) \oplus b = \text{PAR}(X)$. It follows that C' succeeds with probability at least

$$\left(\frac{1}{2} + \varepsilon\right)\left(1 - \varepsilon\right) = \frac{1}{2} + \frac{\varepsilon}{2} - \varepsilon^2 > \frac{1}{2} \tag{30}$$

for every X, which is what we wanted to show. ■

Of course, Theorem 14 implies that $\mathrm{Q}_0(\text{WEAKPAR}_{n,\varepsilon})$, $\mathrm{R}_2(\text{WEAKPAR}_{n,\varepsilon})$, and $\mathrm{R}_0(\text{WEAKPAR}_{n,\varepsilon})$ are $\Omega\left(n/\log\left(1/\varepsilon\right)\right)$ as well. It is curious that we do not get any lower bounds for Q_0, R_2, or R_0 better than for Q_2.

It is, however, illuminating to see what happens if we run the reduction of Theorem 14, starting from the assumption that C is a *zero-error* randomized or quantum algorithm for $\text{WEAKPAR}_{n,\varepsilon}$. Suppose furthermore that C satisfies the same strong property that our zero-error algorithms from Section 4 satisfied: namely, the property that C halts after T queries in expectation on *every* input $X \in \{0,1\}^n$. In that case, one can skip the amplification step in Theorem 14, to produce an algorithm C' with the following properties:

(i) C halts after T queries in expectation on every input X, and

(ii) C guesses $\text{PAR}(X)$ with probability greater than $1/2$ on every input X.

Now, one might think the above would imply $T \geq n/2$ (regardless of ε), thereby contradicting our upper bounds from Section 4! However, the apparent paradox is resolved once we realize that the lower bound

of Beals et al. [5]—showing that $T \geq n/2$ queries are needed to guess $\text{PAR}(X)$ with probability greater than $1/2$ on every input X—says nothing about *expected* query complexity. And indeed, it is trivial to design an algorithm that guesses $\text{PAR}(X)$ with $1/2 + \varepsilon$ probability on every input X, using $2\varepsilon n$ queries in expectation. That algorithm just evaluates $\text{PAR}(X)$ (using n queries) with probability 2ε, and otherwise guesses randomly, without examining X at all!

6 Lower Bound via Sensitivity

Theorem 14 shows that our algorithms from Theorem 13 are close to optimal when ε is reasonably large. Unfortunately, though, Theorem 14 gives nothing when $\varepsilon = 2^{-n}$. Equivalently, it does not even rule out a randomized or quantum algorithm making a *constant* number of queries (!), that correctly decides PARITY on a subset of size $2^{n-1} + 1$. We conjecture that $n^{\Omega(1)}$ randomized or quantum queries are needed for the latter task, but we are unable to prove that conjecture—a state of affairs that Section 7 will help to explain. In this section, we at least prove that $\Omega(\log n)$ randomized queries and $\Omega(\sqrt{\log n})$ quantum queries are needed to solve WEAK PARITY for all $\varepsilon > 0$.

The key is a combinatorial quantity called $\Lambda(n)$, which was introduced by Chung, Füredi, Graham, and Seymour [9]. Abusing notation, we identify the set $\{0,1\}^n$ with the Boolean hypercube graph (where two vertices are adjacent if and only if they have Hamming weight 1), and also identify any subset $G \subseteq \{0,1\}^n$ with the induced subgraph of $\{0,1\}^n$ whose vertex set is G. Let $\Delta(G)$ be the maximum degree of any vertex in G. Then

$$\Lambda(n) := \min_{G \subseteq \{0,1\}^n \,:\, |G| = 2^{n-1}+1} \Delta(G) \tag{31}$$

is the minimum of $\Delta(G)$ over all induced subgraphs G of size $2^{n-1} + 1$.

The following proposition relates $\Lambda(n)$ to WEAK PARITY.

