An Almost-Quadratic Lower Bound for Quantum Formula Size

Vwani P. Roychowdhury† Farrokh Vatan‡
Electrical Engineering Department
UCLA
Los Angeles, CA 90095

Abstract

We show that Nechiporuk’s method [14] for proving lower bound for Boolean formulas can be extended to the quantum case. This leads to an \( \Omega(n^2 / \log^2 n) \) lower bound for quantum formulas computing an explicit function. The only known previous explicit lower bound for quantum formulas [15] states that the majority function does not have a linear-size quantum formula.

1 Introduction

Computational devices based on quantum physics have attracted much attention lately, and quantum algorithms that perform much faster than their classical counterparts have been developed [8, 10, 11]. To provide a systematic study of the computational power of quantum devices, models similar to those for classical computational devices have been proposed. Deutsch [3] formulated the notion of quantum Turing machine. This approach was further developed by Bernstein and Vazirani [8], and the concept of an efficient universal quantum Turing machine was introduced. As in the case of classical Boolean computation, there is

---

*This work was supported in part by grants from the Revolutionary Computing group at JPL (contract #961360), and from the DARPA Ultra program (subcontract from Purdue University #530–1415–01).
†e-mail: vwani@ee.ucla.edu
‡e-mail: vatan@ee.ucla.edu
also a quantum model of computation based on circuits (or networks). Yao [15] proved that the quantum circuit model, first introduced by Deutsch [6], is equivalent to the quantum Turing machine model.

Since every Boolean circuit can be simulated by a quantum circuit, with at most a polynomial factor increase in its size, any nontrivial lower bound for quantum circuits could have far reaching consequences. In classical Boolean circuit theory, all nontrivial lower bounds are for proper subclasses of Boolean circuits such as monotone circuits, formulas, bounded-depth circuits, etc. In the quantum case also it seems that the only hope to prove nontrivial lower bounds is for proper subclasses of quantum circuits. So far the only such known lower bound has been derived by Yao [15] for quantum formulas. The quantum formula is a straightforward generalization of the classical Boolean formula: in both cases, the graph of the circuit is a tree. Yao has proved that the quantum formula size of the majority function $\text{MAJ}_n$ is not linear; i.e., if $L(\text{MAJ}_n)$ denotes the minimum quantum formula size of $\text{MAJ}_n$ then $\lim_{n \to \infty} L(\text{MAJ}_n)/n = \infty$. This bound is derived from a bound on the quantum communication complexity of Boolean functions.

In this paper, we prove an almost quadratic lower bound for quantum formula size. The key step in the derivation of this lower bound is the extension of Nechiporuk’s method to quantum formulas; for a detailed discussion of Nechiporuk’s method in the Boolean setting see [7, 14]. Nechiporuk’s method has been used in several different areas of Boolean complexity (e.g., see [7] for details). It has also been applied to models where the gates do not take on binary or discrete values, but the input/output map still corresponds to a Boolean function. For example, in [12] this method has been used to get a lower bound for arithmetic and threshold formulas. The challenging part of this method is a step that we shall refer to as “path squeezing” (see Section 3 for the exact meaning of it). Although in the case of Boolean gates, this part can be solved easily, in the case of analog circuits it is far from obvious (see [12]). For the quantum formulas “path squeezing” becomes even more complicated, because here we should take care of any quantum entanglement and interference phenomenon. We show that it is still possible to squeeze a path with arbitrary number of constant inputs to a path with a fixed number of inputs. This leads to a lower bound of $\Omega(n^2/\log^2 n)$ on the size of quantum formulas computing a class of explicit functions. For example, we get such a bound for the Element Distinctness function

---

1There are exponential lower bounds on the time of quantum computation for the black–box model (see, e.g., [2]), but they do not apply to the size of quantum circuits.

2The value of $\text{MAJ}_n(x_1, \ldots, x_n)$ is 1 if at least $\lceil n/2 \rceil$ of inputs are 1.
EDₙ. The input of EDₙ is of the form (z₁, . . . , zₗ), where each zₖ is a string of 2 log ℓ bits. Then EDₙ(z₁, . . . , zₗ) = 1 if and only if all these strings are pairwise distinct.

