Characterization of exact one-query quantum algorithms

Weijiang Chen,[1,2,*] Zekun Ye,[1,2,*] and Lvzhou Li[1,3,†]
[1]*Institute of Computer Science Theory, School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 510006, China*
[2]*Center for Quantum Computing, Peng Cheng Laboratory, Shenzhen 518055, China*
[3]*Ministry of Education Key Laboratory of Machine Intelligence and Advanced Computing, Sun Yat-Sen University, Guangzhou 510006, China*

(Received 2 December 2019; accepted 30 January 2020; published 20 February 2020)

The quantum query model is one of the most important models in quantum computing. Several well-known quantum algorithms are captured by this model, including the Deutsch-Jozsa algorithm, the Simon algorithm, the Grover algorithm, and others. In this paper, we characterize the computational power of exact one-query quantum algorithms. It is proved that a total Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ can be exactly computed by a one-query quantum algorithm if and only if $f(x) = x_{i_1}$ or $x_{i_1} \oplus x_{i_2}$ (up to isomorphism). Note that, unlike most work in the literature based on the polynomial method, our proof does not resort to any knowledge about the polynomial degree of f.

DOI: 10.1103/PhysRevA.101.022325

I. INTRODUCTION

The classical decision tree models have been well studied in classical computing and focus on problems such as the following: given a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$, how can we make as few queries as possible to the bits of x in order to output the value of $f(x)$? Quantum analogs, called quantum query models, have also attracted much attention in recent years [1]. The implementation procedure of a quantum query model is a quantum query algorithm, which can be roughly described as follows: it starts with a fixed state $|\psi_0\rangle$, and then performs the sequence of operations $U_0, O_x, U_1, \ldots, O_x, U_t$, where U_i's are unitary operators that do not depend on the input x but the query O_x does. This leads to the final state $|\psi_x\rangle = U_t O_x U_{t-1} \ldots U_1 O_x U_0 |\psi_o\rangle$. The result is obtained by measuring the final state $|\psi_x\rangle$.

The quantum query model can be discussed in two main settings: the exact setting and the bounded-error setting. A quantum query algorithm is said to compute a function f exactly, if its output equals $f(x)$ with probability 1, for all inputs x. It is said to *compute f with bounded error*, if its output equals $f(x)$ with a probability greater than a constant, for all inputs x. Roughly speaking, the query complexity of a function f is the number of queries that an optimal (classical or quantum) algorithm should make in the worst case to compute f. The classical deterministic query complexity of f is denoted by $D(f)$, and the quantum query complexity in the exact setting is denoted by $Q_E(f)$. In this paper, we focus on quantum query algorithms in the exact setting, which have been studied in much work [2–22]. And quantum advantages were shown by comparing $Q_E(f)$ and $D(f)$. For total Boolean functions, Beals *et al.* [23] showed that exact quantum query algorithms can only achieve polynomial speed-up over classical counterparts. On the other hand, Ambainis *et al.* [3] proved that exact quantum algorithms have advantage for almost all Boolean functions. However, the biggest gap between $Q_E(f)$ and $D(f)$ is only a factor of 2 and is achieved by Deutsch's algorithm for a long time. In 2013, a breakthrough result was obtained by Ambainis, showing the first total Boolean function for which exact quantum algorithms have superlinear advantage over classical deterministic algorithms [2]. Moveover, Ambainis [12] improved this result and presented a nearly quadratic separation in 2016. For partial functions, exponential separations between exact quantum and classical deterministic query complexity were obtained in several papers [4,11,24,25]. A typical example is the Deutsch-Jozsa algorithm [4].

In this paper, we consider the following problem: what functions can be computed exactly by one-query quantum algorithms (that can make only one query)? Our motivation comes from the following two aspects.

(i) Characterizing the computational power of a quantum computing model (or a kind of quantum algorithm) is of fundamental interest in the context of quantum complexity theory, and also is critical for discovering quantum advantage. Recently, characterization of one-query quantum algorithms in the bounded-error case has been considered by Aaronson *et al.* [26] and Arunachalam *et al.* [27]. But their results are not applicable to the exact case.

