RAIRO-Theor. Inf. Appl. 46 (2012) 615–641 Available online at:
DOI: 10.1051/ita/2012018 www.rairo-ita.org

SUPERIORITY OF ONE-WAY AND REALTIME QUANTUM MACHINES*,**

Abuzer Yakaryilmaz¹

Abstract. In automata theory, quantum computation has been widely examined for finite state machines, known as quantum finite automata (QFAs), and less attention has been given to QFAs augmented with counters or stacks. In this paper, we focus on such generalizations of QFAs where the input head operates in one-way or realtime mode, and present some new results regarding their superiority over their classical counterparts. Our first result is about the nondeterministic acceptance mode: Each quantum model architecturally intermediate between realtime finite state automaton and one-way pushdown automaton (oneway finite automaton, realtime and one-way finite automata with onecounter, and realtime pushdown automaton) is superior to its classical counterpart. The second and third results are about bounded error language recognition: for any k > 0, QFAs with k blind counters outperform their deterministic counterparts; and, a one-way QFA with a single head recognizes an infinite family of languages, which can be recognized by one-way probabilistic finite automata with at least two heads. Lastly, we compare the nondeterministic and deterministic acceptance modes for classical finite automata with k blind counter(s), and we show that for any k > 0, the nondeterministic models outperform the deterministic ones.

Mathematics Subject Classification. 68Q05, 68Q10, 68Q12, 68Q45.

Keywords and phrases. Quantum computation, quantum automata, blind counter automata, multihead finite automata, nondeterminism, bounded error.

^{*} A preliminary version of this paper appeared in the Proceedings of Third Workshop on Non-Classical Models for Automata and Applications (2011) 209–224.

^{**} This work was partially supported by $T\ddot{U}B\dot{I}TAK$ with grant 108E142 and FP7 FET-Open project QCS.

 $^{^1}$ University of Latvia, Faculty of Computing, Raina bulv. 19, LV-1586 Rīga, Latvia. <code>abuzer@lu.lv</code>

1. Introduction

Quantum computation is a generalization of classical computation [36, 45]. Therefore, it is interesting to investigate the cases in which quantum computation is superior to classical computation. Unfortunately, due to their restricted definitions, early quantum finite automaton (QFA) models were shown to be less powerful than their classical counterparts for some cases (e.g. [1, 5, 18, 23, 49]). These models do not reflect the full power of quantum computation [37]. In this paper, we use "modern" definitions for the quantum models (e.g. [4, 17, 39, 45]) and then we present some new results about how quantumness adds power to oneway and realtime computational models.² We also give a new result on classical computation.

Our first result is on nondeterministic computation. The superiority of quantum models has been known for realtime finite automata and one-way pushdown automata in nondeterministic acceptance mode [10,25,26,42]. We extend this superiority to every model architecturally intermediate between realtime finite automata and one-way pushdown automata.

Main Result 1. In nondeterministic acceptance mode, quantum one-way automata, quantum realtime finite automata with one-counter, quantum one-way finite automata with one-counter, and quantum realtime pushdown automata are superior to their classical counterparts.

Our second result is on finite automata with blind counter(s). On contrary to standard ones [11], the status of *blind* counters are never tested during the computation. This restriction leads to a dramatic decrease in the power of computation [15]. We show how a quantum trick can be used to test the status of a blind counter algorithmically. That is, we present a realtime quantum automaton with one blind counter recognizing a language that cannot be recognized by any realtime deterministic automaton with k blind counter(s), k > 0.

Main Result 2. For any k > 0 and $\epsilon \in (0, \frac{1}{2})$, the class of languages recognized by realtime deterministic automata with k blind counter(s) is a proper subset of the class of languages recognized by realtime quantum automata with k blind counter(s) with error bound ϵ .

Our third result is on one-way (multi-head) finite automata. Rosenberg [30] presented an infinite family of languages such that for any k > 0, we have some languages which cannot be recognized by any one-way k-head deterministic automaton [12, 20]. Freivalds [12] presented a one-way two-head probabilistic automaton recognizing this whole family with any error bound. We show that one head is sufficient for one-way quantum automata to obtain the same result.

² Here is a list of references presenting some cases in which one-way or realtime quantum models are superior to their classical counterparts: [1–3,7–10,14,16,18,19,21,22,24,26,31–33,41–49].

Main Result 3. For any k > 0, there exists a language which cannot be recognized by any one-way k-head deterministic automaton, but it is recognized by a one-way (one-head) quantum automaton.

Our last result is on classical computation. We show that nondeterministic automata are superior to deterministic ones if they both have the same number of blind counter(s).

Main Result 4. For any k > 0, the class of languages recognized by realtime deterministic automata with k blind counter(s) is a proper subset of the class of languages recognized by realtime nondeterministic automata with k blind counter(s).

We refer the reader to [38] for a preliminary version of this paper. We give all definitions and complete proofs in the current version. The definitions of models and other basics are given in Section 2 and our main results are presented in Section 3. We also have an appendix, in which some technical details of quantum models are given for completeness of the paper.

2. Background

In this section, we provide some background information.

2.1. Basic notation

Throughout the paper, we use the following notations: Σ , not containing \mathfrak{c} and \mathfrak{s} (the left and the right end-markers respectively), denotes the input alphabet; $\tilde{\Sigma} = \Sigma \cup \{\mathfrak{c}, \mathfrak{s}\}$; Q is the set of (internal) states; $Q_a \subseteq Q$ is the set of accepting states, q_0 is the initial state (unless otherwise specified); $f_{\mathcal{M}}(w)$ is the accepting probability (or value) of machine \mathcal{M} on w. For any $w \in \Sigma^*$, w_i is the i^{th} symbol of w; |w| is the length of w; $|w|_{\sigma}$ is the number of occurrences of σ in w. The list of abbreviations used for the models in the paper is given below:

- the prefixes "1" and "rt" stand for *one-way* (the head(s) is (are) allowed to be stationary or to move to the right) and *realtime* (the head is allowed to move only to the right at every step) input head(s), respectively;
- the letters "D", "N", "P", and "Q" used after "1" or "rt" stand for deterministic, nondeterministic, probabilistic, and quantum variants, respectively;
- the abbreviations "FA", "kFA", "k'BCA", and "PDA" stand for finite automaton, finite automaton with k input heads, automaton with k' blind counter(s), and pushdown automaton, respectively, where k > 1, k' > 0.

2.2. Language recognition criteria

The language recognition criteria used in the paper are given below:

- a language $L \subseteq \Sigma^*$ is said to be recognized by \mathcal{M} with error bound $\epsilon \in (0, \frac{1}{2})$ if (i) $f_{\mathcal{M}}(w) \geq 1 \epsilon$ for $w \in L$, and (ii) $f_{\mathcal{M}}(w) \leq \epsilon$ for $w \notin L$. More generally, a language $L \subseteq \Sigma^*$ is said to be recognized by \mathcal{M} with bounded error if there is an $\epsilon \in (0, \frac{1}{2})$ satisfying the above condition;
- a language $L \subseteq \Sigma^*$ is said to be recognized by \mathcal{M} with negative one-sided error bound $\epsilon \in (0,1)$ if (i) $f_{\mathcal{M}}(w) = 1$ for $w \in L$ and (ii) $f_{\mathcal{M}}(w) \leq \epsilon$ for $w \notin L$;
- a language $L \subseteq \Sigma^*$ is said to be recognized by \mathcal{M} in nondeterministic mode [42] if (i) $f_{\mathcal{M}}(w) > 0$ for $w \in L$ and (ii) $f_{\mathcal{M}}(w) = 0$ for $w \notin L$.

2.3. A PROLOGUE FOR QUANTUM MACHINES

We refer the reader to [27] for a complete reference on quantum computation. A finite-dimensional quantum system is an n-dimensional Hilbert space, a complex vector space with inner product (\mathcal{H}_n) , where n > 0. The set $\mathcal{B}_n = \{|q_i\rangle \mid 1 \le i \le n\}$ is an orthonormal basis for \mathcal{H}_n , where the i^{th} entry of $|q_i\rangle$ is 1 and the remaining entries are zeros. Any quantum state of the system is described by its state vector, say $|\psi\rangle$, that is a linear combination of basis states³ $|\psi\rangle = \alpha_1|q_1\rangle + \cdots + \alpha_n|q_n\rangle$, where the number α_i is the amplitude of $|q_i\rangle$, whose modulus squared $(|\alpha_i|^2)$ gives the probability of being in state q_i , and $\sum_i |\alpha_i|^2 = 1$ $(1 \le i \le n)$. When $|\psi\rangle$ contains more than one basis state with nonzero amplitude, the system is said to be in a superposition (of the basis states).

The most general operator applied to a quantum system is a superoperator, which generalizes unitary and stochastic operators. Formally, a superoperator \mathcal{E} is composed of a finite number of operation elements, $\mathcal{E} = \{E_l \mid 1 \leq l \leq k \in \mathbb{Z}^+\}$ satisfying

$$\sum_{l=1}^{k} E_l^{\dagger} E_l = I, \tag{2.1}$$

i.e., the columns of the matrix obtained by concatenating each operation element one under the other forms an orthonormal set. If we apply $\mathcal E$ to a quantum system in $|\psi\rangle$, we obtain a mixture of quantum states, $\{(p_l,|\psi_l\rangle)\mid 1\leq l\leq k\}$, where $|\psi_l\rangle=\frac{|\check{\psi}_l\rangle}{\sqrt{p_l}}$, $|\check{\psi}_l\rangle=E_l|\psi\rangle$, and $p_l=\langle\check{\psi}_l|\check{\psi}_l\rangle$ is the probability of observing the system being in $|\psi_l\rangle$. Note that $|\psi_i\rangle$ is obtained by normalizing $|\check{\psi}_i\rangle$, i.e. any "observed" quantum state is normalized. A convenient mathematical tool for representing the mixture of quantum states is density matrix. The density matrix representation of $\{(p_l,|\psi_l\rangle)\mid 1\leq l\leq k\}$ is

$$\rho = \sum_{i=1}^{l} p_l |\psi_l\rangle \langle \psi_l|.$$

³We fixed it as \mathcal{B}_n . However, note that, one can also select any other orthonormal basis.

	$c_1 \cdots c_{ \mathcal{C}_{\mathcal{M}}^w }$
c_1	
:	T
$c_{ \mathcal{C}_{\mathcal{M}}^{w} }$	E_1
c_1	
$c \mid \mathcal{C}_{M}^{w} \mid$	E_2
:	:
c ₁	
$\frac{\cdot}{c} \mathcal{C}_{\mathcal{M}}^{w} $	\boldsymbol{E}_k

FIGURE 1. Matrix $\mathsf{E}_{\mathcal{M}}^w$.

