Quantum Information and Computation, Vol. 3, No. 2 (2003) 165–174
© Rinton Press

QUANTUM LOWER BOUND FOR RECURSIVE FOURIER SAMPLING

SCOTT AARONSON[a]
Computer Science Department, University of California, Berkeley
Berkeley CA 94720 USA

Received November 11, 2002
Revised January 17, 2003

We revisit the oft-neglected 'recursive Fourier sampling' (RFS) problem, introduced by Bernstein and Vazirani to prove an oracle separation between BPP and BQP. We show that the known quantum algorithm for RFS is essentially optimal, despite its seemingly wasteful need to uncompute information. This implies that, to place BQP outside of PH [log] relative to an oracle, one would need to go outside the RFS framework. Our proof argues that, given any variant of RFS, either the adversary method of Ambainis yields a good quantum lower bound, or else there is an efficient classical algorithm. This technique may be of independent interest.

Keywords: quantum computing, lower bounds, query complexity

Communicated by: R Jozsa & J Watrous

1 Introduction

Quantum computing first gained notoriety with Shor's factoring algorithm [1], which built on earlier work of Simon [2]. It is sometimes claimed that, before Simon and Shor's breakthroughs, there was no credible evidence that quantum computers can yield a superpolynomial speedup over classical ones. For, although the Deutsch-Jozsa algorithm [3] was discovered earlier than Simon's algorithm, the former gives a speedup only in the exact case, not the bounded-error case.

However, there is an oft-neglected algorithm that appeared after Deutsch and Jozsa's and before Simon's. This is the recursive Fourier sampling (henceforth RFS) algorithm, which was used by Bernstein and Vazirani [4] to obtain the first oracle separation between BPP and BQP.[b] 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. (We will define RFS in Section 2; 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 (i.e. n versus $n^{\log n}$), rather than exponential as for the period-finding and hidden subgroup problems.

Nevertheless, we believe RFS is due for a comeback. Despite (or perhaps because of) its artificiality, the problem serves as an important link between quantum computing and the techniques of classical complexity theory. One reason is that, although other problems in

[a]Email: aaronson@cs.berkeley.edu. Supported by an NSF Graduate Fellowship and by the Defense Advanced Research Projects Agency (DARPA) and Air Force Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-01-2-0524.

[b]For the definitions of complexity classes used in this paper, see [5]. For background on quantum computing and on the quantum oracle model, see [4, 6].

BQP—such as the factoring, discrete logarithm, and 'shifted Legendre symbol' problems [7]—are thought to be classically intractable, these problems are quite low-level by complexity-theoretic standards. They, or their associated decision problems, are in $\mathsf{NP} \cap \mathsf{coNP}$.[c] By contrast, Bernstein and Vazirani [4] showed that, as an oracle problem, *RFS* lies outside NP and even MA (the latter result is unpublished, though not difficult). Subsequently Watrous [9] gave an oracle A, based on an unrelated problem, for which $\mathsf{BQP}^A \not\subseteq \mathsf{MA}^A$.[d] Also, Green and Pruim [10] gave an oracle B for which $\mathsf{BQP}^B \not\subseteq \mathsf{P}^{\mathsf{NP}^B}$. However, Watrous' problem was shown by Babai [11] 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.

On the other hand, Vazirani [12] and others have conjectured that *RFS* is not in PH, from which it would follow that there exists an oracle A relative to which $\mathsf{BQP}^A \not\subseteq \mathsf{PH}^A$. Proving this is, in our view, one of the central open problems of quantum complexity theory. Its solution is likely to require radically new techniques for circuit lower bounds.[e]

Here we address a different question. If BQP is indeed outside of PH relative to an oracle, how many alternations *are* needed to simulate it classically? The *RFS* problem is trivially in PH [log] (that is, PH with logarithmic alternations). Yet several researchers independently expressed to us the hope that, if *RFS* were modified a bit, a quantum algorithm might be able to handle more than $\log n$ levels of recursion. One might thereby obtain an oracle problem that is in BQP, yet outside PH [log] or even PH [*poly* log]. We were thus led to investigate whether this hope could be realized.

