Information and Computation 209 (2011) 873–892

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/ic

Unbounded-error quantum computation with small space bounds ☆,☆☆

Abuzer Yakaryılmaz *, A.C. Cem Say

Boğaziçi University, Department of Computer Engineering, Bebek 34342 İstanbul, Turkey

ARTICLE INFO

Article history:
Received 21 July 2010
Available online 18 February 2011

Keywords:
Sublogarithmic space
Quantum Turing machines
Probabilistic Turing machines
Quantum finite automata
Probabilistic finite automata
Unbounded error

ABSTRACT

We prove the following facts about the language recognition power of quantum Turing machines (QTMs) in the unbounded error setting: QTMs are strictly more powerful than probabilistic Turing machines for any common space bound s satisfying $s(n) = o(\log \log n)$. For "one-way" Turing machines, where the input tape head is not allowed to move left, the above result holds for $s(n) = o(\log n)$. We also give a characterization for the class of languages recognized with unbounded error by real-time quantum finite automata (QFAs) with restricted measurements. It turns out that these automata are equal in power to their probabilistic counterparts, and this fact does not change when the QFA model is augmented to allow general measurements and mixed states. Unlike the case with classical finite automata, when the QFA tape head is allowed to remain stationary in some steps, more languages become recognizable. We define and use a QTM model that generalizes the other variants introduced earlier in the study of quantum space complexity.

1. Introduction

The investigation of the power of space-bounded quantum computers was initiated by Watrous [1–3], who defined several machine models suitable for the analysis of this problem, and proved that those quantum machines are equivalent to probabilistic Turing machines (PTMs) for any common space-constructible bound $s(n) \in \Omega(\log n)$ in the unbounded error case. Together with Kondacs, Watrous also examined the case of constant space bounds, defining [4] a quantum finite automaton (QFA) variant, which inspired a fruitful line of research [5–15].

In this paper, we answer two open questions posed in the previous study of space-bounded quantum complexity regarding sublogarithmic space bounds. We first show that unbounded-error quantum Turing machines are strictly more powerful than PTMs for any common space bound s satisfying $s(n) = o(\log \log n)$. For "one-way" Turing machines, where the input tape head is not allowed to move left, the above result holds for $s(n) = o(\log n)$. We then give a full characterization of the class of languages recognized with unbounded error by real-time QFAs with restricted measurements. It turns out that these automata have the same power as their classical counterparts, and this fact does not change when the QFA definition is generalized in accordance with the modern approach [13,16]. Unlike the case with classical finite automata, when the QFA tape head is allowed two-way movement, or even just the option of remaining stationary during some steps, more languages become recognizable.

As hinted above, early models of QTMs and QFAs [2,4,17] were unduly restricted in their definitions, and did not reflect the full potential of quantum mechanics in their computational power. This problem was later addressed [3,9,13] by the

☆ A preliminary version of this paper appeared in the *Proceedings of the Fourth International Computer Science Symposium in Russia*, pp. 356–367, 2009.

☆☆ This work was partially supported by the Boğaziçi University Research Fund with Grant 08A102 and by the Scientific and Technological Research Council of Turkey (TÜBİTAK) with Grant 108E142.

* Corresponding author.
E-mail addresses: abuzer@boun.edu.tr, abuzer.yakaryilmaz@gmail.com (A. Yakaryılmaz), say@boun.edu.tr (A.C.C. Say).

0890-5401/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2011.01.008

incorporation of general quantum operations and mixed states into the models. Aiming to provide the most general reasonable machine model for the study of quantum space complexity, while taking into account the peculiarities of small space bounds, we define a QTM variant of our own. The other QTM models are shown to be specializations of our variant. We conjecture that our model is more powerful than the other variants, at least for some space bounds.

The rest of this paper is structured as follows: Section 2 contains relevant background information. The machine models we use, including our new variant, are defined in Section 3. The superiority of QTMs over PTMs for a range of small space bounds is established in Section 4. In Section 5, we characterize the languages recognized with unbounded error by all QFAs that are at least as powerful as the "Kondacs–Watrous" model. Section 6 is a conclusion. Some technical details about our various quantum models are covered in Appendix A and Appendix B.

2. Preliminaries

We start by introducing some notation and terminology that will be used frequently in the remainder of the paper.

2.1. Basic notation

The following is a list of notational items that appear throughout the paper:

- Σ is the input alphabet, not containing the end markers ¢ and \$, and $\tilde{\Sigma} = \Sigma \cup \{¢, \$\}$.
- Γ is the work tape alphabet, containing a distinguished blank symbol denoted #.
- Δ denotes the finite set of measurement outcomes.
- Q is the set of internal states, where q_1 is the initial state.
- δ is the transition function, which determines the behavior of the machine.
- $\diamond$ is the set $\{\leftarrow, \downarrow, \rightarrow\}$, where $\leftarrow$ means that the (corresponding) head moves one square to the left, $\downarrow$ means that the head stays on the same square, and $\rightarrow$ means that the head moves one square to the right.
- $f_{\mathcal{M}}(w)$ is the acceptance probability (or, in one context, the acceptance value) associated by machine $\mathcal{M}$ to input string w.
- ε is the empty string.
- For a given string w, $|w|$ is the length of w, w_i is the ith symbol of w, and $\tilde{w}$ represents the string ¢w\$.
- $\mathbb{N}$ is the set of nonnegative integers.
- $\mathbb{Z}^+$ is the set of positive integers.
- For a given (row or column) vector v, $v[i]$ is the ith entry of v.
- For a given matrix A, $A[i,j]$ is the (i,j)th entry of A.
- Some fundamental conventions in Hilbert space are as follows:
  - v and its conjugate transpose are denoted $|v\rangle$ and $\langle v|$, respectively;
  - the multiplication of $\langle v_1|$ and $|v_2\rangle$ is shortly written as $\langle v_1|v_2\rangle$;
  - the tensor product of $|v_1\rangle$ and $|v_2\rangle$ can also be written as $|v_1\rangle|v_2\rangle$ instead of $|v_1\rangle \otimes |v_2\rangle$,

  where v, v_1, and v_2 are vectors.

2.2. Language recognition

The language $L \subset \Sigma^*$ recognized by machine $\mathcal{M}$ with (strict) cutpoint $\lambda \in \mathbb{R}$ is defined as

$$L = \{w \in \Sigma^* | f_{\mathcal{M}}(w) > \lambda\}. \tag{1}$$

The language $L \subset \Sigma^*$ recognized by machine $\mathcal{M}$ with nonstrict cutpoint $\lambda \in \mathbb{R}$ is defined [18] as

$$L = \{w \in \Sigma^* | f_{\mathcal{M}}(w) \geq \lambda\}. \tag{2}$$

The language $L \subset \Sigma^*$ is said to be recognized by machine $\mathcal{M}$ with unbounded error if there exists a cutpoint $\lambda \in \mathbb{R}$ such that L is recognized by $\mathcal{M}$ with strict or nonstrict cutpoint λ.

3. Space-bounded Turing machines

The Turing machine (TM) models we use in this paper consist of a read-only input tape with a two-way tape head, a read/write work tape with a two-way tape head, and a finite state control. (The quantum versions also have a finite register that plays a part in the implementation of general quantum operations, and is used to determine whether the computation has halted, and if so, with which decision. For reasons of simplicity, this register is not included in the definition of the probabilistic machines, since its functionality can be emulated by a suitable partition of the set of internal states without any loss of computational power.) Both tapes are assumed to be two-way infinite and indexed by $\mathbb{Z}$.

Let w be an input string. On the input tape, $\tilde{w} = ¢w_1 \ldots w_{|w|}\$$ is placed in the squares indexed by $1, \ldots, |\tilde{w}|$, and all remaining squares contain #. When the computation starts, the internal state is q_1, and both heads are placed on the squares indexed by 1. Additionally, we assume that the input tape head never visits the squares indexed by 0 or $|\tilde{w}| + 1$.

The internal state and the symbols scanned on the input and work tapes determine the transitions of the machine. After each of these transitions, the internal state is updated, the symbol on the work tape is overwritten, and the positions of the input and work tape heads are updated with respect to $\Leftrightarrow$. (In the quantum case, the content of the finite register is overwritten, too.)

A TM is said to be *unidirectional* if the movements of input and work tape heads are fixed for each internal state to be entered in any transition. That is, for a unidirectional TM, we have two functions $D_i : Q \rightarrow \Leftrightarrow$ and $D_w : Q \rightarrow \Leftrightarrow$, determining, respectively, the movements of the input and work tape heads.

A configuration of a TM is the collection of

- the internal state of the machine,
- the position of the input tape head,
- the contents of the work tape, and the position of the work tape head.

$\mathcal{C}^w$, or shortly $\mathcal{C}$, denotes the set of all configurations, which is a finite set in our case of space bounded computations. Let c_i and c_j be two configurations. The probability (or amplitude) of the transition from c_i to c_j is given by the transition function δ if c_i is reachable from c_j in one step, and is zero otherwise. (Note that, in probabilistic and quantum computation, more than one outgoing transition can be defined for a single configuration.) A *configuration matrix* is a square matrix whose rows and columns are indexed by the configurations. The (j, i)th entry of the matrix denotes the value of the transition from c_i to c_j.

We say that [1] a TM $\mathcal{M}$ runs in space s, for s a function of the form $s : \mathbb{N} \rightarrow \mathbb{Z}^+$, if the following holds for each input w: there exist $s(|w|)$ contiguous tape squares on the work tape of $\mathcal{M}$ such that there is zero probability that the work tape head of $\mathcal{M}$ leaves these tape squares at any point in its computation on input w.

By restricting the movement of the input tape head to the set $\{\downarrow, \rightarrow\}$, we obtain a one-way machine.

3.1. Probabilistic Turing machines

A probabilistic Turing machine (PTM) is a 7-tuple[1]

$$\mathcal{P} = (Q, \Sigma, \Gamma, \delta, q_1, Q_a, Q_r), \tag{3}$$

where Q_a and Q_r, disjoint subsets of Q not including q_1, are the collections of accepting and rejecting internal states, respectively. Additionally, $Q_n = Q \setminus \{Q_a \cup Q_r\}$.

The transition function δ is specified so that

$$\delta(q, \sigma, \gamma, q', \gamma') \in \tilde{\mathbb{R}} \tag{4}$$

is the probability that the PTM will change its internal state to q', write γ' on the work tape, and update the positions of the input and work tape heads with respect to $D_i(q')$ and $D_w(q')$, respectively,[2] if it scans σ and γ on the input and work tapes, respectively, when originally in internal state q. $\tilde{\mathbb{R}}$ is the set consisting of $p \in \mathbb{R}$ such that there is a deterministic algorithm that computes p to within 2^{-n} in time polynomial in n. We choose $\tilde{\mathbb{R}} \cap [0, 1]$ as our set of possible transition probabilities, rather than the familiar "coin-flipping" set $\{0, \frac{1}{2}, 1\}$, or the set of rational numbers, since these possibilities are not known to be equivalent from the point of view of computational power under the small space bounds that we consider, and we wish to use the most powerful yet "reasonable" models in our analysis.

For each input string $w \in \Sigma^*$, the transition function defines a unique configuration matrix, A^w, or shortly A. A PTM is *well-formed* (i.e. fulfills the commonsense requirement that the probabilities of alternative transitions always add up to 1) if all columns of A are stochastic vectors. This constraint defines the following *local conditions for PTM well-formedness* that δ must obey: for each $q \in Q$, $\sigma \in \tilde{\Sigma}$, and $\gamma \in \Gamma$,

$$\sum_{q', \gamma'} \delta(q, \sigma, \gamma, q', \gamma') = 1, \tag{5}$$

where $q' \in Q$ and $\gamma' \in \Gamma$. A well-formed PTM can be described relatively easily by specifying δ by presenting, for each $\sigma \in \tilde{\Sigma}$, a (left) stochastic transition matrix A_σ, whose rows and columns are indexed by (state, work tape symbol) pairs, and the entry indexed by $((q', \gamma'), (q, \gamma))$ equals $\delta(q, \sigma, \gamma, q', \gamma')$.

