Proceedings of Grammar Systems Week 2004, edited by E. Csuhaj-Varjú and Gy. Vaszil,
MTA SZTAKI, Budapest, pages 15 - 29, 2004.

Distributed 2-Way Finite State Quantum Automata

A. Arun Prasath, Kamala Krithivasan

Dept of Computer Science and Engineering
Indian Institute of Technology, Madras
Chennai, India -600036
kamala@iitm.ernet.in

Abstract

We define Multiple choice two-way quantum automata with multiple observables. Distributed quantum automata are defined with four modes of cooperation. We show that multiple choice two way quantum automata and distributed quantum automata have the same power as that of two way quantum automata(having single choice) with multiple observables.

1 Introduction

The possibility that a quantum computational model can be more powerful than its classical counterpart, as has been elucidated by the celebrated Peter Shor's factoring and discrete logarithm algorithms [2] and Grover's quantum searching algorithm [3] has encouraged researchers to come up with various quantum computational models. Two-way quantum finite state automata (2qfa) have been proposed as the quantum analogue of deterministic (2dfa) finite state automata, and it has been shown that these 2qfa's are strictly more powerful than the 2dfa's, the 2nfa's and the 2pfa's [1].

Also, in the past, there have been attempts to come up with co-operative distributed models of classical automata. In the case of push down stack automata, it has been shown that, distribution results in increased power [4]. In fact, it has been shown that the distributed push down stack automata are equivalent in power to that of a Turing machine and hence are computationally complete. The main aim of this paper is to analyze the effect of distribution on the power of computation in quantum automata.

The 2qfa model proposed by Kondacs and Watrous has one minor restriction. It is that only measurements with respect to one particular decomposition (observable) of the state space is allowed (that which decides if the machine halts with acceptance, halts with rejection or goes on). In this work, the 2qfa model is first generalized to a model ((2qfa-MOb)) that allows measurements with respect to one or several of multiple decompositions (observables) of the state space. The significance of the new model is understood when distribution of computation in quantum automata is attempted. Different modes of cooperation are defined similar to that in distributed finite state automata. Then, it is shown that the distributed quantum finite state automata (D-2qfa-MOb) can be reduced to a 2qfa-MOb, for any of its

modes of acceptance.

First, a multiple choice 2qfa-MOb (MC-2qfa-MOb) is formulated, and it is proved that the behaviour of a multiple choice 2qfa-MOb can be simulated by a 2qfa-MOb to an arbitrarily close accuracy. Then, the D-2qfa-MOb is formulated, and by reducing this to a MC-2qfa-MOb, in each one of its modes of cooperation, it is proved that a D-2qfa-MOb in any mode of cooperation, has the same power as that of a 2qfa-MOb, thus proving the result.

2 2-Way Quantum Finite State Automata (2qfa)

2.1 Definition

Kondacs and Watrous [1] define a 2qfa as 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 alphabet (The tape alphabet Γ is defined as $\Sigma \cup \{\#, \$\}$, where $\#, \$ \notin \Sigma$ are used to mark the left and right ends of the tape respectively), δ is the transition function for the automaton defined below, $q_0 \in Q$ is the initial state and $Q_{acc} \in Q$ and $Q_{rej} \in Q$ are the sets of accepting states and rejecting states respectively (Elements of Q_{acc} and Q_{rej} are halting states and elements of $Q_{non} = Q - (Q_{acc} \cup Q_{rej})$ are non-halting states. Also, $q_0 \in Q_{non}$ and $Q_{acc} \cap Q_{rej} = \emptyset$).

The contents of any tape can be described by a mapping $x : \mathcal{Z}_n \to \Gamma$, n being the number of distinct tape squares on the tape. The number of configurations of the 2qfa M on any tape x of length n is $n|Q|$, since there are n possible locations for the tape head and $|Q|$ internal states. The configuration of a tape can hence be described by a mapping $\mathcal{C}_n = Q \text{ X } \mathcal{Z}_n$. A superposition of M on a tape x of length n is any norm 1 element of the finite dimensional Hilbert space $\mathcal{H}_n = l_2(\mathcal{C}_n)$, which is the space of mappings from $\mathcal{C}_n$ to $\mathbb{C}$ with the usual inner product. Following Dirac's notation, for each $c \in \mathcal{C}_n$, $|c\rangle$ 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 basic vectors. For a superposition $|\psi\rangle = \sum_{c \in \mathcal{C}_n} \alpha_c |c\rangle$, α_c is the probability amplitude associated with c in superposition $|\psi\rangle$.

The transition function δ of the 2qfa M is a mapping of the form,
$\delta : Q \text{ X } \Sigma \text{ X } Q \text{ X } \{-1, 0, 1\} \to \mathbb{C}$. For each $q, q' \in Q$, $\sigma \in \Sigma$ 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 its state to q' and move its tape head in direction d. For any tape x, δ induces a time evolution operator U_δ^x on $\mathcal{H}_{|x|}$ as follows. $U_\delta^x |q, k\rangle = \sum_{q',d} \delta(q, x(k), q', d)|q', k + d(mod|x|)\rangle$ for each $(q, k) \in C_{|x|}$, and is extended to all of $\mathcal{H}_x$ by linearity.

