Quantum Lower Bound for Recursive Fourier Sampling

Scott Aaronson
Institute for Advanced Study, Princeton
aaronson@ias.edu

Abstract

One of the earliest quantum algorithms was discovered by Bernstein and Vazirani, for a problem called Recursive Fourier Sampling. This paper shows that the Bernstein-Vazirani algorithm is not far from optimal. The moral is that the need to "uncompute" garbage can impose a fundamental limit on efficient quantum computation. The proof introduces a new parameter of Boolean functions called the "nonparity coefficient," which might be of independent interest.

Like a classical algorithm, a quantum algorithm can solve problems recursively by calling itself as a subroutine. When this is done, though, the algorithm typically needs to call itself *twice* for each subproblem to be solved. The second call's purpose is to uncompute 'garbage' left over by the first call, and thereby enable interference between different branches of the computation. Of course, a factor of 2 increase in running time hardly seems like a big deal, when set against the speedups promised by quantum computing. The problem is that these factors of 2 multiply, with each level of recursion producing an additional factor. Thus, one might wonder whether the uncomputing step is really necessary, or whether a cleverly designed algorithm might avoid it. This paper gives the first nontrivial example in which recursive uncomputation is provably necessary.

The example concerns a long-neglected problem called *Recursive Fourier Sampling* (henceforth RFS), which was introduced by Bernstein and Vazirani [5] in 1993 to prove the first oracle separation between BPP and BQP. Many surveys on quantum computing pass directly from the Deutsch-Jozsa algorithm [8] to the dramatic results of Simon [14] and Shor [13], without even mentioning RFS. There are two likely reasons for this neglect. First, the RFS problem seems artificial. It was introduced for the sole purpose of proving an oracle result, and is unlike all other problems for which a quantum speedup is known. (I will define RFS in Section 1; but for now, it involves a tree of depth $\log n$, where each vertex is labeled with a function to be evaluated via a Fourier transform.) Second, the speedup for RFS is only quasipolynomial (n versus $n^{\log n}$), rather than exponential as for the period-finding and hidden subgroup problems.

Nevertheless, I believe that RFS merits renewed attention—for it serves as an important link between quantum computing and the ideas of classical complexity theory. One reason is that, although other problems in BQP—such as the factoring, discrete logarithm, and 'shifted Legendre symbol' problems [16]—are thought to be classically intractable, these problems are quite low-level by complexity-theoretic standards. They, or their associated decision problems, are in $\mathsf{NP} \cap \mathsf{coNP}$.[1] By contrast, Bernstein and Vazirani [5] showed that, as an oracle problem, RFS lies outside NP and even MA (the latter result is unpublished, though not difficult). Subsequently Watrous [17] gave an oracle A, based on an unrelated problem, for which $\mathsf{BQP}^A \not\subset \mathsf{MA}^A$.[2] Also, Green and Pruim [10] gave an oracle B for which $\mathsf{BQP}^B \not\subset \mathsf{P}^{\mathsf{NP}^B}$. However, Watrous' problem was shown by Babai [3] to be in AM, while Green and Pruim's problem is in BPP. Thus, neither problem can be used to place BQP outside higher levels of the polynomial hierarchy PH.

On the other hand, Umesh Vazirani and others have conjectured that RFS is not in PH, from which it would follow that there exists an oracle A relative to which $\mathsf{BQP}^A \not\subset \mathsf{PH}^A$. Proving this is, in my view, one of the central open problems in quantum complexity theory. Its solution seems likely to require novel techniques for constant-depth circuit lower bounds.[3]

[1] For the shifted Legendre symbol problem, this is true assuming a number-theoretic conjecture of Boneh and Lipton [6].

[2] Actually, to place BQP outside MA relative to an oracle, it suffices to consider the complement of Simon's problem ("Does $f(x) = f(x \oplus s)$ only when $s = 0$?").