Proposition 15 $\text{R}_2(\text{WEAKPAR}_{n,\varepsilon}) = \Omega(\Lambda(n))$ *and* $\text{Q}_2(\text{WEAKPAR}_{n,\varepsilon}) = \Omega(\sqrt{\Lambda(n)})$ *for all* $\varepsilon > 0$.

Proof. Let U be an algorithm that decides PARITY (with bounded error probability) on a subset $A \subseteq \{0,1\}^n$. Then we claim that U must make $\Omega(\Delta(A))$ randomized or $\Omega(\sqrt{\Delta(A)})$ quantum queries, which is $\Omega(\Lambda(n))$ or $\Omega(\sqrt{\Lambda(n)})$ respectively if $|A| > 2^{n-1}$. To see this, let $X \in A$ be a vertex with degree $\Delta(A)$. Then PARITY, when restricted to X and its neighbors, already yields a Grover search instance of size $\Delta(A)$. But searching a list of N elements is well-known to require $\Omega(N)$ randomized or $\Omega(\sqrt{N})$ quantum queries [6]. ■

To build intuition, it is easy to find an induced subgraph $G \subseteq \{0,1\}^n$ such that $|G| = 2^{n-1}$ but $\Delta(G) = 0$: consider the set of all points with odd Hamming weight. But adding a single vertex to that G increases its maximum degree $\Delta(G)$ all the way to n. More generally, Chung et al. [9] were able to prove the following.[11]

Theorem 16 (Chung et al. [9]) *We have*

$$\Lambda(n) \geq \frac{1}{2}\log_2 n - \frac{1}{2}\log_2 \log_2 n + \frac{1}{2}. \tag{32}$$

[11] Chung et al.'s result is very closely related to an earlier result of Simon [21], which states that if $f : \{0,1\}^n \rightarrow \{0,1\}$ is a Boolean function depending on all n of its inputs, then $\text{s}(f) \geq \frac{1}{2}\log_2 n - \frac{1}{2}\log_2 \log_2 n + \frac{1}{2}$, where $\text{s}(f)$ is the sensitivity. However, neither Chung et al.'s result nor Simon's seems derivable as an immediate corollary of the other.

Combining Theorem 16 with Proposition 15 tells us immediately that

$$\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon}) = \Omega(\log n), \tag{33}$$

$$\mathrm{Q}_2(\textsc{WeakPar}_{n,\varepsilon}) = \Omega(\sqrt{\log n}) \tag{34}$$

for all $\varepsilon > 0$.

Now, the best-known *upper* bound on $\Lambda(n)$, also proved by Chung et al. [9], is $\sqrt{n}+1$, and it is conjectured that this is essentially tight. By Proposition 15, clearly a proof of that conjecture would imply

$$\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon}) = \Omega(\sqrt{n}), \tag{35}$$

$$\mathrm{Q}_2(\textsc{WeakPar}_{n,\varepsilon}) = \Omega\left(n^{1/4}\right) \tag{36}$$

for all $\varepsilon > 0$—and more generally, proving $\Lambda(n) \geq n^{\Omega(1)}$ would imply that $\mathrm{R}(\textsc{WeakPar}_{n,\varepsilon})$ and $\mathrm{Q}(\textsc{WeakPar}_{n,\varepsilon})$ are $n^{\Omega(1)}$.

Unfortunately, proving $\Lambda(n) \geq n^{\Omega(1)}$ will be challenging. To see why, recall the famous *Sensitivity Conjecture* (Conjecture 7), which says that $\mathrm{s}(f)$ is polynomially related to $\mathrm{bs}(f)$ (or equivalently, to $\deg(f)$). In 1992, Gotsman and Linial [13] showed that the Sensitivity Conjecture is equivalent to a statement about the maximum degrees of induced subgraphs of $\{0,1\}^n$:

Theorem 17 (Gotsman-Linial [13]) *Given any growth rate h, we have $\mathrm{s}(f) > h(\deg(f))$ for all Boolean functions $f : \{0,1\}^n \rightarrow \{0,1\}$, if and only if*

$$\max\{\Delta(G), \Delta(\{0,1\}^n \setminus G)\} \geq h(n) \tag{37}$$

for all subsets $G \subseteq \{0,1\}^n$ such that $|G| \neq 2^{n-1}$.

