1-Way Multihead Quantum Finite State Automata

Debayan Ganguly, Kingshuk Chatterjee, Kumar Sankar Ray

Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India
Email: debayan3737@gmail.com, kingshukchaterjee@gmail.com, ksray@isical.ac.in

Abstract

1-way multihead quantum finite state automata (1QFA(k)) can be thought of modified version of 1-way quantum finite state automata (1QFA) and k-letter quantum finite state automata (k-letter QFA) respectively. It has been shown by Moore and Crutchfield as well as Konadacs and Watrous that 1QFA can't accept all regular language. In this paper, we show different language recognizing capabilities of our model 1-way multihead QFAs. New results presented in this paper are the following ones: 1) We show that newly introduced 1-way 2-head quantum finite state automaton (1QFA(2)) structure can accept all unary regular languages. 2) A language which can't be accepted by 1-way deterministic 2-head finite state automaton (1DFA((2)) can be accepted by 1QFA(2) with bounded error. 3) 1QFA(2) is more powerful than 1-way reversible 2-head finite state automaton (1RMFA(2)) with respect to recognition of language.

1. Introduction

Classical finite state automaton is the very basic model of classical finite machine. Likewise a quantum finite state automaton may be seen as basic model of finite state quantum machine. A variety of models of quantum finite state automaton are used. 1-way quantum finite automaton (1QFA) can be seen as the simplest model of quantum automaton .The two most popular models of quantum finite state automaton are quantum finite state automaton introduced by Moore and Crutchfield [1] (measure once quantum finite state automaton) and quan-

tum finite state automaton introduced by Kondacs and Watrous [2] (measure many quantum finite state automaton). They have seemingly small difference, measure once quantum finite state automaton performs the measurement only at the end of computation,but for measure-many quantum finite state automaton the measurement will be performed by the automaton at every step of computation. Ambainis *et al.* [3] showed that measure many one-way quantum finite automata can accept all languages that can be accepted by measure once one-way quantum finite automata. Hence, in this paper,we consider the measure many quantum finite automata described by Kondacs *et al.* [2]. Whenever we mention one-way quantum finite automata we mean the model described by Konadacs *et al.* It has been shown by Kondacs *et al.* that the languages recognized by 1QFA's form a proper subset of the regular languages. Besides these two models of QFA there are also such models of QFA as "enhanced" quantum finite state automaton [4], latvian quantum finite state automaton [5], 1-way QFA with control languages [6], quantum finite state automaton with quantum and classical states (introduced by aharonov, kitaev and Nisan [7]). Some other QFA models can be found in [8] [9]. In [10], A. Nayek proposed a further generalization by allowing the QFA to perform several arbitrary measurements with intermediate unitary transformation at each step. The second model is 2-way quantum finite state automaton (2QFA) [2]. In this model,it is easy to simulate any deterministic automaton and some non-regular languages can be recognized as well; this implies that 2QFA's are strictly more powerful than their classical counterparts. In [7], they propose 2-way finite automaton with quantum and classical states, an intermediate model between 1QFA's and 2QFA's.

Languages accepted by multitape or multihead finite automaton were introduced in [11] and [12]. 1-way reversible and multihead finite automaton [13] and 2-way reversible multihead finite automaton [14] are introduced as a simple model of reversible computing and its language accepting capability is studied.

A konadacs and J Watrous [2] showed that 1QFA can only recognize regular languages,moreover, 1QFA cannot recognize all the regular languages. In [15], they proposed a new model of one way QFA, namely, multiletter QFAs, that is an analogue of quantum automaton with classical memory containing the previously read letters. In these model,the automaton is not limited to seeing only one,the just incoming letter, but can see several earlier received letters as well. So a *k*-letter QFA is not limited to see only one,the just incoming input letter. Daowen Qiu *et al.* [16] further study the decidability of the equivalence and minimization problems of multiletter QFAs. In [17], hierarchy and equivalence of multiletter quantum finite state automaton are studied.

Belovs *et al.* [15] have already showed that regular language $(a,b)^* a$ which can't be accepted by 1QFA can be accepted by a 2-letter QFA.We continue the investigation of of 1-way quantum finite state automaton and k-letter quantum finite state automaton for improving their language accepting capabilities. In this paper, we introduce 1-way multihead quantum finite state automaton (1QFA(k)) by introducing multiple heads combined with existing automaton and study its language recognizing capabilities. It is proved that the newly introduced model 1QFA(2) can accept all unary regular languages.

We know that the language $\left\{ L = w \in \{a,b\}^+ \mid w \neq w^R \right\}$ cannot be recognized by 1DFA(2) ([18] [19]) and 1RMFA(2) ([13]) respectively.Here we show that this language *L* can be recognized by our model 1QFA(k). It has been shown that 1QFA(2) is more powerful compare to 1RMFA(2) respectively.We consider the context-sensitive language: $\left\{ a^n b^n c^n \mid n \geq 1 \right\}$. It has been shown that this languages is also recognized by 1QFA(2).

2. Preliminaries and Definitions

In this section we give different definitions and corresponding results for 1QFA.

2.1. Quantum Finite Automata

One-way quantum finite state automaton can been seen as the simplest model of quantum computation.Quantum finite automata could be of large importance is the fact that quantum memory seems to be very expensive and it is therefore of very much importance to know what can be achieved with limited amounts of quantum resources.

2.1.1. 1-Way Quantum Finite State Automata

One-way quantum finite state automaton seem to model very well the way very simple quantum processors work (Ambainis and freivalds, 1998), and also the way simple classical/quantum processors are expected to work: the classical part reads an input, picks up the corresponding quantum operator (a transition mapping) and performs it on a quantum memory of fixed size, independent of the size of input. 1QFA are very simple but less

powerful than classical 1-way finite automaton.

Measure many quantum finite state automata (1QFA): We consider 1-way quantum finite automata (QFA) as defined in [2].

Definition 1. *Namely, a* 1-*way QFA is a tuple* $M = (Q, \Sigma, \delta, q_0, Q_a, Q_r)$ *where*

1) Q *is a finite set of states,*

2) Σ *is an input alphabet,*

3) δ *is a transition function,*

4) $q_0 \in Q$ *is a starting state,*

5) $Q_a \in Q$ *and* $Q_r \in Q$ *are sets of accepting and rejecting states.*

The states in Q_a *and* Q_r *are called halting states and the states in* $Q_{non} = Q - (Q_a \cup Q_r)$ *are called non-halting states.* # *and* $ *are symbols that do not belong to* Σ. *We use* # *and* $ *as the left and the right end marker, respectively. The working alphabet of M is* $\Gamma = \Sigma \cup \{\#, \$\}$.

A superposition of M is any element of $l_2(Q)$ (the space of mappings from Q to $\mathbb{C}$ with l_2 norm). For $q \in Q$, $|q\rangle$ denotes the unit vector with value 1 at q and 0 elsewhere. All elements of $l_2(Q)$ can be expressed as linear combinations of vectors $|q\rangle$. We will use ψ to denote elements of $l_2(Q)$. The transition function δ maps $Q \times \Gamma \times Q$ to C. The value $\delta(q_1, a, q_2)$ is the amplitude of $|q_2\rangle$ in the superposition of states to which M goes from $|q_1\rangle$ after reading a. For $a \in \Gamma$, V_a is a linear transformation on $l_2(Q)$ defined by

$$V_a(|q_1\rangle) = \sum_{q_2} \delta(q_1, a, q_2)|q_2\rangle \tag{1}$$

The computation of a QFA starts in the superposition $|q_0\rangle$. Then transformations corresponding to the left endmarker #, the letters of the input word x and the right endmarker $ are applied. The transformation corresponding to a $a \in \Gamma$ consists of two steps. 1) First, V_a is applied. The new superposition ψ' is $V_a(\psi)$ where ψ is the superposition before this step. 2) Then ψ' is observed with respect to the observable $E_{acc} \oplus E_{rej} \oplus E_{non}$ where $E_{acc} = \text{span}|q_2\rangle : q \in Q_a$, $E_{rej} = span|q_2\rangle : q \in Q_r$, $E_{non} = span|q_2\rangle : q \in Q_{non}$. This observation gives $x \in E_i$, with the probability equal to the amplitude of the projection of ψ'. After that, the superposition collapses to this peojection. If we get $\psi' \in E_{acc}$, the input is accepted. If we get $\psi' \in E_{rej}$, the input is rejected. If we get $\psi' \in E_{non}$, the next transformation is applied.