Since valid superpositions for the automata are of unit norm, the finite dimensionality of $\mathcal{H}_n$ requires U_δ^x to be a unitary operator, so that any valid superposition will evolve into another valid superposition and the automata M is then well formed. Now, let $V_\sigma : l_2(Q) \to l_2(Q)$ be an unitary operator in the Hilbert space $l_2(Q)$, and let $D : Q \to \{-1, 0, 1\}$. Now, if the transition function δ is defined as

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

M is well-formed iff V_σ is unitary.

2.2 Observables and Measurements

2.2.1 Definition

An observable $\mathcal{O}$ is a decomposition of the Hilbert space $\mathcal{H}_n$ into subspaces: $\mathcal{H}_n = E_1 \oplus \cdots \oplus E_k$, where the E_i are pairwise orthogonal. Let $|\psi_i\rangle$ be the projection of $|\psi\rangle$ onto E_i, for each i, so that, $|\psi\rangle = |\psi_1\rangle + \cdots + |\psi_k\rangle$. Then, the result of measuring observable O is that, the machine will collapse randomly to some outcome j with probability $|| \ |\psi_j\rangle \ ||$ and the new superposition of the machine will be $\frac{1}{|| \ |\psi_j\rangle \ ||}|\psi_j\rangle$.

2.2.2 Deciding the Time of Measurement

The decision as to when to measure an observable, is a crucial requirement, for the automaton to have the intended functionality. It is usually the case that, the machine is in a superposition of states, which means that, it essentially evaluates multiple paths simultaneously, and if the string on the tape has the desired structure, then the paths interfere either constructively or destructively. It is this aspect, that, in fact, distinguishes the qfa from classical automata, and is mainly responsible for their power. The point that is emphasized here, is that, the measurement is generally assumed to be done only after the interference mentioned above is guaranteed to have completed. Any measurement before that, would not guarantee proper functionality of the automaton. The instant of measurement, is also dependent on the string on the tape. In general, it is assumed that, there is some mechanism (that is not part of the unitary evolution of the automaton), which ensures that measurements are done only at appropriate moments. Strictly speaking, the definition of the automaton should include the specifications of the instants of measurement, along with the transition function, for it is the both that together define the functionality of the automaton under consideration.

2.3 The Possibility of Allowing Multiple Observables and Its Significance

The 2-qfa discussed above, allows measurements only with respect to one observable. This observable (that is referred to, here, as the primary observable) is a decomposition of the state space of the machine into subspaces constructed from non-halting states, halting states accepting the string on tape and halting states rejecting the string on tape. This measurement alone decides whether the machine is to further evolve or not, and in case of halting, it decides whether the string is accepted or not. Thus, it is this measurement that is used to define the languages accepted by a quantum automaton.

But, in general, it is possible to have different decompositions of the state space of the automaton (multiple observables). Such quantum automata possess sufficient power that, when distributed computation is attempted over these machines, additional power is not obtained, as shall be proved below. So, an interesting variant of the qfa is proposed below, wherein, multiple observables are allowed.

3 2qfa Allowing Measurements of Multiple Observables (2qfa-MOb)

3.1 Definition

A 2qfa-MOb is defined as a tuple

$$M = (Q, \Sigma, V, D, q_0, Q_{acc}, Q_{rej}, \mathcal{O})$$

where Q,Σ,q_0,Q_{acc} and Q_{rej} are as before and the tape alphabet Γ being defined as before from Σ by including left and right end markers $\{\#, \$\}$. V is a set of unitary operators $V_\sigma : l_2(Q) \to l_2(Q)$ for every $\sigma \in \Gamma$ and $D : Q \to \{-1, 0, 1\}$. Moreover, it is understood that the transition function (δ) for the automata is constructed from V and D as in Eqn.1. Using V and D, instead of δ, is just a notational simplicity that provides convenience when dealing with distributed quantum automata. $\mathcal{O}$ is a set of observables for the automaton, excluding the primary observable (defined as the decomposition into subspaces, $Q_{acc} \oplus Q_{rej} \oplus Q_{non}$, where $Q_{non} = Q - (Q_{acc} \cup Q_{rej})$). Measurements can be performed with respect to any of the observables in $\mathcal{O}$. The instant of measurement for any observable is assumed to be well-defined as in the case of the usual 2-qfa.

3.2 Languages Accepted by 2qfa-MObs

For a given string $w \in \Sigma^*$, a tape x_w is constructed with length $|w|+2$, with $x_w(0) = \#$, $x_w(|w|+1) = \$$ and $x_w(i) = w_i$ for $1 \leq i \leq |w|$. The computation begins in the superposition $|q_0, 0\rangle$ and measurements can be made with respect to the primary observable and observables in $\mathcal{O}$, the instants of these measurements being defined, as mentioned above. It is the measurement of the primary observable that decides the acceptance of w. When any measurement of the primary observable results in a halting state, the computation halts. If the state is in the subspace constructed from Q_{acc}, w is accepted. Otherwise, the state is in the subspace constructed from Q_{rej}, and the string is rejected.

The computation can now be treated in the same manner as for a probabilistic machine. For instance, 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. Just like probabilistic automata, restrictions such as running time and probability of error can be placed on the 2qfa-MOb as well.