Notice that if $|G| \neq 2^{n-1}$, then

$$\max\{\Delta(G), \Delta(\{0,1\}^n \setminus G)\} \geq \Lambda(n). \tag{38}$$

To see this, choose whichever of G or $\{0,1\}^n \setminus G$ is larger, and then discard all but $2^{n-1}+1$ of its elements. Thus, any lower bound on Chung et al.'s combinatorial quantity $\Lambda(n)$ implies the same lower bound on the function $h(n)$ of Theorem 17. For example, if $\Lambda(n) \geq n^{\Omega(1)}$, then $\mathrm{s}(f) \geq \deg(f)^{\Omega(1)}$.

But this means that *any proof of $\Lambda(n) \geq n^{\Omega(1)}$ would imply the Sensitivity Conjecture!*[12] Thus, the conjecture $\Lambda(n) \geq n^{\Omega(1)}$ could be seen as a "common combinatorial core" of the WEAK PARITY and sensitivity versus block sensitivity questions.

As a final note, Andy Drucker (personal communication) found a self-contained proof for $\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon}) = \Omega(\log n)$, one that does not rely on $\Lambda(n)$, and that indeed achieves a better constant than the $\Lambda(n)$-based proof. We include this proof with Drucker's kind permission.

Proposition 18 $\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon}) \geq \log_2 n - O(1)$ *for all $\varepsilon > 0$.*

Proof. Suppose by contradiction that $\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon}) \leq \log_2 n - C$, and let U be a randomized algorithm that achieves the bound. Then we can think of U as just a probability distribution $\mathcal{D}$ over decision trees. Given a decision tree T, let $S_T \subseteq [n]$ be the set of all indices i such that the variable x_i appears anywhere

[12] Interestingly, we do not know the reverse implication.

in T. Then by assumption, each T in the support of $\mathcal{D}$ has depth at most $\log_2 n - C$, and therefore satisfies $|S_T| \leq 2^{\log_2 n - C} = n/2^C$. By averaging, it follows that there exists an $i \in [n]$ such that

$$\Pr_{T \sim \mathcal{D}} [i \in S_T] \leq 2^{-C}. \tag{39}$$

But this means that, for every $X \in \{0,1\}^n$, we must have

$$\left| \Pr [U \text{ accepts } X] - \Pr \left[U \text{ accepts } X^{\{i\}} \right] \right| \leq 2^{-C}, \tag{40}$$

where $X^{\{i\}}$ denotes X with the i^{th} bit flipped (as in Section 3.4). Hence, for (say) $C \geq 2$, either $\Pr [U \text{ accepts } X]$ fails to approximate $\textsc{Par}(X)$, or else $\Pr \left[U \text{ accepts } X^{\{i\}} \right]$ fails to approximate $\textsc{Par}\left(X^{\{i\}}\right)$. But this means that U weakly computes PARITY on at most 2^{n-1} inputs. ■

Interestingly, unlike with our argument based on $\Lambda(n)$, we do not know how to generalize Drucker's argument to prove any lower bound on *quantum* query complexity, nor do we know (even conjecturally) how to push the argument beyond $\Omega(\log n)$.

7 Connection to $\deg(f)$ vs. $\mathrm{Q}(f)$

In the last section, we identified a known combinatorial conjecture ($\Lambda(n) \geq n^{\Omega(1)}$) that would imply that the randomized and quantum query complexities of WEAK PARITY are $n^{\Omega(1)}$ for all $\varepsilon > 0$. However, since $\Lambda(n) \geq n^{\Omega(1)}$ would also imply the Sensitivity Conjecture, it will clearly be difficult to prove.

