On the Power of Quantum Finite State Automata

Attila Kondacs*

Computer Science Department
Eotvos Lorand University
Budapest, Hungary

John Watrous†

Computer Sciences Department
University of Wisconsin
Madison, Wisconsin 53706

Abstract

In this paper, we introduce 1-way and 2-way quantum finite state automata (1qfa's and 2qfa's), which are the quantum analogues of deterministic, nondeterministic and probabilistic 1-way and 2-way finite state automata.

We prove the following facts regarding 2qfa's.

1. *For any $\epsilon > 0$, there is a 2qfa M which recognizes the non-regular language $L = \{a^m b^m \mid m \geq 1\}$ with (one-sided) error bounded by ϵ, and which halts in linear time. Specifically, M accepts any string in L with probability 1 and rejects any string not in L with probability at least $1 - \epsilon$.*
2. *For every regular language L, there is a reversible (and hence quantum) 2-way finite state automaton which recognizes L and which runs in linear time.*

In fact, it is possible to define 2qfa's which recognize the non-context-free language $\{a^m b^m c^m | m \geq 1\}$, based on the same technique used for 1. Consequently, the class of languages recognized by linear time, bounded error 2qfa's properly includes the regular languages. Since it is known that 2-way deterministic, nondeterministic and polynomial expected time, bounded error probabilistic finite automata can recognize only regular languages, it follows that 2qfa's are strictly more powerful than these "classical" models.

In the case of 1-way automata, the situation is reversed. We prove that the class of languages recognizable by bounded error 1qfa's is properly contained in the class of regular languages.

1 Introduction

There is a growing body of evidence which suggests that computational machines whose behavior is governed by quantum physics may be considerably more powerful than their classical counterparts. Undoubtedly the most celebrated of these results are Peter Shor's factoring and discrete logarithm algorithms for quantum computers [23, 24]. Other results include Grover's quantum searching algorithm [12] and various oracle results regarding the power of quantum computers [1, 3, 4, 7, 25].

The above examples regard the power of universal quantum machines (e.g. quantum Turing machines [1, 5], quantum circuits [6, 27], quantum cellular automata [8, 16, 17, 26]). In this paper, we define two new, much more restricted quantum computational models: 1-way and 2-way quantum finite state automata (1qfa's and 2qfa's).

The main focus of this paper will be on 2qfa's, which are the quantum analogue of deterministic, nondeterministic and probabilistic 2-way finite state automata (2dfa's, 2nfa's and 2pfa's). While 2dfa's and 2nfa's are known to be equivalent in power to ordinary (1-way) deterministic automata [13, 20, 22], it was shown by Freivalds in [11] that the non-regular language $\{a^m b^m | m \geq 1\}$ could be recognized by a 2pfa with arbitrarily small error. However, the 2pfa's for $\{a^m b^m \mid m \geq 1\}$ defined by Freivalds require exponential expected time, and it was subsequently shown by Dwork and Stockmeyer [9, 10] that any 2pfa recognizing a non-regular language with bounded error probability must take exponential expected time on infinitely many inputs. Thus, 2dfa's, 2nfa's and polynomial expected time, bounded error 2pfa's recognize exactly the regular languages.

We show that 2qfa's are strictly more powerful than 2dfa's, 2nfa's and 2pfa's in the sense that linear time, bounded error 2qfa's recognize a class of languages which properly includes the regular languages. Specifically, we prove:

1. For any $\epsilon > 0$, there is a 2qfa M which recognizes the non-regular language $L = \{a^m b^m \mid m \geq 1\}$ with (one-sided) error bounded by ϵ, and which halts in

*E-mail: `kondacs@cs.elte.hu`.

†E-mail: `watrous@cs.wisc.edu`. Supported in part by NSF grant CCR-95-10244.

linear time. Specifically, M accepts any string in L with probability 1 and rejects any string not in L with probability at least $1-\epsilon$.

2. For every regular language L, there is a reversible (and hence quantum) 2-way finite state automaton which recognizes L and which runs in linear time.

We prove 1 by exhibiting a sequence of linear time 2qfa's for $\{a^m b^m \mid m \geq 1\}$ which have error probability approaching zero. We also note that the non-context-free language $\{a^m b^m c^m \mid m \geq 1\}$ can be recognized by bounded error, linear time 2qfa's, based on the same technique. In order to prove 2, we apply a technique from a recent result of Lange, McKenzie and Tapp [14] regarding reversible simulation of deterministic Turing machines to the finite automaton case. A corollary of 2 is that reversible 2-way finite state automata are equivalent in power to (1-way or 2-way) deterministic finite state automata. This is in contrast to the fact that 1-way reversible finite state automata are known to be less powerful than deterministic finite state automata [19].

In this paper, we also prove an elementary result regarding 1qfa's; similar to 1-way reversible finite state automata, 1qfa's are strictly less powerful than 1-way deterministic, nondeterministic and probabilistic finite state automata (1dfa's, 1nfa's and 1pfa's). The fact that bounded error 1qfa's can only recognize regular languages can be demonstrated by modifying slightly a proof due to Rabin [21] for the analogous result for probabilistic automata. In order to demonstrate that the containment of the class of languages recognized by bounded error 1qfa's in the regular languages is proper, a simple example of a regular language which evinces this given: we show that $\{a,b\}^*a$ cannot be recognized by any bounded error 1qfa. Independent of this paper, study of a slightly different model of 1-way quantum finite state automata has recently appeared in [18].

The remainder of this paper has the following organization. In Section 2, we define 2-way quantum finite state automata, and in Section 3 we provide a well-formedness criterion for 2qfa's and discuss a method by which well-formed 2qfa's can be easily specified. In Section 4, we present bounded error, linear time 2qfa's for the language $\{a^m b^m \mid m \geq 1\}$, and in Section 5 we show that any regular language can be recognized by a 2-way reversible (and hence quantum) finite state automaton. Finally, we discuss 1-way quantum finite state automata in Section 6, and conclude with mention of some open problems in Section 7.