4 Distributed 2qfa-MOb and Multiple-Choice 2qfa-MOb

4.1 Distributed 2qfa-MOb (D-2qfa-MOb)

4.1.1 Introduction

The D-2qfa-MOb is the quantum computational model corresponding to the classical distributed finite state automata. Different modes of acceptance can be defined here, in lines with the classical distributed automata.

4.1.2 Definition

A Distributed 2qfa-MOb (D-2qfa-MOb) is defined as a tuple,

$$M = (Q, \Sigma, V^1, V^2, \cdots, V^m, D, q_0, Q_{acc}, Q_{rej}, \mathcal{O})$$

where Q,Σ,q_0,Q_{acc} and Q_{rej} are as before, the tape alphabet Γ being defined as before from Σ by including left and right end markers $\{\#, \$\}$ and $\mathcal{O}$ as in 2qfa-MOb is the set of observables excluding the primary observable, with the timings of measurements for every observable, assumed to be well-defined. $V^i = \{V^i_\sigma, \sigma \in \Gamma\}, i \in \{1, 2, \cdots, m\}$, where each $V^i_\sigma : l_2(Q) \to l_2(Q)$. Also, $D : Q \to \{-1, 0, 1\}$ and it is understood that transition functions (δs) for the automaton are constructed from V^i's and D as in Eqn.1, with V^i replacing V in the equation. Each transition function obtained is well defined for any i, since all V^i_σ are unitary. The component of the machine decides which transition function can be chosen at any moment.

During any moment of evolution, the machine is in exactly one of the components $i \in \{1, 2, \cdots, m\}$. If the symbol read by the head is σ, then the next stage of evolution operates V^i_σ on the current internal state of the machine and the machine goes to a new internal state. Different modes of acceptance can be defined for these distributed automata. The difference between the modes arises from the definition of allowed transitions from one component to another, for the machine. Acceptance is defined exactly in the same way as in 2qfa-MOb with respect to the outcome of measurements of the primary observable.

4.1.3 Modes of Acceptance

There are four possible modes of acceptance:

- $*$ - mode - The transition from component i to some other component j can occur at any arbitrary stage of the evolution.
- $= k$ - mode - The transition from component i to some other component j occurs after exactly k transitions using the V^is.
- $\geq k$ - mode - The transition from component i to some other component j occurs after at least k transitions using the V^is.
- $\leq k$ - mode - The transition from component i to some other component j occurs before $k + 1$ transitions using the V^is.

The language accepted by a D-2qfa-MOb M using mode of acceptance $\alpha, \alpha \in \{*, = k, \geq k, \leq k | k \geq 1\}$ is denoted by $L(M, \alpha)$. Note that, in some cases the automata constructed is such that, all strings belonging to L are accepted definitely, whereas strings not in L are accepted with a finite probability bounded by a small value (Automata with one-sided bounded error). In such cases, the bound is also included in the notation as $L(M, \alpha, \text{bound})$.

4.2 Multiple-Choice 2qfa-MOb (MC-2qfa-MOb)

4.2.1 Introduction

The Multiple-Choice-2qfa-MOb (MC-2qfa-MOb) is the quantum computational model corresponding to the classical non-deterministic finite state automata(NFA). ie., There are choices in the evolution at any stage, and one of the choices can be chosen non-deterministically.

4.2.2 Definition

A MC-2qfa-MOb is defined as a tuple,

$$M = (Q, \Sigma, V, D, q_0, Q_{acc}, Q_{rej}, \mathcal{O})$$

where Q,Σ,q_0,Q_{acc} and Q_{rej} are as before, the tape alphabet Γ being defined as before from Σ by including left and right end markers $\{\#, \$\}$ and $\mathcal{O}$ as in 2qfa-MOb is the set of observables excluding the primary observable, with the timings of measurements for every observable, assumed to be well-defined. Now, $V = \{V_\sigma | \sigma \in \Gamma\}$. Each V_σ is now, a finite multiset of unitary operators, unlike the 2qfa-MOb wherein, V_σ is just a single operator. ie., $V_\sigma = \{V_{(\sigma,i)}, i \in \mathbb{N}\}$, and $V_{(\sigma,i)} : l_2(Q) \to l_2(Q)$.

As usual, $D : Q \to \{-1, 0, 1\}$ and it is understood that transition functions (δs) for the automata are constructed from V and D as in Eqn.1, with any $V_{(\sigma,i)}$ replacing V_σ in the equation. Any of the transition functions obtained can be used for the evolution of the automaton.

At any moment of evolution, if the symbol at the head is σ, then one of the operators in V_σ is chosen non-deterministically and is operated on the current internal state.

