Quantum Lower Bounds by Polynomials

Robert Beals
University of Arizona
and
Harry Buhrman
CWI and University of Amsterdam
and
Richard Cleve
University of Calgary
and
Michele Mosca
University of Waterloo
and
Ronald de Wolf
CWI and University of Amsterdam

A preliminary version of this paper was presented at FOCS'98 [Beals et al. 1998]. HB and RdW are partially supported by EU fifth framework program QAIP, IST-1999-11234. RC and MM gratefully acknowledge the hospitality of the CWI, where much of this research took place. RC is partially supported by Canada's NSERC. MM was affiliated with the Centre for Quantum Computation, Oxford (UK) when this work was done. He also thanks CESG for their support.
Name: Robert Beals
Affiliation: University of Arizona
Address: Department of Mathematics, University of Arizona, P.O. Box 210089, 617 N. Santa Rita Ave, Tucson AZ 85721–0089, USA. E-mail: beals@math.arizona.edu.
Name: Harry Buhrman
Affiliation: CWI
Address: CWI, P.O. Box 94079, Amsterdam, The Netherlands. E-mail: buhrman@cwi.nl.
Name: Richard Cleve
Affiliation: University of Calgary
Address: Department of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4. E-mail: cleve@cpsc.ucalgary.ca.
Name: Michele Mosca
Affiliation: University of Waterloo
Address: Centre for Applied Cryptographic Research, University of Waterloo, Canada. E-mail: mmosca@cacr.math.uwaterloo.ca.
Name: Ronald de Wolf
Affiliation: CWI and University of Amsterdam
Address: CWI, P.O. Box 94079, Amsterdam, The Netherlands. E-mail: rdewolf@cwi.nl.

We examine the number of queries to input variables that a quantum algorithm requires to compute Boolean functions on $\{0,1\}^N$ in the *black-box* model. We show that the exponential quantum speed-up obtained for *partial* functions (i.e., problems involving a promise on the input) by Deutsch and Jozsa, Simon, and Shor cannot be obtained for any *total* function: if a quantum algorithm computes some total Boolean function f with small error probability using T black-box queries, then there is a classical deterministic algorithm that computes f exactly with $O(T^6)$ queries. We also give asymptotically tight characterizations of T for all symmetric f in the exact, zero-error, and bounded-error settings. Finally, we give new precise bounds for AND, OR, and PARITY. Our results are a quantum extension of the so-called polynomial method, which has been successfully applied in classical complexity theory, and also a quantum extension of results by Nisan about a polynomial relationship between randomized and deterministic decision tree complexity.

Categories and Subject Descriptors: F.1.1 [**Computation by Abstract Devices**]: Models of Computation; F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms: Theory, Algorithms, Performance

Additional Key Words and Phrases: Quantum computing, query complexity, black-box model, lower bounds, polynomial method

1. INTRODUCTION

The *black-box* model of computation arises when one is given a black-box containing an N-tuple of Boolean variables $X = (x_0, x_1, \ldots, x_{N-1})$. The box is equipped to output the bit x_i on input i. We wish to determine some property of X, accessing the x_i only through the black-box. Such a black-box access is called a *query*. A property of X is any Boolean function that depends on X, i.e., a property is a function $f : \{0,1\}^N \to \{0,1\}$. We want to compute such properties using as few queries as possible. For classical algorithms, this optimal number of queries is known as the *decision tree complexity* of f.

Consider, for example, the case where the goal is to determine whether or not X contains at least one 1, so we want to compute the property $\mathrm{OR}_N(X) = x_0 \vee \ldots \vee x_{N-1}$. It is well known that the number of queries required to compute OR_N by any *classical* (deterministic or probabilistic) algorithm is $\Theta(N)$. Grover [Grover 1996] discovered a remarkable *quantum* algorithm that can be used to compute OR_N with small error probability using only $O(\sqrt{N})$ queries. His algorithm makes essential use of the fact that a quantum algorithm can apply a query to a *superposition* of different i, thereby accessing different input bits x_i at the same time, each with some amplitude. This upper bound of $O(\sqrt{N})$ queries was shown to be asymptotically optimal [Bennett et al. 1997; Boyer et al. 1998; Zalka 1999] (the first version of [Bennett et al. 1997] in fact appeared *before* Grover's algorithm).

Most other existing quantum algorithms can be naturally expressed in the black-box model. For example, in the case of *Simon's problem* [Simon 1997], one is given a function $\tilde{X} : \{0,1\}^n \to \{0,1\}^n$ satisfying the promise that there is an $s \in \{0,1\}^n$ such that $\tilde{X}(i) = \tilde{X}(j)$ iff $i = j \oplus s$, where $\oplus$ denotes bitwise exclusive-or (addition mod 2). The goal is to determine whether $s = 0$ or not. Simon's quantum algorithm yields an *exponential* speed-up over classical algorithms: it requires an expected

number of $O(n)$ applications of $\tilde{X}$, whereas every classical randomized algorithm for the same problem must make $\Omega(\sqrt{2^n})$ queries. Note that the function $\tilde{X}$ can be viewed as a black-box $X = (x_0, \ldots, x_{N-1})$ of $N = n2^n$ bits, and that an $\tilde{X}$-application can be simulated by n queries to X. Thus we see that Simon's problem fits squarely in the black-box setting, and exhibits an exponential quantum-classical separation for this promise-problem. The promise means that Simon's problem $f : \{0,1\}^N \to \{0,1\}$ is *partial*; it is not defined on all $X \in \{0,1\}^N$ but only on X that correspond to an $\tilde{X}$ satisfying the promise. (In the previous example of OR_N, the function is *total*; however, the quantum speed-up is only quadratic instead of exponential.) Something similar holds for the *order-finding problem*, which is the core of Shor's efficient quantum factoring algorithm [Shor 1997]. In this case the promise is the periodicity of a certain function derived from the number that we want to factor (see [Cleve 2000] for the exponential classical lower bound for order-finding). Most other quantum algorithms are naturally expressed in the black-box model as well, see e.g. [Deutsch and Jozsa 1992; Boneh and Lipton 1995; Kitaev 1995; Boyer et al. 1998; Brassard and Høyer 1997; Brassard et al. 1997; Høyer 1999; Mosca and Ekert 1998; Cleve et al. 1998; Brassard et al. 2000; Grover 1998; Buhrman et al. 1998; Dam 1998; Farhi et al. 1999b; Høyer et al. 2001; Buhrman et al. 2001; Dam and Hallgren 2000].

Of course, *upper bounds* in the black-box model immediately yield upper bounds for the *circuit description* model in which the function X is succinctly described as a $(\log N)^{O(1)}$-sized circuit computing x_i from i. On the other hand, *lower bounds* in the black-box model do not imply lower bounds in the circuit model, though they can provide useful guidance, indicating what certain algorithmic approaches are capable of accomplishing. It is noteworthy that, at present, there is no known algorithm for computing OR_N (i.e., satisfiability of a $\log N$-variable propositional formula) in the circuit model that is significantly more efficient than using the circuit solely to make queries. Some better algorithms are known for k-SAT [Schöning 1999] but not for satisfiability in general (though *proving* that no better algorithm exists is likely to be difficult, as it would imply $P \neq NP$).

It should also be noted that the black-box complexity of a function only considers the number of queries; it does not capture the complexity of the *auxiliary* computational steps that have to be performed in addition to the queries. In cases such as the computation of OR, PARITY, MAJORITY, this auxiliary work is not significantly larger than the number of queries; however, in some cases it may be much larger. For example, consider the case of factoring N-bit integers. The best known algorithms for this involve $\Theta(N)$ queries to determine the integer, followed by $2^{N^{\Omega(1)}}$ operations in the classical case but only $N^2(\log N)^{O(1)}$ operations in the quantum case [Shor 1997]. Thus, the number of queries seems not to be of primary importance in the case of factoring. However, the problem that Shor's quantum algorithm actually solves is the order-finding problem, which *can* be expressed in the black-box model as mentioned above.

In this paper, we analyze the black-box complexity of several functions and classes of functions in the quantum computation setting, establishing strong lower bounds. In particular, we show that the kind of exponential quantum speed-up that algorithms like Simon's achieve for partial functions cannot be obtained by any quantum algorithm for any total function: at most a polynomial speed-up is possible. We

also tightly characterize the quantum black-box complexity of all symmetric functions, and obtain exact bounds for functions such as AND, OR, PARITY, and MAJORITY for various error models: exact, zero-error, bounded-error.