Theorem 1. *Let L be any language recognized by* 1*QFA with bounded error. Then L is regular.*

Proof. The proof is in [2].

Proposition 1. *Given a language* $S = \{w_1, w_2, \cdots, w_k\}^*$, *it is not possible in general case to build a* 1*QFA that recognizes this language.*

Proof. The proof is in [20].

Theorem 2. *The language* $L = \{a, b\}^*$ *a cannot be recognized by* 1*QFA with bounded error.*

Proof. This is shown in [2].

2.1.2. 2-Way Quantum Finite State Automata

The model of 2-way quantum finite state automaton (2QFA) is first introduced by Watrous [2]. 2QFA is more powerful than their classical counterpart. A 2QFA consists of a finite state control and a 2-way tape head—which scans a read only input tape.

Definition 2. *Formally, a* 2-*way QFA is specified by 6-tuplet* $M = (Q, \Sigma, \delta, q_0, Q_a, Q_r)$ *where*

1) Q *is a finite set of states.*

2) Σ *is an input alphabet.*

3) δ *is a transition function which has a mapping of the form* $\delta : Q \times \Gamma \times \{-1, 0, 1\} \to \mathbb{C}$. *In addition to input symbols* Σ, # *and* $ *are symbols that do not belong to* Σ. *We use* # *and* $ *as the left and the right end marker, respectively. The working alphabet of M is* $\Gamma = \Sigma \cup \{\#, \$\}$.

4) $q_0 \in Q$ *is a starting state.*

5) $Q_a \in Q$ *are sets of accepting states.*

6) $Q_r \in Q$ *are sets of rejecting states.*

The states in Q_a *and* Q_r *are called halting states and the states in* $Q_{non} = Q - (Q_a \cup Q_r)$ *are called non-halting states.*

The 2QFA satisfies the following conditions (of well-formedness) for any $q_1, q_2 \in Q$, $\sigma, \sigma_1, \sigma_2 \in \Gamma$, $d \in (-1, 0, 1)$:

1) Local probability and orthogonality condition

$$\sum_{q',d} \overline{\delta(q_1,\sigma,q',d)}\delta(q_2,\sigma,q',d) = 1 \text{ if } q_1 = q_2 \text{ otherwise } 0.$$

2) Separability condition I

$$\sum_{q'} \overline{\delta(q_1,\sigma,q',1)}\delta(q_2,\sigma,q',0) = 0.$$

3) Separability condition II

$$\sum_{q'} \overline{\delta(q_1,\sigma_1,q',0)}\delta(q_2,\sigma_2,q',-1) = 0.$$

4) Separability condition III

$$\sum_{q'} \overline{\delta(q_1,\sigma_1,q',1)}\delta(q_2,\sigma_2,q',-1) = 0.$$

In order to process an input word $x \in \Sigma^*$ by M, we assume that the input is written on the tape with the endmarkers in the form $w_x = \#x\$$ and such a tape of length $|x| + 2$ is circular, *i.e.*, the symbol to the right of $\$$ is $\#$.

For an integer n let C_x be the set (of size $(n + 2)|Q|$) of all possible configuration of M, for inputs of length x. $\delta(q,\sigma,q',d)$ 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 U on tape x) on $H_{|x|}$ as follows:

$$U_\delta^x |q,k| = \sum_{q',d} \delta(q,x(k),q',d)|q', \ k + d \mod |x| \text{ for each } (q,k) \in C_{|x|} \text{ and is extended to all } H_{|x|} \text{ by}$$

linearity. Consider the Hilbert space $l_2(Q)$, where Q is the set of internal states of a 2QFA M. Suppose that we have a linear operator $V_\sigma : l_2(Q) \to l_2(Q)$ for each $\sigma \in \Gamma$ and a function $D: Q \to \{-1,0,1\}$. Define transition function δ as

$$\delta(q,\sigma,q',d) = \langle q'|V_\sigma|q \rangle \text{ if } D(q') = d$$

and

$$\delta(q,\sigma,q',d) = 0 \text{ if } D(q') \neq d$$

M is well-formed when V_σ is unitary.

Theorem 3. *Every regular language is accepted by a 2QFA.*

Proof. The proof has been shown in [2].

2.1.3. Multi-Letter Quantum Finite State Automata

Multi-letter quantum finite state automata has been introduced in [15]. In [16], Qie etal. further study the decidability of the equivalence and minimization problems of multiletter QFAs. In [17], hierarchy and equivalence of multiletter quantum finite state automaton are studied. *k*-letter QFA can be thought of as an analogue of quantum automata with classical memory containing the previously read letters. k-letter QFA is not limited to see only one, just incoming input letter, but can see several up to k of the earlier letter as well.

Definition 3. *Formally, a k-letter QFA M is specified by a 5-tuple $M = (Q,Q_a,|q_0\rangle,\Sigma,\delta)$ where*

1) *Q is a finite set of states,*
2) *$Q_a \in Q$ are set of accepting states.*
3) *$|q_0\rangle$ is the initial quantum state from H_Q.*
4) *Σ is an input alphabet.*
5) *δ is a transition function that assign a unitary trasition matrix V_Γ on $C^{|Q|}$ to each string $\Gamma \in (\Sigma \cup \{\Lambda\})^k$*

and $\Lambda \neq \Sigma$ where C^n denotes Euclidean space consisting of all n-dimensional complex vectors.

A *k*-letter QFA M works in the same way as an measure-once 1-way quantum finite state automaton [1] except that it applies unitary transformation corresponding not only to the last letter but to the last k-letters received. When $k = 1$, it is exactly same as measure once 1-way quantum finite state automaton. According to [16], all languages accepted by k-letter QFAs with bounded error are regular language for any *k*.

To calculate the probability $P_M(x)$ that a *k*-letter QFA accepts an input string $x = \sigma_1,\sigma_2,\cdots,\sigma_n \in \Sigma^*$, it fol-

lows that for any $\Gamma \in \left(\Sigma \cup \{\Lambda\}\right)^k$, $\delta(\Gamma)$ is a unitary matrix. By $\overline{\delta}$ they define a map from Σ^* to the set of of all $|Q| \times |Q|$ unitary matrices. $\overline{\delta}$ is induced by δ in the following way. For $x = \sigma_1, \sigma_2, \cdots, \sigma_n \in \Sigma^*$

$$\overline{\delta} = \delta\left(\Lambda^{k-1}\sigma_1\right)\delta\left(\Lambda^{k-2}\sigma_1\sigma_2\right)\cdots\delta\left(\Lambda^{k-m}x\right) \text{ if } m < k$$

and

$$\overline{\delta} = \delta\left(\Lambda^{k-1}\sigma_1\right)\delta\left(\Lambda^{k-2}\sigma_1\sigma_2\right)\cdots\delta\left(\sigma_{k-m+1}\sigma_{k-m+2}\cdots\sigma_m\right) \text{ if } m \geq k$$

which specifies the computing process of M for an input string x. They identify the states in Q with an orthonormal basis of the complex Euclidean space $C^{|Q|}$ and let P_a denote the projection operator on the subspace spanned by Q_a where

$$P_M(x) = \left\|\langle q_0|\overline{\delta}(x)P_a\right\|^2$$

where $\langle q_0|$ denotes the conjugate transpose of vector $|q_0\rangle$.