In this paper we use the notation | · | for two different purposes. When α is a complex number, |α| denotes the absolute value of α; i.e., |α| = \sqrt{α \cdot α^*}. While if X is a set then |X| denotes the cardinality of X.

2 Preliminaries

A quantum circuit is defined as straightforward generalization of acyclic classical (Boolean) circuit (see [3]). For constructing a quantum circuit, we begin with a basis of quantum gates as elementary gates. Each such elementary gate with d inputs and outputs is uniquely represented by a unitary operation on ℂ²ᵈ. The gates are interconnected by quantum “wires”. Each wire represents a quantum bit, qubit, which is a 2–state quantum system represented by a unit vector in ℂ². Let {0|, 1|} be the standard orthonormal basis of ℂ². The 0| and 1| values of a qubit correspond to the classical Boolean 0 and 1 values, but a qubit can also be in a superposition of the form α 0| + β 1|, where α, β ∈ ℂ and |α|² + |β|² = 1. Each gate g with d inputs represents a unitary operation Uₓ ∈ U(2ᵈ). Note that the output of such gate, in general, is not a tensor product of its inputs, but an entangled state; e.g., a state like 1/√2 00| + 1/√2 11| which can not be written as a tensor product.

If the circuit has m inputs, then for each d–input gate g, the unitary operation Uₓ ∈ U(2ᵈ) can be considered in a natural way as an operator in U(2ᵐ) by acting as the identity operator on the other m − d qubits. Hence, a quantum circuit with m inputs computes a unitary operator in U(2ᵐ), which is the product of successive unitary operators defined by successive gates.

In this paper, we consider quantum circuits that compute Boolean functions. Consider a quantum circuit C with m inputs. Suppose that C computes the unitary operator U₃ ∈ U(2ᵐ). We say C computes the Boolean function f: {0, 1}ⁿ → {0, 1} if the following holds. The inputs are labeled by the variables x₁, x₂, . . . , xₙ or the constants 0| or 1| (different inputs may be labeled by the same variable xₗ). We consider one of the output wires, say the first one, as the output of the circuit. To compute the value of the circuit at α = (α₁, . . . , αₙ) ∈ {0, 1}ⁿ, let the value of each input wire with label xₗ be |αₗ|. These inputs, along with the constant inputs to the circuit, define a vector |α⟩ in ℂ²ᵐ. In fact this vector is a standard basis vector of the following form (up to some repetitions and a
permutation)

\[ |\alpha\rangle = |\alpha_1\rangle \otimes \cdots \otimes |\alpha_n\rangle \otimes |0\rangle \otimes \cdots \otimes |1\rangle \]

The act of the circuit \( C \) on the input \( |\alpha\rangle \) is the same as \( U_C(|\alpha\rangle) \). Note that since \( U_C \) is unitary, \( \| U_C(|\alpha\rangle) \| = 1 \). We decompose the vector \( U_C(|\alpha\rangle) \in \mathbb{C}^{2^m} \) with respect to the output qubit. Let the result be

\[ U_C(|\alpha\rangle) = |0\rangle \otimes |A_0\rangle + |1\rangle \otimes |A_1\rangle. \]

Then we define the probability that \( C \) outputs 1 (on the input \( \alpha \)) as \( p_{\alpha} = \| |A_1\rangle \|^2 \), i.e., the square of the length of \( |A_1\rangle \in \mathbb{C}^{2^m-1} \). Finally, we say that the quantum circuit \( C \) computes the Boolean function \( f \) if for every \( \alpha \in \{0, 1\}^n \), if \( f(\alpha) = 1 \) then \( p_{\alpha} > 2/3 \) and if \( f(\alpha) = 0 \) then \( p_{\alpha} < 1/3 \).

Following Yao [15], we define quantum formulas as a subclass of quantum circuits. A quantum circuit \( C \) is a formula if for every input there is a unique path that connects it to the output qubit. To make this definition more clear we define the computation graph of \( C \), denoted by \( G_C \). The nodes of \( G_C \) correspond to a subset of the gates of \( C \). We start with the output gate of \( C \), i.e., the gate which provides the output qubit, and let it be a node of \( G_C \). Once a node \( v \) belongs to \( G_C \) then all gates in \( C \) that provide inputs to \( v \) are considered as adjacent nodes of \( v \) in \( G_C \). Then \( C \) is a formula if the graph \( G_C \) is a tree. Figure 1 provides examples of quantum circuits of both kinds, i.e., circuits that are also quantum formulas, and circuits that are not formulas.