[3] For the RFS function can be represented by a low-degree real polynomial—this follows from the existence of a polynomial-time quantum algorithm for RFS, together with the result of Beals et al. [4] relating quantum algorithms to low-degree polynomials.

In this paper I examine the RFS problem from a different angle. Could Bernstein and Vazirani's quantum algorithm for RFS be improved even further, to give an *exponential* speedup over the classical algorithm? And could we use RFS, not merely to place BQP outside of PH relative to an oracle, but to place it outside of PH with (say) a logarithmic number of alternations?

My answer to both questions is a strong 'no.' I study a large class of variations on RFS, and show that all of them fall into one of two classes:

(1) a trivial class, for which there exists a classical algorithm making only one query, or

(2) a nontrivial class, for which any quantum algorithm needs $2^{\Omega(h)}$ queries, where h is the height of the tree to be evaluated. (By comparison, the Bernstein-Vazirani algorithm uses 2^h queries, because of its need to uncompute garbage recursively at each level of the tree.)

Since n^h queries always suffice classically, this dichotomy theorem implies that the speedup afforded by quantum computers is at most quasipolynomial. It also implies that (nontrivial) RFS is solvable in quantum polynomial time only when $h = O(\log n)$.

The plan is as follows. In Section 1, I define the RFS problem, and give Bernstein and Vazirani's quantum algorithm for solving it. In Section 2, I use the adversary method of Ambainis [2] to prove a lower bound on the quantum query complexity of any RFS variant. This bound, however, requires a parameter that I call the "nonparity coefficient" to be large. Intuitively, given a Boolean function $g : \{0,1\}^n \rightarrow \{0,1\}$, the nonparity coefficient measures how far g is from being the parity of some subset of its input bits—not under the uniform distribution over inputs (the standard assumption in Fourier analysis), but under an adversarial distribution. The crux of the argument is that *either* the nonparity coefficient is zero (meaning the RFS variant in question is trivial), or else it is bounded below by a positive constant. This statement is proved in Section 2, and seems like it might be of independent interest. Section 3 concludes with some open problems.

1 Preliminaries

In ordinary Fourier sampling, we are given oracle access to a Boolean function $A : \{0,1\}^n \rightarrow \{0,1\}$, and are promised that there exists a secret string $s \in \{0,1\}^n$ such that $A(x) = s \cdot x \pmod 2$ for all x. The problem is to find s—or rather, since we need a problem with Boolean output, the problem is to return $g(s)$, where $g : \{0,1\}^n \rightarrow \{0,1\}$ is some known Boolean function. We can think of $g(s)$ as the "hard-core bit" of s, and can assume that g itself is efficiently computable, or else that we are given access to an oracle for g.

To obtain a height-2 recursive Fourier sampling tree, we simply compose this problem. That is, we are no longer given direct access to $A(x)$, but instead are promised that $A(x) = g(s_x)$, where $s_x \in \{0,1\}^n$ is the secret string for another Fourier sampling problem. A query then takes the form (x, y), and produces as output $A_x(y) = s_x \cdot y \pmod 2$. As before, we are promised that there exists an s such that $A(x) = s \cdot x \pmod 2$ for all x, meaning that the s_x strings must be chosen consistent with this promise. Again we must return $g(s)$.

Continuing, we can define height-h recursive Fourier sampling, or RFS_h, recursively as follows. We are given oracle access to a function $A(x_1, \ldots, x_h)$ for all $x_1, \ldots, x_h \in \{0,1\}^n$, and are promised that

(1) for each fixed x_1^*, $A(x_1^*, x_2, \ldots, x_h)$ is an instance of RFS_{h-1} on $x_2, \ldots, x_h$, having answer bit $b(x_1^*) \in \{0,1\}$; and

(2) there exists a secret string $s \in \{0,1\}^n$ such that $b(x_1^*) = s \cdot x_1^* \pmod 2$ for each x_1^*.