3. Multihead Quantum Finite Automata

A k-head quantum finite automaton is a quantum finite automaton having a single read only input tape whose inscription is the input word in between two endmarkers. We define 1-way k-head QFA where k heads of the automaton can move to the right or stay on the current tape square but not beyond the endmarkers.

We show that 1QFA(2) is more powerful than 1RMFA(2).

3.1. 1-Way Multihead Quantum Finite State Automata (1QFA(k))

Definition 4. A 1-*way multihead quantum finite state automaton is a automaton* $M = \left(Q, Q_a, |q_0\rangle, \Sigma, \delta\right)$ *where*

1) Q *is a finite set of states*,
2) $Q_a \in Q$ *are set of accepting states.*
3) $|q_0\rangle$ *is the initial quantum state superposition obeying normalization condition.*
4) Σ *is an input alphabet.*
5) δ *is a transition function that assign a unitary trasition matrix* V_Γ *on* $C^{|Q|}$ *to each string*

$\Gamma \in \left(\Sigma \cup \{\#, \$\}\right)^k$ *where* C^n *denotes Euclidean space consisting of all n-dimensional complex vectors. So* δ *is a mapping of the form* $\delta : Q \times \Gamma^k \times \{0,1\}^k \to \mathbb{C}^Q$ *is the partial transition function where* 1 *means to move the head one square to the right and* 0 *means to keep the head at current square. We use # and \$ as the left and the right end marker,respectively.*

A superposition of M is any element in the Hilbert space $l_2(Q)$. For $q \in Q$, $|q\rangle$ denotes the unit vector with value 1 at q and 0 elsewhere. All elements of $l_2(Q)$ can be expressed as a linear combination of vectors.

The transition function δ maps $Q \times \Gamma^k \times \{0,1\}^k$ to $\mathbb{C}^Q$ where $\mathbb{C}^Q$ denotes the set of complex numbers. The value $\delta\left(q, \sigma_1, \cdots, \sigma_k, q', d_1, \cdots, d_k\right)$ is the amplitude of $|q'\rangle$ in the superposition of states to which M goes from $|q\rangle$ after reading σ_1 by 1st head, σ_2 by 2nd head and so on and moving the heads according to $d_1, d_2, \cdots, d_k$ respectively. The head movement 0 denotes it stays in its position and 1 denotes head is moved to the right. For $\sigma_1, \cdots, \sigma_k \in \Gamma, V_{\sigma_1, \cdots, \sigma_k}$ is a linear transformation on $l_2(Q)$ defined by

$$V_{\sigma_1, \cdots, \sigma_k}\left(|q\rangle\right) = \sum_{q' \in Q} \delta\left(q, \sigma_1, \cdots, \sigma_k, q', d_1, \cdots, d_k\right)|q'\rangle$$

We require all $V_{\sigma_1, \cdots, \sigma_k}$ to be unitary. The check for wellformedness can be done in a similar manner as in [2] in the following way:

Consider the Hilbert space $l_2(Q)$, where Q is the set of internal states of a 1QFA(k) M. Suppose that we have a linear operator $V_{\sigma_1, \cdots, \sigma_k} : l_2(Q) \to l_2(Q)$ for each $\sigma_i \in \Gamma, i = 1 \leq i \leq k$ and a function $D : Q \to \{1, 0\}^k$. Define transition function δ as:

$$\delta\left(q, \sigma, q', d_1, \cdots, d_k\right) = \langle q'|V_{\sigma_1, \cdots, \sigma_k}|q\rangle \quad \text{when } D(q_i) = d_1, \cdots, d_k \tag{2}$$

and

$$\delta\left(q,\sigma,q',d_1,\cdots,d_k\right)=0 \text{ when } D\left(q_i\right)\neq d_1,\cdots,d_k \tag{3}$$

Here $\left\langle q'\middle|V_{\sigma_1,\cdots,\sigma_k}\middle|q\right\rangle$ denotes the coefficient of $\left|q'\right\rangle$ in $V_{\sigma_1,\cdots,\sigma_k}\left|q\right\rangle$. Eventually, M is well-formed if and only if

$$\sum_{q'}\left\langle q''\middle|V_{\sigma_1,\cdots,\sigma_k}\middle|q\right\rangle\left\langle q''\middle|V_{\sigma_1,\cdots,\sigma_k}\middle|q'\right\rangle=1 \text{ if } q=q'$$

and

$$\sum_{q'}\left\langle q''\middle|V_{\sigma_1,\cdots,\sigma_k}\middle|q\right\rangle\left\langle q''\middle|V_{\sigma_1,\cdots,\sigma_k}\middle|q'\right\rangle=0 \text{ if } q\neq q'$$

for each $V_{\sigma_1,\cdots,\sigma_k}$ pair. pair. The condition mentioned is similar to the condition for reversibility in [15].

The input word w begin with # and ends with $. The input is accepted if and only if the computation halts in an accepting states. It halts when the transition function is not defined for the current situation. In all other cases the input is rejected.

3.1.1. Matrices Representation of Different Automaton

In these section we write transition matrices of different automaton and discuss different properties of these automaton in terms of their transition matrices.

1) Deterministic finite state automaton

A deterministic finite automaton [20] consists of five tuple *tuple* $M=\left(Q,\Sigma,\delta,q_0,F\right)$ *where*

1) Q is a finite set of states,
2) Σ is an input alphabet,
3) δ is a transition function that takes as arguments a state and an input symbol and return a state,
4) $q_o \in Q$ is a starting state,
5) $F \in Q$ is a set of final or accepting states.

We design a deterministic finite state automaton $M=\left(\{q_0,q_1\},\{a,b\},\delta,q_0,\{q_1\}\right)$ which accepts all of string having at least one alphabet "*b*" [see **Figure 1**] where

$$\delta\left(q_0,a\right)=q_0$$
$$\delta\left(q_0,b\right)=q_1$$
$$\delta\left(q_1,a\right)=q_1$$
$$\delta\left(q_1,b\right)=q_1$$

The transition matrix of the deterministic finite state automaton is shown in **Figure 2**.

Here each row of each transition matrix contain exactly one non-zero entry *i.e.* 1 for deterministic finite state automaton.
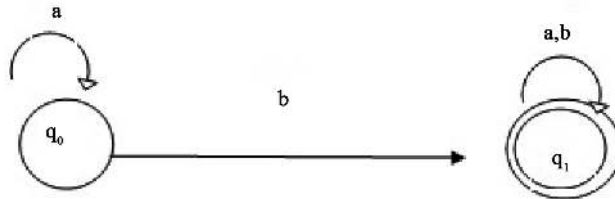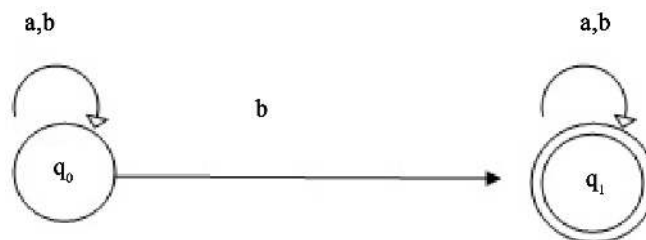

Figure 1. The deterministic finite state automaton accepts all length of string having at least one alphabet "*b*".

$$M_a = \begin{matrix} & q_0 \ \ q_1 \\ \begin{matrix}q_0\\q_1\end{matrix} & \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\end{matrix} \qquad M_b = \begin{matrix} & q_0 \ \ q_1 \\ \begin{matrix}q_0\\q_1\end{matrix} & \begin{pmatrix}0 & 1\\0 & 1\end{pmatrix}\end{matrix}$$

Figure 2. The transition matrix of the deterministic finite state automaton accepts all length of string having at least one alphabet "*b*".