4.2.3 Choice of an Operator at an Instant

It should be noted that V_σ is a multiset since repetition of operators is allowed. The number of times a particular operator recurs in the set affects its probability of choice from the whole set. Suppose, a particular operator $\mathcal{V}$ occurs n times in V_σ, which is to say, $card(\{i | V_{(\sigma,i)} = \mathcal{V}\}) = n$. Then, the probability that $\mathcal{V}$ is chosen for evolution in the next stage is equal to $\frac{n}{card(V_\sigma)}$. (Here, $card(X)$ denotes the cardinality of the set or multiset X). Note that, for any MC-2qfa-MOb $M(Q, \Sigma, V, D, q_0, Q_{acc}, Q_{rej}, \mathcal{O})$, an equivalent MC-2qfa-MOb $M'(Q, \Sigma, V', D, q_0, Q_{acc}, Q_{rej}, \mathcal{O})$ with $card(V'_\sigma) = card(V'_\beta)$ for all $\sigma, \beta \in \Gamma$ (ie., all V'_σ having the same number of operators) can be constructed trivially, in the following way:

Let N be the L.C.M. (Least common multiple) of all the cardinalities, $card(V_\sigma)$. For each $\sigma \in \Gamma$, find $replication(\sigma) = \frac{N}{card(V_\sigma)}$ and construct V'_σ from V_σ by replicating each operator $\mathcal{V} \in V_\sigma$, $replication(\sigma)$ times. (If the operator occurs n times, it is replicated $n \times replication(\sigma)$ times). Now, $card(V'_\sigma) = N$, for any $\sigma \in \Gamma$. Also, this operation leaves the probability of selection of any operator in V_σ unchanged. This is because, for any $\mathcal{V}$ occurring n times in V_σ, the probability of selection of $\mathcal{V}$ from V'_σ is given by $\frac{n \times replication(\sigma)}{N} = \frac{n}{card(V_\sigma)}$. Hence, the probability remains the same. So, M' is equivalent to M and also has the same number of operators in each

$V'(\sigma)$.

From now on, it is assumed without loss of generality that, any MC-2qfa-MOb M has the same number of operators in each of its V_σs, and this number is referred to as the choice dimensionality (number of possible choices) of M. Also, it is understood that when simply written as 2qfa-MOb, it means that, there is no choice for the selection of operators. The language accepted by a MC-2qfa-MOb M is denoted by $L(M)$. In case of automata with one-sided bounded error, the bound is also included in the notation as $L(M, \text{bound})$.

5 Equivalence of MC-2qfa-MOb and 2qfa-MOb

5.1 Nature of Equivalence

In this section, the MC-2qfa-MOb is shown to be equivalent to the 2qfa-MOb. An exact equivalence is got in the case when the choice dimensionality (number of operators to choose from) is of the form 2^n, for some $n \in \mathbb{N}$. When it is not of such a form, then an equivalence can be obtained arbitrarily accurately (with bounded error).

5.2 Derivation of the Result

Theorem 5.2.1 *Any* MC-2qfa-MOb $M(Q, \Sigma, V, D, q_0, Q_{acc}, Q_{rej}, \mathcal{O})$ *can be reduced to an equivalent* 2qfa-MOb $M'(Q', \Sigma, V', D', q_0', Q_{acc}', Q_{rej}', \mathcal{O}')$ *with single choice for evolution operators.*

Proof: M' is constructed as follows:

Let N be the choice dimensionality of M. Two cases are considered.

5.2.1 Case-I: N is of the form 2^n, for some $n \in \mathbb{N}$ - Exact Equivalence

Construction:

Consider the set $C = \{1, 2, \cdots, N\}$. The Hilbert space constructed with elements in C ($l_2(C)$), is denoted by Ω. Now $Q' = Q \otimes C$. ie., Any valid state of M' is of the form $\sum_{q,i} a_{(q,i)}|q, i\rangle, q \in Q, i \in C, \sum a_{(q,i)}^2 = 1$. (Here, the $a_{(q,i)}$'s denote the amplitudes, and $|q, i\rangle$ stands for $|q\rangle \otimes |i\rangle$).

$D'(|q, i\rangle) = D(q)$ for all $q \in Q, i \in C$.

$Q_{acc}' = Q_{acc} \otimes C$.

$Q_{rej}' = Q_{rej} \otimes C$.

$q_0' = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} (|q_0, i\rangle)$.

$\mathcal{O}' = (\mathcal{O} \otimes C) \cup \omega$, where ω is the observable $(Q \otimes |0\rangle) \oplus (Q \otimes |1\rangle) \oplus \cdots \oplus (Q \otimes |N\rangle)$ which is to say that, all the original observables are preserved here by expanding the subspaces in the decompositions to the whole configuration space corresponding to Q', with the addition of a new observable, ω which measures the $|i\rangle$ portion of the superposition $\sum_{q,i} a_{(q,i)}|q, i\rangle$. The original timing information of the observables in $\mathcal{O}$ is still carried over to the corresponding observables in $\mathcal{O}'$. As far as measuring

ω is concerned, a measurement is made after every step in the evolution.

Now, let

$$E(1) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Construct $E(i)$s recursively from $E(i-1)$s using,

$$E(i) = \frac{1}{\sqrt{2}} \begin{pmatrix} (E(i-1)) & (E(i-1)) \\ (E(i-1)) & (-E(i-1)) \end{pmatrix}$$

The $E(i)$ are matrices of order 2^i. It is easily seen that, for any i, $E(i)$ is unitary (since $E(i) \times E(i)^T = I_{2^i}$, where I_{2^i} is the identity matrix of order 2^i). Also, $E(i)$ has the property that, each element in it is either $\frac{1}{\sqrt{2^i}}$ or $-\frac{1}{\sqrt{2^i}}$.