[1] Recall that some notation and terminology which will be used multiple times in this and the following definitions were introduced in Section 2.1.

[2] We define PTMs as unidirectional machines. This causes no loss of computational power, and increases the number of internal states in our machines at most by a factor of 9.

The computation halts and the input is accepted (or rejected) whenever the machine enters an internal state belonging to the set of accepting (or rejecting) states. PrSPACE(s) is the class of the languages that are recognized by a PTM running in space $O(s)$ with unbounded error.

The case of constant space bounds will be given special attention: by removing the work tape of the PTM,[3] we obtain the two-way probabilistic finite state automaton (2PFA), which is formally a 6-tuple

$$\mathcal{P} = (Q, \Sigma, \delta, q_1, Q_a, Q_r). \tag{6}$$

In this case, a well-formed machine can be specified by providing a (left) stochastic matrix A_σ, whose rows and columns are indexed only by internal states, for each $\sigma \in \tilde{\Sigma}$.

In both probabilistic and quantum finite automata [4,19,20], the transition probabilities are traditionally allowed to be uncomputable numbers, and therefore the classes of recognized languages include undecidable ones [19]. TMs, however, are restricted to use computable transition probabilities, as seen in the definition above. Note that the simulation results in this paper do not change when we disallow finite automata to have uncomputable numbers as transition probabilities, since none of our constructions involve such numbers.

If we restrict the range of D_i in 2PFAs with $\{\rightarrow\}$, we obtain the real-time probabilistic finite automaton (RT-PFA) model. A RT-PFA can scan the input only once. Traditionally, RT-PFAs are defined to be able to decide on acceptance or rejection only after the last symbol is read, and just specifying the set of accepting states in their description is therefore sufficient, yielding a 5-tuple

$$\mathcal{P} = (Q, \Sigma, \{A_\sigma | \sigma \in \tilde{\Sigma}\}, q_1, Q_a). \tag{7}$$

The computation of a RT-PFA can be traced by a stochastic state vector, say v, such that $v[i]$ corresponds to state q_i. For a given input string $w \in \Sigma^*$, ($\tilde{w} = ¢w\$$ is placed on the tape)

$$v_i = A_{\tilde{w}_i} v_{i-1}, \tag{8}$$

where $1 \leq i \leq |\tilde{w}|$; $\tilde{w}_i$ is the ith symbol of $\tilde{w}$; v_0 is the initial state vector whose first entry is equal to 1. The transition matrices of a RT-PFA can be extended for any string as

$$A_{w\sigma} = A_\sigma A_w, \tag{9}$$

where $w \in \Sigma^*$, $\sigma \in \tilde{\Sigma}$, and $A_\varepsilon = I$. The probability that w will be accepted by RT-PFA $\mathcal{P}$ is

$$f_{\mathcal{P}}(w) = \sum_{q_i \in Q_a} (A_{\tilde{w}} v_0)[i] = \sum_{q_i \in Q_a} v_{|\tilde{w}|}[i]. \tag{10}$$

A generalization of the RT-PFA is the generalized finite automaton (GFA), which is formally a 5-tuple

$$\mathcal{G} = (Q, \Sigma, \{A_\sigma | \sigma \in \Sigma\}, v_0, f), \tag{11}$$

where

1. A_σ's are $|Q| \times |Q|$-dimensional real valued transition matrices.
2. v_0 and f are real valued *initial* (column) and *final* (row) vectors, respectively.

Similar to what we had for RT-PFAs, the transition matrices of a GFA can be extended for any string. For a given input string, $w \in \Sigma^*$, the acceptance value associated by GFA $\mathcal{G}$ to string w is

$$f_{\mathcal{G}}(w) = f A_{w_{|w|}} \cdots A_{w_1} v_0 = f A_w v_0. \tag{12}$$

RT-PFAs, GFAs [21], and 2PFAs [22] recognize the same class of languages with cutpoint. This is the class of *stochastic languages*, denoted by S. The class of languages recognized by these machines with nonstrict cutpoint is denoted by coS. The class of languages recognized by RT-PFAs, GFAs, and 2PFAs with unbounded error is therefore S $\cup$ coS, and is denoted by uS. Note that PrSPACE(1) $\subsetneq$ uS, since uS contains undecidable languages.

[3] One only needs the single "direction" function D_i in this case.

3.2. Quantum Turing machines

We define a quantum Turing machine (QTM) $\mathcal{M}$ to be a 7-tuple

$$M = (Q, \Sigma, \Gamma, \Omega, \delta, q_1, \Delta), \tag{13}$$

which is distinguished from the PTM by the presence of the items Ω, the finite register alphabet, containing the special initial symbol ω_1, and $\Delta = \{\tau_1, \ldots, \tau_k\}$, the set of possible outcomes associated with the measurements of the finite register. Ω is partitioned into $|\Delta| = k$ subsets $\Omega_{\tau_1}, \ldots, \Omega_{\tau_k}$.

In accordance with quantum theory, a QTM can be in a superposition of more than one configuration at the same time. The "weight" of each configuration in such a superposition is called its amplitude. Unlike the case with PTMs, these amplitudes are not restricted to being positive real numbers, and that is what gives quantum computers their interesting features. A superposition of configurations

$$|\psi\rangle = \alpha_1|c_1\rangle + \alpha_2|c_2\rangle + \cdots + \alpha_n|c_n\rangle \tag{14}$$

can be represented by a column vector $|\psi\rangle$ with a row for each possible configuration, where the ith row contains the amplitude of the corresponding configuration in $|\psi\rangle$.

If our knowledge that the quantum system under consideration is in superposition $|\psi\rangle$ is certain, then $|\psi\rangle$ is called a *pure state*, and the vector notation described above is a suitable way of manipulating this information. However, in some cases (e.g. during classical probabilistic computation), we only know that the system is in state $|\psi_l\rangle$ with probability p_l for an ensemble of pure states $\{(p_l, |\psi_l\rangle)\}$,where $\sum_l p_l = 1$. A convenient representation tool for describing quantum systems in such *mixed states* is the density matrix. The *density matrix*[4] representation of $\{(p_l, |\psi_l\rangle)|1 \leq l \leq M < \infty\}$ is

$$\rho = \sum_l p_l|\psi_l\rangle\langle\psi_l|. \tag{15}$$

We will use both these representations for quantum states in this paper. We refer the reader to [23] for further details.

The initial density matrix of the QTM is represented by $\rho_0 = |c_1\rangle\langle c_1|$, where c_1 is the initial configuration corresponding to the given input string.

The transition function of a QTM is specified so that

$$\delta(q, \sigma, \gamma, q', d_i, \gamma', d_w, \omega) \in \tilde{\mathbb{C}} \tag{16}$$

is the amplitude with which the QTM will change its internal state to q', write γ' on the work tape and ω in the finite register, and update the positions of the input and work tape heads with respect to d_i and d_w, respectively, where $d_i, d_w \in \triangleleft\!\triangleright$, if it scans σ and γ on the input and work tapes, respectively, when originally in internal state q. $\tilde{\mathbb{C}}$ [24] is the set of complex numbers whose real and imaginary parts are in $\tilde{\mathbb{R}}$.

After each transition, the finite register is measured [23] as described by the set of operators

$$P = \left\{ P_\tau | P_\tau = \sum_{\omega \in \Omega_\tau} |\omega\rangle\langle\omega|, \tau \in \Delta \right\}. \tag{17}$$

In its standard usage, Δ is the set $\{a, n, r\}$, and the following actions are associated with the measurement outcomes:

- "n": the computation continues;
- "a": the computation halts, and the input is accepted;
- "r": the computation halts, and the input is rejected.

The finite register is reinitialized to ω_1, irreversibly erasing its previous content, before the next transition of the machine.

Since we do not consider the register content as part of the configuration, the register can be seen as the "environment" interacting with the "principal system" that is the rest of the QTM [23]. The transition function δ therefore induces a set of configuration transition matrices, $\{E_\omega|\omega \in \Omega\}$, where the (i, j)th entry of E_ω, the amplitude of the transition from c_j to c_i by writing $\omega \in \Omega$ on the register, is defined by δ whenever c_j is reachable from c_i in one step, and is zero otherwise. The $\{E_\omega|\omega \in \Omega\}$ form an operator $\mathcal{E}$, with operation elements $\mathcal{E}_{\tau_1} \cup \mathcal{E}_{\tau_2} \cup \cdots \cup \mathcal{E}_{\tau_k}$, where for each $\tau \in \Delta$, $\mathcal{E}_\tau = \{E_\omega|\omega \in \Omega_\tau\}$.

[4] Density matrices are Hermitian positive semidefinite matrices of trace 1.

$$
\begin{array}{c|c|}
 & c_1 \quad c_2 \quad \ldots \quad c_{|\mathcal{C}|} \\
\hline
\begin{array}{c} c_1 \\ c_2 \\ \vdots \\ c_{|\mathcal{C}|} \end{array} & E_{\omega_1} \\
\hline
\begin{array}{c} c_1 \\ c_2 \\ \vdots \\ c_{|\mathcal{C}|} \end{array} & E_{\omega_2} \\
\hline
\begin{array}{c} c_1 \\ c_2 \\ \vdots \\ c_{|\mathcal{C}|} \end{array} & \vdots \\
\hline
\begin{array}{c} c_1 \\ c_2 \\ \vdots \\ c_{|\mathcal{C}|} \end{array} & E_{\omega_{|\Omega|}} \\
\hline
\end{array}
$$

Fig. 1. Matrix E.

According to the modern understanding of quantum computation [25], a QTM is said to be *well-formed* [5] if $\mathcal{E}$ is a superoperator (selective quantum operator), i.e.

$$\sum_{\omega \in \Omega} E_{\omega}^{\dagger} E_{\omega} = I. \tag{18}$$

$\mathcal{E}$ can be represented by a $|\mathcal{C}||\Omega| \times |\mathcal{C}|$-dimensional matrix E (Fig. 1) by concatenating each E_{ω} one under the other, where $\omega \in \Omega$. It can be verified that $\mathcal{E}$ is a superoperator if and only if the columns of E form an orthonormal set.

Let c_{j_1} and c_{j_2} be two configurations with corresponding columns v_{j_1} and v_{j_2} in E. For an orthonormal set to be formed, we must have

$$v_{j_1}^{\dagger} \cdot v_{j_2} = \begin{cases} 1 & j_1 = j_2 \\ 0 & j_1 \neq j_2 \end{cases} \tag{19}$$

for all such pairs. This constraint imposes some easily checkable restrictions on δ. The (quite long) list of these local conditions for QTM well-formedness can be found in [26].

PrQSPACE(s) is the class of languages that are recognized by QTMs running in space $O(s)$ with unbounded error. (Note that this complexity class has been defined and used by Watrous in references [2,3]. As we will demonstrate shortly, our QTM model is at least as powerful as the models used in those papers, and it may well be strictly more powerful than them. Since the aim is to understand the full power of space-bounded quantum computation, we suggest it would make sense to adopt our definition of PrQSPACE as the standard.)

It is a well-established fact [27] that any quantum computational model defined using superoperators can efficiently simulate its classical counterpart, and so PrSPACE(s) $\subseteq$ PrQSPACE(s) for all s. Some early models used in the study of space-bounded quantum computation, which do not make full use of the capabilities allowed by quantum mechanics, can fail to achieve some tasks that are possible for the corresponding classical machines [4,17].

The two-way quantum finite automaton (2QFA) is obtained by removing the work tape of the QTM:

$$\mathcal{M} = (Q, \Sigma, \Omega, \delta, q_1, \Delta). \tag{20}$$

The transition function of a 2QFA is therefore specified so that

$$\delta(q, \sigma, q', d_i, \omega) \in \mathbb{C} \tag{21}$$

[5] We also refer the reader to [24] for a detailed discussion of the well-formedness of QTMs that evolve unitarily.

is the amplitude with which the machine enters state q', writes ω on the register, and updates the position of the input tape with respect to $d_i \in \Leftrightarrow$, if it reads σ on the input tape when originally in state q. See Appendix A for a list of easily checkable local conditions for well-formedness of 2QFAs.