Our conclusion is negative. In this paper we examine a broad class of variations on *RFS*, and show that each of them has either (1) a good quantum lower bound or (2) an efficient classical algorithm—there are no 'in-between' cases. It follows that, to place BQP outside of PH [log] relative to an oracle, one would need to go outside the *RFS* framework as formulated in this paper. That rather arcane-sounding assertion has a broader context. 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. The need to uncompute is what causes the standard *RFS* algorithm to have query complexity 2^h, where h is the height of a tree to be evaluated. One might wonder, though, whether the uncomputing step is really necessary, or whether a cleverly designed algorithm might avoid it. Our result gives, to our knowledge, the first nontrivial example for which recursive uncomputation *is* provably necessary. We conjecture that uncomputation is needed as well for other recursive problems, such as game-tree evaluation.[f]

The plan is as follows. In Section 2 we define the *RFS* problem and show that it lies in BQP. In Section 3, we use the adversary method of Ambainis [18] to prove a lower bound on the quantum query complexity of any *RFS* variant. This bound, however, requires a parameter called the *nonparity coefficient* to be large. The crux of our argument, proven in Section 3, is that the nonparity coefficient is always above a certain threshold, unless the *RFS* variant is trivial (i.e. admits an efficient classical algorithm). In Section 4 we sketch a

[c]For the shifted Legendre symbol problem, this is true assuming a number-theoretic conjecture of Boneh and Lipton [8].

[d]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 $s = 0$?").

[e]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. [6] relating quantum algorithms to low-degree polynomials. As a result, the circuit lower bound technique of Razborov [13] and Smolensky [14], which is based on the nonexistence of low-degree polynomials, seems unlikely to work. Even the random restriction method of Furst et al. [15] can be related to low-degree polynomials, as shown by Linial et al. [16].

[f]Formally, we conjecture that the quantum query complexity of evaluating a game tree increases with depth as the number of leaves is held constant, even if there is at most one winning move per vertex (and hence no need to bound error probability). This would match an upper bound for quantum game-tree search due to Høyer and de Wolf [17].

generalization of this argument to RFS variants involving partial functions. We conclude in Section 5.

2 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. We are asked to return $g(s)$, where $g : \{0,1\}^n \rightarrow \{0,1\}$ is a known Boolean function. We may assume that either g is efficiently computable, or else we are given access to an oracle for g.

Then, 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 we take g to be the same everywhere in the tree. Allowing different g's at different vertices would, we believe, complicate the results without adding anything conceptually new. As an example that will be used later, we could take $g(s) = g_{\bmod 3}(s)$, where $g_{\bmod 3}(s) = 0$ if $|s| \equiv 0 \pmod 3$ and $g_{\bmod 3}(s) = 1$ 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.

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

Bernstein and Vazirani [4] showed that $RFS_{\log n}$, or RFS with height $\log n$ (all logarithms are base 2), is solvable on a quantum computer in time polynomial in n. We include a proof for completeness. Let $A = (A_n)_{n \geq 0}$ be an oracle that, for each n, encodes an instance of $RFS_{\log n}$ whose answer is Ψ_n. Then let L_A be the unary language $\{0^n : \Psi_n = 1\}$.

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

Proof: $RFS_1(n)$ 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. [19]. 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, we denote the i^{th} bit of string x by $x[i]$.

3 Quantum Lower Bound

In this section we 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 (say) the function $g_{\text{mod } 3}(s)$ defined in Section 2. Let $g : \{0,1\}^n \rightarrow \{0,1\}$ be a function of the secret strings, and let RFS_h^g be height-h RFS in which the problem at each vertex is to return $g(s)$. The following notion turns out to be essential.