2 Definition of 2-way quantum finite automata

A 2-way quantum finite state automaton (2qfa) consists of a finite state control and a 2-way tape head which scans a read-only input tape. Formally, a 2qfa is specified by a 6-tuple $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$, where Q is a finite set of states, Σ is a finite input alphabet, δ is a transition function (described below), $q_0 \in Q$ is the initial state, and $Q_{acc} \subset Q$ and $Q_{rej} \subset Q$ are the sets of accepting states and rejecting states, respectively. Elements of Q_{acc} and Q_{rej} are *halting states* and elements of $Q_{non} = Q \backslash (Q_{acc} \cup Q_{rej})$ are *non-halting states*. It is assumed that $q_0 \in Q_{non}$ and $Q_{acc} \cap Q_{rej} = \emptyset$. In addition to the input symbols Σ, there are two symbols $\cent, \$ \notin \Sigma$ which will be used to mark the left and right ends of the input string, respectively. Together with the input alphabet, these symbols form the tape alphabet $\Gamma = \Sigma \cup \{\cent, \$\}$.

Unlike the usual definition of 2-way automata, we will assume that the tape of any 2qfa is circular in the sense that if the machine is scanning the last tape square and subsequently moves its tape head right (or is scanning the first tape square and moves its tape head left), the tape head will then be scanning the first tape square (last tape square, respectively). This is simply a convenient way to restrict 2qfa's from moving outside the boundaries of the input on the tape; the alternative, which is restricting the movement of the tape head for symbols $\cent$ and \$, introduces unnecessary complications in the quantum case. The contents of any tape can be described by a mapping $x : \mathbb{Z}_n \rightarrow \Gamma$, n being the number of distinct tape squares on the tape. Such a mapping will itself be referred to as a *tape*, and n will be identified as the length of that tape (also denoted $|x|$). For technical reasons, we will make the further assumption that all tapes have length at least 3.

The number of configurations of a 2qfa M on any tape x of length n is precisely $n|Q|$, since there are n possible locations for the tape head and $|Q|$ internal states. For fixed M, we denote this set of configurations by C_n, and identify C_n with $Q \times \mathbb{Z}_n$ in the obvious way.

A superposition of M on a tape x of length n is any norm 1 element of the finite-dimensional Hilbert space $\mathcal{H}_n = \ell_2(C_n)$ (i.e. the space of mappings from C_n to $\mathbb{C}$ with the usual inner product). We will use the Dirac notation to express superpositions. For each $c \in C_n$, $|c\rangle$ denotes the unit vector which takes value 1 at c and 0 elsewhere; all other elements of $\mathcal{H}_n$ may be expressed as linear combinations of these basis vectors. For a superposition $|\psi\rangle = \sum_{c \in C_n} \alpha_c |c\rangle$, α_c is the *amplitude* associated with c in superposition $|\psi\rangle$.

We are now ready to describe the transition function δ. This is a mapping of the form

$$\delta : Q \times \Gamma \times Q \times \{-1, 0, 1\} \to \mathbb{C},$$

and is to be interpreted as follows. For each $q, q' \in Q$, $\sigma \in \Gamma$ and $d \in \{-1, 0, 1\}$, $\delta(q, \sigma, q', d)$ represents the amplitude with which a machine currently in state q and scanning symbol σ will change state to q' and move its tape head in direction d. For any tape x, δ induces an operator U_δ^x (called the *time-evolution operator* of M on tape x) on $\mathcal{H}_{|x|}$ as follows.

$$U_\delta^x \, |q, k\rangle = \sum_{q', d} \delta(q, x(k), q', d) \; |q', k + d \; (\mathrm{mod}\; |x|)\rangle$$

for each $(q, k) \in C_{|x|}$, and is extended to all of $\mathcal{H}_{|x|}$ by linearity. Thus, $(U_\delta^x)^t \, |\psi\rangle$ is the superposition which would be obtained if M on tape x were placed in superposition $|\psi\rangle$ and run (unobserved) for t steps.

In order for a superposition to be valid, it must be of unit norm. This restriction is inherent to the quantum theory, and its necessity will be apparent from the section below regarding observables. A machine which guarantees that any valid superposition will evolve into another valid superposition is said to be *well-formed*. Since each $\mathcal{H}_n$ is finite-dimensional, this corresponds to the time-evolution operator U_δ^x for each tape x being a unitary operator. Note that this condition is quite restrictive. For example, arbitrary 2dfa's which are not reversible (i.e. whose "yields" relations are not one-to-one) will not directly correspond to well-formed 2qfa's. In Section 3, we provide a criterion to determine whether or not a given 2qfa is well-formed.

Observables

The time-evolution operator U_δ^x specifies how a 2qfa will evolve given tape x, assuming that the 2qfa is not observed by an outside observer. We must assume, however, that a machine has to be observed in order for it to yield any information about its computation. The information obtained by a particular sort of observation, as well as the effect of that observation on the machine, is described by an *observable*.

An observable $\mathcal{O}$ for a 2qfa M is a decomposition of each Hilbert space $\mathcal{H}_n$ into subspaces: $\mathcal{H}_n = E_1 \oplus \cdots \oplus E_k$, where the E_j are pairwise orthogonal. Corresponding to each $j = 1, \ldots, k$ will be some particular (distinct) outcome; if a 2qfa M on tape x is in superposition $|\psi\rangle$ and is observed using observable $\mathcal{O}$, then one of these outcomes will result and M will be modified in a certain way. Specifically, let $|\psi_j\rangle$ be the projection of $|\psi\rangle$ onto E_j for each j, so that $|\psi\rangle = |\psi_1\rangle + \cdots + |\psi_k\rangle$. Then the result of the observation is as follows.

1. The outcome observed is random, each outcome j being seen with probability $\| |\psi_j\rangle \|^2$.
2. Immediately after the observation, the machine will "collapse" to the superposition $\frac{1}{\| |\psi_j\rangle \|} |\psi_j\rangle$, where j corresponds to the particular outcome which was observed.

An example of an observable is to let $c_1, \ldots, c_k$ be an enumeration of C_n, and let $E_j = \mathrm{span}\{ |c_j\rangle \}$. (The outcome of the observation can be taken to be simply a description of the corresponding configuration.) Now, the probability of seeing a given configuration is the absolute square of the amplitude associated with that configuration, and upon observation the machine will collapse to the superposition $|c\rangle$ for whichever configuration c which was observed.