An important ingredient of our approach is a reduction that translates quantum algorithms that make T queries into multilinear polynomials of degree at most $2T$ over the N variables. This is a quantum extension of the so-called *polynomial method*, which has been successfully applied in classical complexity theory (see e.g. [Nisan and Szegedy 1994; Beigel 1993]). Also, our polynomial relationship between the quantum and the classical complexity is analogous to earlier results by Nisan [Nisan 1991], who proved a polynomial relationship between randomized and deterministic decision tree complexity.

The only quantum black-box lower bounds known prior to this work were Jozsa's limitations on the power of 1-query algorithms [Jozsa 1991], the search-type bounds of [Bennett et al. 1997; Boyer et al. 1998; Zalka 1999], and some bounds derived from communication complexity [Buhrman et al. 1998]. The tight lower bound for PARITY of [Farhi et al. 1998] appeared independently and around the same time as a first version of this work [Beals et al. 1998], but their proof technique does not seem to generalize easily beyond PARITY. After the first appearance of this work, our polynomial approach has been used to derive many other quantum lower bounds, see e.g. [Nayak and Wu 1999; Buhrman et al. 1999; Farhi et al. 1999a; Ambainis 1999; Wolf 2000; Servedio and Gortler 2000]. Recently an alternative quantum lower bound method appeared [Ambainis 2000] which yields good bounds in cases where polynomial degrees are hard to determine (for instance for AND-OR trees), but it seems, on the other hand, that some bounds obtainable using the polynomial method cannot easily be obtained using this new method (see, e.g., [Buhrman et al. 1999]).

2. SUMMARY OF RESULTS

We consider three different settings for computing f on $\{0,1\}^N$ in the black-box model. In the *exact* setting, an algorithm is required to return $f(X)$ with certainty for every X. In the *zero-error* setting, for every X, an algorithm may return "inconclusive" with probability at most $1/2$, but *if* it returns an answer, this must be the correct value of $f(X)$ (algorithms in this setting are sometimes called *Las Vegas* algorithms). Finally, in the *two-sided bounded-error* setting, for every X, an algorithm must correctly return the answer with probability at least $2/3$ (algorithms in this setting are sometimes called *Monte Carlo* algorithms; the $2/3$ is arbitrary and may be replaced by any $1/2+\varepsilon$ for fixed constant $0<\varepsilon<1/2$).

Our main results are:[1]

(1) In the black-box model, the quantum speed-up for *any* total function cannot be more than by a sixth-root. More specifically, if a quantum algorithm computes

[1] All our results remain valid if we consider a *controlled* black-box, where the first bit of the state indicates whether the black-box is to be applied or not. (Thus such a black-box would map $|0,i,b,z\rangle$ to $|0,i,b,z\rangle$ and $|1,i,b,z\rangle$ to $|1,i,b\oplus x_i,z\rangle$.) Also, our results remain valid if we consider *mixed* rather than only pure states. In particular, allowing intermediate measurements in a quantum query algorithm does not give more power, since all measurements can be delayed until the end of the computation at the cost of some additional memory.

f with bounded-error probability by making T queries, then there is a classical deterministic algorithm that computes f exactly making at most $O(T^6)$ queries. If f is *monotone* then the classical algorithm needs at most $O(T^4)$ queries, and if f is *symmetric* then it needs at most $O(T^2)$ queries. If the quantum algorithm is *exact*, then the classical algorithm needs $O(T^4)$ queries.
As a by-product, we also improve the polynomial relation between the *decision tree complexity* $D(f)$ and the *approximate degree* $\widetilde{deg}(f)$ of [Nisan and Szegedy 1994] from $D(f) \in O(\widetilde{deg}(f)^8)$ to $D(f) \in O(\widetilde{deg}(f)^6)$.

(2) We tightly characterize the black-box complexity of all non-constant symmetric functions as follows. In the exact or zero-error settings $\Theta(N)$ queries are necessary and sufficient, and in the bounded-error setting $\Theta(\sqrt{N(N-\Gamma(f))})$ queries are necessary and sufficient, where $\Gamma(f) = \min\{|2k-N+1| : f \text{ flips value if the Hamming weight of the input changes from } k \text{ to } k+1\}$ (this $\Gamma(f)$ is a number that is low if f flips for inputs with Hamming weight close to $N/2$ [Paturi 1992]). This should be compared with the *classical* bounded-error query complexity of such functions, which is $\Theta(N)$. Thus, $\Gamma(f)$ characterizes the speed-up that quantum algorithms give for all total functions.
An interesting example is the THRESHOLD$_M$ function, which is 1 iff its input X contains at least M 1s. This has query complexity $\Theta(\sqrt{M(N-M+1)})$.

(3) For OR, AND, PARITY, MAJORITY, we obtain the bounds in the table below (all given numbers are both necessary and sufficient). These results are all new,

	exact	zero-error	bounded-error
OR_N, AND_N	N	N	$\Theta(\sqrt{N})$
PARITY_N	$N/2$	$N/2$	$N/2$
MAJ_N	$\Theta(N)$	$\Theta(N)$	$\Theta(N)$

Table 1. Some quantum complexities

with the exception of the $\Theta(\sqrt{N})$-bounds for OR and AND in the bounded-error setting, which appear in [Bennett et al. 1997; Boyer et al. 1998; Zalka 1999]. The new bounds improve by polylog(N) factors previous lower bound results from [Buhrman et al. 1998], which were obtained through a reduction from communication complexity. The new bounds for PARITY were independently obtained by Farhi *et al.* [Farhi et al. 1998].
Note that lower bounds for OR imply lower bounds for the *search* problem, where we want to find an i such that $x_i = 1$, if such an i exists. Thus exact or zero-error quantum search requires N queries, in contrast to $\Theta(\sqrt{N})$ queries for the bounded-error case. (On the other hand, if we are promised in advance that the number of solutions is t, then a solution can be found with probability 1 using $O(\sqrt{N/t})$ queries [Brassard et al. 2000].)

3. SOME DEFINITIONS

Our main goal in this paper is to find the number of queries a quantum algorithm needs to compute some Boolean function by relating such algorithms to polynomials. In this section we give some basic definitions and properties of multilinear polynomials and Boolean functions, and describe our quantum setting.

3.1 Boolean Functions and Polynomials

We assume the following setting, mainly adapted from [Nisan and Szegedy 1994]. We have a vector of N Boolean variables $X = (x_0, \ldots, x_{N-1})$, and we want to compute a Boolean function $f : \{0,1\}^N \to \{0,1\}$ of X. Unless explicitly stated otherwise, f will always be total. The Hamming weight (number of 1s) of X is denoted by $|X|$. For example, $\mathrm{OR}_N(X) = 1$ iff $|X| > 0$, $\mathrm{AND}_N(X) = 1$ iff $|X| = N$, $\mathrm{PARITY}_N(X) = 1$ iff $|X|$ is odd, and $\mathrm{MAJ}_N(X) = 1$ iff $|X| > N/2$.

We can represent Boolean functions using N-variate polynomials $p : \mathbb{R}^N \to \mathbb{R}$. Since $x^m = x$ whenever $x \in \{0,1\}$, we can restrict attention to *multilinear* p. If $p(X) = f(X)$ for all $X \in \{0,1\}^N$, then we say that p *represents* f. It is easy to see that every f is represented by a *unique* multilinear polynomial p of degree $\leq N$. We use $deg(f)$ to denote the degree of this p. If $|p(X) - f(X)| \leq 1/3$ for all $X \in \{0,1\}^N$, we say p *approximates* f, and $\widetilde{deg}(f)$ denotes the degree of a minimum-degree p that approximates f. For example, $x_0 x_1 \ldots x_{N-1}$ is a multilinear polynomial of degree N that represents AND_N. Similarly, $1 - (1 - x_0)(1 - x_1)\ldots(1 - x_{N-1})$ represents OR_N. The polynomial $\frac{1}{3}x_0 + \frac{1}{3}x_1$ approximates but does not represent AND_2.

Nisan and Szegedy [Nisan and Szegedy 1994, Theorem 2.1] proved a general lower bound on the degree of any Boolean function that depends on N variables:

THEOREM 3.1 (NISAN & SZEGEDY). *If f is a Boolean function that depends on N variables, then $deg(f) \geq \log N - O(\log\log N)$.*

Let $p : \mathbb{R}^N \to \mathbb{R}$ be a polynomial. If π is some permutation on $\{0, \ldots, N-1\}$, and $X = (x_0, \ldots, x_{N-1})$, then $\pi(X) = (x_{\pi(0)}, \ldots, x_{\pi(N-1)})$. Let S_N be the set of all $N!$ permutations. The *symmetrization* p^{sym} of p averages over all permutations of the input, and is defined as:

$$p^{sym}(X) = \frac{\sum_{\pi \in S_N} p(\pi(X))}{N!}.$$

Note that p^{sym} is a polynomial of degree at most the degree of p. Symmetrizing may actually lower the degree: if $p = x_0 - x_1$, then $p^{sym} = 0$. The following lemma, originally due to [Minsky and Papert 1968], allows us to reduce an N-variate polynomial to a single-variate one.

