Information Processing Letters 112 (2012) 289–291

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Superiority of exact quantum automata for promise problems

Andris Ambainis [1], Abuzer Yakaryılmaz [*,2]

University of Latvia, Faculty of Computing, Raina bulv. 19, Rīga, LV-1586, Latvia

ARTICLE INFO

Article history:
Received 16 August 2011
Received in revised form 28 December 2011
Accepted 2 January 2012
Available online 4 January 2012
Communicated by J. Torán

Keywords:
Computational complexity
Exact quantum computation
Promise problems
Succinctness
Quantum finite automaton
Classical finite automaton

ABSTRACT

In this note, we present an infinite family of promise problems which can be solved exactly by just tuning transition amplitudes of a two-state quantum finite automaton operating in realtime mode, whereas the size of the corresponding classical automata grows without bound.

1. Introduction

The exact quantum computation has been widely examined for both partial (promise) and total functions (e.g. [5,3,2,6,11,7,14,9,16]). On the other hand, in automata theory, only two results have been obtained:

(i) Klauck [11] has shown that realtime quantum finite automata (QFAs) cannot be more concise than realtime deterministic finite automata (DFAs)[3] in case of language recognition, and
(ii) Murakami et al. [14] have shown that there is a promise problem solvable by quantum pushdown automata but not by any deterministic pushdown automaton.

In this note, we consider succinctness of realtime QFAs for promise problems. We present an infinite family of promise problems which can be solved exactly by just tuning transition amplitudes of a two-state realtime QFAs, whereas the size of the corresponding classical automata grows without bound.

2. Background

Throughout the paper,

(i) Σ denotes the input alphabet not containing left- and right-end markers (¢ and \$, respectively), and $\tilde{\Sigma} = \Sigma \cup \{¢, \$\}$,
(ii) ε is the empty string,
(iii) w_i is the ith symbol of a given string w, and
(iv) $\tilde{w}$ represents the string $¢w\$$, for $w \in \Sigma^*$.

Moreover, all machines presented in the paper operate in realtime mode. That is, the input head moves one square to the right in each step, and the computation stops after reading \$.

[*] Corresponding author.
E-mail addresses: ambainis@lu.lv (A. Ambainis), abuzer@lu.lv (A. Yakaryılmaz).

[1] Ambainis was supported by ESF project 1DP/1.1.1.2.0/09/APIA/VIAA/044.

[2] Yakaryılmaz was partially supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) with grant 108E142 and FP7 FET-Open project QCS.

[3] The proof was basically given for Kondacs–Watrous realtime QFA model [12] but it can be extended for any model of realtime QFAs including the most general ones [8,4,10,17].

0020-0190/\$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2012.01.001

A promise problem is a pair $A = (A_{yes}, A_{no})$, where $A_{yes}, A_{no} \subseteq \Sigma^*$ and $A_{yes} \cap A_{no} = \emptyset$ [15]. A promise problem $A = (A_{yes}, A_{no})$ is solved exactly by a machine $\mathcal{M}$ if each string in A_{yes} (resp., A_{no}) is accepted (resp., rejected) exactly by $\mathcal{M}$. Note that language recognition is a special case of solving promise problems, i.e. $A_{yes} \cup A_{no} = \Sigma^*$ in case of language recognition.

We give our quantum result for the most restricted of the known QFA models, i.e. *Moore–Crutchfield quantum finite automaton* (MCQFA) [13] (see [17] for the definition of the most general QFA model).