All circuits that we consider are over some fixed universal quantum basis. The lower bound does not depend on the basis; the only condition is that the number of inputs (and so the number of outputs) of each gate be bounded by some fixed constant number (this condition is usually considered as part of the definition of a quantum basis). For example, this basis can be the set of all 2-input 2-output quantum gates, and as as it is shown in [1], this basis is universal.

For our proof we also need a Shannon–type result for quantum circuits. Knill [9] has proved several theorems about the quantum circuit complexity of almost all Boolean functions. We will use the following theorem.

**Theorem 2.1** [9] The number of different \( n \)-variable Boolean functions that can be computed by size \( N \) quantum circuits \( (n \leq N) \) with \( d \)-input \( d \)-output elementary gates is at most \( 2^{cN \log N} \), where \( c \) is a constant which is a function of \( d \).
Figure 1: Quantum circuits and their computation graphs; the top circuit is not a formula while the bottom one is a formula.

For the sake of completeness, in the Appendix we have provided a proof for a slightly weaker bound. Our approach is different from that in [9] and it seems it is shorter and simpler than his proof. Although the bound that we get is a little weaker than the bound provided by the above theorem (it is of the form $2^{O(nN)}$), our bound results in the same bound if $\log(N) = \Omega(n)$. Thus our result provides the same bound for the complexity of the most difficult function and the bound we get in this paper.

We also need to consider orthonormal bases in the space $\mathbb{C}^{2^n}$ other than the standard basis. In the context of quantum physics, we identify the Hilbert space $\mathbb{C}^{2^n}$ as the tensor product space $\otimes_{j=1}^n \mathbb{C}^2$, and the standard basis consists of the vectors

$$|c_1\rangle \otimes |c_2\rangle \otimes \cdots \otimes |c_n\rangle = |c_1c_2\cdots c_n\rangle, \quad c_j \in \{0, 1\}.$$ 

The next lemma provides a method to construct other sets of mutually orthogonal unit vectors in $\mathbb{C}^{2^n}$.

**Lemma 2.2** Let $|A_j\rangle \in \mathbb{C}^{2^k}$ and $|B_j\rangle \in \mathbb{C}^{2^m}$ for $j = 1, 2$. Then for the length and inner product of the vectors $|A_j\rangle \otimes |B_j\rangle \in \mathbb{C}^{2^k+m}$ we have

$$||A_1\rangle \otimes |B_1\rangle|| = ||A_1\rangle|| \cdot ||B_1\rangle||,$$

and if $|x_1\rangle = |A_1\rangle \otimes |B_1\rangle$ and $|x_2\rangle = |A_2\rangle \otimes |B_2\rangle$ then

$$\langle x_1| x_2 \rangle = \langle A_1| A_2 \rangle \cdot \langle B_1| B_2 \rangle.$$
Proof. Suppose that $|A_1⟩ = \sum_{c \in \{0,1\}^k} \alpha_c |c⟩$ and $|B_1⟩ = \sum_{d \in \{0,1\}^m} \beta_d |d⟩$. Then

$$\| \langle A_1 | \otimes | B_1 \rangle \| = \left\| \sum_{c \in \{0,1\}^k} \sum_{d \in \{0,1\}^m} \alpha_c \beta_d |c⟩ \otimes |d⟩ \right\|$$

$$= \sum_{c \in \{0,1\}^k} \sum_{d \in \{0,1\}^m} |\alpha_c|^2 |\beta_d|^2$$

$$= \left( \sum_{c \in \{0,1\}^k} |\alpha_c|^2 \right) \cdot \left( \sum_{d \in \{0,1\}^m} |\beta_d|^2 \right)$$

$$= \| |A_1⟩ \| \cdot \| |B_1⟩ \|.$$ 

The proof in the case of the inner product is similar. \(\square\)

The above lemma can easily be generalized to the families of more than two vectors, and the generalized version is stated below.