Now, the V' is constructed from V and $E(n)$ (Note that $N = 2^n$) as follows: If the internal state of the automaton is $|q, i\rangle$ and the symbol on the head is σ, then the evolution on applying V'_σ should be such that,

$$V'_\sigma(|q, i\rangle) = V_{(\sigma,i)}(|q\rangle) \otimes E(n)(|i\rangle)$$

The construction is as below when the states and operators are expressed explicitly in matrix form. Let $Q = \{q_1, q_2, \cdots, q_s\}$, $s = card(Q)$. The amplitudes of the the current superimposition of states of the machine can be expressed as a column vector of the amplitudes $a_{(q,i)}$s as

$$(a_{(q_1,1)}, a_{(q_2,1)}, \cdots, a_{(q_s,1)}, a_{(q_1,2)}, a_{(q_2,2)}, \cdots, a_{(q_s,N)})^T.$$

Now, if

$$E(n) = \begin{pmatrix} a_{11} & a_{12} & . & . & . & a_{1N} \\ a_{21} & a_{22} & . & . & . & a_{2N} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a_{N1} & a_{N2} & . & . & . & a_{NN} \end{pmatrix}$$

then

$$V'_\sigma = \begin{pmatrix} (V_{(\sigma,1)}) \times a_{11} & (V_{(\sigma,2)}) \times a_{12} & . & . & . & (V_{(\sigma,N)}) \times a_{1N} \\ (V_{(\sigma,1)}) \times a_{21} & (V_{(\sigma,2)}) \times a_{22} & . & . & . & (V_{(\sigma,N)}) \times a_{2N} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ (V_{(\sigma,1)}) \times a_{N1} & (V_{(\sigma,2)}) \times a_{N2} & . & . & . & (V_{(\sigma,N)}) \times a_{NN} \end{pmatrix}$$

The construction of $V'(\sigma)$ is very similar to that of a tensor product and it can be easily verified that V'_σ is unitary using the fact that, all V_σs and $E(n)$ are unitary.

Proof of equivalence:

The proof rests on the fact that, the $|i\rangle$ component of the internal state $|q, i\rangle$ for M' simulates the operation of $V_{(\sigma,i)}$ on the current state $|q\rangle$ on M, when the symbol

read by the head is σ. Consider any intermediate step in the evolution of M. Let the symbol read on one of the paths at instant $t-1$ be σ and the state be q. Now, one of the matrices in V_σ is chosen non-deterministically. Without loss of generality, we assume that, $V_{(\sigma,j)}$ was chosen. Now, the new state of the machine becomes $V_{(\sigma,j)}(|q\rangle) = |v\rangle$ (say), the head moves in the direction $D(|v\rangle)$ and all observables that can be measured at that instant, as per the timing specification in O can be measured. If the primary observable is measured, then the halting/non-halting and acceptance/rejection decisions are made. The time now becomes t. If not halted, the evolution proceeds in the same manner for time instant t.

The same situation translates into the construction and functionality of M' as follows. At time $t-1$, the symbol read on the path is σ and the state of the automaton is

$$r(t-1) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} (|q,i\rangle).$$

The form above corresponds to the state $|q\rangle$ of M, as will be evident, once the evolution is understood. Now, a measurement of ω is done. This collapses state $r(t-1)$ to the state $|q,j\rangle$ with the same probability $\frac{1}{N}$ as in the case of M. Now, V'_σ is applied on the state $|q,j\rangle$ to give

$$V'_\sigma(|q,j\rangle) = V_{(\sigma,j)}(|q\rangle) \otimes E(n)(|j\rangle) = |v\rangle \otimes \sum_{i=1}^{N} (|i\rangle)$$

which takes the machine to the state

$$r(t) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} (|v,i\rangle)$$

which corresponds to the state $|v\rangle$ of M in the same way as before. This consistent correspondence of the states of M' to those of M is because, $E(n)$ is such that, for any $k \in C$ and correspondingly $|k\rangle \in \Omega$, $E(n)(|k\rangle) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} (|i\rangle)$. The head moves in the direction $D'(|v,i\rangle) = D(|v\rangle)$ for any i. The measurements mentioned above are made with respect to corresponding observables in M' and as seen, the primary observable of M' is defined consistently in terms of Q'_{acc} and Q'_{rej}. Also, the initial state q'_0 is such that a measurement of ω initially would set the state to $|q_0,j\rangle$ for some j. Thus, M' exactly simulates the behaviour of M, and hence the languages accepted by them are exactly equivalent (the exactness here, also, includes acceptance with one-sided bounded error, since the probabilities of the choice of operators have been taken care of, accurately).

5.2.2 Case-II: N is NOT of the form 2^n, for any $n \in \mathbb{N}$ - Equivalence with bounded error

Construction:

The equivalence in the previous case was proved when N is of the form 2^n. The matrices $E(i)$ defined above have order 2^i. Matrices with similar properties (unitarity and equal distribution of amplitudes), but orders not of the form 2^i cannot be constructed. Hence, when the choice dimensionality is not of the form 2^n, equivalence cannot be proved in exactly the same way before. Here, equivalence

is proved allowing small violations in the probability of choice of operators. Such violations in anyway do not affect the class of languages, accepted by automata with certainty. Only the probability of acceptance or rejection, in the case of automata with bounded error, is affected. This deviation can be made arbitrarily small, as shall be shown below. In this way, M' can simulate M as accurately as possible.