A compact way of representing the evolution of a system described by density matrix ρ , on which superoperator \mathcal{E} is applied, is

$$\rho' = \mathcal{E}(\rho) = \sum_{l=1}^{k} E_l \rho E_l^{\dagger}.$$

A projective measurement P is composed of a finite number of projectors, zero-one diagonal matrices, i.e. $P = \{P_l \mid \sum_{l=1}^k P_l = I, 1 \leq l \leq k \in \mathbb{Z}^+\}$, where indices represent the measurement outcomes. If we apply P to a system in ρ or $|\psi\rangle$, the outcome "l" is obtained with probability

$$p_l = \operatorname{Tr}(P_l \rho) \text{ or } p_l = \langle \psi | P_l | \psi \rangle,$$

respectively.

The computational space of a machine on an input string is its configuration sets, *i.e.* a configuration is a complete information of the machine at any computational step. The computation starts in the *initial* configuration and continues with respect to its transition function, which are the set of rules governing the behaviour of the machine locally. Depending on the model type, the evolution of the computational space differs. Note that the evolution of the computational space defines some restrictions on the transition function.

Let \mathcal{M} be a quantum machine. We represent the set of the configurations of \mathcal{M} on a given input $w \in \Sigma^*$ by $\mathcal{C}_{\mathcal{M}}^w$ that forms quantum system $\mathcal{H}_{\mathcal{M}}^w$ with a basis set $\mathcal{B}_{\mathcal{M}}^w$, i.e. $\mathcal{B}_{\mathcal{M}}^w = \{|c_i\rangle \mid c_i \in \mathcal{C}_{\mathcal{M}}^w\}$, where $|c_i\rangle$ is a vector composed by zeros except i^{th} entry which is 1. In order to be a well-formed machine, the evolution of $\mathcal{H}_{\mathcal{M}}^w$ must be governed by a quantum operator, specifically by a superoperator. We give the matrix form of this superoperator, called matrix $\mathsf{E}_{\mathcal{M}}^w$, on $\mathcal{H}_{\mathcal{M}}^w$ in Figure 1. Note that the columns of $\mathsf{E}_{\mathcal{M}}^w$ form an orthonormal set.

The entries of $\mathsf{E}_{\mathcal{M}}^w$ are filled by the transition function of the machine. Due to the condition given in equation (2.1), the transition function must obey some constraints, known as *local conditions for the machine well-formedness* [6, 35, 45].

In this paper, we define quantum models with some simplifications (described later). Thus, the transitions of each model can be grouped for each symbol in $\tilde{\Sigma}$ such that each group forms a superoperator which is defined on the set of internal states if the machine is well-formed. We give the general definitions of quantum models and the technical explanations of such simplifications in Appendix A.

Note that it is an open problem for most quantum models whether such restrictions lead to decrease in computational power of the models [4]. The same restrictions, however, do not change the computational power of classical machines. Since the superoperators generalize the stochastic operators, each simplified quantum model is at least powerful as its classical counterpart [33,45].

2.4. Definitions of models

For all models (except GFAs described below), the input $w \in \Sigma^*$ is placed on a read-only two-way infinite tape as $\tilde{w} = \mathfrak{c}w$ \$ between the cells indexed by 1 to $|\tilde{w}|$. At the beginning, the head(s) is (are) initially placed on the cell indexed by 1 and the value(s) of the counter(s) is (are) set to zero(s).

2.4.1. Realtime finite automata

A realtime probabilistic finite automaton [29] (rtPFA) is formally a 5-tuple

$$\mathcal{P} = (Q, \Sigma, \{A_{\sigma} \mid \sigma \in \tilde{\Sigma}\}, q_0, Q_a),$$

where the $A_{\sigma \in \tilde{\Sigma}}$'s are $|Q| \times |Q|$ -dimensional (left) stochastic matrices. The computation of a rtPFA can be traced by a stochastic state vector, say v, whose i^{th} entry, denoted v[i], corresponds to state q_i . For a given input string $w \in \Sigma^*$,

$$v_i = A_{\tilde{w}_i} v_{i-1},$$

where $1 \leq i \leq |\tilde{w}|$ and v_0 is the initial state vector, whose first entry is 1. The accepting probability of w by \mathcal{P} is defined as

$$f_{\mathcal{P}}(w) = \sum_{q_i \in Q_a} v_{|\tilde{w}|}[i].$$

We call a rtPFA realtime deterministic finite automaton (rtDFA) if its stochastic components contain only zeros and ones. Any rtPFA defined with nondeterministic aceptance mode is also called *realtime* nondeterministic finite automaton (rtNFA).

A generalized finite automaton [34] (GFA), a generalization of rtPFAs, is formally a 5-tuple

$$\mathcal{G} = (Q, \Sigma, \{A_{\sigma} \mid \sigma \in \Sigma\}, v_0, f),$$

where (i) $A_{\sigma \in \Sigma}$'s are $|Q| \times |Q|$ -dimensional real valued transition matrices, and, (ii) v_0 and f are real valued *initial* (column) and *final* (row) vectors, respectively. For an input string, $w \in \Sigma^*$, the acceptance value of w associated by \mathcal{G} is defined as

$$f_{\mathcal{G}}(w) = f A_{w_{|w|}} \cdots A_{w_1} v_0.$$

We call a GFA *Turakainen finite automaton* (TuFA) if all of its components are restricted to be rational numbers.

A realtime quantum finite automaton [17,45] (rtQFA) is a 5-tuple

$$\mathcal{M} = (Q, \Sigma, \{\mathcal{E}_{\sigma} \mid \sigma \in \tilde{\Sigma}\}, q_0, Q_a),$$

where \mathcal{E}_{σ} is a superoperator composed by a collection of state transition matrices $\{E_{\sigma,1},\ldots,E_{\sigma,k}\}$ for some $k\in\mathbb{Z}^+$ satisfying

$$\sum_{i=1}^{k} E_{\sigma,i}^{\dagger} E_{\sigma,i} = I.$$

Additionally, we define the operator

$$P = \left\{ P_a, P_r \mid P_a = \sum_{q \in Q_a} |q\rangle\langle q| \text{ and } P_r = I - P_a \right\}$$

representing the single measurement of the state type at the end of the computation. For a given input string $w \in \Sigma^*$, the overall state of the machine can be traced by

$$\rho_{j} = \mathcal{E}_{\tilde{w}_{j}}(\rho_{j-1}) = \sum_{i=1}^{k} E_{\tilde{w}_{j},i} \rho_{j-1} E_{\tilde{w}_{j},i}^{\dagger},$$

where $1 \leq j \leq |\tilde{w}|$ and $\rho_0 = |q_1\rangle\langle q_1|$ is the initial density matrix. The accepting probability of w by \mathcal{M} is defined as

$$f_{\mathcal{M}}(w) = \operatorname{Tr}(P_a \rho_{|\tilde{w}|}).$$

NQAL is the class of languages recognized by rtQFA in nondeterministic mode [42], i.e., $L \in \text{NQAL}$ if and only if there exits a rtQFA \mathcal{M} such that $f_{\mathcal{M}}(w) > 0$ for all $w \in L$ and $f_{\mathcal{M}}(w) = 0$ for all $w \notin L$.

2.4.2. Realtime automata with blind counters

Let $X \in \{D,N,P,Q\}$ and k and m be nonnegative integers. A rtXkBCA is a rtXFA augmented with k blind counter(s), on which some fixed amount of increment and decrement operations can be made [15]. The term blind refers to a setup such that the automaton never checks the content of the counter(s) and the input is accepted only if the value(s) of the counter(s) is (are) zero(s) at the end of the computation. For simplicity, we assume that the updates on the counters are determined by the states to be entered during computation. We denote this relationship using function

$$D_c: Q \to \{-m, \ldots, m\}^k.$$

That is, when the automaton enters to state $q \in Q$, then $D_c(q)[i]$ is added to the *i*th counter value⁴. So, each rtXkBCA is defined exactly the same as rtXFA. Additionally, the value(s) of the counter(s) is (are) updated after each transition.

For a given string $w \in \Sigma^*$, the configurations of a given rtXkBCA are the pairs of $(q, v) \in Q \times \mathbb{Z}^k$ and $(q_0, \{0\}^k)$ is the initial one, where v is a k-tuple representing the value(s) of the counter(s). In quantum case, a measurement is done on the configuration sets at the end of the computation to check whether all the value(s) of the counter(s) is (are) zero(s). If not, the input is rejected immediately. Otherwise, the decision is given by the standard measurement implemented on the state set.

2.4.3. Multihead finite automata

Let $X \in \{D,N,P,Q\}$ and k > 1. A 1XkFA is a generalization of rtXFA such that (i) it has more than one input head and (ii) each input head is not required to move to the right after each transition [30]. For simplicity, we assume that the movement of the input head(s) is determined by the states to be entered during computation. We denote this relationship using function

$$D_i: Q \to \{``\downarrow", ``\to"\}^k.$$

That is, when the automaton enters state $q \in Q$, then the *i*th head stays on the same square if $D_i(q)[i] = "\downarrow"$ or moves one square to the right if $D_i(q)[i] = "\to"$. Thus, the transition operators over the state set are defined for each element in $\tilde{\Sigma}^k$, *i.e.*, the next state, say q, is determined by the current state and the symbols under the heads and then the positions of the heads are updated by $D_i(q)$. It is always assumed that, none of the heads leaves the right-end marker (\$).

Moreover, we divide Q into three disjoint subsets, Q_a , Q_r , and Q_c , where Q_r and Q_c are the set of rejecting and continuing states. The computation is terminated and the input is accepted (resp., rejected) if the machine enters a state in Q_a (resp., Q_r) during the computation. In the quantum case, a measurement is implemented to make a similar decision. We omit the details here. But, the situation for 1QFAs, given in the next section, can be generalized to 1QkFAs.

2.4.4. One-way quantum finite automata

Since machines of this model have a single head, we denote the simplification property by using arrow symbols such that if the machine enters state \overrightarrow{q} or $\downarrow q$, then the input head is moved one square to the right or stays in the same square, respectively.

A simple one-way quantum finite automaton (1QFA) [45] is a 6-tuple

$$\mathcal{M} = (Q, \Sigma, \{\mathcal{E}_{\sigma} \mid \sigma \in \tilde{\Sigma}\}, q_0, Q_a, Q_r).$$

⁴It is a well-known fact that (e.g. see [47]) for any classical or quantum counter automaton having the capability of updating its counter(s) with increments from the set $\{-m, \ldots, m\}$, there exists an equivalent counter automaton updating its counter(s) with increments from the set $\{-1,0,1\}$ for any m > 1.