**Lemma 2.3** Let $|A_j⟩ \in \mathbb{C}^{2^k}$ and $|B_ℓ⟩ \in \mathbb{C}^{2^m}$ be unit vectors (for $j$ and $ℓ$ in some index sets). If $|A_j⟩$ are pairwise orthogonal and $|B_ℓ⟩$ are pairwise orthogonal then the family

$$\left\{ |A_j⟩ \otimes |B_ℓ⟩ \in \mathbb{C}^{2^{k+m}} : j, ℓ \right\}$$

is an orthonormal set.

The following lemma, although seemingly obvious, is crucial for the “path squeezing” technique in the proof of the lower bound.

**Lemma 2.4** (a) Suppose that $C$ is a subcircuit of a quantum circuit. Let the inputs of $C$ be divided into two disjoint sets of qubits $Q_1$ and $Q_2$. Suppose that each gate of $C$ either acts only on qubits from $Q_1$ or only on qubits from $Q_2$. Then there are subcircuits $C_1$ and $C_2$ such that $C_j$ acts only on qubits from $Q_j$ and the operation of $C$ is the composition of operations of $C_1$ and $C_2$ no matter in which order they act; i.e., $C = C_1 \circ C_2 = C_2 \circ C_1$. So the subcircuit $C$ can be substituted by $C_1$ and $C_2$ (see Figure 2).

(b) Let $C$ be a quantum subcircuit with distinct input qubits $q$ and $r_1, \ldots, r_t$. Suppose that only $t$ gates $g_1, \ldots, g_t$ in $C$ act on $q$ and each $g_j$ acts on $q$ and $r_j$. Then, w.l.o.g., we can assume that each qubit $r_j$ after entering the gate $g_j$ will not interact with any other qubit until the gate $g_t$ is performed (see Figure 3).
Figure 2: Decomposition of a quantum subcircuit acting on disjoint sets of qubits (Lemma 2.4 (a)).

Figure 3: Postponing the gates (Lemma 2.4 (b)).

**Proof.** Part (a) is based on the following simple observation. If $M \in U(2^m)$ and $N \in U(2^n)$ then

$$M \otimes N = (M \otimes I_n) \circ (I_m \otimes N) = (I_m \otimes N) \circ (M \otimes I_n),$$

where $I_t$ is the identity map in $U(2^t)$. Note that the inputs of the subcircuit $C$ may be in an entangled state; but to see that the equality $C = C_1 \circ C_2 = C_2 \circ C_1$ holds, it is enough to check this equality for the standard basis and extend it to the whole space by linearity.

Part (b) follows simply from part (a); as in Figure 4, part (a) can be applied on subcircuit consisting of gates $h_2$ and $h_3$. Note that in this case also input qubits $r_j$ of $g_j$'s may be in an entangled state. Again a linearity argument shows that we have to consider only the case that $r_j$'s are in a product state. □
The lower bound

Let $f(x_1, \ldots, x_n)$ be a Boolean function, and let $X = \{x_1, \ldots, x_n\}$ be the set of the input variables. Consider a partition $\{S_1, \ldots, S_k\}$ of $X$; i.e., $X = \bigcup_{1 \leq j \leq k} S_j$ and $S_j \cap S_{j'} = \emptyset$, for $j_1 \neq j_2$. Let $n_j = |S_j|$, for $j = 1, \ldots, k$. Let $\Sigma_j$ be the set of all subfunctions of $f$ on $S_j$ obtained by fixing the variables outside $S_j$ in all possible ways. We denote the cardinality of $\Sigma_j$ by $\sigma_j$.

As an example, we compute the above parameters for the Element Distinctness function $\text{ED}_n$ (see [4]). Let $n = 2\ell \log \ell$ (so $\ell = \Omega(n/\log n)$) and divide the $n$ inputs of the function into $\ell$ strings each of $2\log \ell$ bits. Then the value of $\text{ED}_n$ is 1 if and only if these $\ell$ strings are pairwise distinct. We consider the partition $\langle S_1, \ldots, S_\ell \rangle$ such that each $S_j$ contains all variables of the same string. Thus $n_j = |S_j| = 2\log \ell$. Each string in $S_j$ represents an integer from the set $\{0, 1, \ldots, \ell^2-1\}$. The function $\text{ED}_n$ is symmetric with respect to $S_j$’s; so $|\Sigma_j| = |\Sigma_{j'}|$. To estimate $|\Sigma_1|$, note that if the strings $(z_2, \ldots, z_\ell)$ in $S_2, \ldots, S_\ell$ represent distinct integers then the corresponding subfunction is different from any subfunction corresponding to any other string. So $\sigma_j = |\Sigma_1| \geq \left(\frac{\ell^2}{\ell - 1}\right) > \ell^{\ell-1}$.