We will use a different observable, however, which will correspond to determining not the entire configuration of a machine, but rather only whether that machine is in an accepting, rejecting or non-halting state. For fixed n, define $C_{acc} = Q_{acc} \times \mathbb{Z}_n$, $C_{rej} = Q_{rej} \times \mathbb{Z}_n$ and $C_{non} = Q_{non} \times \mathbb{Z}_n$, and let $E_{acc} = \mathrm{span}\{ |c\rangle \mid c \in C_{acc} \}$, $E_{rej} = \mathrm{span}\{ |c\rangle \mid c \in C_{rej} \}$, $E_{non} = \mathrm{span}\{ |c\rangle \mid c \in C_{non} \}$. Now, let $\mathcal{O}$ be the observable corresponding to the decomposition $\mathcal{H}_n = E_{acc} \oplus E_{rej} \oplus E_{non}$, where the outcome of any observation is "accept", "reject" or "non-halting" accordingly. For example, if the amplitude associated with every halting configuration in some superposition is 0, then the result of an observation using observable $\mathcal{O}$ will be "non-halting" with probability 1, and the superposition will "collapse" to itself (i.e. will not be altered by the observation).

Languages recognized by 2qfa's

Finally, we can discuss the languages recognized by 2qfa's. For a given input string $w \in \Sigma^*$, we define a corresponding tape x_w which has length $|w| + 2$ and takes the form $x_w(0) = ¢$, $x_w(|w| + 1) = \$$ and $x_w(i) = w_i$ for $1 \le i \le |w|$. (The exceptional case is when the input string is the empty string; in this case the corresponding tape will take the form $x(0) = ¢$ and $x(1) = \$$, with $x(2)$ defined arbitrarily.) Now, to say that a 2qfa M is run on input w means that 1) the tape of M is described by x_w, 2) the computation begins with M in superposition $|q_0, 0\rangle$, and 3) after each step, the machine is observed using observable $\mathcal{O}$ defined in the previous paragraph. The computation continues until the result of an observation is "accept" or "reject", at which time the computation halts. The computation can now be treated in the same manner as for a probabilistic machine: if input w results in "accept" with probability greater than $1/2$, then w is an element of the language recognized by M, otherwise it is not.

As in the probabilistic case, classes of languages may be defined by placing restrictions on the 2qfa's which recognize them, such as running time, probability of error, etc. In this paper we are interested in the class of languages which can be recognized by polynomial time 2qfa's with error probability bounded away from $1/2$.

3 Defining well-formed 2qfa's

We will only be interested in 2qfa's which are well-formed, so it will be necessary to be able to determine whether or not given machines satisfy this condition. The following proposition, which is analogous to Bernstein and Vazirani's criterion for well-formedness of quantum Turing machines [1], allows us to do this.

Proposition 1 *A 2qfa $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ is well-formed if and only if for every choice of $\sigma, \sigma_1, \sigma_2 \in \Gamma$ and $q_1, q_2 \in Q$ the following hold.*

1. $$\sum_{q',d} \overline{\delta(q_1,\sigma,q',d)}\,\delta(q_2,\sigma,q',d) = \begin{cases} 1 & q_1 = q_2 \\ 0 & q_1 \neq q_2 \end{cases}$$
2. $$\sum_{q'} \Big(\overline{\delta(q_1,\sigma_1,q',1)}\,\delta(q_2,\sigma_2,q',0) + \overline{\delta(q_1,\sigma_1,q',0)}\,\delta(q_2,\sigma_2,q',-1)\Big) = 0$$
3. $$\sum_{q'} \overline{\delta(q_1,\sigma_1,q',1)}\,\delta(q_2,\sigma_2,q',-1) = 0$$

Proof. For each x, U_δ^x is unitary if and only if the vectors $U_\delta^x\,|q,k\rangle$ for $q \in Q, k \in \mathbb{Z}_{|x|}$ are orthonormal. Condition 1 is equivalent to the statement that, for every x, we have $\|U_\delta^x\,|q,k\rangle\| = 1$ for each q and k, and $U_\delta^x\,|q_1,k\rangle \perp U_\delta^x\,|q_2,k\rangle$ for $q_1 \neq q_2$. Conditions 2 and 3 are equivalent to $U_\delta^x\,|q_1,k\rangle \perp U_\delta^x\,|q_2,k+1\rangle$ and $U_\delta^x\,|q_1,k\rangle \perp U_\delta^x\,|q_2,k+2\rangle$, respectively, for each x, q_1, q_2 and k, with $|x| \geq 5$. For $3 \leq |x| \leq 4$, conditions 2 and 3 are sufficient to show $U_\delta^x\,|q_1,k\rangle \perp U_\delta^x\,|q_2,k+1\rangle$ and $U_\delta^x\,|q_1,k\rangle \perp U_\delta^x\,|q_2,k+2\rangle$. It is clear that $U_\delta^x\,|q_1,k_1\rangle \perp U_\delta^x\,|q_2,k_2\rangle$ whenever k_1 and k_2 are more than two squares away, since the tape head of a 2qfa moves at most one square per step. Thus, the vectors $U_\delta^x\,|q,k\rangle$, $q \in Q, k \in \mathbb{Z}_{|x|}$ are orthonormal for every x if and only if conditions 1, 2 and 3 are satisfied. ∎

Proposition 1 provides a relatively simple criterion to determine whether a 2qfa is or is not well-formed. However, it will simplify matters to mention a method by which well-formed machines can be more easily specified. In essence, the method is to decompose the transition function δ into two parts: one for transforming states and the other for moving the tape head.