So could there be a *different* way to prove tight lower bounds for $\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon})$ and $\mathrm{Q}_2(\textsc{WeakPar}_{n,\varepsilon})$—a way that wouldn't require us to address any longstanding open problems about Boolean functions? Alas, in this section we largely close off that possibility. In particular, suppose we could prove a strong lower bound on $\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon})$. We will show that this would imply a better polynomial relationship between $\deg(f)$ and $\mathrm{R}_2(f)$ for *all* total Boolean functions f than is currently known. Similar statements hold for R_0, Q_2, and Q_0.

Theorem 19 *Given a constant c, suppose there exists a sequence of functions $\{f_n\}_{n \geq 1}$ such that $\deg(f_n) = n$ and $\mathrm{R}_2(f_n) = O(n^c)$. Then*

$$\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon}) = O\left(\frac{n}{\log^{1-c} 1/\varepsilon} \right). \tag{41}$$

The same holds if we replace R_2 by R_0, Q_2, or Q_0 in both instances.

Proof. We first show that $\mathrm{R}_2(\textsc{WeakPar}_{n,2^{-n}}) = O(n^c)$ in the special case $\varepsilon = 2^{-n}$; then we generalize to larger ε.

Observe that we can assume without loss of generality that each f_n has exactly n inputs. For otherwise, let p be the unique multilinear polynomial representing f_n; then choose a monomial m of p with degree n, and arbitrarily fix all bits that do not appear in m. This yields a subfunction f'_n with n inputs, $\deg(f'_n) = \deg(f_n) = n$, and $\mathrm{R}_2(f'_n) \leq \mathrm{R}_2(f_n)$.

Now by Proposition 6, the statement $\deg(f_n) = n$ is equivalent to the combinatorial statement

$$|\{X : f_n(X) = 1 \text{ and } \textsc{Par}(X) = 0\}| \neq |\{X : f_n(X) = 1 \text{ and } \textsc{Par}(X) = 1\}|. \tag{42}$$

This means that $f_n(X)$ either agrees or disagrees with $\textsc{Par}(X)$ on at least $2^{n-1}+1$ inputs X. By replacing f_n by $1 - f_n$, we can assume without loss of generality that the first case holds. Then if we run the algorithm

for f_n, it will make $O(n^c)$ queries and correctly decide PARITY on at least $2^{n-1}+1$ inputs, which was the desired result.

To generalize to arbitrary ε, we simply need to appeal to Lemma 12, which tells us that if

$$\mathrm{R}_2(\textsc{WeakPar}_{n,2^{-n}}) \leq T(n) = O(n^c), \tag{43}$$

then

$$\mathrm{R}_2(\textsc{WeakPar}_{N,\varepsilon}) \leq \frac{N \cdot T(\log_2 1/\varepsilon)}{\log_2 1/\varepsilon} = O\left(\frac{N}{\log^{1-c} 1/\varepsilon}\right). \tag{44}$$

Finally, since we never used that the algorithm was classical or bounded-error, everything in the proof still works if we replace R_2 by R_0, Q_2, or Q_0 throughout. ■

For clarity, let us state Theorem 19 in contrapositive form.

Corollary 20 *Suppose* $\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon}) = \Omega\left(n/\log^{1-c}(1/\varepsilon)\right)$. *Then for every Boolean function* f, *we have* $\mathrm{R}_2(f) = \Omega(\deg(f)^c)$ *(and similarly for* R_0, Q_2, *and* Q_0*).*

Plugging our $\Omega(n/\log(1/\varepsilon))$ lower bound on $\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon})$ (i.e., Theorem 14) into Corollary 20, we get only the trivial lower bound $\mathrm{R}_2(f) = \Omega(1)$ for non-constant f. On the other hand, suppose we could prove that

$$\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon}) = \Omega\left(\frac{n}{\log^{2/3} 1/\varepsilon}\right). \tag{45}$$