In the remainder of this section, we will examine some specializations of the QTM model that have appeared in the literature.

3.2.1. QTMs with classical heads

Although our definition of space usage as the number of work tape squares used during the computation is standard in the study of small as well as large space bounds [28–30], some researchers prefer to utilize QTM models where the tape head locations are classical (i.e. the heads do not enter quantum superpositions) to avoid the possibility of using quantum resources that increase with input size for the implementation of the heads. For details of this specialization of our model, which we call the QTM with classical heads (CQTM), see Appendix B, which also includes a demonstration of the fact that all quantum machines can simulate their probabilistic counterparts easily.

Watrous' QTM model in [3], which we call Wa03-QTM for ease of reference, is a CQTM variant that has an additional classical work tape and classical internal states. Every Wa03-QTM can be simulated exactly (i.e. preserving the same acceptance probability for every input) by CQTMs with only some time overhead.[6] Note that Wa03-QTMs allow only algebraic transition amplitudes by definition.

Let us consider real-time versions of 2QFAs, whose tape heads are forced by definition to have classical locations [8]. If the quantum machine model used is sufficiently general, then the intermediate measurements can be postponed easily to the end of the algorithm in real-time computation. That final measurement can be performed on the set of internal states, rather than the finite register. Therefore, as with RT-PFAs, we specify a subset of the internal states of the machine as the collection of accepting states, denoted Q_a.

A real-time quantum finite automaton (RT-QFA) [13] is a 5-tuple

$$\mathcal{M} = (Q, \Sigma, \{\mathcal{E}_\sigma | \sigma \in \tilde{\Sigma}\}, q_1, Q_a), \tag{22}$$

where each $\mathcal{E}_\sigma$ is an operator having elements $\{E_{\sigma,1}, \ldots, E_{\sigma,k}\}$ for some $k \in \mathbb{Z}^+$ satisfying

$$\sum_{i=1}^{k} E_{\sigma,i}^{\dagger} E_{\sigma,i} = I. \tag{23}$$

Additionally, we define the projector

$$P_a = \sum_{q \in Q_a} |q\rangle\langle q| \tag{24}$$

in order to check for acceptance. For any given input string $w \in \Sigma^*$, $\tilde{w}$ is placed on the tape, and the computation can be traced by density matrices

$$\rho_j = \mathcal{E}_{\tilde{w}_j}(\rho_{j-1}) = \sum_{i=1}^{k} E_{\tilde{w}_j,i}\,\rho_{j-1} E_{\tilde{w}_j,i}^{\dagger}, \tag{25}$$

where $1 \leq j \leq |\tilde{w}|$, and $\rho_0 = |q_1\rangle\langle q_1|$ is the initial density matrix. This is how density matrices evolve according to superoperators [23]. The transition operators can be extended easily for any string as

$$\mathcal{E}_{w\sigma} = \mathcal{E}_\sigma \circ \mathcal{E}_w, \tag{26}$$

where $w \in \Sigma^*$ and $\mathcal{E}_\varepsilon = I$. Note that, if $\mathcal{E} = \{E_i | 1 \leq i \leq k\}$ and $\mathcal{E}' = \{E'_j | 1 \leq j \leq k'\}$, then

$$\mathcal{E}' \circ \mathcal{E} = \{E'_j E_i | 1 \leq i \leq k, 1 \leq j \leq k'\}. \tag{27}$$

The probability that RT-QFA $\mathcal{M}$ will accept w is

$$f_{\mathcal{M}}(w) = tr(P_a \mathcal{E}_{\tilde{w}}(\rho_0)) = tr(P_a \rho_{|\tilde{w}|}). \tag{28}$$

The class of languages recognized by RT-QFAs with cutpoint is denoted by QAL. The class of languages recognized by these machines with nonstrict cutpoint is denoted by coQAL. QAL ∪ coQAL is denoted by uQAL.

[6] We omit the proof here, but it is not hard to show how to simulate the classical components of a Wa03-QTM within a CQTM.

Let A, B, and C be $n \times n$ dimensional matrices. vec is a linear mapping from $n \times n$ matrices to n^2 dimensional (column) vectors defined as

$$vec(A)[(i-1)n+j] = A[i,j], \tag{29}$$

where $1 \leq i, j \leq N$. One can verify the following properties:

$$vec(ABC) = (A \otimes C^T)vec(B) \tag{30}$$

and

$$tr(A^T B) = vec(A)^T vec(B). \tag{31}$$

Fig. 2. The definition and properties of *vec* (see [3, p. 73]).

Lemma 3.1. *For any RT-QFA $\mathcal{M}$ with n internal states, there exists a GFA $\mathcal{G}$ with n^2 internal states such that $f_{\mathcal{M}}(w) = f_{\mathcal{G}}(w)$ for all $w \in \Sigma^*$.*

Proof. Let $\mathcal{M} = (Q_1, \Sigma, \{\mathcal{E}_\sigma | \sigma \in \tilde{\Sigma}\}, q_1, Q_a)$ be the RT-QFA with n internal states, and let each $\mathcal{E}_\sigma$ have k elements, without loss of generality. We will construct GFA $\mathcal{G} = (Q_2, \Sigma, \{A_\sigma | \sigma \in \Sigma\}, v_0, f)$ with n^2 internal states. We start by building an intermediate GFA $\mathcal{G}' = (Q_3, \Sigma, \{A'_\sigma | \sigma \in \Sigma\}, v'_0, f')$ with the required simulation property but with $2n^2$ states. We will use the mapping *vec* described in Fig. 2 in order to linearize the computation of $\mathcal{M}$, so that it can be traced by $\mathcal{G}'$.

We define

$$v''_0 = vec(\rho_1), \tag{32}$$

where

$$\rho_1 = \mathcal{E}_{¢}(\rho_0) = \sum_{i=1}^{k} E_{¢,i}\rho_0 E_{¢,i}^\dagger. \tag{33}$$

For each $\sigma \in \Sigma$, we define

$$A''_\sigma = \sum_{i=1}^{k} E_{\sigma,i} \otimes E_{\sigma,i}^*, \tag{34}$$

and so we obtain (by Eqs. 25 and 30)

$$vec(\mathcal{E}_\sigma(\rho)) = A''_\sigma vec(\rho) \tag{35}$$

for any density matrix ρ. Finally, we define

$$f'' = vec(P_a)^T \sum_{i=1}^{k} E_{\$,i} \otimes E_{\$,i}^*. \tag{36}$$

It can be verified by using Eq. 31 that for any input string $w \in \Sigma^*$,

$$f'' A''_{w_{|w|}} \cdots A''_{w_1} v''_0 = tr(P_a \mathcal{E}_\$ \circ \mathcal{E}_w \circ \mathcal{E}_{¢}(\rho_0)) = f_{\mathcal{M}}(w). \tag{37}$$

The complex entries of v''_0, $\{A''_\sigma | \sigma \in \Sigma\}$, and f'' can be replaced [17] with 2×2 dimensional real matrices,[7] and so we obtain the equations

$$\begin{pmatrix} f_{\mathcal{M}}(w) & 0 \\ 0 & f_{\mathcal{M}}(w) \end{pmatrix} = f''' A'''_{w_{|w|}} \cdots A'''_{w_1} v'''_0, \tag{38}$$

where the terms with triple primes are obtained from the corresponding terms with double primes. We finish the construction of $\mathcal{G}'$ by stating that

[7] $a + bi$ is replaced with $\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$.

1. v'_0 is the first column of v'''_0,
2. A'_σ is equal to A'''_σ, for each $\sigma \in \Sigma$, and
3. f' is the first row of f'''.

We refer the reader to [17,31,32], that present similar constructions for other types of real-time QFAs. The remainder of this proof is an improvement over these constructions regarding the number of states, and was kindly suggested to us by one of the anonymous referees of this paper, to whom we are indebted.

Since density matrices are Hermitian, all entries on the main diagonal are real, and the entries on the opposite sides of the diagonal are complex conjugates of each other, meaning that one actually needs only n^2 distinct real numbers to represent the $n \times n$ density matrices of $\mathcal{M}$. So the information in the vector v'_0 in the definition of $\mathcal{G}'$ can in fact fit in an n^2-dimensional vector. To perform conversions between these two representations, we can define two linear operators, denoted L and L', such that

- L, an $n^2 \times 2n^2$-dimensional matrix containing entries from the set $\{-1, 0, 1\}$, transforms $2n^2$-dimensional vectors in the format of machine $\mathcal{G}'$ to equivalent n^2-dimensional vectors, and
- L', a $2n^2 \times n^2$-dimensional matrix, performs the reverse transformation.

Hence, the state-efficient GFA $\mathcal{G}$ is constructed by setting

1. $v_0 = Lv'_0$,
2. $A_\sigma = LA'_\sigma L'$, for each $\sigma \in \Sigma$, and
3. $f = f'L'$. $\square$

Corollary 3.1. *QAL = S.*

We therefore have that real-time unbounded-error probabilistic and quantum finite automata are equivalent in power. We will show in Sections 4 and 5 that this equivalence does not carry over to the two-way case.

3.2.2. QTMs with restricted measurements

In another specialization of the QTM model, the *QTM with restricted measurements*, the machine is unidirectional, the heads can enter quantum superpositions, $\Delta = \{n, a, r\}$, and $|\Omega_n| = |\Omega_a| = |\Omega_r| = 1$. The first family of QTMs that was formulated for the analysis of space complexity issues [1,2], which we call the Wa98-QTM, corresponds to such a model, with the added restriction that the transition amplitudes are only allowed to be rational numbers. The finite automaton versions of QTMs with restricted measurements[8] are known as Kondacs–Watrous quantum finite automata, and abbreviated as 2KWQFAs, 1KWQFAs, or RT-KWQFAs, depending on the set of allowed directions of movement for the input head. These are pure state models, since the nonhalting part of the computation is always represented by a single quantum state. Therefore, configuration or state vectors, rather than the density matrix formalism, can be used in order to trace the computation easily. To be consistent with the literature on 2KWQFAs, we specialize the 2QFA model by the following process:

1. The finite register does not need to be refreshed, since the computation continues if and only if the initial symbol is observed.
2. In fact, 2KWQFAs do not need to have the finite register at all, instead, similarly to 2PFAs, the set of internal states of the 2KWQFA is partitioned to sets of nonhalting, accepting, and rejecting states, denoted Q_n, Q_a, and Q_r, respectively, which can be obtained easily by taking the tensor product of the internal states of the 2QFA and the set $\{n, a, r\}$.
3. A configuration is designated as nonhalting (resp. accepting or rejecting), if its internal state is a member of Q_n (resp. Q_a or Q_r). Nonhalting (resp. accepting or rejecting) configurations form the set $\mathcal{C}_n$ (resp. $\mathcal{C}_a$ or $\mathcal{C}_r$) (for a given input string).
4. The evolution of the configuration sets can be represented by a unitary matrix.
5. The measurement is done on the configuration set with projectors P_n, P_a, and P_r, defined as

$$P_\tau = \sum_{c \in \mathcal{C}^w_\tau} |c\rangle\langle c| \tag{39}$$

for a given input string $w \in \Sigma^*$, where $\tau \in \{n, a, r\}$ and the standard actions are associated with the outcomes "n", "a", and "r".

[8] These models, which also allow unrestricted transition amplitudes by the convention in automata theory, are introduced in the paper written by Kondacs and Watrous [4].

Formally, a 2KWQFA is a 6-tuple

$$\mathcal{M} = \{Q, \Sigma, \delta, q_1, Q_a, Q_r\}, \tag{40}$$

where $Q_n = Q \setminus \{Q_a \cup Q_r\}$ and $q_1 \in Q_n$. δ induces a unitary matrix U_σ, whose rows and columns are indexed by internal states for each input symbol σ. Since all 2KWQFAs are unidirectional, we will use the notations $\overleftarrow{q}$, $\downarrow q$, and $\overrightarrow{q}$ for internal state q in order to represent the value of $D_i(q)$ as $\leftarrow$, $\downarrow$, and $\rightarrow$, respectively.