LEMMA 3.2 (MINSKY & PAPERT). *If $p : \mathbb{R}^n \to \mathbb{R}$ is a multilinear polynomial, then there exists a polynomial $q : \mathbb{R} \to \mathbb{R}$, of degree at most the degree of p, such that $p^{sym}(X) = q(|X|)$ for all $X \in \{0,1\}^N$.*

PROOF. Let d be the degree of p^{sym}, which is at most the degree of p. Let V_j denote the sum of all $\binom{N}{j}$ products of j different variables, so $V_1 = x_0 + \ldots + x_{N-1}$, $V_2 = x_0x_1 + x_0x_2 + \ldots + x_{N-1}x_{N-2}$, etc. Since p^{sym} is symmetrical, it can be written as

$$p^{sym}(X) = a_0 + a_1V_1 + a_2V_2 + \ldots + a_dV_d,$$

for some $a_i \in \mathbb{R}$. Note that V_j assumes value $\binom{|X|}{j} = |X|(|X|-1)(|X|-2)\ldots(|X|-j+1)/j!$ on X, which is a polynomial of degree j of $|X|$. Therefore the single-variate

polynomial q defined by

$$q(|X|) = a_0 + a_1\binom{|X|}{1} + a_2\binom{|X|}{2} + \ldots + a_d\binom{|X|}{d}$$

satisfies the lemma. □

A Boolean function f is *symmetric* if permuting the input does not change the function value (i.e., $f(X)$ only depends on $|X|$). Paturi has proved a powerful theorem that characterizes $\widetilde{deg}(f)$ for symmetric f. For such f, let $f_k = f(X)$ for $|X| = k$, and define

$$\Gamma(f) = \min\{|2k - N + 1| : f_k \neq f_{k+1} \text{ and } 0 \leq k \leq N-1\}.$$

$\Gamma(f)$ is low if f_k "jumps" near the middle (i.e., for some $k \approx N/2$). Now [Paturi 1992, Theorem 1] gives:

THEOREM 3.3 (PATURI). *If f is a non-constant symmetric Boolean function on $\{0,1\}^N$, then $\widetilde{deg}(f) \in \Theta(\sqrt{N(N-\Gamma(f))})$.*

For functions like OR_N and AND_N, we have $\Gamma(f) = N-1$ and hence $\widetilde{deg}(f) \in \Theta(\sqrt{N})$. For PARITY_N (which is 1 iff $|X|$ is odd) and MAJ_N (which is 1 iff $|X| > N/2$), we have $\Gamma(f) = 1$ if N is even and $\Gamma(f) = 0$ if N is odd, hence $\widetilde{deg}(f) \in \Theta(N)$ for those functions.

3.2 The Framework of Quantum Networks

Our goal is to compute some Boolean function f of $X = (x_0, \ldots, x_{N-1})$, where X is given as a black-box: calling the black-box on i returns the value of x_i. We want to use as few queries as possible.

A classical algorithm that computes f by using (adaptive) black-box queries to X is called a *decision tree*, since it can be pictured as a binary tree where each node is a query, each node has the two outcomes of the query as children, and the leaves give answer $f(X) = 0$ or $f(X) = 1$. The *cost* of such an algorithm is the number of queries made on the worst-case input X, i.e., the depth of the tree. The *decision tree complexity* $D(f)$ of f is the cost of the best decision tree that computes f. Similarly we can define $R(f)$ as the worst-case number of queries for *randomized* algorithms that compute $f(X)$ with error probability $\leq 1/3$ for all X. By a well-known result of Nisan, the best randomized algorithm can be at most polynomially more efficient than the best deterministic algorithm: $D(f) \in O(R(f)^3)$ for all total f [Nisan 1991, Theorem 4].

For a general introduction to quantum computing we refer to [Nielsen and Chuang 2000]. A *quantum network* (also called *quantum algorithm*) with T queries is the quantum analogue to a classical decision tree with T queries, where queries and other operations can now be made in quantum superposition. Such a network can be represented as a sequence of unitary transformations:

$$U_0, O_1, U_1, O_2, \ldots, U_{T-1}, O_T, U_T,$$

where the U_i are arbitrary unitary transformations, and the O_j are unitary transformations that correspond to queries to X. The computation ends with some measurement or observation of the final state. We assume each transformation

acts on m qubits and each qubit has basis states $|0\rangle$ and $|1\rangle$, so there are 2^m basis states for each stage of the computation. It will be convenient to represent each basis state as a binary string of length m or as the corresponding natural number, so we have basis states $|0\rangle, |1\rangle, |2\rangle, \ldots, |2^m - 1\rangle$. Let K be the index set $\{0, 1, 2, \ldots, 2^m - 1\}$. With some abuse of notation, we will sometimes identify a set of numbers with the corresponding set of basis states. Every state $|\phi\rangle$ of the network can be uniquely written as $|\phi\rangle = \sum_{k \in K} \alpha_k |k\rangle$, where the α_k are complex numbers such that $\sum_{k \in K} |\alpha_k|^2 = 1$. When $|\phi\rangle$ is measured in the above basis, the probability of observing $|k\rangle$ is $|\alpha_k|^2$. Since we want to compute a function of X, which is given as a black-box, the initial state of the network is not very important and we will disregard it hereafter; we may assume the initial state to be $|0\rangle$ always.

The queries are implemented using the unitary transformations O_j in the following standard way. The transformation O_j only affects the leftmost part of a basis state: it maps basis state $|i, b, z\rangle$ to $|i, b \oplus x_i, z\rangle$ ($\oplus$ denotes XOR). Here i has length $\lceil \log N \rceil$ bits, b is one bit, and z is an arbitrary string of $m - \lceil \log N \rceil - 1$ bits. Note that the O_j are all equal.

How does a quantum network compute a Boolean function f of X? Let us designate the rightmost qubit of the final state of the network as the output bit. More precisely, the output of the computation is defined to be the value we observe if we measure the rightmost qubit of the final state. If this output equals $f(X)$ with certainty, for every X, then the network computes f *exactly*. If the output equals $f(X)$ with probability at least $2/3$, for every X, then the network computes f with bounded error probability at most $1/3$. To define the zero-error setting, the output is obtained by observing the *two* rightmost qubits of the final state. If the first of these qubits is 0, the network claims ignorance ("inconclusive"), otherwise the second qubit should contain $f(X)$ with certainty. For every X, the probability of getting "inconclusive" should be less than $1/2$. We use $Q_E(f)$, $Q_0(f)$ and $Q_2(f)$ to denote the minimum number of queries required by a quantum network to compute f in the exact, zero-error and bounded-error settings, respectively. It can be shown that the quantum setting generalizes the classical setting, hence $Q_2(f) \leq Q_0(f) \leq Q_E(f) \leq D(f) \leq N$ and $Q_2(f) \leq R(f) \leq D(f) \leq N$.

4. GENERAL LOWER BOUNDS ON THE NUMBER OF QUERIES

In this section we will provide some general lower bounds on the number of queries required to compute a Boolean function f on a quantum network, either exactly or with zero- or bounded-error probability.

4.1 The Acceptance Probability is a Polynomial

Here we prove that the acceptance probability of a T-query quantum network can be written as a multilinear N-variate polynomial $P(X)$ of degree at most $2T$. The next lemmas relate quantum networks to polynomials; they are the key to most of our results.

LEMMA 4.1. *Let $\mathcal{N}$ be a quantum network that makes T queries to a black-box X. Then there exist complex-valued N-variate multilinear polynomials $p_0, \ldots, p_{2^m-1}$,*

each of degree at most T, such that the final state of the network is the superposition

$$\sum_{k \in K} p_k(X)|k\rangle,$$

for any black-box X.

