Quantum Finite Automata and Weighted Automata

M. V. Panduranga Rao   V. Vinay

IISc-CSA-TR-2004-9
http://archive.csa.iisc.ernet.in/TR/2004/9/

Computer Science and Automation
Indian Institute of Science, India

October 2004

Quantum Finite Automata and Weighted Automata

M. V. Panduranga Rao* V. Vinay†

Abstract

Two way quantum finite automata derive their strength by exploiting complex valued probability amplitudes and interference therein. This paper is a step towards finding the least powerful model which is purely classical and can capture the dynamics of quantum phase. We give a classical simulation of a restriction of the 2-way Finite Automata with Quantum and Classical States (2QCFA) of Ambainis and Watrous [A. Ambainis, J. Watrous, Two-way finite automata with quantum and classical state, Theoretical Computer Science, 287(1): 299-311 (2002)] using weighted automata over the real semiring.

1 Introduction

Quantum Finite Automata (QFA) have been an area of active research in the recent past. Many inequivalent models of QFA have been proposed. The first, proposed by Moore and Crutchfield [7] and Kondacs and Watrous [6], is the Measure-Once (MO) 1-way QFA. In this model, the finite automaton has a read-only input tape, with the input demarcated by a beginning marker ¢ and an end-marker \$. The machine has a finite set of states Q with $|Q| = k$, a constant. We associate a Hilbert space $\mathcal{H}$ with Q, $\{|q\rangle\} = Q$ forming a basis[1]. A state vector of the MO-1QFA is then a vector of unit norm in this space. A series of unitary transformations, each transformation depending on the input symbol, is applied on an initial quantum state as the input tape is scanned from the beginning to the end. The final state vector is a linear superposition of the basis vectors: $|\psi\rangle = \sum_{q \in Q} \alpha_q |q\rangle$, where $\alpha_q \in \mathbb{C}$ and $\sum_{q\in Q} |\alpha_q|^2 = 1$. On reaching the end-marker, a measurement is

*pandurang@csa.iisc.ernet.in

†vinay@csa.iisc.ernet.in

[1] See Section 2 for a quick introduction to quantum computing.

performed. If the state observed is among those designated as accepting, the input is accepted. Crutchfield and Moore [7] and Brodsky and Pippenger [4] showed that this model accepts only a subset of regular languages with bounded error.

A more powerful model, the Measure-Many (MM) 1-way QFA, was proposed by Kondacs and Watrous [6]. In this model, measurements are made after reading every input symbol. Further, Q is partitioned into *accepting, rejecting* and *non-halting* subspaces. Measurements are carried out in such a way that the outcomes correspond to only the subspaces and not the individual states. If an accepting or rejecting subspace is observed, then the input is accepted or rejected respectively. If the outcome is *non-halting*, the computation proceeds from a normalized vector in the non-halting subspace. Even this model has been shown to accept only a subset of regular languages in [6]. A lot of effort has gone into characterizing the languages accepted by this model, cf. [2], [3], [4]. A still more powerful model, the 2-way quantum finite automaton (2QFA) was also proposed by Kondacs and Watrous [6]. This model allows superpositions where the head can be in many positions simultaneously on the input (of length n). Not only can this model recognize all regular languages, but also some non-regular ones like $L_{eq} = \{a^m b^m \mid m \in \mathbb{N}\}$ with bounded error. However, the $O(\log n)$ qubits required to store the position of the head in the "finite" control make this model costly, and goes against the spirit of finite automata.

Seeking to harness the advantages of both quantum states and the ability of the head of the tape to move both ways, Ambainis and Watrous [1] proposed a model intermediate between 1QFA and 2QFA. This model, called the 2-way finite automata with quantum and classical states (2QCFA) was shown by [1] to recognize L_{eq} and $L_{pal} = \{x \in \{a,b\}^* \mid x = x^R\}$, where x^R is the reverse of x, with bounded error in polynomial and exponential time respectively.