A RT-KWQFA is a 6-tuple

$$\mathcal{M} = \{Q, \Sigma, \{U_\sigma | \sigma \in \tilde{\Sigma}\}, q_1, Q_a, Q_r\}, \tag{41}$$

where $\{U_\sigma | \sigma \in \tilde{\Sigma}\}$ are unitary transition matrices. In contrast to the other kinds of real-time finite automata, a RT-KWQFA is measured at each step during computation after the unitary transformation is applied. The projectors are defined as

$$P_\tau = \sum_{q \in Q_\tau} |q\rangle\langle q|, \tag{42}$$

where $\tau \in \Delta$. The nonhalting portion of the computation of a RT-KWQFA can be traced by a state vector, say $|u\rangle$, such that $\langle i|u\rangle$ corresponds to state q_i. The computation begins with $|u_0\rangle = |q_1\rangle$. For a given input string $w \in \Sigma^*$, at step j $(1 \leq j \leq |\tilde{w}|)$:

$$|u_j\rangle = P_n U_{\tilde{w}_j} |u_{j-1}\rangle, \tag{43}$$

the input is accepted with probability

$$||P_a U_{\tilde{w}_j} |u_{j-1}\rangle||^2, \tag{44}$$

and rejected with probability

$$||P_r U_{\tilde{w}_j} |u_{j-1}\rangle||^2. \tag{45}$$

The overall acceptance and rejection probabilities are accumulated by summing up these values at each step. Note that, the state vector representing the nonhalting portion is not normalized in the description given above.

Brodsky and Pippenger [10], who studied various properties of some early models of quantum finite automata, defined the class of languages recognized by RT-KWQFAs with unbounded error, denoted *UMM*, in a way that is slightly different than our approach in this paper: $L \in UMM$ if and only if there exists a RT-KWQFA $\mathcal{M}$ such that

- $f_\mathcal{M}(w) > \lambda$ when $w \in L$ and
- $f_\mathcal{M}(w) < \lambda$ when $w \notin L$,

for some $\lambda \in [0, 1]$.

For descriptions of several other QTM variants, we refer the reader to [33,34].

4. Probabilistic vs. quantum computation with sublogarithmic space

Watrous compared the unbounded-error probabilistic space complexity classes (PrSPACE(s)) with the corresponding classes for both Wa98-QTMs [1,2] and Wa03-QTMs [3] for space bounds $s = \Omega(\log n)$, establishing the identity of the associated quantum space complexity classes with each other, and also with the corresponding probabilistic ones. The case of $s = o(\log n)$ was left as an open question [2]. In this section, we provide an answer to that question.

We already know that QTMs allowing superoperators are at least as powerful as PTMs for any common space bound. We will now exhibit a 1KWQFA which performs a task that is impossible for PTMs with small space bounds.

Consider the nonstochastic context-free language [35]

$$L_{NH} = \{a^x b a^{y_1} b a^{y_2} b \cdots a^{y_t} b | x, t, y_1, \cdots, y_t \in \mathbb{Z}^+ \text{ and } \exists k\ (1 \leq k \leq t), x = \textstyle\sum_{i=1}^{k} y_i\}$$

over the alphabet $\{a, b\}$. Freivalds and Karpinski [30] have proven the following facts about L_{NH}:

Fact 1. *No PTM using space $o(\log\log n)$ can recognize L_{NH} with unbounded error.*

Fact 2. *No 1PTM using space $o(\log n)$ can recognize L_{NH} with unbounded error.*

Stages	$U_{¢}, U_a$	$U_{\$}$
	$U_{¢}\vert\overrightarrow{q_0}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{q_1}\rangle + \frac{1}{\sqrt{2}}\vert\overrightarrow{p_1}\rangle$	
I (path_1)	$U_a\vert\overrightarrow{q_1}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{q_2}\rangle + \frac{1}{2}\vert{\downarrow}A_1\rangle + \frac{1}{2}\vert{\downarrow}R_1\rangle$ $U_a\vert\overrightarrow{q_2}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{q_2}\rangle - \frac{1}{2}\vert{\downarrow}A_1\rangle - \frac{1}{2}\vert{\downarrow}R_1\rangle$	$U_{\$}\vert\overrightarrow{q_1}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_1\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_1\rangle$ $U_{\$}\vert\overrightarrow{q_2}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_2\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_2\rangle$ $U_{\$}\vert\overrightarrow{q_3}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_3\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_3\rangle$
I (path_2)	$U_a\vert\overrightarrow{p_1}\rangle = \vert{\downarrow}w_1\rangle$ $U_a\vert{\downarrow}w_1\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{p_2}\rangle + \frac{1}{2}\vert{\downarrow}A_2\rangle + \frac{1}{2}\vert{\downarrow}R_2\rangle$ $U_a\vert\overrightarrow{p_2}\rangle = \vert{\downarrow}w_2\rangle$ $U_a\vert{\downarrow}w_2\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{p_2}\rangle - \frac{1}{2}\vert{\downarrow}A_2\rangle - \frac{1}{2}\vert{\downarrow}R_2\rangle$	$U_{\$}\vert\overrightarrow{p_1}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_4\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_4\rangle$ $U_{\$}\vert\overrightarrow{p_2}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_5\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_5\rangle$ $U_{\$}\vert\overrightarrow{p_3}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_6\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_6\rangle$
II (path_1)	$U_a\vert\overrightarrow{q_3}\rangle = \vert{\downarrow}w_3\rangle$ $U_a\vert{\downarrow}w_3\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{q_4}\rangle + \frac{1}{2}\vert{\downarrow}A_3\rangle + \frac{1}{2}\vert{\downarrow}R_3\rangle$ $U_a\vert\overrightarrow{q_4}\rangle = \vert{\downarrow}w_4\rangle$ $U_a\vert{\downarrow}w_4\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{q_4}\rangle - \frac{1}{2}\vert{\downarrow}A_3\rangle - \frac{1}{2}\vert{\downarrow}R_3\rangle$	$U_{\$}\vert\overrightarrow{q_4}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_7\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_7\rangle$ $U_{\$}\vert\overrightarrow{q_5}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_8\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_8\rangle$
II (path_2)	$U_a\vert\overrightarrow{p_3}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{p_4}\rangle + \frac{1}{2}\vert{\downarrow}A_4\rangle + \frac{1}{2}\vert{\downarrow}R_4\rangle$ $U_a\vert\overrightarrow{p_4}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{p_4}\rangle - \frac{1}{2}\vert{\downarrow}A_4\rangle - \frac{1}{2}\vert{\downarrow}R_4\rangle$	$U_{\$}\vert\overrightarrow{p_4}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_9\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_9\rangle$ $U_{\$}\vert\overrightarrow{p_5}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_{10}\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_{10}\rangle$
III (path_1)	$U_a\vert\overrightarrow{q_5}\rangle = \vert{\downarrow}w_5\rangle$ $U_a\vert{\downarrow}w_5\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{q_6}\rangle + \frac{1}{2}\vert{\downarrow}A_5\rangle + \frac{1}{2}\vert{\downarrow}R_5\rangle$ $U_a\vert\overrightarrow{q_6}\rangle = \vert{\downarrow}w_6\rangle$ $U_a\vert{\downarrow}w_6\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{q_6}\rangle - \frac{1}{2}\vert{\downarrow}A_5\rangle - \frac{1}{2}\vert{\downarrow}R_5\rangle$	$U_{\$}\vert\overrightarrow{q_6}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_{11}\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_{11}\rangle$
III (path_2)	$U_a\vert\overrightarrow{p_5}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{p_6}\rangle + \frac{1}{2}\vert{\downarrow}A_6\rangle + \frac{1}{2}\vert{\downarrow}R_6\rangle$ $U_a\vert\overrightarrow{p_6}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{p_6}\rangle - \frac{1}{2}\vert{\downarrow}A_6\rangle - \frac{1}{2}\vert{\downarrow}R_6\rangle$	$U_{\$}\vert\overrightarrow{p_6}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_{12}\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_{12}\rangle$
III ($\mathrm{path}_{\mathrm{accept}}$)	$U_a\vert\overrightarrow{a_1}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{a_2}\rangle + \frac{1}{2}\vert{\downarrow}A_7\rangle + \frac{1}{2}\vert{\downarrow}R_7\rangle$ $U_a\vert\overrightarrow{a_2}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{a_2}\rangle - \frac{1}{2}\vert{\downarrow}A_7\rangle - \frac{1}{2}\vert{\downarrow}R_7\rangle$ $U_a\vert\overrightarrow{a_3}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{a_4}\rangle + \frac{1}{2}\vert{\downarrow}A_8\rangle + \frac{1}{2}\vert{\downarrow}R_8\rangle$ $U_a\vert\overrightarrow{a_4}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{a_4}\rangle - \frac{1}{2}\vert{\downarrow}A_8\rangle - \frac{1}{2}\vert{\downarrow}R_8\rangle$	$U_{\$}\vert\overrightarrow{a_1}\rangle = \vert{\downarrow}A_{17}\rangle$ $U_{\$}\vert\overrightarrow{a_3}\rangle = \vert{\downarrow}A_{18}\rangle$ $U_{\$}\vert\overrightarrow{a_2}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_{13}\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_{13}\rangle$ $U_{\$}\vert\overrightarrow{a_4}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_{14}\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_{14}\rangle$
III ($\mathrm{path}_{\mathrm{reject}}$)	$U_a\vert\overrightarrow{r_1}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{r_2}\rangle + \frac{1}{2}\vert{\downarrow}A_9\rangle + \frac{1}{2}\vert{\downarrow}R_9\rangle$ $U_a\vert\overrightarrow{r_2}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{r_2}\rangle - \frac{1}{2}\vert{\downarrow}A_9\rangle - \frac{1}{2}\vert{\downarrow}R_9\rangle$ $U_a\vert\overrightarrow{r_3}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{r_4}\rangle + \frac{1}{2}\vert{\downarrow}A_{10}\rangle + \frac{1}{2}\vert{\downarrow}R_{10}\rangle$ $U_a\vert\overrightarrow{r_4}\rangle = \frac{1}{\sqrt{2}}\vert\overrightarrow{r_4}\rangle - \frac{1}{2}\vert{\downarrow}A_{10}\rangle - \frac{1}{2}\vert{\downarrow}R_{10}\rangle$	$U_{\$}\vert\overrightarrow{r_1}\rangle = \vert{\downarrow}R_{17}\rangle$ $U_{\$}\vert\overrightarrow{r_3}\rangle = \vert{\downarrow}R_{18}\rangle$ $U_{\$}\vert\overrightarrow{r_2}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_{15}\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_{15}\rangle$ $U_{\$}\vert\overrightarrow{r_4}\rangle = \frac{1}{\sqrt{2}}\vert{\downarrow}A_{16}\rangle + \frac{1}{\sqrt{2}}\vert{\downarrow}R_{16}\rangle$

Fig. 3. Specification of the transition function of the 1KWQFA for L_{NH} (part 1).

There exists a one-way *deterministic* TM that recognizes L_{NH} within the optimal space bound $O(\log n)$ [30]. No (two-way) PTM which recognizes L_{NH} using $o(\log n)$ space is known as of the time of writing.

Theorem 4.1. *There exists a 1KWQFA that recognizes L_{NH} with unbounded error.*

Proof. Consider the 1KWQFA $\mathcal{M} = (Q, \Sigma, \delta, q_0, Q_a, Q_r)$, where $\Sigma = \{a, b\}$, and the state sets are as follows:

$$
\begin{aligned}
Q_n &= \{\overrightarrow{q_0}\} \cup \{\overrightarrow{q_i}\,|1 \le i \le 6\} \cup \{\overrightarrow{p_i}\,|1 \le i \le 6\} \cup \{\overrightarrow{a_i}\,|1 \le i \le 4\} \\
&\quad \cup \{\overrightarrow{r_i}\,|1 \le i \le 4\} \cup \{\downarrow w_i|1 \le i \le 6\}, \\
Q_a &= \{\downarrow A_i|1 \le i \le 18\}, \quad Q_r = \{\downarrow R_i|1 \le i \le 18\}.
\end{aligned}
$$

Let each U_σ induced by δ act as indicated in Figs. 3 and 4, and extend each to be unitary.