The construction is done by choosing a large number $N' = 2^m$, for some $m \in \mathbb{N}$, such that $N' > N$. Let f be the quotient and g the remainder when N' is divided by N. For each $\sigma \in \Gamma$, replicate $V_{(\sigma,i)}$, for every i, f times. Then, reinclude the first g operators, $\{V_{(\sigma,i)} | 1 \leq i \leq g\}$. Let the set of operators now be denoted $V"_\sigma$. Consider the automata $M" = (Q, \Sigma, V", D, q_o, Q_{acc}, Q_{rej}, \mathcal{O})$. The number of operators in each of $V"_\sigma$ is now 2^m. Then, the construction is done exactly as in Case-I and the automata M' is got. Now, it is clear that, M' accepts the same class of languages accepted by M, and only in the case of languages accepted with finite one-side bounded error, the probabilities of acceptance or rejection might slightly differ.

The probability of choosing the i^{th} operator, $V_{(\sigma,i)}$ from V_σ at any moment is $\frac{1}{N}$, whereas the probability of choosing the same operator from $V"$ is equal to either $\frac{f}{N'}$ or $\frac{f+1}{N'}$. As shown below, the difference in the two probabilities can be made arbitrarily small, by appropriately choosing N', and hence, $M"$ (and hence M') can be made to simulate M arbitrarily accurately.

Choice of N':

Suppose the bound that we impose on the difference in the probabilities is ε. ie.,

$$\tfrac{f+1}{N'} - \tfrac{1}{N} < \varepsilon.$$

To achieve this bound, N' is chosen as follows. Find numbers m and λ, with $\lambda \leq \varepsilon$ such that,

$$\tfrac{1}{2^m} > \lambda > (\tfrac{N-1}{N})\tfrac{1}{2^m}$$

m and λ can be found for any given ϵ, since it is always possible to choose as high an m as possible (corresponding to lower and lower λ values). Now, $N' = 2^m$. This choice of m and λ gives

$$\tfrac{1}{\lambda} > 2^m (= N') > (\tfrac{N-1}{N})\tfrac{1}{\lambda} \Rightarrow N - NN'\lambda < 1.$$

Since the remainder g is non-zero (at least one), we have

$$1 \leq N' - Nf < N$$

$$\Rightarrow N' - Nf > N - NN'\lambda$$

$$\Rightarrow N(f+1) < N'(1 + N\lambda)$$

which reduces to,

$$\tfrac{f+1}{N'} - \tfrac{1}{N} < \lambda \leq \varepsilon.$$

In an exactly similar way, the choice of N' for the bound,

$$\tfrac{1}{N} - \tfrac{f}{N'} < \varepsilon$$

can be done. In order for both the bounds to hold good, we choose the bound corresponding to the lesser of the two differences, and choose N' based on that.

The bound on the probability difference translates directly to the bound on the difference in the probabilities of one-sided rejection. Hence, if the choice dimensionality is not of the form 2^n, an equivalence with bounded error can be obtained.

5.3 Result

Thus, the MC-2qfa-MOb are reducible to the 2qfa-MOb, exactly when the choice dimensionality is of the form 2^n and arbitrarily closely otherwise. ie., for every MC-2qfa-MOb M, there exists a 2qfa-MOb M' such that, $L(M) = L(M')$. If M has acceptance with bounded error (bound ε), then a 2qfa-MOb M" can be constructed such that, $L(M, \varepsilon) = L(M", \vartheta)$, such that, $|\varepsilon - \vartheta| < \varsigma$, for any arbitrarily small ς.

6 Equivalence of D-2qfa-MOb and MC-2qfa-MOb

6.1 Theorem:

For any D-2qfa-MOb M, there exists a MC-2qfa-MOb M', such that, $L(M, \alpha) = L(M')$, for any $\alpha \in \{*, = k, \leq k, \geq k\}$. In the case of M being a D-2qfa-MOb with one-sided bounded error ε for acceptance, then, a MC-2qfa-MOb M' can be constructed such that, $L(M, \alpha, \varepsilon) = L(M', \varepsilon)$

6.1.1 Proof:

The result is proved by reducing a D-2qfa-MOb M to a MC-2qfa-MOb M', for each mode of transition, as below:
Case-I: $\alpha = *$

Let $M = (Q, \Sigma, V^1, V^2, \cdots, V^m, D, q_0, Q_{acc}, Q_{rej}, \mathcal{O})$. The equivalent MC-2qfa-MOb is constructed as follows. $M' = (Q', \Sigma, V', D', q_0', Q_{acc}', Q_{rej}', \mathcal{O}')$.
Let $C = \{1, 2, \cdots, m\}$. Let $\mathcal{H} = l_2(C)$ denote the Hilbert space constructed using the elements of C as basis. Also, let $s = card(Q)$.