An interesting problem in this context is to identify lesser powerful computational models among the classical ones which can "simulate" the more powerful models of quantum finite automata. In this paper we give a technique of simulating a restricted version of the 2QCFA by weighted automata over the real semiring. Although Ambainis and Watrous define the general 2QCFA in [1], they use only a restricted version to recognize L_{eq} and L_{pal}. As examples, we give simulations of their machines that recognize these two languages. The paper is organized as follows. Section 2 gives a brief introduction to quantum computing, 2QCFA and weighted automata. In Section 3 we give a method of constructing a weighted automaton for any given 2QCFA and apply the method for L_{eq} and L_{pal}. Section 4 concludes the paper.

2 Preliminaries

We begin this section with an introduction to quantum computing. A superposition of k states $\{q_0, q_1, \dots, q_{k-1}\}$ is a vector of unit norm in a k-dimensional Hilbert space $\mathcal{H}$, with $\{|q_0\rangle, |q_1\rangle, \dots, |q_{k-1}\rangle\}$ serving as a basis of elementary unit vectors. Thus, a superposition may be written as $|\psi\rangle = \sum_{i=0}^{k-1} \alpha_i |q_i\rangle$ with each $\alpha_i \in \mathbb{C}$ and $\sum_{i=0}^{k-1} |\alpha_i|^2 = 1$. A unitary operator on $\mathcal{H}$ is a norm preserving linear operator. Application of a unitary operator U on a superposition $|\psi\rangle$ evolves the system to $U|\psi\rangle$. An orthogonal measurement of the system is specified by a set $\{P_i\}$ of operators such that (i) $P_i = P_i^\dagger$ for each i, where $P_i^\dagger$ denotes the adjoint of P_i (ii) $P_i^2 = P_i$ for each i (iii) $P_i P_j = 0$ for $i \neq j$ and (iv) $\sum_i P_i = I$. Measuring a superposition $|\psi\rangle$ through such a set yields i with a probability $||P_i|\psi\rangle||^2$ for each i. The superposition itself collapses to $\frac{1}{||P_i|\psi\rangle||} P_i|\psi\rangle$ for the i that was the yield.

We now define our restriction of the 2QCFA before moving on to weighted automata. For a detailed explanation of 2QCFA and weighted automata and results related to language recognition by them, see [1] and [7] respectively.

Informally, a "Measure at End" ME-2QCFA machine M works as follows. It consists of a classical and a quantum part. The quantum part is described by a Hilbert space of constant size. The head movement is controlled by a classical finite control. Depending on the classical state and the symbol being scanned, the quantum part is subjected to either a unitary transformation or a measurement.

One round of computation consists of scanning the input, performing (a) for every symbol read, a unitary transformation on the quantum part, a change of the classical state and a movement of the head (including the identity transformation, no change of classical state and no movement of the head respectively) (b) a measurement of the quantum part in the end. The outcome of the measurement decides if the input belongs to the language of the machine or not. Since the outcome of a measurement is probabilistic, several rounds may be performed for probability amplification. The model differs from 2QCFA in that the 2QCFA formalism allows measurements in the middle of a computation. Depending on the outcome a 2QCFA machine can either halt, rejecting or accepting the input, or continues with the computation.

2.1 ME-2QCFA

Formally, the ME-2QCFA is a 9 tuple: $M = (Q, S, \Sigma, \Theta, \delta, q_0, s_0, S_{acc}, S_{rej})$ where

- Q and S are finite sets of quantum and classical states respectively.
- Σ is a finite alphabet, $\Gamma = \Sigma \bigcup \{¢, \$\}$ being the tape alphabet with $¢,\$ \notin \Sigma$ as left and right end-markers respectively.
- q_0 and s_0 are the initial quantum and classical states respectively.
- S_{acc}, $S_{rej} \subseteq S$ are the subsets of classical accepting and rejecting states respectively.
- Θ is a two-parameter transition function which governs the evolution of the quantum state. $\Theta(s, \sigma)$, where $s \in S \backslash (S_{acc} \bigcup S_{rej})$ and $\sigma \in \Gamma$, is either a unitary transformation or a measurement.
- δ is a function describing the evolution of the classical state and is defined as follows:

  Unless $\Theta(s, \sigma)$ is a measurement, $\delta(s, \sigma) \in S \times D$, $D = \{-1, 0, 1\}$ where $-1, 0$, and 1 specify the movement of the tape head.