2) Non-deterministic finite state automaton

An non-deterministic finite automaton [20] is represented essentially like a deterministic finite state automaton. It consists of five tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

1) Q is a finite set of states,

2) Σ is an input alphabet,

3) δ is a transition function that takes a state in Q and an input symbol in Σ as arguments and returns a subset of Q,

4) $q_o \in Q$ is a starting state,

5) $F \in Q$ is a set of final or accepting states.

The only difference between an non-deterministic finite state automaton and deterministic finite state automaton is the value of δ that returns a set of states in the case of an non-deterministic finite state automatonand single state in the case of deterministic finite state automaton.We design a non-deterministic finite state automaton $M = (\{q_0, q_1\}, \{a, b\}, \delta, q_0, \{q_1\})$ (shown in **Figure 3**) which accepts all of string having at least one alphabet "*b*" where

$$\delta(q_0, a) = q_0$$

$$\delta(q_0, b) = q_0$$

$$\delta(q_0, b) = q_1$$

$$\delta(q_1, a) = q_1$$

$$\delta(q_1, b) = q_1$$

The transition matrix of the deterministic finite state automaton is shown in **Figure 4**.

There is atleast one row in a transition matrix for non-deterministic automaton which contain more than one non-zero entry.

3) Reversible finite state automaton

An automaton $M = (Q, \Sigma, \delta, I, F)$ [21] is reversible if, for every state p in Q for every letter a in M there existsat most one transition in δ that comes from p (respectively goes to p) with label a. Here

1) Q is a finite set of states,

2) Σ is an input alphabet,

3) δ is a transitions,is a subset of $Q \times \Sigma \times Q$,

4) $I \in Q$ is a set of final states,

5) $F \in Q$ is a set of final states.

Figure 3. The non-deterministic finite state automaton accepts all length of string having at least onealphabet "*b*".

$$M_a = \begin{array}{c} \\ q_0 \\ q_1 \end{array} \begin{array}{cc} q_0 & q_1 \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array} \qquad M_b = \begin{array}{c} \\ q_0 \\ q_1 \end{array} \begin{array}{cc} q_0 & q_1 \\ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \end{array}$$

Figure 4. The transition matrix of the non-deterministic finite state automaton accepts all length of string having at least one alphabet "*b*".

A reversible automaton is a finite automaton in which each letter induces a partial one-to-one map from the set of states into itself. A reversible automaton may have several initial or final states. As a consequence, the minimal automaton of a reversible language may not reversible.

We define reversible automaton $M = (\{q_0, q_1, q_2\}, \{a\}, \delta, q_0, \{q_0\})$ shown in **Figure 5** which accept string of a's of length 3 where

$$\delta(q_0, a) = q_1$$

$$\delta(q_1, a) = q_2$$

$$\delta(q_2, a) = q_0$$

The transition matrix of the above automaton is shown in **Figure 6**.

In case of reversible automaton dot product of any two row is zero and there are no cycles within the transition/output matrix that can't accessed from one of the input states.

4) Probabilistic finite state automaton

A probabilistic finite state automaton [22] over the alphabet Σ is a system consists of $M = (Q, \delta, q_0, F)$ *where*

1) Q is a finite set of states,

2) δ is a transition function from $Q \times \Sigma$ into $[0,1]^{n+1}$ such that for $(q, \sigma) \in Q \times \Sigma$, $\delta(q, \sigma) = (p_0(q, \sigma), \cdots, p_n(q, \sigma)), \ 0 \le p_i(q, \sigma), \ \sum p_i(q, \sigma) = 1$

3) $q_o \in Q$ is a starting state,

4) $F \in Q$ is a set of final or accepting states.

In case of probabilistic finite state automaton we allow the fractional values in transition matrix with the provision that sum of each row give 1 [see **Figure 7**].

5) Quantum finite state automaton

We consider 1-way quantum finite state automata (QFA) as defined in [23] *is a* tuple $M = (Q, \Sigma, \delta, q_0, Q_a, Q_r)$ where

1) Q is a finite set of states,

2) Σ is an input alphabet,
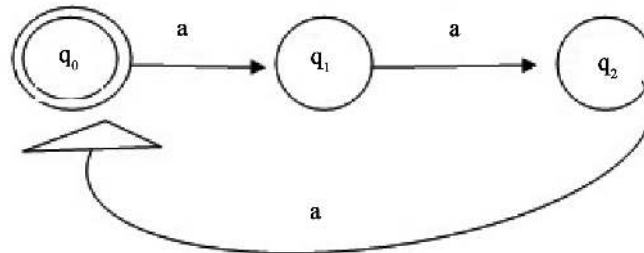
Figure 5. The reversible automaton accept string of *a*'s of length 3.

$$M_a = \begin{array}{c} \\ q_0 \\ q_1 \\ q_2 \end{array} \begin{array}{ccc} q_0 & q_1 & q_2 \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$$

Figure 6. The transition matrix of the reversible automaton accepts string of *a*'s of length of 3.

$$M_a = \begin{array}{c} \\ q_0 \\ q_1 \end{array} \begin{array}{cc} q_0 & q_1 \\ \begin{pmatrix} 0.5 & 0.5 \\ 0 & 1 \end{pmatrix} \end{array} \qquad M_b = \begin{array}{c} \\ q_0 \\ q_1 \end{array} \begin{array}{cc} q_0 & q_1 \\ \begin{pmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{pmatrix} \end{array}$$

Figure 7. The sum of each row give 1.

3) δ is a transition function,

4) $q_0 \in Q$ is a starting state

5) $Q_a \in Q$ and $Q_r \in Q$ are sets of accepting and rejecting states.

The states in Q_a and Q_r are called halting states and the states in $Q_{non} = Q - (Q_a \cup Q_r)$ are called non-halting states. # and $ are symbols that do not belong to Σ. We use # and $ as the left and the right end marker, respectively. The working alphabet of M is $\Gamma = \Sigma \cup \{\#,\$\}$. Quantum finite state automaton is obtainby letting the transition matrix with complex entries.We also require each of the matrices to be unitary.

The transition matrix of the quantum finite state automaton looks like [**Figure 8**]:

The transition matrix is unitary since the sum of the squares of the norms in each row adds up to 1 and the dot product of any two row is 0.If all matrices only have 0 or 1 entries and the matrices are unitary,then the automaton is deterministic and reversible.

6) 1-way multihead deterministic finite state automaton

A 1-way k-head deterministic finite state automaton is a deterministic finite state automaton with k-independent read heads on a single input tape with the end markers. On each move the machine can simultaneously read the k input cells scanned by k-heads,move each head one square to the right or keep stationary.

A 1-way multihead deterministic finite state automaton (1DFA(k)) [13] is a tuple $M = (Q, \Sigma, k, \delta, \#, \$, q_0, F)$ where

1) Q is a finite set of states,

2) Σ is an input alphabet,

3) $k \geq 1$ is the number of heads.

4) $\delta : Q \times (\Sigma \cup \{\#,\$\})^k \to Q \times \{0,1\}^k$ is the partial transition function;where 1 means to move the head one square to the right and 0 means to keep the head on the current square,

5) $\# \notin \Sigma$ is the left and $\$ \notin \Sigma$ is the right endmarkers.

6) $q_o \in Q$ is a starting state,

7) $F \in Q$ is a set of final or accepting states.