PROOF. Let $|\phi_i\rangle$ be the state of the network (using some black-box X) just before the ith query. Note that $|\phi_{i+1}\rangle = U_i O_i |\phi_i\rangle$. The amplitudes in $|\phi_0\rangle$ depend on the initial state and on U_0 but not on X, so they are polynomials of X of degree 0. A query maps basis state $|i, b, z\rangle$ to $|i, b \oplus x_i, z\rangle$. Hence if the amplitude of $|i, 0, z\rangle$ in $|\phi_0\rangle$ is α and the amplitude of $|i, 1, z\rangle$ is β, then the amplitude of $|i, 0, z\rangle$ *after* the query becomes $(1 - x_i)\alpha + x_i\beta$ and the amplitude of $|i, 1, z\rangle$ becomes $x_i\alpha + (1 - x_i)\beta$, which are polynomials of degree 1. (In general, if the amplitudes before a query are polynomials of degree $\leq j$, then the amplitudes after the query will be polynomials of degree $\leq j + 1$.) Between the first and the second query lies the unitary transformation U_1. However, the amplitudes after applying U_1 are just linear combinations of the amplitudes before applying U_1, so the amplitudes in $|\phi_1\rangle$ are polynomials of degree at most 1. Continuing in this manner, the amplitudes of the final states are found to be polynomials of degree at most T. We can make these polynomials multilinear without affecting their values on $X \in \{0,1\}^N$, by replacing all x_i^m by x_i. □

Note that we have not used the assumption that the U_j are unitary, but only their linearity. The next lemma is also implicit in the combination of some proofs in [Fenner et al. 1993; Fortnow and Rogers 1999].

LEMMA 4.2. *Let $\mathcal{N}$ be a quantum network that makes T queries to a black-box X, and B be a set of basis states. Then there exists a real-valued multilinear polynomial $P(X)$ of degree at most $2T$, which equals the probability that observing the final state of the network with black-box X yields a state from B.*

PROOF. By the previous lemma, we can write the final state of the network as

$$\sum_{k \in K} p_k(X)|k\rangle,$$

for any X, where the p_k are complex-valued polynomials of degree $\leq T$. The probability of observing a state in B is

$$P(X) = \sum_{k \in B} |p_k(X)|^2.$$

If we split p_k into its real and imaginary parts as $p_k(X) = pr_k(X) + i \cdot pi_k(X)$, where pr_k and pi_k are real-valued polynomials of degree $\leq T$, then $|p_k(X)|^2 = (pr_k(X))^2 + (pi_k(X))^2$, which is a real-valued polynomial of degree at most $2T$. Hence P is also a real-valued polynomial of degree at most $2T$, which we can make multilinear without affecting its values on $X \in \{0,1\}^N$. □

Letting B be the set of states that have 1 as rightmost bit, it follows that we can write the acceptance probability of a T-query network (i.e., the probability of getting output 1) as a polynomial $P(X)$ of degree $\leq 2T$.

4.2 Lower Bounds for Exact and Zero-Error Quantum Computation

Consider a quantum network that computes f exactly using $T = Q_E(f)$ queries. Its acceptance probability $P(X)$ is a polynomial of degree $\leq 2T$ which equals $f(X)$ for all X. But then $P(X)$ must have degree $deg(f)$, which implies the following lower bound result for $Q_E(f)$:

THEOREM 4.3. *If f is a Boolean function, then $Q_E(f) \geq deg(f)/2$.*

Combining this with Theorem 3.1, we obtain a weak but general lower bound:

COROLLARY 4.4. *If f depends on N variables, then $Q_E(f) \geq \frac{\log N}{2} - O(\log \log N)$.*

For *symmetric* f we can prove a much stronger bound. Firstly for the zero-error setting:

THEOREM 4.5. *If f is non-constant and symmetric, then $Q_0(f) \geq (N+1)/4$.*

PROOF. We assume $f(X) = 0$ for at least $(N+1)/2$ different Hamming weights of X; the proof is similar if $f(X) = 1$ for at least $(N+1)/2$ different Hamming weights. Consider a network that uses $T = Q_0(f)$ queries to compute f with zero-error. Let B be the set of basis states that have 11 as rightmost bits. These are the basis states corresponding to output 1. By Lemma 4.2, there is a real-valued multilinear polynomial P of degree $\leq 2T$, such that for all X, $P(X)$ equals the probability that the output of the network is 11 (i.e., that the network answers 1). Since the network computes f with zero-error and f is non-constant, $P(X)$ is non-constant and equals 0 on at least $(N+1)/2$ different Hamming weights (namely the Hamming weights for which $f(X) = 0$). Let q be the single-variate polynomial of degree $\leq 2T$ obtained from symmetrizing P (Lemma 3.2). This q is non-constant and has at least $(N+1)/2$ zeroes, hence degree at least $(N+1)/2$, and the result follows. □

Thus functions like OR_N, AND_N, PARITY_N, threshold functions etc., all require at least $(N+1)/4$ queries to be computed exactly or with zero-error on a quantum network. Since N queries always suffice, even classically, we have $Q_E(f) \in \Theta(N)$ and $Q_0(f) \in \Theta(N)$ for all non-constant symmetric f.

Secondly, for the exact setting we can prove slightly stronger lower bounds using results by Von zur Gathen and Roche [Gathen and Roche 1997, Theorems 2.6 and 2.8]:

THEOREM 4.6 (VON ZUR GATHEN & ROCHE). *If f is non-constant and symmetric, then $deg(f) = N - O(N^{0.548})$. If, in addition, $N+1$ is prime, then $deg(f) = N$.*

COROLLARY 4.7. *If f is non-constant and symmetric, then $Q_E(f) \geq N/2 - O(N^{0.548})$. If, in addition, $N+1$ is prime, then $Q_E(f) \geq N/2$.*

In Section 6 we give more precise bounds for some particular functions. In particular, this will show that the $N/2$ lower bound is tight, as it can be met for PARITY_N.

4.3 Lower Bounds for Bounded-Error Quantum Computation

Here we use similar techniques to get bounds on the number of queries required for *bounded-error* computation of some function. Consider the acceptance probability

of a T-query network that computes f with bounded-error, written as a polynomial $P(X)$ of degree $\leq 2T$. If $f(X) = 0$ then we have $0 \leq P(X) \leq 1/3$, and if $f(X) = 1$ then $2/3 \leq P(X) \leq 1$. Hence P approximates f, and we obtain:

THEOREM 4.8. *If f is a Boolean function, then $Q_2(f) \geq \widetilde{deg}(f)/2$.*

This result implies that a quantum algorithm that computes f with bounded error probability can be at most polynomially more efficient (in terms of number of queries) than a classical deterministic algorithm: Nisan and Szegedy proved that $D(f) \in O(\widetilde{deg}(f)^8)$ [Nisan and Szegedy 1994, Theorem 3.9], which together with the previous theorem implies $D(f) \in O(Q_2(f)^8)$. The fact that there is a polynomial relation between the classical and the quantum complexity is also implicit in the generic oracle-constructions of Fortnow and Rogers [Fortnow and Rogers 1999]. In Section 5 we will prove the stronger result $D(f) \in O(Q_2(f)^6)$.

Combining Theorem 4.8 with Paturi's Theorem 3.3 gives a lower bound for *symmetric* functions in the bounded-error setting: if f is non-constant and symmetric, then $Q_2(f) \in \Omega(\sqrt{N(N-\Gamma(f))})$. We can in fact prove a matching upper bound, using the following result about quantum counting [Brassard et al. 2000, Theorem 13]:

THEOREM 4.9 (BRASSARD, HØYER, MOSCA, TAPP). *There exists a quantum algorithm with the following property. For every N-bit input X (with $t = |X|$) and number T, the algorithm uses T queries and outputs a number $\tilde{t}$ such that*

$$|t - \tilde{t}| \leq 2\pi \frac{\sqrt{t(N-t)}}{T} + \pi^2 \frac{N}{T^2}$$

with probability at least $8/\pi^2$.

THEOREM 4.10. *If f is non-constant and symmetric, then we have that $Q_2(f) \in \Theta(\sqrt{N(N-\Gamma(f))})$.*

PROOF. We describe a strategy that computes f with small error probability. Let $f_k = f(x)$ for x with $|X| = k$. First note that since $\Gamma(f) = \min\{|2k - N + 1| : f_k \neq f_{k+1} \text{ and } 0 \leq k \leq N-1\}$, f_k must be identically 0 or 1 for $k \in \{\lceil (N-\Gamma(f))/2 \rceil, \ldots, \lceil (N+\Gamma(f)-2)/2 \rceil\}$. Consider some X with $|X| = t$. In order to be able to compute $f(X)$, it is sufficient to know t exactly if $t < \lceil (N-\Gamma(f))/2 \rceil$ or $t > \lceil (N+\Gamma(f)-2)/2 \rceil$, or to *know* that $\lceil (N-\Gamma(f))/2 \rceil \leq t \leq \lceil (N+\Gamma(f)-2)/2 \rceil$ otherwise.