Again the answer bit to be returned is $g(s)$. Note that g is assumed to be the same everywhere in the tree—though using the techniques in this paper, it would be straightforward to generalize to the case of different g's. As an example that will be used later, we could take $g(s) = g_{\text{mod}\,3}(s)$, where $g_{\text{mod}\,3}(s) = 0$ if $|s| \equiv 0 \pmod 3$ and $g_{\text{mod}\,3}(s) = 1$ otherwise, and $|s|$ denotes the Hamming weight of s. We do not want to take g to be the parity of s, for if we did then $g(s)$ could be evaluated using a single query. To see this, observe that if x is the all-1's string, then $s \cdot x \pmod 2$ is the parity of s.

As a result, the circuit lower bound technique of Razborov [12] and Smolensky [15], which is based on the nonexistence of low-degree polynomials, seems unlikely to work. Even the random restriction method of Furst et al. [9] can be related to low-degree polynomials, as shown by Linial et al. [11].

By an 'input,' I will mean a complete assignment for the RFS oracle (that is, $A(x_1,\ldots,x_h)$ for all $x_1,\ldots,x_h$). I will sometimes refer also to an 'RFS tree,' where each vertex at distance ℓ from the root has a label $x_1,\ldots,x_\ell$. If $\ell = h$ then the vertex is a leaf; otherwise it has 2^n children, each with a label $x_1,\ldots,x_\ell,x_{\ell+1}$ for some $x_{\ell+1}$. The subtrees of the tree just correspond to the sub-instances of RFS.

Bernstein and Vazirani [5] showed that $\mathrm{RFS}_{\log n}$, or RFS with height $\log n$ (all logarithms are base 2), is solvable on a quantum computer in time polynomial in n. I include a proof for completeness. Let $A = (A_n)_{n\geq 0}$ be an oracle that, for each n, encodes an instance of $\mathrm{RFS}_{\log n}$ whose answer is Ψ_n. Then let L_A be the unary language $\{0^n : \Psi_n = 1\}$.

Lemma 1 *$L_A \in \mathsf{EQP}^A \subseteq \mathsf{BQP}^A$ for any choice of A.*

Proof. RFS_1 can be solved exactly in four queries, with no garbage bits left over. The algorithm is as follows: first prepare the state

$$2^{-n/2}\sum_{x\in\{0,1\}^n} |x\rangle\,|A(x)\rangle\,,$$

using one query to A. Then apply a phase flip conditioned on $A(x)=1$, and uncompute $A(x)$ using a second query, obtaining

$$2^{-n/2}\sum_{x\in\{0,1\}^n} (-1)^{A(x)}\,|x\rangle\,.$$

Then apply a Hadamard gate to each bit of the $|x\rangle$ register. It can be checked that the resulting state is simply $|s\rangle$. One can then compute $|s\rangle\,|g(s)\rangle$ and uncompute $|s\rangle$ using two more queries to A, to obtain $|g(s)\rangle$. To solve $RFS_{\log n}(n)$, we simply apply the above algorithm recursively at each level of the tree. The total number of queries used is $4^{\log n} = n^2$.

One can further reduce the number of queries to $2^{\log n} = n$ by using the "one-call kickback trick," described by Cleve et al. [7]. Here one prepares the state

$$2^{-n/2}\sum_{x\in\{0,1\}^n} |x\rangle \otimes \frac{|1\rangle - |0\rangle}{\sqrt{2}}$$

and then exclusive-OR's $A(x)$ into the second register. This induces the desired phase $(-1)^{A(x)}$ without the need to uncompute $A(x)$. However, one still needs to uncompute $|s\rangle$ after computing $|g(s)\rangle$. ■

A remark on notation: to avoid confusion with subscripts, I denote the i^{th} bit of string x by $x[i]$.

2 Quantum Lower Bound

In this section I prove a lower bound on the quantum query complexity of RFS. Crucially, the bound should hold for any nontrivial one-bit function of the secret strings, not just a specific function such as $g_{\mathrm{mod}\,3}(s)$ defined in Section 1. Let RFS_h^g be height-h recursive Fourier sampling in which the problem at each vertex is to return $g(s)$. The following notion turns out to be essential.