We define a 1DFA(2) $M = (\{q_0,q_1,q_2\},\{a,b\},2,\delta,\#,\$,q_0,\{q_2\})$ shown in **Figure 9** which accept $L = \{a^n b^n \mid n \geq 1\}$ where

$$\delta(q_0,\{b,a\}) = q_1$$
$$\delta(q_1,\{b,a\}) = q_1$$
$$\delta(q_1,\{a,b\}) = q_2$$
$$\delta(q_2,\{\$,b\}) = q_2$$

$$M = \begin{pmatrix} -1 & 0 \\ 0 & i \end{pmatrix}$$

Figure 8. The transition matrix of the automaton contain complex entry.
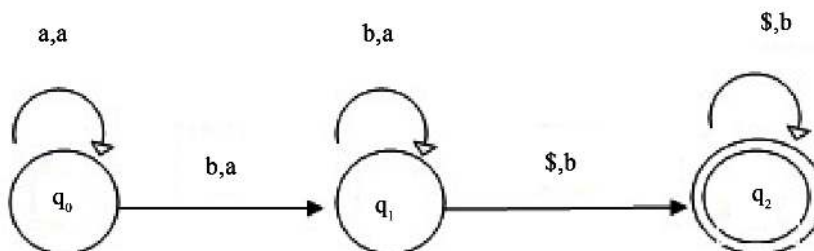
Figure 9. 1DFA(2) accept a language $L = \{a^n b^n \mid n \geq 1\}$ labelover flow.

The transition matrix of the above automaton is [see **Figure 10**].

Each row of transition matrix contain only one 1 which has the same property as deterministic finite state automaton.

7) 1-way Reversible multihead finite state automaton

A 1-way reversible multihead finite state automaton (1REV-DFA(k)) [13] is a tuple $M = (Q, \Sigma, k, \delta, \#, \$, q_0, F)$ which has same structure as 1DFA(k) where

1) Q is a finite set of states,

2) Σ is an input alphabet,

3) $k \geq 1$ is the number of heads.

4) $\delta : Q \times (\Sigma \cup \{\#, \$\})^k \to Q \times \{0,1\}^k$ is the partial transition function;where 1 means to move the head one square to the right and 0 means to keep the head on the current square,

5) $\# \notin \Sigma$ is the left and $\$ \notin \Sigma$ is the right endmarkers.

6) $q_o \in Q$ is a starting state,

7) $F \in Q$ is a set of final or accepting states.

Let M be a 1DFA(k) and D be the set of all reachable configuration that occur in any computation of M beginning with an initial configuration and $(w, q, (p_1, \cdots, p_k)) \in D$ with $w = \sigma_1, \cdots, \sigma_n$, $\sigma_0 = \#$ and $\sigma_{n+1} = \$$. D be the set of all reachable configurations that occur in any computation. M is said to be reversible if the following two conditions are fulfilled:

1) For any two transitions:

$$\delta(q, (\sigma_1, \cdots, \sigma_n)) = (q_1, (d_1, \cdots, d_k))$$

and

$$\delta(q', (\sigma_1, \cdots, \sigma_n)) = (q_1, (d_1', \cdots, d_k'))$$

it holds if $(d_1, \cdots, d_k) = (d_1', \cdots, d_k')$.

2) There is at most one transition of the form

$$\delta(q', (x_{p_1 - d_1}, \cdots, x_{p_k - d_k})) = (q, (d_1, \cdots, d_k)).$$

The non-context free language $L = \{a^n b^n c^n \mid n \geq 1\}$ is accepted by REV-1DFA(2)

$M = (\{q_0, q_1, q_2, q_3, q_f\}, \{a,b,c\}, 2, \delta, \#, \$, q_0, \{q_f\})$ shown in **Figure 11** where the transition function δ is as follows:

$$M_{aa} = \begin{array}{c} \\ q_0 \\ q_1 \\ q_2 \end{array} \begin{array}{ccc} q_0 & q_1 & q_2 \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \qquad M_{ba} = \begin{array}{c} \\ q_0 \\ q_1 \\ q_2 \end{array} \begin{array}{ccc} q_0 & q_1 & q_2 \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \qquad M_{\$b} = \begin{array}{c} \\ q_0 \\ q_1 \\ q_2 \end{array} \begin{array}{ccc} q_0 & q_1 & q_2 \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

Figure 10. The transition matrix of1DFA(2) accept a language $L = \{a^n b^n \mid n \geq 1\}$.

Figure 11. 1REV-DFA(2) accept a language $L = \{a^n b^n c^n \mid n \geq 1\}$.

$$\delta(q_0, \#, \#) = (q_0, 0, 1)$$

$$\delta(q_0, \#, a) = (q_0, 0, 1)$$

$$\delta(q_0, \#, b) = (q_1, 1, 1)$$

$$\delta(q_1, a, b) = (q_1, 1, 1)$$

$$\delta(q_1, a, c) = (q_2, 1, 1)$$

$$\delta(q_2, b, c) = (q_2, 1, 1)$$

$$\delta(q_2, b, \$) = (q_3, 1, 0)$$

$$\delta(q_3, c, \$) = (q_3, 1, 0)$$

$$\delta(q_3, \$, \$) = (q_f, 0, 0)$$

The transition matrix of the above automaton is shown in **Figure 12**.

Dot product of any two row is zero for multihead reversible finite state automaton.

8) 1-way multihead quantum finite state automaton

1-way multihead quantum finite state automaton is a 1-way k-head quantum finite state automaton where k-heads of the automaton can move to the right or stay on the current tape square but not beyond the end markers. The language $L = \{ w \neq w^R \}$ is accepted by the 1QFA(k)

$$M = \left(\{ |q_0\rangle, |q_1\rangle, |q_2\rangle, |q_3\rangle, |q_f\rangle \}, \{ |q_f\rangle \}, \{ |q_0\rangle \}, \{a, b\}, \delta \right) \text{ [see Figure 13] where}$$

$$\delta(|q_0\rangle, \#, \#) = (|q_0\rangle, 0, 1)$$

$$\delta(|q_0\rangle, \#, a) = (|q_0\rangle, 1, 1) + \delta(|q_1\rangle, 1, 1)$$

Figure 12. The transition matrix of 1REV-DFA(2) accept a language $L = \{ a^n b^n c^n \mid n \geq 1 \}$.

$$\delta\big(|q_0\rangle,\#,b\big)=\big(|q_0\rangle,1,1\big)+\delta\big(|q_2\rangle,1,1\big)$$

$$\delta\big(|q_1\rangle,a,b\big)=\big(|q_1\rangle,1,1\big)$$

$$\delta\big(|q_1\rangle,b,a\big)=\big(|q_1\rangle,1,1\big)$$

$$\delta\big(|q_2\rangle,a,b\big)=\big(|q_2\rangle,1,1\big)$$

$$\delta\big(|q_2\rangle,b,a\big)=\big(|q_2\rangle,1,1\big)$$

$$\delta\big(|q_1\rangle,b,\$\big)=\big(|q_f\rangle,0,0\big)$$

$$\delta\big(|q_2\rangle,a,\$\big)=\big(|q_f\rangle,0,0\big)$$

The transition matrix of the above automaton is shown in **Figure 14**.

The sum of the square of the norms in each row adds up to 1 and dot product of any two row is zero formultihead quantum finite state automaton.

3.1.2. Recognition of Language Class

In this section we show that 1QFA(k) has more language recognizing power than 1QFA. 1QFA(2) can recognize regular language $L=(a,b)^{*}a$ and context-sensitive language $L=\{a^{n}b^{n}c^{n}\,|\,n\geq1\}$ respectively.

Theorem 4. 1QFA(2) can accept all unary regular languages.

Proof. In [12] it has been shown that any unary regular language is accepted by some 1-way reversible 2-headdeterministic finite automaton. We find from the previous section that in a 1-way multihead quantum

Figure 13. 1QFA(2) accept a language $\{w\neq w^{R}\}$ with acceptance probability $p > 0$.

Figure 14. The transition matrix of 1QFA(2) accept a language $\{w\neq w^{R}\}$ with acceptance probability $p > 0$.

finitestate automaton where the transition matrices are only 0 and 1 entry, it is essentially a 1-way reversible multihead finite state automaton. So 1-way 2-head quantum finite state automaton accept all unary language.