For a given string $w \in \Sigma^*$, the configurations of \mathcal{M} are the pairs of $(q, x) \in Q \times \{1, \ldots, |\tilde{w}|\}$ and $(q_0, 1)$ is the initial one, where x stands for the head position. Let $\Delta = \{a, r, c\}$. We partitioned the configuration set into three disjoint subsets as $\mathcal{C}^w_{\mathcal{M}, a}$ (accepting), $\mathcal{C}^w_{\mathcal{M}, r}$ (rejecting), and $\mathcal{C}^w_{\mathcal{M}, c}$ (continuing), *i.e.*

$$\mathcal{C}^{w}_{\mathcal{M},\tau} = \{ |c\rangle \mid c \in Q_{\tau} \times \{1, \dots, |\tilde{w}|\} \},$$

where $\tau \in \Delta$. After each step of the transition, the projective measurement

$$P = \left\{ P_{\tau} \mid P_{\tau} = \sum_{c \in \mathcal{C}_{\mathcal{M}, \tau}^{w}} |c\rangle\langle c|, \tau \in \Delta \right\},\,$$

is applied to the configuration set and (i) the computation is continued if "c" is observed, (ii) the computation is terminated, otherwise, and the input is accepted (resp., rejected) if "a" (resp., "r") is observed.

2.5. N-WAY QUANTUM FOURIER TRANSFORM

In our algorithms, we use a special kind of quantum transformation, N-way QFT (quantum Fourier transform) [18, 40, 47]. Let N > 1 be an integer and E be an operation element implementing N-way QFT. The N-way QFT is a set of transition rules defined from N source states s_1, \ldots, s_N to N target states t_1, \ldots, t_N as follows:

$$E|s_j\rangle = \frac{1}{\sqrt{N}} \sum_{l=1}^N e^{\frac{2\pi i}{N}jl} |t_l\rangle, \quad 1 \le j \le N,$$

where t_N is called the distinguished target state. Let ω denote $e^{\frac{2\pi i}{N}}$. Then, the matrix form of the transformation can be given as follows:

Note that the last row summation is $\frac{N}{\sqrt{N}}$ and the other row summations are equal to zeros, *i.e.*, $\sum_{l=1}^{N} (\omega^{j})^{l} = 0$ for $1 \leq j < N$ and $\sum_{l=1}^{N} (\omega^{j})^{l} = \frac{N}{\sqrt{N}}$ for j = N. We use this fact in our algorithms.

3. Main results

We classify our results under four titles. We give our quantum results in the first three subsections, and then we present our result regarding classical computation in the last subsection.

3.1. Superiority of quantum nondeterminism

It is already known that the class of languages recognized by rtPFAs (resp., 1PPDAs) is a proper subset of the class of languages recognized by rtQFAs (resp., 1QPDAs) in nondeterministic mode [10, 25, 26, 42]. We give a stronger version of these results by using the noncontextfree language

$${\tt IJK} = \{a^i b^j c^k \mid i \neq j, i \neq k, j \neq k, 0 \leq i, j, k\}.$$

Fact 3.1 (Lem. 3.1 and Thm. 3.1 of [42]). $L \subseteq \Sigma^* \in \mathsf{NQAL}$ if and only if L is defined by a GFA, say \mathcal{G} , as follows: (i) $f_{\mathcal{G}}(w) > 0$ for $w \in L$ and (ii) $f_{\mathcal{G}}(w) = 0$ for $w \notin L$.

Fact 3.2 (p. 147 in [28]). By tensoring two GFAs, $\mathcal{G}_1 = (Q_1, \Sigma, \{A_{\sigma} \mid \sigma \in \Sigma\}, v_0, f)$ and $\mathcal{G}_2 = (Q_2, \Sigma, \{B_{\sigma} \mid \sigma \in \Sigma\}, u_0, g)$, we obtain a new GFA \mathcal{G}' $(\mathcal{G}_1 \otimes \mathcal{G}_2)$, specified as

$$\mathcal{G}' = (Q_1 \times Q_2, \Sigma, \{A_{\sigma} \otimes B_{\sigma} \mid \sigma \in \Sigma\}, v_0 \otimes u_0, f \otimes g),$$

such that for any $w \in \Sigma$, $f_{\mathcal{G}'}(w) = f_{\mathcal{G}_1}(w)f_{\mathcal{G}_2}(w)$.

Theorem 3.3. In nondeterministic acceptance mode, the class of the languages recognized by classical (probabilistic) machines is a proper subset of the class of the languages recognized by their quantum counterparts for any model architecturally intermediate between realtime finite automaton and one-way (one-head) pushdown automaton, i.e., one-way finite automaton, realtime and one-way finite automata with one-counter, and realtime pushdown automaton.

Proof. We fix ourselves to nondeterministic acceptance mode. Since IJK cannot be recognized by any 1PPDAs, it is sufficient to show that IJK can be recognized by rtQFA, *i.e.*, IJK is in NQAL. That is, if IJK cannot be recognized by any 1PPDAs, then it cannot be recognized by any restricted model of 1PPDA. On the other hand, if IJK can be recognized by rtQFA, then it can be recognized by any generalized model of rtQFA.

Let $\Sigma = \{a, b, c\}$ be the input alphabet. We design a TuFA \mathcal{T}_{a-b} to calculate the value of $(|w|_a - |w|_b)$ as its accepting value for any $w \in \Sigma^*$, *i.e.*,

$$\mathcal{T}_{a-b} = (Q, \Sigma, \{A_{\sigma} \mid \sigma \in \Sigma\}, v_0, f),$$

where $Q = \{q_1, q_2\}, v_0 = (0 \ 1)^T, f = (1 \ 0), and$

$$A_a = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad A_b = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}, \quad A_c = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

At the beginning, the weights of q_1 and q_2 are 0 and 1, respectively. Whenever an a (resp., a b) is read, the weight of q_1 (resp., q_2) is increased (resp., decreased) by 1. At the end, the weight of q_1 is assigned as the accepting value. That is, $f_{\mathcal{T}_{a-b}}(w) = |w|_a - |w|_b$.

Similarly, we can design two TuFAs \mathcal{T}_{a-c} and \mathcal{T}_{b-c} to calculate the values of $(|w|_a - |w|_c)$ and $(|w|_b - |w|_c)$ as their accepting values, respectively, for any $w \in \Sigma^*$.

Moreover, we can design a TuFA $\mathcal{T}_{a^+b^+c^+}$ to assign 1 as the accepting value for the strings of the form $a^+b^+c^+$ and 0 otherwise:

$$\mathcal{T}_{a^+b^+c^+} = (Q, \Sigma, \{A_\sigma \mid \sigma \in \Sigma\}, v_0, f),$$

where $Q = \{q_1, q_2, q_3, q_4\}, v_0 = (1 \ 0 \ 0 \ 0)^T, f = (0 \ 0 \ 0 \ 1), and$

Now, based on Fact 3.2, we can obtain a TuFA \mathcal{T}_{IJK} for IJK as

$$\mathcal{T}_{\text{IJK}} = \mathcal{T}_{a^+b^+c^+} \otimes (\mathcal{T}_{a-b} \otimes \mathcal{T}_{a-b}) \otimes (\mathcal{T}_{a-c} \otimes \mathcal{T}_{a-c}) \otimes (\mathcal{T}_{b-c} \otimes \mathcal{T}_{b-c}),$$

which calculates the value of

$$(|w|_a - |w|_b)^2 (|w|_a - |w|_c)^2 (|w|_b - |w|_c)^2$$

for the strings of the form $a^+b^+c^+$ and returns 0 otherwise. So, $f_{\mathcal{I}_{\text{IJK}}}(w)$ is a positive integer if w is a member of IJK and it is zero if w is not a member of IJK. That is, IJK \in NQAL due to Fact 3.1.

3.2. Quantumness allows intermediate zero-test on blind counters

Let UPAL be the language $\{a^nb^n \mid n \geq 0\}$ and UPAL* be its Kleene closure, *i.e.*

$$\mathtt{UPAL}^* = \{\varepsilon\} \cup \{a^{n_1}b^{n_1} \cdots a^{n_k}b^{n_k} \mid n_i > 0 \ (1 \le i \le k), \ k \ge 1\}.$$

Although UPAL* can be recognized by rtDFAs with a single counter, this is not the case for 1DFAs with blind counters whatever the number of the counters is [15], i.e., UPAL* cannot recognized by any 1DkBCA, k > 0.

In the quantum case, we show that a rtQFA with a single blind counter can recognize the same language with any desired error bound. The main idea is that a rtQ1BCA can create a superposition of counters with different updating strategy and then "cleverly" interfere them in the middle of computation to test whether their values are zeros or not. Note that this test is achieved algorithmically, and the "blindness" constraint is not violated.

Theorem 3.4. For any k > 0 and $\epsilon \in (0, \frac{1}{2})$, the class of languages recognized by rtDkBCAs is a proper subset of the class of the languages recognized by rtQkBCAs with error bound ϵ .

Proof. We give a simple rtQ1BCA algorithm recognizing UPAL* with negative one-sided error bound ϵ for any $\epsilon \in (0, \frac{1}{2})$. Then, the proof follows from the fact that UPAL* cannot be recognized by any rtDkBCA (k > 0).

Let $N \geq 2$ and $\mathcal{M}_{\mathtt{UPAL}^*,N} = (Q, \Sigma, \{\mathcal{E}_{\sigma} \mid \sigma \in \tilde{\Sigma}\}, q_0, Q_a)$ be a rtQ1BCA, where $Q = \{q_0, a_0\} \cup \{q_j \cup q'_j \cup p_j \mid 1 \leq j \leq N\} \cup \{r_j \mid 0 \leq j \leq N-1\}$, and $Q_a = \{a_0, p_N\}$. The details of the transition matrices are given in Figure 2. (The missing parts can easily be completed.) Note that some superoperators have more than one operation elements so that they can implement some *irreversible* tasks. One explanatory example is that $\mathcal{M}_{\mathtt{UPAL}^*,N}$ enters state q'_j after reading a b when it is originally in q_j or q'_j , and so we use two operation elements $E_{b,1}$ and $E_{b,2}$, where $1 \leq j \leq N$ (see Eqs. (3.3) and (3.4)).

Let $w \in \Sigma^*$ be an input string. We begin with two trivial cases:

- if $w = \varepsilon$, then it is accepted with probability 1 (Eq. (3.7));
- if w starts with a b, (i) $\mathcal{M}_{\text{UPAL}^*,N}$ enters reject-path₀ (Eq. (3.8)), (ii) the computation stays there until the end of the computation (Eq. (3.10)), and then (iii) w is rejected with probability 1 (Eq. (3.10)).