Definition 2 *Given a Boolean function $g : \{0,1\}^n \to \{0,1\}$ (partial or total), the nonparity coefficient $\mu(g)$ is the largest μ^* for which there exist distributions D_0 over the 0-inputs of g, and D_1 over the 1-inputs, such that for all $z \in \{0,1\}^n$, all 0-inputs $\widehat{s}_0$, and all 1-inputs $\widehat{s}_1$, we have*

$$\Pr_{s_0\in D_0, s_1\in D_1}\left[s_0\cdot z \equiv \widehat{s}_1\cdot z\,(\mathrm{mod}\,2) \;\;\vee\;\; s_1\cdot z \equiv \widehat{s}_0\cdot z\,(\mathrm{mod}\,2)\right] \geq \mu^*.$$

Loosely speaking, the nonparity coefficient is high if there exist distributions over 0-inputs and 1-inputs that make g far from being a parity function of a subset of input bits. The following proposition develops some intuition about $\mu(g)$.

Proposition 3

(i) $\mu(g) \leq 3/4$ for all nonconstant g.

(ii) $\mu(g) = 0$ *if and only if* g *can be written as the parity (or the NOT of the parity) of a subset* B *of input bits.*

Proof.

(i) Given any $s_0 \neq \widehat{s}_1$ and $s_1 \neq \widehat{s}_0$, a uniform random z will satisfy

$$\Pr_z \left[s_0 \cdot z \not\equiv \widehat{s}_1 \cdot z \pmod 2 \quad \wedge \quad s_1 \cdot z \not\equiv \widehat{s}_0 \cdot z \pmod 2 \right] \geq \frac{1}{4}.$$

(If $s_0 \oplus \widehat{s}_1 = s_1 \oplus \widehat{s}_0$ then this probability will be $1/2$; otherwise it will be $1/4$.) So certainly there is a fixed choice of z that works for random s_0 and s_1.

(ii) For the 'if' direction, take $z[i] = 1$ if and only if $i \in B$, and choose $\widehat{s}_0$ and $\widehat{s}_1$ arbitrarily. This ensures that $\mu^* = 0$. For the 'only if' direction, if $\mu(g) = 0$, we can choose D_0 to have support on all 0-inputs, and D_1 to have support on all 1-inputs. Then there must be a z such that $s_0 \cdot z$ is constant as we range over 0-inputs, and $s_1 \cdot z$ is constant as we range over 1-inputs. Take $i \in B$ if and only if $z[i] = 1$.

■

If $\mu(g) = 0$, then RFS_h^g is easily solvable using a single classical query. Theorem 5 will show that for all g (partial or total),

$$\mathrm{Q}_2\left(\mathrm{RFS}_h^g\right) = \Omega\left(\left(\frac{1}{1-\mu(g)}\right)^{h/2}\right),$$

where $\mathrm{Q}_2(f)$ is the bounded-error quantum query complexity of f as defined by Beals et al. [4]. In other words, any RFS problem with μ bounded away from 0 requires a number of queries exponential in the tree height h.

However, there is an essential further part of the argument, which restricts the values of $\mu(g)$ itself. Suppose there existed a family $\{g_n\}$ of 'pseudoparity' functions: that is, $\mu(g_n) > 0$ for all n, yet $\mu(g_n) = O(1/\log n)$. Then the best bound obtainable from Theorem 5 would be $\Omega\left((1+1/\log n)^{h/2}\right)$, suggesting that $\mathrm{RFS}_{\log^2 n}^g$ might still be solvable in quantum polynomial time. On the other hand, it would be unclear a priori how to solve $\mathrm{RFS}_{\log^2 n}^g$ classically with a logarithmic number of alternations. Theorem 7 will rule out this scenario by showing that pseudoparity functions do not exist: if $\mu(g) < 0.146$ then g is a parity function, and hence $\mu(g) = 0$.

