DESIGN AND SIMULATION OF TWO-WAY QUANTUM FINITE AUTOMATA

by

Fatih Mehmet ATAK

B.S., Computer Engineering, Boğaziçi University, 2003

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University

2006

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Prof. Cem Say, for his invaluable guidance and help. This would never be done without his efforts.

I am grateful to my friends, which supported and helped me with their experiences and insightful discussions.

Also, I am grateful to ITG Ltd, especially Celil Germeyan, for the tolerance and support during preparation of the thesis.

And finally, I want to thank my family, especially my wife, for their unlimited support and motivation from the beginning to the end. Without their support, I would have fallen at the beginning, and it is to them that I would like to dedicate this thesis.

ABSTRACT

DESIGN AND SIMULATION
OF
TWO-WAY QUANTUM FINITE AUTOMATA

In this thesis, we review classical finite state automata, (FSAs and TWAs) and 2-way quantum finite state automata (2QFAs). We examine fundamental theorems and their proofs.

We develop a software simulator of quantum finite state automata. We introduce the simulator and by the help of the simulator, we examine some non-regular languages like $\{0^n1^n \mid n > 0\}$ (type 2) and $\{0^n1^n2^n \mid n > 0\}$ (type 1).

We also propose a new technique to enhance the language recognition probability of 2QFAs. This method may allow some languages to be recognized with bounded error, if we have an algorithm for the unbounded error version. A sample construction for such a case is inspected in detail. Using this technique, we enhance the complexity of a 2QFA which originally recognize string s with error probability of 1/2, beyond a well known 2QFA in the literature for the same language.

ÖZET

KUANTUM ÇİFT YÖNLÜ SONLU DURUM MAKİNELERİNİN DİZAYN VE SİMULASYONU

Bu tezde, klasik sonlu durum makineleri (tek ve çift yönlü) ve kuantum çift yönlü sonlu durum makineleri ve bunlarla ilgili temel kuram ve ispatlar gözden geçirilmiştir.

Ayrıca, kuantum çift yönlü sonlu durum makineleri için bir simülatör geliştirilmiştir. Geliştirilen simülatör kullanılarak, $\{0^n1^n \mid n > 0\}$ (tip 2) ve $\{0^n1^n2^n \mid n > 0\}$ (tip 1) gibi düzenli olmayan olan çeşitli diller incelenmiştir.

Bunların yanı sıra, kuantum sonlu durum makineleri için dil tanıma olasılığını arttıracak bir metot üzerinde çalışılmıştır. Bu metot sayesinde, orijinal halde bir dili sınırlandırılmış hata ile tanıyamayan makineler, tanıyabilecek hale getirilebilmektedir. Bu metodu uygulayarak, literatürdeki bilinen bir makine üzerinde iyileştirmeler yapılmıştır. Ayrıca, çeşitli diğer makineler için de metodun uygulanması ve sonuçları incelenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Up to now, a lot of work has been done to explore the details of classical finite accepters, but their quantum counterparts have not been explored that deeply. In this thesis, we examine Quantum Finite Accepters.

To aid us in this project, we decided to use a simulation software. This way, we could have a better understanding of their working principles. But because of some reasons described later, we could not use the existing simulators, so we implemented a new 2QFA simulator. This simulator had some properties which makes life easier while dealing with 2QFAs. We have implemented 2QFAs for a number of languages which have previously been presented in the literature, and a number of languages which were not previously discussed.

We also developed a method for enhancing language recognition probability of some specific 2QFAs and inspected the applicability of our method to some 2QFAs we have implemented.

To sum up, our work can be grouped in three parts:

I. Writing a 2QFA simulator

II. Designing some machines recognizing some famous languages, and verifying that the construction is feasible.

III. A proposal to enhance the language recognition probability of some specific 2QFAs.

In the remainder of this thesis, we will overview classical finite automata and inspect quantum finite automata in detail in chapter 2. We will inspect the existing simulators and our simulator in chapter 3. In chapter 4, we will present our experiments with the simulator, which includes implementations of some famous languages in the literature, and design of some languages which are not in the literature. In chapter 5, we discuss our new approach for enhancing the language recognition probability of 2QFAs. Chapter 6 is the conclusion.

2. BACKGROUND

We will present a short overview of classical and quantum finite accepters.

2.1. Classical Case

Definition 2.1. A nondeterministic Finite State Accepter (FSA) is a five-tuple M = (Q, S, P, I, F), in which

Q is finite set of states,

S is a finite input alphabet,

$I \subseteq Q$ is the set of initial states,

$F \subseteq Q$ is the set of final states,

$P \subseteq Q \times S \times Q$ is the *transition relation* of M:

whenever (q, s, q') is an element of P, then

$q =^{s}=> q'$ is a transition of M.

Definition 2.2. If an FSA M has only one initial state and the transitions P of M form a function defined as P: $Q \times S => Q$, then it is called a Deterministic FSA or shortly DFA.

Definition 2.3. A FSA processes an input string by consuming its symbols in order and changing state according to the transitions for each such symbol. If a FSA M is at a final state after processing an input string w, then w is accepted by M. Otherwise, if it is not on a final state, or there are no transitions to process whole string, then w is rejected by M.

We will not go into more details of usual FSA, but just present the following as is. If you require details, please refer to the references: [3]

- The language recognition powers of nondeterministic FSA and deterministic FSA are the same. They recognize all and only the type 3 (regular) languages.

- Even if we add lambda (empty string) transitions to the FSA, this does not improve the language recognition capability of the accepter. They will still recognize only the regular languages.

We now slightly modify the accepter model, and we add a tape for storing the input.

Definition 2.4. A one way deterministic finite state accepter (1DFA) is a five-tuple $M = (S, \Sigma, \mu, s_0, F)$, in which

S is finite set of states,

Σ is a finite input alphabet,

$\mu \subseteq Q \times S \times Q$ is the *transition function* of M:

whenever $\mu(q, s) = q'$ holds, then $q =^{s}=> q'$ is a transition of M.

$s_0 \subseteq Q$ is the initial state,

$F \subseteq Q$ is the set of final states,

Basically a 1DFA is a FSA, but it is deterministic and it reads its input from a one way traversable tape. If we allow the tape to be traversable in two ways, this will reduce the required state count.

Definition 2.5. A two way accepter (TWA), which may also be referred as 2DFA/2NFA (either deterministic or non-deterministic) is a five-tuple M = (Q, S, P, I, F), where

Q is a finite set of internal states, including initial states I and final states F

S is a finite input alphabet

P is a program

Each instruction in the program P is of the form

q] left (s, q') or **q] right (s, q')**

where

$q, q' \in Q$ and

$s \in S \cup \{\#\}$, # is the blank symbol and **$\{\#\} \notin S$.**

Each state has only one type of instructions, either left or right. And initially the tape head is on position 0, i.e. the input string begins on the square on the right of the tape head.

Definition 2.6. The configuration of an automaton allows us to uniquely determine the complete information about the automaton at a certain point of the computation. A TWA has two main parts, a tape and a computational unit. The tape is read only, so we only need to know the head position to determine the tape state at a point. The computation is not history dependent, so we only need to know the current state. Therefore, the configuration of a TWA is a 2 tuple, (q, k), where q is the current state and k is the current tape head position.

Definition 2.7. The execution principle of a TWA M with the input w can be summarized as follows:

1. An instruction q]right(s, q') will be executed in a configuration (q, k), if the k+1st tape square contains the symbol s. After the execution, the tape head is moved one square to the right, the k+1st position, and the current state is changed to q'. This kind of transition will be shown as $(q,k) \xrightarrow{s} (q',k+1)$.

2. An instruction q]left(s, q') will be executed in a configuration (q, k), if k > 0 and the k–1st tape square contains the symbol s. After the execution, the tape head is moved one square to the left, the k–1st position, and the current state is changed to q'. This kind of transition will be shown as $(q,k) \xrightarrow{s} (q',k-1)$.

Processing continues in this way, until a final state is reached or the machine enters a state without any possible next state.

If M has a sequence of transitions for the input w, $(q,0) \Rightarrow (q',k)$, $q \in \mathbf{I}$, $q' \in \mathbf{F}$, $k \geq 0$ then the string w is accepted by M.

If there is always at most one next configuration for a TWA, then the TWA is deterministic. And if there are no next configurations for a non-final configuration, then it is called a dead configuration. If a TWA enters such a configuration, it halts without accepting the input string. TWAs can enter infinite loops. In such a case, the input is rejected. Infinite loops can also move the tape head to the right infinitely.

Definition 2.8. Standard Form TWA: The class of TWAs, which never move the tape head out of the boundaries of the input and always halt –if they halt– on the first blank symbol after the input, either entering a final state, or a dead state, which has no outgoing transitions.

Theorem 2.1. It is possible to convert any TWA into an equivalent one in Standard Form.

Proof: We assume, without loss of generality, that our TWA M never attempts to go beyond the initial and final blanks around the input string. There are only a few possible ways for a TWA to halt without reaching the end of the tape.

1. Any final state of the TWA can be entered without moving the tape head to the end of the input. In this case, we add a temporary state to move the tape head to the right, until it sees the first blank symbol. Then it jumps to the actual halting state. (Figure 2.1)

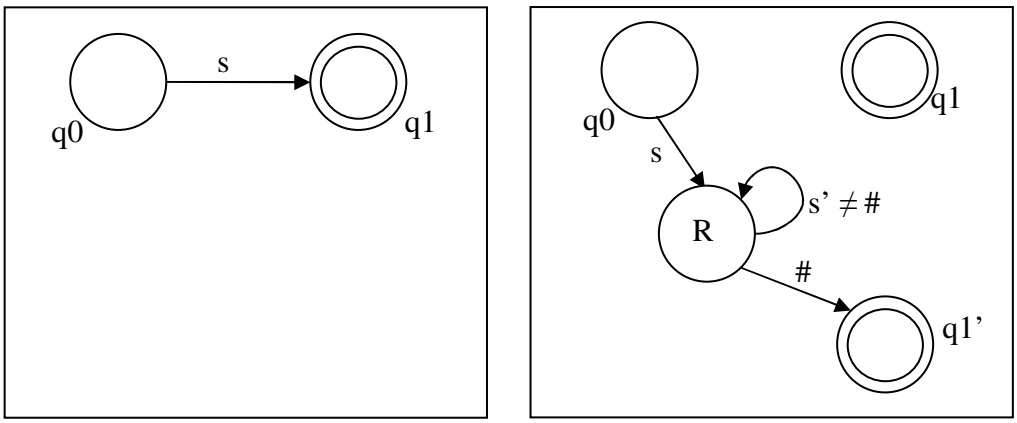

Figure 2.1: Alternation of final states (Final states are shown with double circles)

2. Some of the states may have transitions to the same direction with the blank symbol (#). When the machine follows this transition, the tape head will be moved out of the bounds of the input. These kind of transitions will be changed so that they will pass the control to a reject state, because the original transition will cause M to enter a dead configuration. (Fig 2.2) A standard form TWA does not need to have any transitions with the blank symbol, other than the initial and the final ones.

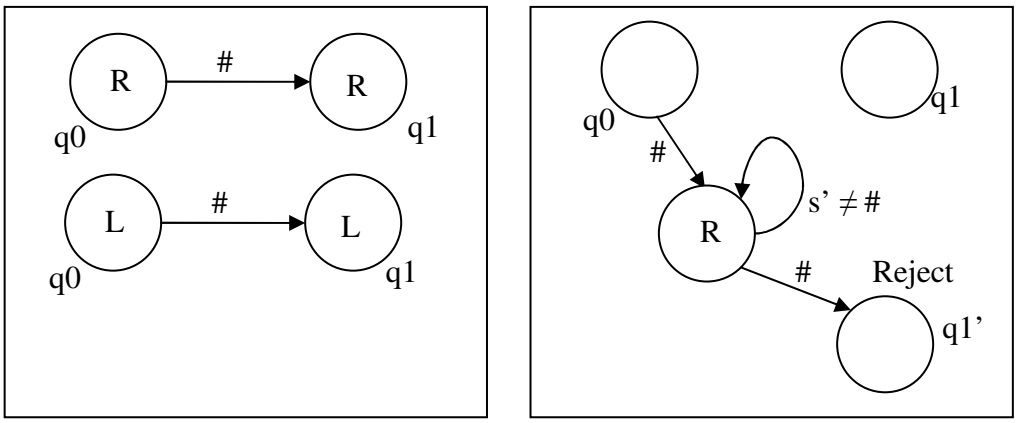

Figure 2.2: Alternation of transitions to account for some dead configurations

3. And the last possible source is the states without outgoing transitions. These states are called dead states, and since execution can not continue from those states, they are viewed as reject states. These states are also altered in a similar way. (Figure 2.3)

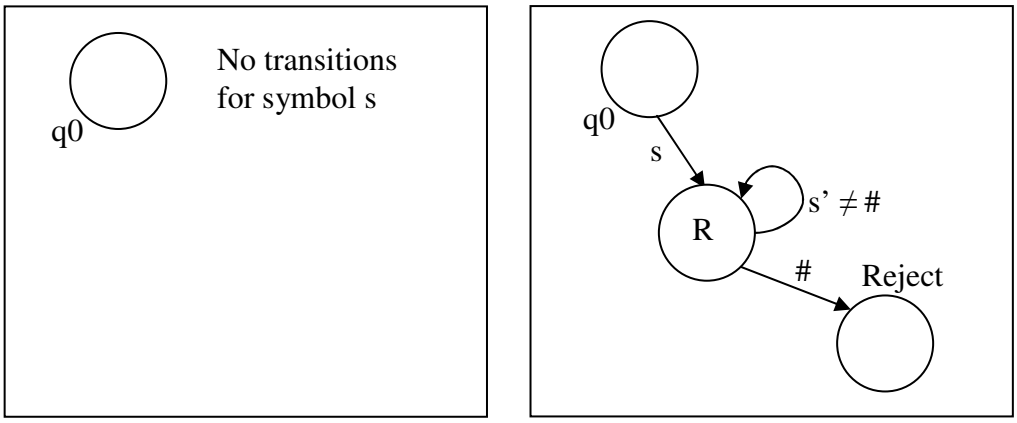

Figure 2.3: Alternation of transitions to account for some other dead configurations

As it is seen from the figures, the final states are constructed in a similar way. So, we also combine the different final states into a single one. Therefore, we can have a single reject and a single accept state, where the reject state is simply a dead state.

Corollary 2.1. A Standard Form TWA can move the tape head over the input w, as well as the tape start and end symbols. So, it can only have |Q|.(|w|+2) distinct and meaningful configurations for an input string w, where |Q| is the cardinality of state set and |w| is the length of input string.

Theorem 2.2. TWAs have exactly the same language recognition power as FSAs.

Proof Idea: Before giving the detailed proof, let us have a short overview of the actual idea behind the proof. A TWA processes input in such a way that it can go back and forth on the input, so it can re-read the input many times, and update some of its internal states accordingly. But TWAs have a limited amount of memory, so they can not extract more than a finite amount of information from the input. For any prefix t of the input, processing will scan the input for some limited number of times. If we keep track of these scans, and at a single scan, store all the information gained at different scans, then we will be able to get a FSA. Since the number of possible configurations of a Standard Form TWA is finite, this is possible with a FSA having sufficiently large number of states.

Detailed Proof: This proof is based on the proof in [3]. The proof of TWAs being equal in power to the FSAs consists of 2 parts. We should be able to convert a FSA to a TWA and a TWA to a FSA.

The first transition is straightforward. For any FSA, we can build a corresponding TWA by adding a RIGHT moving state for each FSA state, with exactly the same transitions as the FSA. While performing this conversion, we will need to take care of a few things.

Final states: In an FSA, final states can be entered any number of times, without a problem. In a TWA this is not the case. To handle this, we will perform an operation similar to the one we performed previously while converting a TWA into standard form. We will add a transition to the accept state of the TWA with the blank symbol, to the states corresponding to final states in FSA, This way, the TWA will perform a transition to the accept state with the blank symbol (#), if and only if the original machine finishes processing the input at a final state.

Initial/Final Blank Symbols: For the initial and final blank symbols, we will need to add some transitions. The one which we need to add to the initial state of the TWA will lead the execution to the state corresponding to the initial state of the FSA, The ones added to the states corresponding to the accept states of the FSA will lead the execution to the accept state of the TWA. And all the other non-final states will have a transition to the reject state with the blank symbol, so that if the processing of input finishes at a non-final state, it will pass control to the reject state.

This way, we can convert any FSA to a TWA.

The second transition is a bit tricky, since TWAs are much more compact representations for regular languages than FSAs.

Let M = (Q, S, P, I, F) be a deterministic two way accepter in standard form, where Q is the state set, S is the input alphabet, P is the program (the transition function), I is the initial state and F is the final state set. As seen in Figure 2.4, for an input w, M may scan the input more than once. At each pass, M may set some facts to its internal states about the re-processed prefix of the input. But since M is a finite state accepter, the number of different facts that M can store for any prefix α of w will be finite. This is the essential point of the proof. Since these facts are finite, we can read all of these facts in a single pass. And if we can build a machine, which reads and records all these facts at a single pass, then we have an equivalent FSA.

Let Q_L and Q_R be the left and right moving state sets of M respectively, where $Q_L \cup Q_R = Q$ and $Q_L \cap Q_R = \varnothing$. Figure 2.4 shows a snapshot of the processing, at which the tape head traverses the boundary x back and forth for a few times. Let us inspect the first traversal, which is marked as "**at state q**" and on symbol s. At this point, M just scanned symbol s, and moved the tape head to the left.

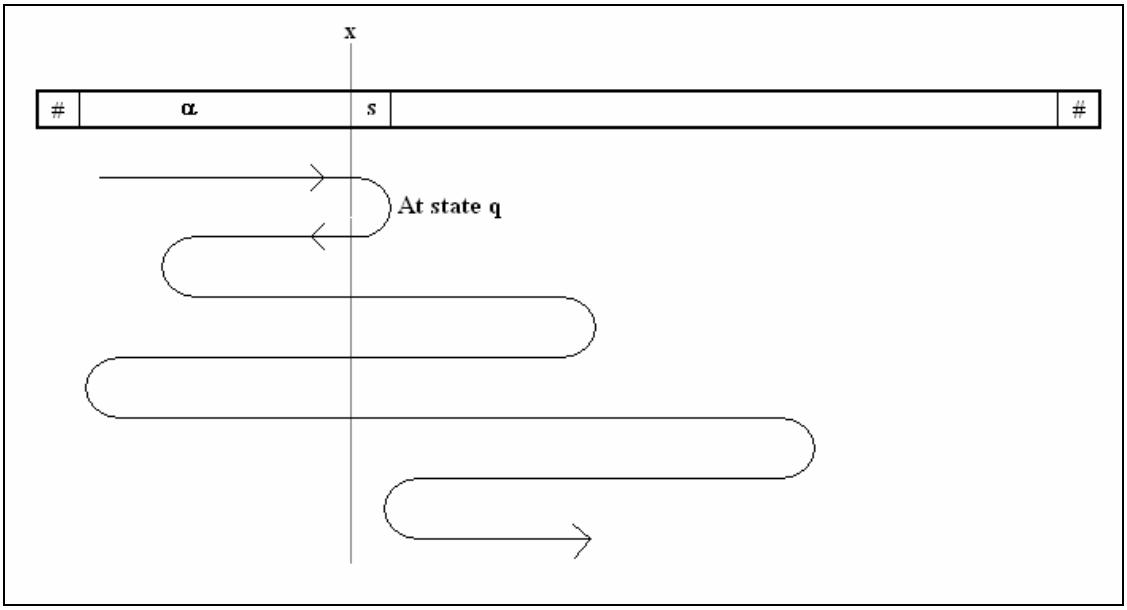

Figure 2.4: Tape scanning behavior of a TWA

Now, there are two possible future behaviors for M:

1. It will perform some processing on the prefix α and eventually re-cross boundary x at a later time.
2. It will enter an infinite loop in the prefix α, so will never cross boundary x.

There are no other possible alternatives, because M is in standard form, so it can not stop at any point other than the final # at the tape end.

The behavior of M, while processing left of boundary x, is completely dependent on state q, the input prefix α of length k and P, the program of M.

Now let us define a transformation T using these:

$$T_\alpha = Q_L \rightarrow Q_R \cup \{\Delta\} \qquad (2.1)$$

This transformation is interpreted as follows: For an input prefix α and a left moving state q, if the machine M never re-crosses the boundary x, thus enters an infinite loop, then the transformation is mapped to Δ, which means no states. Else, if M re-crosses

boundary x, T is mapped to the right moving state at which M re-crosses boundary for the first time. Thus,

$$T_\alpha(q) = \begin{cases} q' & \text{if M has a move sequence } (q, k+1) \Rightarrow (q', k), \text{ where } q' \text{ is the first} \\ & \text{right state entered by M during the sequence while at tape position k} \\ \Delta & \text{otherwise} \end{cases}$$

At first glance this transformation may seem a bit useless since there are infinitely many different prefixes α, there seem to be infinitely many different transformations. But this is not true, since the transformation is from a finite set Q_L to another finite set $Q_R \cup \{\Delta\}$, and there exist only a finite number of such transformations. If there are m left states and n right states, the number of distinct functions that can be defined from Q_L to $Q_R \cup \{\Delta\}$ is $(n+1)^m$. So, there can be no more than $(n+1)^m$ transformations for M. Let ℑ be the set of these transformations:

$$\Im = \{T \mid T: Q_L \rightarrow Q_R \cup \{\Delta\}\} \tag{2.2}$$

We also define an equivalence relation for each string in S*:

$$\omega \sim \varphi \qquad \text{if and only if } T_\omega = T_\varphi \tag{2.3}$$

Let [α] denote the equivalence class of the string α. And also let the notation $T_{[\alpha]}$ represent the transformation in ℑ that applies for each string in the equivalence class [α]. At this point we can make an interpretation about T: $T_{[\alpha]}$ represents the total knowledge of M about the prefix α.

To build FSA M' which is the equivalent of M, we will provide M' with sufficient number of states, which will keep track of the information gathered from the transitions. For each symbol s presented to M' as an input, M' makes a transition from the state representing $T_{[\alpha]}$ to the state representing $T_{[\alpha s]}$, where α represents the symbols presented to M' before s.

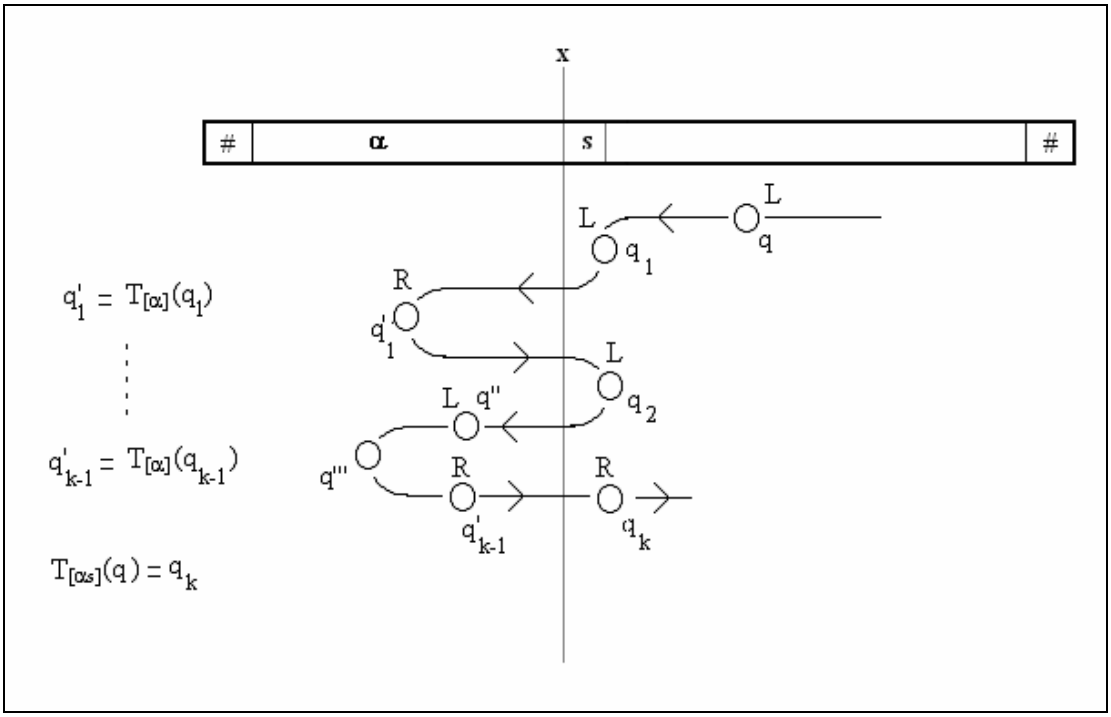

Figure 2.5: Relation between $T_{[\alpha s]}$ and $T_{[\alpha]}$.

