Lower bound on the quantum query complexity of read-once functions: abstract

Howard Barnum* and Michael Saks†
* *CCS-3, MS B256, Los Alamos National Laboratory, Los Alamos, NM 87545;* barnum@lanl.gov
† *Dept. of Mathematics-Hill Center, 110 Frelinghuysen Road, Rutgers University, New Brunswick, NJ;* saks@math.rutgers.edu
(Dated: Aug 19, 2002)

We establish a lower bound of $\Omega(\sqrt{n})$ on the bounded-error quantum query complexity of read-once Boolean functions, providing evidence for the conjecture that $\Omega(\sqrt{D(f)})$ is a lower bound for all Boolean functions.

In the *quantum query* model of computation, the goal is to compute a function f of the string x which is accessed via queries. A state of the computer is: $\sum_{x,i,z} \alpha_{x,i,z}|x\rangle|i\rangle|z\rangle$, (where $\sum_{x,i,z} |\alpha_{x,i,z}|^2 = 1$). Here $|x\rangle \in H_{In}$ with $x \in \{0,1\}^n$; H_{In} is called the input register. We require $i \in \{0,1\}^{\lceil \log n+1 \rceil}$; $|i\rangle$ belongs to a *query register* containing an index specifying what bit of x will be queried if the oracle is called (or a null query), and $|z\rangle$ belongs to an auxiliary register with an unspecified (but finite, for a given computation) number of qubits. The query and auxiliary registers together form the *workspace* H_W.

The unitary operator O, the *oracle*, operates as follows:

$$O|x,i,z\rangle = (-1)^{x_i}|x,i,z\rangle \tag{1}$$

An algorithm is specified by (1) an arbitrary sequence $U_1,\ldots,U_t$ of unitary operators acting nontrivially only on the H_W and (2) a pair of orthogonal projectors P_0 on H_W satisfying $P_0 + P_1 = I_W$. It executes starting with the input register set to the input x and all other registers set to 0. Then the sequence $U_1, O, U_2, O, \ldots, U_t, O$ is applied to the computer. Thus the state of the computer is always of the form $|x\rangle \otimes |\Psi_x(t)\rangle$ where x is the input and $|\Psi_x(t)\rangle$ is a vector of H_W (generally not a standard basis state). If the final state of the computation is $|x\rangle \otimes |\Psi\rangle$ then the computation outputs j with probability equal to $\|P_j|\Psi\rangle\|^2$.

The complexity of the algorithm is measured by the number of calls t to the oracle. For $0 < \epsilon < 1/2$ a computation is said to *ϵ-compute f* if for every input, the probability that the algorithm gives the wrong answer for that input, is no greater than ϵ. The ϵ-error quantum query complexity of f, denoted $Q_\epsilon(f)$ is the minimum number of steps in an algorithm that ϵ-computes f. Up to a multiplicative constant, it is independent of $\epsilon \in (0,1/2)$.

In the bounded error model, quantum computation speeds up $f = OR$ of N variables quadratically (to $\sqrt{N}$ queries) over deterministic (and randomized) classical decision trees. This is the best speedup result known for a boolean total function.

Conjecture 1 *For any boolean function f and $\epsilon \in (0,1/2)$, $Q_\epsilon(f) = \Omega(D(f)^{1/2})$.*

The best known result of this type says that for any *total* f $Q_\epsilon(f) = \Omega(D(f)^{1/6})$. It was obtained in [2], via an extension to the quantum setting of the polynomials methods introduced in [3].

Our main result is to prove the conjecture for the class of *read-once functions*, those expressible by a boolean formula in which each variable (bit x_i of the input string) appears once. This is a quantum counterpart to the lower bounds on the randomized decision tree complexity of read-once functions given in [4] and [5].

The result is proved by an inductive argument, together with an extension of a lower bound method of Ambainis. We have learned of work by Høyer, Neerbek, and Shih [1] which applies a similar extension of Ambainis' method to other problems.

Ambainis' method involves viewing a quantum algorithm as inducing as a mapping from inputs x to quantum states $|\Psi_x(t)\rangle$ of the workspace, which changes as the computation proceeds. Initially, the mapping is constant: the initial state of the machine is independent of the input. If the algorithm ϵ-computes f then at the end of the computation the mapping must satisfy that the two states associated with any pair of inputs having different f values are nearly orthogonal. One carefully selects a set of f-distinguished pairs. The sum of the inner products of the corresponding states $|\psi_x(t)\rangle$ must decrease significantly during the computation. By deriving an upper bound on the decrease in the sum during a single step, one can obtain a lower bound on the number of steps.

We extend Ambainis' bound by considering general weighted sums of f-distinguished pairs. This allows one to optimize the tradeoff in the sum, for example, between abundant pairs of states requiring a certain variable x_i to be queried in order to distinguish them, and less abundant pairs (which might otherwise have too little impact on the sum) requiring a different variable to be queried, by giving the latter a larger relative weight μ_{xy}. We then prove the result for read-once functions by induction on the number of variables, where the induction step involves a careful choice of weights depending on f to optimize the lower bound attained.