The theorem of Ambainis that we need is his "most general" lower bound from [2], which he introduced to show that the quantum query complexity of inverting a permutation is $\Omega(\sqrt{n})$. That theorem can be stated as follows.

Theorem 4 (Ambainis) *Let* $X \subseteq f^{-1}(0)$ *and* $Y \subseteq f^{-1}(1)$ *be sets of inputs to function* f*. Let* $R(x,y) \geq 0$ *be a symmetric real-valued relation function, and for* $x \in X$*,* $y \in Y$*, and index* i*, let*

$$\theta(x,i) = \frac{\sum_{y^* \in Y \,:\, x[i] \neq y^*[i]} R(x,y^*)}{\sum_{y^* \in Y} R(x,y^*)},$$

$$\theta(y,i) = \frac{\sum_{x^* \in X \,:\, x^*[i] \neq y[i]} R(x^*,y)}{\sum_{y^* \in Y} R(x^*,y)},$$

where the denominators are all nonzero. Then $\mathrm{Q}_2(f) = O(1/\upsilon)$ *where*

$$\upsilon = \max_{x \in X,\ y \in Y,\ i \,:\, R(x,y)>0,\ x[i] \neq y[i]} \sqrt{\theta(x,i)\,\theta(y,i)}.$$

We are now ready to prove a lower bound for RFS.

Theorem 5 *For all* g *(partial or total),* $\mathrm{Q}_2\left(\mathrm{RFS}_h^g\right) = \Omega\left((1-\mu(g))^{-h/2}\right)$.

Proof. Let X be the set of all 0-inputs to RFS_h^g, and let Y be the set of all 1-inputs. We will weight the inputs using the distributions D_0, D_1 from the definition of the nonparity coefficient $\mu(g)$. For all $x \in X$, let $p(x)$ be the product, over all vertices v in the RFS tree for x, of the probability of the secret string s at v, if s is drawn from $D_{g(s)}$ (where we condition on v's output bit, $g(s)$). Next, say that $x \in X$ and $y \in Y$ *differ minimally* if, for all vertices v of the RFS tree, the subtrees rooted at v are identical in x and in y whenever the answer bit $g(s)$ at v is the same in x and in y. If x and y differ minimally, then we will set $R(x,y) = p(x)\,p(y)$; otherwise we will set $R(x,y) = 0$. Clearly $R(x,y) = R(y,x)$ for all $x \in X, y \in Y$. Furthermore, we claim that $\theta(x,i)\,\theta(y,i) \leq (1-\mu(g))^h$ for all x, y that differ minimally and all i such that $x[i] \neq y[i]$. For suppose $y^* \in Y$ is chosen with probability proportional to $R(x,y^*)$, and $x^* \in X$ is chosen with probability proportional to $R(x^*,y)$. Then $\theta(x,i)\,\theta(y,i)$ equals the probability that we would notice the switch from x to y^* by monitoring i, times the probability that we would notice the switch from y to x^*.

Let v_j be the j^{th} vertex along the path in the RFS tree from the root to the leaf vertex i, for all $j \in \{1,\ldots,h\}$. Also, let $z_j \in \{0,1\}^n$ be the label of the edge between v_{j-1} and v_j, and let $s_{x,j}$ and $s_{y,j}$ be the secret strings at v_j in x and y respectively. Then since x and y differ minimally, we must have $g(s_{x,j}) \neq g(s_{y,j})$ for all j—for otherwise the subtrees rooted at v_j would be identical, which contradicts the assumption $x[i] \neq y[i]$. So we can think of the process of choosing y^* as first choosing a random $s'_{x,1}$ from D_1 so that $1 = g\left(s'_{x,1}\right) \neq g(s_{x,1}) = 0$, then choosing a random $s'_{x,2}$ from $D_{1-g(s_{x,2})}$ so that $g\left(s'_{x,2}\right) \neq g(s_{x,2})$, and so on. Choosing x^* is analogous, except that whenever we used D_0 in choosing y^* we use D_1, and vice versa. Since the $2h$ secret strings $s_{x,1},\ldots,s_{x,h},s_{y,1},\ldots,s_{y,h}$ to be updated are independent of one another, it follows that