Definition 1 *The nonparity coefficient $\mu(g)$ of g is the maximum μ^* for which the following holds. There exist distributions D_0, D_1 over $g^{-1}(0)$ and $g^{-1}(1)$ respectively such that for all $z \in \{0,1\}^n \setminus \{0^n\}$, $\widehat{s}_0 \in g^{-1}(0)$ and $\widehat{s}_1 \in g^{-1}(1)$,*

$$\begin{aligned} &\Pr_{s_0 \in D_0}[s_0 \cdot z \equiv \widehat{s}_1 \cdot z \,(\text{mod}\, 2)] \geq \mu^* \text{ and} \\ &\Pr_{s_1 \in D_1}[s_1 \cdot z \equiv \widehat{s}_0 \cdot z \,(\text{mod}\, 2)] \geq \mu^*. \end{aligned}$$

One can verify that $\mu(g) \in [0, 1/2]$ for all g. Intuitively, the nonparity coefficient is high if the parity of some subset of bits of s is never strongly correlated with $g(s)$. For example, we show in Proposition 3 that $\mu(g_{\text{mod } 3}(s)) = 1/2 - O(1/n)$. At the other extreme, $\mu(g) = 0$ if and only if g is a parity function or a restriction thereof; note that RFS_h^g is solvable in a single classical query if $\mu(g) = 0$. In Theorem 2 we show that for all g,

$$Q_2(RFS_h^g) = \Omega\left(\left(\frac{1}{1 - \mu(g)}\right)^h\right),$$

where Q_2 is bounded-error quantum query complexity as defined by Beals et al. [6]. In other words, any RFS problem with μ bounded away from 0 requires a number of queries exponential in the tree height h. To show this we use the adversary method of Ambainis [18].

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 lower bound we could obtain from Theorem 2 would be $\Omega\left((1 + 1/\log n)^h\right)$, suggesting that $RFS_{\log^2 n}^g$ might still be solvable in quantum polynomial time. On the other hand, it would be unclear a priori how to solve $RFS_{\log^2 n}^g$ classically with a logarithmic number of alternations. In Theorem 4 we rule out this scenario by showing that pseudoparity functions do not exist: if $\mu(g) < 0.0732$ then g is a parity function, and hence $\mu(g) = 0$.

The theorem of Ambainis [18] that we need is the following.

Theorem 1 (Ambainis) *Let $f : \{0,1\}^N \rightarrow \{0,1\}$ be a Boolean function and X, Y two sets of inputs such that $f(x) \neq f(y)$ if $x \in X$ and $y \in Y$. Let $R \subset X \times Y$ be such that*

(1) For every $x \in X$, there are at least m different $y \in Y$ such that $(x,y) \in R$.

(2) For every $y \in Y$, there are at least m' different $x \in X$ such that $(x,y) \in R$.

(3) For every $x \in X$ and $i \in \{1,\dots,N\}$, there are at most l different $y \in Y$ such that $(x,y) \in R$ and $x[i] \neq y[i]$.

(4) For every $y \in Y$ and $i \in \{1,\dots,N\}$, there are at most l' different $x \in X$ such that $(x,y) \in R$ and $x[i] \neq y[i]$.

Then $Q_2(f) = \Omega\left(\sqrt{mm'/ll'}\right)$.

We will actually use a weighted variant of Theorem 1:

Corollary 1 *Let f, X, Y, and R be as described in Theorem 1, and let $\mathcal{X}$ and $\mathcal{Y}$ be probability distributions over X and Y respectively. Let*

$$\alpha = \max_{x\in X,\ i\in\{1,\dots,N\}} \Pr_{y\in\mathcal{Y}\,:\,(x,y)\in R}\left[x[i] \neq y[i]\right],$$
$$\beta = \max_{y\in Y,\ i\in\{1,\dots,N\}} \Pr_{x\in\mathcal{X}\,:\,(x,y)\in R}\left[x[i] \neq y[i]\right].$$

Then $Q_2(f) = \Omega\left(1/\sqrt{\alpha\beta}\right)$.