(ii) Actually, the well-known Deutsch algorithm and Deutsch-Jozsa algorithm belong to the class of exact one-query quantum algorithms. Then it is natural to ask what kind of functions (problems) can be computed exactly by one-query quantum algorithms.

We show that a total Boolean function f can be computed exactly by a one-query quantum algorithm if and only if $f(x) = x_{i_1}$ or $x_{i_1} \oplus x_{i_2}$ (up to isomorphism). It is worth pointing out that, unlike most work in the literature based on

*Both authors contributed equally to this work.
†lilvzh@mail.sysu.edu.cn

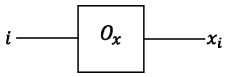

FIG. 1. Classical oracle.

the polynomial method, our proof does not depend on any knowledge about the polynomial degree of f. We hope this will illuminate a more general problem: what functions can be computed exactly by k-query quantum algorithms?

The remainder of this paper is organized as follows. The query models and the problem we consider are given in Sec. II. The main results of this paper are presented in Sec. III. Finally, a conclusion is made in Sec. IV and some further problems are proposed.

II. PRELIMINARIES

In this paper, we consider Boolean functions $f : \{0,1\}^n \to \{0,1\}$. Without special explanation, a function always means a total function, that is, it is defined for all $x \in \{0,1\}^n$. We will also refer to partial functions that are defined on a subset $D \subset \{0,1\}^n$. Throughout this paper, a function is assumed to be nonconstant, since the query complexity of a constant function is trivially zero. In the following, we first give an introduction about the query models, including both classical and quantum cases, and then we describe the problem to be discussed.

Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, suppose $x = x_1 x_2 \ldots x_n \in \{0,1\}^n$ is an input of f and we use x_i to denote its ith bit. The goal of a query algorithm is to compute $f(x)$, given queries to the bits of x.

In the classical case, the process of querying to x is implemented by using the black box, which we call the *query oracle*, as shown in Fig. 1. We want to compute $f(x)$ by using the query oracle as little as possible. A classical deterministic algorithm for computing f can be described by a *decision tree*. For example, suppose that we want to use a classical deterministic algorithm to compute $f(x) = x_1 \wedge (x_2 \vee x_3)$. Then a decision tree T for that is depicted in Fig. 2. Given an input x, the tree is evaluated as follows. It starts at the root. At each node, if it is a leaf, then its label is output as the result for $f(x)$; otherwise, it queries its label variable x_i. If $x_i = 0$, then we recursively evaluate the left subtree. Otherwise, we recursively evaluate the right subtree. The query complexity of tree T denoted by $D(T)$ is its depth, and we have $D(T) = 3$ in this example. Given f, there exist different decision trees to compute it. The query complexity of f, denoted by $D(f)$, is defined as

$$D(f) = \min_T D(T).$$

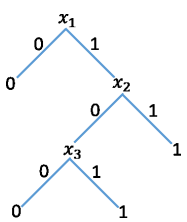

FIG. 2. A decision tree T for computing $f(x) = x_1 \wedge (x_2 \vee x_3)$.

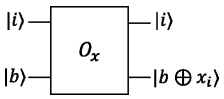

FIG. 3. Quantum oracle.

In the quantum case, the oracle is defined as $O_x|i,b\rangle = |i, b \oplus x_i\rangle$, where $i \in \{1,\ldots,n\}$, as shown in Fig. 3. Note that in this case we are able to query more than one bit each time due to quantum superposition. A T-query quantum algorithm can be seen as a sequence of unitaries $U_T O_x U_{T-1} O_x \ldots O_x U_0$, where U_i's are fixed unitaries and O_x depends on x (see Fig. 4).

The process of computation is as follows.

(1) Start with an initial state $|\psi_0\rangle$.