In Figure 2.5, the relation between $T_{[\alpha s]}$ and $T_{[\alpha]}$ is shown. Suppose that at a left state q_1, a transition is made and the boundary x is crossed. Some steps later the process comes to state q'_1, which is a right state. And when it scans symbol s, it goes to state q_2, but since it is again a left state, so it re-crosses boundary x, and goes to state q''. The process goes on in a similar fashion until one of the following conditions holds:

1. Some state is repeated in the sequence $q_1..q_k$, in this case it means an infinite loop.
2. $T_{[\alpha]}(q_k) = \Delta$; which means M crosses the boundary to the left, and never crosses right of it.
3. $q_k \in Q_R$.

For the first two cases, $T_{[\alpha s]}(q) = \Delta$. And for the third case, $T_{[\alpha s]}(q) = q_k$.

We can summarize $T_{[\alpha s]}(q)$ as follows, where $q \xrightarrow{s} q'$ denotes a transition from q to q' with symbol s:

$$T_{[\sigma s]}(q) = \begin{cases} q_k & \begin{cases} \text{if M has transitions} \\ q \xrightarrow{s} q_1 \\ q_1' \xrightarrow{s} q_2 \\ \ldots \\ q_{k-1}' \xrightarrow{s} q_k \\ \text{such that} \\ \left.\begin{array}{l} q_1,\ldots,q_k \text{ are distinct,} \\ q_i \in Q_L \text{ and } q_i' \in Q_R, \\ T_{[\alpha]}(q_i) = q_i', \end{array}\right\} i = 1,\ldots,k-1 \\ q_k \in Q_R \end{cases} \\ \Delta & \text{otherwise} \end{cases}$$

The empty string transformation $T_{[\lambda]}$ maps all the left states of M to the initial right state of M on scanning of the initial #. Standard form machines never move the tape head to the left of the first blank symbol on the tape.

$T_{[\lambda]}(q) = q'$, where M has a transition from state q to q' with symbol #, q in Q_L and q' in Q_R. Since M is deterministic and in standard form, there will be exactly one such transition for each state in Q_L.

After this point, the construction of M' is straightforward. We will identify the states of M' as pairs, one is the corresponding state of M, and the other is the transformation belonging to a specific input. Hence, after M' processes a prefix α of the input string of length k, the state will be $(q, T_{[\alpha]})$, where $q \in Q_R$, and $T_{[\alpha]} \in \Im$. Here q is the right moving state, at which M scans the $k+1^{st}$ symbol for the first time, and $T_{[\alpha]}$ is the transformation which belongs to the boundary between the k^{th} and $k+1^{st}$ squares. And at the next step, according to the $k+1^{st}$ symbol on the tape, we will choose the next state. Even if M moves the tape head back to previous tape positions, and visits other states, at some point it will reach the $k+1^{st}$ state via a right state. We already identified that state as our transition. So, the next state will be the pair consisting of that state and the transformation: $(q', T_{[\alpha s]})$, where q' is the first right moving state moving the tape head to position k+1 containing the symbol s. This way, we will model all going back and returning operations via single state change. This will allow us to process all the inputs in a

single pass, but it will require many more states in some cases. And we will have another extra state, which is q_Δ, a trap state for the non accepting and infinite loop configurations in M.

Here is a detailed construction table of M' from M:

Table 2.1: Detailed construction table [3]

TWA	**FSA**
M = (Q, S, P, q_i, F) $Q = Q_L \cup Q_R$	M' = (Q', S, P', q'_i, F'), $Q' = (Q_R \times \Im) \cup \{q_\Delta\}$ $\Im = \{ T : Q_L \rightarrow Q_R \cup \{\Delta\}\}$ $q'_i = (q_1, T_{[\lambda]})$
Transitions:	
$q \xrightarrow{s} q_0, q_0 \in Q_R$	$(q, T_{[\alpha]}) \xrightarrow{s} (q_0, T_{[\alpha s]})$
$q \xrightarrow{s} q_1$ $q'_1 \xrightarrow{s} q_2 = T_{[\alpha]}(q_1)$ ... $q'_{k-1} \xrightarrow{s} q_k = T_{[\alpha]}(q_{k-1})$ such that $\left.\begin{array}{l} q_1,...,q_k \text{ are distinct,} \\ q_i \in Q_L \text{ and } q'_i \in Q_R, \\ T_{[\alpha]}(q_i) = q'_i, \end{array}\right\} i = 1,...,k-1$ $q_k \in Q_R$	$(q, T_{[\alpha]}) \xrightarrow{s} (q_k, T_{[\alpha s]})$
$q \xrightarrow{s} q_1$ *and if there is no sequence q_1 to q_k as above* $q'_1 \xrightarrow{\#} q_2 = T_{[\alpha]}(q_1)$ ... $q'_{k-1} \xrightarrow{\#} q_k = T_{[\alpha]}(q_{k-1})$ such that $\left.\begin{array}{l} q_1,...,q_k \text{ are distinct,} \\ q_i \in Q_L \text{ and } q'_i \in Q_R, \\ T_{[\alpha]}(q_i) = q'_i, \end{array}\right\} i = 1,...,k-1$ $q_k \in F$	$(q, T_{[\alpha]}) \xrightarrow{s} q_\Delta$ $q_\Delta \xrightarrow{s} q_\Delta$, *for each* $s \in S$ $(q_0, T_{[\alpha]}) \in F'$

Now let us examine the contents of Table 2.1 in detail: If M scans an input symbol s for the first time at a right-moving state q, then execution flow continues from state q_0. If q_0 is also a right moving state, then the corresponding transformation in M' is updated accordingly. $T_{[\alpha]}$ is the transformation up to that point. And at that point, since we again moved to a right-moving state, the transformation function will be updated and it will be $T_{[\alpha s]}$. But if the next state is not a right moving state, then to find the state q_k, which is the first right-moving state in the sequence of states which visits the next tape square after the prefix α, we use the previous transformation $T_{[\alpha]}$ and q_k.

A state (q_0, $T_{[\alpha]}$) is a final state in M' if and only if q_0 is a right-moving state, and there is a transition with the symbol # from q_0 to q_1, and either q_1 is a final state or is the start of a sequence of transitions leading to a final state.

Using this technique, one can build an equivalent FSA for any given TWA. This proves that FSAs are equivalent in terms of computation power to TWAs.

Now, let us examine a sample machine conversion described in [3]:

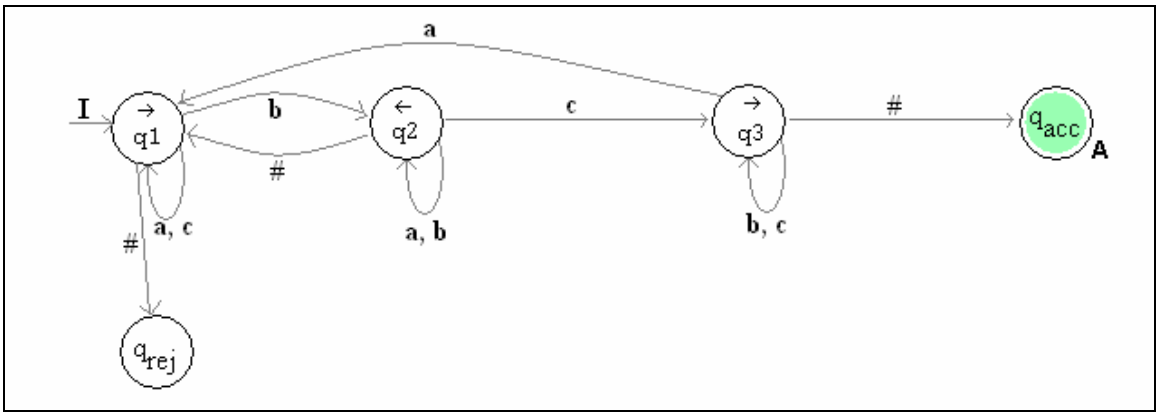

Figure 2.6: A sample TWA to be converted

The sample TWA is already in standard form, so the accept and reject states are already added. Execution starts from the state q1, which is also marked as I. The TWA has one left and two right states:

$$Q_L = \{q_2\} \qquad Q_R = \{ q_1, q_3 \} \tag{2.4}$$

So, each transformation $T_{[\alpha]}$ will be of the form T : $\{q_2\} \rightarrow \{ q_1, q_3, \Delta\}$.

We can enumerate the possible transformations as follows:

$$\Im = \{ (q_2, q_1), (q_2, q_3), (q_2, \Delta) \} \tag{2.5}$$

Let us name these as

$$T_{21} = \{ (q_2, q_1) \}, \qquad T_{23} = \{ (q_2, q_3) \}, \qquad T_{2\Delta} = \{ (q_2, \Delta) \}. \tag{2.6}$$

A # symbol will scanned by a left move only at state q_2 in the instruction q_2]left(#, q1). Hence, $T_{[\lambda]} = T_{21}$.

Table 2.2: $T_{[\alpha s]}$ in terms of $T_{[\alpha]}$ and s

$T_{[\alpha]}$ \ s	**a**	**B**	**c**
$\mathbf{T_{21}}$	T_{21}	$T_{2\Delta}$	T_{23}
$\mathbf{T_{23}}$	T_{21}	T_{23}	T_{23}
$\mathbf{T_{2\Delta}}$	$T_{2\Delta}$	$T_{2\Delta}$	T_{23}

Table 2.2 shows how each transformation $T_{[\alpha]}$ must be updated by a transition $(q, T_{[\alpha]}) \xrightarrow{s} (q', T_{[\alpha s]})$ of M'. For example, suppose that $T_{[\alpha]}=T_{21}$. To determine $T_{[\alpha a]}$, let the symbol *a* be on tape square k, and let M be in state q_2, with the tape head on position k+1. The behaviour will be:

$$(q_2, k+1) \xrightarrow{a} (q_2, k) \overset{T_{21}}{\Rightarrow} (q_1, k-1) \xrightarrow{a} (q_1, k) \tag{2.7}$$

Since state q_1 is a right moving state, we conclude that $T_{[\alpha a]}(q_2) = q_1$, and therefore $T_{[\alpha a]}=T_{21}$. One can see that $T_{[\alpha c]}=T_{23}$ in a similar way. Now, let us determine $T_{[\alpha b]}$. To do this, let the square k contain symbol b, and let M be in state q_2 with tape head on position k+1.

$$(q_2, k+1) \xrightarrow{b} (q_2, k) \overset{T_{21}}{\Rightarrow} (q_1, k-1) \xrightarrow{b} (q_2, k) \overset{T_{21}}{\Rightarrow} (q_1, k-1)... \tag{2.8}$$

We see that M is in a loop; thus $T_{[\alpha b]} = T_{2\Delta}$.

The second and third rows in Table 2.2 are determined in a similar fashion. Now let us see how each state q must be updated by a transition $(q, T_{[\alpha]}) \xrightarrow{s} (q', T_{[\alpha s]})$ of M'. Table 2.3 shows the details:

Table 2.3: q' in terms of $T_{[\alpha]}$, q and s

q \ s	a	b	c
q_1	q_1	*	q_1
q_3	q_1	q_3	q_3

*: if $q = q_1$ and s = b then q' in terms of $T_{[\alpha]}$ is as follows:

$T_{[\alpha]}$	q'
T_{21}	Q_Δ
T_{23}	q_3
$T_{2\Delta}$	Q_Δ

If q is a right moving state, and M is positioned at symbol s in square k, then q' is the state in which M moves right to scan square k+1 for the first time. In our example, q' depends on $T_{[\alpha]}$ only for state q_1 and symbol b, because **q1]right(b, q2)** is the only right instruction that puts M in a left moving state. If b is written in square k+1, and $T_{[\alpha]} = T_{21}$ applies to the boundary between squares k and k+1, we have

$$(q_1, k) \xrightarrow{b} (q_2, k+1) \overset{T_{21}}{\Rightarrow} (q_1, k) \xrightarrow{b} (q_2, k+1) \tag{2.9}$$

Since M is in a loop, the new state of M' is q_Δ. The cases in which $T_{[\alpha]}$ is T_{23} or $T_{2\Delta}$ are handled similarly.

After the conversion, the newly built FSA M' is shown in Figure 2.7.

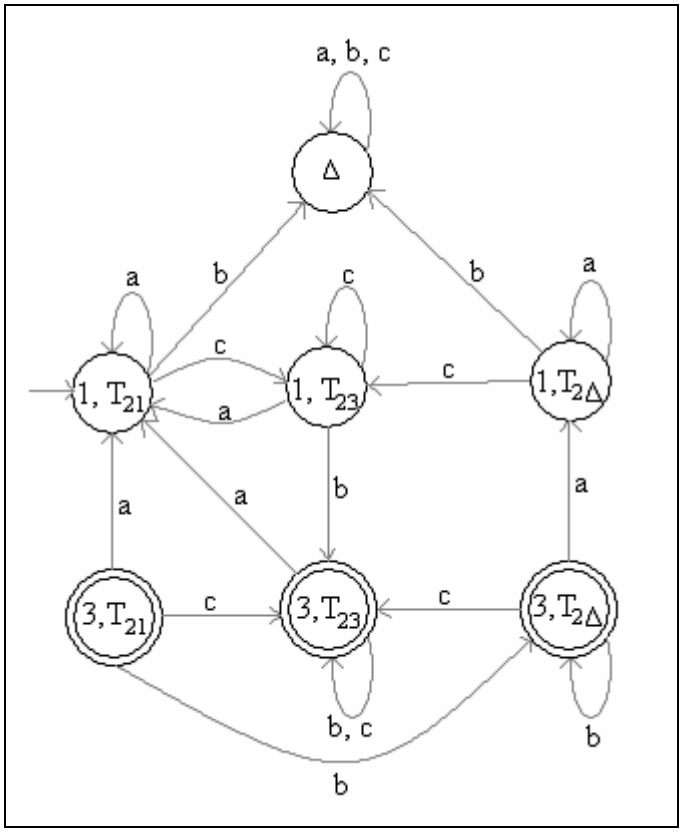

Figure 2.7: Converted FSA (To make the figure more readable, the q symbol is omitted in state names.)

All the accepters mentioned above have the same language recognition power. The following languages are not regular, i.e. they cannot be recognized by the above machines.

- The matching language $\{0^n1^n \mid n > 0\}$ is not a regular language. It is a type-2 (Context Free) language according to the Chomsky hierarchy. And it cannot be recognized with finite accepters. We require at least a Pushdown Acceptor to recognize it. [3]

- The language $\{0^n1^n2^n \mid n > 0\}$ is a type 1 (Context Sensitive) language. This class of languages cannot even be recognized by a PDA. To recognize this class of languages, we require a linear bounded Turing machine. [3]

2.2. Quantum Case

For all kinds of classical automata, one can define a corresponding quantum automaton. The main difference is the type of transitions we have in the quantum case. A Quantum Automaton is analogous to the classical counterpart, but with the property of being able to make quantum transitions between its states. Also there are some other differences related to the nature of quantum physics, forcing some restrictions on construction. For example, the transition matrices are required to be unitary. In the classical case, changing the movement capability of the tape head does not affect the language recognition power of the finite automata. But in the quantum version, this is not the case. If the construction only allows one way movement then the class of finite memory machines you can build will recognize a proper subset of regular languages. [1] If we are allowed to use two way movements, then we can build much stronger machines. In the classical version, to recognize a non-regular language of type 2 or higher, we always need an infinite rewritable storage capacity. So, the standard examples for machines that can recognize a type 2 language are PDAs. (Push Down Automata) These machines use a stack as their internal memory, and the language class they recognize corresponds exactly to the type 2 languages. But in the quantum case, we have an interesting situation. Although the quantum counterpart of a TWA does not have memory other than what is required to store the input, and it can not overwrite the input, it can recognize some of the non-regular languages. This is because of the quantum transitions in 2QFAs. This is not the only difference. The classical versions are deterministic. That means, a classical machine accepts a string or rejects a string, no matter how many times you run the machine with the same input. But Quantum Automata are not deterministic. They accept according to some probabilities. To get a reliable result, you should run the machine multiple times.

Definition 2.9. A Two-way Quantum Finite Automaton 2QFA is a six-tuple $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$, where,

Q is a finite set of states,

Σ is a finite input alphabet,

δ is the transition function,

$q_0 \in Q$ is the initial state,

Q_{acc} is the set of accepting states,

Q_{rej} is the set of rejecting states

The accepting and rejecting states are halting states and the remaining states are non-halting states. $Q_{acc} \subseteq Q$ and $Q_{rej} \subseteq Q$. Each state has an associated direction with it, either left, right or not-moving. The tape head is moved according to the direction of the entered state during execution. The tape is assumed to be circular. That means, if tape head moves beyond the tape end symbol, it will continue its motion from the tape start symbol.

The transition function $\boldsymbol{\delta}$ is a mapping,

$$\mathbf{Q} \times \sum \cup \{\#, \$\} \times \mathbf{Q} \times \{-1, 0, +1\} \rightarrow \mathbf{C} \qquad (2.10)$$

where **C** is the set of complex numbers, # and $ are the tape start and end symbols respectively.

It can be shown as $\boldsymbol{\delta(q_i, s, q_f, d) = c}$ and read as, the amplitude of transition from state $\mathbf{q_i}$ to $\mathbf{q_f}$ while moving tape head to direction **d**, with the tape symbol **s** is **c**. If the state entered during execution is a right moving state, then **d** is taken to be +1, if it is a left moving state, then **d** is taken to be -1 and if it is a not moving state, then **d** is taken to be 0. There is a little difference between our definition of the classical TWAs and 2QFAs about the tape. In classical TWAs, the statement to be executed is dependent on the symbol in the target tape cell. In 2QFAs, it is dependent on the current tape cell. The transition function of a 2QFA can be expressed as a set of matrices of complex numbers. Each matrix will have n rows and columns, where n is the number of states in 2QFA. Each number $\alpha_{i,j}$ of the transition matrix M_s will correspond to the transition amplitude from state q_i to q_j, with the symbol s.

The accepting states and rejecting states of a 2QFA are analogous to the classical counterparts. If, when started on an input, the 2QFA reaches an accepting state, it halts and accepts that string, if it reaches a rejecting state, then it halts and rejects that string. The difference is that, since 2QFAs are not deterministic, the 2QFA may not be in a single state, but it can be in many states with different probabilities, depending on the computation history so far.

Definition 2.10. The conjugate of a complex number **c = a + bi** is **a – bi** and shown as c*.

Definition 2.11. A complex square matrix M is called unitary, if $M.M^{CT} = I$ and $M^{CT}.M = I$, where I is the identity matrix of the same dimension with M, and M^{CT} is the conjugate transpose of M.

Definition 2.12. If a 2QFA is well formed, then all of the transition matrices must be unitary. The reverse is also true. If all of the transition matrices are unitary then the 2QFA is well formed.

Definition 2.13. A configuration of a 2QFA is analogous to the configuration of a TWA. Since the tape contents can not be changed, there can be at most |Q|.|w|, configurations, where |Q| is cardinality of the state set and |w| is the length of the input, including the start and end symbols.

Definition 2.14. For any 2QFA, one can define an n dimensional Hilbert Space H_n, where n is the number of different configurations. Thus, each norm 1 basis vector of the H_n corresponds to a distinct configuration of the 2QFA.

Definition 2.15. A superposition S of a 2QFA, is a vector in the related Hilbert Space H_n. The most widely used notation for superpositions is Dirac notation. In Dirac notation, $|c\rangle$ denotes a vector, which has all elements 0, except the c^{th} one, which is exactly 1, where $c \in Z$ and $1 \le c \le n$. The set of the vectors $B = \{ |c\rangle \mid c \in Z, 1 \le c \le n \}$ is a basis for H_n. Any vector in H_n can be expressed as linear combinations of the vectors in B. For a superposition $|\psi\rangle = \sum_{c=1,n} \alpha_c |c\rangle$, α_c is the amplitude associated with the configuration c in $|\psi\rangle$. When we want to make the two components of configuration c explicit, we write $|q, k\rangle$ instead of $|c\rangle$, meaning that the 2QFA is currently at state $|q\rangle$ and the tape head is on position k. In this case, k ranges between 0 and |w|, where |w| is the length of the input, including the start and end tape symbols.

Definition 2.16. A subspace of a Hilbert Space H_n is another Hilbert Space H_m, where $m \leq n$. It uses some of the basis vectors with H_n. H_n can have a maximum of n mutually orthogonal subspaces.

Definition 2.17. An observation is the process of determining the quantum state of a system. An observable is a specific observation for a specific computational model. Observables are used as an abstraction layer between the physical world and mathematical model used by us. This abstraction is required, because underlying physical construction of a quantum device will not produce a numerical value, such as the quantum systems utilizing the polarization of photons. Thus, an observation is actually the process of converting the quantum device state into a numerical value.

To define mathematically, an observable O for a 2QFA, is the set of subspaces $\{E_1 \ldots E_k\}$ and a mapping $\{E_1 \ldots E_k\} \rightarrow R$, where n is the number of states and H_n is the related Hilbert Space, and H_n is composed of mutually orthogonal subspaces E_1 to E_k.
$H_n = E_1 \oplus \ldots. \oplus E_k$, where $\oplus$ is the orthogonal sum symbol.

For each different $1 \leq j \leq n$, there is a corresponding different outcome determined by the mapping. A measurement of a 2QFA is projecting the superposition vector over the orthogonal subspaces, and determining the outcome based on this projection. The result will depend on the superposition. After the projection, a random output is observed, where the probability of each possible outcome is determined by the magnitude of the projection result over the related subspace. After the observation is done, the non-observed projections are destroyed and the observed projection is normalized. Therefore, as a side effect, the observation will modify the machine configuration.

If $|\psi_j\rangle$ is the projection of $|\psi\rangle$ onto E_j, and $|\psi\rangle = |\psi_1\rangle + \ldots. + |\psi_k\rangle$, then an observation will have following results:

- Any of the outcomes has a certain probability of being seen, which is equal to the square of the modulus of the related configuration's amplitude in the measured superposition, which is $\left\| |\psi_j\rangle \right\|^2$.

- After the observation made, as a side effect, the machine will "collapse" onto the superposition $\frac{1}{\left\| \left| \psi_j \right\rangle \right\|} \left| \psi_j \right\rangle$. That means any other configurations will be removed from the superposition.

A possible choice is setting $E_j = |c_j\rangle$. So, any observation outcome will only contain a single configuration. An observation with this outcome set will give us an exact configuration, and machine will collapse onto the configuration observed $\left| c_j \right\rangle$. While this kind of an observable is good for learning the exact configuration, in our work, a different kind of observable is used. This kind of observations are not used to determine the exact configuration of the machine, but to determine whether the machine is in a reject state, accept state or non-halting state. So, we will have three different observation results, E_{acc}, E_{rej}, E_{non}, which correspond to accepting, rejecting and non-halting configurations respectively. E_{acc} is a subspace whose base vectors are the vectors corresponding to configuration vectors of accept states. Similarly, E_{rej}'s base vectors are the vectors, corresponding to configuration vectors of reject states. And the remaining subspace in H_n, which corresponds to non-halting states are the base vectors of E_{non}. For a superposition $|\psi\rangle$, an observation result j is simply the projection of $|\psi\rangle$ over the subspace corresponding to E_j.

Definition 2.18. If there exists a fixed ε such that a 2QFA Q accepts any string $x \in$ L with a probability $\frac{1}{2} + \varepsilon, \varepsilon > 0$, and rejects any string $x \notin$ L with a probability $\frac{1}{2} + \varepsilon, \varepsilon > 0$ then Q is said to recognize L with bounded error.

A simple 2QFA: To be able to analyze the exact behavior of the 2QFA, we need to inspect them. So from now on, we will use the simulator we build to examine the 2QFAs that we will study. The diagrams of the 2QFAs can be interpreted in a similar manner to a classical 2FA. States are shown as circles, with names written in them. Arrows stand for the transitions, and the transition amplitudes are listed in the source state for each symbol. Direction of the arrows also shows the direction of the transitions. Hence, if there is an

arrow from state q to q' with an associated amplitude α, it means there is a transition from q to q' with amplitude α. Accept states are shown in green and marked with an A on the lower right, reject states are shown in red and marked with an R on the lower right. The current state is shown in blue and marked with a C on the lower right. (For detailed information, please refer to the section about our simulator.)

Figure 2.8 shows a simple 2QFA for the language 0*1*.