Example 1. *A* 1-*way* 2-*head quantum finite state automaton is a automaton* $M = (Q, Q_a, |q_0\rangle, \Sigma, \delta)$ *can accept* $L = (a, b)^* a$ *in the folowing manner*:

Let, $Q = \{q_0, q_1, q_f\}$, $\Sigma = \{a, b\}$, $Q_a = \{q_f\}$

Define:

$$V_{\#\#}|q_0\rangle = |q_0\rangle$$

$$V_{\#x}|q_0\rangle = |q_1\rangle, x \in (a, b)$$

$$V_{xy}|q_1\rangle = |q_1\rangle, x, y \in \{a, b\}$$

$$V_{a\$}|q_1\rangle = |q_f\rangle$$

$$D(q_0) = (0, 1)$$

$$D(q_1) = (1, 1)$$

$$D(q_f) = (1, 0)$$

The automaton acts as follows: *Initially both heads of the automaton M are at #. After reading the input symbols, the automaton M remain at state* $|q_0\rangle$. *The first head remain stationary where the second head move one square to the right of the input tape whenever the automaton reaches at state* $|q_0\rangle$. *The movement of heads are similar as the previous case. Due to the movement of heads the second head may be at "a" or "b".*

For both cases the automaton M move to state $|q_1\rangle$ *from state* $|q_0\rangle$ *and both heads move one square to theright of the input tape. When the first head at "a" and second head at "$", the automaton M goes to final state* $|q_f\rangle$ *from* $|q_1\rangle$ *and the string will be accepted by the automaton M with probability 1.*

Consider a string w not in L. As w is not in L the heads of the automaton M will arrived in such a way that for that particular position of heads and state, no transition rules are defined. So, for a string w, which M does not accept, there is no sequence of transitions that makes M to its final state after consumption of w. So, M rejects with probability 1. Each pairs of $V_{\sigma_1 \sigma_2}$ *is unitary.*

Example 2. *A* 1-*way* 2-*head quantum finite state automaton is a automaton* $M = (Q, Q_a, |q_0\rangle, \Sigma, \delta)$ *can accept* $L = \{a^n b^n c^n \mid n \geq 1\}$ *in the following manner*:

Let, $Q = \{q_0, q_1, q_2, q_3, q_f\}$, $\Sigma = \{a, b\}$, $Q_a = \{q_f\}$

Define:

$$V_{\#\#}|q_0\rangle = |q_0\rangle$$

$$V_{\#a}|q_0\rangle = |q_0\rangle$$

$$V_{\#b}|q_0\rangle = |q_1\rangle$$

$$V_{ab}|q_1\rangle = |q_1\rangle$$

$$V_{ac}|q_1\rangle = |q_2\rangle$$

$$V_{bc}|q_2\rangle = |q_2\rangle$$

$$V_{b\$}|q_2\rangle = |q_3\rangle$$

$$V_{c\$}|q_3\rangle = |q_3\rangle$$

$$V_{\$\$}|q_3\rangle = |q_f\rangle$$

$$D(q_0) = (0, 1)$$

$$D(q_1) = (1, 1)$$

$$D(q_2) = (1,1)$$

$$D(q_3) = (1,0)$$

$$D(q_f) = (0,0)$$

The automaton acts as follows: *Initially both heads of the automaton M are at #. After reading the input symbols, the automaton M remains at $|q_0\rangle$ and the first head remain stationary and the second head moveone square to the right of the input tape. Whenever the automaton M reach at state $|q_0\rangle$ the movement of heads are similar as the above case. The automaton M performs similar task as previous one for all inputletter "a" until the next input letter read by second head is "b". For reading input letter "b" by the second head of the automaton M, it moves to state $|q_1\rangle$ and remains at state $|q_1\rangle$ until the next input letter read by thesecond head is "c". This steps counts the number of letter "b" with number of letter "a". Both heads move one square to the right of the input string for state $|q_1\rangle$. After reading the letter "c" by second head where the first is reading letter "a" of the input string, the automaton M goes to state $|q_2\rangle$ from state $|q_1\rangle$. Both heads moveone square to the right of the input string whenever $|q_1\rangle$ state will be reached. The automaton M now checks the number of letter "b" and "c" of the input string with two heads of the automaton until the second head readthe right endmarker $\$$, for which the automaton goes to state $|q_3\rangle$. When the automaton M is in state $|q_3\rangle$ and both heads read the right endmarker $\$$, then the automaton accept the input string with probability 1.*

Consider a string w not in L. As w is not in L the heads of the automaton M will arrived in such a way that for that particular position of heads and state, no transition rules are defined. So, for a string w, which M does not accept,there is no sequence of transitions that makes M to its final state after consumption of w. So, M rejects with probability 1. Condition of unitarity is satisfied for all pairs of $V_{\sigma_1\sigma_2}$.

Theorem 5. *1QFA(2) is more powerful than 1QFA with respect to recognition of language.*

Proof. In Theorem 2 it was proved that the language $L = \{a,b\}^* a$ cannot be recognized by 1QFA with bounded error. In Example 1 we proved that it can be done with 1QFA(2).

Theorem 6. *Given a language $S = \{w_1, \cdots, w_k\}^*$ it is possible in general case to build a 1QFA(k) that recognizes this language.*

Proof. In [24] it has been shown that this language consists of subset of words from $L = \{0,1\}^* 1$ language whichis recognize by 1QFA(2) as shown in Example 1.

Example 3. *A 1-way 2-head quantum finite state automaton is a automaton $M = (Q, Q_a, |q_0\rangle, \Sigma, \delta)$ can accept $L = \{w \neq w^R\}$ in the following manner:*

Let, $Q = \{q_0, q_a, q_b, q_{acc}\}$, $\Sigma = \{a,b\}$, $Q_a = \{q_{acc}\}$

Define:

$$V_{\#\#}|q_0\rangle = |q_0\rangle$$

$$V_{\#a}|q_0\rangle = \frac{1}{\sqrt{2}}|q_0\rangle + \frac{1}{\sqrt{2}}|q_a\rangle$$

$$V_{\#b}|q_0\rangle = \frac{1}{\sqrt{2}}|q_0\rangle + \frac{1}{\sqrt{2}}|q_b\rangle$$

$$V_{xy}|q_a\rangle = |q_a\rangle, x, y \in \{a,b\}$$

$$V_{xy}|q_b\rangle = |q_b\rangle, x, y \in \{a,b\}$$

$$V_{b\$}|q_a\rangle = |q_{acc}\rangle$$

$$V_{a\$}|q_b\rangle = |q_{acc}\rangle$$

$$D(q_0) = (0,1)$$

$$D(q_a) = (1,1)$$

$$D(q_b) = (1,1)$$

$$D(q_{acc}) = (0,0)$$

The automaton acts as follows: at each reading of the symbol $x \in \Sigma$ the automaton goes into a super-position of two states $|q_0\rangle$ and $|q_x\rangle$ with amplitude a $\frac{a}{\sqrt{2}}$ respectively.This is done with the objective that at each reading of the symbol the automaton guesses x to be the first character which does not match. Thus if the guess is rightthen the path corresponding to $|q_x\rangle$ goes to an accepting states and the guess is wrong the path correspondingto $|q_x\rangle$ goes to an rejecting state. Even if the guess is wrong the path corresponding to $|q_0\rangle$ allows us to makeanother guess. So if the input word belongs to L and k^{th} letter does not match with theend.