(2) Perform the operators $U_0, O_x, U_1, O_x \ldots U_T$ in sequence, and then we obtain the state $|\psi_x\rangle = U_T O_x U_{T-1} O_x \ldots U_0|\psi_0\rangle$.

(3) Measure $|\psi_x\rangle$ with a 0–1 positive operator-valued measurement [28]. The measurement result is regarded as the output of the algorithm.

In the above, we use $r(x)$ to denote the measurement result of $|\psi_x\rangle$. Let $P[\mathcal{A}]$ denote the probability that event $\mathcal{A}$ occurs. If it satisfies

$$\forall x, P[r(x) = f(x)] \geqslant 1 - \epsilon,$$

where $\epsilon < \frac{1}{2}$, then the quantum query algorithm is said to compute $f(x)$ with bounded error ϵ. If it satisfies

$$\forall x, P[r(x) = f(x)] = 1,$$

then it is said to compute $f(x)$ exactly.

The exact quantum query complexity of f, denoted by $Q_E(f)$, is the minimum number of queries that a quantum query algorithm needs to compute f. The gap between $D(f)$ and $Q_E(f)$ is usually used to exhibit quantum advantage.

In this paper, we want to characterize those functions f that satisfy $Q_E(f) = 1$. In other words, we consider this problem: what functions f can be computed exactly by a one-query quantum algorithm? In this case a quantum query algorithm is as shown in Fig. 5.

III. MAIN RESULT

Two functions f and g over $\{0,1\}^n$ are *isomorphic* if they are equal up to negations and permutations of the input variables, and negation of the output variable. It is easy to see that for any two isomorphic Boolean functions f and g we have $Q_E(f) = Q_E(g)$. Now our main result is as follows.

Theorem 1. A total Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be computed exactly by a one-query quantum algorithm, if and only if $f(x) = x_{i_1}$ or $x_{i_1} \oplus x_{i_2}$ (up to isomorphism).

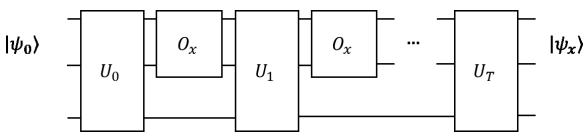

FIG. 4. T-query quantum algorithm.

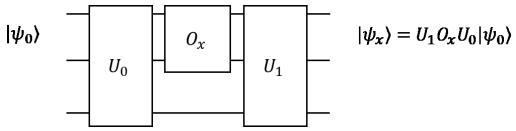

FIG. 5. One-query quantum algorithm.

Remark 1. In the above theorem, we consider only total functions. A more interesting problem is to characterize the partial functions (promise problems) that can be computed exactly by one-query quantum algorithms. This problem seems to be more complicated, and the method used here may not be applicable to partial functions.

First, it is easy to see the sufficiency. If $f(x) = x_{i_1} \oplus x_{i_2}$, then it can be computed exactly by the Deutsch algorithm. If $f(x) = x_{i_1}$, there obviously exists a one-query quantum algorithm to do that. Therefore, the key to prove Theorem 1 is the necessity. For that we will prove the following: (a) if f can be exactly computed by a one-query quantum algorithm, then it depends on at most two variables, and (b) furthermore it must be in the above form.

Definition 1. For a total Boolean function $f : \{0,1\}^n \to \{0,1\}$, f is said to depend on the ith variable of the input, if there exists $x \in \{0,1\}^n$ such that $f(x) \neq f(x^i)$, where x^i is the same as x except for the ith bit being flipped.

For example, if $f(x) = x_{i_1} \oplus x_{i_2}$, then f depends on the i_1th and i_2th variables.

Proof of necessity

Now suppose that $f : \{0,1\}^n \to \{0,1\}$ can be computed exactly by a one-query quantum algorithm as shown in Fig. 5. We denote

$$U_0|\psi_0\rangle = \sum_{i,j,k} \alpha_{ijk}|i\rangle|j\rangle|k\rangle,$$

where $i \in \{1,2,\ldots,n\}$, $j \in \{0,1\}$, and $|k\rangle$ is an arbitrary ancilla register, corresponding to the third line in Fig. 5. We have the following observation.