Consider the Hilbert space $\ell_2(Q)$, where Q is the set of internal states of a 2qfa M. Suppose that we have a linear operator $V_\sigma : \ell_2(Q) \to \ell_2(Q)$ for each $\sigma \in \Gamma$, and a function $D : Q \to \{-1, 0, 1\}$. Define transition function δ as

$$\delta(q,\sigma,q',d) = \begin{cases} \langle q' \,|\, V_\sigma \,|\, q\rangle & D(q') = d \\ 0 & D(q') \neq d. \end{cases} \tag{1}$$

(Here, $\langle q' \,|\, V_\sigma \,|\, q\rangle$ denotes the coefficient of $|q'\rangle$ in $V_\sigma\,|q\rangle$.) We see from Proposition 1 that M is well-formed if and only if

$$\sum_{q'} \overline{\langle q' \,|\, V_\sigma \,|\, q_1\rangle}\,\langle q' \,|\, V_\sigma \,|\, q_2\rangle = \begin{cases} 1 & q_1 = q_2 \\ 0 & q_1 \neq q_2, \end{cases}$$

for each $\sigma \in \Gamma$. Equivalently, M is well-formed when every V_σ is unitary.

Example: a 2qfa for a^*b^*

To illustrate the above method, we will show how a 2qfa for the language a^*b^* can be defined. (Note that it is not immediate that there is a well-formed 2qfa for this language, as a "typical" 1dfa or 2dfa for a^*b^* will likely not be reversible.)

Define $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ as follows. Let $Q = \{q_0, q_1, q_2, q_3, q_4\}$, $\Sigma = \{a, b\}$, $Q_{acc} = \{q_3\}$ and $Q_{rej} = \{q_4\}$. Define

$$\begin{array}{lll}
V_{¢}\,|q_0\rangle = |q_0\rangle, & V_a\,|q_0\rangle = |q_0\rangle, & V_b\,|q_0\rangle = |q_1\rangle, \\
V_{¢}\,|q_1\rangle = |q_2\rangle, & V_a\,|q_1\rangle = |q_2\rangle, & V_b\,|q_1\rangle = |q_0\rangle, \\
V_{¢}\,|q_2\rangle = |q_4\rangle, & V_a\,|q_2\rangle = |q_4\rangle, & V_b\,|q_2\rangle = |q_2\rangle, \\
V_{¢}\,|q_3\rangle = |q_3\rangle, & V_a\,|q_3\rangle = |q_3\rangle, & V_b\,|q_3\rangle = |q_3\rangle, \\
V_{¢}\,|q_4\rangle = |q_1\rangle, & V_a\,|q_4\rangle = |q_1\rangle, & V_b\,|q_4\rangle = |q_4\rangle,
\end{array}$$

$$\begin{array}{ll}
V_{\$}\,|q_0\rangle = |q_1\rangle, & D(q_0) = +1, \\
V_{\$}\,|q_1\rangle = |q_0\rangle, & D(q_1) = -1, \\
V_{\$}\,|q_2\rangle = |q_3\rangle, & D(q_2) = +1, \\
V_{\$}\,|q_3\rangle = |q_2\rangle, & D(q_3) = 0, \\
V_{\$}\,|q_4\rangle = |q_4\rangle, & D(q_4) = 0,
\end{array}$$

and define δ as in (1). Each V_σ is unitary by inspection, so M is well-formed.

Consider first inputs not in a^*b^*. For example, suppose that the input string is "abba", so the corresponding tape x satisfies: $x(0) = ¢$, $x(1) = a$, $x(2) = b$, $x(3) = b$, $x(4) = a$, $x(5) = \$$. We have the following sequence of superpositions when M is run:

$$\begin{aligned}&|q_0,0\rangle \mapsto |q_0,1\rangle \mapsto |q_0,2\rangle\\&\quad\mapsto |q_1,1\rangle \mapsto |q_2,2\rangle \mapsto |q_2,3\rangle \mapsto |q_2,4\rangle \mapsto |q_4,4\rangle.\end{aligned}$$

After each step except for the last, observation with our observable $\mathcal{O}$ yields "non-halting" with certainty, and after the last step the result of the observation is "reject" with certainty (and thus, the input is rejected). Other inputs not in a^*b^* are rejected in a similar manner.

For any input $w \in a^*b^*$, the reader may verify that the machine will enter superposition $|q_3,|w|+1\rangle$ after $|w|+4$ steps, and will not have previously been in a halting state (and will therefore accept w).

Note that many values of $V_\sigma |q_j\rangle$ define transitions which are not encountered during a computation on any input w. Here, we have defined these values arbitrarily in such a way that each V_σ is unitary. In general, we need only specify those values which matter; so long as these vectors are orthonormal, the remaining values can always be assigned in arbitrary fashion so that the resulting operator is unitary.

4 A 2qfa for $\{a^m b^m \mid m \geq 1\}$

In this section, we will show that for any error bound $\epsilon > 0$, there exists a 2qfa which accepts the non-regular language $L = \{a^m b^m \mid m \geq 1\}$ with error bounded by ϵ in linear time.

For each N, define $M_N = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ as follows. Let $\Sigma = \{a, b\}$,

$$\begin{aligned}Q &= \{q_0, q_1, q_2, q_3,\}\\&\quad\cup \{r_{j,k} \mid 1 \leq j \leq N, 0 \leq k \leq \max(j, N-j+1)\}\\&\quad\cup \{s_j \mid 1 \leq j \leq N\},\end{aligned}$$

$Q_{acc} = \{s_N\}$ and $Q_{rej} = \{q_3\} \cup \{s_j \mid 1 \leq j < N\}$. Let each V_σ take the values indicated in figure 1, and extend each to be unitary on $\ell_2(Q)$ (for each σ, the vectors $\{V_\sigma |q\rangle\}$ are orthonormal by inspection). Also define D as in figure 1, and let δ be defined in the manner described in Section 3.

Proposition 2 *Let $w \in \{a,b\}^*$. For every positive integer N, if $w \in \{a^m b^m \mid m \geq 1\}$ then M_N accepts w with probability 1, and otherwise M_N rejects w with probability at least $1 - 1/N$. In either case M_N halts after $O(N|w|)$ steps with certainty.*

Proof. The computation of each M_N consists of two phases. The first phase rejects any input not of the form $a^u b^v$ for $u, v \geq 1$, and the second phase rejects, with some probability, those inputs for which $u \neq v$.

The first phase is straightforward, similar to the example in section 3. If the input is not of the indicated form, a reject state is entered and the computation ends. Otherwise, the second phase begins with the machine in state q_2 with the tape head reading the right end-marker.