We continue with the inputs started by an a, i.e. $w = (a^+b^+)^*(a^+)$ or $w = (a^+b^+)^+$. That is, the input is formed by

- 1. $t \ge 0$ block(s) of a^+b^+ and then a block of a^+ or
- 2. t > 0 block(s) of a^+b^+ .

The main idea (for the remaining part) is that we run a procedure, called *block-test*, on each block. The input is accepted exactly if each *block-test* succeeds. Otherwise, the input is rejected exactly if one *block-test* fails and it is rejected with high probability if at least two *block-test*s fail. The details of *block-test* are given below.

A block-test is started with reading an a after $\mathfrak c$ or a b. The latter case will be described soon. In the former case, the computation splits into N different paths, i.e. path_j $(1 \le j \le N)$, with equal amplitude, and the counter value is increased by j (Eq. (3.1)). The counter value is increased by j in path_j as long as a's are being read (Eq. (3.2)). If the next symbol is the right end-marker (\$), then the current block-test has failed and so the input is rejected with probability 1 (Eq. (3.9)). Otherwise – the next symbol is a b – path_j changes its update strategy (Eq. (3.3)) such that the counter value is decreased by j in path_j as long as reading b's (Eqs. (3.3) and (3.4)). Now the next symbol can be either an a or the \$. If it is \$, all paths make the QFT given in equation (3.6). If it is an a, they make the QFT given in equation (3.5), in which the computation again splits into N paths from the distinguished target, i.e. a new block-test is started (the related transitions are $\left(\frac{1}{\sqrt{N}}\sum_{j=1}^N |q_j\rangle\right)$ in Eq. (3.5)).

Since a block formed by a^+ is rejected certainly, we consider the blocks of the form a^+b^+ . Assume that $w=a^{m_1}b^{n_1}\cdots a^{m_t}b^{n_t}$, where t>0.

If $w \in \text{UPAL}^*$: Before making the first QFT, path_j is in $\frac{1}{\sqrt{N}}|q'_j, j(m_1 - n_1)\rangle$, which is $\frac{1}{\sqrt{N}}|q'_j, 0\rangle$ since $m_1 = n_1$. Therefore, all the configurations are interfered with each other during the QFT and only the distinguished target survives

$$\begin{cases} \mathcal{E}_{\mathbb{C}} = \{E_{\mathbb{C}}\} \\ \mathcal{E}_{a} = \{E_{a,1}, E_{a,2}, E_{a,3}, E_{a,r}\} \\ \mathcal{E}_{b} = \{E_{b,1}, E_{b,2}, E_{b,r}\} \\ \mathcal{E}_{\$} = \{E_{\$,1}, E_{\$,r}\} \end{cases} \begin{vmatrix} D_{c}(q_{0}) = 0, & D_{c}(a_{0}) = 0 \\ D_{c}(q_{j}) = j, & D_{c}(q'_{j}) = -j, & 1 \leq j \leq N \\ D_{c}(p_{j}) = 0, & 1 \leq j \leq N \\ D_{c}(r_{j}) = 0, & 0 \leq j < N \end{cases}$$

Initial

$$E_{\mathfrak{C}}|q_0\rangle = |q_0\rangle.$$

FIRST-SPLIT

$$E_{a,1}|q_0\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |q_i\rangle. \tag{3.1}$$

UPDATE

 $\overline{\mathsf{path}_i} \ (1 \le j \le N)$:

$$E_{a,2}|q_j\rangle = |q_j\rangle \tag{3.2}$$

$$E_{b,1}|q_i\rangle = |q_i'\rangle \tag{3.3}$$

$$E_{b,2}|q_i'\rangle = |q_i'\rangle. \tag{3.4}$$

 $\frac{\text{QFT-SPLIT}}{\mathsf{path}_i \ (1 \le j \le N)}:$

$$E_{a,3}|q_j'\rangle = \frac{1}{\sqrt{N}} \left(\sum_{l=1}^{N-1} e^{\frac{2\pi i}{N}jl} |r_l\rangle \right) + \underbrace{\frac{1}{\sqrt{N}} e^{2\pi i j} \left(\frac{1}{\sqrt{N}} \sum_{j=1}^{N} |q_j\rangle \right)}_{=\frac{1}{N} \sum_{l=1}^{N} |q_j\rangle}$$
(3.5)

 $\frac{\text{LAST-QFT}}{\mathsf{path}_i \ (1 \le j \le N)}$:

$$E_{\$,1}|q_j'\rangle = \frac{1}{\sqrt{N}} \sum_{l=1}^N e^{\frac{2\pi i}{N}jl} |p_l\rangle.$$
 (3.6)

ACCEPT

$$E_{\$,1}|q_0\rangle = |a_0\rangle. \tag{3.7}$$

Reject

$$E_{b,1}|q_0\rangle = |r_0\rangle \tag{3.8}$$

 $\mathsf{path}_i \ (1 \le j \le N)$:

$$E_{\$,1}|q_i\rangle = |q_i\rangle \tag{3.9}$$

reject-path_j $(0 \le j < N)$:

$$E_{\sigma,r}|r_j\rangle = |r_j\rangle, \quad \sigma \in \{a, b, \$\}.$$
 (3.10)

FIGURE 2. The details of the transition function of $\mathcal{M}_{\text{UPAL}^*,N}$.

(with probability 1). Such a QFT is called *successful*. That is, (i) if the QFT happens on an a, a new *block-test* is started with probability 1 (Eq. (3.5)) or (ii) the input is accepted with probability 1 otherwise (Eq. (3.6)). We are done with the case t=1. In case of t>1, each *block-test* starts with a fresh counter. Hence, the input is accepted exactly after the last QFT.

If $w \notin \text{UPAL}^*$: There exists a minimum t' $(1 \le t' \le t)$ such that $m_{t'} \ne n_{t'}$. We already know form the above that the block-test for $a^{m_{t'}}b^{m_{t'}}$ starts with a fresh counter. So, before making the QFT, path_j is in $\frac{1}{\sqrt{N}}|q_j', j(m_{t'} - n_{t'})\rangle$ and $j_1(m_{t'} - n_{t'}) = j_2(m_{t'} - n_{t'})$ if and only if $j_1 = j_2$, where $1 \le j_1, j_2 \le N$. Therefore, no configuration interferes with the others during the QFT: (i) N different configurations whose state components are distinguished targets are obtained, each of which with probability $\frac{1}{N^2}$, (the overall probability is $\frac{1}{N}$) and (ii) $N^2 - N$ different configurations whose state components are other targets are obtained (the overall probability is $1 - \frac{1}{N}$). If the QFT happens on \$, the input is rejected exactly, since none of the counter values is zero. Otherwise, in the configurations whose state components are distinguished targets, N different new block-tests for $a^{m_{t'+1}}b^{m_{t'+1}}$ are started, each of which has a different counter value, and, in the configurations whose state components are other targets, the computation enters some reject-path (Eq. (3.5)), in which the computation goes on until the end and then the input is rejected exactly (Eq. (3.10)), where $1 \le l < N$.

It is obvious that having a single unbalanced block does not give a bound on the error since the value of each counter in the superposition is nonzero. On the other hand, each unbalanced block leads to reduce the overall probability of surviving block-tests by $\frac{1}{N}$. Therefore, we consider the case that there is one more unbalanced block. Without loss of generality, we assume that it is the last block $(m_t \neq n_t)$. Let $c' = (m_{t'} - n_{t'})$, $c = (m_t - n_t)$, block-test_{j'} be the (j')th block-test having a counter value of j'c' at the beginning, and $\operatorname{path}_{j',j}$ be a path created in block-test_{j'}, where $1 \leq j, j' \leq N$. So, $\operatorname{path}_{j',j}$ is in $\frac{1}{N\sqrt{N}}|q'_j,j'c'+jc\rangle$ before the QFT. For a fixed j', the value of j'c'+jc can be zero for at most one j $(1 \leq j \leq N)$. If c=-c'=1, then we have a j for each j'. Thus, $\operatorname{path}_{j',j}$ ends in a state having $|p_N,0\rangle$ with amplitude with $\frac{1}{N^2}$. The part of state, in which the input is accepted, is $\frac{1}{N}|p_N,0\rangle$ – after (constructive) interference. That is, the input is accepted with probability $\frac{1}{N^2}$ and it is the upper bound on the error. By selecting suitable N, we obtain any desired error bound ϵ .

We suspect that randomization cannot help to recognize UPAL* in the bounded error case.

Conjecture 3.5. UPAL* cannot be recognized by any 1PkBCA with bounded error, for any k > 0.

3.3. A 1QFA ALGORITHM FOR AN INFINITE FAMILY OF LANGUAGES

We begin with the definitions of two languages:

$$UPAL(t) = \{a^{n_1}b \cdots ba^{n_t}ba^{n_t}b \cdots ba^{n_1} \mid n_i \ge 0, 1 \le i \le t\}$$

and

$$UPAL'(t) = \{a^{n_1}b \cdots ba^{n_t}ba^{n_t}b \cdots ba^{n_1} \mid n_i > 0, 1 \le i \le t\}.$$

Note that there is always an a before and after each b in the latter language but this is not required in the former language. It was shown in [12, 20] that for any k, there exists a t>0 such that language $\mathtt{UPAL}'(\mathtt{t})$ cannot be recognized by any $\mathtt{1D}k\mathtt{FA}$. The same argument also holds for $\mathtt{UPAL}(\mathtt{t})$ since any $\mathtt{1D}k\mathtt{FA}$ recognizing $\mathtt{UPAL}(\mathtt{t})$ can be converted to a $\mathtt{1D}k\mathtt{FA}$ recognizing $\mathtt{UPAL}'(\mathtt{t})$ in a straightforward way.

In [12], Freivalds gave a 1P2FA algorithm for these languages. In this section, we show that a 1QFA does not need any other head to recognize UPAL(t). For pedagogical reasons, we consider firstly the case of t = 1 and then the case of t > 0.

Lemma 3.6. Language UPAL(1) can be recognized by a simple 1QFA with negative one-sided error bound $\frac{1}{N}$, where N > 2.