Lemma 1. For a pair of inputs x, y such that $f(x) \neq f(y)$, let $S = \{i | x_i \neq y_i\}$. Then it holds that

$$\sum_{i\in S}\sum_{k} |\alpha_{i0k} - \alpha_{i1k}|^2 = 1. \tag{1}$$

Proof. Note that $|\psi_x\rangle = U_1 O_x U_0|\psi_0\rangle$ and let $|\phi_x\rangle = O_x U_0|\psi_0\rangle$. Then we get

$$|\phi_x\rangle = O_x \sum_{i,j,k} \alpha_{ijk}|i\rangle|j\rangle|k\rangle = \sum_{i,j,k} \alpha_{ijk}|i\rangle|j \oplus x_i\rangle|k\rangle.$$

The assumption that f can be computed exactly implies that $|\psi_x\rangle$ and $|\psi_y\rangle$ can be perfectly distinguished for x, y satisfying $f(x) \neq f(y)$. Thus, the two states are mutually orthogonal, that is,

$$\langle\psi_x|\psi_y\rangle = 0.$$

Since unitary operators do not change the orthogonality between two states, equivalently, there is

$$\langle\phi_x|\phi_y\rangle = 0,$$

from which it follows that

$$\begin{aligned}\langle\phi_x|\phi_y\rangle &= \sum_{i,j,k} \alpha^*_{ijk}\langle i|\langle j \oplus x_i|\langle k| \sum_{p,q,r} \alpha_{pqr}|p\rangle|q \oplus y_p\rangle|r\rangle \\ &= \sum_{i,j,k,q} \alpha^*_{ijk}\alpha_{iqk}\langle j \oplus x_i | q \oplus y_i\rangle = 0. \end{aligned} \tag{2}$$

Furthermore, Eq. (2) can be rewritten as

$$\sum_{i\notin S}\sum_{k}(|\alpha_{i0k}|^2 + |\alpha_{i1k}|^2) + \sum_{i\in S}\sum_{k}(\alpha^*_{i0k}\alpha_{i1k} + \alpha^*_{i1k}\alpha_{i0k}) = 0. \tag{3}$$

Moreover, note that $\sum_{i,j,k} |\alpha_{ijk}|^2 = 1$, that is,

$$\sum_{i\notin S}\sum_{k}(|\alpha_{i0k}|^2 + |\alpha_{i1k}|^2) + \sum_{i\in S}\sum_{k}(|\alpha_{i0k}|^2 + |\alpha_{i1k}|^2) = 1. \tag{4}$$

Therefore, by subtracting Eq. (3) from Eq. (4), we obtain

$$\sum_{i\in S}\sum_{k}[(|\alpha_{i0k}|^2 + |\alpha_{i1k}|^2) - (\alpha^*_{i0k}\alpha_{i1k} + \alpha^*_{i1k}\alpha_{i0k})] = 1,$$

which is equivalent to

$$\sum_{i\in S}\sum_{k} |\alpha_{i0k} - \alpha_{i1k}|^2 = 1.$$

■

Now we are ready for proving the necessity.

Lemma 2. If a total Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be computed exactly by a one-query quantum algorithm, then f depends on at most two variables of x, and furthermore f is in the form $f(x) = x_{i_1}$ or $x_{i_1} \oplus x_{i_2}$ (up to isomorphism).