  If $\Theta(s, \sigma)$ is a measurement, $\delta(s, \sigma) \in S_{acc} \bigcup S_{rej}$.

The multi-scan version: A further restriction would be to restrict the head movement to multiple scans of the input only. The head is allowed to move left if and only if it is at the end of the input, and even then, only back to the beginning. The modified transition function is

$\delta(s, \sigma) = (s', 1)$ where $s, s' \in S \backslash (S_{acc} \bigcup S_{rej})$ and $\sigma \in \Sigma \bigcup ¢$ and

$\delta(s, \$)$: if $\Theta(s, \$)$ is a measurement, then the machine goes into a state in $(S_{acc} \bigcup S_{rej})$; else it moves the head back to ¢, the beginning of the input.

Let $p_{acc}(x)$ and $p_{rej}(x)$ be the probabilities that the computation enters an accepting or rejecting state respectively. A given 2QCFA M is said to accept a language $L \subseteq \Sigma^*$ with one-sided error ϵ if for all $x \in \Sigma^*$ we have the following– (a) $p_{acc}(x) + p_{rej}(x) = 1$ (b) $p_{acc}(x) = 1$ if $x \in L$ and $p_{rej}(x) \geq 1 - \epsilon$ if $x \notin L$.

2.2 Weighted Automata

Weighted automata are similar to deterministic finite automata (DFA) but have transitions labelled with weights in addition to symbols of the alphabet. The set to which the weights belong needs to have the algebraic structure of a semiring, for various operations to be well defined.

Definition 1 *A system* $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ *is a semiring if:*

- $(\mathbb{K}, \oplus, \bar{0})$ *is a commutative monoid.*
- $(\mathbb{K}, \otimes, \bar{1})$ *is a monoid with* $\bar{1}$ *as the identity element for* $\otimes$.
- $\forall a, b, c \in \mathbb{K}$, $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$, *and* $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
- $\forall a \in \mathbb{K}$, $a \otimes \bar{0} = \bar{0} \otimes a = \bar{0}$.

While most applications use one-way weighted automata, we define and use a two-way version. The definition is similar, except that now the machine reads input off a tape and the head can travel in both directions. A two-way weighted automaton is defined as follows.

Definition 2 *A 2-way weighted finite automaton over the semiring* $\mathbb{K}$ *is a 7-tuple*
$W = (S, \Sigma, I, F, \Delta, \lambda, \rho)$ *where*

- S *is a finite set of states.*
- Σ *is a finite alphabet,* $\Gamma = \Sigma \bigcup \{ \text{¢}, \$ \}$ *being the tape alphabet.*
- $I, F \subseteq S$ *are sets of initial and final states respectively.*
- $\Delta : S \times \Gamma \times \mathbb{K} \times S \to D$ *is a finite set of transitions, where* $D = \{-1, 0, +1\}$ *is the direction of movement of the head on the tape.*
- $\lambda : I \to \mathbb{K}$ *and* $\rho : F \to \mathbb{K}$.

A weighted automaton recognizes languages in the following manner. For a transition $e \in \Delta$ we denote by $w[e]$ its weight and by $i[e]$ its input label. We denote a *path* $e_1 \ldots e_k \in \Delta^*$ by π. $first[\pi]$ is the originating state of the transition e_1 and $last[\pi]$ is the destination state of e_k. The label of the path, $label[\pi]$ is the string obtained by concatenating the labels of the transitions $e_1, \ldots, e_k$ in that order and the weight of the path $w[\pi] = w[e_1] \otimes \ldots \otimes w[e_k]$. If $P(q, q')$ is the set of all paths between states q and q', we say

$$P(I, F) = \bigcup_{q \in I, q' \in F} P(q, q').$$

If Δ is such that the movement of the head is always deterministic, as will be the case in this paper, then we can define:

$$\Pi(x') = \{ \pi \in P(I, F) | label[\pi] = x' \}.$$

where $x' = scan_W(x)$ is the sequence symbols of the input x to come under the head. It is important to note that since the head is two-way (or multi-scan), the same symbol of the input may occur more than once in $scan_W(x)$.

