On the Power of 2-Way Quantum Finite State Automata

John Watrous

Technical Report #1350

April 1997

On the Power of 2-Way Quantum Finite State Automata

John Watrous
Computer Sciences Department
University of Wisconsin
Madison, Wisconsin 53706
watrous@cs.wisc.edu

April 28, 1997

Abstract

In this paper, we introduce 2-way quantum finite state automata (2qfa's), which are the quantum analogue of deterministic, nondeterministic and probabilistic 2-way finite state automata (2dfa's, 2nfa's and 2pfa's). We prove the following facts regarding 2qfa's.

1. For any $\epsilon > 0$, there exists a 2qfa M which recognizes the non-regular language $L = \{a^m b^m \,|\, m \geq 1\}$ with (one-sided) error bounded by ϵ, and which halts in polynomial 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 exists a reversible (hence quantum) 2-way finite state automaton M for L running in polynomial 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 polynomial time, bounded error 2qfa's properly includes the regular languages. Since it is known that 2dfa's, 2nfa's and polynomial expected time, bounded error 2pfa's can recognize only regular languages, it follows that 2qfa's are strictly more powerful than these "classical" models.

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 [22, 23]. Other results include Grover's quantum searching algorithm [12] and various oracle results regarding the power of quantum computers [3, 7, 1, 24, 4]. (For further information regarding quantum computation, the reader is referred to [5, 1, 2].)

The above examples regard the power of universal quantum machines (e.g. quantum Turing machines [5, 1], quantum circuits [6, 26], quantum cellular automata [18, 17, 25, 8]). In this paper, we define a new, much more restricted quantum computational model: 2-way quantum finite state automata (2qfa's). 2qfa's 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 [20, 21, 14], 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 $\{a^m b^m\}$ defined by Freivalds require exponential expected time, and it was subsequently shown by Dwork and Stockmeyer [9, 10] and independently by Kaneps and Freivalds [13] 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.

In this paper, we show that 2qfa's are strictly more powerful than 2dfa's, 2nfa's and 2pfa's in the sense that polynomial 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 exists a 2qfa M which recognizes the non-regular language $L = \{a^m b^m \,|\, m \geq 1\}$ with (one-sided) error bounded by ϵ, and which halts in polynomial 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 exists a reversible (hence quantum) 2-way finite state automaton M for L running in polynomial time.

We prove 1 simply by exhibiting a sequence of polynomial time 2qfa's for $\{a^m b^m\}$ 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, polynomial 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 [15] regarding reversible simulation of deterministic Turing machines to the finite automaton case. As a corollary of 2, we have 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].

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, polynomial time 2qfa's for the language $\{a^m b^m\}$, and in Section 5 we prove that any regular language can be recognized by a 2-way reversible (hence quantum) finite state automaton. Finally, we conclude with Section 6 which mentions some related open problems.

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. It is assumed that $q_0 \notin Q_{acc} \cup Q_{rej}$ and $Q_{acc} \cap Q_{rej} = \emptyset$. Elements of Q_{acc} and Q_{rej} are *halting states* and all other elements of Q are *non-halting states*. In addition to the input symbols Σ, there are two symbols ¢, $\$ \notin \Sigma$ 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 $\Gamma = \Sigma \cup \{¢, \$\}$.

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 ¢ and \$, introduces unnecessary complications in the quantum case. The contents of any tape can therefore be described by a mapping $x : \mathbb{Z}_n \to \Gamma$, 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 $|\psi\rangle = \sum_{c \in C_n} \alpha_c |c\rangle$, we say that 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.

$$U_\delta^x |q, k\rangle = \sum_{\substack{q' \in Q \\ d \in \{-1,0,1\}}} \delta(q, x(k), q', d) \; |q', k + d \pmod{|x|}\rangle$$

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.

Definition 1 A 2qfa M is *well-formed* if and only if U_δ^x is unitary for every tape x.

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 and languages recognized by 2qfa's

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 $\left\| |\psi_j\rangle \right\|^2$.
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.