Proof. We denote the Hamming distance of x and y by $d(x,y)$. For a total function f, if f depends on some variable x_i, then there exist $x, y \in \{0,1\}^n$ such that $d(x,y) = 1$, $x_i \neq y_i$ and $f(x) \neq f(y)$. Thus, in this case $S = \{i\}$, and by Lemma 1 we have

$$\sum_{k} |\alpha_{i0k} - \alpha_{i1k}|^2 = 1.$$

Now suppose f depends on t variables $x_{i_1}, x_{i_2}, \ldots, x_{i_t}$. It follows that

$$\sum_{r=1}^{t}\sum_{k} |\alpha_{i_r 0k} - \alpha_{i_r 1k}|^2 = t.$$

Furthermore, we have

$$\sum_{r=1}^{t}\sum_{k} |\alpha_{i_r 0k} - \alpha_{i_r 1k}|^2 \leqslant 2\sum_{r=1}^{t}\sum_{k} |\alpha_{i_r 0k}|^2 + |\alpha_{i_r 1k}|^2 \leqslant 2,$$

where the first equality follows from the observation that $|a-b|^2 \leqslant 2(|a|^2 + |b|^2)$ for any two complex numbers a, b, and the second equality holds because we have $\sum_{i,j,k} |\alpha_{ijk}|^2 = 1$. Thus, we get $t \leqslant 2$, that is, f depends on at most two variables.

Next, we prove $f(x) = x_{i_1}$ or $x_{i_1} \oplus x_{i_2}$ (up to isomorphism). Suppose that f depends on at most two variables x_{i_1} and x_{i_2}. Denote $C_{00} = \{x | x_{i_1} = x_{i_2} = 0\}$, $C_{01} = \{x | x_{i_1} = 0, x_{i_2} = 1\}$, $C_{10} = \{x | x_{i_1} = 1, x_{i_2} = 0\}$, $C_{11} = \{x | x_{i_1} = 1, x_{i_2} = 1\}$. Let $S_0 = \{x | f(x) = 0\}$ and $S_1 = \{x | f(x) = 1\}$. Without loss of generality, suppose $C_{00} \subseteq S_0, C_{10} \subseteq S_1$. Below we show

$f = x_{i_1}$ or $f = x_{i_1} \oplus x_{i_2}$. Other cases can be discussed in a similar way.

(1) $C_{01} \subseteq S_0, C_{11} \subseteq S_1$. In this case, we have $f = x_{i_1}$.

(2) $C_{01} \subseteq S_1, C_{11} \subseteq S_0$. In this case, we have $f = x_{i_1} \oplus x_{i_2}$.

(3) $C_{01} \subseteq S_0, C_{11} \subseteq S_0$. In this case, $f = x_{i_1} \wedge \neg x_{i_2}$, which is isomorphic to AND_2 that cannot be computed by one-query quantum algorithms as indicated in [3]. Thus, this case is impossible.

(4) $C_{01} \subseteq S_1, C_{11} \subseteq S_1$. In this case, $f = x_{i_1} \vee x_{i_2}$, which is also isomorphic to AND_2. Thus, this case is impossible. ■

IV. CONCLUSION

In this paper we have characterized the power of exact one-query quantum algorithms for total functions. In conclusion, a total Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ can be computed by exactly a one-query algorithm if and only if $f(x) = x_i$ or $x_{i_1} \oplus x_{i_2}$ (up to isomorphism). Note that, unlike most work in the literature based on the polynomial method, our proof does not resort to any knowledge about the polynomial degree of f. We hope it will illuminate two more general problems that are worthy of further consideration as follows.

A. Characterization of partial functions that can be computed exactly by a one-query quantum algorithm

Given a partial Boolean function $f : D \to \{0, 1\}$, where $D \subset \{0, 1\}^n$, how do we determine whether there exists a one-query quantum algorithm that computes it exactly? Furthermore, can we discover all those partial functions f that can be computed exactly by a one-query quantum algorithm?

B. Characterization of functions that can be computed exactly by a k-query quantum algorithm

A more interesting problem is to discuss the power of exact k-query quantum algorithms, although for some functions with a specific property we know their quantum query complexity. There is no a general conclusion to characterize the power of exact k-query quantum algorithms. Figuring out this problem is useful for further understanding quantum query algorithms and inspiring us to find more problems with quantum advantage.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 61772565), the Natural Science Foundation of Guangdong Province of China (Grant No. 2017A030313378), the Science and Technology Program of Guangzhou City of China (Grant No. 201707010194), the Key Research and Development project of Guangdong Province (Grant No. 2018B030325001), and the Open Research Fund from State Key Laboratory of High Performance Computing of China (Grant No. 201901-03).