Then Corollary 20 would reproduce the result of Nisan [16] that $\mathrm{R}_2(f) = \Omega(\deg(f)^{1/3})$ for all Boolean functions f. Likewise, if we could prove that

$$\mathrm{Q}_2(\textsc{WeakPar}_{n,\varepsilon}) = \Omega\left(\frac{n}{\log^{5/6} 1/\varepsilon}\right), \tag{46}$$

then Corollary 20 would reproduce the result of Beals et al. [5] that $\mathrm{Q}_2(f) = \Omega(\deg(f)^{1/6})$ for all f. Any *better* lower bounds than those on $\mathrm{R}_2(\textsc{WeakPar}_{n,\varepsilon})$ or $\mathrm{Q}_2(\textsc{WeakPar}_{n,\varepsilon})$ would imply better general lower bounds on $\mathrm{R}_2(f)$ or $\mathrm{Q}_2(f)$ than are currently known. So for example, suppose we could prove that

$$\mathrm{Q}_2(\textsc{WeakPar}_{n,\varepsilon}) = \Omega\left(\frac{n}{\sqrt{\log 1/\varepsilon}}\right); \tag{47}$$

i.e., that the quantum algorithm of Theorem 13 was optimal. Then we would prove the longstanding conjecture that $\mathrm{Q}_2(f) = \Omega(\sqrt{\deg(f)})$ for all Boolean functions f (the bound being saturated when $f = \mathrm{OR}$).

One might wonder: can we also go in the other direction, and use the known polynomial relationships between $\deg(f)$ and query complexity measures to prove better lower bounds for WEAK PARITY? At present, we cannot quite do that, but we can do something close. Recall from Section 1 that, in defining WEAK PARITY, we did not impose any requirement that our algorithm's acceptance probability $p(X)$ approximate a total Boolean function. However, suppose we *do* impose that requirement. Then we can easily show the following:

Proposition 21 *Fix any* $\varepsilon > 0$. *Suppose an algorithm's acceptance probability must satisfy* $p(X) \in [0,1/3] \cup [2/3,1]$ *for all* $X \in \{0,1\}^n$. *Then any randomized algorithm for* $\textsc{WeakPar}_{n,\varepsilon}$ *makes* $\Omega\left(n^{1/3}\right)$ *queries, and any quantum algorithm makes* $\Omega\left(n^{1/6}\right)$ *queries.*

Suppose further that the acceptance probability must satisfy $p(X) \in \{0,1\}$ *for all* X. *Then any randomized algorithm for* $\textsc{WeakPar}_{n,\varepsilon}$ *makes* $\Omega\left(n^{1/2}\right)$ *queries in expectation, and any quantum algorithm makes* $\Omega\left(n^{1/3}\right)$ *queries.*

Proof. Let $f(X) = \lfloor p(X) \rceil$ be the total Boolean function approximated by $p(X)$. Then since the algorithm solves WEAK PARITY,

$$|\{X : f(X) = 1 \text{ and } \textsc{Par}(X) = 1\}| > |\{X : f(X) = 1 \text{ and } \textsc{Par}(X) = 0\}|. \tag{48}$$

So by Proposition 6, we must have $\deg(f) = \mathrm{D}(f) = n$. By Theorems 2, 4, and 5, this means that

$$\mathrm{R}_2(f) = \Omega(\mathrm{D}(f)^{1/3}) = \Omega(n^{1/3}), \tag{49}$$

$$\mathrm{Q}_2(f) = \Omega(\mathrm{D}(f)^{1/6}) = \Omega(n^{1/6}), \tag{50}$$

$$\mathrm{R}_0(f) = \Omega(\mathrm{D}(f)^{1/2}) = \Omega(n^{1/2}), \tag{51}$$

$$\mathrm{Q}_0(f) = \Omega(\mathrm{D}(f)^{1/3}) = \Omega(n^{1/3}). \tag{52}$$

■

8 Weak Algorithms for Other Functions

In this section, we begin the investigation of weak algorithms for Boolean functions other than PARITY. Our main result is the following:

Theorem 22 *Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be any Boolean function. Then we can guess $f(X)$ on $2^{n-1}+1$ inputs X using a bounded-error quantum algorithm that makes $O(\sqrt{n})$ queries, a zero-error randomized algorithm that makes $O\left(n^{0.754}\right)$ queries, or a zero-error quantum algorithm that makes $O(\sqrt{n}\log^2 n)$ queries.*

Proof. Assume without loss of generality that n has the form 2^d for $d \geq 2$ (this will not affect the asymptotics). There are two cases. First, suppose f is *unbalanced*: that is,

$$|\{X : f(X) = 1\}| \neq 2^{n-1}. \tag{53}$$

Then we can trivially agree with f on at least $2^{n-1}+1$ inputs X, by either always outputting 0 or always outputting 1.

Second, suppose f is balanced. Note that the OR function outputs 1 on an odd number of inputs X. It follows that $|\{X : f(X) = \mathrm{OR}(X)\}|$ must be odd as well, and cannot equal 2^{n-1}. So either $\mathrm{OR}(X)$ or $1 - \mathrm{OR}(X)$ must agree with $f(X)$ on at least $2^{n-1}+1$ inputs X. Thus, Grover's algorithm gives us the desired bounded-error quantum algorithm making $O(\sqrt{n})$ queries.

For the other algorithms, recall Proposition 10, which tells us that the AND/OR tree T_d also outputs 1 on an odd number of inputs X. So by the same reasoning as above, either $\mathrm{T}_d(X)$ or $1 - \mathrm{T}_d(X)$ must agree with $f(X)$ on at least $2^{n-1}+1$ inputs X. Hence, we can use Theorem 8 to get the desired zero-error randomized algorithm making $O\left(n^{0.754}\right)$ queries, and use Theorem 9 to get a zero-error quantum algorithm making $O(\sqrt{n}\log^2 n)$ queries. ■

Interestingly, unlike for PARITY, for arbitrary f it is unclear whether we can get any nontrivial algorithms when ε is larger than 2^{-n}. Our proof of Lemma 12 relied essentially on PARITY's downward self-reducibility, so it does not generalize to other functions.

Note also that we cannot hope to prove any general *lower* bound on the weak query complexity of f, even assuming that f is balanced and that its quantum query complexity is $\Omega(n)$. As a counterexample, let $\mathrm{H}(X) = 1$ if X has Hamming weight at least $2n/3$ and $\mathrm{H}(X) = 0$ otherwise; then consider

$$f(x_1, \ldots, x_n) := x_1 \oplus \mathrm{H}(x_2, \ldots, x_n). \tag{54}$$

9 Open Problems

The obvious problem is to close the gaps between our upper and lower bounds on the query complexity of WEAK PARITY. We have seen that this problem is intimately related to longstanding open problems in the study of Boolean functions, including polynomial degree versus query complexity, the Sensitivity Conjecture, and lower-bounding Chung et al.'s [9] combinatorial quantity $\Lambda(n)$. Perhaps the surprising relationships among these problems could motivate renewed attacks.

In the meantime, can we reprove our $\Omega(n/\log(1/\varepsilon))$ lower bound for WEAK PARITY (or better yet, improve it) *without* exploiting PARITY's random self-reducibility? How far can we get by using (say) the polynomial or adversary methods directly? It would also be great if we could say something about weak algorithms for functions other than PARITY, beyond what we said in Section 8: for example, what happens if $\varepsilon > 2^{-n}$?

Let us end with three more specific questions:

(1) Do we ever get faster algorithms for WEAK PARITY, if we drop the constraint that the algorithm's acceptance probability approximates a total Boolean function f?[13]

(2) Can we "interpolate" between our two different ways of proving lower bounds for WEAK PARITY, to get better lower bounds than $\Omega(\log n)$ or $\Omega(\sqrt{\log n})$ when ε is small but still larger than 2^{-n}?