Machine $\mathcal{M}$ starts computation on symbol ¢ by branching into two paths, $\mathtt{path}_1$ and $\mathtt{path}_2$, with equal probability. Each path and their subpaths, to be described later, check whether the input is of the form $(aa^*b)(aa^*b)(aa^*b)^*$. The different stages of the program indicated in Figs. 3 and 4 correspond to the subtasks of this regular expression check. Stage I ends successfully if the input begins with (aa^*b). Stage II checks the second (aa^*b). Finally, Stage III controls whether the input ends with $(aa^*b)^*$.

The reader will note that many transitions in the machine are of the form

$$U_\sigma|q_i\rangle = |\psi\rangle + \alpha|A_k\rangle + \alpha|R_k\rangle,$$

where $|\psi\rangle$ is a superposition of configurations such that $\langle\psi|\psi\rangle = 1 - 2\alpha^2$, $A_k \in Q_a$, $R_k \in Q_r$. The equal-probability transitions to the "twin halting states" A_k and R_k are included to ensure that the matrices are unitary, without upsetting the "accept/reject balance" until a final decision about the membership of the input in L_{NH} is reached. If the regular expression

Stages	U_b
I (path$_1$)	$U_b\|\overrightarrow{q_1}\rangle = \frac{1}{\sqrt{2}}\|{\downarrow}A_1\rangle + \frac{1}{\sqrt{2}}\|{\downarrow}R_1\rangle$ $U_b\|\overrightarrow{q_2}\rangle = \|\overrightarrow{q_3}\rangle$ $U_b\|\overrightarrow{q_3}\rangle = \frac{1}{\sqrt{2}}\|{\downarrow}A_2\rangle + \frac{1}{\sqrt{2}}\|{\downarrow}R_2\rangle$
I (path$_2$)	$U_b\|\overrightarrow{p_1}\rangle = \frac{1}{\sqrt{2}}\|{\downarrow}A_3\rangle + \frac{1}{\sqrt{2}}\|{\downarrow}R_3\rangle$ $U_b\|\overrightarrow{p_2}\rangle = \|\overrightarrow{p_3}\rangle$ $U_b\|\overrightarrow{p_3}\rangle = \frac{1}{\sqrt{2}}\|{\downarrow}A_4\rangle + \frac{1}{\sqrt{2}}\|{\downarrow}R_4\rangle$
II (path$_1$)	$U_b\|\overrightarrow{q_4}\rangle = \frac{1}{2}\|\overrightarrow{q_5}\rangle + \frac{1}{2\sqrt{2}}\|\overrightarrow{a_1}\rangle + \frac{1}{2\sqrt{2}}\|\overrightarrow{r_1}\rangle + \frac{1}{2}\|{\downarrow}A_{11}\rangle + \frac{1}{2}\|{\downarrow}R_{11}\rangle$ $U_b\|\overrightarrow{q_5}\rangle = \frac{1}{\sqrt{2}}\|{\downarrow}A_5\rangle + \frac{1}{\sqrt{2}}\|{\downarrow}R_5\rangle$
II (path$_2$)	$U_b\|\overrightarrow{p_4}\rangle = \frac{1}{2}\|\overrightarrow{p_5}\rangle + \frac{1}{2\sqrt{2}}\|\overrightarrow{a_1}\rangle - \frac{1}{2\sqrt{2}}\|\overrightarrow{r_1}\rangle + \frac{1}{2}\|{\downarrow}A_{12}\rangle + \frac{1}{2}\|{\downarrow}R_{12}\rangle$ $U_b\|\overrightarrow{p_5}\rangle = \frac{1}{\sqrt{2}}\|{\downarrow}A_6\rangle + \frac{1}{\sqrt{2}}\|{\downarrow}R_6\rangle$
III (path$_1$)	$U_b\|\overrightarrow{q_6}\rangle = \frac{1}{2}\|\overrightarrow{q_5}\rangle + \frac{1}{2\sqrt{2}}\|\overrightarrow{a_1}\rangle + \frac{1}{2\sqrt{2}}\|\overrightarrow{r_1}\rangle - \frac{1}{2}\|{\downarrow}A_{11}\rangle - \frac{1}{2}\|{\downarrow}R_{11}\rangle$
III (path$_2$)	$U_b\|\overrightarrow{p_6}\rangle = \frac{1}{2}\|\overrightarrow{p_5}\rangle + \frac{1}{2\sqrt{2}}\|\overrightarrow{a_1}\rangle - \frac{1}{2\sqrt{2}}\|\overrightarrow{r_1}\rangle - \frac{1}{2}\|{\downarrow}A_{12}\rangle - \frac{1}{2}\|{\downarrow}R_{12}\rangle$
III (path$_{\text{accept}}$)	$U_b\|\overrightarrow{a_2}\rangle = \frac{1}{\sqrt{2}}\|\overrightarrow{a_3}\rangle + \frac{1}{2}\|{\downarrow}A_{13}\rangle + \frac{1}{2}\|{\downarrow}R_{13}\rangle$ $U_b\|\overrightarrow{a_1}\rangle = \frac{1}{\sqrt{2}}\|{\downarrow}A_7\rangle + \frac{1}{\sqrt{2}}\|{\downarrow}R_7\rangle$ $U_b\|\overrightarrow{a_4}\rangle = \frac{1}{\sqrt{2}}\|\overrightarrow{a_3}\rangle - \frac{1}{2}\|{\downarrow}A_{13}\rangle - \frac{1}{2}\|{\downarrow}R_{13}\rangle$ $U_b\|\overrightarrow{a_3}\rangle = \frac{1}{\sqrt{2}}\|{\downarrow}A_8\rangle + \frac{1}{\sqrt{2}}\|{\downarrow}R_8\rangle$
III (path$_{\text{reject}}$)	$U_b\|\overrightarrow{r_2}\rangle = \frac{1}{\sqrt{2}}\|\overrightarrow{r_3}\rangle + \frac{1}{2}\|{\downarrow}A_{14}\rangle + \frac{1}{2}\|{\downarrow}R_{14}\rangle$ $U_b\|\overrightarrow{r_1}\rangle = \frac{1}{\sqrt{2}}\|{\downarrow}A_9\rangle + \frac{1}{\sqrt{2}}\|{\downarrow}R_9\rangle$ $U_b\|\overrightarrow{r_4}\rangle = \frac{1}{\sqrt{2}}\|\overrightarrow{r_3}\rangle - \frac{1}{2}\|{\downarrow}A_{14}\rangle - \frac{1}{2}\|{\downarrow}R_{14}\rangle$ $U_b\|\overrightarrow{r_3}\rangle = \frac{1}{\sqrt{2}}\|{\downarrow}A_{10}\rangle + \frac{1}{\sqrt{2}}\|{\downarrow}R_{10}\rangle$

Fig. 4. Specification of the transition function of the 1KWQFA for L_{NH} (part 2).

check mentioned above fails, each path in question splits equiprobably to one rejecting and one accepting configuration, and the overall probability of acceptance of the machine turns out to be precisely $\frac{1}{2}$. If the input is indeed of the form $(aa^*b)(aa^*b)(aa^*b)^*$, whether the acceptance probability will exceed $\frac{1}{2}$ or not depends on the following additional tasks performed by the computation paths in order to test for the equality mentioned in the definition of L_{NH}:

1. path$_1$ walks over the a's at the speed of one tape square per step until reading the first b. After that point, path$_1$ pauses for one step over each a before moving on to the next symbol.
2. path$_2$ pauses for one step over each a until reading the first b. After that point, path$_2$ walks over each a at the speed of one square per step.
3. On each b except the first one, path$_1$ and path$_2$ split to take the following two courses of action with equal probability:
   (a) In the first alternative, path$_1$ and path$_2$ perform a two-way quantum Fourier transform (QFT) [4]:
      (i) The targets of the QFT are two new computational paths, i.e. path$_{\text{accept}}$ and path$_{\text{reject}}$. Disregarding the equal-probability transitions to the twin halting states mentioned above, the QFT is realized as:

$$\text{path}_1 \rightarrow \frac{1}{\sqrt{2}}\text{path}_{\text{accept}} + \frac{1}{\sqrt{2}}\text{path}_{\text{reject}}$$

$$\text{path}_2 \rightarrow \frac{1}{\sqrt{2}}\text{path}_{\text{accept}} - \frac{1}{\sqrt{2}}\text{path}_{\text{reject}}$$

      (ii) path$_{\text{accept}}$ and path$_{\text{reject}}$ continue computation at the speed of path$_2$, walking over the b's without performing the QFT any more.
   (b) In the second alternative, path$_1$ and path$_2$ continue computation without performing the QFT.
4. On symbol \$, path$_{\text{accept}}$ enters an accepting state, path$_{\text{reject}}$ enters a rejecting state, path$_1$ and path$_2$ enter accepting and rejecting states with equal probability.

Suppose that the input is of the form

$$w = a^x ba^{y_1} ba^{y_2} b \cdots a^{y_t} b,$$

where $x, t, y_1, \cdots, y_t \in \mathbb{Z}^+$.

path_1 reaches the first b earlier than path_2. Once it has passed the first b, path_2 becomes faster, and may or may not catch up with path_1, depending on the number of a's in the input after the first b. The two paths can meet on the symbol following the xth a after the first b, since at that point path_1 will have paused for the same number of steps as path_2. Only if that symbol is a b, the two paths will perform a QFT in the same place and at the same time. To paraphrase, if there exists a k $(1 \leq k \leq t)$ such that $x = \sum_{i=1}^{k} y_i$, path_1 and path_2 meet over the $(k+1)$th b, and perform the QFT at the same step. If there is no such k, the paths either never meet, or meet over an a without a QFT.

The $\mathrm{path}_{\mathrm{accept}}$ and $\mathrm{path}_{\mathrm{reject}}$s that are offshoots of path_1 continue their traversal of the string faster than path_1. On the other hand, the offshoots of path_2 continue their traversal at the same speed as path_2.

By definition, the twin halting states reached during the computation contribute equal amounts to the acceptance and rejection probabilities. path_1 and path_2 accept and reject equiprobably when they reach the end of the string. If path_1 and path_2 never perform the QFT at the same time and in the same position, every QFT produces two equal-probability paths which perform identical tasks, except that one accepts and the other one rejects at the end.

The overall acceptance and rejection probabilities are equal, $\frac{1}{2}$, unless a $\mathrm{path}_{\mathrm{reject}}$ with positive amplitude and a $\mathrm{path}_{\mathrm{reject}}$ with negative amplitude can meet and therefore cancel each other. In such a case, the surviving $\mathrm{path}_{\mathrm{accept}}$'s will contribute the additional acceptance probability that will tip the balance. As described above, such a cancellation is only possible when path_1 and path_2 perform the QFT together.

Therefore, if $w \in L_{NH}$, the overall acceptance probability is greater than $\frac{1}{2}$. If $w \notin L_{NH}$, the overall acceptance probability equals $\frac{1}{2}$. $\square$

Corollary 4.1. *For any space bound s satisfying* $s(n) = o(\log \log n)$,

$$PrSPACE(s) \subsetneq PrQSPACE(s).$$

Corollary 4.2. *For any space bound s satisfying* $s(n) = o(\log n)$, *the class of languages recognized with unbounded error by 1PTMs is a proper subclass of the class of languages recognized with unbounded error by 1QTMs.*

In the next section, we will prove a fact which will allow us to state a similar inclusion relationship between the classes of languages recognized by QTMs with restricted measurements and PTMs using constant space.

Theorem 4.2. *The language*

$$L_{YS} = \{a^{n-1} b a^{kn} | n > 1, k > 0\}$$

is nonstochastic, and can be recognized by a 2KWQFA with unbounded error.

Proof. Suppose that L_{YS} is stochastic. Then, it is not hard to show that $\{a\} \cdot L_{YS}$ is stochastic, too. However, as stated in [36, p. 88], $\{a\} \cdot L_{YS}$ is nonstochastic.