A simple example of an observable is to let $c_1, \ldots, c_k$ be an enumeration of C_n, and let $E_j = \text{span}\left\{|c_j\rangle\right\}$. (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 somewhat 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 $C_{non} = Q \backslash (Q_{acc} \cup Q_{rej}) \times \mathbb{Z}_n$, and let $E_{acc} = \text{span}\left\{|c\rangle : c \in C_{acc}\right\}$, $E_{rej} = \text{span}\left\{|c\rangle : c \in C_{rej}\right\}$ and $E_{non} = \text{span}\left\{|c\rangle : c \in C_{non}\right\}$. 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).

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 form $x_w(0) = ¢$, $x_w(|w| + 1) = \$$

and $x_w(i) = w_i$ for $1 \leq i \leq |w|$. 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 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 $\mathcal{O}$. 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 by small values of $\epsilon > 0$.

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

Proof. For each x, U_δ^x is unitary if and only if $\{U_\delta^x |q, k\rangle : s \in Q, k \in \mathbb{Z}_{|x|}\}$ is an orthonormal set. 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 every x, q_1, q_2 and k, with $|x| \geq 5$. For $3 \leq |x| \leq 4$, it can be shown that conditions 2 and 3 are sufficient to yield $U_\delta^x |q_1, k\rangle \perp U_\delta^x |q_2, k+1\rangle$ and $U_\delta^x |q_1, k\rangle \perp U_\delta^x |q_2, k+2\rangle$. Since the tape head of a 2qfa cannot move more than one square per step, $U_\delta^x |q_1, k_1\rangle \perp U_\delta^x |q_2, k_2\rangle$ whenever k_1 and k_2 are more than two squares away, regardless of how δ is defined. Thus, $\{U_\delta^x |q, k\rangle : s \in Q, k \in \mathbb{Z}_{|x|}\}$ is an orthonormal set 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, as before, 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

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

and

$$D(q_0) = +1, \quad D(q_1) = -1, \quad D(q_2) = +1, \quad D(q_3) = 0, \quad D(q_4) = 0,$$

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:

$$|q_0, 0\rangle \mapsto |q_0, 1\rangle \mapsto |q_0, 2\rangle \mapsto |q_1, 1\rangle \mapsto |q_2, 2\rangle \mapsto |q_2, 3\rangle \mapsto |q_2, 4\rangle \mapsto |q_4, 4\rangle.$$

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 $\{a^m b^m\}$

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 polynomial time. Specifically, for each $k \in \mathbb{N}$ we will define a 2qfa M_k which accepts strings in L with probability 1, and rejects strings not in L with probability at least $1 - 4^{-k}\binom{2k}{k}$. Since $4^{-k}\binom{2k}{k} \to 0$ as $k \to \infty$, the error probability can be made arbitrarily small by appropriate choice of k.

For fixed $k \in \mathbb{N}$, define

$$Q = \{q_0, q_1, q_2, q_3, q_{rej}\} \cup \left\{ r_0^{(j)}, \ldots, r_8^{(j)}, r_{rej}^{(j)} \;\middle|\; j = 1, \ldots, k \right\} \cup \{r_{acc}\},$$

take $\Sigma = \{a, b\}$ and define $Q_{acc} = \{r_{acc}\}$ and $Q_{rej} = \{q_{rej}\} \cup \left\{ r_{rej}^{(j)} \;\middle|\; j = 1, \ldots, k \right\}$. Define