The input is associated to a weight in $\mathbb{K}$ by the automaton W as

$$W \circ x' = \bigoplus_{\pi \in \Pi(x)} \lambda(first[\pi]) \otimes w[\pi] \otimes \rho(last[\pi]) \tag{1}$$

In this paper, we use the semiring $(\mathbb{C}, +, *, 0 + i0, 1 + i0)$. This, however, is only for convenience. Any $n \times n$ complex matrix can be simulated by a $2n \times 2n$ real valued matrix, which is also unitary if the original matrix is. For example, the complex number $z = a + ib$ can be represented as $\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$.

Therefore, the semiring being used (while simulating unitary transformations) is essentially the real semiring, $(\mathbb{R}, +, *, 0, 1)$. Further, we also use the modulus operator $\mid z \mid^2 = a^2 + b^2$.

In case of the complex semiring we modify the association of weight to an input by the weighted automaton–

$$W \circ x' = \bigoplus_{\pi \in \Pi(x')} \mid \lambda(first[\pi]) \otimes w[\pi] \otimes \rho(last[\pi]) \mid .$$

With this in hand, we can now define language recognition by a weighted automaton.

Definition 3 *Let $\mathbb{J} \subseteq \mathbb{K}$. We say that a string $x \in \Sigma^*$ is $\mathbb{J}$-recognized by the weighted automaton W if $W \circ scan_W(x) \in \mathbb{J}$.*

Therefore, recognition involves two steps- computation of $W \circ x'$ and checking if $W \circ x' \in \mathbb{J}$. If $\mathbb{J}$ is a small set, then the second step is trivial. In this paper, $\mathbb{J}$ will be a singleton set. The first step is shown in [5] to take $|W|.|x'|.(T_\otimes + T_\oplus)$, where $|W|$ is the size of the automaton, and $T_\otimes$ and $T_\oplus$ are the cost of the $\oplus$ and $\otimes$ operations of the semiring. Example weighted automata are non-deterministic automata over the boolean semiring $\{\mathbb{B}, \vee, \wedge, 0, 1\}$, and probabilistic finite automata over the real semiring with $\mathbb{J} = \{y \mid \kappa < y \leq 1\}$ where κ is the "cut-point".

3 The Simulation

We begin by defining a useful term.

Definition 4 *A quantum state q observing which causes the classical part to reject the input is called a* rejecting *quantum state. We denote the set of such states by Q_{rej}.*

The Construction The weighted automata W consists of $|S|$ blocks, one for each state in S, the set of classical states in the 2QCFA being simulated. We first describe the construction of the blocks. Then we will show the blocks are linked together. The idea is to localize in one block all the unitary operations that occur while the ME-2QCFA is in the same classical state. The i^{th} block has states $q_0^{i_1}, q_1^{i_1}, \ldots, q_{k-1}^{i_1}, q_0^{i_2}, q_1^{i_2}, \ldots, q_{k-1}^{i_2}$, where $k = |Q|$.

Depending on σ, $\Theta(s_i, \sigma)$, $s_i \in S$, is one of the following unitary operators: $U(s_i, ¢)$, $U(s_i, a)$, $U(s_i, b)$ or $U(s_i, \$)$. These unitary operators can be written as

$$U(s_i, ¢) : | q_0\rangle \rightarrow \sum_{j=0}^{k-1} \phi_j^0 \mid q_j\rangle \quad \ldots \quad U(s_i, ¢) : | q_{k-1}\rangle \rightarrow \sum_{j=0}^{k-1} \phi_j^{k-1} \mid q_j\rangle$$