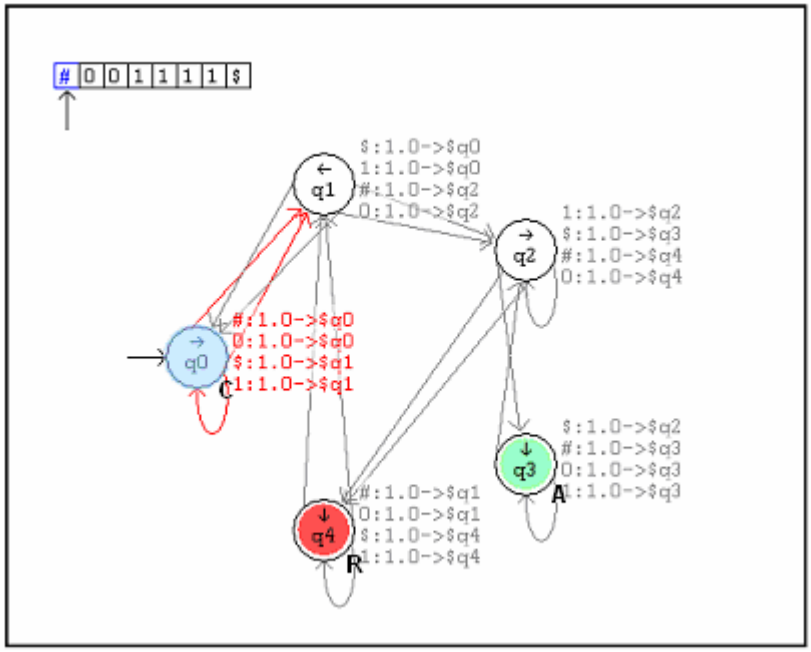

Figure 2.8: A simple 2QFA construction for the language 0*1*

Transition Matrices:

Table 2.4: Transition matrices of the 2QFA in Figure 2.8

Transition Matrix for Symbol #

#	q0	q1	q2	q3	q4
q0	1.0	0.0	0.0	0.0	0.0
q1	0.0	0.0	1.0	0.0	0.0
q2	0.0	0.0	0.0	0.0	1.0
q3	0.0	0.0	0.0	1.0	0.0
q4	0.0	1.0	0.0	0.0	0.0

Transition Matrix for Symbol **0**

0	q0	q1	q2	q3	q4
q0	1.0	0.0	0.0	0.0	0.0
q1	0.0	0.0	1.0	0.0	0.0
q2	0.0	0.0	0.0	0.0	1.0
q3	0.0	0.0	0.0	1.0	0.0
q4	0.0	1.0	0.0	0.0	0.0

Transition Matrix for Symbol **1**

1	q0	q1	q2	q3	q4
q0	0.0	1.0	0.0	0.0	0.0
q1	1.0	0.0	0.0	0.0	0.0
q2	0.0	0.0	1.0	0.0	0.0
q3	0.0	0.0	0.0	1.0	0.0
q4	0.0	0.0	0.0	0.0	1.0

Transition Matrix for Symbol **$**

$	q0	q1	q2	q3	q4
q0	0.0	1.0	0.0	0.0	0.0
q1	1.0	0.0	0.0	0.0	0.0
q2	0.0	0.0	0.0	1.0	0.0
q3	0.0	0.0	1.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	1.0

Transition values are shown both in the diagram and in the matrices. Each matrix shows the transition amplitudes associated with a specific symbol. The matrices are not symmetric. Each row contains the outgoing amplitudes for the related state, and columns contain the incoming amplitudes.

Now let us examine the execution of this 2QFA for some simple inputs. For execution trace of input 011, we will use the matrices and vectors since number of states and symbols are relatively small, and tracing is possible.

Initially, our 2QFA is in configuration $|\psi_0\rangle = |q_0, 0\rangle$. For convenience, we will present the simulation step number as a sub index.

q0	1
q1	0
q2	0
q3	0
q4	0

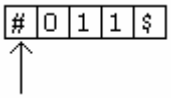

Figure 2.9: Initial state vector and the tape, corresponding to superposition $|q_0, 0>$

Step 1: To find the next configuration, we need to the multiply the state vector and the transition matrix related to the symbol under the tape head. Initially, the tape head is over the tape start symbol, at position 0.

$$|\psi_1\rangle = M_{\#}.|\psi_0\rangle = |q_0, 1\rangle \tag{2.11}$$

$|\psi_1\rangle$ is the resulting state configuration, which turns out to be in the same state as the initial one. But since q0 was a right state, the tape head is moved to position 1.
At this point we perform an observation. As we discussed earlier, we have three possible outcomes for an observation. To figure out each observation result, we will perform a projection of the current superposition.

Accept: $E_{acc}. |\psi_1\rangle = 0$

Reject: $E_{rej}. |\psi_1\rangle = 0$

Non-Halting: $E_{non}. |\psi_1\rangle = |q_0, 1\rangle$

As it is seen, after the projection, the probabilities of seeing an accept or reject configuration are zero. So, the machine is not halted yet. And we do not have any changes in the current superposition.

q0	1
q1	0
q2	0
q3	0
q4	0

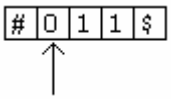

Figure 2.10: State vector and the tape after the first step ($|q_0, 1\rangle$)

Step 2: We again perform the same process. But this time, we use the transition matrix for symbol 0, since the symbol under the tape head is 0.

$$|\psi_2\rangle = M_0.|\psi_1\rangle = |q_0, 2\rangle \tag{2.12}$$

As the result, the state vector is still the same. But the tape head is moved to position 2. After the state transitions, we perform another observation.

Accept: $E_{acc}. |\psi_2\rangle = 0$

Reject: $E_{rej}. |\psi_2\rangle = 0$

Non-Halting: $E_{non}. |\psi_2\rangle = |q_0, 2\rangle$

As it is seen, observation had no effect at this step.

q0	1
q1	0
q2	0
q3	0
q4	0

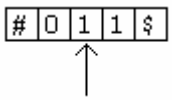

Figure 2.11: State vector and the tape after the 2nd step ($|q_0, 2\rangle$)

Step 3: We again perform the same process. But this time, we use the transition matrix for symbol 1, since the symbol under the tape head is 1.

$$|\psi_3\rangle = M_1.|\psi_2\rangle = |q_1, 1\rangle \quad (2.13)$$

As the result, the state will be changed. Process will continue from state q1. And since q1 is a left moving state, tape head will be moved back to position 1. After the state transitions, we perform another observation.

Accept: $E_{acc}.\ |\psi_3\rangle = 0$

Reject: $E_{rej}.\ |\psi_3\rangle = 0$

Non-Halting: $E_{non}.\ |\psi_3\rangle = |q_1, 1\rangle$

As it is seen, observation had no effect at this step.

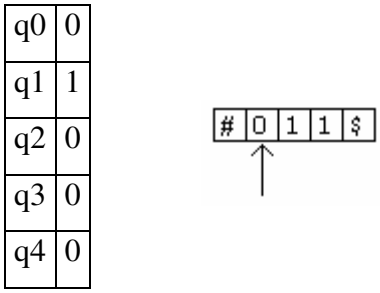

Figure 2.12: State vector and the tape after the 3rd step

Step 4: We again perform the same process. We will use the transition matrix for symbol 0.

$$|\psi_4\rangle = M_0.|\psi_3\rangle = |q_2, 2\rangle \quad (2.14)$$

As the result, the state vector will be changed. The process will continue from state q2. And since q2 is a right moving state, the tape head will be moved to position 2 again. After the state transitions, we perform an observation.

Accept: $E_{acc}.\ |\psi_4\rangle = 0$

Reject: $E_{rej}.\ |\psi_4\rangle = 0$

Non-Halting: $E_{non}.\ |\psi_4\rangle = |q_2, 2\rangle$

As it is seen, the observation had no effect at this step.

q0	0
q1	0
q2	1
q3	0
q4	0

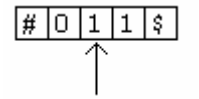

Figure 2.13: State vector and the tape after the 4th step

Step 5: We again perform the same process. We will use the transition matrix for symbol 1.

$$|\psi_5\rangle = M_1.|\psi_4\rangle = |q_2, 3\rangle \quad (2.15)$$

The state vector will not be changed. The process will continue from state q2. The tape head will be moved to position 3. After the state transitions, we perform an observation.

Accept: $E_{acc}.\ |\psi_5\rangle = 0$

Reject: $E_{rej}.\ |\psi_5\rangle = 0$

Non-Halting: $E_{non}.\ |\psi_5\rangle = |q_2, 3\rangle$

As it is seen, observation had no effect at this step.

q0	0
q1	0
q2	1
q3	0
q4	0

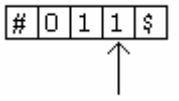

Figure 2.14: State vector and the tape after the 5th step

Step 6: We again perform the same process. We will again use the transition matrix for symbol 1.

$$|\psi_6\rangle = M_1.|\psi_5\rangle = |q_2, 4\rangle \quad (2.16)$$

The state vector will not be changed. Process will continue from state q2. The tape head will be moved to position 4. After the state transitions, we perform an observation.

Accept: $E_{acc}.\ |\psi_6\rangle = 0$

Reject: $E_{rej}.\ |\psi_6\rangle = 0$

Non-Halting: $E_{non}.\ |\psi_6\rangle = |q_2, 4\rangle$

As it is seen, observation had no effect at this step.

q0	0
q1	0
q2	1
q3	0
q4	0

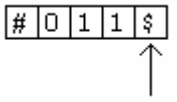

Figure 2.15: State vector and the tape after the 6th step

Step 7: We again perform the same process. We will again use the transition matrix for symbol $.

$$|\psi_7\rangle = M_1.|\psi_6\rangle = |q_3, 4\rangle \quad (2.17)$$

The state vector will be changed. Process will continue from state q3. The tape head will not be moved. After the state transitions, we perform an observation.

Accept: $E_{acc}.\ |\psi_7\rangle = |q_3, 4\rangle$

Reject: $E_{rej}.\ |\psi_7\rangle = 0$

Non-Halting: $E_{non}. |\psi_7\rangle = 0$

This time, the observation had a side effect. Projection corresponding to non halting configuration has an amplitude of 0, so we can say the machine is halted. And since only the accepting configuration's projection is non-zero, the input is ACCEPTED by the machine.

q0	0
q1	0
q2	0
q3	1
q4	0

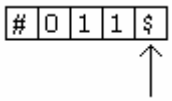

Figure 2.16: State vector and the tape after the 7th step

Theorem 2.3. 2QFAs can recognize all regular languages.[1]

Proof: To do this, we will utilize the machine building method. We will demonstrate a conversion algorithm from 1DFAs to 2QFAs. This way, one could build a 2QFA for any given regular language, since all the regular languages have a corresponding 1DFA.

A 1DFA can be specified by a 5-tuple (S, Σ, μ, s_0, F), where S is the state set, Σ is the input alphabet, μ is the transition function, s_0 is the initial state and F is the final state set. Given such a 1DFA A, we will define a 2QFA $M = (Q, \sum, \delta, q_0, Q_{acc}, Q_{rej})$, which will recognize the same language as A.

In the 2QFA, we have two additional symbols on the tape, namely # and $ (tape start and end markers). To allow M to behave correctly, we will make a small modification over A, so that it will take care of the # and $. To do this, we will extend S to S' and μ to μ'.

S' = S ∪ {s'$_0$, s_{acc}, s_{rej}}, where s'$_0$, s_{acc} and s_{rej} are not elements of S.

$$\mu'(s,\sigma) = \begin{cases} \mu(s,\sigma) & s \in S, \sigma \in \Sigma \\ s_0 & s = s_0', \sigma = \# \\ s_{acc} & s \in F, \sigma = \$ \\ s_{rej} & s \in S \setminus F, \sigma = \$ \end{cases} \quad (2.18)$$

And for other values, let μ' be undefined.

We also define the following two sets:

$$I_{s,\sigma} = \{s' \in S \mid \mu'(s', \sigma) = \mu'(s, \sigma)\} \quad (2.19)$$

$$J_{s,\sigma} = \{s' \in S \mid \mu'(s', \sigma) = s\} \quad (2.20)$$

which means, $I_{s,\sigma}$ is the set of all states which have a transition with the symbol σ to the same state as state s. And $J_{s,\sigma}$ is the set of all states which have a transition with the symbol σ to the state s.

We also define an ordering on the state set. The specific ordering is not important. Let us also have **min** and **max** define the least and maximum elements according to this ordering. And we also define the function **succ** over this ordering: **succ**(s, T) is the smallest element –if it exists– that is larger than s in T according to our ordering, where T ⊆ S'.

Now we can define M. Let Q = S' × {-1, +1}, and q_0 = (s'$_0$, +1). Q_{acc} = {(s_{acc}, +1)} and Q_{rej} = {(s_{rej}, +1)}.

For the transition function, we will not use a single big matrix, but we will use separate smaller matrices for each input symbol, and we will use V_s to denote these matrices, where s is the related symbol of the matrix.

For each s∈ S', σ∈Γ, where Γ=Σ ∪ {#, $}, and μ'(s, σ) is defined, let

$$V_\sigma \left|(s,+1)\right\rangle = \begin{cases} \left|(succ(s, I_{s,\sigma}),-1)\right\rangle & s \neq \max(I_{s,\sigma}) \\ \left|(\mu'(s,\sigma),+1)\right\rangle & s = \max(I_{s,\sigma}) \end{cases}, \tag{2.21}$$

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

$$V_\sigma \left|(s,-1)\right\rangle = \begin{cases} \left|(s,+1)\right\rangle & J_{s,\sigma} = \varnothing \\ \left|(\min(J_{s,\sigma}),-1)\right\rangle & J_{s,\sigma} \neq \varnothing \end{cases} \tag{2.22}$$

The remaining rows and columns of transition matrices could be completed arbitrarily, but not violating the unitarity condition.

And the directions of the states are defined as D((s, +1)) = +1 and D((s, -1)) = -1.

Using the matrices and D, let the transition function δ be defined in the usual way. Now we can examine the simulation of the 1DFA with the new 2QFA we have built. First let us define the processing graph G. The vertices of G form the set $S' \times Z_{n+2}$, namely the different configurations of A. And there is an edge between two vertices if and only if A can move from the corresponding configuration to the other configuration. Thus, (s_1, k) and $(s_2, k+1)$ are connected only if $\mu'(s_1, w_k) = s_2$, where w_k is the k^{th} symbol on the tape. Let us try to visualize the processing of the input by the 1DFA as a tree. This tree is a subgraph of G. In this tree, the root will be one of the newly added accept or reject states, depending on whether the input is accepted or rejected. The first level of the tree will contain the states which can possibly have a transition to the final state, depending on the input. And each of these states will have the possible previous states, which can have a transition to those states with the input symbol on the corresponding tape cell.

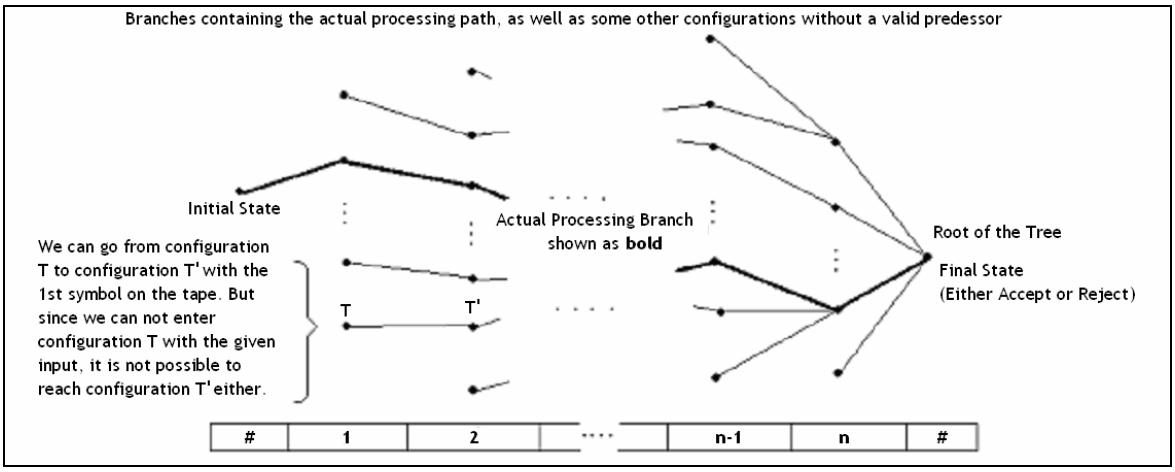

Figure 2.17: Visualization of the processing tree

For any given input w of length n, we can find a path from the level n to one of the halting states in G.

The tree is completely determined from the definition of A. And G is determined from the input w. So, for any symbol and any given configuration, one can determine the next state and possible previous states by using the machine specification. Our newly built 2QFA will simulate traversing the tree belonging to the machine A. The traversal may or may not be identical to the exact execution path of the 1DFA, but it will traverse the tree, and at the end it will arrive at the same state with the original 1DFA, either accept or reject.The 2QFA may visit more configurations than the original 1DFA to satisfy reversibility, but since the number of possible next and previous configurations are limited, the execution will be completed in O(n) steps.

A can have a total of |S'|.(n + 2) different configurations. In M, we have 2 configurations for each different configuration (s, k) of A, namely ((s, -1), k-1) and ((s, +1), k). These two configurations are pre and post states of a tree traversal. The sub-tree G_0 of the directed graph G, has configuration (s,k) as its root node. This configuration corresponds to the state s of the machine A, since our initial graph is based on that machine also. In the 2QFA, when the configuration ((s, -1), k-1) is entered, the sub-tree G_0 is just about to be traversed. And when the machine enters the configuration ((s, +1), k), then G_0 has just been traversed. Traversing these sub-trees means the 2QFA will eventually simulate every possible configuration that the 1DFA can enter for a given input. At each

step, the 2QFA traverses the possible previous configurations, according to the ordering we have used. When there are no possible states to traverse in the same level, the machine enters a configuration on the upper level. That means, we have processed one more symbol on the tape, and all the possible paths possible with that symbol.

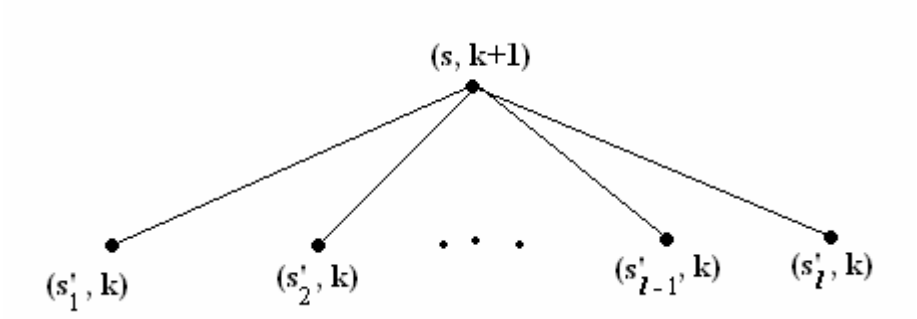

Figure 2.18: The traverse tree of the 1DFA

Figure 2.18 shows the sub-tree rooted at (s, k+1). At the time the machine enters that configuration, the tape symbol is w_k, where w is the input. The details of the processing can be inspected in a few cases:

- All possible previous configurations of configuration (s, k+1) are listed as s'_1 thru s'_l. When the 2QFA is at any configuration $((s'_c, -1), k)$, $c < l$, the next configuration is $((\mathbf{succ}(s'_c, I_{s'_c, w_k})-1), k-1) = ((s'_{c+1}, -1), k-1)$, where c is incremented according to our predefined ordering of S and k is decremented according to state transition definitions. After this process, we are in a position that, we will process the sub-tree rooted at node (s'_{c+1}, k). When $c=l$, and since is the maximum state according to our predefined ordering, the next state is (s, k+1).

- Let us assume M is in configuration ((s,-1), k). That means, the sub-tree rooted at vertex (s, k) will be traversed next. And next configuration is $((\mathbf{min}(J_{s', w_k}), -1), k-1) = ((s'_1, -1), k-1)$. And since we know once it enters a series of states s'_c, it will process until it reaches the last one.

- And when (s, k+1) has no predecessors, then it will immediately jump from state ((s, -1), k) to state ((s, +1), k+1). That means it has traversed that vertex.

So, as it is seen, the configuration (s, k+1) can only be visited after each child of (s,k+1) is visited. That means, our 2QFA will traverse thru all the vertices shown in Figure 2.17 and 2.18, until it reaches either the accepting state or the rejecting state. Since there are no cycles in the tree, the processing will not enter an infinite loop. It will eventually stop in O(|w|) steps.

So, since any regular language can be recognized by a 1DFA, we have proved that one can build a 2QFA for any given regular language.

To make things clearer, here is a sample conversion of a 1DFA to an equivalent 2QFA:

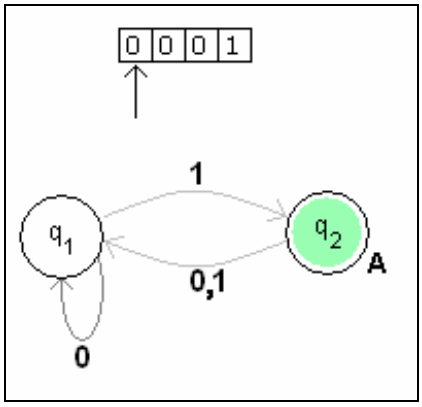

Figure 2.19: A 1DFA for the language of strings of 0s and 1s which end with an odd number of 1s. $\{1^n \mid n = 1 \pmod 2\} \cup \{\{0,1\}^*01^n \mid n = 1 \pmod 2\}$

Here is the transition function for the original 1DFA:

$$\mu(s,\sigma) = \begin{cases} q_1 & s = q1,\ \sigma = 0 \\ q_2 & s = q1,\ \sigma = 1 \\ q_1 & s = q2,\ \sigma = 0 \\ q_1 & s = q2,\ \sigma = 1 \end{cases} \tag{2.23}$$

The following is the transition function for the modified 1DFA:

$$\mu'(s,\sigma)=\begin{cases}\mu(s,\sigma) & s\in\{q1,q2\},\sigma\in\{0,1\}\\ q_1 & s=q_0,\sigma=\#\\ q_3 & s\in F,\sigma=\$\\ q_4 & s\in S\setminus F,\sigma=\$\end{cases} \tag{2.24}$$

And here are the relations I and J:

$$\begin{aligned} I_{q_1,0}&=\{q_1,q_2\} & J_{q_1,0}&=\{q_1,q_2\}\\ I_{q_1,1}&=\{q_1\} & J_{q_1,1}&=\{q_2\}\\ I_{q_2,0}&=\{q_1,q_2\} & J_{q_2,0}&=\varnothing\\ I_{q_2,1}&=\{q_2\} & J_{q_2,1}&=\{q_1\}\end{aligned} \tag{2.25}$$

Let us define the ordering on S' as follows:

$$q_0<q_1<q_2<q_3<q_4 \tag{2.26}$$

To be able to simulate the newly built 2QFA in our simulator, we will use the following mapping:

$$\begin{aligned}(q_0,+1)&\Rightarrow q_0\\ (q_1,+1)&\Rightarrow q_1\\ (q_2,+1)&\Rightarrow q_2\\ (q_3,+1)&\Rightarrow q_3\\ (q_4,+1)&\Rightarrow q_4\\ (q_0,-1)&\Rightarrow q_5\\ (q_1,-1)&\Rightarrow q_6\\ (q_2,-1)&\Rightarrow q_7\\ (q_3,-1)&\Rightarrow q_8\\ (q_4,-1)&\Rightarrow q_9\end{aligned} \tag{2.27}$$

Transition matrices of the 2QFA can be seen in Table 2.5. And the actual machine can be seen in Figure 2.20.

Table 2.5: Transition matrices for the sample 1DFA for the language $\{1^n \mid n = 1 \pmod 2\} \cup \{(0,1)^*01^n \mid n = 1 \pmod 2\}$

Transition Matrix for Symbol **#**

#	q0	Q1	q2	q3	q4	q5	q6	q7	q8	q9
q0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q2	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0
q5	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0
q6	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0
q7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0
q8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0
q9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1

Transition Matrix for Symbol **0**

0	q0	Q1	q2	q3	q4	q5	q6	q7	q8	q9
q0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0
q2	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0
q5	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0
q6	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0
q7	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0
q9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1

Transition Matrix for Symbol **1**

1	q0	Q1	q2	q3	q4	q5	q6	q7	q8	q9
q0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q2	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0
q5	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0
q6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0
q7	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0
q8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0
q9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1

Transition Matrix for Symbol **$**

$	q0	Q1	q2	q3	q4	q5	q6	q7	q8	q9
q0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0
q2	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0
q5	0.0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0
q6	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q7	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0.0
q9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1

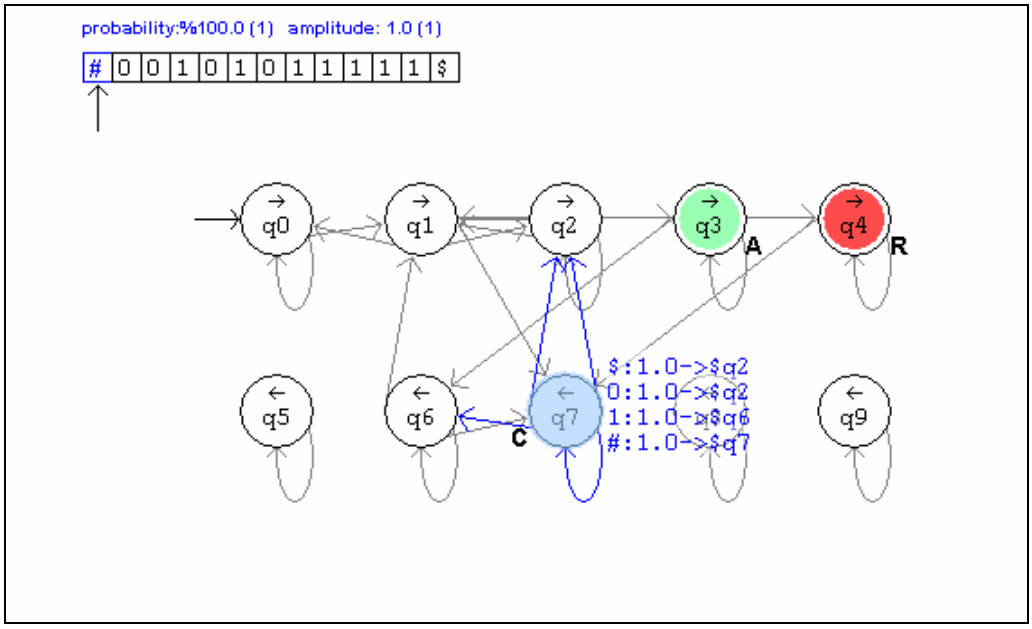

Figure 2.20: Diagram for the constructed 2QFA for the language $\{1^n \mid n = 1 \pmod 2\} \cup \{\{0,1\}^*01^n \mid n = 1 \pmod 2\}$

Theorem 2.4. 2QFAs can recognize some non-regular languages.