**Theorem 3.1** Every quantum formula computing $f$ has size

$$\Omega\left(\sum_{1 \leq j \leq k} \frac{\log(\sigma_j)}{\log(\log(\sigma_j))}\right).$$

**Proof.** We give a proof for any basis consisting of 2–input 2–output quantum gates. The proof for the other bases is a simple generalization of this proof.

Let $F$ be a formula computing $f$. Let $\overline{S}_j$ be the set of input wires of $F$ labeled by a variable from $S_j$, and let $s_j = |\overline{S}_j|$. Then

$$\text{size}(F) = \Omega\left(\sum_{1 \leq j \leq k} s_j\right). \quad (1)$$

3 The lower bound

![Figure 4: Changing the order of gates (Lemma 2.4 (b)).](image-url)
Figure 5: The qubits $q_1$ and $q_2$ are strong companions at step $\ell$, the qubits $q_1$ and $q_3$ are companions at step $\ell + 2$.

We want to consider the formulas obtained from $F$ by letting the variable inputs not in $\overline{S_j}$ to some constant value $|0\rangle$ or $|1\rangle$. In this regard, let $P_j$ be the set of all paths from an input wire in $\overline{S_j}$ to the output of $F$. Finally, let $G_j$ be the set of gates of $F$ where two paths from $P_j$ intersect. Then $|G_j| \leq s_j$.

Let $\rho$ be an assignment of $|0\rangle$ or $|1\rangle$ to the input variable wires not in $\overline{S_j}$. We denote the resulting formula by $F_{\rho}$. Thus $F_{\rho}$ computes a Boolean function $f_{\rho} : \{0, 1\}^{n_j} \rightarrow \{0, 1\}$ which is a subfunction of $f$ and a member of $\Sigma_j$. Consider a path

$$\pi = (g_1, g_2, \ldots, g_m), \quad m > 2, \quad (2)$$

in $F_{\rho}$, where $g_1$ is an input wire or a gate in $G_j$, $g_m$ is a gate in $G_j$ or the output wire of $F$, and $g_\ell \notin G_j$ for $1 < \ell < m$.

To show how we can squeeze paths like (2) (this is the essence of the Nechiporuk’s method), we introduce the following notations. We consider a natural ordering $\gamma_1, \gamma_2, \ldots, \gamma_t$ on the gates of the formula $F_{\rho}$, and regard $F_{\rho}$ as a computation in $t$ steps where at step $\ell$ the corresponding gate $\gamma_\ell$ is performed. We say two qubits $q_1$ and $q_2$ are strong companions of each other at step $\ell$ if there is a gate $\gamma_j$ such that $j \leq \ell$ and $q_1$ and $q_2$ are inputs of $\gamma_j$. We say qubits $q_1$ and $q_2$ are companions of each other at step $\ell$ if there exists a sequence $r_1, r_2, \ldots, r_p$ of qubits such that $r_1 = q_1$, $r_p = q_2$, and $r_j$ and $r_{j+1}$ (for $1 \leq j \leq p - 1$) are strong companions of each other at step $\ell$ (see Figure 5). If $q_1$ and $q_2$ are companions at step $\ell$ then they are also companions at any step after $\ell$. For a gate $g$, we define the set of companions of $g$ as the union of all companions of input qubits of $g$.