$$\begin{aligned}
\Pr[y^*[i] \neq x[i]] \Pr[x^*[i] \neq y[i]] &= \prod_{j=1}^{h} \Pr_{s \in D_0}[s \cdot z_j \not\equiv s_{x,j} \cdot z_j] \Pr_{s \in D_1}[s \cdot z_j \not\equiv s_{y,j} \cdot z_j] \\
&\leq \prod_{j=1}^{h} (1-\mu(g)) \\
&= (1-\mu(g))^h
\end{aligned}$$

by the definition of $\mu(g)$. Therefore

$$\text{Q}_2\left(\text{RFS}_h^g\right) = \Omega\left((1-\mu(g))^{-h/2}\right)$$

by Theorem 4. ■

Before continuing further, let me show that there is a natural, explicit choice of g—the function $g_{\text{mod}\,3}(s)$ from Section 1—for which the nonparity coefficient is almost 3/4. Thus, for $g = g_{\text{mod}\,3}$, the algorithm of Lemma 1 is essentially optimal.

Proposition 6 $\mu(g_{\text{mod}\,3}) = 3/4 - O(1/n)$.

Proof. Let $n \geq 6$. Let D_0 be the uniform distribution over all s with $|s| = 3\lfloor n/6 \rfloor$ (so $g_{\text{mod}\,3}(s) = 0$); likewise let D_1 be the uniform distribution over s with $|s| = 3\lfloor n/6 \rfloor + 2$ ($g_{\text{mod}\,3}(s) = 1$). We consider only the case of s drawn from D_0; the D_1 case is analogous. We will show that for any z,

$$\left| \Pr_{s \in D_0}[s \cdot z \equiv 0] - \frac{1}{2} \right| = O\left(\frac{1}{n}\right)$$

(all congruences are mod 2). The theorem then follows, since by the definition of the nonparity coefficient, given any z the choices of $s_0 \in D_0$ and $s_1 \in D_1$ are independent.

Assume without loss of generality that $1 \leq |z| \leq n/2$ (if $|z| > n/2$, then replace z by its complement). We apply induction on $|z|$. If $|z| = 1$, then clearly

$$\Pr[s \cdot z \equiv 0] = 3\lfloor n/6 \rfloor / n = \frac{1}{2} \pm O\left(\frac{1}{n}\right).$$

For $|z| \geq 2$, let $z = z_1 \oplus z_2$, where z_2 contains only the rightmost 1 of z and z_1 contains all the other 1's. Suppose the proposition holds for $|z|-1$. Then

$$\begin{aligned}
\Pr[s \cdot z \equiv 0] = {} & \Pr[s \cdot z_1 \equiv 0] \Pr[s \cdot z_2 \equiv 0 | s \cdot z_1 \equiv 0] + \\
& \Pr[s \cdot z_1 \equiv 1] \Pr[s \cdot z_2 \equiv 1 | s \cdot z_1 \equiv 1],
\end{aligned}$$

where

$$\Pr[s \cdot z_1 \equiv 0] = \frac{1}{2} + \alpha, \;\; \Pr[s \cdot z_1 \equiv 1] = \frac{1}{2} - \alpha$$

for some $|\alpha| = O(1/n)$. Furthermore, even conditioned on $s \cdot z_1$, the expected number of 1's in s outside of z_1 is $(n - |z_1|)/2 \pm O(1)$ and they are uniformly distributed. Therefore

$$\Pr[s \cdot z_2 \equiv b | s \cdot z_1 \equiv b] = \frac{1}{2} + \beta_b$$

for some $|\beta_0|, |\beta_1| = O(1/n)$. So

$$\begin{aligned} \Pr[s \cdot z \equiv 0] &= \frac{1}{2} + \frac{\beta_0}{2} + \alpha\beta_0 - \frac{\beta_1}{2} - \alpha\beta_1 \\ &= \frac{1}{2} \pm O\left(\frac{1}{n}\right). \end{aligned}$$

■

Finally it must be shown that pseudoparity functions do not exist. That is, if g is too close to a parity function for the bound of Theorem 5 to apply, then g actually *is* a parity function, from which it follows that RFS_h^g admits an efficient classical algorithm.