Proof: In previous work, it was shown that a 2QFA could recognize some languages that are not type 3. This is very surprising, since in classical FA, we do not have any known abstractions that recognize a non-regular language without using a read/write memory. The first non-regular language that was proven to be of this kind is the famous $\{0^n1^n \mid n > 0\}$ language, which is type 2 [1]. Also in the same paper it was stated that, a similar construction would allow recognizing $\{ 0^n1^n2^n \mid n > 0\}$ language, which is a non-context free language. We will now present the detailed construction of both machines.

Table 2.6: Formal definition of the original 2QFA for the language $\{0^n1^n \mid n > 0\}$ [1]

$$V_{\#}|q_0\rangle = |q_0\rangle, \qquad V_b|q_0\rangle = |q_1\rangle,$$
$$V_{\#}|q_1\rangle = |q_3\rangle, \qquad V_b|q_2\rangle = |q_2\rangle,$$
$$V_{\#}|r_{j,0}\rangle = \frac{1}{\sqrt{N}}\sum_{l=1}^{N}\exp(\frac{2\pi i}{N}jl)|s_l\rangle, 1 \le j \le N, \qquad V_b|r_{j,0}\rangle = |r_{j,N-j+1}\rangle, 1 \le j \le N,$$
$$V_a|r_{j,k}\rangle = |r_{j,k-1}\rangle, 1 \le k \le N-j+1, 1 \le j \le N,$$
$$V_a|q_0\rangle = |q_0\rangle,$$
$$V_a|q_1\rangle = |q_2\rangle,$$
$$V_a|q_2\rangle = |q_3\rangle, \qquad V_\$|q_0\rangle = |q_3\rangle,$$
$$V_a|r_{j,0}\rangle = |r_{j,j}\rangle, 1 \le j \le N, \qquad V_\$|q_2\rangle = \frac{1}{\sqrt{N}}\sum_{j=1}^{N}|r_{j,0}\rangle,$$
$$V_a|r_{j,k}\rangle = |r_{j,k-1}\rangle, 1 \le k \le j, 1 \le j \le N,$$

$$D(q_0) = +1, \qquad D(r_{j,0}) = -1, 1 \le j \le N,$$
$$D(q_1) = -1, \qquad D(r_{j,k}) = 0, 1 \le j \le N, k > 0,$$
$$D(q_2) = +1, \qquad D(s_j) = 0, 1 \le j \le N,$$
$$D(q_3) = 0,$$

$\{0^n1^n \mid n > 0\}$: This is the original language that was used [1] to prove that a 2QFA can accept non regular languages. Table 2.6 is a parametric definition of a machine that can be used to recognize this language at a desired accuracy. And the construction is shown in Figure 2.19 for the case N=3.

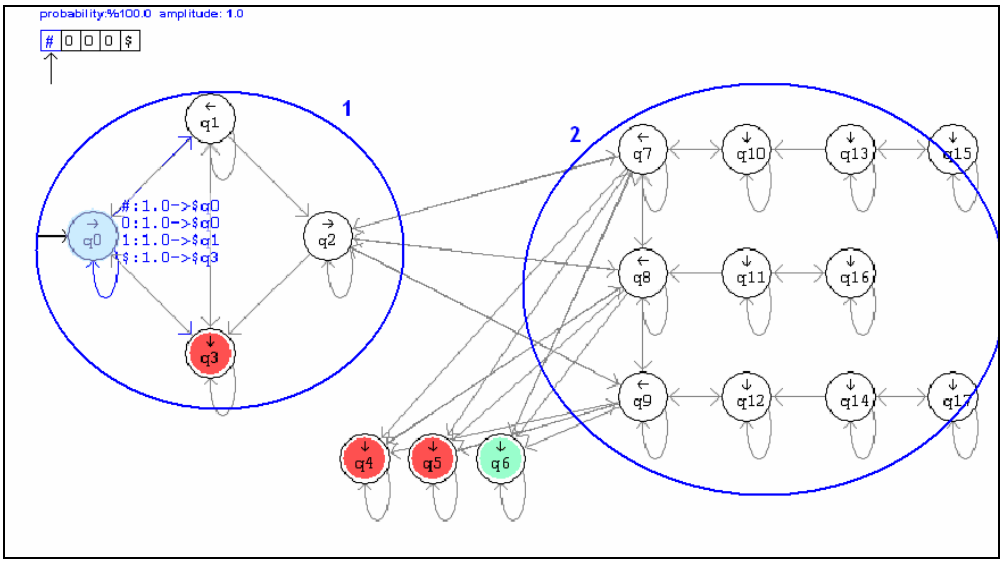

Figure 2.21: Construction of the 2QFA for the language $\{0^n1^n \mid n > 0\}$.

[For transition table, refer to Appendix A: Machine 1]

Figure 2.21 is a visualization of the construction that recognizes this language. It works in two phases. In the first phase, it validates that the input string is in 0*1* by a scan from left to right. If it is not of the form 0*1*, then it will be rejected with probability 1. Then in the second phase, it validates the equality of the number of 0's and 1's. This is done by using a special construction in the state transactions. According to this construction, the execution branches into N (in Figure 2.21 it is three) branches at initial state of this phase. (You can see these as three rows of states on the right) Each branch has a different leftward speed on the tape. The speed also differs according to the symbols. The branch i will move the tape head after i steps for symbol 1, and it will move after (N-i+1) steps for symbol 0. So, if in the initial input there are a 0s and b 1s, then it will take $(i+1)a+(N-i+2)b+1$ steps to reach from one end to the other end of the tape. If $i \neq j$ then the number of steps to reach end of the tape for any two different branches is equal, i.e.

$$(i+1)a+(N-i+2)b+1=(j+1)a+(N-j+2)b+1 \tag{2.28}$$

is valid only if $a=b$. At the beginning of the tape, each branch will again branch to the final states, (q1, q2 and q3 in this case) with equal probability. If $a=b$ then all branches will approach to the tape start and the second branching state at the same step. Then, because of the special branching, the amplitudes on reject states will cancel each other, and the amplitudes on the accept state will be added. This special branching is called Quantum Fourier Transform. If $a \neq b$, then all branches will reach the start of the tape at different steps, since the speeds of branches are different. So, each branch will separately re-branch to the final states, and will be rejected with $\frac{N-1}{N}$ probability, since only one of N final states is an accept state. So, at the end, total false accept ratio will be $\frac{1}{N}$, false reject ratio will be 0.

This construction is a parametric construction, so the builder can choose the required correctness probability. The construction in Figure 2.21 is for N=3, and it yields an error probability of 1/3. It accepts the strings which are elements of the language with a

probability of 1. And it rejects the non-members with a probability of $\frac{N-1}{N}$, which is 2/3 in above case.

Even though the machine seems complex, there is a simple yet important point in the construction. Let us overview it shortly: The 2QFA creates N threads to verify the string. And in case the string is illegal, these N threads will be out of synchronization. That would cause the string to be rejected. If the string is legal, then the threads will be in synchronization, and they will cancel each other on rejecting states, and the string will be accepted. This construction utilizes the technique Quantum Fourier Transform, and it will be examined in detail in the following section.

Detailed Execution Description: Now, we will examine the working principle of the 2QFA by using a simple example input "**01**". This way, we will be able to follow the program flow. Initially, there is only one active configuration of the 2QFA. The tape head is at position 0, and q0 is the active state.

Step 1. Current symbol is tape begin symbol (#). Transition function maps q0 to q0 with a tape content of # ($\delta(q0, \#) = q0$). And q0 is a right moving state, so it moves the tape head to the first tape square. We perform an observation, and since none of the halting states are active, i.e. none of the halting states has non-zero amplitude in the current superposition, observation yields no changes. Resulting superposition: $|\psi_1\rangle=|q_0, 1\rangle$

Step 2. Symbol: 0. $\delta(q0, 0) = q0$, so, we move tape head to position 2 while staying at state q0. We perform another observation, but no effect, since no halting states have non zero amplitudes. Resulting superposition: $|\psi_2\rangle = |q_0, 2\rangle$

Step 3. Symbol: 1; $\delta(q0, 1) = q1$. So tape head is moved back to position 1, and state is changed to q1. Resulting superposition: $|\psi_3\rangle = |q_1, 1\rangle$

Step 4. Symbol: 0; $\delta(q1, 0) = q2$. So tape head is moved to position 2 again and state is changed to q2. Resulting superposition: $|\psi_4\rangle = |q_2, 2\rangle$

Step 5. Symbol: 1; δ(q2, 1) = q2. So tape head is moved to position 3 and state is not changed. Resulting superposition: $|\psi_1\rangle = |q_2, 3\rangle$

Step 6. Symbol $; $\delta(q2, \$) = \frac{1}{\sqrt{3}}|q7\rangle + \frac{1}{\sqrt{3}}|q8\rangle + \frac{1}{\sqrt{3}}|q9\rangle$. In this step, there is not a single transition, but there are 3 transitions. From now on, we have a superposition of 3 configurations. To identify each different execution flow, we will use the term **thread**. So, in this case, we have 3 threads at this step. In the simulator, you will see 3 tabs in the execution window, each one corresponding to a different thread.

Thread 1: This thread continues execution from state q7. Its amplitude is $\frac{1}{\sqrt{3}}$ and its probability is $\frac{1}{3}$. Tape head is on position 2.

Thread 2: This thread continues execution from state q8. Its amplitude is $\frac{1}{\sqrt{3}}$ and its probability is $\frac{1}{3}$. Tape head is on position 2.

Thread 3: This thread continues execution from state q9. Its amplitude is $\frac{1}{\sqrt{3}}$ and its probability is $\frac{1}{3}$. Tape head is on position 2.

Resulting superposition: $|\psi_6\rangle = \frac{1}{\sqrt{3}}|q7,2\rangle + \frac{1}{\sqrt{3}}|q8,2\rangle + \frac{1}{\sqrt{3}}|q9,2\rangle$

Step 7. There are 3 threads, so we need to inspect them separately.

Thread 1: Symbol: 1; δ(q7, 1) = q15. So tape head is not moved and still on position 2. State is changed to q15.

Thread 2: Symbol: 1; δ(q8, 1) = q16. So tape head is not moved and still on position 2. State is changed to q16.

Thread 3: Symbol: 1; δ(q9, 1) = q12. So tape head is not moved and still on position 2. State is changed to q12.

Resulting superposition: $|\psi_7\rangle = \frac{1}{\sqrt{3}}|q15,2\rangle + \frac{1}{\sqrt{3}}|q16,2\rangle + \frac{1}{\sqrt{3}}|q12,2\rangle$

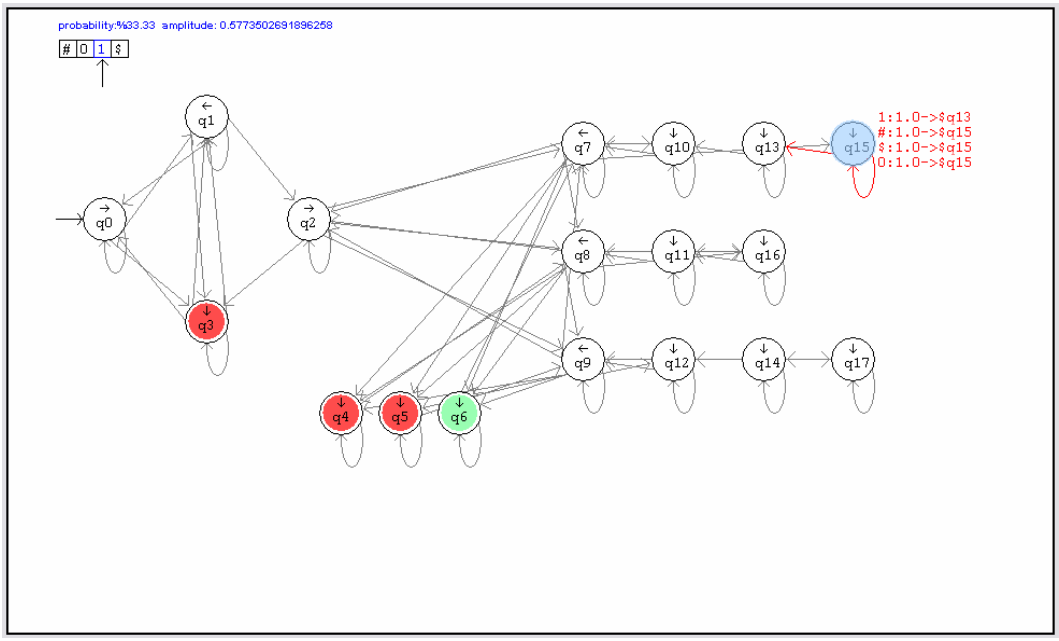

Figure 2.22: 2QFA state after step 7. Marked state is the current state for thread 1.

Step 8. There are 3 threads, so we need to inspect them separately.

Thread 1: Symbol: 1; δ(q15, 1) = q13. So tape head is not moved and still on position 2. State is changed to q13.

Thread 2: Symbol: 1; δ(q16, 1) = q11. So tape head is not moved and still on position 2. State is changed to q11.

Thread 3: Symbol: 1; δ(q12, 1) = q9. So tape head is moved to position 1 and state is changed to q9.

Resulting superposition: $|\psi_8\rangle = \frac{1}{\sqrt{3}}|q13,2\rangle + \frac{1}{\sqrt{3}}|q11,2\rangle + \frac{1}{\sqrt{3}}|q9,1\rangle$

Step 9. There are 3 threads, so we need to inspect them separately.

Thread 1: Symbol: 1; δ(q13, 1) = q10. So tape head is not moved and still on position 2. State is changed to q10.

Thread 2: Symbol: 1; δ(q11, 1) = q8. So tape head is moved to position 1 and state is changed to q8.

Thread 3: Symbol: 0; δ(q9, 0) = q17. So tape head is not moved and still on position 1. State is changed to q17.

Resulting superposition: $|\psi_9\rangle = \frac{1}{\sqrt{3}}|q10,2\rangle + \frac{1}{\sqrt{3}}|q8,1\rangle + \frac{1}{\sqrt{3}}|q17,1\rangle$

Step 10. There are 3 threads, so we need to inspect them separately.

Thread 1: Symbol: 0; δ(q10, 0) = q7. So tape head is moved to position 1 and state is changed to q7.

Thread 2: Symbol: 0; δ(q8, 0) = q16. So tape head is not moved and still on position 1. State is changed to q16.

Thread 3: Symbol: 0; δ(q17, 0) = q14. So tape head is not moved and still on position 1. State is changed to q14.

Resulting superposition: $|\psi_{10}\rangle = \frac{1}{\sqrt{3}}|q7,1\rangle + \frac{1}{\sqrt{3}}|q16,1\rangle + \frac{1}{\sqrt{3}}|q14,1\rangle$

Step 11. There are 3 threads, so we need to inspect them separately.

Thread 1: Symbol: 1; δ(q7, 1) = q10. So tape head is not moved and still on position 1. State is changed to q10.

Thread 2: Symbol: 0; δ(q16, 0) = q11. So tape head is not moved and still on position 1. State is changed to q11.

Thread 3: Symbol: 0; δ(q14, 0) = q12. So tape head is not moved and still on position 1. State is changed to q12.

Resulting superposition: $|\psi_{11}\rangle = \frac{1}{\sqrt{3}}|q10,1\rangle + \frac{1}{\sqrt{3}}|q11,1\rangle + \frac{1}{\sqrt{3}}|q12,1\rangle$

Step 12. There are 3 threads, so we need to inspect them separately.

Thread 1: Symbol: 0; δ(q10, 0) = q7. So tape head is moved to position 0. State is changed to q7.

Thread 2: Symbol: 0; δ(q11, 0) = q8. So tape head is moved to position 0. State is changed to q8.

Thread 3: Symbol: 0; δ(q12, 0) = q9. So tape head is moved to position 0. State is changed to q9.

Resulting superposition: $|\psi_{12}\rangle = \frac{1}{\sqrt{3}}|q7,0\rangle + \frac{1}{\sqrt{3}}|q8,0\rangle + \frac{1}{\sqrt{3}}|q9,0\rangle$

Step 13. At this stage, we will see a QFT (Quantum Fourier Transform) type transition. That means, three threads will proceed to same states at the same step with specially calculated amplitudes. This special calculation allows some of the threads cancel each other or merge together.

Thread 1: Symbol: #; δ(q7, #) = $(-\frac{1}{2\sqrt{3}}+\frac{i}{2})|q4\rangle+(-\frac{1}{2\sqrt{3}}-\frac{i}{2})|q5\rangle+\frac{1}{\sqrt{3}}|q6\rangle$. So tape head is not moved. It is branched to q4, q5 and q6.

Thread 2: Symbol: #; δ(q8, #) = $(-\frac{1}{2\sqrt{3}}-\frac{i}{2})|q4\rangle+(-\frac{1}{2\sqrt{3}}+\frac{i}{2})|q5\rangle+\frac{1}{\sqrt{3}}|q6\rangle$. So tape head is not moved. It is branched to q4, q5 and q6.

Thread 3: Symbol: #; δ(q9, #) = $\frac{1}{\sqrt{3}}|q4\rangle+\frac{1}{\sqrt{3}}|q5\rangle+\frac{1}{\sqrt{3}}|q6\rangle$. So tape head is not moved. It is branched to q4, q5 and q6.

Observation for step 13: We have all 3 threads to split into 3 again, to states q4, q5 and q6. If any thread reaches to reject states (q4 or q5), then that thread will halt and will be identified as rejected. If any thread reaches accept states (q6), then it will halt and will be identified as accepted.

The amplitudes of the split threads to states q4 and q5 add up to 0, thus canceling each other. So, even though it seems they reach the states q4 and q5, they disappear. But for state q6, the amplitudes do not cancel. So the processing continues to states q6 meanwhile merging all the threads into a single one. And that single one is an Accept state, so the input is accepted with 100% probability. If we inspect the superposition amplitudes, we will see the same result:

Resulting superposition:

$$\begin{aligned}|\psi_{13}\rangle = &(-\frac{1}{2\sqrt{3}}+\frac{i}{2})|q4,0\rangle + (-\frac{1}{2\sqrt{3}}-\frac{i}{2})|q5,0\rangle + \frac{1}{\sqrt{3}}|q6,0\rangle \\ &+(-\frac{1}{2\sqrt{3}}-\frac{i}{2})|q4,0\rangle + (-\frac{1}{2\sqrt{3}}+\frac{i}{2})|q5,0\rangle + \frac{1}{\sqrt{3}}|q6,0\rangle \\ &+\frac{1}{\sqrt{3}}|q4,0\rangle + \frac{1}{\sqrt{3}}|q5,0\rangle + \frac{1}{\sqrt{3}}|q6,0\rangle \\ &=0.|q4,0\rangle + 0.|q5,0\rangle + 1.|q6,0\rangle \\ &=|q6,0\rangle\end{aligned} \tag{2.23}$$

All threads are merged on state q6, and accepted with 100% probability.

If we inspect the execution of the same machine for a different input that will be rejected, for example the input string “011”, then we will see that, execution will be exactly same for each thread until the last step. But in the last step, the threads will not be in synchronization. The construction is adjusted so that, any two thread will never arrive at states r_j (q7, q8, q9 in this machine) at the same time when number of 0s and 1s are different. This way, QFT will not cancel other threads, so each thread will be accepted by probability 1/N. (where N=3 in this case). And since the threads are already split into 3, each individual thread will accept by 1/9 probability. And result will be 2/3 reject and 1/3 accept.

$\{0^n1^n2^n \mid n > 0\}$: This language was proposed to be recognized with a regular 2QFA, but its construction details were omitted in [1]. Here we present the construction. (For transition matrices, please refer to Appendix A: Machine 3) This language has the property of being a non-context free language. With classical automata, one can not recognize this language even with a PDA. [3]

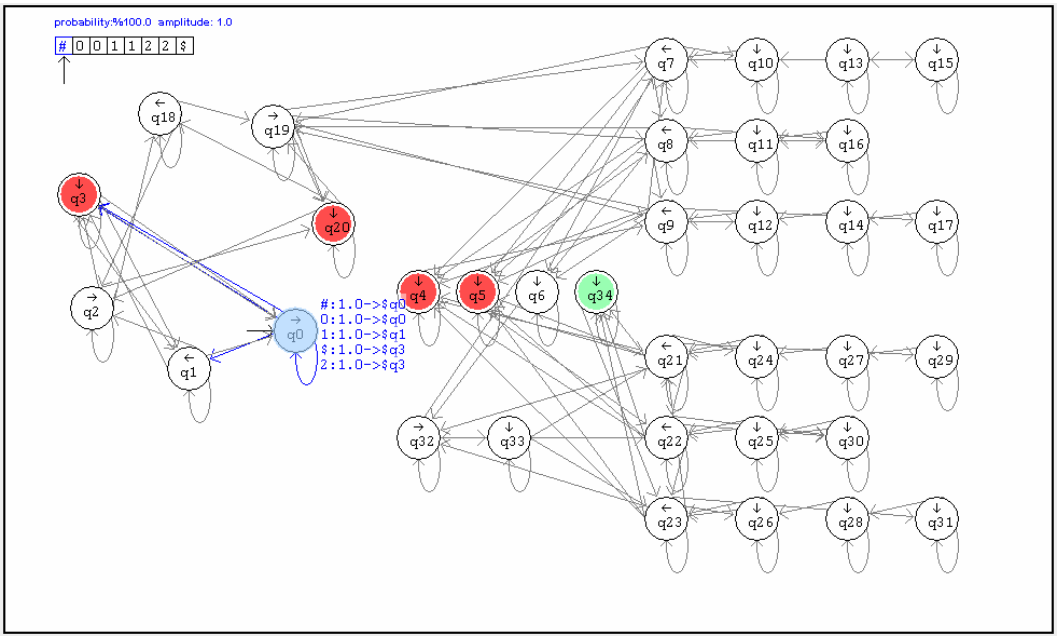

Figure 2.23: A construction for a 2QFA recognizing the language $\{0^n1^n2^n \mid n > 0\}$

Figure 2.23 is a visualization of the construction that recognizes this language. It works in three phases. In the first phase, it checks the validity of the string, i.e. whether it is in 0*1*2* form. In second phase, it checks whether the 1^n2^n part is valid. If that is

invalid, it rejects with probability $\frac{N-1}{N}$. Else, it continues to the phase three. In phase three, it checks for the validity of 0^n1^n part. If it is invalid, then it again rejects with the probability $\frac{N-1}{N}$. If it is not of the form 0*1*2*, then it will be rejected with probability 1. If it is a legal string it will be accepted with probability 1.

3. SIMULATORS

3.1. Previous Work

Although we do not have 2QFA devices ready to experiment with, we can still inspect those devices with the help of simulators. These tools allow us to simulate the behavior of an actual 2QFA, but, with a penalty of slowdown, since they cannot take the advantage of quantum speedup. For small enough examples, this slowdown is acceptable and simulators are very useful.

During the time we have started research, there were not too many quantum simulators available. And the few available ones were not fit for our requirements.

Here is a list of simulators we have found on the Internet:

- QUASI [8]
- Quack! [9]
- libquantum [10]
- QCL [11]
- Quantum Entanglement [12]
- Fraunhofer Quantum Computing Simulator [13]
- QDENSITY [14]
- Quantum Information Suite [15]
- M-fun [21]
- Qubit4matlab [22]
- Quantum Octave [23]
- Open Qubit [24]
- Qubiter [25]
- QuaSi 1 / 2 (Quantum Circuit Simulator) [26]
- Quantum Algorithm Designer [27]
- jQuantum [28]

- Quantum Computer Emulator [29]
- And more for details please refer to [30], which is updated frequently.

Here our problems with the simulators:

- The simulation method was not what we wanted. Almost all of them have chosen the circuit implementation of quantum algorithms for simulation, and some of them [11], [25] implemented a high level language which had quantum language constructs like QFT. These methods were not the way we wanted. We planned to inspect QFA simulation in a more similar way to the classical FSA representation.