To see that Corollary 1 follows from Theorem 1, note that we could simulate probabilities by, for example, letting X and Y contain multiple copies of each input. We can then obtain the lower bound for RFS.

Theorem 2 *For all total g, $Q_2\left(RFS_h^g\right) = \Omega\left((1-\mu(g))^{-h}\right)$.*

Proof: Let X be the set of all inputs to RFS_h with output 0, and Y the set of all inputs with output 1. Say that $x \in X$ and $y \in Y$ *differ minimally* if, at every vertex v of the RFS tree, if the answer bit $g(s)$ at v is the same for x and y, then the subtrees rooted at v are identical in x and y. We will take $(x,y) \in R$ if and only if x and y differ minimally.

We weight the inputs using the distributions D_0, D_1 from the definition of the nonparity coefficient. In particular, the probability of input x being drawn from $\mathcal{X}$ is the product, over all vertices v in the tree, 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, $g(s)$). Fix $i \in \{1,\dots,n\}$ and a pair $(\widehat{x},\widehat{y}) \in R$. Then

$$\Pr_{x\in\mathcal{X}\,:\,(x,\widehat{y})\in R}\left[x[i] \neq \widehat{y}[i]\right] \leq (1-\mu)^h,$$
$$\Pr_{y\in\mathcal{Y}\,:\,(\widehat{x},y)\in R}\left[\widehat{x}[i] \neq y[i]\right] \leq (1-\mu)^h.$$

This says the following. Suppose we choose x from $\mathcal{X}$, conditioned on it differing minimally from $\widehat{y}$. Then the probability that x differs from $\widehat{y}$ at any particular bit i decreases exponentially in h. To see this, observe that $x[i] = \widehat{y}[i]$ if $s_x \cdot z \equiv s_{\widehat{y}} \cdot z \pmod 2$ at *any* vertex v along the path from the root to i, where s_x and $s_{\widehat{y}}$ are the secret strings of x and $\widehat{y}$ respectively at v. Similar reasoning applies to the probability that y differs from $\widehat{x}$.

This implies that, in Corollary 1, $\alpha \leq (1-\mu)^h$ and $\beta \leq (1-\mu)^h$. Therefore

$$Q_2\left(RFS_h^g\right) = \Omega\left(\sqrt{1/\alpha\beta}\right) = \Omega\left((1-\mu)^{-h}\right).$$

□

We now show that there is a natural choice of g—the function $g_{\mathrm{mod}\,3}(s)$ defined in Section 2—for which the nonparity coefficient is almost 1/2. Thus, for $g = g_{\mathrm{mod}\,3}$, the algorithm of Lemma 1 is essentially optimal.

Proposition 3 $\mu\left(g_{\mathrm{mod}\,3}\right)=1/2-O\left(1/n\right)$.

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

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

(all congruences are mod 2), from which the proposition follows.

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

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

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

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

where

$$\Pr\left[s\cdot z_1\equiv 0\right]=1/2+\alpha,\ \ \Pr\left[s\cdot z_1\equiv 1\right]=1/2-\alpha$$

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

$$\Pr\left[s\cdot z_2\equiv b|s\cdot z_1\equiv b\right]=1/2+\beta_b$$

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

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

□

Finally we need to show that pseudoparity functions do not exist. That is, if g is too close to a parity function for the bound of Theorem 2 to apply, then g actually *is* a parity function, from which it follows that RFS_h^g admits an efficient classical algorithm. We need the following lemma about finite groups.