$$U(s_i, a) : | q_0\rangle \rightarrow \sum_{j=0}^{k-1} \alpha_j^0 \mid q_j\rangle \quad \ldots \quad U(s_i, a) : | q_{k-1}\rangle \rightarrow \sum_{j=0}^{k-1} \alpha_j^{k-1} \mid q_j\rangle$$

$$U(s_i, b) : | q_0\rangle \rightarrow \sum_{j=0}^{k-1} \beta_j^0 \mid q_j\rangle \quad \ldots \quad U(s_i, b) : | q_{k-1}\rangle \rightarrow \sum_{j=0}^{k-1} \beta_j^{k-1} \mid q_j\rangle$$

$$U(s_i, \$) : | q_0\rangle \rightarrow \sum_{j=0}^{k-1} \omega_j^0 \mid q_j\rangle \quad \ldots \quad U(s_i, \$) : | q_{k-1}\rangle \rightarrow \sum_{j=0}^{k-1} \omega_j^{k-1} \mid q_j\rangle.$$

We assign a weight α_j^l, β_j^l, ϕ_j^l or ω_j^l to the edge from $q_l^{i_1}$ to $q_j^{i_2}$ if it is labelled by a, b, ¢, or \$ respectively. Indeed, if the weights associated with all labels between two states $q_l^{i_1}$ and $q_j^{i_2}$ are 0, then there is no edge between them. A transition along any of these edges is not accompanied by a movement of the tape head.

Now we connect different blocks. Let $\delta(s_i, \sigma) = (s_p, d)$. Then, the weighted automaton has edges from $q_j^{i_2}$ to $q_j^{p_1}$, $0 \leq j \leq k-1$, labelled with σ and weight 1. The head moves by d.

The set of initial states I is the singleton set $\{q_0^{1_1}\}$. The set of final states F is the set $Q_{rej}^{i_2}$ for every block i, corresponding to the set of quantum rejecting states Q_{rej}. λ and ρ trivially assign a weight of 1 to all initial and final states respectively. Thus the input is associated with a weight as

$$W \circ x' = \bigoplus_{\pi \in \Pi(x')} \mid w[\pi] \mid . \tag{2}$$

This completes the construction.

By construction, it is evident that the head of the 2-way weighted automaton traverses the input symbols in exactly the same order as the ME-2QCFA. Hence $x' = scan_W x = scan_M x$ where $scan_M x$ is the scan sequence of the ME-2QCFA on input x.

Observation 1 *All paths of the computation of an ME-2QCFA end in the states of the same block i at any given step.*

We can therefore talk of the computation being *in a block i* at a given step.

Lemma 1 *Suppose $\mid \psi\rangle = \sum_{j=0}^{k-1} \gamma_j(x') \mid q_j\rangle$ is the state vector of the ME-2QCFA after reading the input $x' = scan_M(x)$. Then,*

$$\gamma_j(x') = \bigoplus_{\pi \in P_{x'}(q_0^{1_1}, q_j^{l'_2})} w[\pi]$$

where $P_{x'}(q_0^{1_1}, q_j^{l'_2})$ is the set of all paths labelled by x' from the $q_0^{1_1}$ state of the first block to the $q_j^{l'_2}$ state of l', the last block entered.

Proof 1 *We induct on the number of blocks transited during the computation.*

Basis: *Consider the first block immediately after the computation starts. The machine enters the first block into $q_0^{1_1}$. By the construction, it is evident that within a block i, the computation is exactly the application of the unitary matrices $U(s_i, \sigma)$, $\sigma \in \Gamma$. Just before the computation leaves block 1, we have*

$$\bigoplus_{\pi \in P_{x_0}(q_0^{1_1}, q_0^{1_2})} w[\pi] = \gamma_0(x_0) \qquad \textit{(over all paths from } q_0^{1_1} \textit{ to } q_0^{1_2}\textit{)}$$

$$\bigoplus_{\pi \in P_{x_0}(q_0^{1_1}, q_1^{1_2})} w[\pi] = \gamma_1(x_0) \qquad \textit{(over all paths from } q_0^{1_1} \textit{ to } q_1^{1_2}\textit{)}$$

$$\vdots$$

$$\bigoplus_{\pi \in P_{x_0}(q_0^{1_1}, q_{k-1}^{1_2})} w[\pi] = \gamma_{k-1}(x_0) \qquad \textit{(over all paths from } q_0^{1_1} \textit{ to } q_{k-1}^{1_2}\textit{)}$$

where x_0 is the sequence of symbols scanned while in block 1.