- One of the common problems with the existing simulators is the user interface. Some of them [8], [26], [27], [28], [29] contains a built in GUI, which is usually a complex or unintuitive one. And most of others do not have a GUI at all. [9], [10], [11],.. Some of the simulators available [10], [11], [24] are just designed to be used as a library or a command line tool. Doing this way has its own advantages for sure, but it requires a certain level of experience with the library even to perform simple tests. For not getting into such a situation, we decided to add a default user interface to the simulator, while not preventing it to be used as a library, without use of the default interface.

- Another common problem with the simulators having a user interface is the complexity of the interface. [8], [26], [27], [28], [29] To learn the user interface of the simulators usually needs extra training, even if you have a strong conceptual background. Because of this reason, one of the design criteria of our simulator is user friendly interface, which would be easy to use not only for experienced users, but the ones who has only classical automaton background.

- Another issue is the interpretation of the quantum world. There are many different interpretations of the quantum superposition phenomena. We decided to use the multiple-world interpretation for our simulator. [20] This way, one can easily view all possible configurations in any stage of the computation.

3.2. Our 2QFA Simulator

3.2.1. Design Criteria:

- It may be used as a library package for simulating a 2QFA.
- It will include a default GUI, which will be user friendly, meanwhile presenting all the functionality of the library to the user.
- It will follow Multiple World Interpretation of quantum phenomena.
- It will do most of the common machine building tasks automatically, whenever it is applicable. (Checking for the unitarity of the transition matrices; completing not-mentioned transitions etc.)

3.2.2. Current Features

Our simulator is still in development. We continuously add it new features. Up to now, we have covered most of the design criteria. We are trying to automate the remaining parts of the process.

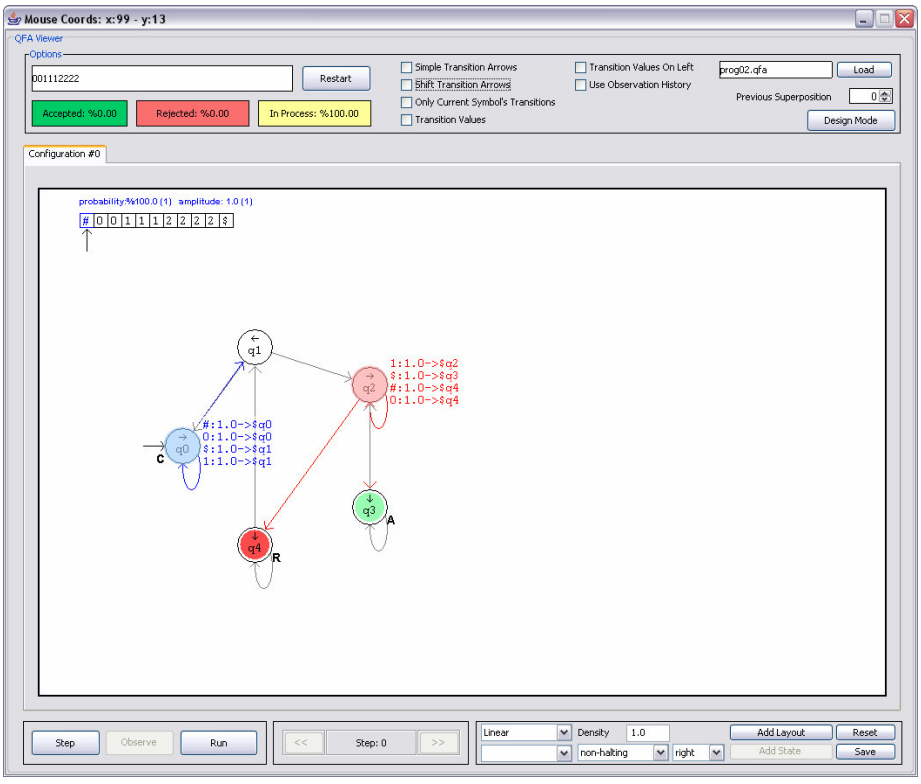

Figure 3.1: General Layout of the Simulator

Our simulator has three main parts. Namely, the configuration part, the display part and the control part.

3.2.2.1. Configuration Part. In the configuration part, you can control the 2QFA definition file, tape content, history traversing and change some display options. It also shows the Accept, Reject and In-process probabilities of the 2QFA since the beginning of the simulation.

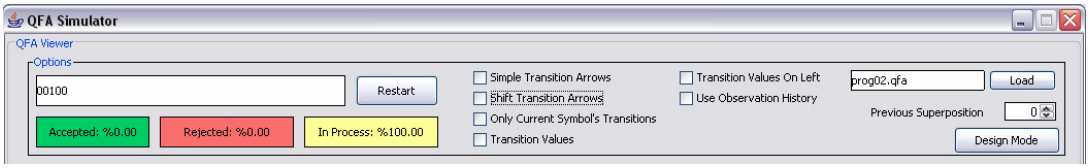

Figure 3.2: Configuration part

Here, the first thing you will want to do is to load a 2QFA definition file. To do that, write the name of the file into the **Input File** box, and press **Restart** button to start over the simulation. For the format of the files, please refer to Appendix C: File format.

Figure 3.3: Loading part

You may also want to change contents of the tape. To do that, simply enter the contents to the input box. Start (#) and end ($) symbols are appended automatically. After entering appropriate tape data, press **Restart** button to start over simulation.

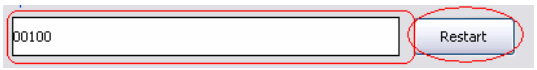

Figure 3.4: Tape input part

3.2.2.2. Options.

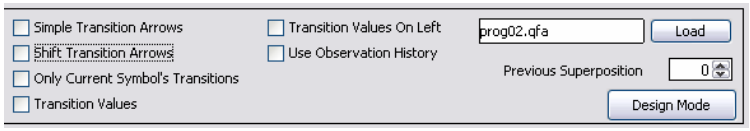

Figure 3.5: Options part

- *Simple Transition Arrows*: Having all of the transition arrows sometimes makes the view too confusing to be understood. When this option is selected, simulator does not display the transition arrows other than the current state and the selected state.

- *Shift Transition Arrows:* If two arrows conflict, it may be difficult to see the arrows. Selecting this option will prevent any conflict of the arrows, by shifting each arrow by a certain amount.

- *Only Current Symbol's Transitions:* Having all the transitions for all symbols shown in the view can make the view complicated. Selecting this option will prevent displaying any transition information other than the symbol under the tape head. This reduces the confusion.

- *Transition Values:* Selecting this option will display numeric values for the transitions, when Simple Transition Arrows options is not selected.

- *Transition Values on Left:* This option will move the numeric display to the left side of the states. It may be useful when the state is at right end or there is a visual conflict at the current display position.

- *Use Observation History:* This option is related to the computation history view. The superposition values used as the history are read from at two different points. One is just after a new super position is created, and the other is just before an observation is made. This option allows choosing which source will be used for history viewing.

- *Previous Superposition:* This allows seeing the computation history for the current input. You may change the value between 0 (corresponds to the current configuration) and number of steps taken (corresponds to the initial configuration). This way you may see all the computation history step by step. When viewing the history information, computational options are disabled. To enable them, enter 0.

3.2.2.3. Global Probabilities. Accepted shows the probability of having the 2QFA stopped in an accepting configuration. Rejected is similar to the Accept, only it counts the probability of rejecting the input. And In-Process is the probability of not-halting so far (which is %100.0 in the case).

Figure 3.6: Global Accept / Reject / In process probabilities

3.2.2.4. Control Part. Simulation of the 2QFA is controlled from the control part. The execution of a 2QFA step consists of 2 operations. In first one, a transition is made. And in the second one, an observation is done. And these two operations are repeated until the machine comes to a halting state. In the simulator, you can either run it step by step, or run it until it halts, or a predefined maximum number of simulation steps have been executed. In each simulation step, a new superposition is created according to the current superposition of the 2QFA. All the previously created superpositions can be accessed using the GUI. A superposition may contain a single configuration as well as multiple configurations. After a simulation step is processed, an observation must be done. Making an observation collapses the current superposition to non-halting configurations. If there are halting configurations, then number of the configurations will be decreased, so you may or may not see the configurations in their previous tabs in GUI. If there are any halting configurations, it also updates the global halting probabilities of ACCEPT and REJECT. Then the collapsed probabilities are re-scaled to keep a total of 1.

Figure 3.7: Execution control part

The **Step** button performs one simulation step. Once the Step button is clicked, a single step of simulation will be performed and it will be disabled until The Observe button is clicked. The Observe button performs the observation and collapses the current superposition to the non-halting configurations. And it will be disabled until Step button is clicked again.

3.2.2.5. Display Part: In the display part, there are two possible modes: Simulation mode and the design mode. In design mode, you can design some details of the 2QFA as well as the visual layout.

In the simulation mode, the current configuration of the 2QFA is shown. If there is more than one possible configuration, then all are displayed in separate tabs. Tabs are named beginning from Configuration #0 up to the number of configurations contained in the current superposition of the 2QFA. Each configuration tab displays the probability, tape contents and head position, 2QFA states, transition arrows, and transition amplitudes of that configuration. You can select (activate) a state by clicking on it. If a state is selected, then it will be highlighted with a red color. The selected state's transition amplitudes will be shown with a more noticeable color. You can pan the display area by dragging with the mouse.

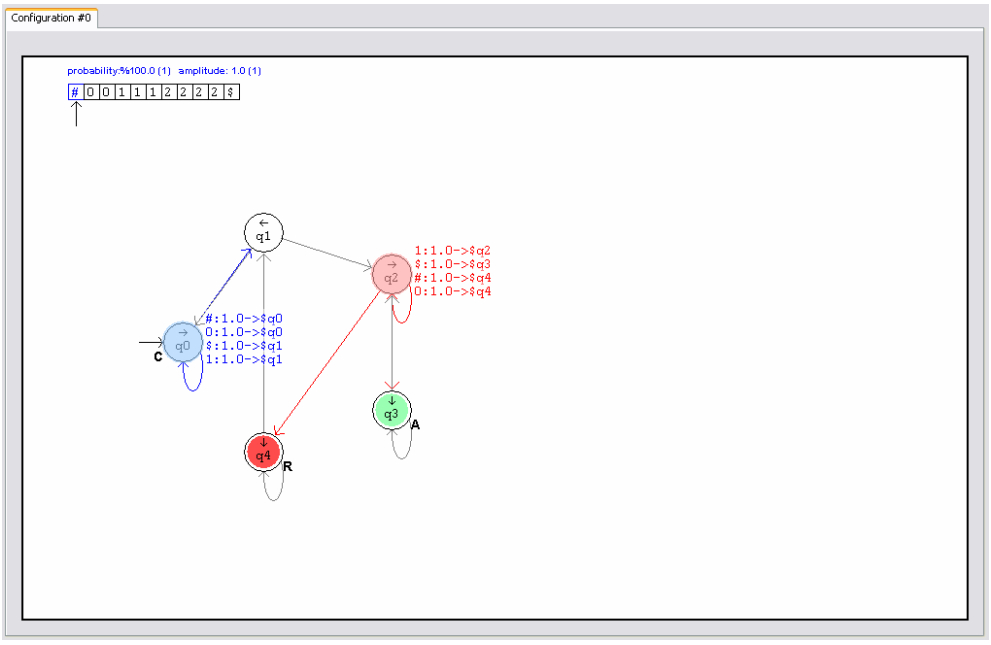

Figure 3.8: State diagram part

The title of the tab shows the configuration number. In top left corner, tape contents and the tape head position is displayed. Active tape cell is also colored blue. On top of tape, the probability of this configuration is displayed. (Since there is a single configuration in this case, its probability is 100%)

The current state is marked with a blue circle around it, and a bunch of numbers corresponding to the amplitudes of transitions from the current state. The transition arrows are colored blue.

The selected state has also numerical display of the transition probabilities. Also the transition arrows are colored red.

4. EXPERIMENTS WITH THE SIMULATOR

Here are some 2QFA implementations for some famous languages. All implementations are presented with machine graph, transition table and a small algorithm explanation.

4.1. Machine 1

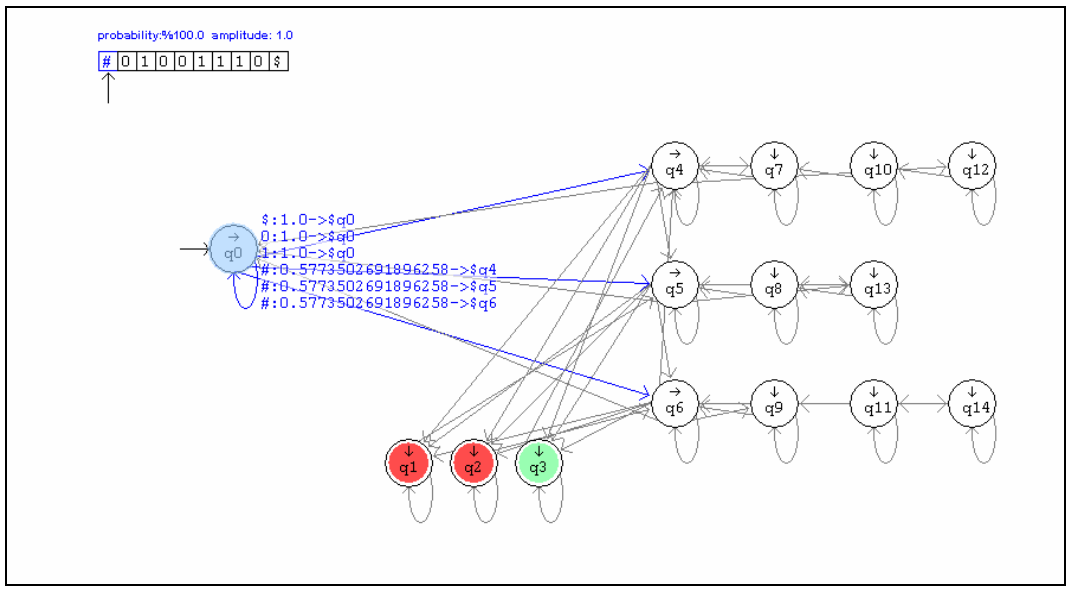

Figure 4.1: Diagram for Machine 1.

(For transition table, refer to Appendix A: Machine 1)

This machine recognizes the language L, consisting of {0,1}* form strings having equal number of 0s and 1s. If input is element of L, then it is accepted with 100% probability. If not, it is rejected (N-1)/N probability.

The construction checks the equality of the numbers of 0s and 1s. For a more detailed overview please refer to the previous section, in which the detailed construction of the machine for language $\{0^n1^n \mid n > 0\}$ discussed.

4.2. Machine 2

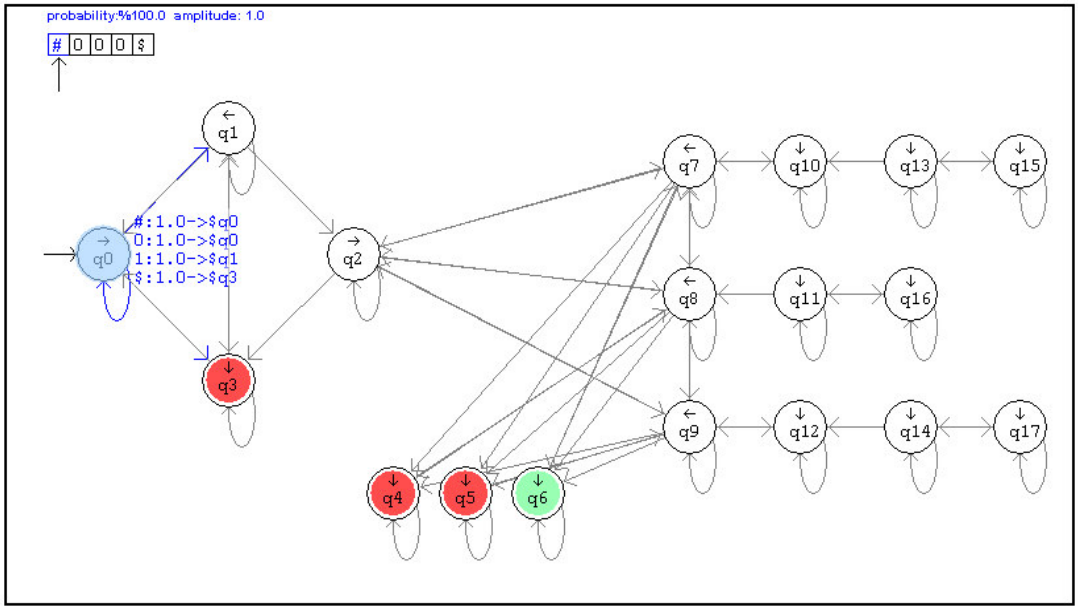

Figure 4.2: Diagram for Machine 2.

(For transition table, refer to Appendix A: Machine 2)

This is the previously examined machine for recognizing the language $\{0^n1^n \mid n > 0\}$. Since the algorithm of the 2QFA is inspected in detail in Section 2.2, we will not go into detail here.

4.3. Machine 3

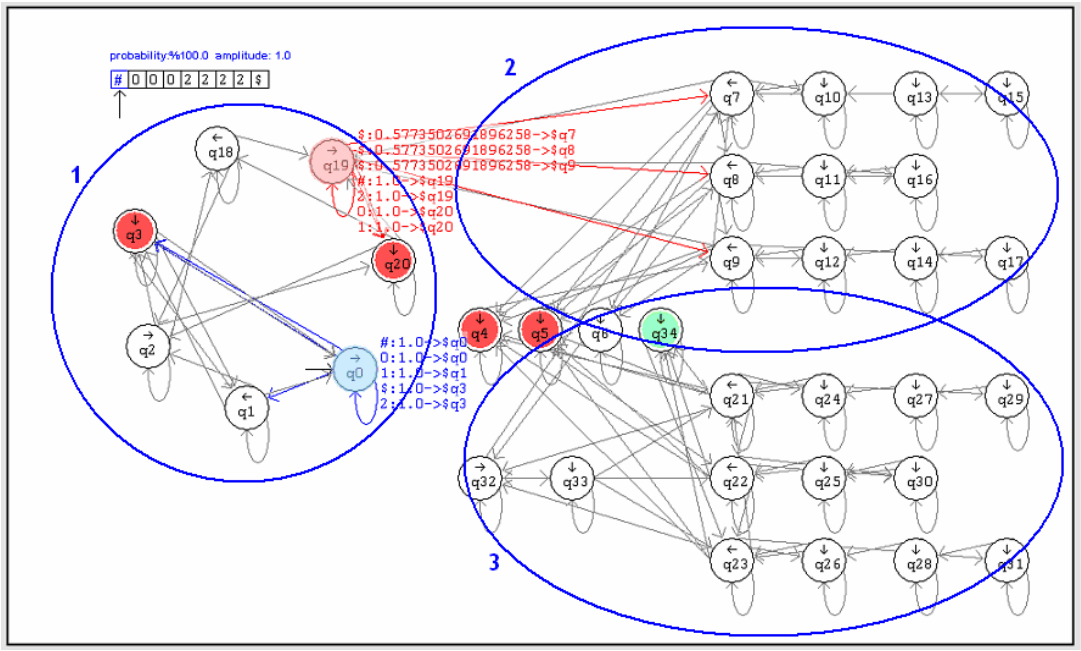

Figure 4.3: Diagram for Machine 3.

(For transition table, refer to Appendix A: Machine 3)

This machine is to recognize the language $\{0^n1^n2^n \mid n > 0\}$. A sketch of an algorithm for this machine is given in [1]. But there were no actual implementations. We have implemented the machine.

4.4. Machine 4

This machine recognizes the non-regular context free language, 0^n1^{2n}.

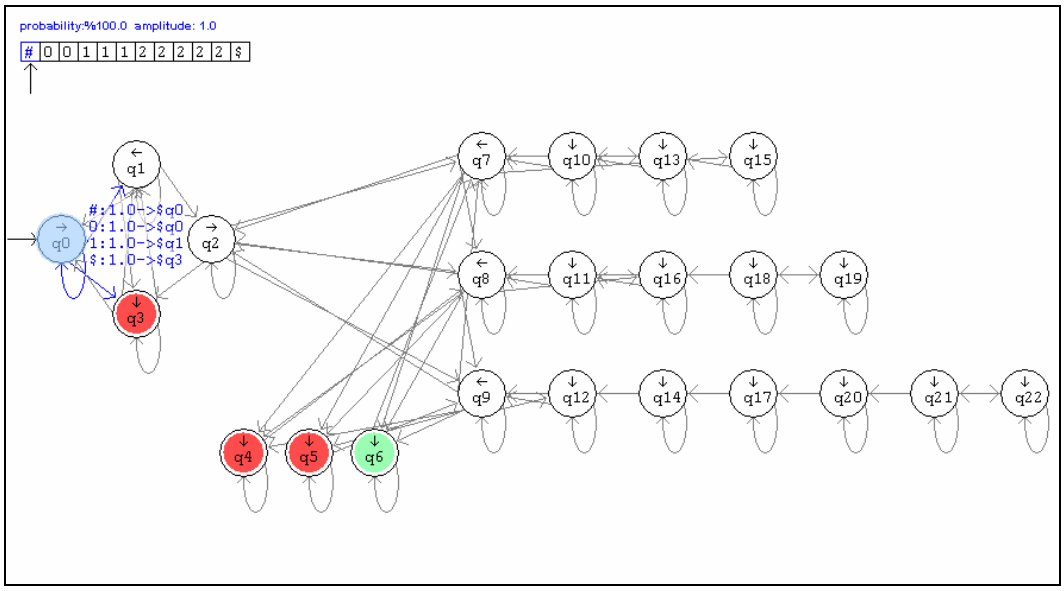

Figure 4.4: Diagram for Machine 4.

(For transition table, refer to Appendix A: Machine 4)

The algorithm of this construction is similar to the previous ones, but with a slight difference. In this construction, the speed of the branches for symbol 1 is halved. That way, the equation will only hold if 1s are twice the number of zeros.

4.5. Machine 5

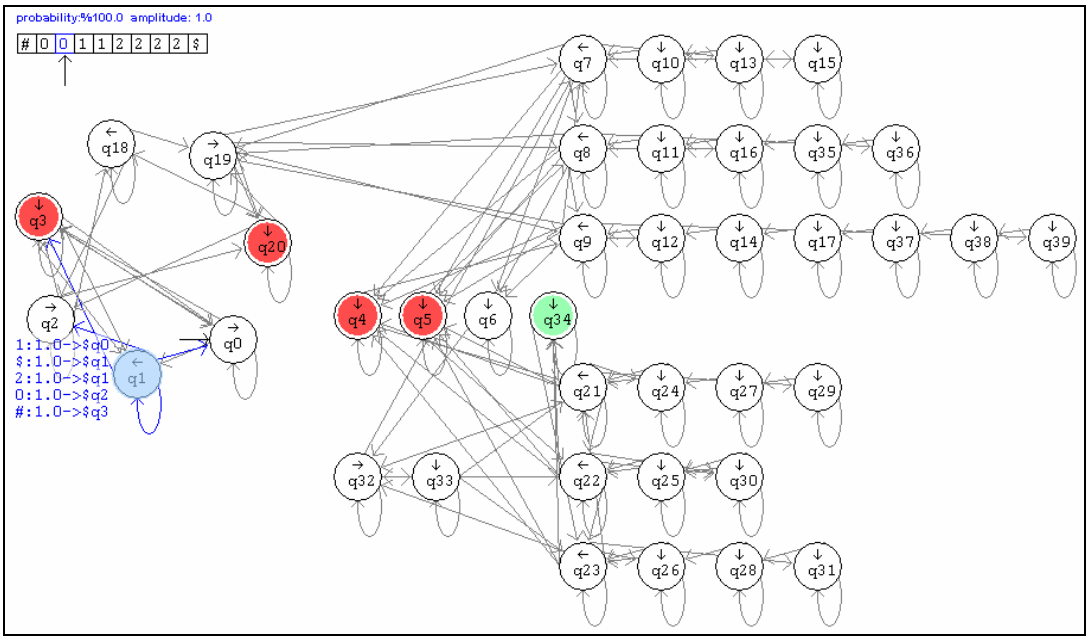

Figure 4.5: Diagram for Machine 5.

(For transition table, refer to Appendix A: Machine 5)

This machine is to recognize another non-context free language, $\{0^n1^n2^{2n} \mid n > 0\}$.