Suppose that in the path (2) $g_1 = \gamma_{j_0}$, $g_m = \gamma_{j_1}$, the inputs of $g_1$ are $q_0$ and $q_1$, the output of $\gamma_{j_0}$ from the path (2) is the qubit $q_0$, and the input of $\gamma_{j_1}$ not from the path (2) is the qubit $q_2$. Note that $q_0$ is the companion of $q_2$ at step $j_1$. Let $Q_{\pi}$ be union of all sets of companions of $g_1, \ldots, g_{m-1}$ at step $j_1$ minus $q_0$ and $q_1$ (see Figure 5). Let $C_0$ be the circuit defined by the gates $g_1, \ldots, g_{m-1}$ from the path (2). Suppose that $|Q_{\pi}| = v$ and consider $C_0$ as an operation acting on $\mathcal{H} = \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^{2^n}$, where $|\alpha_0 \rangle \otimes |\alpha_1 \rangle \otimes |\alpha \rangle \in \mathcal{H}$ denotes the state of $q_0, q_1$, and the companion qubits in $Q_{\pi}$, respectively. Note that all qubits in $Q_{\pi}$ are
constant inputs of $F_\rho$ and do not intersect any other path like (2), because $F$ is a formula. So the input $|\alpha\rangle$ of the subcircuit $C_0$ is the same for all $|\alpha_0\rangle$ and $|\alpha_1\rangle$. Therefore, we could substitute the subcircuit $C_0$ by $\tilde{C}_0$ such that on input $|\alpha_0\rangle |\alpha_1\rangle |0\cdots0\rangle$, the subcircuit $\tilde{C}_0$ first computes $|\alpha_0\rangle |\alpha_1\rangle |\alpha\rangle$ then applies the action of $C_0$. Suppose that the act of $\tilde{C}_0$ be defined as follows

$$\sum_{c_0,c_1=\{0,1\}} |c_0\rangle \otimes |c_1\rangle \otimes |A^{\alpha_0,\alpha_1}_{c_0,c_1}\rangle,$$

where $\alpha_0,\alpha_1 \in \{0,1\}$ and $|A^{\alpha_0,\alpha_1}_{c_0,c_1}\rangle \in \mathbb{C}^{2^n}$ may be not a unit vector. Let $A_\pi \subseteq \mathbb{C}^{2^n}$ be the vector space spanned by $|A^{\alpha_0,\alpha_1}_{c_0,c_1}\rangle$, for $\alpha_0,\alpha_1, c_0, c_1 \in \{0,1\}$ and $d = \dim(A_\pi)$. Then $1 \leq d \leq 16$. Let $|A_1^\top\rangle, \ldots, |A_d^\top\rangle$ be an orthonormal basis for $A_\pi$. Then we can rewrite (3) as follows

$$\sum_{c_0,c_1=\{0,1\}} \sum_{1 \leq j \leq d} \lambda^{\alpha_0,\alpha_1}_{j,c_0,c_1} |c_0\rangle \otimes |c_1\rangle \otimes |A_j^\top\rangle.$$

Let $M_\pi$ be the set of those unitary operations that are performed after one of the gates $g_1, \ldots, g_{m-1}$ on some qubits in $Q_\pi$ before the step $j_1$. Since qubits in $Q_\pi$ do not interact with any other path of the form (2), by Lemma 2.4 (b), we can postpone all operations in $M_\pi$ after we computed the output of $g_m$. Let $\pi_1, \ldots, \pi_k$ be a natural ordering on the paths in $P_j$ (i.e., the last gate of $\pi_{j+1}$ is not performed before the last gate of $\pi_j$). Consider the sets of postponed operations $M_{\pi_1}, \ldots, M_{\pi_k}$. Once again Lemma 2.4 implies that we can postpone operations in $M_{\pi_1}$ after the last gate of $\pi_2$, and so on. Repeating this argument shows that we can postpone all operations in $M_{\pi_1}, \ldots, M_{\pi_k}$ after we compute the output qubit. In this way, the state of the output qubit, before the postponed operations $M_{\pi_1}, \ldots, M_{\pi_k}$ is
applied, is of the form
\[ |0\rangle \otimes |M\rangle + |1\rangle \otimes |N\rangle, \tag{5} \]
where the first qubit is the output qubit and \(|M\rangle\) and \(|N\rangle\) are superpositions of tensor products of orthonormal vectors \(|A^\pi_j\rangle\) used in (4). By Lemma 2.3, these tensor products of the vectors \(|A^\pi_j\rangle\) are unit vectors and pairwise orthogonal. The unitary operations in the sets \(M_{\pi_j}\) (for paths \(\pi_j\) of the form (2)), which are postponed to the end, do not change the lengths of \(|M\rangle\) and \(|N\rangle\). Thus, as far as the computation of the Boolean function \(f_\rho\) is concerned, we can ignore all the postponed unitary operations. For this reason we construct the circuit \(\overline{F}_\rho\) from the formula \(F_\rho\) by eliminating all postponed operations in \(M_{\pi_j}\), substituting for each path \(\pi_j\) of the form (2) the companion qubits in \(Q_{\pi_j}\) by four new qubits, and the unitary operation (4) by the operation defined as
\[ |\alpha_0\rangle \otimes |\alpha_1\rangle \otimes |0000\rangle \longrightarrow \sum_{\alpha_0, c_1, c_2} \sum_{0 \leq j \leq 15} \lambda_{j, c_0, c_1}^{\alpha_0, \alpha_1} |\alpha_0\rangle \otimes |c_0\rangle \otimes |c_1\rangle \otimes |j\rangle. \tag{6} \]
The output of the circuit \(\overline{F}_\rho\), instead of (5), is of the form
\[ |0\rangle \otimes |M'\rangle + |1\rangle \otimes |N'\rangle, \tag{7} \]
where \(||M|| = ||M'||\) and \(||N|| = ||N'||\). So the circuit \(\overline{F}_\rho\) computes \(f_\rho\). Moreover,
\[ \text{size}(\overline{F}_\rho) = O(s_j), \]
and for another assignment \(\tau\), the corresponding circuit \(\overline{F}_\tau\) differs from \(\overline{F}_\rho\) only at unitary operations defined by (3).