Run the quantum counting algorithm for $\Theta(\sqrt{(N-\Gamma(f))N})$ steps to count the number of 1s in X. If t is in one of the two tails ($t < \lceil (N-\Gamma(f))/2 \rceil$ or $t > \lceil (N+\Gamma(f)-2)/2 \rceil$), then with high probability the algorithm gives us an exact count of t. If $\lceil (N-\Gamma(f))/2 \rceil \leq t \leq \lceil (N+\Gamma(f)-2)/2 \rceil$, then with high probability the counting algorithm returns some $\tilde{t}$ that is in this interval as well. Thus with high probability $f_{\tilde{t}}$ equals $f_t = f(X)$. This shows that we can compute f using only $O(\sqrt{N(N-\Gamma(f))})$ queries. □

Theorem 4.10 implies that the above-stated result about quantum counting (Theorem 4.9) is optimal, since a better upper bound for counting would give a better upper bound on $Q_2(f)$ for symmetric f, whereas we already know that Theorem 4.10 is tight. In contrast to Theorem 4.10, it can be shown that a randomized

classical strategy needs $\Theta(N)$ queries to compute any non-constant symmetric f with bounded-error.

Moreover, it can be shown that almost all functions f satisfy $deg(f) = N$, see [Buhrman and Wolf 2001], hence almost all f have $Q_E(f) \geq N/2$. After reading the preliminary version of this paper [Beals et al. 1998], Andris Ambainis [Ambainis 1999] proved a similar result for the approximate case: almost all f satisfy $\widetilde{deg}(f) \geq N/2 - O(\sqrt{N}\log N)$ and hence have $Q_2(f) \geq N/4 - O(\sqrt{N}\log N)$. On the other hand, Wim van Dam [Dam 1998] proved that with good probability we can learn all N variables in the black-box using only $N/2 + \sqrt{N}$ queries. This implies the general upper bound $Q_2(f) \leq N/2 + \sqrt{N}$ for *every* f. This bound is almost tight, as we will show later on that $Q_2(f) = \lceil N/2 \rceil$ for $f = \text{PARITY}$.

4.4 Lower Bounds in Terms of Block Sensitivity

Above we gave lower bounds on the number of queries used, in terms of degrees of polynomials that represent or approximate the function f that is to be computed. Here we give lower bounds in terms of the *block sensitivity* of f, a measure introduced in [Nisan 1991].

DEFINITION 4.11. *Let $f : \{0,1\}^N \to \{0,1\}$ be a function, $X \in \{0,1\}^N$, and $B \subseteq \{0,\ldots,N-1\}$ a set of indices. Let X^B denote the string obtained from X by flipping the variables in B. We say that f is sensitive to B on X if $f(X) \neq f(X^B)$. The block sensitivity $bs_X(f)$ of f on X is the maximum number t for which there exist t disjoint sets of indices $B_1,\ldots,B_t$ such that f is sensitive to each B_i on X. The block sensitivity $bs(f)$ of f is the maximum of $bs_X(f)$ over all $X \in \{0,1\}^N$.*

For example, $bs(\text{OR}_N) = N$, because if we take $X = (0,\ldots,0)$ and $B_i = \{i\}$, then flipping B_i in X flips the value of OR_N from 0 to 1.

We can adapt the proof of [Nisan and Szegedy 1994, Lemma 3.8] on lower bounds of polynomials to get lower bounds on the number of queries in a quantum network in terms of block sensitivity.[2] The proof uses a theorem from [Ehlich and Zeller 1964; Rivlin and Cheney 1966]:

THEOREM 4.12 (EHLICH & ZELLER; RIVLIN & CHENEY). *Let $p : \mathbb{R} \to \mathbb{R}$ be a polynomial such that $b_1 \leq p(i) \leq b_2$ for every integer $0 \leq i \leq N$, and the derivative p' satisfies $|p'(x)| \geq c$ for some real $0 \leq x \leq N$. Then $deg(p) \geq \sqrt{cN/(c + b_2 - b_1)}$.*

THEOREM 4.13. *If f is a Boolean function, then*

$$Q_E(f) \geq \sqrt{\frac{bs(f)}{8}} \text{ and } Q_2(f) \geq \sqrt{\frac{bs(f)}{16}}.$$

PROOF. We prove the lower bound on $Q_2(f)$ here, the bound on $Q_E(f)$ is completely analogous. Consider a network using $T = Q_2(f)$ queries that computes f with error probability $\leq 1/3$. Let p be the polynomial of degree $\leq 2T$ that approximates f, obtained as for Theorem 4.8. Note that $p(X) \in [0,1]$ for all $X \in \{0,1\}^N$, because p represents a probability.

[2] This theorem can also be proved by an argument similar to the lower bound proof for quantum searching in [Bennett et al. 1997], see e.g. [Vazirani 1998].

Let $b = bs(f)$, and Z and $B_1, \ldots, B_b$ be the input and sets that achieve the block sensitivity. We assume without loss of generality that $f(Z) = 0$. We transform $p(x_0, \ldots, x_{N-1})$ into a polynomial $q(y_1, \ldots, y_b)$ by replacing every x_j in p as follows:

(1) $x_j = (1 - z_j)y_i + z_j(1 - y_i)$ if $j \in B_i$
(2) $x_j = z_j$ if j occurs in none of the B_i

Now it is easy to see that q has the following properties:

(1) q is a multilinear polynomial of degree $\leq d \leq 2T$
(2) $q(Y) \in [0,1]$ for all $Y \in \{0,1\}^b$
(3) $q(\vec{0}) = p(Z) \in [0, 1/3]$
(4) $q(e_i) = p(Z^{B_i}) \in [2/3, 1]$ for all unit vectors $e_i \in \{0,1\}^b$

Let r be the single-variate polynomial of degree $\leq d$ obtained from symmetrizing q over $\{0,1\}^b$ (Lemma 3.2). Note that $0 \leq r(i) \leq 1$ for every integer $0 \leq i \leq b$, and for some $x \in [0,1]$ we have $r'(x) \geq 1/3$ (because $r(0) \leq 1/3$ and $r(1) \geq 2/3$). Applying Theorem 4.12 we obtain $d \geq \sqrt{(1/3)b/(1/3 + 1 - 0)} = \sqrt{b/4}$, hence $T \geq \sqrt{b/16}$. □

We can generalize this result to the computation of *partial* Boolean functions, which are only defined on a domain $\mathcal{D} \subset \{0,1\}^N$ of inputs that satisfy some promise, by generalizing the definition of block sensitivity to partial functions in the obvious way.

5. POLYNOMIAL RELATION FOR CLASSICAL AND QUANTUM COMPLEXITY

Here we will compare the classical complexities $D(f)$ and $R(f)$ with the quantum complexities. First some separations: in the next section we show $Q_2(\text{PARITY}_N) = \lceil N/2 \rceil$ while $D(\text{PARITY}_N) = N$. In the bounded-error setting $Q_2(\text{OR}_N) \in \Theta(\sqrt{N})$ by Grover's algorithm, while $R(\text{OR}_N) \in \Theta(N)$ and $D(\text{OR}_N) = N$, so we have a quadratic gap between $Q_2(f)$ on the one hand and $R(f)$ and $D(f)$ on the other.[3]

Nisan proved that the randomized complexity is at most polynomially better than the deterministic complexity: $D(f) \in O(R(f)^3)$ [Nisan 1991]. As mentioned in Section 4, we can prove that also the *quantum* complexity can be at most polynomially better than the best deterministic algorithm: $D(f) \in O(Q_2(f)^8)$. Here we give the stronger result that $D(f) \in O(Q_2(f)^6)$. In other words, if we can compute some function quantumly with bounded-error using T queries, we can compute it classically error-free using $O(T^6)$ queries. We will need the notion of *certificate complexity*:

DEFINITION 5.1. *Let C be an assignment $C : S \to \{0,1\}$ of values to some subset S of the N variables. We say that C is consistent with $X \in \{0,1\}^N$ if $x_i = C(i)$ for all $i \in S$.*

[3] In the case of randomized decision trees, no function is known for which there is a quadratic gap between $D(f)$ and $R(f)$. The best known separation is for complete binary AND/OR-trees, where $D(f) = N$ and $R(f) \in \Theta(N^{0.753\ldots})$, and it has been conjectured that this is the largest gap possible. This holds both for zero-error randomized trees [Saks and Wigderson 1986] and for bounded-error trees [Santha 1991].

For $b \in \{0,1\}$, a b-certificate for f is an assignment C such that $f(X) = b$ whenever X is consistent with C. The size of C is $|S|$.

The certificate complexity $C_X(f)$ of f on X is the size of a smallest $f(X)$-certificate that is consistent with X. The certificate complexity of f is $C(f) = \max_X C_X(f)$. The 1-certificate complexity of f is $C^{(1)}(f) = \max_{\{X|f(X)=1\}} C_X(f)$, and similarly we define $C^{(0)}(f)$.