An MCQFA is a 5-tuple

$$\mathcal{M} = \big(Q, \Sigma, \{U_\sigma \mid \sigma \in \tilde{\Sigma}\}, q_1, Q_a\big),$$

where $Q = \{q_1, \ldots, q_n\}$ is the set of states, q_1 is the initial state, $Q_a \subseteq Q$ is the set of accepting states, and U_σ's are unitary operators. The computation of an MCQFA on a given input string $w \in \Sigma^*$ can be traced by a $|Q|$-dimensional vector. This vector is initially set to $|v_0\rangle = (1\ 0\ \cdots\ 0)^T$, and evolves according to

$$|v_i\rangle = U_{\tilde{w}_i}|v_{i-1}\rangle, \quad 1 \leqslant i \leqslant |\tilde{w}|.$$

At the end of the computation, w is accepted (resp., rejected) with probability $\|P_a|v_{|\tilde{w}|}\rangle\|^2$ (resp., $\|P_r|v_{|\tilde{w}|}\rangle\|^2$), where $P_a = \sum_{q \in Q_a} |q\rangle\langle q|$ and $P_r = I - P_a$.

If we replace each unitary operator with a left stochastic operator, we obtain a realtime probabilistic finite automaton (which we call simply a PFA). A PFA is a 5-tuple

$$\mathcal{P} = \big(Q, \Sigma, \{A_\sigma \mid \sigma \in \tilde{\Sigma}\}, q_1, Q_a\big),$$

where A_σ's are left stochastic operators. The computation of a PFA on a given input string $w \in \Sigma^*$ can be traced by a $|Q|$-dimensional vector. This vector is initially set to $v_0 = (1\ 0\ \cdots\ 0)^T$, and evolves according to

$$v_i = A_{\tilde{w}_i} v_{i-1}, \quad 1 \leqslant i \leqslant |\tilde{w}|.$$

At the end of the computation, w is accepted (resp., rejected) with probability $\sum_{q_i \in Q_a} v_{|\tilde{w}|}[i]$ (resp., $\sum_{q_i \in (Q \setminus Q_a)} v_{|\tilde{w}|}[i]$). If we allow only zero–one entries in each stochastic operator, we obtain a realtime DFA (which we call simply a DFA).

3. The main results

Let $A^k_{yes} = \{a^{i2^k} \mid i$ is a nonnegative *even* integer$\}$ and $A^k_{no} = \{a^{i2^k} \mid i$ is a positive *odd* integer$\}$ be two unary languages, where k is a positive integer. We will show that a two-state MCQFA can solve promise problem $A^k = (A^k_{yes}, A^k_{no})$ exactly, but any DFA must have at least 2^{k+1} states to solve the same problem.

Theorem 1. *Promise problem $A^k = (A^k_{yes}, A^k_{no})$ can be solved by a two-state MCQFA $\mathcal{M}_k$ exactly.*

Proof. We will use a well-known technique given in [1]. Let $N = 2^k$ and $\mathcal{M}_k = (Q, \Sigma, \{U_\sigma \mid \sigma \in \tilde{\Sigma}\}, q_1, Q_a)$, where $Q = \{q_1, q_2\}$, $\Sigma = \{a\}$, $Q_a = \{q_1\}$, $U_{¢} = U_{\$} = I$, and U_a is a rotation in $|q_1\rangle$–$|q_2\rangle$ plane with angle $\theta = \frac{\pi}{2N}$, i.e.

$$U_a = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

The computation begins with $|q_1\rangle$ and, after reading each block of N a's, the following pattern is followed by $\mathcal{M}_k$:

$$|q_1\rangle \xrightarrow{a^N} |q_2\rangle \xrightarrow{a^N} -|q_1\rangle \xrightarrow{a^N} -|q_2\rangle \xrightarrow{a^N} |q_1\rangle \xrightarrow{a^N} \cdots.$$

Therefore, it is obvious that $\mathcal{M}_k$ solves promise problem A^k exactly. □

Lemma 1. *Any DFA solving $A^k = (A^k_{yes}, A^k_{no})$ must have at least 2^{k+1} states.*

Note. We give a classical lower bound for DFAs since PFAs cannot be concise than DFAs in the case of solving promise problem exactly.[4]

Proof. Let $N = 2^k$ and $\mathcal{D}$ be an m-state DFA solving A^k. We show that m cannot be less than $2N$.