At the start of the second phase, the computation branches into N paths, indicated by the states $r_{1,0}, \ldots, r_{N,0}$, each with amplitude $1/\sqrt{N}$. For each of these paths, the tape head moves deterministically to the left end-marker in the following way: along the jth path, if the tape head reads the symbol a it remains stationary for j steps and then moves left; if it reads the symbol b it remains stationary for $N-j+1$ steps and then it moves left. Thus, on input $a^u b^v$, the tape head requires precisely $(j+1)u + (N-j+2)v + 1$ steps to move from the right to the left end-marker along the jth path. Under the assumption that $j \neq j'$, we have $(j+1)u+(N-j+2)v+1 = (j'+1)u+(N-j'+2)v+1$ if and only $u = v$, from which it follows that any two distinct computation paths will reach the ¢ symbol at the same time if and only if $u = v$ (i.e. the input is of the form $a^m b^m$).

Upon reaching the ¢ symbol, each computation path again splits according to the quantum Fourier transform, yielding either the single accepting state s_N or one of the rejecting states $\{s_j \mid 1 \leq j < N\}$.

Consider first the case that the input is of the form $a^m b^m$. Since each of the N computation paths reaches the ¢ symbol at the same time, we have that the superposition of the machine immediately after performing the quantum Fourier transform is

$$\frac{1}{N}\sum_{j=1}^{N}\sum_{l=1}^{N} \exp\left(\frac{2\pi i}{N}\, j\, l\right) |s_l, 0\rangle = |s_N, 0\rangle.$$

Hence, our observable yields the result *accept* with probability 1.

Now suppose that the input is not of the form $a^m b^m$. Each of the N computation paths reaches the ¢ symbol at a different time, and so there is no cancellation between the rejecting states. For each of the N possible path lengths, the conditional probability that an observation results in *accept* at the time corresponding to that path length is $1/N$, given that some halting state was observed. It follows that the total probability that an observation results in *accept* is also $1/N$, and consequently the input is rejected with probability $1 - 1/N$.

Each possible computation path clearly has length $O(N|w|)$. Since each path ends in a halting configuration, M_N must halt after $O(N|w|)$ steps with certainty. ■

$$
\begin{array}{ll}
V_{¢}\,|q_0\rangle = |q_0\rangle, & V_{\$}\,|q_0\rangle = |q_3\rangle, \\
V_{¢}\,|q_1\rangle = |q_3\rangle, & V_{\$}\,|q_2\rangle = \frac{1}{\sqrt{N}}\sum_{j=1}^{N}|r_{j,0}\rangle, \\
V_{¢}\,|r_{j,0}\rangle = \frac{1}{\sqrt{N}}\sum_{l=1}^{N}\exp\left(\frac{2\pi i}{N}\,j\,l\right)|s_l\rangle,\ 1\le j\le N, & \\
 & \\
V_a\,|q_0\rangle = |q_0\rangle, & V_b\,|q_0\rangle = |q_1\rangle, \\
V_a\,|q_1\rangle = |q_2\rangle, & V_b\,|q_2\rangle = |q_2\rangle, \\
V_a\,|q_2\rangle = |q_3\rangle, & V_b\,|r_{j,0}\rangle = |r_{j,N-j+1}\rangle,\ 1\le j\le N, \\
V_a\,|r_{j,0}\rangle = |r_{j,j}\rangle,\ 1\le j\le N, & V_b\,|r_{j,k}\rangle = |r_{j,k-1}\rangle,\ 1\le k\le N-j+1,\ 1\le j\le N, \\
V_a\,|r_{j,k}\rangle = |r_{j,k-1}\rangle,\ 1\le k\le j,\ 1\le j\le N, & \\
 & \\
D(q_0) = +1, & D(r_{j,0}) = -1,\ 1\le j\le N, \\
D(q_1) = -1, & D(r_{j,k}) = 0,\ 1\le j\le N,\ k\ne 0, \\
D(q_2) = +1, & D(s_j) = 0,\ 1\le j\le N. \\
D(q_3) = 0, &
\end{array}
$$

Figure 1: Specification of the transition function of M_N.

This method can be extended to show that non-context-free languages can be recognized by bounded error, linear time 2qfa's as well.

Corollary 3 *For each $\epsilon > 0$, there exists a 2qfa M which recognizes the language $\{a^m b^m c^m \mid m \geq 1\}$ with one-sided error bounded by ϵ, and which halts in linear time.*

Proof. [Sketch] For each N, a 2qfa M can be defined which functions similarly to M_N above, except that M runs in three phases rather than two: M first checks to see that the input is of the form $a^+b^+c^+$, then checks the last part of the string to see that it is in $\{b^m c^m \mid m \geq 1\}$, and finally checks the initial part of the string to see that it is in $\{a^m b^m \mid m \geq 1\}$. Details will appear in the final version of this paper. ∎

5 Reversible simulation of 1dfa's

In this section, we use a technique from a recent result due to Lange, McKenzie and Tapp [14] regarding space-efficient reversible simulation of deterministic Turing machines to show that an arbitrary 1-way deterministic finite automaton (1dfa) can be simulated by a 2-way reversible finite automaton (2rfa) (which we may simply define as a well-formed 2qfa whose transition amplitudes may only take the values 0 and 1). Here, the construction is considerably simpler than in the Turing machine case.

A 1dfa can be formally specified by a quintuple $A = (S, \Sigma, \mu, s_0, F)$ in the familiar way (see [15], for example). Given such an A, we define a 2qfa $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ which will accept the same language as A.