We construct a 2KWQFA $\mathcal{M} = (Q, \Sigma, \delta, q_0, Q_a, Q_r)$, where $\Sigma = \{a, b\}$, and the state sets are

$$Q_n = \{\overrightarrow{q_0}, \overrightarrow{q_1}, \downarrow w_1, \downarrow w_2, \overleftarrow{p_1}, \overrightarrow{p_2}, \overrightarrow{r_1}, \overrightarrow{r_2}, \overleftarrow{r_3}\},$$
$$Q_a = \{\downarrow A_i | 1 \leq i \leq 5\}, \; Q_r = \{\downarrow R_i | 1 \leq i \leq 5\}.$$

Let each U_σ induced by δ act as indicated in Fig. 5, and extend each to be unitary.

If the input string does not begin with an a, or if it contains no b's, the machine halts, and the input is accepted with probability just $\frac{1}{2}$. Otherwise, the head moves to the right until it scans the first b, on which the computation splits to two equiprobable paths, say, $\mathrm{path}_{\mathrm{left}}$ and $\mathrm{path}_{\mathrm{right}}$. Let the number of a's before the first b be $n - 1 > 0$.

$\mathrm{path}_{\mathrm{left}}$ starts with two dummy stationary moves, and then enters an infinite loop. In each iteration of this loop, the head goes to the left end-marker and then comes back to the b at the speed of one step per symbol. At the end of the kth iteration, exactly $2nk + n + 3$ steps after the start of computation, the head scans the b again, and $\mathrm{path}_{\mathrm{left}}$ splits to the superposition of configurations

$$\alpha_k | \overleftarrow{p_1}, n\rangle + \frac{\alpha_k}{2} | \downarrow A_2, n+1\rangle + \frac{\alpha_k}{2} | \downarrow R_2, n+1\rangle + \frac{\alpha_k}{2} | \downarrow A_3, n+1\rangle + \frac{\alpha_k}{2} | \downarrow R_3, n+1\rangle,$$

where $\alpha_k = \left(\frac{1}{\sqrt{2}}\right)^{n+k+1}$, and $|s, h\rangle$ denotes the configuration with state s and head position h.

$\mathrm{path}_{\mathrm{right}}$ checks whether the postfix of the input after the first b is of the form a^+. If not, the machine halts, and the input is accepted with probability $\frac{1}{2}$. Otherwise, the head walks to the right end-marker and then comes back to the b at the speed

Stages	$U_{\cent}, U_a$	$U_b, U_\$$
I	$U_{\cent}\lvert\overrightarrow{q_0}\rangle = \lvert\overrightarrow{q_0}\rangle$ $U_a\lvert\overrightarrow{q_0}\rangle = \frac{1}{\sqrt{2}}\lvert\overrightarrow{q_1}\rangle + \frac{1}{2}\lvert\downarrow A_1\rangle + \frac{1}{2}\lvert\downarrow R_1\rangle$ $U_a\lvert\overrightarrow{q_1}\rangle = \frac{1}{\sqrt{2}}\lvert\overrightarrow{q_1}\rangle - \frac{1}{2}\lvert\downarrow A_1\rangle - \frac{1}{2}\lvert\downarrow R_1\rangle$	$U_b\lvert\overrightarrow{q_0}\rangle = \frac{1}{\sqrt{2}}\lvert\downarrow A_1\rangle + \frac{1}{\sqrt{2}}\lvert\downarrow R_1\rangle$ $U_b\lvert\overrightarrow{q_1}\rangle = \frac{1}{\sqrt{2}}\lvert\downarrow w_1\rangle + \frac{1}{\sqrt{2}}\lvert\overrightarrow{r_1}\rangle$ $U_\$\lvert\overrightarrow{q_0}\rangle = \frac{1}{\sqrt{2}}\lvert\downarrow A_1\rangle + \frac{1}{\sqrt{2}}\lvert\downarrow R_1\rangle$ $U_\$\lvert\overrightarrow{q_1}\rangle = \frac{1}{\sqrt{2}}\lvert\downarrow A_2\rangle + \frac{1}{\sqrt{2}}\lvert\downarrow R_2\rangle$
	$U_{\cent}, U_a, U_b$	
II (path$_{\text{left}}$)	$U_{\cent}\lvert\overleftarrow{p_1}\rangle = \lvert\overrightarrow{p_2}\rangle$ $U_a\lvert\overleftarrow{p_1}\rangle = \lvert\overleftarrow{p_1}\rangle$ $U_a\lvert\overrightarrow{p_2}\rangle = \lvert\overrightarrow{p_2}\rangle$ $U_b\lvert\downarrow w_1\rangle = \lvert\downarrow w_2\rangle$ $U_b\lvert\downarrow w_2\rangle = \frac{1}{\sqrt{2}}\lvert\overleftarrow{p_1}\rangle - \frac{1}{2\sqrt{2}}\lvert\downarrow A_2\rangle - \frac{1}{2\sqrt{2}}\lvert\downarrow R_2\rangle - \frac{1}{2\sqrt{2}}\lvert\downarrow A_3\rangle - \frac{1}{2\sqrt{2}}\lvert\downarrow R_3\rangle$ $U_b\lvert\overrightarrow{p_2}\rangle = \frac{1}{\sqrt{2}}\lvert\overleftarrow{p_1}\rangle + \frac{1}{2\sqrt{2}}\lvert\downarrow A_2\rangle + \frac{1}{2\sqrt{2}}\lvert\downarrow R_2\rangle + \frac{1}{2\sqrt{2}}\lvert\downarrow A_3\rangle + \frac{1}{2\sqrt{2}}\lvert\downarrow R_3\rangle$	
	$U_a, U_b, U_\$$	
II (path$_{\text{right}}$)	$U_a\lvert\overrightarrow{r_1}\rangle = \frac{1}{\sqrt{2}}\lvert\overrightarrow{r_2}\rangle + \frac{1}{2}\lvert\downarrow A_2\rangle + \frac{1}{2}\lvert\downarrow R_2\rangle$ $U_a\lvert\overrightarrow{r_2}\rangle = \frac{1}{\sqrt{2}}\lvert\overrightarrow{r_2}\rangle - \frac{1}{2}\lvert\downarrow A_2\rangle - \frac{1}{2}\lvert\downarrow R_2\rangle$ $U_a\lvert\overleftarrow{r_3}\rangle = \lvert\overleftarrow{r_3}\rangle$ $U_b\lvert\overrightarrow{r_1}\rangle = \frac{1}{\sqrt{2}}\lvert\downarrow A_4\rangle + \frac{1}{\sqrt{2}}\lvert\downarrow R_4\rangle$ $U_b\lvert\overrightarrow{r_2}\rangle = \frac{1}{\sqrt{2}}\lvert\downarrow A_5\rangle + \frac{1}{\sqrt{2}}\lvert\downarrow R_5\rangle$ $U_b\lvert\overleftarrow{r_3}\rangle = \frac{1}{\sqrt{2}}\lvert\downarrow A_2\rangle - \frac{1}{\sqrt{2}}\lvert\downarrow R_2\rangle$ $U_\$\lvert\overrightarrow{r_1}\rangle = \frac{1}{\sqrt{2}}\lvert\downarrow A_3\rangle + \frac{1}{\sqrt{2}}\lvert\downarrow R_3\rangle$ $U_\$\lvert\overrightarrow{r_2}\rangle = \lvert\overleftarrow{r_3}\rangle$	

Fig. 5. Specification of the transition function of the 2KWQFA for L_{YS}.

of one step per symbol. Let the number of a's after the b be $m > 0$. At the $(2m+n+3)$th step, the head scans the b, and path$_{\text{right}}$ splits to the superposition of configurations

$$\left(\frac{1}{\sqrt{2}}\right)^{m+n+1} \lvert \downarrow A_2, n+1\rangle - \left(\frac{1}{\sqrt{2}}\right)^{m+n+1} \lvert \downarrow R_2, n+1\rangle.$$

The two paths can meet and interfere with each other only if

$$2nk + n + 3 = n + 2m + 3$$

or

$$nk = m.$$

This is the case precisely for the members of L_{YS}, where the acceptance probability exceeds the rejection probability, similarly to what we had in the proof of Theorem 4.1. □

We do not know of a one-way QFA for L_{YS}. Note that the somewhat simpler language $L_{fre} = \{a^n b a^n \mid n \in \mathbb{Z}^+\}$ can be recognized with bounded error by a 2PFA [37].

The class C$_=$SPACE(s) is defined [2] as follows: A language L is in C$_=$SPACE(s) if there exists a PTM that runs in space $O(s)$, halts absolutely,[9] and accepts each input w with probability precisely equal to $\frac{1}{2}$ if and only if $x \in L$. We define the analogous family of quantum classes.

Definition 4.1. A language L is in C$_=$QSPACE(s) if there exists a QTM that runs in space O(s), halts absolutely, and accepts each input w with probability precisely equal to $\frac{1}{2}$ if and only if $x \in L$.

Corollary 4.3. *coC$_=$SPACE*$(1) \subsetneq$ *coC$_=$QSPACE*(1).

Proof. Since coC$_=$SPACE(1) is a proper subset of S [20], L_{NH} is not a member of coC$_=$SPACE(1). On the other hand, as shown in Theorem 4.1, L_{NH} is also a member of coC$_=$QSPACE(1). □

[9] That is, for every input w, there exists an integer $k(w)$, such that the PTM halts with probability 1 within $k(w)$ steps.

5. Languages recognized by RT-KWQFAs with unbounded error

In this section, we settle an open problem of Brodsky and Pippenger [10], giving a complete characterization of the class of languages recognized with unbounded error by RT-KWQFAs. It turns out that these restricted RT-QFAs, which are known to be inferior to RT-PFAs in the bounded error case, are equivalent to them in the unbounded error setting.

Lemma 5.1. *Any language recognized with cutpoint (or nonstrict cutpoint) $\frac{1}{2}$ by a RT-PFA with n internal states can be recognized with cutpoint (or nonstrict cutpoint) $\frac{1}{2}$ by a RT-KWQFA with $O(n)$ internal states.*

Proof. Let L be a language recognized by an n-state RT-PFA

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

with (nonstrict) cutpoint $\frac{1}{2}$. We will construct a RT-KWQFA

$$\mathcal{M} = (R, \Sigma, \{U_\sigma | \sigma \in \tilde{\Sigma}\}, r_1, R_a, R_r)$$

which has $3n + 6$ internal states, and recognizes L with (nonstrict) cutpoint $\frac{1}{2}$. The idea is to "embed" the (not necessarily unitary) matrices A_σ of the RT-PFA within the larger unitary matrices U_σ of the RT-KWQFA.

We define Q', v'_0, and $\{A'_\sigma | \sigma \in \tilde{\Sigma}\}$ as follows:

1. $Q' = Q \cup \{q_{n+1}, q_{n+2}\}$;
2. $v'_0 = (1, 0, \ldots, 0)^T$ is an $(n+2)$-dimensional column vector;
3. Each A'_σ is a $(n+2) \times (n+2)$-dimensional matrix: for each $\sigma \in \Sigma \cup \{¢\}$,

$$A'_\sigma = \left(\begin{array}{c|c} A_\sigma & 0_{n\times 2} \\ \hline 0_{2\times n} & I_{2\times 2} \end{array}\right)$$

and

$$A'_\$ = \left(\begin{array}{c|c} 0_{n\times n} & 0_{2\times n} \\ \hline T_{2\times n} & I_{2\times 2} \end{array}\right)\left(\begin{array}{c|c} A_\$ & 0_{n\times 2} \\ \hline 0_{2\times n} & I_{2\times 2} \end{array}\right),$$

where $T(1, i) = 1$ and $T(2, i) = 0$ when $q_i \in Q_a$, and $T(1, i) = 0$ and $T(2, i) = 1$ when $q_i \notin Q_a$, for $1 \le i \le n$.