4.6. Machine 6

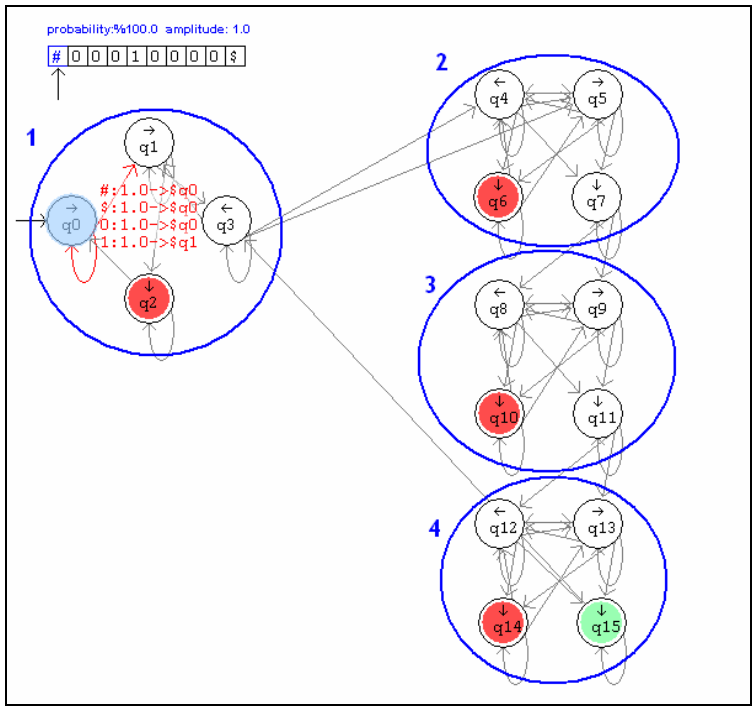

Figure 4.6: Diagram for Machine 6.

(For transition table, refer to Appendix A: Machine 6)

This machine is to recognize the language $\{0^a10^a \mid a > 0\}$, where. This language is a non-regular context free language.

It uses a different algorithm from the previous constructions. It will be inspected in detail in the next section.

4.7. Machine 7

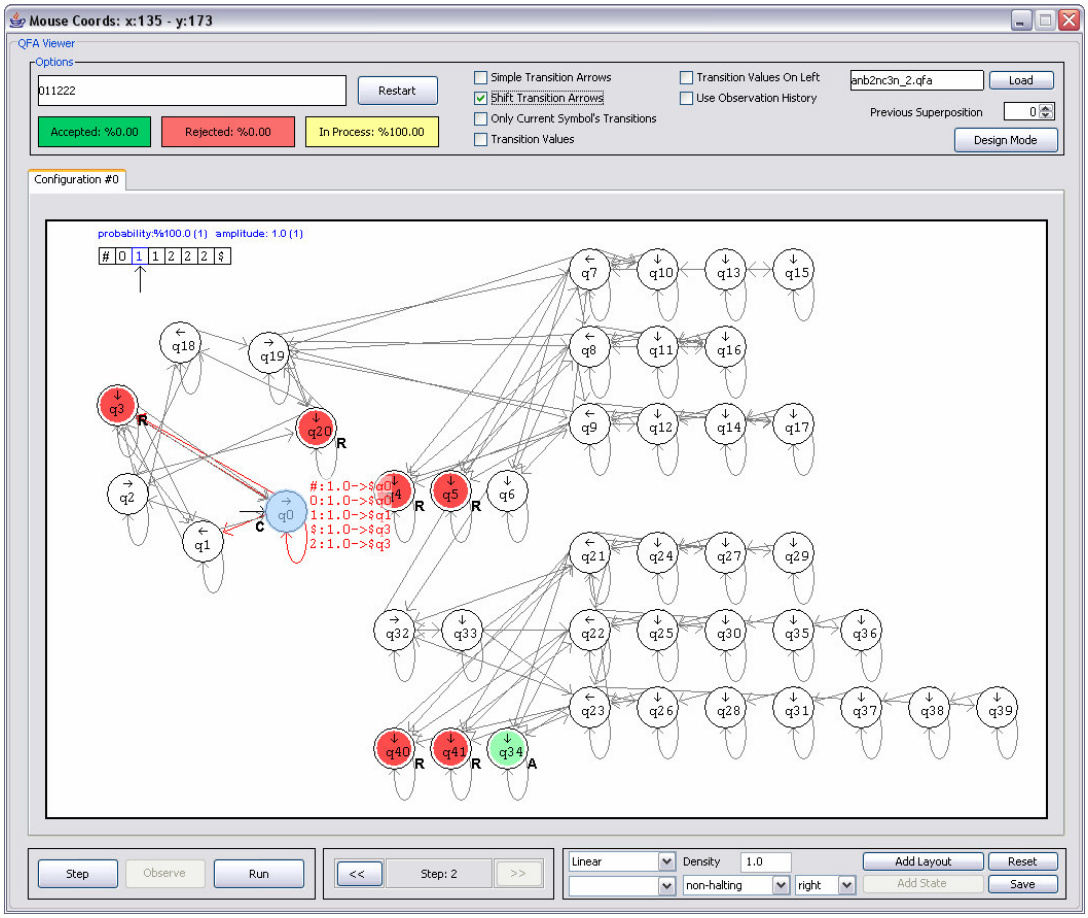

Figure 4.7: Diagram for Machine 7.

(For transition table, refer to Appendix A: Machine 7)

This is another machine in the first style. This machine recognizes the context free language $\{0^n1^{2n}2^{3n} \mid n > 0\}$. Algorithm details are similar to the previously built machines, so we will not go into detail here.

5. A NEW APPROACH TO ENHANCE 2QFA PROGRAMS

Our aim is to enhance 2QFA programs' language recognition probabilities. We will try to use a different construction to get better performance from some specific QFA programs. We will inspect the results and analyze the enhancement we gained.

5.1. Motivation

We can enhance the behavior of the 2QFA programs in the one sided error cases. In these programs, we can assure the correctness of either ACCEPT or REJECT (but not both) with 100% assurance, but we can have a small error in the other one. At this place, our aim to use the information we have at the end of the computation branch giving the wrong answer. If we can re-process the wrong answering thread, then we can reduce the probability of the mistake further. If the problematic thread gives wrong answer with 1/3 probability, then if we can process this 1/3 one more, we can reduce the probability to 1/9 in the best case scenario.

5.2. Implementation

Original 2QFA:

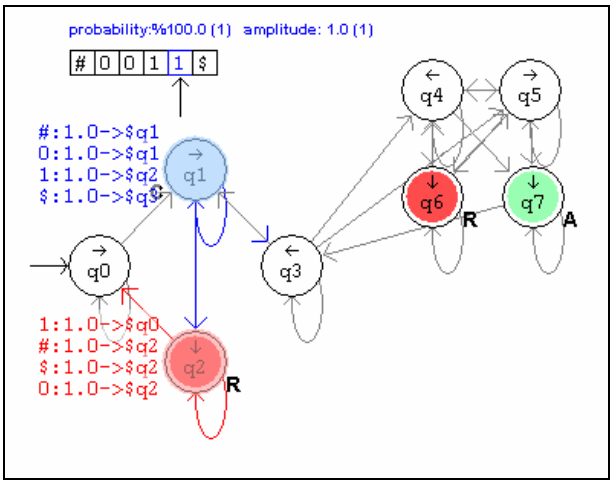

Figure 5.1: State diagram of the original 2QFA for recognizing language $\{0^n10^n \mid n \geq 0\}$

The states q2 and q6, which are shown in red, are reject states. The state q7, that is shown in green, is an accept state. This machine is designed for the language $\{0^n10^n \mid n \geq 0\}$.

Here are the transition matrices of the 2QFA:

Table 5.1: Transition matrices of the 2QFA in Figure 5.1

Transition Matrix for Symbol **#**

#	q0	q1	q2	q3	q4	q5	q6	q7
q0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
q2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
q5	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
q6	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
q7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Transition Matrix for Symbol **0**

0	q0	q1	q2	q3	q4	q5	q6	q7
q0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
q2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
q5	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
q6	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
q7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Transition Matrix for Symbol **1**

1	q0	q1	q2	q3	q4	q5	q6	q7
q0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
q2	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	0.0	0.0	0.0	$2^{-1/2}$	$2^{-1/2}$	0.0	0.0
q4	0.0	0.0	0.0	0.0	0.0	0.0	$-2^{-1/2}$	$2^{-1/2}$
q5	0.0	0.0	0.0	0.0	0.0	0.0	$2^{-1/2}$	$2^{-1/2}$
q6	0.0	0.0	0.0	0.0	$-2^{-1/2}$	$2^{-1/2}$	0.0	0.0
q7	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0

Transition Matrix for Symbol **$**

$	q0	q1	q2	q3	q4	q5	q6	q7
q0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
q2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
q5	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
q6	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
q7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

The algorithm is as follows: The machine works in 2 phases. In the first phase, it validates the input to be of the form 0*10*. If it succeeds, then it moves the tape head to the 1 at the middle. And it splits into 2 threads, processing the left and right parts of the tape. At each step, both threads move the tape head to the end of the tape. At the end of the tape, the threads change direction, and begin moving to the 1 at the middle. When they reach the 1, they perform a QFT. If both the threads came at the same time, amplitudes for q6 cancel each other so the input string is accepted. If the string has unequal number of zeroes at each side of the 1, then the threads will arrive at the middle 1 at different times. So, cancellation on state q6 will not occur. So, the input will be accepted by 50% probability and rejected by 50% probability.

Table 5.3 is a trace for the execution of this 2QFA. At each step, first we apply a transition, then we apply an observation. At the observation, the halting states are removed. And the amplitudes of the remaining states in the superposition is normalized. Table 5.3 shows the simulation of the 2QFA for the input 0100. At any step, all state/tape position pairs of the superposition are listed one in a row. So, the table should be interpreted in this way: Let us inspect the row labeled step 9:

Table 5.2: A sample trace of the 2QFA

Step 9	**State:**q4	**Amp:**$2^{-1/2}$	**TapePos:**1
	State:q5	**Amp:**$2^{-1/2}$	**TapePos:**3
Observation after step 9	**State:**q4	**Amp:**$2^{-1/2}$	**TapePos:**1
	State:q5	**Amp:**$2^{-1/2}$	**TapePos:**3

In the first part of step 9, the transitions are performed. And after the transitions, we have two states in our superposition, q4 and q5. They both have the same amplitude $2^{-1/2}$, and one of them is on tape position 1 and the other one is on tape position 3. After that, we perform an observation. If q4 or q5 was a halting state, then it would be removed and the other one would be normalized accordingly. But since that is not the case, nothing has been changed during observation. All the table could be interpreted in a similar way.

Table 5.3: A sample trace of the machine for input 0100

	State	Amplitude	Tape Position
Step 1	q0	1	1
Observation after step 1	q0	1	1
Step 2	q0	1	2
Observation after step 2	q0	1	2
Step 3	q1	1	3
Observation after step 3	q1	1	3
Step 4	q1	1	4
Observation after step 4	q1	1	4
Step 5	q1	1	5
Observation after step 5	q1	1	5
Step 6	q3	1	4
Observation after step 6	q3	1	4
Step 7	q3	1	3
Observation after step 7	q3	1	3
Step 8	q3	1	2
Observation after step 8	q3	1	2
Step 9	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	3
Observation after step 9	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	3
Step 10	q4	$2^{-1/2}$	0
	q5	$2^{-1/2}$	4
Observation after step 10	q4	$2^{-1/2}$	0
	q5	$2^{-1/2}$	4

Step 11	q5	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5
Observation after step 11	q5	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5
Step 12	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Observation after step 12	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Step 13	q6	1 / 2	2
	q7	1 / 2	2
	q4	$2^{-1/2}$	3
Observation after step 13	q4	1	3
	Here, system collapsed on state q4, since q6 and q7 are halting states		
Step 14	q4	1	2
Observation after step 14	q4	1	2
Step 15	q6	$-2^{-1/2}$	2
	q7	$2^{-1/2}$	2
Observation after step 15	Simulator Halted. There are no active non-halting states.		
	Here, simulation is ended, since both q6 and q7 are halting states. Result: 50% Accept, 50% Reject		

Table 5.4: A sample trace of the machine for input 00100

	State	Amplitude	Tape Position
Step 1	q0	1	1
Observation after step 1	q0	1	1
Step 2	q0	1	2
Observation after step 2	q0	1	2
Step 3	q0	1	3
Observation after step 3	q0	1	3
Step 4	q1	1	4
Observation after step 4	q1	1	4
Step 5	q1	1	5
Observation after step 5	q1	1	5

Step 6	q1	1	6
Observation after step 6	q1	1	6
Step 7	q3	1	5
Observation after step 7	q3	1	5
Step 8	q3	1	4
Observation after step 8	q3	1	4
Step 9	q3	1	3
Observation after step 9	q3	1	3
Step 10	q4	$2^{-1/2}$	2
	q5	$2^{-1/2}$	4
Observation after step 10	q4	$2^{-1/2}$	2
	q5	$2^{-1/2}$	4
Step 11	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5
Observation after step 11	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5
Step 12	q4	$2^{-1/2}$	0
	q5	$2^{-1/2}$	6
Observation after step 12	q4	$2^{-1/2}$	0
	q5	$2^{-1/2}$	6
Step 13	q5	$2^{-1/2}$	1
	q4	$2^{-1/2}$	5
Observation after step 13	q5	$2^{-1/2}$	1
	q4	$2^{-1/2}$	5
Step 14	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Observation after step 14	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Step 15	q5	$2^{-1/2}$	3
	q4	$2^{-1/2}$	3
Observation after step 15	q5	$2^{-1/2}$	3
	q4	$2^{-1/2}$	3
Step 16	q6	0	3
	q7	1	3

	Simulator Halted. There are no active non-halting states.
Observation after step 16	Here, simulation is ended, since q7 is a halting state. Result: 100% Accept

With the current design, it has a false accept ratio of 50%. With this ratio, this construction can not be used to validate input strings for the given language, since it does not converge. To be able to use a machine, to validate a string with desired probability, the machine must have a failure ratio ε less than ½. This way, after a number of repetitions, the majority result will very likely be the correct one, since the probability of having the wrong result in the majority gets smaller exponentially.

We will make some enhancements to this machine to decrease the failure ratio. Our method is simple. In the q7 state, instead of accepting it as is, we will restart processing. To do that, we will put a clone of the phase 2 of the machine at this point (maybe with some modifications) and it will process the thread that will accept otherwise. So, that should decrease the error ratio.

Figure 5.2 shows a sample construction. This is a 2 level construction. We only appended a single copy for this. We may add as many as we wish.

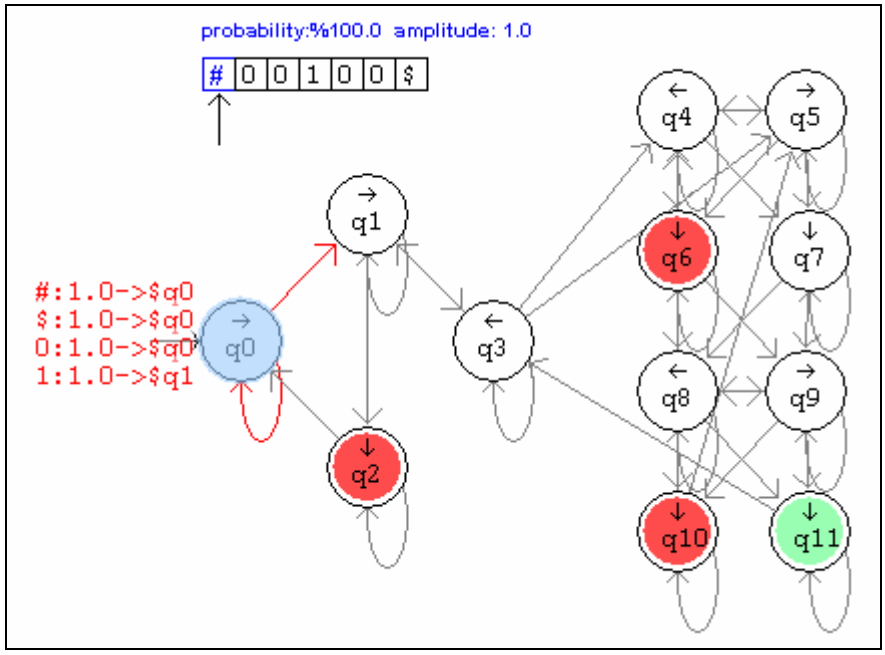

Figure 5.2: Construction sample of 2 levels

In q7 state, instead of accepting, we add a clone of the processing part, and we re-process the input. Since the tape head will be returned to the initial position of processing states (onto the 1 in the middle) after processing the input, we do not need any extra states to rewind the tapes etc. With the new construction, we may expect to have an error ratio of ½ * ½ = ¼, since at each iteration step we can reject 50% with confidence. But when we execute this on the simulator, the results are different: it has a false accept ratio of 37.50%.

Table 5.5: A sample trace of the machine for input 0100

	State	Amplitude	Tape Position
Step 1	q0	1	1
Observation after step 1	q0	1	1
Step 2	q0	1	2
Observation after step 2	q0	1	2
Step 3	q1	1	3
Observation after step 3	q1	1	3
Step 4	q1	1	4
Observation after step 4	q1	1	4
Step 5	q1	1	5
Observation after step 5	q1	1	5
Step 6	q3	1	4
Observation after step 6	q3	1	4
Step 7	q3	1	3
Observation after step 7	q3	1	3
Step 8	q3	1	2
Observation after step 8	q3	1	2
Step 9	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	3
Observation after step 9	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	3
Step 10	q4	$2^{-1/2}$	0
	q5	$2^{-1/2}$	4
Observation after step 10	q4	$2^{-1/2}$	0

	q5	$2^{-1/2}$	4
Step 11	q5	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5
Observation after step 11	q5	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5
Step 12	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Observation after step 12	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Step 13	q6	1 / 2	2
	q7	1 / 2	2
	q4	$2^{-1/2}$	3
Observation after step 13	q7	$3^{-1/2}$	2
	q4	$(6^{-1/2}) * 2$	3
	Here, system collapsed on q7 and q4, since q6 is a halting state		
Step 14	q8	$6^{-1/2}$	1
	q9	$6^{-1/2}$	3
	q4	$(6^{-1/2}) * 2$	2
Observation after step 14	q8	$6^{-1/2}$	1
	q9	$6^{-1/2}$	3
	q4	$(6^{-1/2}) * 2$	2
Step 15	q8	$6^{-1/2}$	0
	q9	$6^{-1/2}$	4
	q6	$-3^{-1/2}$	2
	q7	$3^{-1/2}$	2
Observation after step 15	q8	1 / 2	0
	q9	1 / 2	4
	q7	$2^{-1/2}$	2
	Since q6 is a halting state, system collapsed on q7, q8, q9		
Step 16	q9	1 / 2	1
	q9	1 / 2	5
	q8	1 / 2	1
	q9	1 / 2	3
Observation after step 16	q9	1 / 2	1
	q9	1 / 2	5

	q8	1 / 2	1
	q9	1 / 2	3
Step 17	q9	1 / 2	2
	q8	1 / 2	4
	q8	1 / 2	0
	q9	1 / 2	4
Observation after step 17	q9	1 / 2	2
	q8	1 / 2	4
	q8	1 / 2	0
	q9	1 / 2	4
Step 18	q10	$(2^{-1 / 2}) / 2$	2
	q11	$(2^{-1 / 2}) / 2$	2
	q8	1 / 2	3
	q9	1 / 2	1
	q9	1 / 2	5
Observation after step 18	q8	$3^{-1 / 2}$	3
	q9	$3^{-1 / 2}$	1
	q9	$3^{-1 / 2}$	5
	Since q10 and q11 are halting states, system collapses on q8 and q9.		
Step 19	q8	$3^{-1 / 2}$	2
	q9	$3^{-1 / 2}$	2
	q8	$3^{-1 / 2}$	4
Observation after step 19	q8	$3^{-1 / 2}$	2
	q9	$3^{-1 / 2}$	2
	q8	$3^{-1 / 2}$	4
Step 20	q10	0	2
	q11	$(6^{-1 / 2}) * 2$	2
	q8	$3^{-1 / 2}$	3
Observation after step 20	q8	1	3
	Since q10 and q11 are halting states, system collapses on q8.		
Step 21	q8	1	2
Observation after step 21	q8	1	2
Step 22	q10	$-2^{-1 / 2}$	2
	q11	$2^{-1 / 2}$	2

Observation after step 22	Simulator Halted. There are no active non-halting states. Here, simulation is ended, since both q10 and q11 are halting states. Result: 37.5% Accept 62.5% Reject

Table 5.6: A sample trace of the machine for input 00100

	State	Amplitude	Tape Position
Step 1	q0	1	1
Observation after step 1	q0	1	1
Step 2	q0	1	2
Observation after step 2	q0	1	2
Step 3	q0	1	3
Observation after step 3	q0	1	3
Step 4	q1	1	4
Observation after step 4	q1	1	4
Step 5	q1	1	5
Observation after step 5	q1	1	5
Step 6	q1	1	6
Observation after step 6	q1	1	6
Step 7	q3	1	5
Observation after step 7	q3	1	5
Step 8	q3	1	4
Observation after step 8	q3	1	4
Step 9	q3	1	3
Observation after step 9	q3	1	3
Step 10	q4	$2^{-1/2}$	2
	q5	$2^{-1/2}$	4
Observation after step 10	q4	$2^{-1/2}$	2
	q5	$2^{-1/2}$	4
Step 11	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5
Observation after step 11	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5
Step 12	q4	$2^{-1/2}$	0

	q5	$2^{-1/2}$	6
Observation after step 12	q4	$2^{-1/2}$	0
	q5	$2^{-1/2}$	6
Step 13	q5	$2^{-1/2}$	1
	q4	$2^{-1/2}$	5
Observation after step 13	q5	$2^{-1/2}$	1
	q4	$2^{-1/2}$	5
Step 14	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Observation after step 14	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Step 15	q5	$2^{-1/2}$	3
	q4	$2^{-1/2}$	3
Observation after step 15	q5	$2^{-1/2}$	3
	q4	$2^{-1/2}$	3
Step 16	q6	0	3
	q7	1	3
Observation after step 16	q6	0	3
	q7	1	3
Step 17	q8	$2^{-1/2}$	2
	q9	$2^{-1/2}$	4
Observation after step 17	q8	$2^{-1/2}$	2
	q9	$2^{-1/2}$	4
Step 18	q8	$2^{-1/2}$	1
	q9	$2^{-1/2}$	5
Observation after step 18	q8	$2^{-1/2}$	1
	q9	$2^{-1/2}$	5
Step 19	q8	$2^{-1/2}$	0
	q9	$2^{-1/2}$	6
Observation after step 19	q8	$2^{-1/2}$	0
	q9	$2^{-1/2}$	6
Step 20	q9	$2^{-1/2}$	1
	q8	$2^{-1/2}$	5
Observation after step 20	q9	$2^{-1/2}$	1
	q8	$2^{-1/2}$	5

Step 21	q9	$2^{-1/2}$	2
	q8	$2^{-1/2}$	4
Observation after step 21	q9	$2^{-1/2}$	2
	q8	$2^{-1/2}$	4
Step 22	q9	$2^{-1/2}$	3
	q8	$2^{-1/2}$	3
Observation after step 22	q9	$2^{-1/2}$	3
	q8	$2^{-1/2}$	3
Step 23	q10	0	3
	q11	1	3
Observation after step 23	Simulator Halted. There are no active non-halting states. Accepted with 100% probability.		

The previous result, 50%, was useless. But the new result is much better, since we can use it to test any input with an assurance level we choose.

The reason of the difference between the expected probability versus actual probability is related to the architecture of the machine itself. There are some canceling superpositions between the first layer and the second layer. For example, in the Table 5.6 for input 0100, we can observe this behavior at step 19 and 20. At step 19, we have 3 states in our superposition. The first and second one are on tape position 2, and are just about to perform a QFT to states q10 and q11. If we trace these threads backwards, we will see that one of them is the thread which processed the longer side of the input in the first level, and then processed the shorter side of the input in the second level, and the other one processed the shorter side in the first level and longer side in the second level. So, the total number of processed symbols for each thread is the same and they finish processing level 2 at the same time. And because of the construction, since they finish at the same time, both threads end up in the accepting state of the 2QFA. So this causes us to lose some of the assurance level. This loss reveals another question: “If this causes a loss in the assurance, will not it cause a worse result than 50% in some specific conflicts?” The answer is no. Because, at the first step the simulation goes from layer one to layer two, we already guarantee a correct reject ratio of 50%. (Since we can not re-accept in a thread that has already rejected). Even if all the remaining part cancels each other –which is very unlikely– we still have our original assurance level. But this full cancellation is very

unlikely if you construct the machine well. For this machine, full cancellation is impossible. To have full cancellation, the process must flow to the state q11, regardless of the input in second level. But this is not possible unless the input has equal number of 0s on both sides of the symbol 1 in the middle, because of the construction of the machine. In this case, the input is already a valid input, so it will be accepted with 100% probability. Also with some minor changes, this cancellation can be eliminated or reduced.