First, in order to allow M to behave correctly when reading the ¢ and \$ symbols, extend S and μ to S' and μ' as follows (essentially converting A to an equivalent 2dfa). Let $S' = S \cup \{s'_0, s_{acc}, s_{rej}\}$, where s'_0, s_{acc} and s_{rej} are not elements of S, and define

$$
\mu'(s,\sigma) = \begin{cases} \mu(s,\sigma) & s \in S,\ \sigma \in \Sigma \\ s_0 & s = s'_0,\ \sigma = ¢ \\ s_{acc} & s \in F,\ \sigma = \$ \\ s_{rej} & s \in S \backslash F,\ \sigma = \$. \end{cases}
$$

For all other values, let μ' be undefined. Also define

$$
\begin{aligned}
I_{s,\sigma} &= \{s' \in S \mid \mu'(s',\sigma) = \mu'(s,\sigma)\}, \\
J_{s,\sigma} &= \{s' \in S \mid \mu'(s',\sigma) = s\},
\end{aligned}
$$

and fix some ordering of the set S'. Let *max* and *min* denote the maximum and minimum functions relative to this ordering, and for any subset $T \subseteq S'$ let $\mathrm{succ}(s, T)$ be the least element larger than s in T (assuming there is such an element).

Now, we may define M. Let $Q = S' \times \{-1, +1\}$, and let $q_0 = (s'_0, +1)$, $Q_{acc} = \{(s_{acc}, +1)\}$ and $Q_{rej} = \{(s_{rej}, +1)\}$. We will again use the technique from Section 3 to define δ. For each $s \in S'$ and $\sigma \in \Gamma$

for which $\mu'(s,\sigma)$ is defined, let

$$V_\sigma \left|(s,+1)\right\rangle = \begin{cases} \left|(\mathrm{succ}(s, I_{s,\sigma}), -1)\right\rangle & s \neq \max(I_{s,\sigma}) \\ \left|(\mu'(s,\sigma), +1)\right\rangle & s = \max(I_{s,\sigma}), \end{cases}$$

and for every $s \in S'$ and $\sigma \in \Gamma$ let

$$V_\sigma \left|(s,-1)\right\rangle = \begin{cases} \left|(s,+1)\right\rangle & J_{s,\sigma} = \emptyset \\ \left|(\min(J_{s,\sigma}), -1)\right\rangle & J_{s,\sigma} \neq \emptyset. \end{cases}$$

Note that each V_σ can be extended to be a permutation of $\{ |q\rangle \mid q \in Q\}$, inducing a unitary operator on $\ell_2(Q)$. Define $D((s,+1)) = +1$, $D((s,-1)) = -1$ and let δ be defined as in (1).

Proposition 4 *For any 1dfa A, let M be as defined above. For any $w \in \Sigma^*$, if A accepts w then M accepts w in $O(|w|)$ steps, and if A does not accept w, then M rejects w in $O(|w|)$ steps.*

Proof. Viewing A as a 2dfa which only moves its tape head to the right, we have that the set of configurations of A on any input w of length n is $S' \times \mathbb{Z}_{n+2}$. For given A and w, let G be an undirected graph with set of vertices $S' \times \mathbb{Z}_{n+2}$, and an edge between vertices (s_1, k) and $(s_2, k+1)$ if and only if $\mu'(s_1, w_k) = s_2$ (i.e. G is an undirected graph representing the "yields" relation of A on input w). Let G_0 be the connected component of G which contains the initial configuration $(s'_0, 0)$. There can be no cycles in G_0, and G_0 must contain exactly one vertex corresponding to a halting state (s_{acc} or s_{rej}). So, we may view G_0 as being a tree with the single halting configuration vertex as the root and the leaves of the tree including the vertex representing the initial configuration (as well as possibly many other configurations which have no predecessors). M simulates A on input w by traversing G_0 in a reversible manner.

The specific manner in which M performs this traversal is now described. For each configuration (s,k) of A, there correspond two configurations of M: $((s,+1),k)$ and $((s,-1),k-1)$, which are to be interpreted as follows. When M is in configuration $((s,+1),k)$, this indicates that the subtree of G_0 rooted at vertex (s,k) has just been traversed, and when M is in configuration $((s,-1),k-1)$, the subtree of G_0 rooted at vertex (s,k) is now about to be traversed. Consider figure 2. Here we have $J_{s,w_k} = I_{s'_i,w_k} = \{s'_1, \ldots, s'_l\}$ for each $i = 1, \ldots, l$, and we assume that $s'_1 < s'_2 < \cdots < s'_l$ according to our ordering of S'. Suppose that M

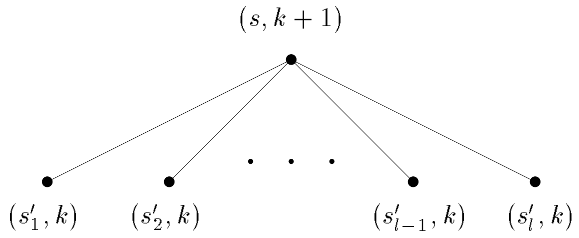

Figure 2: Vertex $(s, k+1)$ and its children.

is in configuration $((s'_i,+1),k)$ for $i < l$. Since $s'_i \neq \max(I_{s'_i,w_k})$, the next configuration of M is $((\mathrm{succ}(s'_i, I_{s'_i,w_k}),-1),k-1) = ((s'_{i+1},-1),k-1)$. (And now the tree rooted at (s'_{i+1},k) is about to be traversed.) Now suppose that M is in configuration $((s'_l,+1),k)$. Since $s'_l = \max(I_{s'_l,w_k})$, the next configuration will be $((\mu(s'_l,w_k),+1),k+1) = ((s,+1),k+1)$. Hence, M enters configuration $((s,+1),k+1)$ only after each of the subtrees rooted at its children have been traversed. Next, suppose that M is in configuration $((s,-1),k)$. The next configuration of M is $((\min(J_{s,w_k}),-1),k-1) = ((s'_1,-1),k-1)$, and so the subtree rooted at vertex (s'_1,k) is now to be traversed. Finally, in the case that $(s,k+1)$ has no predecessors (such as when $k=0$ and $s \neq s_0$), we have $J_{s,w_k} = \emptyset$, and so the configuration of M which immediately follows $((s,-1),k)$ is $((s,+1),k+1)$. (The subtree rooted at $(s,k+1)$ consists of a single vertex in this case, and hence has been traversed.)

By traversing G_0 in this manner, M will eventually enter one of the two configurations $((s_{acc},+1),0)$ or $((s_{rej},+1),0)$, and consequently accepts or rejects accordingly. It is clear that M halts after $O(|w|)$ steps, since there are $O(|w|)$ configurations of M and no configuration may be entered more than once before a halting configuration is reached. (This is true of any 2rfa which eventually halts.) ∎

Since any regular language can be recognized by some 1dfa, we have the following result.