Then at the kth depth $|q_x\rangle$ will reach an accepting states with probability $\left(\frac{1}{\sqrt{2}}\right)^k$. But if the input word is an palindrome and does notbelong to L then no matter what depth we traversed $|q_x\rangle$ will never go to an accepting state. As the aboveautomaton is 1-way and the length of the input word is finite, the above automaton will always halt in finitetime and the input string belonging to the language is accepted by the automaton if the probability of state $|q_{acc}\rangle$ is greater than 0. We arrive at this conclusion because if the word is palindrome the probability of $|q_{acc}\rangle$ will be 0. Each $V_{\sigma_1\sigma_2}$ is unitary by inspection. So M is well-formed. **Figure 15** and **Figure 16** shown below describe the working steps of the automaton in pictorial form.
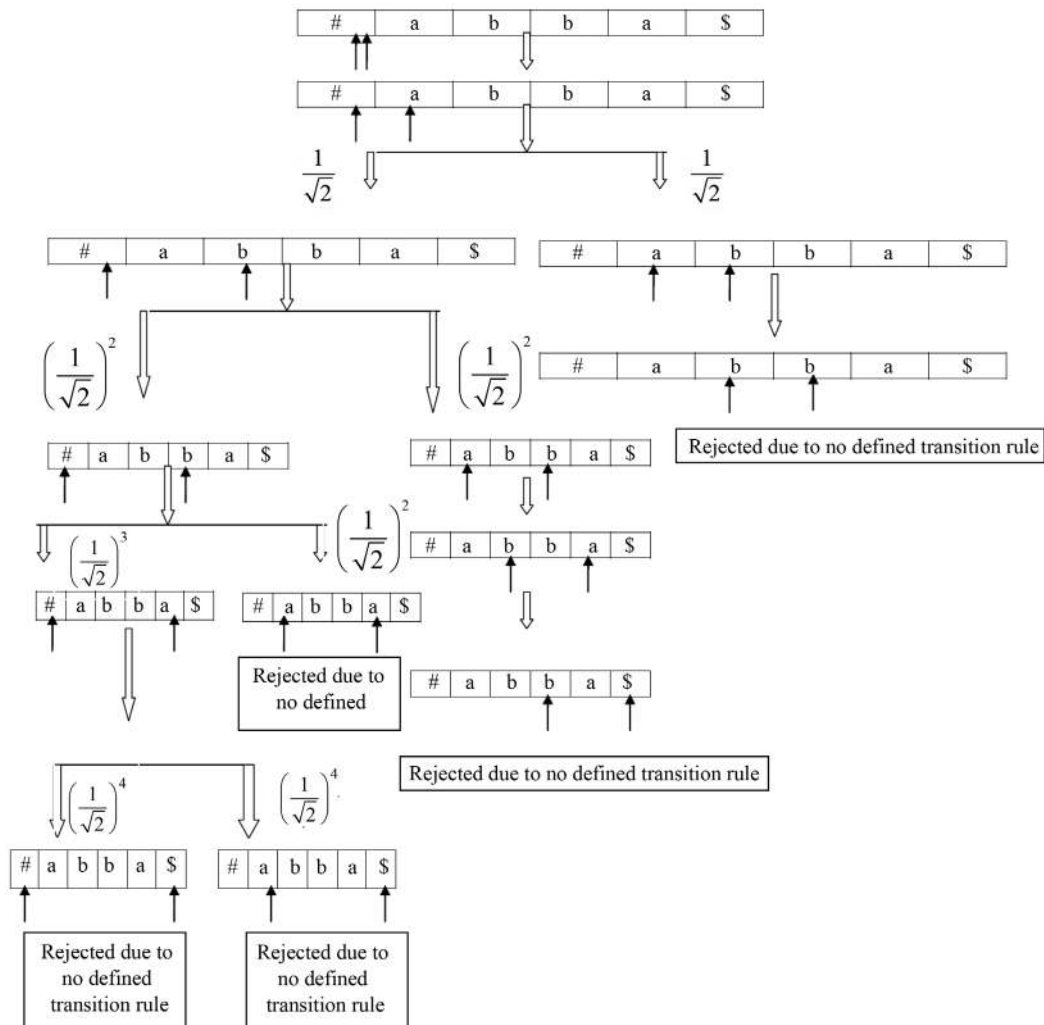
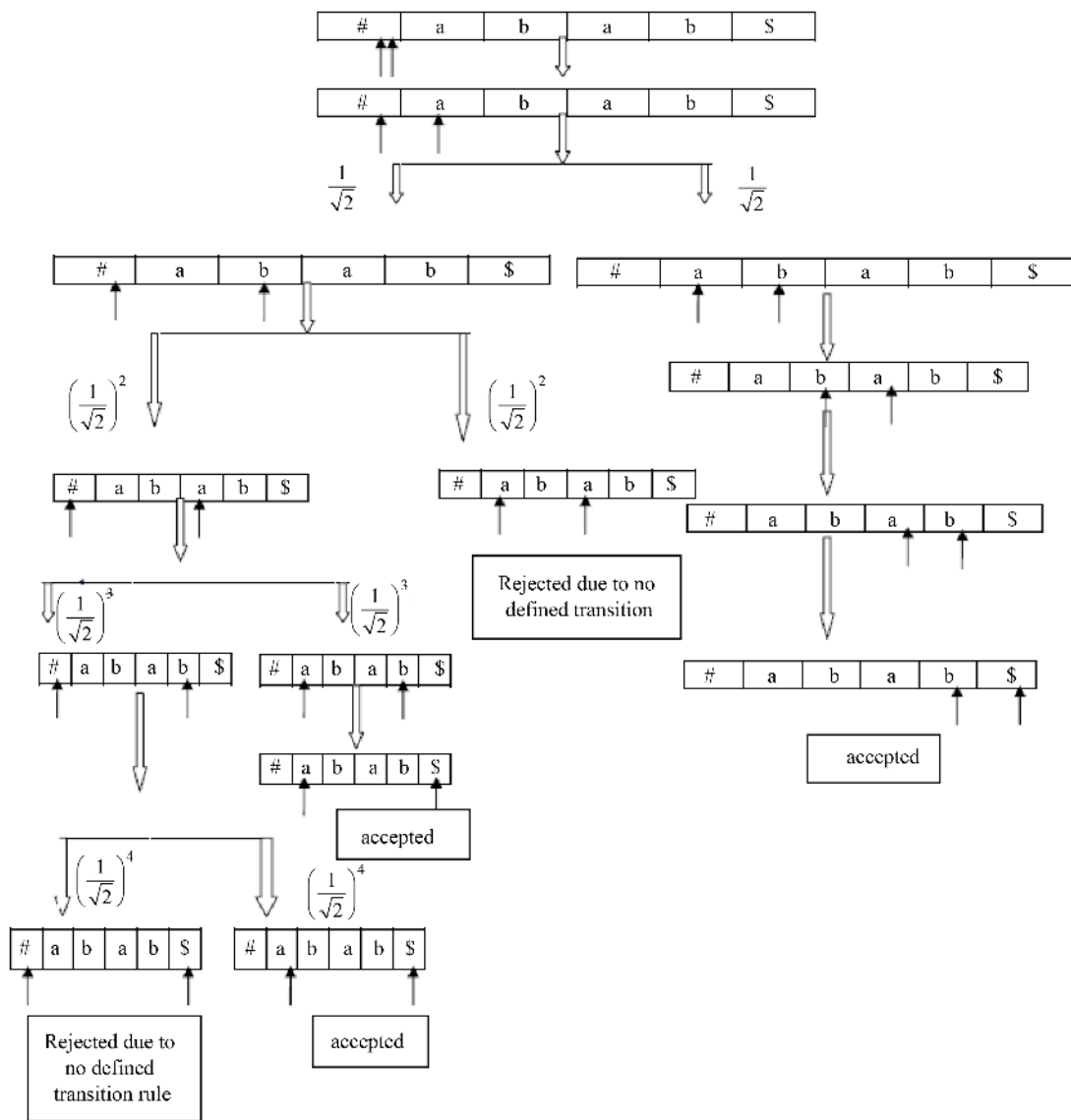**Figure 15.** Input "abba" is palindrome and hence it is not accepted.

Figure 16. Input "abba" is not palindrome and hence it is accepted.