For example, if f is the OR-function, then the certificate complexity on the input $(1,0,0,\ldots,0)$ is 1, because the assignment $x_0 = 1$ already forces the OR to 1. The same holds for the other X for which $f(X) = 1$, so $C^{(1)}(f) = 1$. On the other hand, the certificate complexity on $(0,0,\ldots,0)$ is N, so $C(f) = N$.

The first inequality in the next lemma is obvious from the definitions, the second inequality is [Nisan 1991, Lemma 2.4]. We include the proof for completeness.

LEMMA 5.2 (NISAN). $C^{(1)}(f) \leq C(f) \leq bs(f)^2$.

PROOF. Consider an input $X \in \{0,1\}^N$ and let $B_1, \ldots, B_b$ be disjoint *minimal* sets of variables that achieve the block sensitivity $b = bs_X(f) \leq bs(f)$. We will show that $C : \cup_i B_i \to \{0,1\}$ that sets variables according to X, is a certificate for X of size $\leq bs(f)^2$.

Firstly, if C were not an $f(X)$-certificate then let X' be an input that agrees with C, such that $f(X') \neq f(X)$. Let $X' = X^{B_{b+1}}$. Now f is sensitive to B_{b+1} on X and B_{b+1} is disjoint from $B_1, \ldots, B_b$, which contradicts $b = bs_X(f)$. Hence C is an $f(X)$-certificate.

Secondly, note that for $1 \leq i \leq b$ we must have $|B_i| \leq bs_{X^{B_i}}(f)$: if we flip one of the B_i-variables in X^{B_i} then the function value must flip from $f(X^{B_i})$ to $f(X)$ (otherwise B_i would not be minimal), so every B_i-variable forms a sensitive set for f on input X^{B_i}. Hence the size of C is $|\cup_i B_i| = \sum_{i=1}^b |B_i| \leq \sum_{i=1}^b bs_{X^{B_i}}(f) \leq bs(f)^2$. □

The crucial lemma is the following, which we prove along the lines of [Nisan 1991, Lemma 4.1].

LEMMA 5.3. $D(f) \leq C^{(1)}(f)bs(f)$.

PROOF. The following describes an algorithm to compute $f(X)$, querying at most $C^{(1)}(f)bs(f)$ variables of X (in the algorithm, by a "consistent" certificate C or input Y at some point we mean a C or Y that agrees with the values of all variables queried up to that point).

(1) Repeat the following at most $bs(f)$ times:
    Pick a consistent 1-certificate C and query those of its variables whose X-values are still unknown (if there is no such C, then return 0 and stop); if the queried values agree with C then return 1 and stop.
(2) Pick a consistent $Y \in \{0,1\}^N$ and return $f(Y)$.

The nondeterministic "pick a C" and "pick a Y" can easily be made deterministic by choosing the first C resp. Y in some fixed order. Call this algorithm **A**. Since **A** runs for at most $bs(f)$ stages and each stage queries at most $C^{(1)}(f)$ variables, **A** queries at most $C^{(1)}(f)bs(f)$ variables.

It remains to show that **A** always returns the right answer. If it returns an answer in step 1, this is either because there are no consistent 1-certificates left (and hence $f(X)$ must be 0) or because X is found to agree with a particular 1-certificate C; in both cases **A** gives the right answer.

Now consider the case where **A** returns an answer in step 2. We will show that all consistent Y must have the same f-value. Suppose not. Then there are consistent Y, Y' with $f(Y) = 0$ and $f(Y') = 1$. **A** has queried $b = bs(f)$ 1-certificates $C_1, C_2, \ldots, C_b$. Furthermore, Y' contains a consistent 1-certificate C_{b+1}. We will derive from these C_i disjoint sets B_i such that f is sensitive to each B_i on Y. For every $1 \leq i \leq b+1$, define B_i as the set of variables on which Y and C_i disagree. Clearly, each B_i is non-empty. Note that Y^{B_i} agrees with C_i, so $f(Y^{B_i}) = 1$ which shows that f is sensitive to each B_i on Y. Let v be a variable in some B_i $(1 \leq i \leq b)$, then $X(v) = Y(v) \neq C_i(v)$. Now for $j > i$, C_j has been chosen consistent with all variables queried up to that point (including v), so we cannot have $X(v) = Y(v) \neq C_j(v)$, hence $v \notin B_j$. This shows that all B_i and B_j are disjoint. But then f is sensitive to $bs(f)+1$ disjoint sets on Y, which is a contradiction. Accordingly, all consistent Y in step 2 must have the same f-value, and **A** returns the right value $f(Y) = f(X)$ in step 2, because X is one of those consistent Y. □

The inequality of the previous lemma is tight, because if $f = \mathrm{OR}$, then $D(f) = N$, $C^{(1)}(f) = 1$, $bs(f) = N$.

The previous two lemmas imply $D(f) \leq bs(f)^3$. Combining this with Theorem 4.13 ($bs(f) \leq 16\, Q_2(f)^2$), we obtain the main result:

THEOREM 5.4. *If f is a Boolean function, then $D(f) \leq 4096\, Q_2(f)^6$.*

We do not know if the $D(f) \in O(Q_2(f)^6)$-relation is tight, and suspect that it is not. The best separation we know is for OR and similar functions, where $D(f) = N$ and $Q_2(f) \in \Theta(\sqrt{N})$. However, for such symmetric Boolean function we can do no better than a quadratic separation: $D(f) \leq N$ always holds, and we have $Q_2(f) \in \Omega(\sqrt{N})$ by Theorem 4.10, hence $D(f) \in O(Q_2(f)^2)$ for symmetric f. For *monotone* Boolean functions, where the function value either increases or decreases monotonically if we set more input bits to 1, we can use [Nisan 1991, Proposition 2.2] ($bs(f) = C(f)$) to prove $D(f) \leq 256\, Q_2(f)^4$. For the case of exact computation we can also give a better result: Nisan and Smolensky proved $D(f) \leq 2\, deg(f)^4$ for any f (they never published this, but allowed their proof to be included in [Buhrman and Wolf 2001]). Together with our Theorem 4.3 this yields

THEOREM 5.5. *If f is a Boolean function, then $D(f) \leq 32\, Q_E(f)^4$.*

As a by-product, we improve the polynomial relation between $D(f)$ and $\widetilde{deg}(f)$. Nisan and Szegedy [Nisan and Szegedy 1994, Theorem 3.9] proved $\widetilde{deg}(f) \leq D(f) \leq 1296\, \widetilde{deg}(f)^8$. Using our result $D(f) \leq bs(f)^3$ and Nisan and Szegedy's $bs(f) \leq 6\, \widetilde{deg}(f)^2$ [Nisan and Szegedy 1994, Lemma 3.8] we obtain

COROLLARY 5.6. $\widetilde{deg}(f) \leq D(f) \leq 216\, \widetilde{deg}(f)^6$.

6. SOME PARTICULAR FUNCTIONS

In this section we consider the precise complexity of various specific functions.

First we consider the OR-function, which is related to search. By Grover's well-known search algorithm [Grover 1996; Boyer et al. 1998], if at least one x_i equals 1, we can find an index i such that $x_i = 1$ with high probability of success in $O(\sqrt{N})$ queries. This implies that we can also compute the OR-function with high success probability in $O(\sqrt{N})$: let Grover's algorithm generate an index i, and return x_i. Since $bs(\mathrm{OR}_N) = N$, Theorem 4.13 gives us a lower bound of $\frac{1}{4}\sqrt{N}$ on computing OR_N with bounded error probability (this $\Omega(\sqrt{N})$ bound was first shown in [Bennett et al. 1997] and is given in a tighter form in [Boyer et al. 1998; Zalka 1999], but the way we obtained it here is rather different from those proofs). Thus $Q_2(\mathrm{OR}_N) \in \Theta(\sqrt{N})$, where classically we require $\Theta(N)$ queries. Now suppose we want to get rid of the probability of error: can we compute OR_N exactly or with zero-error using $O(\sqrt{N})$ queries? If not, can quantum computation give us at least *some* advantage over the classical deterministic case? Both questions have a negative answer:

PROPOSITION 6.1. $Q_0(\mathrm{OR}_N) = N$.