Inductive hypothesis: *The amplitude $\gamma_i(x_1)$ associated with $\mid q_i\rangle$ is $\bigoplus_{\pi \in P_{x_1}(q_0^{1_1}, q_i^{l_2})} w[\pi]$ where x_1 is the input sequence traversed thus far, of which x_0 is a prefix, the computation being in block l.*

Let $V_{P_{x_1}(q_0^{1_1}, q_i^{l_2})}$ be a vector the entries of which are the weights of paths in $P_{x_1}(q_0^{1_1}, q_i^{l_2})$. Then,

$$\gamma_i(x_1) = \bigoplus_{\pi \in P_{x_1}(q_0^{1_1}, q_i^{l_2})} w[\pi] = \sum_k V_{P_{x_1}(q_0^{1_1}, q_i^{l_2})}[k]$$

Induction step: *Suppose the computation exits block l and enters the next block (say l') on reading a symbol $\sigma \in \Gamma$. Let x_2 be the additional sequence of symbols traversed by the head while in block l'. Let $V_{P_{x_2}(q_i^{l'_1}, q_j^{l'_2})}$ be the vector of weights of paths in $P_{x_2}(q_i^{l'_1}, q_j^{l'_2})$.*

Define multiplication of two vectors V_1 and V_2 as $V_1 * V_2 = \sum_{i,j} V_1[i]V_2[j]$. Then,

$$\gamma_j(x') = \bigoplus_i (V_{P_{x_1}(q_0^{1_1}, q_i^{l_2})} * V_{P_{x_2}(q_i^{l'_1}, q_j^{l'_2})}).$$

Note that $\bigoplus_i (V_{P_{x_1}(q_0^{1_1}, q_i^{l_2})} * V_{P_{x_2}(q_i^{l'_1}, q_j^{l'_2})})$ is nothing but $\bigoplus_{\pi \in P_{x'}(q_0^{1_1}, q_j^{l'_2})} w[\pi]$. Therefore, $\gamma_j(x') = \bigoplus_{\pi \in P_{x'}(q_0^{1_1}, q_j^{l'_2})} w[\pi]$.

Lemma 2

$$A \circ x' = \bigoplus_{q \in Q_{rej}^{l'_2}} \mid \bigoplus_{\pi \in P_{x'}(q_0^{1_1}, q)} w(\pi) \mid^2 = 0$$

if and only if $x \in L$.

Proof 2 $\sum_{q \in Q_{rej}} \mid \gamma_q \mid^2 = 0$ *if and only if* $\mid \gamma_q \mid^2 = 0 \ \forall q \in Q_{rej}$. *If $p_{acc}(x) = 1$, then $p_{rej}(x) = 0$. Further, $p_{rej}(x) = 0$ implies $\mid \gamma_q \mid^2 = 0$, $\forall q \in Q_{rej}$. Thus,*

$$\sum_{q \in Q_{rej}} \mid \gamma_q \mid^2 = 0 \tag{3}$$

By the previous lemma, we know that if the last block is l', $\gamma_q = \bigoplus_{\pi \in P_{x'}(q_0^{1_1}, q^{l'_2})} w(\pi)$. Substituting in equation 3, we have $\sum_{q \in Q_{rej}^{l'_2}} \mid \bigoplus_{\pi \in P_{x'}(q_0^{1_1}, q)} w(\pi) \mid^2 = 0$. Identifying $\sum$ with $\bigoplus$, $A \circ x' = \bigoplus_{q \in Q_{rej}^{l'_2}} \mid \bigoplus_{\pi \in P_{x'}(q_0^{1_1}, q)} w(\pi) \mid^2 = 0$.