Lemma 2 *Partition a finite group G into (A,B). Then either for all $a\in A$, there exist $b_1,b_2\in B$ such that $a=b_1\cdot b_2$, or for all $b\in B$, there exist $a_1,a_2\in A$ such that $b=a_1\cdot a_2$.*

Proof: Suppose without loss of generality that $1\in A$. Then either $|A|-1\leq|B|$ or $|A|-1\geq|B|$. Suppose the former case, and choose an $a\in A$. Let the elements of G be vertices of a directed graph, with an edge from g_1 to g_2 if and only if $g_2=g_1^{-1}a$. Then, since each vertex has indegree and outdegree 1, the graph is a union of disjoint cycles. Furthermore, after we remove the cycle $(1,a)$, there are more vertices in B than in A. Hence there exists an edge from some $b_1\in B$ to $b_2\in B$, and $b_1b_2=a$. Similarly, if $|A|-1\geq|B|$, then for each $b\in B$, after we remove the cycle $(1,b)$ there are more vertices in A than in B. □

We can now prove the main result.

Theorem 4 *Suppose g is total and $\mu\left(g\right)<0.0732$. Then $\mu\left(g\right)=0$ (or equivalently, g is a parity function).*

Proof: By linear programming duality, there exists a distribution D over $z,\widehat{s}\in\{0,1\}^n$, $z\neq 0^n$ such that for all $s\in\{0,1\}^n$,

$$\Pr_{(z,\widehat{s})\in D:g(s)\neq g(\widehat{s})}\left[s\cdot z\equiv\widehat{s}\cdot z\left(\mathrm{mod}\,2\right)\right]<\mu.$$

It follows that there exist distributions $\widehat{D}_0$ and $\widehat{D}_1$ over $z \in \{0,1\}^n$, as well as functions $b_0(z), b_1(z) \in \{0,1\}$ (which we can take to be deterministic without loss of generality), such that for all $s_0 \in g^{-1}(0)$ and $s_1 \in g^{-1}(1)$,

$$\Pr_{z \in \widehat{D}_0}[s_0 \cdot z \equiv b_0(z) \pmod 2] > 1 - \mu,$$
$$\Pr_{z \in \widehat{D}_1}[s_1 \cdot z \equiv b_1(z) \pmod 2] > 1 - \mu.$$

Observe that if s is drawn uniformly at random from $\{0,1\}^n$, then for any $b_i(z)$,

$$\Pr_s\left[\Pr_{z \in \widehat{D}_1}[s \cdot z \equiv b_i(z) \pmod 2] > \frac{1}{2}\right] \le \frac{1}{2}$$

by symmetry. Hence $\left|g^{-1}(i)\right| \le 2^{n-1}$ for $i \in \{0,1\}$, and therefore $\left|g^{-1}(0)\right| = \left|g^{-1}(1)\right| = 2^{n-1}$.

Now identify $\{0,1\}^n$ with the group $\mathbb{Z}_2^n$. By Lemma 2, either for all $s_0 \in g^{-1}(0)$ there exist $s_\alpha, s_\beta \in g^{-1}(1)$ such that $s_0 = s_\alpha \oplus s_\beta$, or for all $s_1 \in g^{-1}(1)$ there exist $s_\alpha, s_\beta \in g^{-1}(0)$ such that $s_1 = s_\alpha \oplus s_\beta$. Suppose the former without loss of generality; then for all $s_0 \in g^{-1}(0)$,

$$s_0 \cdot z \equiv (s_\alpha \cdot z) \oplus (s_\beta \cdot z) \pmod 2.$$

Here we have the parity of two binary random variables, $s_\alpha \cdot z$ and $s_\beta \cdot z$, which are not necessarily independent, but both of which take the value $b_1(z)$ with probability greater than $1 - \mu$. Hence, by the union bound,

$$\Pr_{z \in \widehat{D}_1}[s_0 \cdot z \equiv 1 \pmod 2] < 2\mu.$$

for all $s_0 \in g^{-1}(0)$.