PROOF. Consider a zero-error network for OR_N that uses $T = Q_0(\mathrm{OR}_N)$ queries. By Lemma 4.1, there are complex-valued polynomials p_k of degree at most T, such that the final state of the network on black-box X is

$$|\phi^X\rangle = \sum_{k \in K} p_k(X)|k\rangle.$$

Let B be the set of all basis states having 10 as rightmost bits (i.e., where the output is the answer 0). Then for every $k \in B$ we must have $p_k(X) = 0$ if $X \neq \vec{0} = (0, \ldots, 0)$, otherwise the probability of getting the incorrect answer 0 on $|\phi^X\rangle$ would be non-zero. On the other hand, there must be at least one $k' \in B$ such that $p_{k'}(\vec{0}) \neq 0$, since the probability of getting the correct answer 0 on $|\phi^{\vec{0}}\rangle$ must be non-zero. Let $p(X)$ be the real part of $1 - p_{k'}(X)/p_{k'}(\vec{0})$. This polynomial p has degree at most T and represents OR_N. But then p must have degree $deg(\mathrm{OR}_N) = N$, so $T \geq N$. □

COROLLARY 6.2. *A quantum network for exact or zero-error search requires N queries.*

In contrast, under the promise that the number of solutions is either 0 or t, for some fixed known t, exact search can be done in $O(\sqrt{N/t})$ queries [Brassard et al. 2000]. A partial block sensitivity argument (see the comment following Theorem 4.13) shows that this is optimal up to a multiplicative constant.

Like the OR-function, PARITY has $deg(\mathrm{PARITY}_N) = N$, so by Theorem 4.3 exact computation requires at least $\lceil N/2 \rceil$ queries. This is also sufficient. It is well known that the XOR of 2 variables can be computed using only one query [Cleve et al. 1998]. Assuming N even, we can group the variables of X as $N/2$ pairs: $(x_0, x_1), (x_2, x_3), \ldots, (x_{N-2}, x_{N-1})$, and compute the XOR of all pairs using $N/2$ queries. The parity of X is the parity of these $N/2$ XOR values, which can be computed without any further queries. If we allow bounded-error, then $\lceil N/2 \rceil$

queries of course still suffice. It follows from Theorem 4.8 that this cannot be improved, because $\widetilde{deg}(\text{PARITY}_N) = N$ [Minsky and Papert 1968]:

LEMMA 6.3 (MINSKY, PAPERT). $\widetilde{deg}(\text{PARITY}_N) = N$.

PROOF. Let f be PARITY on N variables. Let p be a polynomial of degree $\widetilde{deg}(f)$ that approximates f. Since p approximates f, its symmetrization p^{sym} also approximates f. By Lemma 3.2, there is a polynomial q, of degree at most $\widetilde{deg}(f)$, such that $q(|X|) = p^{sym}(X)$ for all inputs. Thus we must have $|f(X) - q(|X|)| \leq 1/3$, so

> $q(0) \leq 1/3$, $q(1) \geq 2/3$, ..., $q(N-1) \geq 2/3$, $q(N) \leq 1/3$ (assuming N even).

We see that the polynomial $q(x) - 1/2$ must have at least N zeroes, hence q has degree at least N and $\widetilde{deg}(f) = N$. □

PROPOSITION 6.4. *If f is PARITY on $\{0,1\}^N$, then $Q_E(f) = Q_0(f) = Q_2(f) = \lceil N/2 \rceil$.*[4]

Note that this result also implies that Theorems 4.3 and 4.8 are tight. For *classical* algorithms, N queries are necessary in the exact, zero-error, and bounded-error settings. Note that while computing PARITY on a quantum network is much harder than OR in the *bounded-error* setting ($\lceil N/2 \rceil$ versus $\Theta(\sqrt{N})$), in the *exact* setting PARITY is actually easier ($\lceil N/2 \rceil$ versus N).

The upper bound on PARITY uses the fact that the XOR connective can be computed with only one query. Using polynomial arguments, it turns out that XOR and its negation are the *only* examples among all 16 connectives on 2 variables where quantum gives an advantage over classical computation.

Since OR_N can be reduced to MAJORITY on $2N-1$ variables (if we set the first $N-1$ variables to 1, then the MAJORITY of all variables equals the OR of the last N variables) and OR requires N queries to be computed exactly or with zero-error, it follows that MAJ_N takes at least $(N+1)/2$ queries. Hayes, Kutin, and Van Melkebeek [Hayes et al. 1998] found an exact quantum algorithm that uses at most $N + 1 - w(N)$ queries, where $w(N)$ is the number of 1s in the binary representation of N; this can save up to $\log N$ queries. This also follows from classical results [Saks and Werman 1991; Alonso et al. 1993] that show that an item with the majority value can be identified classically deterministically with $N - w(N)$ *comparisons* between bits (a comparison between two input bits is the parity of the two bits, which can be computed with 1 quantum query). For the zero-error case, the same $(N+1)/2$ lower bound applies; Van Melkebeek, Hayes and Kutin give a zero-error quantum algorithm that works in roughly $\frac{2}{3}N$ queries. For the bounded-error case, we can apply Theorem 4.10: $\Gamma(\text{MAJ}_N) = 1$, so we need $Q_2(\text{MAJ}_N) \in \Theta(N)$ queries. The best upper bound we have here is $N/2 + \sqrt{N}$, which follows from [Dam 1998].

[4] This has also been proved independently by Farhi, Goldstone, Gutmann, and Sipser [Farhi et al. 1998], using a different technique. As noted independently by Terhal [Terhal 1997] and [Farhi et al. 1998], this result immediately implies results by Ozhigov [Ozhigov 1998] to the effect that no quantum computer can significantly speed up the computation of *all* functions (this follows because no quantum computer can significantly speed up the computation of PARITY).

The $\Omega(N)$ lower bound for MAJORITY also implies a lower bound for the number of comparisons required to *sort* N totally ordered elements. It is well known that $N \log N + \Theta(N)$ comparisons between elements are necessary and sufficient for sorting on a classical computer. Note that if we can sort then we can compute MAJORITY: if we sort the N-bit black-box then the bit at the $(N/2)$th position gives the MAJORITY-value (a comparison between 2 black-box bits can easily be simulated by a few queries). Hence our $\Omega(N)$-bound for MAJORITY implies:

COROLLARY 6.5. *Sorting N elements on a quantum computer takes at least $\Omega(N)$ comparisons.*

An $\Omega(N)$ lower bound for sorting was also derived independently in [Farhi et al. 1999a], via a different application of our polynomial-based method. The bound has recently been improved to the optimal $\Omega(N \log N)$ [Høyer et al. 2001].

ACKNOWLEDGMENTS

We thank Lance Fortnow for stimulating discussions on many of the topics treated here; Alain Tapp for sending us a preliminary version of [Brassard et al. 1998] and subsequent discussions about quantum counting; Andris Ambainis for sending us his proof that most functions cannot be computed with bounded-error using significantly fewer than N queries; Noam Nisan for sending us his and Roman Smolensky's proof that $D(f) \leq 2\ deg(f)^4$; Dieter van Melkebeek, Tom Hayes, and Sandy Kutin for their algorithms for MAJORITY; Hayes and Kutin for the reference to [Gathen and Roche 1997]; and two anonymous referees for some comments that improved the presentation of the paper.

REFERENCES

ALONSO, L., REINGOLD, E. M., AND SCHOTT, R. 1993. Determining the majority. *Information Processing Letters 47*, 5, 253–255.

AMBAINIS, A. 1999. A note on quantum black-box complexity of almost all Boolean functions. *Information Processing Letters 71*, 1, 5–7. quant-ph/9811080.

AMBAINIS, A. 2000. Quantum lower bounds by quantum arguments. In *Proceedings of 32nd ACM STOC* (2000), pp. 636–643. quant-ph/0002066.

BEALS, R., BUHRMAN, H., CLEVE, R., MOSCA, M., AND WOLF, R. DE 1998. Quantum lower bounds by polynomials. In *Proceedings of 39th IEEE FOCS* (1998), pp. 352–361. quant-ph/9802049.

BEIGEL, R. 1993. The polynomial method in circuit complexity. In *Proceedings of the 8th IEEE Structure in Complexity Theory Conference* (1993), pp. 82–95.

BENNETT, C. H., BERNSTEIN, E., BRASSARD, G., AND VAZIRANI, U. 1997. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing 26*, 5, 1510–1523. quant-ph/9701001.

BONEH, D. AND LIPTON, R. J. 1995. Quantum cryptanalysis of hidden linear functions (extended abstract). In *Advances in Cryptology (CRYPTO'95)*, Volume 963 of *Lecture Notes in Computer Science* (1995), pp. 424–437. Springer.

BOYER, M., BRASSARD, G., HØYER, P., AND TAPP, A. 1998. Tight bounds on quantum searching. *Fortschritte der Physik 46*, 4–5, 493–505. Earlier version in Physcomp'96. quant-ph/9605034.

BRASSARD, G. AND HØYER, P. 1997. An exact quantum polynomial-time algorithm for Simon's problem. In *Proceedings of the 5th Israeli Symposium on Theory of Computing and Systems (ISTCS'97)* (1997), pp. 12–23. quant-ph/9704027.