For a given input $w \in \Sigma^*$,

$$v'_{|\tilde{w}|} = A'_\$ A'_{w_{|w|}} \cdots A'_{w_1} A'_¢ v'_0. \tag{46}$$

It can easily be verified that

$$v'_{|\tilde{w}|} = (0_{1\times n} | f_{\mathcal{P}}(w), 1 - f_{\mathcal{P}}(w))^T.$$

For each A'_σ, we will construct a $(n+2) \times (n+2)$-dimensional upper triangular matrix B_σ so that the columns of

$$\frac{1}{l}\left(\begin{array}{c} A'_\sigma \\ \hline B_\sigma \end{array}\right) \tag{47}$$

form an orthonormal set, where l will be defined later. For this purpose, the entries of B_σ, say $b_{i,j}$ representing $B_\sigma[i, j]$ for $1 \le i, j \le n+2$, can be computed iteratively using the following procedure:

1. Initialize all entries of B_σ to 0.
2. Update the entries of B_σ to make the length of each column of $\left(\begin{array}{c} A'_\sigma \\ \hline B_\sigma \end{array}\right)$ equal to l and also to make the columns of $\left(\begin{array}{c} A'_\sigma \\ \hline B_\sigma \end{array}\right)$ pairwise orthogonal, by executing the following loop:

(i) for $i = 1$ to $n + 2$
(ii) set l_i to the current length of the ith column
(iii) set $b_{i,i}$ to $-\sqrt{l^2 - l_i^2}$
(iv) for $j = i + 1$ to $n + 2$
(v) set $b_{i,j}$ to some nonnegative value so that the ith and jth columns can become orthogonal.

The loop does not work properly if the value of l_i, calculated at the (ii)nd step, is greater than l. Therefore, the value of l should be set carefully. For instance, by setting l to $2n + 7$, the following bounds can be easily verified for each iteration of the loop:

- $l_i < 2$ at the (ii)nd step;
- $2n + 6 < |b_{i,i}| < 2n + 7$ at the (iii)rd step;
- $0 \le b_{j,i} < \frac{1}{n+3}$ at the (v)th step.

We define

$$U_\sigma = \left(\begin{array}{c|c} A''_\sigma & \\ \hline B'_\sigma & D_\sigma \\ \hline B''_\sigma & \end{array} \right), \tag{48}$$

where $A''_\sigma = \frac{1}{l} A'_\sigma$, $B'_\sigma = B''_\sigma = \frac{1}{\sqrt{2}l} B_\sigma$, and the entries of D_σ are selected to make U_σ a unitary matrix.

The state set $R = R_n \cup R_a \cup R_r$ is specified as:

1. $r_{n+1} \in R_a$ corresponds to state q_{n+1};
2. $r_{n+2} \in R_r$ corresponds to state q_{n+2};
3. $\{r_1, \ldots, r_n\} \in R_n$ correspond to the states of Q, where r_1 is the start state;
4. All the states defined for the rows of B'_σ and B''_σ are, respectively, accepting and rejecting states.

$\mathcal{M}$ simulates the computation of $\mathcal{P}$ for the input string w by multiplying the amplitude of each nonhalting state with $\frac{1}{l}$ in each step. Hence, the top $n + 2$ entries of the state vector of $\mathcal{M}$ equal

$$\left(\frac{1}{l}\right)^{|w|+2} (0_{1\times n} | f_{\mathcal{P}}(w), 1 - f_{\mathcal{P}}(w))^T$$

just before the last measurement on the right end-marker. Note that, the halting states, except q_{n+1} and q_{n+2}, will come in accept/reject pairs, so that transitions to them during the computation will add equal amounts to the overall acceptance and rejection probabilities, and therefore will not affect the decision on the membership of the input in L. We conclude that

$$f_{\mathcal{M}}(w) > \frac{1}{2} \text{ if and only if } f_{\mathcal{P}}(w) > \frac{1}{2} \tag{49}$$

and

$$f_{\mathcal{M}}(w) \ge \frac{1}{2} \text{ if and only if } f_{\mathcal{P}}(w) \ge \frac{1}{2}. \quad \Box \tag{50}$$

Theorem 5.1. *The class of languages recognized by RT-KWQFAs with unbounded error is uS (uQAL).*

Proof. Follows from Lemma 5.1, Lemma 3.1 and [21]. □

Corollary 5.1. *UMM = QAL ∩ coQAL = S ∩ coS.*

Proof. It is obvious that UMM ⊆ QAL ∩ coQAL. Let $L \in$ QAL ∩ coQAL. Then, there exist two RT-KWQFAs $\mathcal{M}_1$ and $\mathcal{M}_2$ such that for all $w \in L$, $f_{\mathcal{M}_1}(w) > \frac{1}{2}$ and $f_{\mathcal{M}_2}(w) \ge \frac{1}{2}$, and for all $w \notin L$, $f_{\mathcal{M}_1}(w) \le \frac{1}{2}$ and $f_{\mathcal{M}_2}(w) < \frac{1}{2}$. Let $\mathcal{M}_3$ be a RT-KWQFA running $\mathcal{M}_1$ and $\mathcal{M}_2$ with equal probability. Thus, we obtain that for all $w \in L$, $f_{\mathcal{M}_3}(w) > \frac{1}{2}$, and for all $w \notin L$, $f_{\mathcal{M}_3}(w) < \frac{1}{2}$. Therefore, $L \in$ UMM. □

Considering this result together with Theorem 4.1, we conclude that, unlike classical deterministic and probabilistic finite automata, allowing the tape head to “stay put” for some steps during its left-to-right traversal of the input increases the language recognition power of quantum finite automata in the unbounded error case.

Since unbounded-error RT-PFAs and 2PFAs are equivalent in computational power [22], we are now able to state the following corollary to Theorem 4.1:

Corollary 5.2. *The class of languages recognized with unbounded error by constant-space PTMs is a proper subclass of the respective class for QTMs with restricted measurements.*

Also note that, since the algorithm described in the proof of Theorem 4.1 is presented for a 1KWQFA, Corollary 4.3 is still valid when $\mathrm{coC_=QSPACE}(1)$ is defined for QTMs with restricted measurements.

6. Concluding remarks

In this paper, we examined the capabilities of quantum Turing machines operating under small space bounds in the unbounded error setting. We proved that QTMs are strictly superior to PTMs for all common space bounds that are $o(\log\log n)$, and this superiority extends to all sublogarithmic bounds when the machines are allowed only one-way input head movement. We also gave a full characterization of the class of languages recognized by real-time QFAs employing restricted measurements; they turn out to be equivalent to their probabilistic counterparts. It was also shown that allowing the tape head to “stay put” for some steps during its left-to-right traversal of the input increases the language recognition power of quantum finite automata in the unbounded error case, allowing them to recognize some nonstochastic languages. This means that two-way (and even one-way) QFAs are strictly more powerful than RT-QFAs; whereas 2DFAs and unbounded-error 2PFAs are known to be equivalent in power to their real-time versions [22,38].

While we have established some new results relating to the relationship of probabilistic and quantum complexity classes in this paper, the work reported here also gives rise to some new open questions. As already mentioned, Watrous proved the equality $\mathrm{PrQSPACE}(s)$=$\mathrm{PrSPACE}(s)$ $(s \in \Omega(\log n))$ for the cases where PrQSPACE is defined in terms of Wa98-QTMs [1,2], and Wa03-QTMs [3]. We do not know how to prove these results for our more general QTMs, and so the most that we can say about the relationship among these classes now is $\mathrm{PrSPACE}(s) \subsetneq \mathrm{PrQSPACE}(s)$ $(s \in o(\log\log n))$, and $\mathrm{PrSPACE}(s) \subseteq \mathrm{PrQSPACE}(s)$ for all s. The only efficient simulation technique of a quantum machine by a probabilistic machine that remains valid for our definitions is that of Lemma 3.1.

The reader may wonder why we did not present QTMs to be unidirectional by definition. The reason is that the known techniques [24] for converting QTMs with arbitrary head movements to unidirectional QTMs do not work for the space-bounded case when the stationary “move” $(\downarrow)$ is included in the set of allowed head directions, and we stuck to the general definition to avoid any possibility of an unnecessary limitation of computational power.

After it was discovered in the context of this research, the simulation method presented in Lemma 5.1 has been modified and used in several contexts [15,39,40] to help establish relationships between many different machine models.

Several real-time QFA variants have appeared in the literature. Our results show that all of these which are at least as general as the RT-KWQFA ([5,8,11,13], and the real-time version of the machines of [9]) have the same computational power in the unbounded error case. The class of languages recognized with unbounded error by the weakest variant, the Moore–Crutchfield QFA [17], is known [41] to be a proper subset of uS. One important model for which no such characterization is yet known is the Latvian QFA [12]. Another question left open in this work is the relationship between the computational powers of 1QFAs and 2QFAs.

Acknowledgments

We are grateful to Andris Ambainis and John Watrous for their helpful comments on earlier versions of this paper. We also thank Rūsiņš Freivalds for his answers to our questions regarding nonstochastic languages, Maris Ozols for directing us to Watrous' lecture notes, Evgeniya Khusnitdinova for her kind help with Russian translations, and the two anonymous referees for their very useful suggestions.

Appendix A. Well-formedness conditions

A.1. Local conditions for 2QFA well-formedness

Let c_{j_1} and c_{j_2} be two configurations, and v_{j_1} and v_{j_2} be the corresponding columns of E (See Fig. 1). The value of $v_{j_1}[i]$ is determined by δ if the ith entry of v_{j_1} corresponds to a configuration to which c_{j_1} can evolve in one step, and it is zero otherwise. Let x_1 and x_2 be the positions of the input tape head for the configurations c_{j_1} and c_{j_2}, respectively. In order to evolve to the same configuration in one step, the difference between x_1 and x_2 must be at most 2. Therefore, we obtain a total of three different cases, listed below, that completely define the restrictions on the transition function. Note that, by taking the conjugates of each summation, we handle the symmetric cases that are shown in the parentheses.

For all $q_1, q_2 \in Q$; $\sigma \in \tilde{\Sigma}$; (the summations are taken over $q' \in Q$; $d \in \diamond$; and $\omega \in \Omega$),

1. $x_1 = x_2$:

$$\sum_{q' \in Q, d \in \diamond, \omega \in \Omega} \overline{\delta(q_1, \sigma, q', d, \omega)} \delta(q_2, \sigma, q', d, \omega) = \begin{cases} 1 & q_1 = q_2 \\ 0 & \text{otherwise} \end{cases} \tag{51}$$

2. $x_1 = x_2 - 1$ ($x_1 = x_2 + 1$):

$$\sum_{q' \in Q, \omega \in \Omega} \overline{\delta(q_1, \sigma, q', \rightarrow, \omega)} \delta(q_2, \sigma, q', \downarrow, \omega) + \overline{\delta(q_1, \sigma, q', \downarrow, \omega)} \delta(q_2, \sigma, q', \leftarrow, \omega) = 0. \tag{52}$$

3. $x_1 = x_2 - 2$ ($x_1 = x_2 + 2$):

$$\sum_{q' \in Q, \omega \in \Omega} \overline{\delta(q_1, \sigma, q', \rightarrow, \omega)} \delta(q_2, \sigma, q', \leftarrow, \omega) = 0. \tag{53}$$

A.2. Unidirectional machines

The well-formedness of unidirectional QTMs can be checked using the simple conditions in Fig. 6. Removing the reference to worktape symbols, we obtain the analogous constraints for unidirectional 2QFAs as shown in Fig. 7.

For $q_1, q_2 \in Q$; $\sigma \in \tilde{\Sigma}$; $\gamma_1, \gamma_2 \in \Gamma$,

$$\sum_{q' \in Q, \gamma' \in \Gamma, \omega \in \Omega} \overline{\delta(q_1, \sigma, \gamma_1, q', \gamma', \omega)} \delta(q_2, \sigma, \gamma_2, q', \gamma', \omega) = \begin{cases} 1 & q_1 = q_2 \text{ and } \gamma_1 = \gamma_2 \\ 0 & \text{otherwise} \end{cases}. \tag{54}$$

Fig. 6. The local conditions for unidirectional QTM well-formedness.