$$\begin{array}{llll}
V_{¢}|q_0\rangle = |q_0\rangle, & V_a|q_0\rangle = |q_0\rangle, & V_b|q_0\rangle = |q_1\rangle, & V_{\$}|q_0\rangle = |q_{rej}\rangle, \\
V_{¢}|q_1\rangle = |q_{rej}\rangle, & V_a|q_1\rangle = |q_2\rangle, & V_b|q_2\rangle = |q_2\rangle, & V_{\$}|q_2\rangle = |q_3\rangle, \\
 & V_a|q_2\rangle = |q_{rej}\rangle, & V_b|q_3\rangle = |q_3\rangle, & \\
 & V_a|q_3\rangle = \left|r_0^{(1)}\right\rangle, & & \\
V_{¢}\left|r_4^{(j)}\right\rangle = \left|r_6^{(j)}\right\rangle, & V_a\left|r_0^{(j)}\right\rangle = \frac{1}{\sqrt{2}}\left|r_1^{(j)}\right\rangle + \frac{1}{\sqrt{2}}\left|r_2^{(j)}\right\rangle, & V_b\left|r_2^{(j)}\right\rangle = \left|r_4^{(j)}\right\rangle, & V_{\$}\left|r_3^{(j)}\right\rangle = \left|r_5^{(j)}\right\rangle, \\
 & V_a\left|r_1^{(j)}\right\rangle = \left|r_3^{(j)}\right\rangle, & V_b\left|r_3^{(j)}\right\rangle = \left|r_3^{(j)}\right\rangle, & \\
 & V_a\left|r_4^{(j)}\right\rangle = \left|r_4^{(j)}\right\rangle, & V_b\left|r_5^{(j)}\right\rangle = \left|r_5^{(j)}\right\rangle, & \\
 & V_a\left|r_5^{(j)}\right\rangle = \left|r_7^{(j)}\right\rangle, & V_b\left|r_6^{(j)}\right\rangle = \left|r_8^{(j)}\right\rangle, & \\
 & V_a\left|r_6^{(j)}\right\rangle = \left|r_6^{(j)}\right\rangle, & & \\
 & V_a\left|r_7^{(j)}\right\rangle = \frac{1}{\sqrt{2}}\left|r_0^{(j+1)}\right\rangle + \frac{1}{\sqrt{2}}\left|r_{rej}^{(j)}\right\rangle & \multirow{2}{*}{\Big\} \text{ if } j < k,} & \\
 & V_a\left|r_8^{(j)}\right\rangle = \frac{1}{\sqrt{2}}\left|r_0^{(j+1)}\right\rangle - \frac{1}{\sqrt{2}}\left|r_{rej}^{(j)}\right\rangle & & \\
 & V_a\left|r_7^{(k)}\right\rangle = \frac{1}{\sqrt{2}}|r_{acc}\rangle - \frac{1}{\sqrt{2}}\left|r_{rej}^{(k)}\right\rangle, & & \\
 & V_a\left|r_8^{(k)}\right\rangle = \frac{1}{\sqrt{2}}|r_{acc}\rangle + \frac{1}{\sqrt{2}}\left|r_{rej}^{(k)}\right\rangle, & &
\end{array}$$

and extend each V_σ to be unitary on $\ell_2(Q)$ in an arbitrary fashion (for each σ, the vectors $\{V_\sigma|q\rangle\}$ defined above are orthonormal by inspection). Define D as

$$\begin{array}{llllllllllll}
D(q_0) & = & +1, & D(q_1) & = & -1, & D(q_2) & = & +1, & D(q_3) & = & -1, \\
D\left(r_0^{(j)}\right) & = & 0, & D\left(r_3^{(j)}\right) & = & +1, & D\left(r_6^{(j)}\right) & = & +1, & D(q_{rej}) & = & 0, \\
D\left(r_1^{(j)}\right) & = & 0, & D\left(r_4^{(j)}\right) & = & -1, & D\left(r_7^{(j)}\right) & = & 0, & D(r_{acc}) & = & 0, \\
D\left(r_2^{(j)}\right) & = & +1, & D\left(r_5^{(j)}\right) & = & -1, & D\left(r_8^{(j)}\right) & = & -1, & D\left(r_{rej}^{(j)}\right) & = & 0,
\end{array}$$

for each j, and let δ be defined as in (1).

Theorem 2 *Let $w \in \{a,b\}^*$. For every $k \in \mathbb{N}$, if $w \in \{a^m b^m \,|\, m \geq 1\}$ then M_k accepts w with probability 1, and otherwise M_k rejects w with probability at least $1 - 4^{-k}\binom{2k}{k}$. In either case, M_k enters a halting state after $O(k|w|)$ steps.*

Proof. The computation of each M_k consists of two phases. The first phase (indicated by the "q-states", i.e. q_0, q_1, etc.) rejects any input not of the form $a^u b^v$ for $u, v \geq 1$, and the second phase (indicated by the "r-states") rejects, with some probability, those inputs $a^u b^v$ for which $u \neq v$.