Proof. Let $\mathcal{M}_{\mathtt{UPAL}(1),N} = (Q, \Sigma, \{\mathcal{E}_{\sigma} \mid \sigma \in \tilde{\Sigma}\}, q_0, Q_a, Q_r)$ be the simple 1QFA recognizing UPAL(1) with negative one-sided error bound $\frac{1}{N}$, where each \mathcal{E}_{σ} contains exactly one unitary operation element⁵, E_{σ} , and $Q = Q_c \cup Q_a \cup Q_r$, i.e.,

$$\begin{split} Q_c &= \{\overrightarrow{q_0}\} \cup \{\overrightarrow{q_j} \cup \overrightarrow{q_j'} \mid 1 \leq j \leq N\} \\ & \cup \ \{ \downarrow p_{j,l} \mid 1 \leq l \leq j, 1 \leq j \leq N\} \\ & \cup \ \{ \downarrow p_{j,l}' \mid 1 \leq l \leq N - j + 1, 1 \leq j \leq N\} \\ Q_a &= \{ \downarrow p_N \}, \ \text{and} \\ Q_r &= \{ \downarrow p_j \mid 1 \leq j < N \} \cup \{ \downarrow r_j \mid 1 \leq j \leq N \}. \end{split}$$

The technical details of the transition matrices are given in Figure 3. (The missing parts can easily be completed.)

On symbol ¢, the computation is split into N different paths, say path_j $(1 \le j \le N)$, with amplitude $\frac{1}{\sqrt{N}}$ (Eq. (3.11)). Before reading a b, path_j enters a j-step waiting loop on each a (Eqs. (3.12)–(3.14)). If these paths never read a b, then the input is rejected exactly on \$ (Eq. (3.21)). Otherwise, the waiting strategy changes after reading a b (Eq. (3.15)). From now on, path_j enters an (N-j+1)-step waiting loop on each a (Eqs. (3.16)–(3.18)). If these paths read another b before \$, then the input is again rejected exactly on this b (Eq. (3.20)). Otherwise, all paths make a QFT on \$ such that the input is accepted by distinguished target and it is rejected otherwise (Eq. (3.19)).

Let $w\in \Sigma^*$ be an input. If w is not of the form a^*ba^* , then it is rejected certainly. If so, the decision is given by the QFT. Assume that $w=a^mba^n$ $(m,n\geq 0)$. So, path_j arrives on \$ at the $((j+1)m+(N-j+2)n+2)^{th}$ steps. It can be rewritten as

$$j(m-n) + m + (N+2)n + 2.$$

⁵Such quantum models are also known as *Kondacs-Watrous* or *Measure-Many* type [18, 45].

$$E_{\mathfrak{C}}|\overrightarrow{q_0}\rangle = \sum_{j=1}^N \frac{1}{\sqrt{N}}|\overrightarrow{q_j}\rangle.$$
 (3.11)

Wait

 $\overline{\mathsf{path}_i} \ (1 \le j \le N)$:

$$E_a|\overrightarrow{q_j}\rangle = |\downarrow p_{j,1}\rangle \tag{3.12}$$

$$E_a | \downarrow p_{j,l} \rangle = | \downarrow p_{j,l+1} \rangle, \quad 1 \le l < j$$
 (3.13)

$$E_a|\downarrow p_{j,j}\rangle = |\overrightarrow{q_j}\rangle \tag{3.14}$$

 $\mathsf{path}_i \ (1 \le j \le N)$:

$$E_b|\overrightarrow{q_j}\rangle = |\overrightarrow{q_j'}\rangle \tag{3.15}$$

 $\mathsf{path}_i \ (1 \le j \le N)$:

$$E_a|\overrightarrow{q_j'}\rangle = |\downarrow p_{j,1}'\rangle \tag{3.16}$$

$$E_a |\downarrow p'_{i,l} \rangle = |\downarrow p'_{i,l+1} \rangle, \quad 1 \le l < N - j + 1 \tag{3.17}$$

$$E_a |\downarrow p'_{j,N-j+1} \rangle = |\overrightarrow{q'_j} \rangle.$$
 (3.18)

Qft

 $\overline{\mathsf{path}}_i \ (1 \le j \le N)$:

$$E_{\$}|\overrightarrow{q_j'}\rangle = \frac{1}{\sqrt{N}} \sum_{l=1}^{N} e^{\frac{2\pi i}{N}jl} |\downarrow p_l\rangle.$$
 (3.19)

Reject

 $\overline{\mathsf{path}_i} \ (1 \le j \le N)$:

$$E_b|\overrightarrow{q_i'}\rangle = |\downarrow r_j\rangle \tag{3.20}$$

$$E_{\$}|\overrightarrow{q_j}\rangle = |\downarrow r_j\rangle. \tag{3.21}$$

FIGURE 3. The details of the transition matrices of $\mathcal{M}_{\text{IIPAL}(1),N}$.

If m=n, then all paths arrive on \$ at the same time. So, all paths are interfered with each other, and only the distinguished target survives. That is, any input in UPAL(1) is accepted with probability 1. If $m \neq n$, then each path arrives at a different step. So, none of path is interfered with others and each distinguished target survives with amplitude $\frac{1}{N}$. That is, the input is accepted with probability $\frac{1}{N}$ since there are N different distinguished targets and each of them leads the input to be accepted with probability $\frac{1}{N^2}$. So, any input not in UPAL(1) is rejected with a probability at least $1-\frac{1}{N}$.

Theorem 3.7. For any t > 1, language UPAL(t) can be recognized by simple 1QFA $\mathcal{M}_{\text{UPAL(t)},N}$ with negative one-sided error bound $\frac{1}{N}$, where N > 2.

Proof. $\mathcal{M}_{\text{UPAL}(t),N} = (Q, \Sigma, \{\mathcal{E}_{\sigma} \mid \sigma \in \tilde{\Sigma}\}, q_0, Q_a, Q_r)$, where each \mathcal{E}_{σ} contains exactly one *unitary* operation elements, E_{σ} , and $Q = Q_c \cup Q_a \cup Q_r$, *i.e.*,

$$Q_{c} = \{\overrightarrow{q_{0}}\} \cup \{\overrightarrow{q_{j_{1},...,j_{k}}} \cup \overrightarrow{q'_{j_{1},...,j_{k}}} \mid (j_{1},...,j_{k}) \in \{1,...,N\}^{k}, 1 \leq k \leq t\}$$

$$\cup \{\downarrow p_{j_{1},...,j_{k},l} \mid 1 \leq l \leq j_{k}, (j_{1},...,j_{k}) \in \{1,...,N\}^{k}, 1 \leq k \leq t\}$$

$$\cup \{\downarrow p'_{j_{1},...,j_{k},l} \mid 1 \leq l \leq N - j_{k} + 1, (j_{1},...,j_{k}) \in \{1,...,N\}^{k}, 1 \leq k \leq t\},$$

$$Q_{r} = \{\downarrow r_{j_{1},...,j_{k},l} \cup \downarrow r'_{j_{1},...,j_{k},l} \mid (j_{1},...,j_{k}) \in \{1,...,N\}^{k}, 1 \leq k \leq t\} \cup$$

$$\{p_{l} \mid 1 \leq l \leq N - 1\}, \text{ and}$$

$$Q_{a} = \{p_{N}\}.$$

The technical parts of the transition matrices are given in Figure 4. (The missing parts can easily be completed.)

The proof is a generalization of the technique presented in the Proof of Lemma 3.6. Since we can count the number of b's, the machine can easily detect the pairs that are compared with each other. Suppose that the input is of the form

where we enumerate all b's, $m_i, n_i \ge 0$, and $1 \le i \le t$. If the input is not in this form, it is rejected exactly, which will be described soon.

Let $\mathsf{P}_{(\mathsf{m}_i,\mathsf{n}_i)}$ be a procedure to compare numbers m_i and n_i $(1 \leq i \leq t)$. We initially define $\mathsf{P}_{(\mathsf{m}_t,\mathsf{n}_t)}$ and then define the others iteratively. $\mathsf{P}_{(\mathsf{m}_t,\mathsf{n}_t)}$ is executed on

$$b_{t-1}a^{m_t}b_ta^{n_t}b_{t+1}$$

almost the same way as $\mathcal{M}_{\mathtt{UPAL}(1),N}$, given in the Proof of Lemma 3.6, where b_{t-1} , b_t , and b_{t+1} are viewed as the left end-marker, the middle b, and the right end-marker, respectively. The only difference is that the computation is not terminated in the distinguished target of the QFT. If $m_t = n_t$, then only a single distinguished target of the QFT survives with probability 1. On the other hand, if $m_t \neq n_t$, the input is rejected with probability $1 - \frac{1}{N}$.

Similarly, $P_{(m_{t-1},n_{t-1})}$ is executed on

$$b_{t-2}a^{m_{t-1}}b_{t-1}a^{m_t}b_ta^{n_t}b_{t+1}a^{n_{t-1}}b_{t+2},$$

where b_{t-2} and b_{t+2} are viewed as the left end-marker and the right end-marker, respectively. Since its main task is to compare numbers m_{t-1} and n_{t-1} , $\mathsf{P}_{(\mathsf{m}_{t-1},\mathsf{n}_{t-1})}$ indeed sees the input as

$$b_{t-2}a^{m_{t-1}}\mathbf{b}a^{n_{t-1}}b_{t+2}$$

such that \mathbf{b} $(b_{t-1}a^{m_t}b_ta^{n_t}b_{t+1})$ is viewed as the middle b. So, the execution of $\mathsf{P}_{(\mathsf{m}_{t-1},\mathsf{n}_{t-1})}$ on $b_{t-2}a^{m_{t-1}}\mathbf{b}a^{n_{t-1}}b_{t+2}$ is also the same as $\mathcal{M}_{\mathtt{UPAL}(1),N}$ except that each subpath of $\mathsf{P}_{(\mathsf{m}_{t-1},\mathsf{n}_{t-1})}$ calls a copy of $\mathsf{P}_{(\mathsf{m}_{i},\mathsf{n}_{i})}$ on b_{t-1} and each of these copies leaves

Split

$$E_{\mathbb{C}}|q_0\rangle = \sum_{j=1}^{N} \frac{1}{\sqrt{N}} |\overrightarrow{q_j}\rangle$$
 (3.23)

 $\mathsf{path}_{j_1, \dots, j_k} \ ((j_1, \dots, j_k) \in \{1, \dots, N\}^k \ \text{and} \ 1 \le k < t)$:

$$E_b|\overrightarrow{q_{j_1,\dots,j_k}}\rangle = \sum_{j_{k+1}=1}^N \frac{1}{\sqrt{N}}|\overrightarrow{q_{j_1,\dots,j_k,j_{k+1}}}\rangle.$$
 (3.24)