$Q' = Q \otimes \mathcal{H}$. ie., Q' can be represented as $\{[q, i] | 1 \leq i \leq m, q \in Q\}$. The amplitudes of the current superimposition of states of the machine can be expressed as a column vector of amplitudes,
$(a^1_{q_1}, a^1_{q_2}, \cdots, a^1_{q_s}, a^2_{q_1}, \cdots, a^m_{q_s})^T$, where $Q = \{q_i | 1 \leq i \leq s\}$ and $a^i_{q_j}$ is the amplitude corresponding to the internal state $q_j \in Q$, when the machine is in component i. Since the machine can be in only one of the components at any instant, the amplitudes corresponding to all other components are zeroes.

$D'([q, i]) = D(q)$, for all $q \in Q$ and $i \in C$.

$q_0' = [q_0, 1]$.

$Q_{acc}' = \{[q, i] | q \in Q_{acc}, i \in C\}$.

$Q_{rej}' = \{[q, i] | q \in Q_{rej}, i \in C\}$.

$\mathcal{O}'$ is obtained as follows. For an observable (decomposition) $O \in \mathcal{O}$ with $O = S_1 \oplus S_2 \oplus \cdots \oplus S_l$, for some l, then an observable $O' = (S_1 \otimes \mathcal{H}) \oplus (S_2 \otimes \mathcal{H}) \oplus \cdots \oplus (S_l \otimes \mathcal{H})$ is included in O', with the same timing specifications as that of O.

To construct V', the various possible transitions among the components of V are represented as permutations over the set C . A permutation over C is a one-one mapping $P : C \to C$. It can be conveniently represented as a m-tuple $(P(1), P(2), \cdots, P(m))$. For any given mode of transition in M, there are many allowed permutations. Let P_{al} denote the set of all allowed permutations over C for the mode of transition defined in M. As shown below, each allowed permutation corresponds to a choice in the operators in V'. When the machine is in component i, the head reads symbol σ and the permutation (m-tuple) $\mathcal{P} = (P(1), P(2), \cdots, P(m))$ is chosen, the following is done.

- The operator V_σ^i is used to transform the current internal state of the machine. This is achieved through the function matrices described below.
- The machine then transits to the component $P(i)$ in the next step. This is achieved through the transition matrices described below.

Let $\mathcal{P} = (P(1), P(2), \cdots, P(m))$ be an allowed permutation over C, for the mode of transition in M. $\mathcal{P}$ induces a transition matrix $T(\mathcal{P})$ which specifies the transformation of the state in M' corresponding to the change of component in M. Also, for each $\sigma \in \Gamma$, there is a function matrix $F(\sigma)$ which specifies the evolution of the state in M', when the symbol read is σ. $F(\sigma)$ is essentially derived from the Vs of the various components in M..

To understand the construction of these matrices, it is helpful to think of them as $m \times m$ grids of cells, with matrices of order $s \times s$ in each cell. Let $I(s)$ and $O(s)$ denote identity and zero matrices of order $s \times s$ respectively. The grid cell at the k^{th} row and l^{th} column is denoted by $C(k, l)$.

Now, $F(\sigma)$ is defined as the matrix of cells $C_F(k, l), 1 \leq k, l \leq$ given below.

$$C_F(k,l) = \begin{cases} O(s) & k \neq l \\ V_\sigma^k & k = l \end{cases}$$

$T(\mathcal{P})$ is defined as the matrix of cells $C_T(k, l), 1 \leq k, l \leq$ given below.

$$C_T(k,l) = \begin{cases} I(s) & P(l) = k \\ O(s) & \text{otherwise} \end{cases}$$

For instance, the construction of the matrices are shown for the case $m = 4$, $s = 2$ and $\mathcal{P} = (4, 1, 3, 2)$ (Take component 1 to component 4, component 2 to component 1, etc...) is shown below.

$$F(\sigma) = \begin{pmatrix} (V_\sigma^1) & (O(2)) & (O(2)) & (O(2)) \\ (O(2)) & (V_\sigma^2) & (O(2)) & (O(2)) \\ (O(2)) & (O(2)) & (V_\sigma^3) & (O(2)) \\ (O(2)) & (O(2)) & (O(2)) & (V_\sigma^4) \end{pmatrix}$$

$$T(\mathcal{P}) = \begin{pmatrix} (O(2)) & (I(2)) & (O(2)) & (O(2)) \\ (O(2)) & (O(2)) & (O(2)) & (I(2)) \\ (O(2)) & (O(2)) & (I(2)) & (O(2)) \\ (I(2)) & (O(2)) & (O(2)) & (O(2)) \end{pmatrix}$$

ie.,

$$T(\mathcal{P}) = \begin{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \end{pmatrix}$$

$$T(\mathcal{P}) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

since

$$I(2) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad and \quad O(2) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Now suppose the machine M was in component 2 in state $(a_1, a_2)^T$, where a_1, a_2 are the probability amplitudes corresponding to states q_1, q_2, the only states in Q and the current symbol read is σ. Then the state of the machine M' is the column vector $\alpha = (0, 0, a_1, a_2, 0, 0, 0, 0)^T$. Now, the choice of the permutation $\mathcal{P} = (4, 1, 3, 2)$ would mean that, the component should change to 1 after the application of V_σ^2. This is essentially accomplished by the operation $T(\mathcal{P}) \times (F(\sigma) \times \alpha)$. Suppose $V_\sigma^2 \times (a_1, a_2)^T = (b_1, b_2)$, then the above operation $(T \times F \times \alpha)$ would result in the state $\beta = (b_1, b_2, 0, 0, 0, 0, 0, 0)^T$ in consistence with the description of the correspondence of states in M' to states in M.