The "q-phase" is straightforward (similar to the example in Section 3). The reader may verify that any input w not of the form $a^u b^v$, $u, v \geq 1$, will result in the machine entering a superposition of the form $|q_{rej}, l\rangle$ for some l, and will therefore be rejected. For $w = a^u b^v$, $u, v \geq 1$, the machine will enter superposition $\left|r_0^{(1)}, u\right\rangle$ after precisely $u + 2v + 5$ steps (thus beginning the "r-phase").

Now we will analyse the "r-phase". Assume $w = a^u b^v$, $u, v \geq 1$, and consider first the case $k = 1$. The machine begins this phase in superposition $\left|r_0^{(1)}, u\right\rangle$, so that in one step the machine will evolve into the superposition $\frac{1}{\sqrt{2}}\left|r_1^{(1)}, u\right\rangle + \frac{1}{\sqrt{2}}\left|r_2^{(1)}, u+1\right\rangle$. Supposing that the machine were placed in superposition $\left|r_1^{(1)}, u\right\rangle$, the evolution would proceed as follows:

$$\begin{aligned}\left|r_1^{(1)}, u\right\rangle &\mapsto \left|r_3^{(1)}, u+1\right\rangle \mapsto \cdots \mapsto \left|r_3^{(1)}, u+v+1\right\rangle \\ &\mapsto \left|r_5^{(1)}, u+v\right\rangle \mapsto \cdots \mapsto \left|r_5^{(1)}, u\right\rangle \mapsto \left|r_7^{(1)}, u\right\rangle \mapsto \frac{1}{\sqrt{2}}\left|r_{acc}, u\right\rangle - \frac{1}{\sqrt{2}}\left|r_{rej}^{(1)}, u\right\rangle,\end{aligned}$$

the tape head moving to the right end-marker and back, branching into two possible states when it returns to its original location over the last a. The total number of steps required for this transformation is $2v+4$. Letting U denote the time-evolution operator of M_1 on the tape corresponding to input w, we can write this more concisely as $U^{2v+4}\left|r_1^{(1)}, u\right\rangle = \frac{1}{\sqrt{2}}\left|r_{acc}, u\right\rangle - \frac{1}{\sqrt{2}}\left|r_{rej}^{(1)}, u\right\rangle$. Similarly, we have $U^{2u+4}\left|r_2^{(1)}, u+1\right\rangle = \frac{1}{\sqrt{2}}\left|r_{acc}, u\right\rangle + \frac{1}{\sqrt{2}}\left|r_{rej}^{(1)}, u\right\rangle$; when starting from $\left|r_2^{(1)}, u+1\right\rangle$, the tape head moves to the left end-marker and back rather than to the right. Thus, in the case $u = v\ (= m)$ we have $U^{2m+5}\left|r_0^{(1)}, u\right\rangle = \left|r_{acc}, u\right\rangle$; the computation paths leading to the rejecting configuration interfere destructively, due to the fact that their amplitudes sum to zero. Hence, M_1 accepts w with certainty. However, when $u \neq v$, this interference does not occur since the rejecting configuration is not entered at the same time for the two possible paths. For example, in the case $u < v$, we have $U^{2u+5}\left|r_0^{(1)}, u\right\rangle = \frac{1}{2}\left|r_{acc}, u\right\rangle + \frac{1}{2}\left|r_{rej}^{(1)}, u\right\rangle + \frac{1}{\sqrt{2}}\left|s, l\right\rangle$, where s is one of $r_3^{(1)}$, $r_5^{(1)}$ or $r_7^{(1)}$, depending on the size of v. At this point, our observable results in "accept" with probability 1/4, "reject" with probability 1/4, and "non-halting" with probability 1/2. If the result is either "accept" or "reject", the computation is over; otherwise the computation continues, the superposition collapsing to $|s, l\rangle$. This superposition subsequently evolves into $\frac{1}{\sqrt{2}}\left|r_{acc}, u\right\rangle - \frac{1}{\sqrt{2}}\left|r_{rej}^{(1)}, u\right\rangle$, which results in either "accept" or "reject", each with probability 1/2. Thus, the total probability of acceptance is 1/2 and the total probability of rejection is 1/2.