Since both A^k_{yes} and A^k_{no} contain infinitely many unary strings, there must be a chain of t states, say $s_0, \ldots, s_{t-1}$, such that, for sufficiently long strings, $\mathcal{D}$ enters this chain in which $\mathcal{D}$ transmits from s_i to $s_{(i+1 \bmod t)}$ when reading an a, where $0 \leqslant i \leqslant t-1$ and $0 < t \leqslant m$.

Without loss of generality, we assume that $\mathcal{D}$ accepts the input if it is in s_0 before reading \$. Thus, $\mathcal{D}$ rejects the input if it is in $s_{(N \bmod t)}$ before reading \$. Let S_a be the set $\{s_{(i2N \bmod t)} \mid i \geqslant 0\}$. Then, $\mathcal{D}$ accepts the input if it is in one of the states in S_a before reading \$. Note that $s_{(N \bmod t)} \notin S_a$.

Let $d = \gcd(t, 2N)$, $t' = \frac{t}{d}$, and S' be the set $\{s_{id} \mid 0 \leqslant i < t'\}$.

Claim 1. $S_a = S'$.

Proof. Since $S_a \subseteq S'$ and $|S'| = t'$, we can obtain $S_a = S'$ if we show $|S_a| \geqslant t'$. We show $|S_a| \geqslant t'$ in three steps:

1. Firstly, we show that each i satisfying ($i2N \equiv 0 \bmod t$) must be a multiple of t': For such an i, there exists a j such that $i2N = jt$. By dividing both sides with $t = dt'$, we get $\frac{i}{t'}\frac{2N}{d} = j$. This implies that i must be a multiple of t' since the left side must be an integer and $\gcd(t', 2N) = 1$.
2. Secondly, we show that there is no i_1 and i_2, i.e. $t' > i_1 > i_2 \geqslant 0$, such that ($i_1 2N \equiv i_2 2N \bmod t$). If so, we have ($i_1 2N - i_2 2N \equiv 0 \bmod t$), and then (($i_1 - i_2)2N \equiv 0 \bmod t$). This implies that $(i_1 - i_2)$ must be a multiple of t'. This is a contradiction.
3. Thus, for each $i \in \{0, \ldots, t'-1\}$, we obtain a different value of ($i2N \bmod t$) and so $|S_a|$ contains at least t' elements, i.e. $|S_a| \geqslant t'$. □

[4] Let $A = (A_{yes}, A_{no})$ be a promise problem exactly solvable by a PFA. Then, we can easily convert this PFA to a DFA by keeping one of probabilistic choice in each transition and removing all the other probabilistic choices. Thus, the DFA also solves A since it always accepts (resp., rejects) the strings in A_{yes} (resp., A_{no}).

Suppose that $\gcd(t, N) = d$. Since d divides $(N \bmod t)$, $s_{(N \bmod t)}$ becomes a member of S' by definition. Due to Claim 1 ($S_a = S'$), $s_{(N \bmod t)}$ also becomes a member of S_a. But we know that $s_{(N \bmod t)} \notin S_a$. Therefore, $\gcd(t, N)$ must be different than $d = \gcd(t, 2N)$.

Since $N = 2^k$ and $2N = 2^{k+1}$, this is only possible if t is divisible by $2^{k+1} = 2N$. □

Since a 2^{k+1}-state DFA solving promise problem A^k can be constructed in a straightforward way, we obtain the following theorem.

Theorem 2. *The minimal DFA solving the promise problem* $A^k = (A^k_{yes}, A^k_{no})$ *has* 2^{k+1} *states.*

4. Concluding remarks

In this paper, we identify a case in which the superiority of quantum computation to classical one cannot be bounded. For this purpose, we use an infinite family of two unary disjoint languages containing the strings of the form $(a^{2n})^*$ and $a^n(a^{2n})^*$, respectively, where n is a power of 2.