[1] H. Buhrman and R. de Wolf, Theor. Comput. Sci. **288**, 21 (2002).

[2] A. Ambainis, in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing* (ACM Press, New York, 2013), pp. 891–900.

[3] A. Ambainis, J. Gruska, and S. Zheng, Quantum Inf. Comput. **15**, 435 (2015).

[4] D. Deutsch and R. Jozsa, Proc. R. Soc. Lond. A **439**, 553 (1992).

[5] G. Midrijanis, arXiv:quant-ph/0403168 (2004).

[6] T. Mihara and S. Sung, Comput. Complex. **12**, 162 (2003).

[7] X. He, X. Sun, G. Yang, and P. Yuan, arXiv:1801.05717 (2018).

[8] A. Montanaro, R. Jozsa, and G. Mitchison, Algorithmica **71**, 775 (2015).

[9] A. Ambainis, J. Iraids, and J. Smotrovs, in *Proceedings of the Eighth Conference on the Theory of Quantum Computation, Communication, and Cryptography*, edited by S. Severini and F. Brandao (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2013), Vol. 22, pp. 263–269.

[10] A. Ambainis, J. Iraids, and D. Nagaj, in *Proceedings of the 43rd International Conference on Current Trends in Theory and Practice of Computer Science*, Theoretical Computer Science and General Issues (Springer International Publishing, Switzerland AG, 2017), pp. 243–255.

[11] G. Cai and D. Qiu, J. Comput. Syst. Sci. **97**, 83 (2018).

[12] A. Ambainis, SIAM J. Comput. **45**, 617 (2016).

[13] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Proc. R. Soc. Lond. A **454**, 339 (1998).

[14] A. Vasilieva and T. Mischenko-Slatenkova, arXiv:quant-ph/0607022 (2006).

[15] S. Aaronson, S. Ben-David, and R. Kothari, in *Proceedings of the 48th Annual ACM Symposium on Theory of Computing* (ACM Press, New York, 2016), pp. 863–876.

[16] A. Ambainis, K. Balodis, A. Belovs, T. Lee, M. Santha, and J. Smotrovs, J. ACM **64**, 1 (2017).

[17] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Phys. Rev. Lett. **81**, 5442 (1998).

[18] T. Hayes, S. Kutin, and D. van Melkebeek, Algorithmica **34**, 480 (2002).

[19] T. Mischenko-Slatenkova, A. Vasilieva, I. Kucevalovs, and R. Freivalds, in *Descriptional Complexity of Formal Systems: DCFS 2015*, edited by J. Shallit and A. Okhotin, Lecture Notes in Computer Science, Vol. 9118 (Springer, Cham, New York, 2015), pp. 177–184.

[20] S. L. Braunstein, B. Choi, S. Ghosh, and S. Maitra, J. Phys. A. **40**, 8441 (2007).

[21] G. Brassard and P. Høyer, in *Proceedings of the Fifth Israel Symposium on Theory of Computing and Systems, ISTCS 1997* (IEEE Press, New York, 1997), pp. 12–23.

[22] D. Qiu and S. Zheng, Phys. Rev. A **97**, 062331 (2018).

[23] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf, J. ACM. **48**, 778 (2001).

[24] P. W. Shor, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE Press, New York, 1994), pp. 124–134.

[25] D. R. Simon, SIAM J. Comput. **26**, 1474 (1997).

[26] S. Aaronson, A. Ambainis, J. Iraids, M. Kokainis, and J. Smotrovs, in *Proceedings of the 31st Conference on Computational Complexity (CCC 2016)*, edited by Ran Raz (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany), pp. 25:1–25:19.

[27] S. Arunachalam, J. Briet, and C. Palazuelos, SIAM J. Comput. **48**, 903 (2019).

[28] M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2002).