For $k > 1$, the situation becomes more complicated. Strings $a^u b^v$ for which $u = v$ are still accepted with probability 1, since $U^{2m+5}\left|r_0^{(j)}, u\right\rangle = \left|r_0^{(j+1)}, u\right\rangle$ for $j < k$, and $U^{2m+5}\left|r_0^{(k)}, u\right\rangle = \left|r_{acc}, u\right\rangle$ (similar to above). However, the probability of rejecting strings with

$u \neq v$ does not increase to $1 - 2^{-k}$, say, because we may have unwanted interference based on the fact that multiple computation paths may lead to a given state $r_{rej}^{(j)}$ after the same number of steps. For example, there are four computation paths leading from $\left(r_0^{(1)}, u\right)$ to $\left(r_{rej}^{(2)}, u\right)$; one of length $4u + 10$, two of length $2u + 2v + 10$ and one of length $4v + 10$. Paths will interfere (constructively or destructively) whenever they have the same length; the two paths of length $2u + 2v + 10$ have amplitudes $1/4$ and $-1/4$, and will therefore interfere destructively. In general, there will be 2^j paths from $\left(r_0^{(1)}, u\right)$ to $\left(r_{rej}^{(j)}, u\right)$ with possible lengths $i(2u+5) + (j-i)(2v+5)$ for each $i = 0, \ldots, j$. The amplitude of each such path will be $\pm 2^{-j}$, where the sign depends on the direction which the tape head moved on the last cycle (i.e. negative if the transition from $r_0^{(j)}$ to $r_1^{(j)}$ was performed, and positive if the transition from $r_0^{(j)}$ to $r_2^{(j)}$ was performed). The number of paths of length $i(2u+5) + (j-i)(2v+5)$ is $\binom{j}{i}$, where $\binom{j-1}{i-1}$ of the paths have positive amplitude and $\binom{j-1}{i}$ have negative amplitude (following the convention $\binom{j-1}{j} = \binom{j-1}{-1} = 0$). From this, we can determine that the total probability that the machine rejects is

$$\sum_{j=1}^{k} \sum_{i=0}^{j} \left| 2^{-j} \left(\binom{j-1}{i-1} - \binom{j-1}{i} \right) \right|^2 . \qquad (2)$$

Using well-known binomial sums along with induction, (2) can be simplified to $1 - 4^{-k}\binom{2k}{k}$.

From the above, it is clear that each M_k enters a halting state after $O(k|w|)$ steps. ■

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

Proof. [Sketch] For fixed k, a 2qfa M can be defined which functions similarly to M_k above, except that M runs in 3 phases rather than 2: M first checks to see that the input is of the form $a^+b^+c^+$, then checks the initial part of the string to see that it is in $\{a^m b^m \mid m \geq 1\}$, and finally checks the last part of the string to see that it is in $\{b^m c^m \mid m \geq 1\}$. In short, the second phase of M runs exactly as the second phase of M_k, except that the c symbol is treated as if it were \$, and a new state (q_0', say) is entered instead of the state r_{acc}. The third phase (starting in state q_0' and first moving the tape head to the last b), runs as the second phase does, except treating a as if it were ¢, b as if it were a and c as if it were b.

It is fairly straightforward to define such a machine M, and to verify that the correct probabilities are obtained. Details will appear in the final version of this paper. ■

It is interesting to note that the error probability $4^{-k}\binom{2k}{k} \sim \frac{1}{\sqrt{\pi k}}$ is considerably worse than 2^{-k}, which is the bound that would result if machine M_1 were run k times, accepting if and only if M_1 accepted every time. In essence, the manner in which we have defined 2qfa's, as well as the particular observable we are using, does not allow the machine to "reset" itself in order to do this. By modifying the definition of 2qfa's somewhat, allowing machines to have not only a quantum part, but also a classical part (capable of modifying the state of the entire machine depending on results of observations), the model could be made more robust in this sense. For simplicity, we will not do this in this paper.