Furthermore, if s is drawn uniformly from $\{0,1\}^n$ then the expectation of $s \cdot z$ is $1/2$ under any distribution over $z \neq 0$. Thus, if s_1 is drawn uniformly from $g^{-1}(1)$ then

$$\Pr_{z \in \widehat{D}_1,\ s_1}[s_1 \cdot z \equiv 1 \pmod 2] > 1 - 2\mu.$$

It follows by the union bound that

$$\Pr_{z \in \widehat{D}_1}[b_1(z) \equiv 1 \pmod 2] > 1 - 3\mu.$$

and hence, for all $s_1 \in g^{-1}(1)$,

$$\Pr_{z \in \widehat{D}_1}[s_1 \cdot z \equiv 1 \pmod 2] > 1 - 4\mu.$$

We have established that g is a bounded-error threshold function of parity functions: there exist p_z summing to 1 such that for all s,

$$\Psi(s) = \sum_{z \in \{0,1\}^n \setminus \{0^n\}} p_z (s \cdot z) \text{ is } \begin{cases} > 1 - 4\mu & \text{if } g(s) = 1 \\ < 2\mu & \text{if } g(s) = 0. \end{cases} .$$

We will draw s uniformly at random from $\{0,1\}^n$ and consider $var(\Psi)$, the variance of $\Psi(s)$ under this distribution. First, if $p_z \ge 1/2$ for any z, then $g(s) = s \cdot z \oplus b_z$ and $\mu(g) = 0$. Second, suppose $p_z < 1/2$ 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

$$var(\Psi) = (1/4) \textstyle\sum_z p_z^2 < 1/8.$$

Also, we derived previously that if s is drawn uniformly from $g^{-1}(1)$, then the expectation of $\Psi(s)$ is at least $1-2\mu$. It follows that

$$var(\Psi) > (1/2 - 2\mu)^2 .$$

Combining,

$$\mu > \left(2-\sqrt{2}\right)/8 > 0.0732.$$

□

4 Generalization to Partial Functions

What if the function g is partial—that is, in addition to the RFS promise, there is the additional promise that each secret string s in the tree satisfies $s \in \Omega_g$ for some $\Omega_g \subseteq \{0,1\}^n$? Here we sketch a generalization of our results to this case. We first define a *two-sided* variant of $\mu(g)$: that is, one in which strings are drawn from the distributions D_0 and D_1 simultaneously.

Definition 2 *The two-sided nonparity coefficient $\mu_2(g)$ of g is the maximum μ_2^* for which the following holds. There exist distributions D_0, D_1 over $g^{-1}(0), g^{-1}(1)$ respectively such that for all $z \in \{0,1\}^n \setminus \{0^n\}$, $\widehat{s}_0 \in g^{-1}(0)$ and $\widehat{s}_1 \in g^{-1}(1)$,*

$$\Pr_{s_0 \in D_0, s_1 \in D_1}\left[s_0 \cdot z \equiv \widehat{s}_1 \cdot z \pmod 2 \;\;\vee\;\; s_1 \cdot z \equiv \widehat{s}_0 \cdot z \pmod 2\right] \geq \mu_2^*.$$

One can verify that $\mu_2 \in [0, 3/4]$ for all g, and (the analogue of Proposition 3) that $\mu_2(g_{\mathrm{mod}\,3}) = 3/4 - O(1/n)$. We also have $\mu_2(g) \geq \mu(g)$, since if two events both occur with probability at least μ then their disjunction does as well. On the other hand, $\mu_2(g)$ can be much greater than $\mu(g)$, and indeed if g is partial then $\mu_2(g) = 0$ implies that g is a parity function, whereas $\mu(g) = 0$ does not. To illustrate, for even n let $g_{LR}(s) = 0$ if $s[1], \ldots, s[n/2]$ are all 0, and $g_{LR}(s) = 1$ if $s[n/2+1], \ldots, s[n]$ are all 0, under the promise that exactly one of these is the case. Then:

Proposition 5 *$\mu_2(g_{LR}) = 1/2 + \Theta\left(2^{-n/2}\right)$, whereas $\mu(g_{LR}) = 0$.*