What happens if n is not an exact power of 2? For quantum case, we can still solve the same problem with 2 states. On the other hand, for the classical case, the minimum number of states is determined by the biggest factor of the number, which is a power of 2. Let $N = 2^k(2l + 1)$ and $A^N = (A^N_{yes}, A^N_{no})$ be a promise problem such that $A^N_{yes} = \{a^{iN} \mid i$ is a nonnegative *even* integer$\}$ and $A^N_{no} = \{a^{iN} \mid i$ is a positive *odd* integer$\}$, where $k, l \geqslant 0$.

Corollary 1. *The minimal DFA solving promise problem* $A^N = (A^N_{yes}, A^N_{no})$ *has* 2^{k+1} *states.*[5]

Therefore, if N is an odd integer, a DFA only needs 2 states to solve the related promise problems.

Acknowledgements

We thank the anonymous referees for their helpful comments.

References

[1] A. Ambainis, R. Freivalds, 1-way quantum finite automata: strengths, weaknesses and generalizations, in: FOCS'98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, 1998, pp. 332–341.

[2] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf, Quantum lower bounds by polynomials, in: FOCS'98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, 1998, pp. 352–361.

[3] E. Bernstein, U. Vazirani, Quantum complexity theory, SIAM Journal on Computing 26 (5) (1997) 1411–1473.

[4] A. Bertoni, C. Mereghetti, B. Palano, Quantum computing: 1-way quantum automata, in: Z. Ésik, Z. Fülöp (Eds.), Developments in Language Theory, in: Lecture Notes in Computer Science, vol. 2710, 2003, pp. 1–20.

[5] G. Brassard, P. Hoyer, An exact quantum polynomial-time algorithm for Simon's problem, in: ISTCS'97: Proceedings of the Fifth Israel Symposium on the Theory of Computing Systems, 1997, pp. 12–23.

[6] H. Buhrman, R. Cleve, R. de Wolf, C. Zalka, Bounds for small-error and zero-error quantum algorithms, in: FOCS'99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science, 1999, pp. 358–359.

[7] H. Buhrman, R. de Wolf, Quantum zero-error algorithms cannot be composed, Information Processing Letters 87 (2003) 79–84.

[8] M.P. Ciamarra, Quantum reversibility and a new model of quantum automaton, in: FCT'01: Proceedings of the 13th International Symposium on Fundamentals of Computation Theory, 2001, pp. 376–379.

[9] R. Freivalds, K. Iwama, Quantum queries on permutations with a promise, in: CIAA'09: Proceedings of the 14th International Conference on Implementation and Application of Automata, 2009, pp. 208–216.

[10] M. Hirvensalo, Quantum automata with open time evolution, International Journal of Natural Computing Research 1 (1) (2010) 70–85.

[11] H. Klauck, On quantum and probabilistic communication: Las Vegas and one-way protocols, in: STOC'00: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, pp. 644–651.

[12] A. Kondacs, J. Watrous, On the power of quantum finite state automata, in: FOCS'97: Proceedings of the 38th Annual Symposium on Foundations of Computer Science, 1997, pp. 66–75.

[13] C. Moore, J.P. Crutchfield, Quantum automata and quantum grammars, Theoretical Computer Science 237 (1–2) (2000) 275–306.

[14] Y. Murakami, M. Nakanishi, S. Yamashita, K. Watanabe, Quantum versus classical pushdown automata in exact computation, IPSJ Digital Courier 1 (2005) 426–435.

[15] J. Watrous, Quantum computational complexity, in: R.A. Meyers (Ed.), Encyclopedia of Complexity and Systems Science, Springer, 2009, pp. 7174–7201.

[16] A. Yakaryılmaz, R. Freivalds, A.C.C. Say, R. Agadzanyan, Quantum computation with devices whose contents are never read, in: Unconventional Computation, in: Lecture Notes in Computer Science, vol. 6079, 2010, pp. 164–174.

[17] A. Yakaryılmaz, A.C.C. Say, Unbounded-error quantum computation with small space bounds, Information and Computation 209 (6) (2011) 873–892.

[5] The proof can be obtained by using almost the same technique given in Section 3.