Figure 5.3 is the changed version of the same machine, with an extra state, to remove the cancellation:

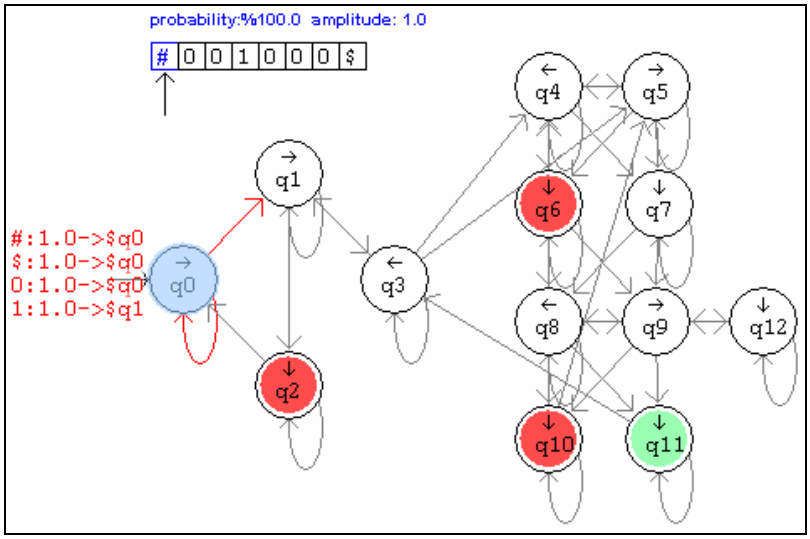

Figure 5.3: Changed version of the machine

Table 5.7: Transition matrices for the 2QFA in Figure 5.3

Transition Matrix for Symbol #

#	q0	q1	q2	q3	Q4	q5	q6	q7	q8	q9	q10	q11	q12
q0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q5	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q6	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
q7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
q8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
q9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
q10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
q11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
q12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Transition Matrix for Symbol **0**

0	q0	q1	q2	q3	Q4	q5	q6	q7	q8	q9	q10	q11	q12
q0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q5	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q6	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
q7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
q8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
q9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
q10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
q11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
q12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0

Transition Matrix for Symbol **1**

1	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12
q0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q2	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	0.0	0.0	0.0	$2^{-1/2}$	$2^{-1/2}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	0.0	0.0	$-2^{-1/2}$	$2^{-1/2}$	0.0	0.0	0.0	0.0	0.0
q5	0.0	0.0	0.0	0.0	0.0	0.0	$2^{-1/2}$	$2^{-1/2}$	0.0	0.0	0.0	0.0	0.0
q6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	$-2^{-1/2}$	$2^{-1/2}$	0.0	0.0	0.0
q7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	$2^{-1/2}$	$2^{-1/2}$	0.0	0.0	0.0
q8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	$-2^{-1/2}$	$2^{-1/2}$	0.0
q9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	$2^{-1/2}$	$2^{-1/2}$	0.0
q10	0.0	0.0	0.0	0.0	$-2^{-1/2}$	$2^{-1/2}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q11	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Transition Matrix for Symbol **$**

$	q0	q1	q2	q3	Q4	q5	q6	q7	q8	q9	q10	q11	q12
q0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q1	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q3	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q4	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q5	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
q6	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
q7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
q8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
q9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
q10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
q11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
q12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

This version has the failure ratio of 25%. That means, it rejects nonmembers with ½ at first layer, and with ½ of the remaining in the second layer. And it accepts nonmembers with ¼ probability.

Table 5.8: A sample trace of the machine for input 0100

	State	Amplitude	Tape Position
Step 1	q0	1	1
Observation after step 1	q0	1	1
Step 2	q0	1	2
Observation after step 2	q0	1	2
Step 3	q1	1	3
Observation after step 3	q1	1	3
Step 4	q1	1	4
Observation after step 4	q1	1	4
Step 5	q1	1	5
Observation after step 5	q1	1	5
Step 6	q3	1	4
Observation after step 6	q3	1	4
Step 7	q3	1	3
Observation after step 7	q3	1	3
Step 8	q3	1	2
Observation after step 8	q3	1	2
Step 9	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	3
Observation after step 9	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	3
Step 10	q4	$2^{-1/2}$	0
	q5	$2^{-1/2}$	4
Observation after step 10	q4	$2^{-1/2}$	0
	q5	$2^{-1/2}$	4
Step 11	q5	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5
Observation after step 11	q5	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5

Step 12	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Observation after step 12	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Step 13	q6	1 / 2	2
	q7	1 / 2	2
	q4	$2^{-1/2}$	3
Observation after step 13	q7	$3^{-1/2}$	2
	q4	$(6^{-1/2}) * 2$	3
Step 14	q8	$6^{-1/2}$	1
	q9	$6^{-1/2}$	3
	q4	$(6^{-1/2}) * 2$	2
Observation after step 14	q8	$6^{-1/2}$	1
	q9	$6^{-1/2}$	3
	q4	$(6^{-1/2}) * 2$	2
Step 15	q8	$6^{-1/2}$	0
	q12	$6^{-1/2}$	3
	q6	$-3^{-1/2}$	2
	q7	$3^{-1/2}$	2
Observation after step 15	q8	1 / 2	0
	q12	1 / 2	3
	q7	$2^{-1/2}$	2
Step 16	q9	1 / 2	1
	q9	1 / 2	4
	q8	1 / 2	1
	q9	1 / 2	3
Observation after step 16	q9	1 / 2	1
	q9	1 / 2	4
	q8	1 / 2	1
	q9	1 / 2	3
Step 17	q12	1 / 2	1
	q12	1 / 2	4
	q8	1 / 2	0
	q12	1 / 2	3
Observation after step 17	q12	1 / 2	1
	q12	1 / 2	4

	q8	1 / 2	0
	q12	1 / 2	3
Step 18	q9	1 / 2	2
	q9	1 / 2	5
	q9	1 / 2	1
	q9	1 / 2	4
Observation after step 18	q9	1 / 2	2
	q9	1 / 2	5
	q9	1 / 2	1
	q9	1 / 2	4
Step 19	q10	$(2^{-1 / 2}) / 2$	2
	q11	$(2^{-1 / 2}) / 2$	2
	q8	1 / 2	4
	q12	1 / 2	1
	q12	1 / 2	4
Observation after step 19	q8	$3^{-1 / 2}$	4
	q12	$3^{-1 / 2}$	1
	q12	$3^{-1 / 2}$	4
Step 20	q8	$3^{-1 / 2}$	3
	q9	$3^{-1 / 2}$	2
	q9	$3^{-1 / 2}$	5
Observation after step 20	q8	$3^{-1 / 2}$	3
	q9	$3^{-1 / 2}$	2
	q9	$3^{-1 / 2}$	5
Step 21	q8	$3^{-1 / 2}$	2
	q10	$6^{-1 / 2}$	2
	q11	$6^{-1 / 2}$	2
	q8	$3^{-1 / 2}$	4
Observation after step 21	q8	$2^{-1 / 2}$	2
	q8	$2^{-1 / 2}$	4
Step 22	q10	-1 / 2	2
	q11	1 / 2	2
	q8	$2^{-1 / 2}$	3
Observation after step 22	q8	1	3
Step 23	q8	1	2

Observation after step 23	q8	1	2
Step 24	q10	$-2^{-1/2}$	2
	q11	$2^{-1/2}$	2
Observation after step 24	Simulator Halted. There are no active non-halting states. It rejected with probability %75.		

Table 5.9: A sample trace of the machine for input 00100

	State	Amplitude	Tape Position
Step 1	q0	1	1
Observation after step 1	q0	1	1
Step 2	q0	1	2
Observation after step 2	q0	1	2
Step 3	q0	1	3
Observation after step 3	q0	1	3
Step 4	q1	1	4
Observation after step 4	q1	1	4
Step 5	q1	1	5
Observation after step 5	q1	1	5
Step 6	q1	1	6
Observation after step 6	q1	1	6
Step 7	q3	1	5
Observation after step 7	q3	1	5
Step 8	q3	1	4
Observation after step 8	q3	1	4
Step 9	q3	1	3
Observation after step 9	q3	1	3
Step 10	q4	$2^{-1/2}$	2
	q5	$2^{-1/2}$	4
Observation after step 10	q4	$2^{-1/2}$	2
	q5	$2^{-1/2}$	4
Step 11	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5
Observation after step 11	q4	$2^{-1/2}$	1
	q5	$2^{-1/2}$	5

Step 12	q4	$2^{-1/2}$	0
	q5	$2^{-1/2}$	6
Observation after step 12	q4	$2^{-1/2}$	0
	q5	$2^{-1/2}$	6
Step 13	q5	$2^{-1/2}$	1
	q4	$2^{-1/2}$	5
Observation after step 13	q5	$2^{-1/2}$	1
	q4	$2^{-1/2}$	5
Step 14	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Observation after step 14	q5	$2^{-1/2}$	2
	q4	$2^{-1/2}$	4
Step 15	q5	$2^{-1/2}$	3
	q4	$2^{-1/2}$	3
Observation after step 15	q5	$2^{-1/2}$	3
	q4	$2^{-1/2}$	3
Step 16	q6	0	3
	q7	1	3
Observation after step 16	q6	0	3
	q7	1	3
Step 17	q8	$2^{-1/2}$	2
	q9	$2^{-1/2}$	4
Observation after step 17	q8	$2^{-1/2}$	2
	q9	$2^{-1/2}$	4
Step 18	q8	$2^{-1/2}$	1
	q12	$2^{-1/2}$	4
Observation after step 18	q8	$2^{-1/2}$	1
	q12	$2^{-1/2}$	4
Step 19	q8	$2^{-1/2}$	0
	q9	$2^{-1/2}$	5
Observation after step 19	q8	$2^{-1/2}$	0
	q9	$2^{-1/2}$	5
Step 20	q9	$2^{-1/2}$	1
	q12	$2^{-1/2}$	5
Observation after step 20	q9	$2^{-1/2}$	1

	q12	$2^{-1/2}$	5
Step 21	q12	$2^{-1/2}$	1
	q9	$2^{-1/2}$	6
Observation after step 21	q12	$2^{-1/2}$	1
	q9	$2^{-1/2}$	6
Step 22	q9	$2^{-1/2}$	2
	q8	$2^{-1/2}$	5
Observation after step 22	q9	$2^{-1/2}$	2
	q8	$2^{-1/2}$	5
Step 23	q12	$2^{-1/2}$	2
	q8	$2^{-1/2}$	4
Observation after step 23	q12	$2^{-1/2}$	2
	q8	$2^{-1/2}$	4
Step 24	q9	$2^{-1/2}$	3
	q8	$2^{-1/2}$	3
Observation after step 24	q9	$2^{-1/2}$	3
	q8	$2^{-1/2}$	3
Step 25	q10	0	3
	q11	1	3
Observation after step 25	Simulator Halted. There are no active non-halting states. It accepted the input with 100% probability.		

The reason of this improvement can be summarized as follows: In the original machine, the thread processing the shorter substring of 0s in first level and longer part in second phase, and the thread processing longer part in the first phase and shorter part in the second phase perform equal numbers of tape movements, which almost equal twice the length of the input. To be exact, this number is (|short| + |short| + |long| + |long| + 2) in both cases. (Note the informal but comprehensible usage of the string length operator |.| in this discussion.) To prevent this from happening, we have added a spin wait state. A *spin wait state* is a dummy state with the sole purpose of preventing the unwanted synchronization between levels of the 2QFA. When we add the dummy state, it will move the tape head one cell at each step while processing from right to left. But it will move the tape head once in two steps while processing from left to right. This way, processing will take a bit longer. Because in the second level the threads will perform different number of operations

than in the first one: The length of the thread traversing the left part is 3*|left| and the length of the thread traversing the right part is 3*|right|. So the overall lengths of the four threads that may exist in the last level are: 5*|left|, 5*|right|, 2*|left|+3*|right|, and 3*|left|+2*|right|. It is easy to see that no two of these can be of equal length, unless |left| = |right|. Therefore there will not be any unwanted cancellations.

This way, we can increase the number of layers, and get higher accept ratios. But when the number of levels are increased, the chance of cancellations increase. To prevent any cancellations, we may need to add more spin wait states at different levels. Any added states should be checked so that, they do not cause equal number of operations at different levels of the 2QFA, which can cancel each other.

5.3. Importance

By use of this method we can have these benefits:

1. For single sided ½ probability assured constructions: These devices can not be used for validation, since we can not repeat the process again and again to get a required assurance level. But with this change, these machines become usable.
2. For any given 2QFA, if we can isolate the falsely accepting state, then we can apply this technique for that machine. We can append a copy of the machine at that state, and try to perform extra refinement on the possibly false accepting thread. But this may not lead to considerable gain if unwanted cancellations are high.

We now test whether this technique will bring any such improvement to machines previously presented in the literature. For the first test, we choose the first machine we have examined, which was supposed to recognize the language $\{0^n1^n \mid n > 0\}$ with 1/N error ratio, where N = 3. We will append a copy of the original machine to the accepting branch. But since addition of the first phase of the original machine will not increase our accept ratio nor effect functioning of the 2QFA, we just append the second phase. Beside appending it, we also took care of the rewinding tape head. So if the string is already a member of the language, it will re-process the input in the same way and it will accept with 100% assurance. Therefore, we will not lose anything for the members of the language.

And if the input is not a member of the language, the rejecting branches will reject by 2/3 probability. But it will accept the remaining 1/3. So, at that point, we append a copy of the 2QFA to the accepting state to re-process that 1/3 part. A naive expectation is, re-processing the input and re-rejecting 2/3, so a false accept ratio of 1/9. If we build the machine of [1] for the language $\{0^n1^n \mid n > 0\}$ with N=9 to get 1/9 probability, we would need 95 states. Also its complexity would be worse than our expected complexity.

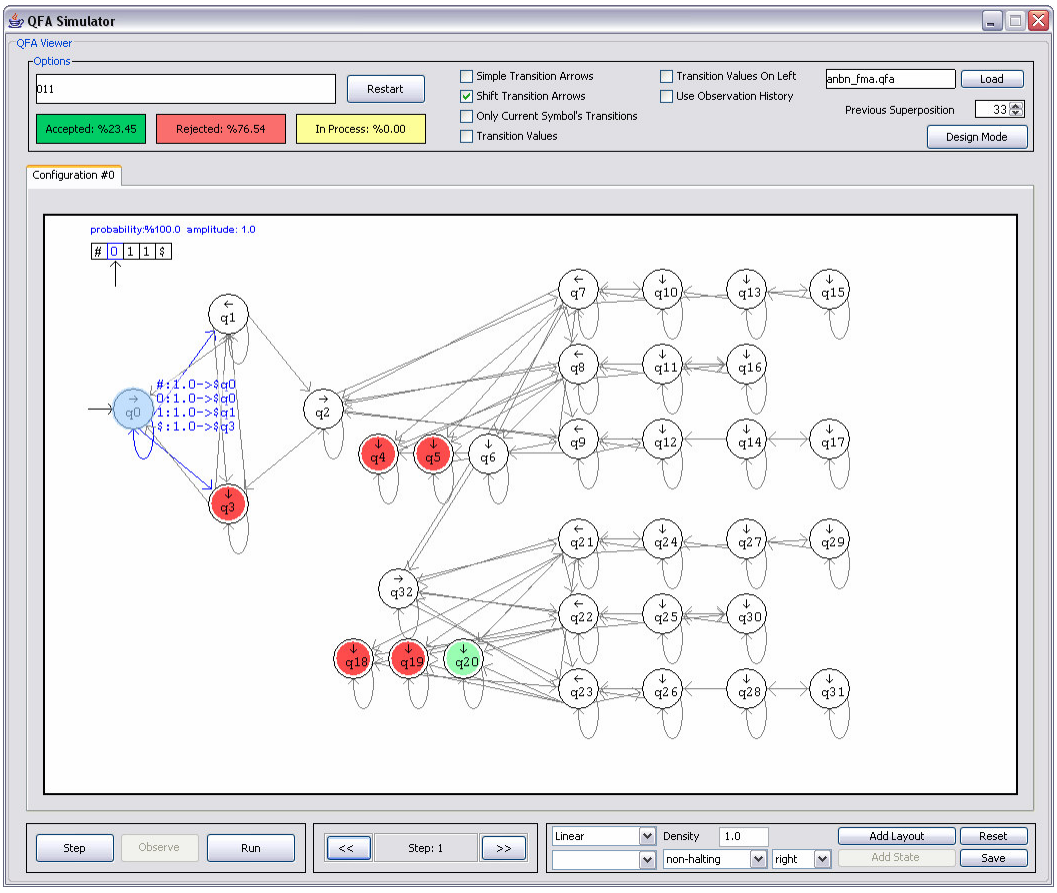

Figure 5.4: Proposed Construction

After building the new 2QFA according to our newly proposed construction, we see that, the wrong accept ratio was not near that expectation. We were expecting 1/9 (11.11%) but the outcome is 23.45%. The reason is, once again, un-intended cancellations of the threads in second layer.

In the first layer, the construction is set up so that all threads will arrive at different steps to the last branching state since the number of 0s and 1s are different. But in the

second layer, since they split into 3 again, some of the faster threads on stage 1 catch some of the slower ones on stage 2.

To be more specific, let's say that the difference between the number of 0s and 1s in the input string is k. If k is 0, then the input will be accepted with 100% probability. So, there is no problem for k=0. If k > 0 then the threads will finish the processing of the input at the same step. Let us say, the fastest thread comes to the start of the tape at M steps, the second fastest will come at M + k steps, and the third one in M+2k steps. For example, for the input 0111, where k=2, the first thread proceeds to the second level at the 18^{th} step, the second one at the 20^{th} step and the third one at the 22^{nd} step. In the second level, each thread will again split into three different threads and will process through the input again. The second level will take M steps for the fastest thread. And the other two threads will process M+k and M+2k steps. At this point one can see that, the slowest thread of level 2 which is split from the fastest thread of level 1 will take M+(M+2k) steps to complete. And the second fastest thread in stage 2, which is split from the second fastest thread in stage 1 will take (M+k) + (M+k) = M+M+2k steps. These two numbers are equal. That means they interfere each other at stage 2, since both arrive to the end of the input at the same time.

Rather than try to fix this machine using the "spin wait" method described earlier, we adopted a new architecture based on the machine we built to recognize the language $\{0^n10^n \mid n > 0\}$ for building a different 2QFA for $\{0^n1^n \mid n > 0\}$.

When the machine in Figure 5.5 runs, following the first phase of checking for 0*1*, in each level (three levels in this case), it splits into two threads at the rightmost occurrence of symbol 0. One thread processes over 0s and the other thread processes over 1s. When they reach the ends of the string, they turn back. And at the rightmost occurrence of symbol 0 on input they perform a QFT. If the number of 0s and 1s are same, then they reach that point at the same step, and reject state will not be reached. So, it will process thru the next level without being rejected. But if the number of 0s and 1s are different, then threads will reach at different steps, and a total of 50% will be rejected. The un-rejected amplitude will continue to the level 2. Since the threads could not match at the middle, they will not join, so there will be 2 threads beginning processing on level 2. For keeping things simple, we will name the threads by the substrings they walk on; L means the left

substring (consisting of 0s) and R means the right substring (consisting of 1s). Let us call the lengths of these substrings l and r, respectively. So, L runs for 2l steps, and R runs for 2r steps. When L passes to the second level, it again splits into two, LL and LR. Similarly R splits into two, RL and RR. Since we do not want LR and RL (or, indeed, any pair) to meet each other when $l \neq r$, what we did in the machine of Figure 5.5 to solve this problem is to change the number of steps processed in different levels, so they will not be equal, therefore they will not cancel each other. To change the number of steps required, we choose to slow down the process in the second pass of the tape in second level. So, when a thread processes to level two, it will process both sides of the tape towards the ends of the tape as usual, but while returning the tape head to the center of the tape, the tape head will move one square after two steps. In case of a legal input, since both sides are slowed down equally, they will still meet at the correct step. And in case of an illegal input, LR will take 2l + 3r steps, and RL will take 2r + 3l steps. So, they will never meet unless l = r. When they reach to the right most 0 symbol, the threads again split into two, and begin processing on level three. These threads are named LRL, LRR, RLL, RLR, RRR, RRL, LLR, and LLL. In the third level, there are two spin wait states. So, when a thread processes to level three, it will process both sides of the tape towards the ends of the tape as usual, but while returning the tape head to the center of the tape, the tape head will move one square after three steps. Again a legal input will still be accepted since both threads are slowed down equally. Table 5.10 shows the number of steps taken by each thread at the end of level three:

Table 5.10: Steps per thread

Threads	**Steps**
LLL	9l
LLR	5l + 4r
LRL	6l + 3r
LRR	2l + 7r
RLL	7l + 2r
RLR	3l + 6r
RRL	4l + 5r
RRR	9l

As it is seen from Table 5.10, any two threads can not meet each other unless l = r, which implies that the input string is actually an element of the language.

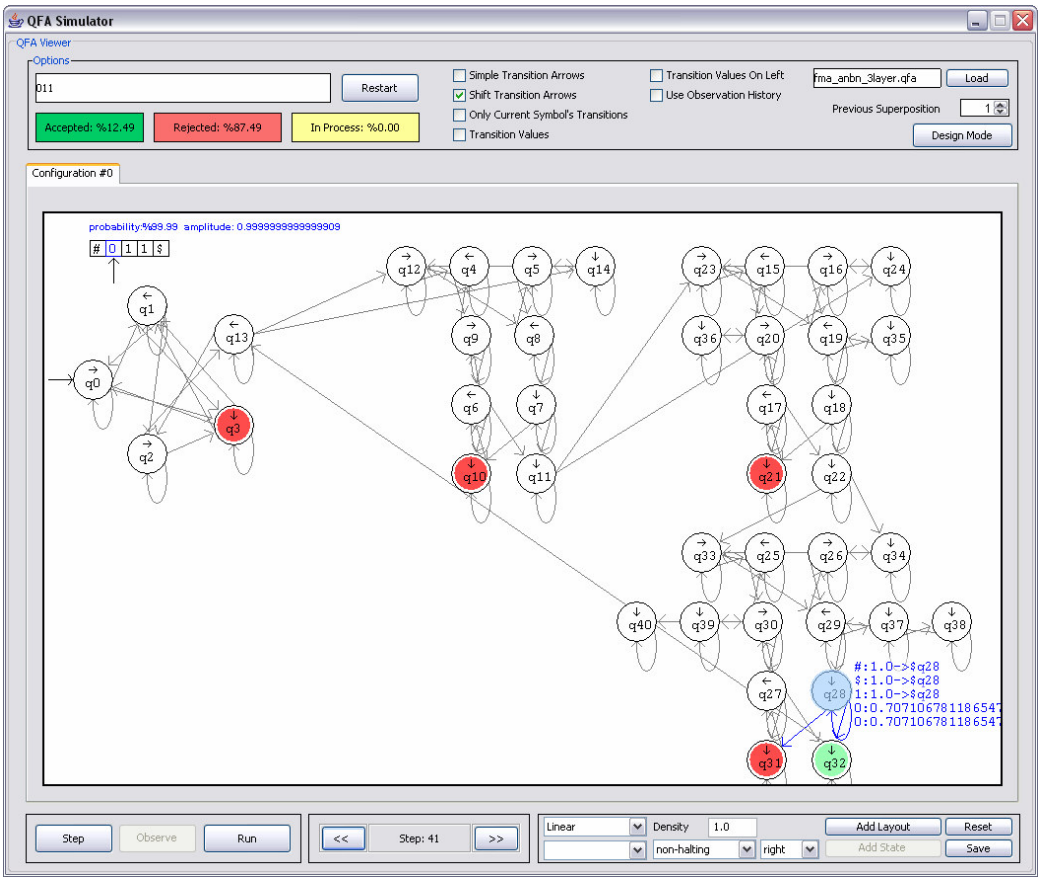

Figure 5.5: Our final 3 level implementation of the machine for recognizing the language $\{0^n1^n \mid n \geq 0\}$.

For detailed information about the transition matrices, please refer to Appendix B.

With this new construction, we prevented the unintended cancellations. So, the false accept ratio is 12.50% (1/8). At each stage 1/2 is rejected.

Now, let us compare this result with the previous machines in the literature. We can build a machine having comparable size and reliability using the parametric construction with N=7. The machine will be similar to Figure 5.6.

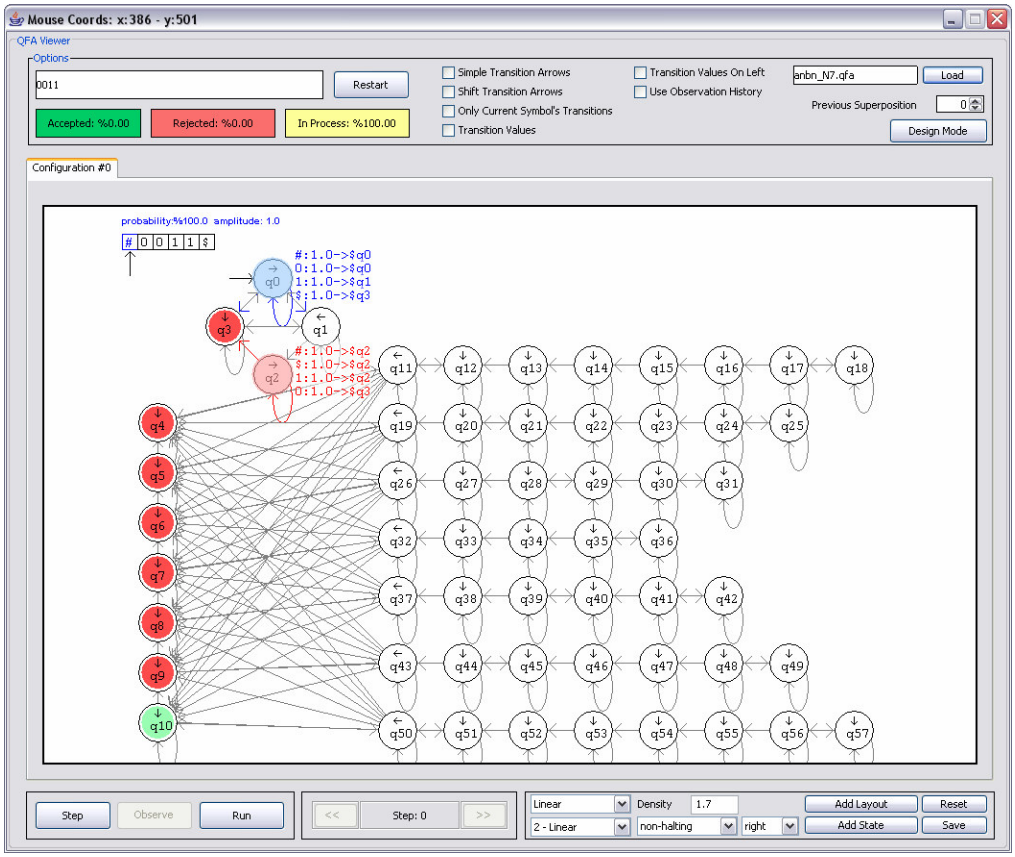

Figure 5.6: A sample construction of the parametric machine defined in [1] for N=7.