Theorem 1 *If a ME-2QCFA M recognizes a language L, then a two way weighted automata W constructed as above from the description of M can recognize L in time $O(\mid x \mid)$.*

Proof 3 *By the previous lemma, checking whether an input x is in L amounts to checking if $A \circ x = \bigoplus_{q \in Q^l_{rej}} \mid \bigoplus_{\pi \in P_x(q^1_0, q)} w(\pi) \mid^2$ equals 0. If it does, we accept the string, else we reject it. Hence, the constructed automaton 0-recognizes L. It is significant that the $\mathbb{J}$ is such a small subset of $\mathbb{K}$, as it reduces the membership test to constant time. The blow-up in the size of the machine is also only by a constant factor. So, the total time taken is $O(\mid x \mid)$, owing to a constant number of scans.*

We now give simulations of the machines of Ambainis and Watrous [1] as examples. Figures 1 and 2 show the weighted automata constructed from the 2QCFA's for L_{eq} and L_{pal} respectively.

Both 2QCFA algorithms are of the multi-scan type. To recognize L_{eq}, the machine starts with the quantum part (a 2-dimensional Hilbert space spanned by $|q_0\rangle$ and $|q_1\rangle$) initially in the state $|q_0\rangle$. One round of the algorithm consists of scanning the input from the beginning to the end once. That the input is indeed of the form a^*b^* is ensured by the classical part of the automaton. On reading an "a" the state vector of the quantum part is rotated by $+\sqrt{2}\pi$ and on reading a "b", by $-\sqrt{2}\pi$. Only one classical state is required for this. Hence, in the construction of the corresponding weighted automaton, we need only one non-trivial block, as shown in figure 1. The weights on the transitions are those that cause rotations in the 2QCFA.

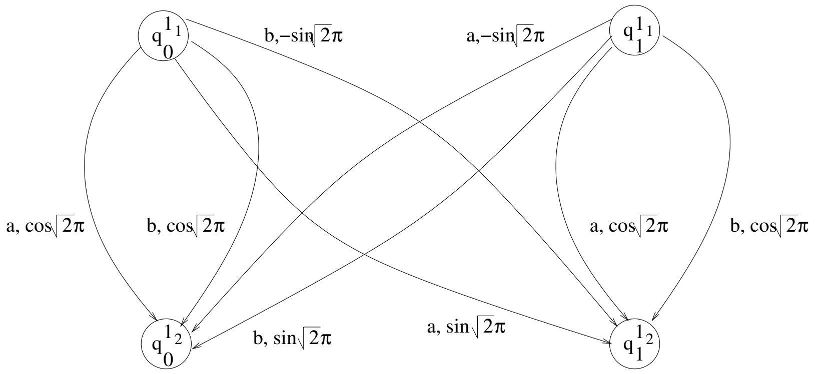

Figure 1: Recognizing L_{eq}. $q_0^{1_1}$ and $q_1^{1_2}$ are the start and final states respectively.

On the other hand, one round of the 2QCFA to recognize L_{pal} consists of two scans of the input. For convenience, a three dimensional quantum part with real amplitudes is used in [1]. Then, a map from the unit sphere in three-dimensional in Euclidean space to a two-dimensional complex unit sphere is