Proof: Let D_0 and D_1 be the uniform distributions over 0 and 1 inputs respectively. Choose any $z \neq 0^n$ for which (say) $z[n/2+1], \ldots, z[n]$ are all 0. Then $s_0 \cdot z \equiv 0 \pmod 2$ for every $s_0 \in g^{-1}(0)$, but

$$\Pr_{s_1 \in D_1}\left[s_1 \cdot z \equiv 1 \pmod 2\right] = \frac{1}{2 - 2^{-(n/2-1)}}$$

since $s[1], \ldots, s[n/2]$ are chosen uniformly at random conditioned on not being all 0. Hence $\mu_2(g_{LR}) = 1/2 + O\left(2^{-n/2}\right)$. This is the best bound achievable, since if z has 1 bits on both the left and right halves then $s \cdot z$ is close to uniformly random for both $s \in D_0$ and $s \in D_1$.

In the one-sided case, for any s we can choose a $z \neq 0^n$ such that $|z| \equiv 1 \pmod 2$ and $s[i] = 1$ whenever $z[i] = 1$. Then $s \cdot z$ disagrees with $\widehat{s} \cdot z$ for every $\widehat{s}$ such that $g_{LR}(\widehat{s}) \neq g_{LR}(s)$, and hence $\mu(g_{LR}) = 0$. □

In the partial setting, the theorem of Ambainis [18] that we need is his 'most general' lower bound, which he uses to show that the quantum query complexity of inverting a permutation is $\Omega(\sqrt{n})$. Using it we can obtain the following variant of Theorem 2; note that the exponent is weakened from $-h$ to $-h/2$.

Theorem 6 *For all g, $Q_2\left(RFS_h^g\right) = \Omega\left((1-\mu_2(g))^{-h/2}\right)$.*

On the other hand, we obtain a variant of Theorem 4 with the threshold twice as large:

Theorem 7 *Suppose $\mu_2(g) < 0.146$. Then g is a parity function (equivalently, $\mu_2(g) = 0$).*

The proofs of Theorems 6 and 7 are omitted for brevity.

5 Concluding Remarks

An intriguing open problem is whether Theorem 2 can be proved using the polynomial method of Beals et al. [6], rather than the adversary method of Ambainis [18]. It is known that one can lower-bound polynomial degree in terms of *block sensitivity*, which is (roughly) 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. See [20] for more about the quantum query complexity of such functions.

We believe the constants of Theorems 4 and 7 can be improved. The smallest nonzero $\mu(g)$ and $\mu_2(g)$ values we know of are both attained by $g = OR(s[1], s[2])$:

Proposition 8 $\mu(OR) = \mu_2(OR) = 1/3$.

Proof: First, $\mu(OR) \geq 1/3$, 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 $\widehat{s}_0 \in g^{-1}(0)$, $s \cdot z \not\equiv \widehat{s}_0 \cdot z$ with probability at most 2/3. Second, $\mu_2(OR) \leq 1/3$, since applying linear programming duality, we can let the first two bits of z and $\widehat{s}_1$, $(z[1]z[2], \widehat{s}_1[1]\widehat{s}_1[2])$, be $(01,01)$, $(10,10)$, or $(11,10)$ each with probability 1/3, while $z[3], \ldots, z[n]$ are all 0. Then $s_0 \cdot z \not\equiv \widehat{s}_1 \cdot z$ always, and for any $s_1 \in g^{-1}(1)$, $s_1 \cdot z \not\equiv \widehat{s}_0 \cdot z$ with probability 2/3. $\square$

A final note. Define *affine recursive Fourier sampling*, or $ARFS_h$, similarly to RFS_h, except that at each level, we are guaranteed there exists a secret string s and additive constant $c \in \{0,1\}$ such that $A_n(x) = (s \cdot x + c) \pmod 2$ for all $x \in \{0,1\}^n$. As before, the goal at each level is to compute $g(s)$; we do not care about c. Clearly the algorithm of Lemma 1 works also for $ARFS_h$, since c is simply encoded into the phase when we apply the Hadamard gates. On the other hand, we can show a quantum lower bound for $ARFS_h^g$ (for any g) much more easily than for RFS_h^g. For observe that we can encode the parity of 2^h bits into $ARFS_h^g$; thus, by a result of Beals et al. [6], a quantum algorithm for $ARFS_h^g$ requires $\Omega(2^h)$ queries.