W<u>ait</u>

 $\overline{\mathsf{path}_{j_1,...,j_k}} \ ((j_1,...,j_k) \in \{1,...,N\}^k \ \text{and} \ 1 \le k \le t)$:

$$E_a|\overrightarrow{q_{j_1,\dots,j_k}}\rangle = |\downarrow p_{j_1,\dots,j_k,1}\rangle \tag{3.25}$$

$$E_a |\downarrow p_{j_1, \dots, j_k, l} \rangle = |\downarrow p_{j_1, \dots, j_k, l+1} \rangle, \quad 1 \le l < j_k$$

$$(3.26)$$

$$E_a|\downarrow p_{j_1,\dots,j_k,j_k}\rangle = |\overrightarrow{q_{j_1,\dots,j_k}}\rangle \tag{3.27}$$

 $\mathsf{path}_{j_1,\dots,j_t}\ ((j_1,\dots,j_t)\in\{1,\dots,N\}^t)\colon$

$$E_b|\overrightarrow{q_{j_1,\ldots,j_t}}\rangle = |\overrightarrow{q'_{j_1,\ldots,j_t}}\rangle.$$
 (3.28)

 $\mathsf{path}_{j_1, \dots, j_k} \ ((j_1, \dots, j_k) \in \{1, \dots, N\}^k \ \text{and} \ 1 \le k \le t)$:

$$E_a|\overrightarrow{q'_{i_1,\ldots,i_k}}\rangle = |\downarrow p'_{i_1,\ldots,i_k,1}\rangle$$
 (3.29)

$$E_a | \downarrow p'_{j_1, \dots, j_k, l} \rangle = | \downarrow p_{j_1, \dots, j_k, l+1} \rangle, \quad 1 \le l < N - j_k + 1$$
(3.22)

$$E_a |\downarrow p'_{j_1,\dots,j_k,N-j_k+1}\rangle = |\overrightarrow{q'_{j_1,\dots,j_k}}\rangle. \tag{3.31}$$

 $\frac{\text{QFT}}{\mathsf{path}_{j_1,...,j_k}} \; ((j_1,\ldots,j_k) \in \{1,\ldots,N\}^k \text{ and } 1 < k \leq t \;)$:

$$E_b|\overrightarrow{q'_{j_1,\dots,j_k}}\rangle = \frac{1}{\sqrt{N}} \left(\sum_{l=1}^{N-1} e^{\frac{2\pi i}{N}jl} |\downarrow r_{j_1,\dots,j_{k-1},l}\rangle \right) + \frac{1}{\sqrt{N}} e^{2\pi i j} |\overrightarrow{q'_{j_1,\dots,j_{k-1}}}\rangle$$
(3.32)

 $path_{j_1} \ (1 \le j_1 \le N)$:

$$E_{\$}|\overrightarrow{q'_{j_1}}\rangle = \frac{1}{\sqrt{N}} \sum_{l=1}^{N} e^{\frac{2\pi i}{N}jl} |\downarrow p_l \rangle.$$
 (3.33)

 $\frac{\text{Reject}}{\mathsf{path}_{j_1}} \ (1 \leq j_1 \leq N) :$

$$E_b|\overrightarrow{q'_{j_1}}\rangle = |\downarrow r_{j_1}\rangle$$
 (3.34)

 $\mathsf{path}_{j_1,\ldots,j_k}\ ((j_1,\ldots,j_k)\in\{1,\ldots,N\}^k \text{ and } 1\leq k\leq t)$:

$$E_{\$}|\overrightarrow{q_{j_1,\ldots,j_k}}\rangle = |\downarrow r_{j_1,\ldots,j_k}\rangle \tag{3.35}$$

 $\mathsf{path}_{j_1, \dots, j_k} \ ((j_1, \dots, j_k) \in \{1, \dots, N\}^k \ \mathrm{and} \ 1 < k \leq t)$:

$$E_{\$}|\overrightarrow{q'_{j_1,\ldots,j_k}}\rangle = |\downarrow r'_{j_1,\ldots,j_k}\rangle.$$
 (3.36)

FIGURE 4. The details of the transition matrices of $\mathcal{M}_{UPAL(t),N}$.

the computation to the subpath calling itself after reading b_{t+1} if computation is not terminated in this copy of $P_{(m_t,n_t)}$.

Case $m_t \neq n_t$. The input is rejected with probability $1 - \frac{1}{N}$ by $\mathsf{P}_{(\mathsf{m}_t,\mathsf{n}_t)}$. Since it gives the desired upper bound on the error, we no longer care about the decision of $\mathsf{P}_{(\mathsf{m}_{t-1},\mathsf{n}_{t-1})}$.

Case $m_t = n_t$. $P_{(m_i,n_i)}$ leaves the computation to its parent exactly. Moreover, the balance between subpaths of $P_{(m_{t-1},n_{t-1})}$ is still preserved. Thus, if $m_{t-1} = n_{t-1}$, then only a single distinguished target of the QFT survives with probability 1. On the other hand, if $m_{t-1} \neq n_{t-1}$, the input is rejected with probability $1 - \frac{1}{N}$.

In an iterative way, $P_{(m_{t-2},n_{t-2})}$ is defined by $P_{(m_{t-1},n_{t-1})}$ and so on. If the input has not been rejected before, the last decision is given on \$ by $P_{(m_1,n_1)}$ and the input is accepted exactly if $m_1 = n_1$. (Note that the input can only be accepted after the last QFT.) In any other case, there exists a maximum i providing $m_i \neq n_i$, the input is rejected with a probability at least $1 - \frac{1}{N}$.

Any input that does not fit the form given in equation (3.22), *i.e.* the input does not contain exactly (2t-1) bs, can be deterministically detected. Therefore, it can be rejected exactly.

In the remaining part, we give the technical details. The computation starts with $P_{(m_1,n_1)}$. After reading ¢, the computation splits into N paths with equal amplitudes, say path_{j_1} $(1 \le j_1 \le N)$ (Eq. (3.23)). These are subpaths of $\mathsf{P}_{(\mathsf{m}_1,\mathsf{n}_1)}$. After reading the first b, each path_{j_1} calls a copy of $\mathsf{P}_{(\mathsf{m}_2,\mathsf{n}_2)}$ and so the computation is again split into N paths with equal amplitudes, i.e. $path_{i_1}$ is split into N paths path_{j_1,j_2} $(1 \leq j_1,j_2 \leq N)$. This process is repeated until reading the (t-1)th b (Eq. (3.24)). Thus, after reading the (t-1)th b, each path has t indexes, i.e. $\mathsf{path}_{j_1,\ldots,j_t}$ $(1 \le j_k \le N \text{ and } 1 \le k \le t)$. Note that, any path with index $(j_1, j_2, \ldots, j_{k'})$ $(1 \leq k' \leq t)$ has responsibility of comparing numbers $m_{k'}$ and $n_{k'}$. Before (resp., after) reading the tth b, if j_k is the last index of the path, then, it waits j_k (resp., $N-j_k+1$) steps over each a (Eqs. (3.25)–(3.31)). After reading the tth b, all paths start to make N-way QFT over each b in order to compare the numbers under their responsibility (Eqs. (3.32) and (3.33)). After the QFT (except the last one), the computation continues with the paths, from which the current paths were created in the previous steps (i.e. technically the rightmost index is dropped) with probability 1 in case of successful QFT and with probability $\frac{1}{N}$ otherwise. After the last QFT, the input is accepted in case of successful QFT (Eq. (3.33)). In any case of scanning less than 2t-1 b's (Eqs. (3.35) and (3.36)) or more than 2t-1 b's (Eq. (3.34)), the input is rejected exactly.

3.4. A NEW CLASSICAL SEPARATION RESULT

In this section, we present an example for how the studies in quantum computation can help to find new results on classical computation. We give a separation result between the languages recognized by rtDkBCA and the languages recognized by rtNkBCA for any k > 0. Note that both models recognize regular languages in case of k = 0.

Theorem 3.8. For any k > 0, if L is recognized by a rtDkBCA, then $\overline{L} \in NQAL$.

Proof. Let $\mathcal{D} = (Q, \Sigma, \{A_{\sigma} \mid \sigma \in \tilde{\Sigma}\}, q_1, Q_a)$ be a simple rtDkBCA recognizing L, where $Q = \{q_1, \ldots, q_n\}$ (n > 0). For a given input string $w \in \Sigma^*$, the state-transition of \mathcal{D} is traced by a (stochastic) column vector, *i.e.*

$$v_i = A_{\tilde{w}_i} v_{i-1}$$

and

$$v_{|\tilde{w}|} = A_{\$} A_{w_{|w|}} \cdots A_{w_1} A_{\complement} v_0,$$

where $v_0 = (1 \ 0 \ \cdots \ 0)^T$ is an *n*-dimensional initial column vector and $1 \le i \le |\tilde{w}|$. It can be easily verified that \mathcal{D} is in state q_j at the end of the computation if and only if $v_{|\tilde{w}|}[j] = 1$, where $1 \le j \le n$. (Note that each intermediate v_i is also a stochastic zero-one vector, where $1 \le i < |\tilde{w}|$.)

The probability of \mathcal{D} finishing in an accepting state on w, which can be either 1 or 0, is calculated as $fv_{|\tilde{w}|}$, where f is an n-dimensional row vector such that f[i] = 1 if $q_i \in Q_a$.

Let p_l be the lth smallest prime $(1 \leq l \leq k)$. In the above schema, the counter operations of \mathcal{D} can be simulated by using a simple number-theoretic method: when the lth counter of \mathcal{D} is updated by 1 (resp., 0 or -1), the nonzero entry of v_i is updated by multiplying with p_l (resp., 1 or $\frac{1}{p_l}$), where $1 \leq l \leq k$ and $1 \leq i \leq |\tilde{w}|$. This method can be embedded into the transition matrices. That is, if the value(s) of counter(s) is (are) updated with respect to $c \in \{-1,0,1\}^k$ when entering state $q_j \in Q$, in each $A_{\sigma \in \tilde{\Sigma}}$, the nonzero entries on the j^{th} row are replaced with

$$\prod_{l=1}^k (p_l)^{c[l]}.$$

We denote these new matrices as $A'_{\sigma \in \tilde{\Sigma}}$.

Suppose that the value(s) of the counter(s) is (are) $C \in \mathbb{Z}^k$ at the end of the computation, and the computation ends in $q_j \in Q$ on input $w \in \Sigma^*$. Then, it can be verified in a straightforward way that

$$v_{|\tilde{w}|}[j] = \prod_{l=1}^{k} (p_l)^{C[l]}, \tag{3.37}$$

which is 1 if and only if each counter value is zero.