Corollary 5 *For any regular language L, there exists a 2rfa which recognizes L.*

6 1-way quantum finite automata

In this section, we will briefly discuss 1-way quantum finite automata (1qfa's). In contrast to the fact that 2-way quantum finite automata are more powerful than classical 2-way finite automata, 1qfa's are shown to be strictly less powerful than classical 1-way finite automata.

In the interest of simplicity, we will define 1qfa's to be that subset of all 2qfa's for which the tape head may only move to the right, but with the added restriction that the computation may continue for only $|w|+2$ steps on input w (hence each symbol, including the two end-markers, is read exactly once). The end-markers are not superfluous for 1qfa's; the existence of the left end-marker allows the model to simulate the situation in which the result of an observation does not necessarily correspond directly to a classical state (i.e. the observable may correspond to any orthogonal decomposition $\ell_2(Q) = E_{acc} \oplus E_{rej} \oplus E_{non}$), and the existence of the right end-marker allows the model to simulate the situation in which a second observable may be used after the entire input has been read.

It is not difficult to show that for any well-formed 1qfa $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$, there must be a unitary operator V_σ acting on $\ell_2(Q)$, for each $\sigma \in \Gamma$, such that

$$\delta(q, \sigma, q', d) = \begin{cases} \langle q' \,|\, V_\sigma \,|\, q \rangle & d = 1 \\ 0 & d \neq 1. \end{cases}$$

This fact yields a means by which analysis of 1qfa's is simplified, which we will now discuss.

Define $\mathcal{V} = \ell_2(Q) \times \mathbb{R} \times \mathbb{R}$. Elements of $\mathcal{V}$ will represent total states of M as follows: a machine described by $(\psi, p_{acc}, p_{rej}) \in \mathcal{V}$ has, thus far in its computation, accepted with probability p_{acc}, rejected with probability p_{rej} and neither with probability $\|\psi\|_2^2$, in which case the current superposition of internal states is $|\psi\rangle$ (normalized). Let P_{acc}, P_{rej} and P_{non} be the projections of $\ell_2(Q)$ onto $span\{\,|q\rangle \mid q \in Q_{acc}\}$, $span\{\,|q\rangle \mid q \in Q_{rej}\}$, and $span\{\,|q\rangle \mid q \in Q_{non}\}$ respectively. For each $\sigma \in \Gamma$, the evolution of M as symbol σ is read can be described by an operator T_σ on $\mathcal{V}$, defined as follows.

$$T_\sigma : (\psi, p_{acc}, p_{rej}) \mapsto \left(P_{non} V_\sigma \psi, p_{acc} + \|P_{acc} V_\sigma \psi\|_2^2, p_{rej} + \|P_{rej} V_\sigma \psi\|_2^2\right).$$

For $x = \sigma_1 \cdots \sigma_n \in \Gamma^*$, define $T_x = T_{\sigma_n} \cdots T_{\sigma_1}$. So, for example, if $(\psi, p_{acc}, p_{rej}) = T_{¢w\$}\left(|q_0\rangle, 0, 0\right)$, then M accepts w with probability p_{acc}, etc.

Define a norm on $\mathcal{V}$ as

$$\|(\psi, p_{acc}, p_{rej})\| = \frac{1}{2}(\|\psi\|_2 + |p_{acc}| + |p_{rej}|),$$

and let $\mathcal{B} = \{v \in \mathcal{V} \mid \|v\| \leq 1\}$. Clearly, any v which represents the state of a valid 1qfa must be in $\mathcal{B}$. A straightforward calculation reveals that there exists a fixed constant c such that $\|T_x v - T_x v'\| \leq c\|v - v'\|$ for every $v, v' \in \mathcal{B}$ and $x \in \Gamma^*$. Furthermore, it can be shown that if a set $A \subseteq \mathcal{B}$ satisfies the property that there exists a $\xi > 0$ such that for all $v, v' \in A$ we have $\|v - v'\| > \xi$, then there can be at most finitely many elements in A.

Proposition 6 *Let L be any language recognized by a 1qfa with bounded error. Then L is regular.*

Proof. [Sketch] The proof is essentially the same as the proof of Theorem 3 in [21], adjusted to the quantum case.

Let $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ be a 1qfa which recognizes L with probability of error bounded by $1/2 - \epsilon$. Write $w \equiv_L w'$ if, for all $y \in \Sigma^*$, we have $wy \in L$ if and only if $w'y \in L$. The relation $\equiv_L$ is an equivalence relation, and partitions Σ^* into finitely many equivalence classes if and only if L is regular [20].

Let $W \subseteq \Sigma^*$ be any set of strings which are pairwise inequivalent with respect to $\equiv_L$. In order to prove the proposition, it suffices to show that W must be finite. For $w \neq w' \in W$, there must exists $y \in \Sigma^*$ such that $wy \in L$ if and only if $w'y \notin L$. Hence, for $v = T_{¢w}(|q_0\rangle, 0, 0)$ and $v' = T_{¢w'}(|q_0\rangle, 0, 0)$ we have $\|T_{y\$}v - T_{y\$}v'\| > 2\epsilon$, since M has error probability bounded away from $1/2$ by ϵ. Consequently, we have $\|v - v'\| > 2\epsilon/c$, so that the set $\{T_{¢w}\left(|q_0\rangle, 0, 0\right) \mid w \in W\}$ must be finite. From this it follows that W must be finite as well. ∎

Finally, we note that the containment of the class of languages recognized by 1qfa's with bounded error in the regular languages is proper. One rather simple example of a regular language not recognizable by a bounded error 1qfa is provided by the following proposition.