Theorem 7 *Suppose $\mu(g) < 0.146$. Then g is a parity function (equivalently, $\mu(g) = 0$).*

Proof. By linear programming duality, there exists a joint distribution $\mathcal{D}$ over $z \in \{0,1\}^n$, 0-inputs $\widehat{s}_0 \in g^{-1}(0)$, and 1-inputs $\widehat{s}_1 \in g^{-1}(1)$, such that for all $s_0 \in g^{-1}(0)$ and $s_1 \in g^{-1}(1)$,

$$\Pr_{(z,\widehat{s}_0,\widehat{s}_1)\in\mathcal{D}}[s_0 \cdot z \equiv \widehat{s}_1 \cdot z \pmod 2 \;\; \vee \;\; s_1 \cdot z \equiv \widehat{s}_0 \cdot z \pmod 2] < \mu(g).$$

Furthermore $\widehat{s}_0 \cdot z \not\equiv \widehat{s}_1 \cdot z \pmod 2$, since otherwise we could violate the hypothesis by taking $s_0 = \widehat{s}_0$ or $s_1 = \widehat{s}_1$. It follows that there exists a joint distribution $\mathcal{D}'$ over $z \in \{0,1\}^n$ and $b \in \{0,1\}$ such that

$$\Pr_{(z,b)\in\mathcal{D}'}[s \cdot z \equiv b \pmod 2] > 1 - \mu(g)$$

for all $s \in g^{-1}(0)$, and

$$\Pr_{(z,b)\in\mathcal{D}'}[s \cdot z \not\equiv b \pmod 2] > 1 - \mu(g)$$

for all $s \in g^{-1}(1)$. But this implies that g is a bounded-error threshold function of parity functions. More precisely, there exist probabilities p_z, summing to 1, as well as $b_z \in \{0,1\}$ such that for all $s \in \{0,1\}^n$,

$$\Psi(s) = \sum_{z\in\{0,1\}^n} p_z((s \cdot z) \oplus b_z) \text{ is } \begin{cases} > 1 - \mu(g) & \text{if } g(s) = 1 \\ < \mu(g) & \text{if } g(s) = 0. \end{cases}$$

We will consider $\mathrm{var}(\Psi)$, the variance of the above quantity $\Psi(s)$ if s is drawn uniformly at random from $\{0,1\}^n$. First, if $p_z \geq 1/2$ for any z, then $g(s) = (s \cdot z) \oplus b_z$ is a parity function and hence $\mu(g) = 0$. So we can assume without loss of generality that $p_z < 1/2$ for all z. Then since s is uniform, for each $z_1 \neq z_2$ we know that $(s \cdot z_1) \oplus b_{z_1}$ and $(s \cdot z_2) \oplus b_{z_2}$ are pairwise independent $\{0,1\}$ random variables, both with expectation $1/2$. So

$$\mathrm{var}(\Psi) = \frac{1}{4}\sum_z p_z^2 < \frac{1}{4}\left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = \frac{1}{8}.$$

On the other hand, since $\Psi(s)$ is always less than μ or greater than $1 - \mu$,

$$\mathrm{var}(\Psi) > \left(\frac{1}{2} - \mu\right)^2.$$

Combining,

$$\mu > \frac{2 - \sqrt{2}}{4} > 0.146.$$

■

3 Open Problems

An intriguing open problem is whether Theorem 5 can be proved using the polynomial method of Beals et al. [4], rather than the adversary method of Ambainis [2]. It is known that one can lower-bound polynomial degree in terms of block sensitivity, or the maximum number of disjoint changes to an input that change the output value. The trouble is that the RFS function has block sensitivity 1—the "sensitive blocks" of each input tend to have small intersection, but are not disjoint. For this reason, I implicitly used the "quantum certificate complexity" as defined in [1] rather than block sensitivity to prove a lower bound.