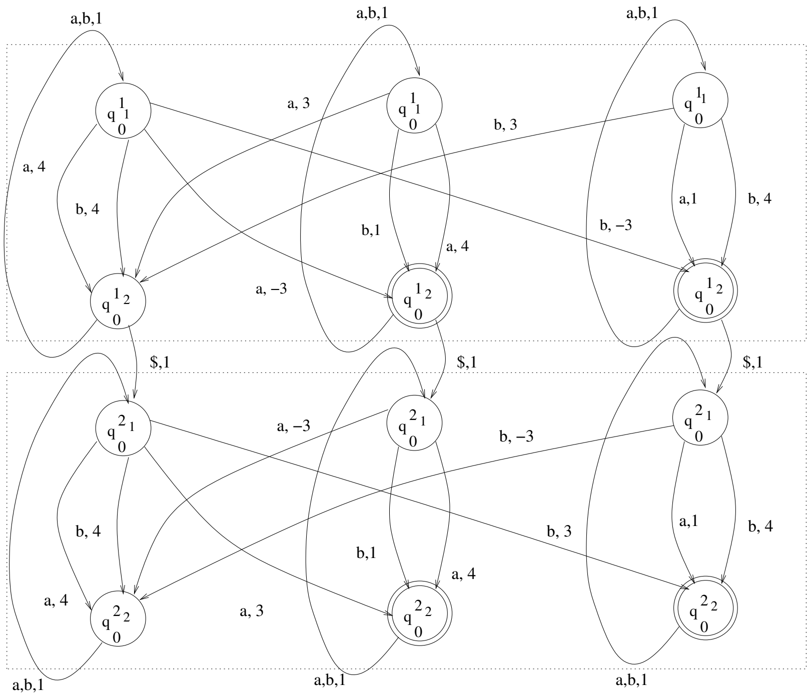

Figure 2: Recognizing L_{pal}. The boxes are shown enclosing individual blocks. q_0^{11} is the start state and the concentric circles denote final states in each block.

invoked, so that the computation can be done with one qubit. Initially, the quantum part is in the state $|q_0\rangle$. In the first scan, on reading an "a", the matrix A is applied and B on a "b". In the second scan, A^{-1} on reading an "a" and B^{-1} on reading a "b" where

$$\mathbf{A} = \frac{1}{5}\begin{pmatrix} 4 & 3 & 0 \\ -3 & 4 & 0 \\ 0 & 0 & 5 \end{pmatrix} \mathbf{B} = \frac{1}{5}\begin{pmatrix} 4 & 0 & 3 \\ 0 & 5 & 0 \\ -3 & 0 & 4 \end{pmatrix}$$

Two classical states are required for the two scans. Hence, the corresponding weighted automaton has two non-trivial blocks, one for each scan by the 2QCFA. The weights in one block (the upper one in figure 2) are taken from A and B and in the other, from A^{-1} and B^{-1}.

4 Conclusions

In this paper we showed a method of constructing a 2-way weighted automaton that efficiently accepts the same language as a given ME-2QCFA, a restriction version of 2QCFA of Ambainis and Watrous. An interesting problem is to construct weighted automata for unrestricted 2QCFA. It is possible that weighted automata of simple structure exist that can accept the same languages as 2QCFA. In that case, can we characterize languages accepted by 2QCFA in terms of the structural properties of the corresponding weighted automata?

References

[1] A. Ambainis, J. Watrous, Two-way finite automata with quantum and classical states, Theor. Comput. Sci., 287(1), pp. 299-311, 2002.

[2] A. Ambainis, A. Kikusts, Exact results for accepting probabilities of quantum automata, Theor. Comput. Sci., 295(1-3), pp. 3-25, 2003.

[3] A. Ambainis, A. Kikusts, M. Valdats, On the Class of Languages Recognizable by 1-way Quantum Finite Automata, Proc. of the 18th Ann. Symp. on Theor. Aspects of Comp. Sci. (STACS), pp. 75-86, 2001.

[4] A. Brodsky and N. Pippenger, Characterizations of 1-Way Quantum Finite Automata, SIAM Journal on Computing, 31(5), 1456-1478, 2002.

[5] C. Cortes, M. Mohri, Context-Free Recognition with Weighted Automata, Grammars 3(2/3), pp. 133-150, 2000.

[6] A. Kondacs, J. Watrous, On the Power of Quantum Finite State Automata, Proc. of the 38th Ann. Symp. on Found. of Comp. Sci. (FOCS), pp. 66-75, 1997.

[7] C. Moore, J.P. Crutchfield, Quantum Automata and Quantum Grammars, Theor. Comput. Sci., 237(1-2), pp. 275-306, 2000.

M. V. Panduranga Rao

Department of Computer Science and Automation

Indian Institute of Science

Bangalore 560 012

India

pandurang@csa.iisc.ernet.in

V. Vinay
Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560 012
India
vinay@csa.iisc.ernet.in