The above discussion implies that \(\sigma_j\), the number of subfunctions on \(S_j\), is at most the number of different Boolean functions computed by size \(O(s_j)\) quantum circuits. Therefore, by Theorem 2.1, we get
\[ \sigma_j \leq 2^{O(s_j \log s_j)}. \]
So \(s_j = \Omega(\log(\sigma_j)/\log \log(\sigma_j))\). Now the theorem follows from (1). \(\square\)

To apply the general bound of the above theorem, we could consider any of the several explicit functions used in the case of Boolean formulas (see [7] and [14]). As we mentioned in the beginning of this section, we consider the Element Distinctness function \(ED_n\). For this function \(\sigma_j > \ell^{\ell - 1}\), where \(\ell = \Omega(n/\log n)\). Therefore, we get the lower bound \(\Omega(\ell^2) = \Omega(n^2/\log^2 n)\) for the formula size.

**Theorem 3.2** Any quantum formula computing \(ED_n\) has size \(\Omega(n^2/\log^2 n)\).
4 Concluding Remarks

We extend a classical technique for proving lower bound for Boolean formula size to quantum formulas. The difficult part was to effectively deal with the phenomenon of entanglement of qubits. While we have been successful in extending a classical technique to the quantum case, the challenges encountered indicate that in general the problem of extending methods of Boolean case to the quantum case may not have simple solutions. For example, even the seemingly simple issue of the exact relationship between quantum formulas and quantum circuits has not been resolved. In the Boolean case, simulation of circuits by formulas is a simple fact, but in the quantum case it is not clear whether every quantum circuit can be simulated by a quantum formula. In particular, it is not clear that in the process of going from quantum circuits to formulas, how we can modify the underlying entanglement of qubits while keeping the probability of reaching to the final answer the same.

References

[1] A. Barenco, C. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, “Elementary gates for quantum computation,” Phys. Rev. A, 52(1995), pp. 3457–3467.

[2] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf, Quantum lower bounds by polynomials, in Proceedings 39th IEEE Annual Symposium on Foundations of Computer Science, pp. 352-361, 1998.

[3] E. Bernstein and U. Vazirani, “Quantum complexity theory,” SIAM J. Computing, 26(1997), pp. 1411–1473.

[4] R. B. Boppana and M. Sipser, “The complexity of finite functions,” in Handbook of Theoretical Computer Science, Vol. A, “Algorithms and Complexity” (J. van Leeuwen, Ed.), pp. 757–804, Elsevier Science, New York, MIT Press, Cambridge, MA, 1990.

[5] D. Deutsch, “Quantum theory, the Church–Turing principle and the universal quantum computer,” Proc. Roy. Soc. London Ser. A., 400(1985), pp. 97–117.