I believe the constant of Theorem 7 can be improved. The smallest nonzero $\mu(g)$ value I know of is attained when $n = 2$ and $g = \mathrm{OR}(s[1], s[2])$:

Proposition 8 $\mu(\mathrm{OR}) = 1/3$.

Proof. First, $\mu(\mathrm{OR}) \geq 1/3$, since D_1 can choose $s[1]\,s[2]$ to be 01, 10, or 11 each with probability 1/3; then for any $z \neq 0$ and the unique 0-input $\widehat{s}_0 = 00$, we have $s_1 \cdot z \not\equiv \widehat{s}_0 \cdot z$ with probability at most 2/3. Second, $\mu(\mathrm{OR}) \leq 1/3$, since applying linear programming duality, we can let the pair $(z, \widehat{s}_1)$ equal $(01, 01)$, $(10, 10)$, or $(11, 10)$ each with probability 1/3. Then $0 \equiv s_0 \cdot z \not\equiv \widehat{s}_1 \cdot z \equiv 1$ always, and for any 1-input s_1, we have $s_1 \cdot z \equiv 1 \not\equiv \widehat{s}_0 \cdot z$ with probability 2/3. ■

Finally, I conjecture that uncomputation is unavoidable not just for RFS but for many other recursive problems, such as game-tree evaluation. Formally, the conjecture is that the quantum query complexity of evaluating a game tree increases exponentially with depth as the number of leaves is held constant, even if there is at most one winning move per vertex (so that the tree can be evaluated with zero probability of error).

4 Acknowledgments

I thank Lisa Hales, Umesh Vazirani, Ronald de Wolf, and the anonymous reviewers for helpful comments. This work was done while I was a graduate student at UC Berkeley, supported by an NSF Graduate Fellowship.

References

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

[2] A. Ambainis. Quantum lower bounds by quantum arguments. *J. Comput. Sys. Sci.*, 64:750–767, 2002. Earlier version in ACM STOC 2000. quant-ph/0002066.

[3] L. Babai. Bounded round interactive proofs in finite groups. *SIAM J. Discrete Math*, 5(1):88–111, 1992.

[4] 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. quant-ph/9802049.

[5] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997. First appeared in ACM STOC 1993.

[6] D. Boneh and R. Lipton. Algorithms for black box fields and their application to cryptography. In *Proceedings of CRYPTO*, volume 109, pages 283–297. Lecture Notes in Computer Science, 1996.

[7] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proc. Roy. Soc. London*, A454:339–354, 1998. quant-ph/9708016.

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

[9] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial time hierarchy. *Math. Systems Theory*, 17:13–27, 1984.

[10] F. Green and R. Pruim. Relativized separation of EQP from P^{NP}. *Inform. Proc. Lett.*, 80(5):257–260, 2001.

[11] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993.

[12] A. A. Razborov. Lower bounds for the size of circuits of bounded depth with basis $\{\&, \oplus\}$. *Mathematicheskie Zametki*, 41(4):598–607, 1987. English translation in *Math. Notes. Acad. Sci. USSR* 41(4):333–338, 1987.

[13] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. Earlier version in IEEE FOCS 1994. quant-ph/9508027.

[14] D. Simon. On the power of quantum computation. In *Proc. IEEE FOCS*, pages 116–123, 1994.

[15] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. ACM STOC*, pages 77–82, 1987.

[16] W. van Dam, S. Hallgren, and L. Ip. Algorithms for some hidden shift problems. In *Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 489–498, 2003. quant-ph/0211140.

[17] J. Watrous. Succinct quantum proofs for properties of finite groups. In *Proc. IEEE FOCS*, pages 537–546, 2000. cs.CC/0009002.