Our lower bound on $Q_\epsilon(f)$ is expressed in terms of a complex vector $|\alpha\rangle$ of length 2^n indexed by inputs and a $2^n \times 2^n$ nonnegative real symmetric matrix Γ indexed by pairs of inputs, satisfying $\Gamma_{xy} = 0$ if $f(x) = f(y)$. Our weights decompose as $\mu_{xy} = \alpha_x \Gamma_{xy} \alpha_y$. For each

$i \in \{1,\ldots,n\}$ we define the Γ-dependent quantity

$$\nu_{x,i} = \sum_{y:x_i \neq y_i} \Gamma_{xy}, \quad (2)$$

the total Γ-weight of inputs differing from x on variable i. Further, for $i \in \{1,\ldots,n\}$ we define:

$$\nu_i = \max_{(x,y):\Gamma_{xy}\neq 0, x_i \neq y_i} \nu_{x,i}\nu_{y,i}$$
$$\nu = \max_{i\in\{1,\ldots,n\}} \nu_i.$$

Theorem 1 *Let f be an n-variate boolean function. Let $|\alpha\rangle$ be a nonnegative real valued vector indexed by $\{0,1\}^n$ and Γ be a nonnegative real symmetric matrix indexed by $\{0,1\}^n \times \{0,1\}^n$ satisfying $\Gamma_{x,y} = 0$ whenever $f(x) = 0$ or $f(y) = 1$. If there is a quantum algorithm that ϵ-computes f using t queries, then*

$$t \geq \frac{\langle\alpha|\Gamma|\alpha\rangle(1-2\sqrt{\epsilon(1-\epsilon)})}{\sqrt{\nu}} = \Omega\left(\frac{\langle\alpha|\Gamma|\alpha\rangle}{\sqrt{\nu}}\right). \quad (3)$$

(Buhrman and Szegedy (personal communication) have independently obtained a similar result.) This should be compared to Theorem 6 of [6].

For read-once functions, it suffices for Γ to be the characteristic function of a relation, but the nonuniformity of the coefficients α (even for inputs giving the same value of f) will be crucial. Then applying Theorem 1 gives:

Theorem 2 *$\Omega(\sqrt{n})$ is a lower bound on the bounded-error quantum query complexity of all read-once Boolean functions.*

A read-once function can be represented a rooted tree with n leaves, each corresponding to a different variable (with some possibly negated), with each internal node labeled either AND or OR. Each AND (OR) node in the tree is associated to a function which is defined recursively as the AND (OR) of the functions computed by its children. WLOG, we assume that all of the children of an AND node are OR nodes, and vice versa. Also, we consider *monotone* functions, those for which the leaves are nonnegated variables, since Q_ϵ is preserved under negation of variables.

The idea of the proof is to represent f by an AND/OR tree, and express f as $g^1 \wedge \ldots \wedge g^r$ where g^i are the functions computed at the children of the root labeled by AND. In choosing a Γ and $|\alpha\rangle$ for applying Theorem 1, we focus on *critical inputs*. An input is critical if for each AND node, at most one child evaluates to 0 and for each OR node, at most one child evaluates to 1. These are, intuitively, the inputs on which f is hardest to compute. (These play a similar role in the lower bound proofs for the randomized query complexity of read-once functions [4] [5]).

Evaluating each node in a tree on a given input gives the *evaluated tree* corresponding to that input, called a *critical tree* if the input is critical. A *critical child* of an node in a critical tree is one such that negating its value negates the value of its parent. Two critical trees are recursively defined to be neighbors if exactly one critical child of the root of one tree is negated compared to the other tree, and the subtrees rooted at that child are neighbors, while the subtrees rooted at the other children are identical. Two neighbors differ on exactly one input variable and consequently, for any critical input w and any $j \in \{1,\ldots,n\}$, w has at most one neighbor that differs from it on variable j.

We take $\Gamma_{xy} = 1$ if x, y are critical neighbors, 0 otherwise. Then ν of Theorem 1 is equal to 1 (for by the last sentence of the previous paragraph, the $\nu_{w,j}$ is at most 1 for any critical input w and $j \in \{1,\ldots,n\}$).

We choose $|\alpha\rangle$ to maximize the lower bound of Theorem 1 (given our particular choice of Γ), using Lagrange multipliers to obtain a set of first-order conditions. We then inductively construct $|\alpha\rangle$ that satisfies the first order conditions. Assume (essentially WLOG) that the root is an AND with r children, the i-th of which computes g^i. Write n_i for the number of (boolean) variables in g^i, so $n := \sum_{i=1}^r n_i$ is the number of (boolean) variables in f. Assume we have determined the optimal $|\alpha^i\rangle$ for each of the g_i. We construct $|\alpha\rangle$ in terms of these. Further we show that if $|\alpha^i\rangle$ gives a bound of $\kappa\sqrt{n_i}$ for each of the g_i, then $|\alpha\rangle$ gives a bound of $\kappa\sqrt{n}$ for f. We assume the coefficients α_x, α_y to be chosen so as to make the lower bound of theorem 1 (given the assumptions on Γ) maximal, not only for f, but for each of the g_i. With the base case of one query to evaluate $f(x_i) = x_i$, this inductively implies the bound of $\sqrt{n}$ for any AND/OR tree. In establishing the induction step, we make extensive use of the first-order conditions for Lagrange multiplier optimization of Theorem 1's lower bound expression (as simplified by our additional assumptions), both for f and for the g^i.

[1] P. Høyer, J. Neerbek, and Y. Shih, *Proc 28th Intl Colloq Automata, Languages, and Programming*, 2001.

[2] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf, *FOCS '98*, pp. 352–361, 1998.

[3] N. Nisan and M. Szegedy, *Comp Cplx*, pp. 301–313, 1994.

[4] M. Saks and A. Wigderson, *Proc 27th IEEE FOCS*, pp. 29–38, 1986.

[5] M. Santha, *Proceedings of the 6th IEEE Structure in Complexity Theory*, pp. 180–187, 1991.

[6] A. Ambainis, *Proc 32nd Ann ACM STOC*, p. 636, 2000.

[7] M. Szegedy, H. Barnum and M. Saks "Quantum decision trees and semidefinite programming," submitted, 2001.