Let $A''_{\sigma \in \tilde{\Sigma}}$ be $(n+1) \times (n+1)$ -dimensional matrices obtained from A'_{σ} as

$$A''_{\sigma} = \begin{pmatrix} & 0 \\ \vdots & \vdots \\ 0 \\ \hline 0 \cdots 0 \\ 1 \end{pmatrix}.$$

We design a TuFA \mathcal{T} based on \mathcal{D} as follows:

$$\mathcal{T} = (Q', \Sigma, \{B_{\sigma} \mid \sigma \in \Sigma\}, v'_0, f'),$$

where $Q' = Q \cup \{q_{n+1}\},\$

$$v_0' = A_{\mathbb{C}}'' \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix}, \quad f' = (f \mid 1)A_{\$}'', \text{ and } B_{\sigma} = A_{\sigma}''.$$

Hence, by using the scenario given in (3.37), we can verify that:

- if $q_j \in Q_a$, then $f_{\mathcal{T}}(w) = \left(\prod_{l=1}^k (p_l)^{C[l]}\right) 1$, i.e.
 - $-f_{\mathcal{T}}(w)=0$ if each counter value is zero, and
 - $-f_{\mathcal{T}}(w) \neq 0$ if at least one counter value is not zero;
- if $q_i \notin Q_a$, then $f_{\mathcal{T}}(w) = -1$.

Let $\mathcal{T}^2 = \mathcal{T} \otimes \mathcal{T}$ (see Fact 3.2). Then, for $w \in L$ (resp., $w \notin L$), $f_{\mathcal{T}^2}(w) = 0$ (resp., $f_{\mathcal{T}^2}(w) > 0$). Thus, $\overline{L} \in \mathsf{NQAL}$.

In [13], it is shown that SAY = $\{w \mid \exists u_1, u_2, v_1, v_2 \in \{a, b\}^*, w = u_1bu_2 = v_1bv_2, |u_1| = |v_2|\}$ cannot be recognized by a rtQFA with unbounded error (and so SAY \notin NQAL and $\overline{\text{SAY}} \notin$ NQAL [45]). In other words, SAY cannot be recognized by any rtDkBCA (k > 0). However, it can be easily shown that SAY can be recognized by a rtN1BCA: two b's (those can also be the same) can be selected nondeterministically and by using a blind counter, the lengths of the substrings before the first b and after the second b are compared.

Corollary 3.9. For any k > 0, the class of languages recognized by rtDkBCAs is a proper subset of the class of languages recognized by rtNkBCAs.

In [2], the two-way⁶ classical-head⁷ quantum finite automaton (2QCFA) was introduced (see also [43,45]).

Fact 3.10 ([43]). Let L be language, and T be a TuFA such that $f_{\mathcal{T}}(w) > 0$ for $w \in L$ and $f_{\mathcal{T}}(w) = 0$ for $w \notin L$. Then both L and \overline{L} can be recognized by 2QCFAs with any error bound.

Based on this fact and Theorem 4, we can conclude the following corollary.

Corollary 3.11. Any language recognized by a rtDkBCA can also be recognized by a 2QCFA with any error bound.

 $^{^6\}mathrm{The}$ input head can move one square to the right or to the left or stay on the same square.

⁷ The quantum part of the machine is just a fixed-size quantum register. For example, this is not the case for 1QFAs.

Acknowledgements. We thank A.C. Cem Say for his comments on a draft version of this paper, Rūsiņš Freivalds and Alexanders Rivoss for their answers to our questions regarding the content of [12], and Juraj Hromkovič for his answers to our questions regarding blind counters. We are also grateful to the anonymous referees for their very useful suggestions.

APPENDIX A. GENERAL DEFINITION OF QUANTUM MACHINES

Superoperators are implemented by interacting the system in interest with a finite-size ancilla quantum register [27]. More specifically:

- (1) the finite register is initialized to a predefined state;
- (2) both systems are combined;
- (3) a unitary operator is applied to the composed system; and
- (4) the finite register is discarded.

In some cases, a measurement is done on the finite register in computational basis before the discarding process. Therefore, apart from classical ones, quantum models are defined with this finite register in order to explicitly specify the local transitions on this register. We denote the set of symbols for the finite register by Ω with a special initial symbol, denoted by ω_1 , the initial symbol. In case of measurement, we divide Ω into $|\Delta|$ disjoint subsets, i.e.

$$\Omega = \{\Omega_{\tau_1}, \dots, \Omega_{\tau_{|\Delta|}}\}$$

where Δ is the set of outcomes. Then, before the discarding process, the projective measurement

$$P = \left\{ P_{\tau} \mid P_{\tau} = \sum_{\omega \in \Omega_{\tau}} |\omega\rangle\langle\omega|, \tau \in \Delta \right\}$$

is applied to the finite register.

Now, we give the definitions of general rtQ1BCA and 1QFA with the constraints on their transition functions. Then, we show how their simplified counterparts can be obtained.

A.1. General realtime quantum automaton with one blind counter

A realtime quantum automaton with one blind counter is a 6-tuple

$$\mathcal{M} = (Q, \Sigma, \Omega, \delta, q_0, Q_a),$$

where δ is the transition function, which is specified as

$$\delta(q,\sigma) = \sum_{\alpha_{(p,c,\omega)}} \alpha_{(p,c,\omega)}(p,c,\omega)$$

(or alternatively $\delta(q, \sigma, p, c, \omega) = \alpha_{(p,c,\omega)}$) such that \mathcal{M} changes the internal state to $p \in Q$, update the counter value by $c \in \{-m, \ldots, m\}$, and writes $\omega \in \Omega$ on

the finite register with amplitude $\alpha_{(p,c,\omega)}$ when it is originally in state $q \in Q$ and reads $\sigma \in \Sigma$ on the input tape. Since the value of m can be arbitrarily chosen, we set it to 1 in the remaining part.

The transition function δ must satisfy the following local conditions of well-formedness (see also [33]): for any choice of $q_1, q_2 \in Q$ and $\sigma \in \tilde{\Sigma}$,

$$\sum_{q' \in Q, c \in \{-1, 0, 1\}, \omega \in \Omega} \overline{\delta(q_1, \sigma, q', c, \omega)} \delta(q_2, \sigma, q', c, \omega) = \begin{cases} 1 & \text{if } q_1 = q_2 \\ 0 & \text{otherwise,} \end{cases}$$
(A.1)

$$\sum_{q' \in Q, \omega \in \Omega} \overline{\delta(q_1, \sigma, q', +1, \omega)} \delta(q_2, \sigma, q', 0, \omega) + \overline{\delta(q_1, \sigma, q', 0, \omega)} \delta(q_2, \sigma, q', -1, \omega) = 0,$$
(A.2)

and

$$\sum_{q' \in Q, \omega \in \Omega} \overline{\delta(q_1, \sigma, q', +1, \omega)} \delta(q_2, \sigma, q', -1, \omega) = 0.$$
(A.3)

As can be easily verified, if the machine is simple, then the constraints given in equations (A.2) and (A.3) becomes trivial and the constraint given in equation (A.1) can be replaced by

$$\sum_{q' \in Q, \omega \in \Omega} \overline{\delta(q_1, \sigma, q', \omega)} \delta(q_2, \sigma, q', \omega) = \begin{cases} 1 & \text{if } q_1 = q_2 \\ 0 & \text{otherwise.} \end{cases}$$
(A.4)

Now, we show how to construct a superoperator for each $\sigma \in \tilde{\Sigma}$. Let $q_1, \ldots, q_{|Q|}$ be an ordering of the states. We define $E_{\sigma,\omega}$ such that the (j,i)th-entry of the matrix is equal to $\delta(q_i,\sigma,q_j,\omega)$, where $\omega \in \Omega$, $1 \leq i,j \leq |Q|$. Therefore, we obtain a bijection between the matrices and the transition function, *i.e.*

$$E_{\sigma,\omega}(j,i) = \delta(q_i,\sigma,q_j,\omega).$$

By grouping these matrices by $\sigma \in \tilde{\Sigma}$, we obtain $\mathcal{E}_{\sigma} = \{E_{\sigma,\omega_1}, \ldots, E_{\sigma,\omega_{|\Omega|}}\}$. If we concatenating each element of \mathcal{E}_{σ} one under the other, we get a new matrix, *i.e.*

$$\mathsf{E}_{\sigma,\omega_{1}}$$

$$\mathsf{E}_{\sigma} = \boxed{\vdots}$$

$$E_{\sigma,\omega_{|\Omega|}}$$

whose columns form an orthonormal set if and only if the constraint in equation (A.4) is satisfied. This is exactly what a superoperator is. Since the presence of the finite register is not mandatory, we prefer to drop it from the formal definition of simple rtQ1BCA.

A.2. General one-way quantum automaton

A one-way quantum finite automaton [45] is a 6-tuple

$$\mathcal{M} = (Q, \Sigma, \Omega, \delta, q_0, \Delta),$$

where $\Delta = \{a, r, c\}$, and so Ω is partitioned into 3 disjoint subsets, *i.e.* $\Omega = \Omega_c \cup \Omega_a \cup \Omega_r$. As described above, a projective measurement is applied to the finite register at each step of the computation and the following actions are followed: (i) the computation continues if "c" is observed and (ii) the computation is terminated and the input is accepted (resp., rejected) if "a" (resp., "r") is observed.

The transition of \mathcal{M} is specified as:

$$\delta(q,\sigma) = \sum_{\alpha_{(p,d,\omega)}} \alpha_{(p,d,\omega)}(p,d,\omega)$$

(or alternatively $\delta(q, \sigma, p, d, \omega) = \alpha_{(p,d,\omega)}$) such that \mathcal{M} changes the internal state to $p \in Q$, update the position of the input head with respect to $d \in \{\downarrow, \rightarrow\}$, and writes $\omega \in \Omega$ on the finite register with amplitude $\alpha_{(p,d,\omega)}$ when it is originally in state $q \in Q$ and reads $\sigma \in \Sigma$ on the input tape.