Acknowledgements

I thank Lisa Hales, Umesh Vazirani, Ronald de Wolf, and the anonymous reviewers for helpful comments.

References

1. P. Shor (1997), *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing 26(5), pp. 1484–1509. quant-ph/9508027 (Available at `www.arxiv.org`).
2. D. Simon (1994), *On the power of quantum computation*, Proceedings of IEEE FOCS'94, pp. 116–123.
3. D. Deutsch and R. Jozsa (1992), *Rapid solution of problems by quantum computation*, Proceedings of the Royal Society of London A439, pp. 553–558.
4. E. Bernstein and U. Vazirani (1997), *Quantum complexity theory*, SIAM Journal on Computing 26(5), pp. 1411–1473.
5. S. Aaronson, *The Complexity Zoo.* http://www.cs.berkeley.edu/˜aaronson/zoo.html.
6. R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf (1998), *Quantum lower bounds by polynomials*, Proceedings of IEEE FOCS'98, pp. 352–361. quant-ph/9802049.
7. W. van Dam, S. Hallgren, and L. Ip (2003), *Algorithms for hidden coset problems*, Proceedings of SODA'2003. quant-ph/0211140.
8. D. Boneh and R. Lipton (1996), *Algorithms for black box fields and their application to cryptography*, Proceedings of CRYPTO'96, Lecture Notes in Computer Science Vol. 1109, Springer-Verlag, pp. 283–297.
9. J. Watrous (2000), *Succinct quantum proofs for properties of finite groups*, Proceedings of IEEE FOCS'2000, pp. 537–546. cs.CC/0009002.
10. F. Green and R. Pruim (2001), *Relativized separation of EQP from P^{NP}*, Information Processing Letters 80(5), pp. 257–260.

11. L. Babai (1992), *Bounded round interactive proofs in finite groups*, SIAM Journal on Discrete Math 5(1), pp. 88–111.
12. U. Vazirani (2001), personal communication.
13. A. A. Razborov (1987), *Lower bounds for the size of circuits of bounded depth with basis* $\{\&, \oplus\}$, Mathematicheskie Zametki 41(4), pp. 598–607. English translation in Math. Notes. Acad. Sci. USSR 41(4), pp. 333–338, 1987.
14. R. Smolensky (1987), *Algebraic methods in the theory of lower bounds for Boolean circuit complexity*, Proceedings of ACM STOC'87, pp. 77–82.
15. M. Furst, J. Saxe, and M. Sipser (1981), *Parity, circuits, and the polynomial time hierarchy*, Proceedings of IEEE FOCS'81, pp. 260–270.
16. N. Linial, Y. Mansour, and N. Nisan (1993), *Constant depth circuits, Fourier transform, and learnability*, Journal of the ACM 40(3), pp. 607–620.
17. P. Høyer and R. de Wolf (2002), manuscript.
18. A. Ambainis (2000), *Quantum lower bounds by quantum arguments*, Proceedings of ACM STOC'2000, pp. 636–643. Journal version in *Journal of Computer and System Sciences* 64, pp. 750-767, 2002. quant-ph/0002066.
19. R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca (1998), *Quantum algorithms revisited*, Proceedings of the Royal Society of London A454, pp. 339–354. quant-ph/9708016.
20. S. Aaronson (2002), *Quantum certificate complexity*, submitted. quant-ph/0210020.