5 Reversible simulation of 1dfa

In this section we prove that an arbitrary 1-way deterministic finite automaton (1dfa) can be simulated in polynomial time by a two-way reversible finite automaton (2rfa) (which we may simply define as a well-formed 2qfa whose transition amplitudes are elements of the set $\{0,1\}$). In order to prove this fact, we use a technique from a recent result due to Lange, McKenzie and Tapp [15] regarding space-efficient reversible simulation of deterministic Turing machines. Here, the construction is considerably simpler than in the Turing machine case, due to the fact that 1dfa's are much simpler than Turing machines.

A 1dfa can be formally specified by a quintuple $A = (S, \Sigma, \mu, s_0, F)$ in the familiar way (see [16], 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, we will 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 $I_{s,\sigma} = \{s' \in S \mid \mu'(s',\sigma) = \mu'(s,\sigma)\}$, $J_{s,\sigma} = \{s' \in S \mid \mu'(s',\sigma) = s\}$ 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 $Q_{rej} = \{(s_{rej}, +1)\}$. Finally, we will 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

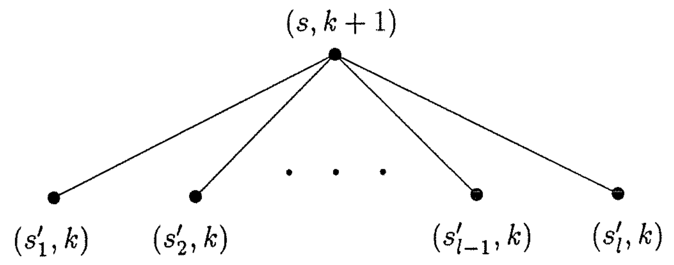
$$V_\sigma \left|(s,-1)\right\rangle = \begin{cases} \left|(s,+1)\right\rangle & J_{s,\sigma} = \emptyset \\ \left|(\min(J_{s,\sigma}), -1)\right\rangle & J_{s,\sigma} \neq \emptyset. \end{cases}$$

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).

Lemma 4 *Let $A = (S, \Sigma, \mu, s_0, F)$ be a 1dfa, let $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ be as defined above and let $w \in \Sigma^*$. If A accepts w then M accepts w in $O(|w|)$ steps. Otherwise, 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 defined as follows. Let G have set of vertices $S' \times \mathbb{Z}_{n+2}$, and for every s_1, s_2 and k let there be an edge between vertices (s_1, k) and $(s_2, k+1)$ if and only if $\mu'(s_1, w_k) = s_2$. In other words, 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 $(s_0', 0)$. A is a deterministic machine which always moves its tape head to the right, so exactly one of the two vertices $(s_{acc}, n+1)$ and $(s_{rej}, n+1)$ must belong to G_0, and there can be no cycles in G_0. We will 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 the appropriate tree G_0 in a reversible manner.

We will now describe the specific manner in which M performs this traversal. 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 the following diagram, representing a vertex $(s, k+1)$ along with its children.



Here we have that $J_{s,w_k} = I_{s_i',w_k} = \{s_1', \ldots, s_l'\}$ 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 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 $((\text{succ}(s_i', I_{s_i',w_k}), -1), k-1)$ $= ((s_{i+1}', -1), k-1)$. (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 the tree G_0 in this manner, M will eventually enter one of the two configurations $((s_{acc}, +1), n+2) = ((s_{acc}, +1), 0)$ or $((s_{rej}, +1), n+2) = ((s_{rej}, +1), 0)$. Consequently, M 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.

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

6 Open problems

One question which has been left open by this paper is whether there is an interesting characterization of the class of languages recognized by bounded error 2qfa's running in polynomial time. Also, the question of how the power of polynomial time, bounded error 2qfa's compares to that of 2pfa's and 2qfa's not restricted to polynomial time, and to other variations of finite automata, is an interesting one.

Another area which may be interesting is the study of 1-way quantum finite automata, which we have not touched on in this paper.

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. To appear.

[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 *28th Annual ACM Symposium on the Theory of Computing*, pages 212–219, 1996.

[13] J. Kaneps and R. Freivalds. Minimal nontrivial space complexity of probabilistic one-way turing machines. In *Proceedings of the Conference on Mathematical Foundations of Computer Science*, volume 452 of *Lecture Notes in Computer Science*, pages 355–361. Springer-Verlag, 1990.

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

[15] 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.

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

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

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

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

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

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

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

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

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