Proposition 7 *The language $L = \{a, b\}^* a$ cannot be recognized by a 1qfa with bounded error.*

Proof. Let M be a 1qfa that recognizes L. For each $x = \sigma_1 \cdots \sigma_n \in \Gamma^*$, write

$$\psi_x = (P_{non} V_{\sigma_n}) \cdots (P_{non} V_{\sigma_1}) \,|q_0\rangle,$$

and let $\mu = \inf\{\|\psi_{¢w}\| \mid w \in \{a, b\}^*\}$. Since $wa \in L$ and $wb \notin L$ for each w, we can conclude that if $\mu = 0$, then M does not recognize L with bounded error. So assume $\mu > 0$. Let $\xi > 0$, and choose w such that $\|\psi_{¢w}\| < \mu + \xi$. It follows that $\|\psi_{¢wy}\| \in [\mu, \mu + \xi)$ for every $y \in \{a, b\}^*$. In particular, for any nonnegative integer j we have

$$\left\|(P_{non} V_b)^j \, \psi_{¢wa}\right\| \in [\mu, \mu + \xi). \tag{2}$$

The sequence $\left\{(P_{non}V_b)^j \psi_{¢wa}\right\}$ is a bounded sequence in a finite dimensional Hilbert space, and must therefore have a limit point. Thus, there must exist integers $j \geq 0$ and $k \geq 1$ such that $\left\|(P_{non}V_b)^j \left(\psi_{¢wa} - (P_{non}V_b)^k \psi_{¢wa}\right)\right\| < \xi$. Using (2), it can be shown that this implies that there is a fixed constant c' (independent of ξ) such that $\left\|\psi_{¢wa} - (P_{non}V_b)^k \psi_{¢wa}\right\| < c'\xi^{1/4}$, and from this it follows that

$$\left\|T_{¢wa\$}\left(\left|q_0\right\rangle, 0, 0\right) - T_{¢wab^k\$}\left(\left|q_0\right\rangle, 0, 0\right)\right\| < c''\xi^{1/4},$$

for fixed c''. Since ξ may be chosen arbitrarily small, and since M must accept wa and reject wab^k, M cannot have error probability bounded away from 1/2. ∎

7 Open problems

A number of questions have been left open by this paper. One interesting question is whether there are languages recognizable by polynomial time, bounded error 2qfa's but not by bounded error (exponential time) 2pfa's (or vice versa). Along similar lines, are there languages recognized by exponential time 2qfa's but not polynomial time 2qfa's? Various generalizations of 2qfa's, such as multi-head 2qfa's, multi-dimensional 2qfa's and 2qfa's with more general types of observables can be defined, as can quantum analogues of other devices based on finite automata, such as interactive proof systems with 2qfa verifiers. What are the relations of these models to one another and to their classical counterparts?

Acknowledgments

We would like to thank Eric Bach, Anne Condon, Katalin Friedl, Mike Siff, and Gábor Tardos for their helpful comments and discussions.

References

[1] E. Bernstein and U. Vazirani. Quantum complexity theory (preliminary abstract). In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 11–20, 1993.

[2] A. Berthiaume. Quantum computation. In *Complexity Theory Retrospective II*. Springer-Verlag, 1997.

[3] A. Berthiaume and G. Brassard. The quantum challenge to structural complexity theory. In *Proceedings of the 7th Annual IEEE Conference on Structure in Complexity*, pages 132–137, 1992.

[4] G. Brassard and P. Høyer. An exact quantum polynomial-time algorithm for Simon's problem. Preprint. To appear in *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems (ISTCS'97)*, 1997.

[5] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London*, A400: 97–117, 1985.

[6] D. Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London*, A425: 73–90, 1989.

[7] D. Deutsch and R. Jozsa. Rapid solutions of problems by quantum computation. *Proceedings of the Royal Society of London*, A439: 553–558, 1992.

[8] C. Dürr, H. Lê Thanh and M. Santha. A decision procedure for well-formed linear quantum cellular automata. In *Proceedings of the Thirteenth Symposium on Theoretical Aspects of Computer Science*, pages 281–292, 1996.

[9] C. Dwork and L. Stockmeyer. On the power of 2-way probabilistic finite state automata. In Proceedings of the 30th Annual Symposium on Foundations of Computer Science, pages 480–485, 1989.

[10] C. Dwork and L. Stockmeyer. A time-complexity gap for two-way probabilistic finite state automata. *SIAM Journal of Computing*, 19: 1011–1023, 1990.

[11] R. Freivalds. Probabilistic two-way machines. In *Proceedings of the International Symposium on Mathematical Foundations of Computer Science*, volume 188 of *Lecture Notes in Computer Science*, pages 33–45. Springer-Verlag, 1981.

[12] L. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pages 212–219, 1996.

[13] R. Ladner, R. Lipton and L. Stockmeyer. Alternating pushdown and stack automata. *SIAM Journal on Computing*, 13(1): 135–155, 1984.

[14] K. Lange, P. McKenzie and A. Tapp. Reversible space equals deterministic space (extended abstract). In *Proceedings of the 12th IEEE Conference on Computational Complexity*, 1997. To appear.

[15] H. Lewis and C. Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall, 1981.

[16] S. Lloyd. A potentially realizable quantum computer. *Science*, 261: 1569–1571, 1993.

[17] N. Margolus. Quantum computation. *Annals of the New York Academy of Science*, 480: 287–297, 1986.

[18] C. Moore and J. Crutchfield. Quantum automata and quantum grammars. Santa Fe Institute Working Paper 97-07-062, 1997.

[19] J. Pin. On the languages accepted by finite reversible automata. In *14th International Colloquium on Automata, Languages and Programming*, volume 267 of *Lecture Notes in Computer Science*, pages 237–249. Springer-Verlag, 1987.

[20] M. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3: 114–125, 1959.

[21] M. Rabin. Probabilistic automata. *Information and Control*. 6: 230–245, 1963.

[22] J. Shepherdson. The reduction of two-way automata to one-way automata. *IBM Journal of Research and Development*, 3: 198–200, 1959.

[23] P. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[24] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. Preprint, 1996.

[25] D. Simon. On the power of quantum computation. In *35th Annual Symposium on Foundations of Computer Science*, pages 116–123, 1994.

[26] J. Watrous. On one-dimensional quantum cellular automata. In *36th Annual Symposium on Foundations of Computer Science*, pages 528–537, 1995.

[27] A. Yao. Quantum circuit complexity. In *34th Annual Symposium on Foundations of Computer Science*, pages 352–361, 1993.