(3) Can we show that an $n^{\Omega(1)}$ lower bound for WEAK PARITY is directly implied by the Sensitivity Conjecture, rather than the related conjecture that $\Lambda(n) \geq n^{\Omega(1)}$?

10 Acknowledgments

We thank Ronald de Wolf, both for helpful discussions and for his comments on a draft; Andy Drucker for allowing us to include Proposition 18; and the anonymous reviewers for their suggestions.

References

[1] S. Aaronson. Algorithms for Boolean function query properties. *SIAM J. Comput.*, 32(5):1140–1157, 2003.

[2] S. Aaronson. Quantum certificate complexity. In *Proc. IEEE Conference on Computational Complexity*, pages 171–178, 2003. ECCC TR03-005, quant-ph/0210020.

[3] A. Ambainis, A. Childs, F. Le Gall, and S. Tani. The quantum query complexity of certification. *Quantum Information and Computation*, 10(3-4), 2010. arXiv:0903.1291.

[4] A. Ambainis, A. M. Childs, B. W. Reichardt, R. Špalek, and S. Zhang. Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. In *Proc. IEEE FOCS*, 2007. quant-ph/0703015 and arXiv:0704.3628.

[5] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. Earlier version in IEEE FOCS 1998, pp. 352-361. quant-ph/9802049.

[6] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. quant-ph/9701001.

[13] If the answer is no, then of course Proposition 21 already gives us good lower bounds for WEAK PARITY when $\varepsilon = 2^{-n}$.

[7] H. Buhrman, R. Cleve, R. de Wolf, and Ch. Zalka. Bounds for small-error and zero-error quantum algorithms. In *Proc. IEEE FOCS*, pages 358–368, 1999. cs.CC/9904019.

[8] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Comput. Sci.*, 288:21–43, 2002.

[9] F. R. K. Chung, Z. Füredi, R. L. Graham, and P. Seymour. On induced subgraphs of the cube. *J. Comb. Theory Ser. A*, 49(1):180–187, 1988.

[10] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proc. Roy. Soc. London*, A439:553–558, 1992.

[11] E. Farhi, J. Goldstone, and S. Gutmann. A quantum algorithm for the Hamiltonian NAND tree. *Theory of Computing*, 4(1):169–190, 2008. quant-ph/0702144.

[12] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. A limit on the speed of quantum computation in determining parity. *Phys. Rev. Lett.*, 81:5442–5444, 1998. quant-ph/9802045.

[13] C. Gotsman and N. Linial. The equivalence of two problems on the cube. *J. Comb. Theory Ser. A*, 61(1):142–146, 1992.

[14] P. Hatami, R. Kulkarni, and D. Pankratov. Variations on the sensitivity conjecture. *Theory of Computing Library Graduate Surveys*, 4, 2011.

[15] G. Midrijanis. On randomized and quantum query complexities. quant-ph/0501142, 2005.

[16] N. Nisan. CREW PRAMs and decision trees. *SIAM J. Comput.*, 20(6):999–1007, 1991.

[17] N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994.

[18] D. Rubinstein. Sensitivity vs. block sensitivity of Boolean functions. *Combinatorica*, 15(2):297–299, 1995.

[19] M. Saks and A. Wigderson. Probabilistic Boolean decision trees and the complexity of evaluating game trees. In *Proc. IEEE FOCS*, pages 29–38, 1986.

[20] M. Santha. On the Monte-Carlo decision tree complexity of read-once formulae. *Random Structures and Algorithms*, 6(1):75–87, 1995.

[21] H.-U. Simon. A tight $\Omega(\log\log n)$ bound on the time for parallel RAM's to compute non-degenerate Boolean functions. In *Foundations of Computing Theory*, volume 158, pages 439–444. Springer-Verlag, 1983.

[22] A. Tal. Properties and applications of Boolean function composition. In *Proc. Innovations in Theoretical Computer Science (ITCS)*, pages 441–454, 2013. ECCC TR12-163.