Since V_σ^i is unitary for all i, $F(\sigma)$ is also unitary $(F(\sigma)F^T(\sigma) = I)$. Also, it is evident that, the transition matrices are unitary too (since there is exactly one 1 in each row (column) with the rest being zeroes, the rows (columns) are linearly independent). Now,

$$V'_\sigma = \{T(\mathcal{P}) \times F(\sigma) \mid \mathcal{P} \in P_{al}\}$$

Thus, M' simulates the function of M, provided the allowed permutations in M' clearly reflect the mode of transition of M. In the "*-mode" of transition, since a transition from the current component i to some component j can occur at any instant, P_{al} includes all permutations over C. ie., all permutations are valid, since any permutation specifies a valid transition between components for M. Hence, when V' is constructed as above, using the permutations in P_{al}, M' simulates M exactly. Since the event of choosing any of the matrices from V'_σ is equally likely, all

permutations are equally likely, which means that, the probability of switching from the current component i to a component j is the same as that of switching to any other component k (There are equal number of permutations that take component i to component j, as there are for transitions from i to k). Also, it is easy to see that, acceptance in M' is defined in exact correspondence to observables in M, and the initial state is defined consistently.

Thus, M' is exactly equivalent in behaviour to M, and hence
$L(M, " * -mode") = L(M')$.
Case-II: $\alpha = " = k"$

The proof for the "=k mode" is very similar to that of the "*-mode". The derivation is looked in the following way. The system M, is looked upon as an equivalent system
$M" = (Q, \Sigma, V^{11}, V^{12}, \cdots, V^{1k}, V^{21}, \cdots, V^{mk}, D, q_0, Q_{acc}, Q_{rej}, \mathcal{O})$, where for all i, V^{ik} of $M"$ is a component created by an exact replication of V^i of M, and now, the transitions are to be forced between components 11 and 12, etc... whereas, there is a choice in transition between ik and $j1$, for any $i, j \in C$. This is accomplished by constructing the MC-2qfa-MOb M' in the same way as before, except for a change in P_{al}. All permutations are not allowed, since as said above, some transitions are to be forced. Hence, only permutations that translate ij to $i(j+1)$, if $j < k$ and to $m1$ otherwise, are allowed. The transition matrices corresponding to these permutations are constructed and V'_σ are constructed as before, by premultiplying the function matrix with these matrices. Then, M' simulates M exactly, and hence, $L(M, " = k - mode") = L(M')$.
Cases-III and IV: $\alpha = " \leq k" | " \geq k"$

The proof for these cases is exactly similar to that of the previous cases. The key to the proof lies in the fact that, the mode of transition in M can be exactly simulated in M' by appropriately defining P_{al} by identifying the permutations that specify valid transitions of components for M.

6.2 Result

Hence, any D-2qfa-MOb is exactly reducible to a MC-2qfa-MOb, for any of its modes of acceptance. Combining this result with the equivalence of MC-2qfa-MOb and 2qfa-MOb, the following result is also true. For any D-2qfa-MOb M, there exists a 2qfa-MOb M', such that, $L(M, \alpha) = L(M')$, for any $\alpha \in \{*, = k, \leq k, \geq k\}$. In the case of M being a D-2qfa-MOb with one-sided bounded error ε for acceptance, then, a 2qfa-MOb M' can be constructed with, $L(M, \alpha, \varepsilon) = L(M', \vartheta)$ such that, $|\varepsilon - \vartheta| < \varsigma$, for any arbitrarily small ς.

7 D-1qfa-MOb

In a way exactly similar to 2-qfa, distributed and multiple choice 1-qfa can also be defined. The corresponding results also hold good here, once multiple observables are allowed. Thus, both D-1qfa-MOb and MC-1qfa-MOb are equivalent in power to the 1qfa-MOb with single choice, and acceptance with certainty, whereas, in the case

of automata with bounded error, D-1qfa-MOb and MC-1qfa-MOb can simulate the behaviour of 1qfa-MOb arbitrarily closely.

8 Conclusion

Thus, in this work, first, quantum automata were generalized to include multiple observables. Then, distribution was attempted. Multiple Choice quantum automata and Distributed quantum automata were considered, in lines with classical NFA and distributed finite state automata, and it is found that, the power of these automata are the same as that of the simple quantum automata with multiple observables and single choice. But the question as to whether allowing multiple observables results in increase in power (whether the 2qfa-MOb are more powerful than 2qfa) has not been answered in this work.

References

[1] A. Kondacs and J. Watrous. On the power of quantum finite state automata. *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 66-75, 1997.

[2] P. Shor. Algorithm for quantum computation: discrete logarithm and factoring. *35th Annual Symposium on Foundations of Computer Science*, pages 124-134, 1994.

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

[4] K. Krithivasan, M. Sakthi Balan and Prahlad Harsha. Distributed processing in automata. *International Journal of Foundations of Computer Science*, Vol 10, No.4, pages 443-464, 1999.