Theorem 7. $L_{1QFA(2)} - L_{1DFA(2)} \neq \varnothing$ *where* $L_{1QFA(2)}$ *is the set of all languages accepted by* 1-*way* 2-*head quantum finite state automaton and* $L_{1DFA(2)}$ *is the set of all languages accepted by* 1-*way* 2-*headdeterministic finite state automaton.*

Proof. The language $L = \{w \neq w^R\}$ cannot be recognized by 1DFA(2) ([18] [19]). In Example 3 we proved that it can be done with 1QFA(2).

Theorem 8. *For every* 1-*way reversible* 2-*head finite state automaton M which accepts a language L, thereexists a* 1-*way* 2-*head quantum finite state automaton M' which accepts the same language L.*

Proof. We know that the transition matrix of 1-way reversible multihead finite state automaton has the following properties:

1) Dot product of any two row is zero for 1-way reversible multihead finite state automaton.

2) All matrices only have 0 or 1 entries.

Therefore the above two properties of the transition matrix ensures that the transition matrix is also unitary. As a result given a 1-way reversible 2-head finite state automaton M we get a 1-way 2-head quantum finite state automaton M' which has the same transition matrix,same set of states, same set of accepting states and start state as M. as the transition matrix, start state and accepting states of M and M' are same,they accept the same

language.

Theorem 9. *The set of all languages accepted by* 1*-way reversible* 2*-head finite state automata* (1RMFA(2)) *is a proper subset of set of all language accepted by* 1*-way* 2*-head quantum finite state automata.* (1QFA(2))

Proof. Theorem 8 tells us that for every 1RMFA(2) which accept a language L there exist 1QFA(2) which accept the same language. So, the set of all languages accepted by 1RMFA(2) is a subset of set of all languages accepted by 1QFA(2). From ([18] [19]) we know that the language $L = \left\{ w \neq w^R \right\}$ is not accepted by any 1DFA(2). Also from ([13]) we know that the set of all languages accepted by 1RMFA(2) is a proper subset of 1DFA(2). Therefore, there is no 1RMFA(2) which accepts the language $L = \left\{ w \neq w^R \right\}$. In Example 3 it has been shown 1QFA(2) accept the language L. Thus, the subset relation is proper.

Corollary 1. 1QFA(2) is computationally more powerful than 1RMFA(2).

4. Conclusion

In this paper, we studied characteristics of 1QFA(k) with their language accepting capability. There are still many non-regular context free context sensitive languages accepted by 1QFA(k) other than shown in this paper. We show that $L_{1QFA(2)} - L_{1DFA(2)} \neq \varnothing$ where $L_{1QFA(2)}$ is the set of all languages accepted by 1-way 2-head quantum finite state automaton and $L_{1DFA(2)}$ is the set of all languages accepted by 1-way 2-head deterministic finite state automaton. We also show that 1QFA(2) is more powerful than 1RMFA(2) with respect to recognition of language. But though 1QFA(2) can accepts some non-regular languages but it is still not be proved that whether 1QFA(2) can accepts all regular languages. We can explore it by using the superposition property of 1QFA(k).

Acknowledgements

References

[1] Moore, C. and Crutchfield, J. (1997) Quantum Automata and Quantum Grammars. *Theoretical Computer Science*, **237**, 275-306. http://dx.doi.org/10.1016/S0304-3975(98)00191-1

[2] Kondacs, A. and Watrous, J. (1997) On the Power of Quantum Finite State Automata. *Proceedings of the* 38*th Annual Symposium on Foundations of Computer Science*, Miami, 66-75. http://dx.doi.org/10.1109/SFCS.1997.646094

[3] Ambainis, A. and Freivalds, R. (1998) One-Way Quantum Finite Automata: Strengths, Weakness and Generalizations. *IEEE* 39*th Annual Symposium on Foundations of Computer Science*, 332-342. http://dx.doi.org/10.1109/SFCS.1998.743469

[4] Ambainis, A., Bonner, R.F., Freivalds, R. and Kikusts, A. (1999) Probabilities to Accept Languages by Quantum Finite Automata. *COCOON*, 174-183. http://dx.doi.org/10.1007/3-540-48686-0_17

[5] Ambainis, A., Bcandry, M., Golovkins, M., Kikusts, A., Mercer, M. and Therien, D. (2004) Algebric Results on Quantum Automata. *STACS*, 93-104.

[6] Bertoni, A., Mereghetti, C. and Palano, B. (2003) Quantum Computing: 1 way Quantum Automata. *Developments Language Theory*, 1-20.

[7] Ambainis, A. and Watrous, J. (2002) Two Way Finite Automata with Quantum and Classical States. *Theoretical Computer Science*, **287**, 299-311.

[8] Ambainis, A., Beaudry, M., Golovkins, M., Kikusts, A., Mercer, M. and Therien, D. (2004) Algebraic Results on Quantum Automata. In: Diekert, V. and Habib, M., Eds., *STACS*, *LNCS*, Vol. 2996, Springer, Heidelberg, 93-104.

[9] Dzelme, I. (2003) Kvantu Automar Jauktajiem Stavokliem. Technical Report, University of Latvia.

[10] Nayak, A. (1999) OPtimal Lower Bounds for Quantum Automata and Random Access Codes. *Proceedings of the* 40*th Annual Symposium on Foundations of Computer Science*, 369-377. http://dx.doi.org/10.1109/sffcs.1999.814608

[11] Rabin, M.O. and Scott, D. (1964) Finite Automata and Their Decision Problems. *Sequential Machines*, Selected Papers, Addition-Wesley, 63-91.

[12] Rosenberg, A.L. (1966) On Multihead Finite Automata. *IBM Journal of Research and Development*, **10**, 388-394. http://dx.doi.org/10.1147/rd.105.0388

[13] Kutrib, M. and Malchar, A. (2013) One-Way Reversible Multi-Head Finite Automata. *Reversible Computation*, *Lec-*

ture Notes in Computer Science, **7581**, 14-28. http://dx.doi.org/10.1007/978-3-642-36315-3_2

[14] Morita, K. (2011) Two-Way Reversible Multi-Head Finite Automata. *Fundamenta Informaticae*, IOS Press, 241-254.

[15] Belovs, A., Rosmanis, A. and Smotrovs, J. (2007) Multi-Letter Reversible and Quantum Finite Automata. *Proceedings of the* 13*th International Conference on Developments in Language Theory* (*DLT'*2007), *Lecture Notes in Computer Science*, Vol. 4588, Springer, Berlin, 60-71. http://dx.doi.org/10.1007/978-3-540-73208-2_9

[16] Qiu, D., Li, L., Zou, X., Mateus, P. and Gruska, J. (2011) Multi-Letter Quantum Finite Automata: Decidability of the Equivalence and Minimization of States. *Acta Informatica*, 271-290.

[17] Qiu, D. and Yu, S. (2006) Hierarchy and Equivalence of Multi-Letter Quantum Finite Automata. *Theoretical Computer Science*, **356**, 190-199.

[18] Ibarra, H.O. and Ravikumar, B. (2009) On Partially Blind Multihead Automata. *Theoretical Computer Science*, **410**, 3006-3017.

[19] Wagner, K. and Wechsung, G. (1986) Computational Complexity. D Reidel Publishing Company.

[20] Hopcroft, E.J.,Motwani, R. and Ullman, D.J. (2012) Introduction to Automata Theory. *Languages*, *and Computation*, Pearson, 3rd Edition.

[21] Sylvain, L. (2002) On the Construction of Reversible Automata for Reversible Language. *Automata*, *Languages and Programming*, *Volume* 2380 *of the Series Lecture Notes in Computer Science*, 170-182.

[22] Rabin, O.M. (1963) Probabilistic Automata. *Information and Control*, **6**, 230-245. http://dx.doi.org/10.1016/S0019-9958(63)90290-0

[23] Gruska, J. (2000) Quantum Computing. McGraw Hill, 153-157.

[24] Skuskovniks, A. On Languages Not Recognizable by One-Way Measure Many Quantum Finite Automaton. Supported by ESF Project 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044.