[6] D. Deutsch, “Quantum computational networks,” Proc. Roy. Soc. London Ser. A., 425(1989), pp. 73–90.
A Appendix: Counting the number of Boolean functions computed by quantum circuits of a given size

In this appendix we prove the following upper bound.

Theorem A.1 The number of different $n$–variable Boolean functions that can be computed by size $N$ quantum circuits ($n \leq N$) with $d$–input $d$–output elementary gates (for some constant $d$) is at most $2^{O(nN) + O(N \log N)}$.

Our proof is based on Warren’s bound on the number of different sign–assignments to real polynomials [13]. We begin with some necessary notations.

Let $P_1(x_1, \ldots, x_t), \ldots, P_m(x_1, \ldots, x_t)$ be real polynomials. A sign–assignment to these polynomials is a system of inequalities

$$P_1(x_1, \ldots, x_t) \Delta_1 0, \ldots, P_m(x_1, \ldots, x_t) \Delta_m 0,$$

where $\Delta_i$ are symbols that can be $0$ or $1$. We may replace $\Delta_i$ by $\mu_i x_i$ where $\mu_i$ is an arbitrary real number. We can then group the $\mu_i$ into $d$ groups and replace $\mu_i$ by $\mu_i$, where $\mu_i$ is a bounded set of real numbers. This replaces the above inequalities by the following system of inequalities

$$P_1(x_1, \ldots, x_t) \Delta_1 0, \ldots, P_m(x_1, \ldots, x_t) \Delta_m 0.$$
where each $\Delta_j$ is either “<” or “>”. The sign-assignment (8) is called consistent if this system has a solution in $\mathbb{R}^t$.

**Theorem A.2** (Warren [13]) Let $P_1(x_1, \ldots, x_t), \ldots, P_m(x_1, \ldots, x_t)$ be real polynomials, each of degree at most $d$. Then there are at most $(4edm/t)^t$ consistent sign-assignments of the form (8).

We are now ready to prove Theorem 2.1. We consider the class of quantum circuits of size $N$ with $d$-bit gates computing $n$-variable Boolean functions. Without loss of generality, we can assume that $n'$, the number of input wires of such circuits, is at most $d \cdot N$. We define an equivalence relation $\cong$ on such circuits: we write $C_1 \cong C_2$ if and only if $C_1$ and $C_2$ differ only in the label of their gates; in another word, $C_1$ and $C_2$ have the same underlying graph but the corresponding gates in these circuits may compute different unitary operations. The number of different equivalence classes is at most $(n' d)^N \leq (dN)^{dN} = 2^{O(N \log N)}$.

Now we find an upper bound for the number of different Boolean functions that can be computed by circuits in the same equivalence class. Fix an equivalence class $\mathcal{E}$. We use the variables $a_1 + ib_1, a_2 + ib_2, \ldots, a_\mu + ib_\mu$, where $\mu = 2^{dN}$, to denote the entries of the matrices of the gates of a circuit $C$ in $\mathcal{E}$. By substituting appropriate values to the variables $a_1, \ldots, a_\mu, b_1, \ldots, b_\mu$, we get all circuits in $\mathcal{E}$. On input $\alpha = (\alpha_1, \ldots, \alpha_n) \in \{0, 1\}^n$, the probability that $C$ outputs 1 can be represented by a real polynomial $P_\alpha(a_1, \ldots, a_\mu, b_1, \ldots, b_\mu)$. The degree of $P_\alpha$ is at most $N^2$. There are $2^n$ polynomials $P_\alpha$ and the number of different Boolean functions can be computed by $C$ by changing the unitary operators of its gates is at most the number of different consistent sign-assignments to the following system:

$$P_\alpha(a_1, \ldots, a_\mu, b_1, \ldots, b_\mu) - \frac{2}{3} \text{ and } P_\alpha(a_1, \ldots, a_\mu, b_1, \ldots, b_\mu) - \frac{1}{3}, \quad \alpha \in \{0, 1\}^n.$$ 

By Theorem A.2 this number is bounded from the above by

$$\left(\frac{4eN^2 2^{n+1}}{2^\mu}\right)^{2\mu} = 2^{O(nN)+O(N \log N)}. \quad \square$$