For $q_1, q_2 \in Q$; $\sigma \in \tilde{\Sigma}$,

$$\sum_{q' \in Q, \omega \in \Omega} \overline{\delta(q_1, \sigma, q', \omega)} \delta(q_2, \sigma, q', \omega) = \begin{cases} 1 & q_1 = q_2 \\ 0 & \text{otherwise} \end{cases}. \tag{55}$$

Fig. 7. The local conditions for unidirectional 2QFA well-formedness.

As is the case with PTMs, the transition function of a unidirectional QTM can be specified easily by transition matrices of the form $\{E_{\sigma,\omega}\}$, whose rows and columns are indexed by (internal state, work tape symbol) pairs for each $\sigma \in \tilde{\Sigma}$ and $\omega \in \Omega$. It can be verified that the well-formedness condition is then equivalent to the requirement that, for each $\sigma \in \tilde{\Sigma}$,

$$\sum_{\omega \in \Omega} E^{\dagger}_{\sigma,\omega} E_{\sigma,\omega} = I. \tag{56}$$

Similarly, for each $\sigma \in \tilde{\Sigma}$ and $\omega \in \Omega$, well-formed unidirectional 2QFAs can be described by transition matrices of the form $\{E_{\sigma,\omega}\}$, whose rows and columns are indexed by internal states, such that for each $\sigma \in \tilde{\Sigma}$,

$$\sum_{\omega \in \Omega} E^{\dagger}_{\sigma,\omega} E_{\sigma,\omega} = I. \tag{57}$$

Appendix B. CQTMs

To specialize our general QTM model in order to ensure that the head positions are classical, we associate combinations of head movements with measurement outcomes. There are nine different pairs of possible movement directions ($\diamond^2 =$

$\{\leftarrow, \downarrow, \rightarrow\} \times \{\leftarrow, \downarrow, \rightarrow\}$) for the input and work tape heads, and so we can classify register symbols with the function

$$D_r : \Omega \rightarrow \diamondsuit^2. \tag{58}$$

We have $D_r(\omega) = (\downarrow, \downarrow)$ if $\omega \in \Omega_a \cup \Omega_r$. We split Ω_n into nine parts, i.e.

$$\Omega_n = \bigcup_{d_i, d_w \in \diamondsuit} \Omega_{n,d_i,d_w}, \tag{59}$$

where

$$\Omega_{n,d_i,d_w} = \{\omega \in \Omega_n | D_r(\omega) = (d_i, d_w)\}. \tag{60}$$

Therefore, the outcome set will have 11 elements, represented as triples, specified as follows:

1. "(n, d_i, d_w)": the computation continues and the positions of the input and work tape heads are updated with respect to d_i and d_w, respectively;
2. "$(a, \downarrow, \downarrow)$": the computation halts and the input is accepted with no head movement;
3. "$(r, \downarrow, \downarrow)$": the computation halts and the input is rejected with no head movement.

The transition function of CQTMs will be specified so that when the CQTM is in state q and reads σ and γ, respectively, on the input and work tapes, it will enter state q', and write γ' and ω, respectively, on the work tape and the finite register with the amplitude

$$\delta(q, \sigma, \gamma, q', \gamma', \omega) \in \tilde{\mathbb{C}}. \tag{61}$$

Since the update of the positions of the input and work tape heads is performed classically, it is no longer a part of the transitions. Note that the transition function of 2QFAs with classical head (2CQFAs) [9] is obtained by removing the mention of the work tape from the above description.

Moreover, as with unidirectional QTMs (resp. unidirectional 2QFAs), for each $\sigma \in \tilde{\Sigma}$ and $\omega \in \Omega$, CQTMs (2CQFAs) can be described by transition matrices $\{E_{\sigma,\omega}\}$ satisfying the same properties. (See Appendix A.)

As also argued in [3], CQTMs are sufficiently general for simulating any classical TM. We will present a trivial simulation.

Lemma B.1. *CQTMs can simulate any PTM exactly.*

Proof. Let $\mathcal{P} = (Q, \Sigma, \Gamma, \delta_{\mathcal{P}}, q_1, Q_a, Q_r)$ be a PTM. We build a CQTM $\mathcal{M} = (Q, \Sigma, \Gamma, \Omega, \delta_{\mathcal{M}}, q_1, \Delta)$. For each $(q, \gamma, q', \gamma') \in Q \times \Gamma \times Q \times \Gamma$, we define a register symbol $\omega_{(q,\gamma,q',\gamma')}$ such that

1. if $q' \in Q_a$: $\omega_{(q,\gamma,q',\gamma')} \in \Omega_{(a,\downarrow,\downarrow)}$;
2. if $q' \in Q_r$: $\omega_{(q,\gamma,q',\gamma')} \in \Omega_{(r,\downarrow,\downarrow)}$;
3. if $q' \in Q_n$: $\omega_{(q,\gamma,q',\gamma')} \in \Omega_{(n,D_i(q'),D_w(q'))}$.

We conclude with setting

$$\delta_{\mathcal{M}}(q, \sigma, \gamma, q', \gamma', \omega_{(q,\gamma,q',\gamma')}) = \sqrt{\delta_{\mathcal{P}}(q, \sigma, \gamma, q', \gamma')} \tag{62}$$

for each $\sigma \in \Sigma$, and setting the values of $\delta_{\mathcal{M}}$ that are still undefined to zero. $\square$

This result is also valid for two-way and real-time finite automata:

Corollary B.1. *2CQFAs (RT-QFAs) can simulate any 2PFA (RT-PFA) exactly.*

References

[1] J. Watrous, Space-bounded quantum computation, Ph.D. Thesis, University of Wisconsin–Madison, USA, 1998.
[2] J. Watrous, Space-bounded quantum complexity, Journal of Computer and System Sciences 59 (2) (1999) 281–326.
[3] J. Watrous, On the complexity of simulating space-bounded quantum computations, Computational Complexity 12 (1–2) (2003) 48–84.
[4] 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.
[5] A. Nayak, Optimal lower bounds for quantum automata and random access codes, in: FOCS'99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science, 1999, pp. 369–376.

[6] 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.
[7] M. Amano, K. Iwama, Undecidability on quantum finite automata, in: STOC'99: Proceedings of the 31st Annual ACM Symposium on Theory of Computing, 1999, pp. 368–375.
[8] K. Paschen, Quantum finite automata using ancilla qubits, Tech. Rep., University of Karlsruhe, 2000. Available from: <http://digbib.ubka.uni-karlsruhe.de/volltexte/1452000>.
[9] A. Ambainis, J. Watrous, Two-way finite automata with quantum and classical states, Theoretical Computer Science 287 (1) (2002) 299–311.
[10] A. Brodsky, N. Pippenger, Characterizations of 1-way quantum finite automata, SIAM Journal on Computing 31 (5) (2002) 1456–1478.
[11] A. Bertoni, C. Mereghetti, B. Palano, Quantum computing: 1-way quantum automata, in: Z. Ésik, Z. Fülöp (Eds.), Developments in Language Theory, LNCS, vol. 2710, 2003, pp. 1–20.
[12] A. Ambainis, M. Beaudry, M. Golovkins, A. Ķikusts, M. Mercer, D. Thérien, Algebraic results on quantum automata, Theory of Computing Systems 39 (1) (2006) 165–188.
[13] M. Hirvensalo, Various aspects of finite quantum automata, in: DLT'08: Proceedings of the 12th International Conference on Developments in Language Theory, 2008, pp. 21–33.
[14] A. Yakaryılmaz, A.C.C. Say, Efficient probability amplification in two-way quantum finite automata, Theoretical Computer Science 410 (20) (2009) 1932–1941.
[15] A. Yakaryılmaz, A.C.C. Say, Succinctness of two-way probabilistic and quantum finite automata, Discrete Mathematics and Theoretical Computer Science 12 (4) (2010) 19–40.
[16] A. Ambainis, A. Yakaryılmaz, Automata: from Mathematics to Applications, Ch. Automata and quantum computing, in preparation.
[17] C. Moore, J.P. Crutchfield, Quantum automata and quantum grammars, Theoretical Computer Science 237 (1–2) (2000) 275–306.
[18] V.D. Blondel, E. Jeandel, P. Koiran, N. Portier, Decidable and undecidable problems about quantum automata, SIAM Journal on Computing 34 (6) (2005) 1464–1473.
[19] M.O. Rabin, Probabilistic automata, Information and Control 6 (1963) 230–243.
[20] A. Paz, Introduction to Probabilistic Automata, Academic Press, New York, 1971.
[21] P. Turakainen, Generalized automata and stochastic languages, Proceedings of the American Mathematical Society 21 (1969) 303–309.
[22] J. Kaņeps, Stochasticity of the languages acceptable by two-way finite probabilistic automata, DiscreteMathematics and Applications 1 (1991) 405–421. (The original version in Russian appeared in Diskretnaya Matematika 1 (1989) 63–77.)
[23] M.A. Nielsen, I.L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000.
[24] E. Bernstein, U. Vazirani, Quantum complexity theory, SIAM Journal on Computing 26 (5) (1997) 1411–1473.
[25] D. Aharonov, A. Kitaev, N. Nisan, Quantum circuits with mixed states, in: STOC'98: Proceedings of the 13th Annual ACM Symposium on Theory of Computing, 1998, pp. 20–30.
[26] A. Yakaryılmaz, Classical and quantum computation with small space bounds, Ph.D. Thesis, Boğaziçi University, 2011, arXiv:1102.0378.
[27] J. Watrous, Quantum computational complexity, in: R.A. Meyers (Ed.), Encyclopedia of Complexity and Systems Science, Springer, 2009, pp. 7174–7201.
[28] M. Sipser, Introduction to the Theory of Computation, second ed., Thomson Course Technology, United States of America, 2006.
[29] A. Szepietowski, Turing Machines with Sublogarithmic Space, Springer-Verlag, 1994.
[30] R. Freivalds, M. Karpinski, Lower space bounds for randomized computation, in: ICALP'94: Proceedings of the 21st International Colloquium on Automata, Languages and Programming, LNCS, vol. 820, 1994, pp. 580–592.
[31] L. Li, D. Qiu, Determining the equivalence for one-way quantum finite automata, Theoretical Computer Science 403 (1) (2008) 42–51.
[32] A. Yakaryılmaz, A.C.C. Say, Language recognition by generalized quantum finite automata with unbounded error, Tech. Rep., 2009, arXiv:0901.2703 (Poster presentation at the TQC2009).
[33] D. van Melkebeek, T. Watson, A quantum time-space lower bound for the counting hierarchy, Electronic Colloquium on Computational Complexity (ECCC) 15 (017). Available from: <http://eccc.hpi-web.de/eccc-reports/2008/TR08-017/index.html>.
[34] F. Le Gall, Exponential separation of quantum and classical online space complexity, Theory of Computing Systems 45 (2) (2009) 188–202.
[35] M. Nasu, N. Honda, A context-free language which is not acceptable by a probabilistic automaton, Information and Control 18 (3) (1971) 233–236.
[36] A. Salomaa, M. Soittola, Automata-Theoretic Aspects of Formal Power Series, Texts and Monographs in Computer Science, Springer-Verlag, NewYork, 1978.
[37] R. Freivalds, Probabilistic two-way machines, in: Proceedings of the International Symposium on Mathematical Foundations of Computer Science, LNCS, vol. 118, 1981, pp. 33–45.
[38] J.C. Shepherdson, The reduction of two-way automata to one-way automata, IBM Journal of Research and Development 3 (1959) 198–200.
[39] A. Yakaryılmaz, A.C.C. Say, Languages recognized by nondeterministic quantum finite automata, Quantum Information and Computation 10 (9 and 10) (2010) 747–770.
[40] A. Yakaryılmaz, A.C.C. Say, Probabilistic and quantum finite automata with postselection, Tech. Rep., 2011, arXiv:1102.0666 (A preliminary version of this paper appeared in the Proceedings of Randomized and Quantum Computation (Satellite Workshop of MFCS and CSL 2010), pp. 14–24, 2010).
[41] A. Bertoni, M. Carpentieri, Analogies and differences between quantum and stochastic automata, Theoretical Computer Science 262 (1–2) (2001) 69–81.