Brassard, G., Høyer, P., Mosca, M., and Tapp, A. 15 May 2000. Quantum amplitude amplification and estimation. quant-ph/0005055. This is the upcoming journal version of [Brassard et al. 1998; Mosca 1998], to appear in Quantum Computation and Quantum Information: A Millennium Volume, AMS Contemporary Mathematics Series.

Brassard, G., Høyer, P., and Tapp, A. 1997. Quantum algorithm for the collision problem. *ACM SIGACT News (Cryptology Column) 28*, 14–19. quant-ph/9705002.

Brassard, G., Høyer, P., and Tapp, A. 1998. Quantum counting. In *Proceedings of 25th ICALP*, Volume 1443 of *Lecture Notes in Computer Science* (1998), pp. 820–831. Springer. quant-ph/9805082.

Buhrman, H., Cleve, R., and Wigderson, A. 1998. Quantum vs. classical communication and computation. In *Proceedings of 30th ACM STOC* (1998), pp. 63–68. quant-ph/9802040.

Buhrman, H., Cleve, R., Wolf, R. de, and Zalka, Ch. 1999. Bounds for small-error and zero-error quantum algorithms. In *Proceedings of 40th IEEE FOCS* (1999), pp. 358–368. cs.CC/9904019.

Buhrman, H., Dürr, Ch., Heiligman, M., Høyer, P., Magniez, F., Santha, M., and Wolf, R. de 2001. Quantum algorithms for element distinctness. In *Proceedings of 16th IEEE Conference on Computational Complexity* (2001), pp. 131–137. quant-ph/0007016.

Buhrman, H. and Wolf, R. de 2001. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*. To appear.

Cleve, R. 2000. The query complexity of order-finding. In *Proceedings of 15th IEEE Conference on Computational Complexity* (2000), pp. 54–59. quant-ph/9911124.

Cleve, R., Ekert, A., Macchiavello, C., and Mosca, M. 1998. Quantum algorithms revisited. In *Proceedings of the Royal Society of London*, Volume A454 (1998), pp. 339–354. quant-ph/9708016.

Dam, W. van 1998. Quantum oracle interrogation: Getting all information for almost half the price. In *Proceedings of 39th IEEE FOCS* (1998), pp. 362–367. quant-ph/9805006.

Dam, W. van and Hallgren, S. 15 Nov 2000. Efficient quantum algorithms for shifted quadratic character problems. quant-ph/0011067.

Deutsch, D. and Jozsa, R. 1992. Rapid solution of problems by quantum computation. In *Proceedings of the Royal Society of London*, Volume A439 (1992), pp. 553–558.

Ehlich, H. and Zeller, K. 1964. Schwankung von Polynomen zwischen Gitterpunkten. *Mathematische Zeitschrift 86*, 41–44.

Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M. 19 Jan 1999b. Invariant quantum algorithms for insertion into an ordered list. quant-ph/9901059.

Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M. 1998. A limit on the speed of quantum computation in determining parity. *Physical Review Letters 81*, 5442–5444. quant-ph/9802045.

Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M. 1999a. How many functions can be distinguished with k quantum queries? *Physical Review A 60*, 6, 4331–4333. quant-ph/9901012.

Fenner, S., Fortnow, L., Kurtz, S., and Li, L. 1993. An oracle builder's toolkit. In *Proceedings of the 8th IEEE Structure in Complexity Theory Conference* (1993), pp. 120–131.

Fortnow, L. and Rogers, J. 1999. Complexity limitations on quantum computation. *Journal of Computer and Systems Sciences 59*, 2, 240–252. Earlier version in Complexity'98. Also cs.CC/9811023.

Gathen, J. von zur and Roche, J. R. 1997. Polynomials with two values. *Combinatorica 17*, 3, 345–362.

Grover, L. K. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th ACM STOC* (1996), pp. 212–219. quant-ph/9605043.

Grover, L. K. 1998. A framework for fast quantum mechanical algorithms. In *Proceedings of 30th ACM STOC* (1998), pp. 53–62. quant-ph/9711043.

Hayes, T., Kutin, S., and van Melkebeek, D. 1998. On the quantum complexity of majority. Technical Report TR-98-11, University of Chicago, Computer Science Department.

HØYER, P. 1999. Conjugated operators in quantum algorithms. *Physical Review A 59*, 5, 3280–3289.

HØYER, P., NEERBEK, J., AND SHI, Y. 2001. Quantum complexities of ordered searching, sorting, and element distinctness. In *Proceedings of 28th ICALP*, Volume 2076 of *Lecture Notes in Computer Science* (2001), pp. 346–357. Springer. quant-ph/0102078.

JOZSA, R. 1991. Characterizing classes of functions computable by quantum parallelism. In *Proceedings of the Royal Society of London*, Volume A435 (1991), pp. 563–574.

KITAEV, A. Y. 12 Nov 1995. Quantum measurements and the Abelian stabilizer problem. quant-ph/9511026.

MINSKY, M. AND PAPERT, S. 1968. *Perceptrons*. MIT Press, Cambridge, MA. Second, expanded edition 1988.

MOSCA, M. 1998. Quantum searching, counting and amplitude amplification by eigenvector analysis. In *MFCS'98 workshop on Randomized Algorithms* (1998), pp. 90–100.

MOSCA, M. AND EKERT, A. 1998. The hidden subgroup problem and eigenvalue estimation on a quantum computer. In *Proceedings of 1st NASA QCQC conference*, Volume 1509 of *Lecture Notes in Computer Science* (1998), pp. 174–188. Springer. quant-ph/9903071.

NAYAK, A. AND WU, F. 1999. The quantum query complexity of approximating the median and related statistics. In *Proceedings of 31st ACM STOC* (1999), pp. 384–393. quant-ph/9804066.

NIELSEN, M. A. AND CHUANG, I. L. 2000. *Quantum Computation and Quantum Information*. Cambridge University Press.

NISAN, N. 1991. CREW PRAMs and decision trees. *SIAM Journal on Computing 20*, 6, 999–1007. Earlier version in STOC'89.

NISAN, N. AND SZEGEDY, M. 1994. On the degree of Boolean functions as real polynomials. *Computational Complexity 4*, 4, 301–313. Earlier version in STOC'92.

OZHIGOV, Y. 1998. Quantum computer can not speed up iterated applications of a black box. In *Proceedings of 1st NASA QCQC conference*, Volume 1509 of *Lecture Notes in Computer Science* (1998). Springer. quant-ph/9712051.

PATURI, R. 1992. On the degree of polynomials that approximate symmetric Boolean functions (preliminary version). In *Proceedings of 24th ACM STOC* (1992), pp. 468–474.

RIVLIN, T. J. AND CHENEY, E. W. 1966. A comparison of uniform approximations on an interval and a finite subset thereof. *SIAM Journal on Numerical Analysis 3*, 2, 311–320.

SAKS, M. AND WIGDERSON, A. 1986. Probabilistic Boolean decision trees and the complexity of evaluating game trees. In *Proceedings of 27th IEEE FOCS* (1986), pp. 29–38.

SAKS, M. E. AND WERMAN, M. 1991. On computing majority by comparisons. *Combinatorica 11*, 4, 383–387.

SANTHA, M. 1991. On the Monte Carlo decision tree complexity of read-once formulae. In *Proceedings of the 6th IEEE Structure in Complexity Theory Conference* (1991), pp. 180–187.

SCHÖNING, U. 1999. A probabilistic algorithm for k-SAT and constraint satisfaction problems. In *Proceedings of 40th IEEE FOCS* (1999), pp. 410–414.

SERVEDIO, R. A. AND GORTLER, S. J. 12 Jul 2000. Quantum versus classical learnability. quant-ph/0007036.

SHOR, P. W. 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing 26*, 5, 1484–1509. Earlier version in FOCS'94. quant-ph/9508027.

SIMON, D. 1997. On the power of quantum computation. *SIAM Journal on Computing 26*, 5, 1474–1483. Earlier version in FOCS'94.

TERHAL, B. December 1997. Personal communication.

VAZIRANI, U. 1998. On the power of quantum computation. In *Proceedings of the Royal Society of London*, Volume A356 (1998), pp. 1759–1768.

WOLF, R. DE 2000. Characterization of non-deterministic quantum query and quantum communication complexity. In *Proceedings of 15th IEEE Conference on Computational Complexity* (2000), pp. 271–278. cs.CC/0001014.

Zalka, Ch. 1999. Grover's quantum searching algorithm is optimal. *Physical Review A 60*, 2746–2751. quant-ph/9711070.