The transition function δ must satisfy the following local conditions of well-formedness (see also [45]): for any choice of $q_1, q_2 \in Q$ and $\sigma \in \tilde{\Sigma}$,

$$\sum_{q' \in Q, d \in \{\downarrow, \to\}, \omega \in \Omega} \overline{\delta(q_1, \sigma, q', d, \omega)} \delta(q_2, \sigma, q', d, \omega) = \begin{cases} 1 & \text{if } q_1 = q_2 \\ 0 & \text{otherwise} \end{cases}$$
(A.5)

and

$$\sum_{q' \in Q, \omega \in \Omega} \overline{\delta(q_1, \sigma, q', \to, \omega)} \delta(q_2, \sigma, q', \downarrow, \omega) = 0.$$
(A.6)

As can be easily verified, if the machine is simple, then constrains given in equation (A.6) becomes trivial and the constraint given in equation (A.5) can be replaced by

$$\sum_{q' \in \mathcal{Q}, \omega \in \Omega} \overline{\delta(q_1, \sigma, q', \omega)} \delta(q_2, \sigma, q', \omega) = \begin{cases} 1 & \text{if } q_1 = q_2 \\ 0 & \text{otherwise.} \end{cases}$$
(A.7)

By using the same procedure described at the end of the previous section, we can obtain a superoperator for each $\sigma \in \tilde{\Sigma}$. On the other hand, in order to drop the presence of the finite register from the formal definition of simple 1QFAs, we need one more step:

We define a state set as $Q' = Q \times \Delta$. Let $Q'_{\tau} = Q \times \{\tau\}$ and $(q_1, n), \ldots, (q_{|Q|}, n), (q_1, a), \ldots, (q_{|Q|}, a), (q_1, r), \ldots, (q_{|Q|}, r)$ be an ordering of the new states. We define new $E'_{\sigma,\omega}$ by using $E_{\sigma,\omega}$ as follows

1	$E_{\sigma,\omega}$	*	*	0	*	*	0		*	*	
	0	*	*	$E_{\sigma,\omega}$	*	*	0		*	*	
	0	*	*	0	*	*	E_{σ}	,ω	*	*	,
,		_		$\overline{}$	_		$\overline{}$	$\overline{}$	_		
if $\omega \in \Omega_n$			if $\omega \in \Omega_a$			if $\omega \in \Omega_r$					

where the entries of "*" can be filled in so that each $\mathcal{E}'_{\sigma} = \{E'_{\sigma,\omega} \mid \omega \in \Omega\}$ forms a superoperator. If we apply projective measurement

$$P = \left\{ P_{\tau} \mid P_{\tau} = \sum_{q \in Q'_{\tau}} |q\rangle\langle q| \right\}$$

to the state set instead of the finite register, we obtain an identical computation. Note that, since no transition defined for $q' \in Q'_a \cup Q'_r$ is actually implemented, the entries of "*" do not affect the computation.

References

- A. Ambainis and R. Freivalds, 1-way quantum finite automata: strengths, weaknesses and generalizations, in Proc. of 39th Annual Symposium on Foundations of Computer Science, FOCS'98 (1998) 332-341.
- [2] A. Ambainis and J. Watrous, Two-way finite automata with quantum and classical states. Theoret. Comput. Sci. 287 (2002) 299–311.
- [3] A. Ambainis and A. Yakaryılmaz, Superiority of exact quantum automata for promise problems. Inf. Process. Lett. 112 (2012) 289–291.
- [4] A. Ambainis and A. Yakaryılmaz, Automata: from Mathematics to Applications, in *Automata and quantum computing*, in preparation.
- [5] A. Ambainis, A. Nayak, A. Ta-Shma and U. Vazirani, Dense quantum coding and quantum finite automata. J. ACM 49 (2002) 496-511.
- [6] E. Bernstein and U. Vazirani, Quantum complexity theory. SIAM J. Comput. 26 (1997) 1411–1473.
- [7] A. Bertoni and M. Carpentieri, Analogies and differences between quantum and stochastic automata. Theor. Comput. Sci. 262 (2001) 69–81.
- [8] A. Bertoni, C. Mereghetti and B. Palano, Small size quantum automata recognizing some regular languages. Theor. Comput. Sci. 340 (2005) 394–407.
- [9] R. Bonner, R. Freivalds and M. Kravtsev, Quantum versus probabilistic one-way finite automata with counter, in Proc. of SOFSEM 2007: Theory and Practice of Computer Science. Lett. Notes Comput. Sci. 2234 (2001) 181–190.
- [10] A. Brodsky and N. Pippenger, Characterizations of 1-way quantum finite automata. SIAM J. Comput. 31 (2002) 1456–1478.
- [11] P.C. Fischer, A.R. Meyer and A.L. Rosenberg, Counter machines and counter languages. Math. Syst. Theory 2 (1968) 265–283.
- [12] R. Freivalds, Language recognition using finite probabilistic multitape and multihead automata. Problemy Peredachi Informatsii 15 (1979) 99–106.
- [13] R. Freivalds, A. Yakaryılmaz and A.C.C. Say, A new family of nonstochastic languages. Inf. Process. Lett. 110 (2010) 410–413.

- [14] M. Golovkins, Quantum pushdown automata, in Proc. of SOFSEM'00: Theory and Practice of Informatics. Lett. Notes Comput. Sci. 1963 (2000) 336–346.
- [15] S.A. Greibach, Remarks on blind and partially blind one-way multicounter machines. Theoret. Comput. Sci. 7 (1978) 311–324.
- [16] K. Hamaguchi M. Nakanishi and T. Indoh, On the power of quantum pushdown automata with a classical stack and 1.5-way quantum finite automata. Technical Report NAIST-IS-TR2001005, Nara Institute of Science and Technology (2001). Available on http://isw3. naist.jp/IS/TechReport/report/2001005.ps.
- [17] M. Hirvensalo, Quantum automata with open time evolution. Int. J. Natural Comput. Res. 1 (2010) 70–85.
- [18] A. Kondacs and J. Watrous, On the power of quantum finite state automata, in Proc. of 38th Annual Symposium on Foundations of Computer Science, FOCS'97. Lett. Notes Comput. Sci. (1997) 66-75.
- [19] M. Kravtsev, Quantum finite one-counter automata, in Proc. of SOFSEM'99: Theory and Practice of Computer Science. Lect. Notes Comput. Sci. 1725 (1999) 431–440.
- [20] M. Kutyłowski, Multihead one-way finite automata. Theoret. Comput. Sci. 85 (1991) 135– 153.
- [21] C. Mereghetti and B. Palano, On the size of one-way quantum finite automata with periodic behaviors. Theoret. Inform. Appl. 36 (2002) 277–291.
- [22] C. Mereghetti and B. Palano, Quantum automata for some multiperiodic languages. Theoret. Comput. Sci. 387 (2007) 177–186.
- [23] C. Moore and J.P. Crutchfield, Quantum automata and quantum grammars. Theoret. Comput. Sci. 237 (2000) 275–306.
- [24] Y. Murakami, M. Nakanishi, S. Yamashita and K. Watanabe, Quantum versus classical pushdown automata in exact computation. IPSJDC 1 (2005) 426–435.
- [25] M. Nakanishi, T. Indoh, K. Hamaguchi and T. Kashiwabara, On the power of nondeterministic quantum finite automata. *IEICE Trans. Inf. Syst.* E85-D (2002) 327–332.
- [26] M. Nakanishi, K. Hamaguchi and T. Kashiwabara, Expressive power of quantum pushdown automata with classical stack operations under the perfect-soundness condition. *IEICE Trans. Inf. Syst.* E89-D (2006) 1120–1127.
- [27] M.A. Nielsen and I.L. Chuang, Quantum Computation and Quantum Information. Cambridge University Press (2000).
- [28] A. Paz, Introduction to Probabilistic Automata. Academic Press, New York (1971).
- [29] M.O. Rabin, Probabilistic automata. Inf. Control 6 (1963) 230–243.
- [30] A.L. Rosenberg, On multi-head finite automata. IBM J. Res. Devel. 10 (1966) 388–394.
- [31] A.C.C. Say and A. Yakaryılmaz, Quantum function computation using sublogarithmic space (2010) (Poster presentation at QIP2010) [arXiv:1009.3124].
- [32] A.C.C. Say and A. Yakaryılmaz, Computation with narrow CTCs, in Unconventional Computation. Lect. Notes Comput. 6714 (2011) 201–211.
- [33] A.C.C. Say and A. Yakaryılmaz, Quantum counter automata. Int. J. Found. Comput. Sci. To appear.
- [34] P. Turakainen, Generalized automata and stochastic languages. Proc. Am. Math. Soc. 21 (1969) 303–309.
- [35] J. Watrous, Space-bounded quantum complexity. J. Comput. Syst. Sci. 59 (1999) 281–326.
- [36] J. Watrous, Quantum computational complexity, in Encyclopedia of Complexity and Systems Science, edited by R.A. Meyers. Springer (2009) 7174–7201.
- [37] J. Watrous, Personal communication (2009).
- [38] A. Yakaryılmaz, Superiority of one-way and realtime quantum machines and new directions, in Third Workshop on Non-Classical Models for Automata and Applications (2011) 209–224.
- [39] A. Yakaryılmaz, Classical and Quantum Computation with Small Space Bounds. Ph.D. thesis, Boğaziçi University (2011) [arXiv:1102.0378].
- [40] A. Yakaryılmaz and A.C.C. Say, Efficient probability amplification in two-way quantum finite automata. Theoret. Comput. Sci. 410 (2009) 1932–1941.

- [41] A. Yakaryılmaz and A.C.C. Say, Languages recognized with unbounded error by quantum finite automata, in Proc. of 4th International Computer Science Symposium in Russia, CSR'09. Lect. Notes Comput. Sci. 5675 (2009) 356–367.
- [42] A. Yakaryılmaz and A.C.C. Say, Languages recognized by nondeterministic quantum finite automata. Quantum Inf. Comput. 10 (2010) 747–770.
- [43] A. Yakaryılmaz and A.C.C. Say, Succinctness of two-way probabilistic and quantum finite automata. Discrete Math. Theoret. Comput. Sci. 12 (2010) 19–40.
- [44] A. Yakaryılmaz and A.C.C. Say, Probabilistic and quantum finite automata with postselection. Technical Report [arXiv:1102.0666] (2011). (A preliminary version of this paper appeared in Proc. of Randomized and Quantum Computation (satellite workshop of MFCS and CSL)) (2010) 14–24.
- [45] A. Yakaryılmaz and A.C.C. Say, Unbounded-error quantum computation with small space bounds. Inf. Comput. 279 (2011) 873–892.
- [46] A. Yakaryılmaz and A.C.C. Say, Proving the power of postselection. Fundam. Inform. (2012). [arXiv:1111.3125].
- [47] A. Yakaryılmaz, R. Freivalds, A.C.C. Say and R. Agadzanyan, Quantum computation with write-only memory. *Natural Comput.* 11 (2012) 81–94.
- [48] T. Yamasaki, H. Kobayashi, Y. Tokunaga and H. Imai, One-way probabilistic reversible and quantum one-counter automata. *Theoret. Comput. Sci.* 289 (2002) 963–976.
- [49] T. Yamasaki, H. Kobayashi and H. Imai, Quantum versus deterministic counter automata. Theoret. Comput. Sci. 334 (2005) 275–297.

Communicated by C. Mereghetti.

Received November 8, 2011. Accepted June 8, 2012.