In this machine, the false accept ratio is 1/N=1/7 (~0.14285714). And this machine has a total of 58 states. And our previous machine of Figure 5.5 has 41 states, and a false accept ratio of 0.125.

Here is a detailed comparison table of characteristics of our machine, this machine, and the machine defined in [1] for N=8. We will not present the details of the machine for N=8 since it is very similar to the machine for N=7.

Table 5.11: State comparison table

	Our Machine	N=7	N=8
Number Of States	41	58	72
Probability Of False Accept	0.125	0.143	0.125

Table 5.12: Runtime comparison table

	011			0011111111			00000000011			00000000000000000000000000011		
	A	B	C	A	B	C	A	B	C	A	B	C
Steps	42	26	28	109	83	91	113	92	111	283	245	271

A: Machine in Figure 5.5

B: Machine for N = 7

C: Machine for N = 8

As it is seen from the tables 5.11 and 5.12, our construction has the least number of states, but the error ratio is better than the error ratio of the machine for N=7 and equal to the error ratio of the machine for N=8. Their runtime characteristics are similar, but our machine performs about 3% to 15% more steps depending on the input. That means, for this specific language, our solution is an enhancement over the existing solutions in terms of state complexity, without sacrificing runtime complexity too much.

5.4. Important Points

Usually for this construction to work, we will need some interesting properties. Our 2QFA should not suffer from the cancellations. It should have relatively small number of states to append its copy, so that we will not need to add too many states. Adding another copy to the newly added copies is also possible but mathematical analysis will get much difficult if the cancellations are in place. And to get better results, we may need to make some slight changes in the construction.

6. CONCLUSION

In this work, we have inspected finite and quantum automata. To get a better understanding of quantum automata, we designed and implemented a 2 way quantum finite state accepter simulator. We have implemented many languages from the literature using the simulator. We also implemented some languages, which were referenced in literature, but not implemented.

On the other hand, we have studied on machine design. We designed some machines for some non-regular languages which were not in literature previously. We also developed a method for enhancing the complexity characteristics of some class of 2QFAs. Our enhancement allows some specific 2QFAs which are initially useless as bounded error accepters, to be used as bounded error accepters.

6.1. Future Work

There are some ideas we thought during the research, but we did not implemented because of time constraints. These could be developed in future work.

- The duplication process may be optimized depending on the machine. For example one can alter the duplicated copy so that it processes the tape backwards, so we will not need a state to rewind the tape head. That will also save processing time.
- There are some other machines which we have implemented but did not inspect in detail since they were similar to the previous ones, like for the languages $\{0^n1^{n+k} \mid n > 0\}$, $\{0^n1^{n-k} \mid n > 0\}$, $\{0^n1^{n+k}2^{n+l} \mid n > 0\}$… etc. These machines could be inspected in more detail.
- Addition of spin wait states could be changed. Maybe instead of changing the increasing the runtime related to length of the input, we could add some constant factor. This may or may not be possible, before making a final decision, one should inspect the side effects.
- There is also a completely different family of 2QFAs, which we worked on which we did not mention in this thesis. These class of 2QFAs will only accept

or reject an input with 100% probability. But they can enter infinite loops, or response time may take too long. These class of machines will have an expected runtime for an input, but they are not bounded. These machines may be inspected in future.

- The enhancement method we mentioned, could be applied to quantum Turing machines also. That would require much more complex analysis since the input tape is not read only, but it may lead to some improvements over some existing machines.
- The simulator software may be optimized for better performance. For the time being, clear code took precedence on optimized code.
- Some user friendly features may be added to the simulator software, so that designing and simulating 2QFAs will be easier. For example, auto-completion of the unitary matrices may be added so one will not need to specify the transitions that would not be used, but required for satisfying unitarity.
- The simulator software may be developed so that it will also support QTMs and other types of devices, not only 2QFAs.

APPENDIX A

Transition Amplitudes of the machines

Table A.1. Machine 1: (Machine for equal number of 0s and 1s)

Transition Matrix for Symbol **#**

#	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14
q0					0.577	0.577	0.577								
q1		1.0													
q2			1.0												
q3				1.0											
q4	0.577				0.577	-0.577									
q5	0.577					0.577	-0.577								
q6	0.577				-0.577		0.577								
q7								1.0							
q8									1.0						
q9										1.0					
q10											1.0				
q11												1.0			
q12													1.0		
q13														1.0	
q14															1.0

Transition Matrix for Symbol **0**

0	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14
q0	1.0														
q1		1.0													
q2			1.0												
q3				1.0											
q4								1.0							
q5														1.0	
q6															1.0
q7					1.0										
q8						1.0									
q9							1.0								
q10											1.0				
q11										1.0					
q12													1.0		
q13									1.0						
q14												1.0			

Transition Matrix for Symbol **1**

1	q0	q1	q2	q3	q4	q5	q6	Q7	q8	q9	q10	q11	q12	q13	q14
q0	1.0														
q1		1.0													
q2			1.0												
q3				1.0											
q4													1.0		
q5														1.0	
q6										1.0					
q7					1.0										
q8						1.0									
q9							1.0								
q10								1.0							
q11												1.0			
q12											1.0				
q13									1.0						
q14															1.0

Transition Matrix for Symbol **$**

$	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14
q0	1.0														
q1						1.0									
q2							1.0								
q3					1.0										
q4		-0.289 + i * 0.5	-0.289 - i * 0.5	0.577											
q5		-0.289 - i * 0.5	-0.289 + i * 0.5	0.577											
q6		0.577	0.577	0.577											
q7								1.0							
q8									1.0						
q9										1.0					
q10											1.0				
q11												1.0			
q12													1.0		
q13														1.0	
q14															1.0

Table A.2. Machine 2: (Machine for language $\{0^n1^n \mid n > 0\}$)

Transition Matrix for Symbol **#**

#	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17
q0	1.0																	
q1				1.0														
q2			1.0															
q3		1.0																
q4									1.0									
q5										1.0								
q6								1.0										
q7					-0.289 +i* 0.5	-0.289 -i*0.5	0.577											
q8					-0.289 -i*0.5	-0.289 +i*0.5	0.577											
q9					0.577	0.577	0.577											
q10											1.0							
q11												1.0						
q12													1.0					
q13														1.0				
q14															1.0			
q15																1.0		
q16																	1.0	
q17																		1.0

Transition Matrix for Symbol **0**

0	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17
q0	1.0																	
q1			1.0															
q2				1.0														
q3		1.0																
q4					1.0													
q5						1.0												
q6							1.0											
q7											1.0							
q8																	1.0	
q9																		1.0
q10								1.0										
q11									1.0									
q12										1.0								
q13													1.0					
q14												1.0						
q15															1.0			
q16											1.0							
q17														1.0				

Transition Matrix for Symbol **1**

1	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17
q0		1.0																
q1	1.0																	
q2			1.0															
q3				1.0														
q4					1.0													
q5						1.0												
q6							1.0											
q7																1.0		
q8																	1.0	
q9													1.0					
q10								1.0										
q11									1.0									
q12										1.0								
q13											1.0							
q14															1.0			
q15														1.0				
q16												1.0						
q17																		1.0

Transition Matrix for Symbol **$**

$	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17
q0				1.0														
q1		1.0																
q2								0.577	0.577	0.577								
q3	1.0																	
q4					1.0													
q5						1.0												
q6							1.0											
q7			0.577					0.577	-0.577									
q8			0.577						0.577	-0.577								
q9			0.577					-0.577		0.577								
q10											1.0							
q11												1.0						
q12													1.0					
q13														1.0				
q14															1.0			
q15																1.0		
q16																	1.0	
q17																		1.0

Table A.3. Machine 3: (Machine for language $\{0^n1^n2^n \mid n > 0\}$)

Transition Matrix for Symbol #

#	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16
q0	1.0																
q1				1.0													
q2			1.0														
q3		1.0															
q4																	
q5																	
q6							1.0										
q7								1.0									
q8									1.0								
q9										1.0							
q10											1.0						
q11												1.0					
q12													1.0				
q13														1.0			
q14															1.0		
q15																1.0	
q16																	1.0
q17																	
q18																	
q19																	
q20																	
q21					-0.289 +i*0.5	-0.289 -i*0.5											
q22					-0.289 -i*0.5	-0.289 +i*0.5											
q23					0.577	0.577											
q24																	
q25																	
q26																	
q27																	
q28																	
q29																	
q30																	
q31																	
q32																	
q33																	
q34																	

Continuation of Transition Matrix for Symbol #

#	q17	q18	q19	q20	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34
q0																		
q1																		
q2																		
q3																		
q4					1.0													
q5						1.0												
q6																		
q7																		
q8																		
q9																		
q10																		
q11																		
q12																		
q13																		
q14																		
q15																		
q16																		
q17	1.0																	
q18		1.0																
q19			1.0															
q20				1.0														
q21																		0.577
q22																		0.577
q23																		0.577
q24								1.0										
q25									1.0									
q26										1.0								
q27											1.0							
q28												1.0						
q29													1.0					
q30														1.0				
q31															1.0			
q32																1.0		
q33																	1.0	
q34							1.0											

Transition Matrix for Symbol **0**

0	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16
q0	1.0																
q1			1.0														
q2				1.0													
q3		1.0															
q4										1.0							
q5									1.0								
q6																	
q7					-0.289 +i*0.5	-0.289 -i*0.5	0.577										
q8					-0.289 -i*0.5	-0.289 +i*0.5	0.577										
q9					0.577	0.577	0.577										
q10											1.0						
q11												1.0					
q12													1.0				
q13														1.0			
q14															1.0		
q15																1.0	
q16																	1.0
q17																	
q18																	
q19																	
q20																	
q21																	
q22																	
q23																	
q24																	
q25																	
q26																	
q27																	
q28																	
q29																	
q30																	
q31																	
q32								1.0									
q33																	
q34																	

Continuation of Transition Matrix for Symbol **0**

0	q17	q18	q19	q20	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34
q0																		
q1																		
q2																		
q3																		
q4																		
q5																		
q6																1.0		
q7																		
q8																		
q9																		
q10																		
q11																		
q12																		
q13																		
q14																		
q15																		
q16																		
q17	1.0																	
q18		1.0																
q19				1.0														
q20			1.0															
q21								1.0										
q22														1.0				
q23															1.0			
q24					1.0													
q25						1.0												
q26							1.0											
q27											1.0							
q28										1.0								
q29													1.0					
q30									1.0									
q31												1.0						
q32																		
q33																	1.0	
q34																		1.0

Transition Matrix for Symbol **1**

1	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16
q0		1.0															
q1	1.0																
q2			1.0														
q3				1.0													
q4					1.0												
q5						1.0											
q6							1.0										
q7											1.0						
q8																	1.0
q9																	
q10								1.0									
q11									1.0								
q12										1.0							
q13														1.0			
q14													1.0				
q15																1.0	
q16												1.0					
q17															1.0		
q18																	
q19																	
q20																	
q21																	
q22																	
q23																	
q24																	
q25																	
q26																	
q27																	
q28																	
q29																	
q30																	
q31																	
q32																	
q33																	
q34																	

Continuation of Transition Matrix for Symbol **1**

1	q17	q18	q19	q20	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34
q0																		
q1																		
q2																		
q3																		
q4																		
q5																		
q6																		
q7																		
q8																		
q9	1.0																	
q10																		
q11																		
q12																		
q13																		
q14																		
q15																		
q16																		
q17																		
q18			1.0															
q19				1.0														
q20		1.0																
q21													1.0					
q22														1.0				
q23										1.0								
q24					1.0													
q25						1.0												
q26							1.0											
q27								1.0										
q28												1.0						
q29											1.0							
q30									1.0									
q31															1.0			
q32																1.0		
q33																	1.0	
q34																		1.0

Transition Matrix for Symbol **2**

2	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17
q0				1.0														
q1		1.0																
q2																		
q3	1.0																	
q4					1.0													
q5						1.0												
q6							1.0											
q7																1.0		
q8																	1.0	
q9													1.0					
q10								1.0										
q11									1.0									
q12										1.0								
q13											1.0							
q14															1.0			
q15														1.0				
q16												1.0						
q17																		1.0
q18			1.0															
q19																		
q20																		
q21																		
q22																		
q23																		
q24																		
q25																		
q26																		
q27																		
q28																		
q29																		
q30																		
q31																		
q32																		
q33																		
q34																		

Continuation of Transition Matrix for Symbol **2**

2	q18	q19	q20	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34
q0																	
q1																	
q2	1.0																
q3																	
q4																	
q5																	
q6																	
q7																	
q8																	
q9																	
q10																	
q11																	
q12																	
q13																	
q14																	
q15																	
q16																	
q17																	
q18																	
q19		1.0															
q20			1.0														
q21				0.577	-0.577										0.577		
q22					0.577	-0.577									0.577		
q23				-0.577		0.577									0.577		
q24							1.0										
q25								1.0									
q26									1.0								
q27										1.0							
q28											1.0						
q29												1.0					
q30													1.0				
q31														1.0			
q32																1.0	
q33				0.577	0.577	0.577											
q34																	1.0

Transition Matrix for Symbol $

$	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16
q0				1.0													
q1		1.0															
q2																	
q3	1.0																
q4					1.0												
q5						1.0											
q6							1.0										
q7								0.577	-0.577								
q8									0.577	-0.577							
q9								-0.577		0.577							
q10											1.0						
q11												1.0					
q12													1.0				
q13														1.0			
q14															1.0		
q15																1.0	
q16																	1.0
q17																	
q18																	
q19								0.577	0.577	0.577							
q20			1.0														
q21																	
q22																	
q23																	
q24																	
q25																	
q26																	
q27																	
q28																	
q29																	
q30																	
q31																	
q32																	
q33																	
q34																	

Continuation of Transition Matrix for Symbol **$**

$	q17	q18	q19	q20	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34
q0																		
q1																		
q2				1.0														
q3																		
q4																		
q5																		
q6																		
q7			0.577															
q8			0.577															
q9			0.577															
q10																		
q11																		
q12																		
q13																		
q14																		
q15																		
q16																		
q17	1.0																	
q18		1.0																
q19																		
q20																		
q21					1.0													
q22						1.0												
q23							1.0											
q24								1.0										
q25									1.0									
q26										1.0								
q27											1.0							
q28												1.0						
q29													1.0					
q30														1.0				
q31															1.0			
q32																1.0		
q33																	1.0	
q34																		1.0

Table A.4. Machine 4: Machine for the non-context free language, $\{0^n1^{2n} \mid n > 0\}$.

Transition Matrix for Symbol #

#	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20	q21	q22
q0	1.0																						
q1				1.0																			
q2			1.0																				
q3		1.0																					
q4									1.0														
q5										1.0													
q6								1.0															
q7					-0.289 + i * 0.5	-0.289 - i * 0.5	0.577																
q8					-0.289 - i * 0.5	-0.289 + i * 0.5	0.577																
q9					0.577	0.577	0.577																
q10											1.0												
q11												1.0											
q12													1.0										
q13														1.0									
q14															1.0								
q15																1.0							
q16																	1.0						
q17																		1.0					
q18																			1.0				
q19																				1.0			
q20																					1.0		
q21																						1.0	
q22																							1.0

Transition Matrix for Symbol **0**

0	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20	q21	q22
q0	1.0																						
q1			1.0																				
q2				1.0																			
q3		1.0																					
q4					1.0																		
q5						1.0																	
q6							1.0																
q7														1.0									
q8																				1.0			
q9																							1.0
q10								1.0															
q11									1.0														
q12										1.0													
q13											1.0												
q14													1.0										
q15																1.0							
q16												1.0											
q17															1.0								
q18																	1.0						
q19																			1.0				
q20																		1.0					
q21																					1.0		
q22																						1.0	

Transition Matrix for Symbol **1**

1	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20	q21	q22
q0		1.0																					
q1	1.0																						
q2			1.0																				
q3				1.0																			
q4					1.0																		
q5						1.0																	
q6							1.0																
q7																1.0							
q8																	1.0						
q9													1.0										
q10								1.0															
q11									1.0														
q12										1.0													
q13											1.0												
q14															1.0								
q15														1.0									
q16												1.0											
q17																		1.0					
q18																			1.0				
q19																				1.0			
q20																					1.0		
q21																						1.0	
q22																							1.0

Transition Matrix for Symbol $

$	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20	q21	q22
q0				1.0																			
q1		1.0																					
q2								0.577	0.577	0.577													
q3	1.0																						
q4					1.0																		
q5						1.0																	
q6							1.0																
q7			0.577					0.577	-0.577														
q8			0.577						0.577	-0.577													
q9			0.577					-0.577		0.577													
q10											1.0												
q11												1.0											
q12													1.0										
q13														1.0									
q14															1.0								
q15																1.0							
q16																	1.0						
q17																		1.0					
q18																			1.0				
q19																				1.0			
q20																					1.0		
q21																						1.0	
q22																							1.0

Table A.5. Machine 5: Machine for the non-context free language, $\{0^n1^n2^{2n} \mid n > 0\}$.

Transition Matrix for Symbol #

#	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	Q15	q16	q17	q18	q19
q0	1.0																			
q1				1.0																
q2			1.0																	
q3		1.0																		
q4																				
q5																				
q6							1.0													
q7								1.0												
q8									1.0											
q9										1.0										
q10											1.0									
q11												1.0								
q12													1.0							
q13														1.0						
q14															1.0					
q15																1.0				
q16																	1.0			
q17																		1.0		
q18																			1.0	
q19																				1.0
q20																				
q21					-0.289 +i*0.5	-0.289 -i*0.5														
q22					-0.289 -i*0.5	-0.289 +i*0.5														
q23					0.577	0.577														
q24																				
q25																				
q26																				
q27																				
q28																				
q29																				
q30																				
q31																				
q32																				
q33																				
q34																				
q35																				
q36																				
q37																				
q38																				
q39																				

Continuation of Transition Matrix for Symbol #

#	q20	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39
q0																				
q1																				
q2																				
q3																				
q4		1.0																		
q5			1.0																	
q6																				
q7																				
q8																				
q9																				
q10																				
q11																				
q12																				
q13																				
q14																				
q15																				
q16																				
q17																				
q18																				
q19																				
q20	1.0																			
q21															0.577					
q22															0.577					
q23															0.577					
q24					1.0															
q25						1.0														
q26							1.0													
q27								1.0												
q28									1.0											
q29										1.0										
q30											1.0									
q31												1.0								
q32													1.0							
q33														1.0						
q34				1.0																
q35																1.0				
q36																	1.0			
q37																		1.0		
q38																			1.0	
q39																				1.0

Transition Matrix for Symbol **0**

0	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19
q0	1.0																			
q1			1.0																	
q2				1.0																
q3		1.0																		
q4										1.0										
q5									1.0											
q6																				
q7					-0.289 +i*0.5	-0.289 -i*0.5	0.577													
q8					-0.289 -i*0.5	-0.289 +i*0.5	0.577													
q9					0.577	0.577	0.577													
q10											1.0									
q11												1.0								
q12													1.0							
q13														1.0						
q14															1.0					
q15																1.0				
q16																	1.0			
q17																		1.0		
q18																			1.0	
q19																				
q20																				1.0
q21																				
q22																				
q23																				
q24																				
q25																				
q26																				
q27																				
q28																				
q29																				
q30																				
q31																				
q32								1.0												
q33																				
q34																				
q35																				
q36																				
q37																				
q38																				
q39																				

Continuation of Transition Matrix for Symbol 0

0	q20	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39
q0																				
q1																				
q2																				
q3																				
q4																				
q5																				
q6													1.0							
q7																				
q8																				
q9																				
q10																				
q11																				
q12																				
q13																				
q14																				
q15																				
q16																				
q17																				
q18																				
q19	1.0																			
q20																				
q21					1.0															
q22											1.0									
q23												1.0								
q24		1.0																		
q25			1.0																	
q26				1.0																
q27								1.0												
q28							1.0													
q29										1.0										
q30						1.0														
q31									1.0											
q32																				
q33														1.0						
q34															1.0					
q35																1.0				
q36																	1.0			
q37																		1.0		
q38																			1.0	
q39																				1.0

Transition Matrix for Symbol **1**

1	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19
q0		1.0																		
q1	1.0																			
q2			1.0																	
q3				1.0																
q4					1.0															
q5						1.0														
q6							1.0													
q7														1.0						
q8																				
q9																				
q10								1.0												
q11									1.0											
q12										1.0										
q13											1.0									
q14													1.0							
q15																1.0				
q16												1.0								
q17															1.0					
q18																				1.0
q19																				
q20																			1.0	
q21																				
q22																				
q23																				
q24																				
q25																				
q26																				
q27																				
q28																				
q29																				
q30																				
q31																				
q32																				
q33																				
q34																				
q35																	1.0			
q36																				
q37																		1.0		
q38																				
q39																				

Continuation of Transition Matrix for Symbol **1**

1	q20	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39
q0																				
q1																				
q2																				
q3																				
q4																				
q5																				
q6																				
q7																				
q8																	1.0			
q9																				1.0
q10																				
q11																				
q12																				
q13																				
q14																				
q15																				
q16																				
q17																				
q18																				
q19	1.0																			
q20																				
q21										1.0										
q22											1.0									
q23							1.0													
q24		1.0																		
q25			1.0																	
q26				1.0																
q27					1.0															
q28									1.0											
q29								1.0												
q30						1.0														
q31												1.0								
q32													1.0							
q33														1.0						
q34															1.0					
q35																				
q36																1.0				
q37																				
q38																		1.0		
q39																			1.0	

Transition Matrix for Symbol **2**

2	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19
q0				1.0																
q1		1.0																		
q2																			1.0	
q3	1.0																			
q4					1.0															
q5						1.0														
q6							1.0													
q7																1.0				
q8																	1.0			
q9													1.0							
q10								1.0												
q11									1.0											
q12										1.0										
q13											1.0									
q14															1.0					
q15														1.0						
q16												1.0								
q17																		1.0		
q18			1.0																	
q19																				1.0
q20																				
q21																				
q22																				
q23																				
q24																				
q25																				
q26																				
q27																				
q28																				
q29																				
q30																				
q31																				
q32																				
q33																				
q34																				
q35																				
q36																				
q37																				
q38																				
q39																				

Continuation of Transition Matrix for Symbol **2**

2	q20	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39
q0																				
q1																				
q2																				
q3																				
q4																				
q5																				
q6																				
q7																				
q8																				
q9																				
q10																				
q11																				
q12																				
q13																				
q14																				
q15																				
q16																				
q17																				
q18																				
q19																				
q20	1.0																			
q21		0.577	-0.577										0.577							
q22			0.577	-0.577									0.577							
q23		-0.577		0.577									0.577							
q24					1.0															
q25						1.0														
q26							1.0													
q27								1.0												
q28									1.0											
q29										1.0										
q30											1.0									
q31												1.0								
q32														1.0						
q33		0.577	0.577	0.577																
q34															1.0					
q35																1.0				
q36																	1.0			
q37																		1.0		
q38																			1.0	
q39																				1.0

Transition Matrix for Symbol **$**

$	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19
q0				1.0																
q1		1.0																		
q2																				
q3	1.0																			
q4					1.0															
q5						1.0														
q6							1.0													
q7								0.577	-0.577											0.577
q8									0.577	-0.577										0.577
q9								-0.577		0.577										0.577
q10											1.0									
q11												1.0								
q12													1.0							
q13														1.0						
q14															1.0					
q15																1.0				
q16																	1.0			
q17																		1.0		
q18																			1.0	
q19								0.577	0.577	0.577										
q20			1.0																	
q21																				
q22																				
q23																				
q24																				
q25																				
q26																				
q27																				
q28																				
q29																				
q30																				
q31																				
q32																				
q33																				
q34																				
q35																				
q36																				
q37																				
q38																				
q39																				

Continuation of Transition Matrix for Symbol **$**

$	q20	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39
q0																				
q1																				
q2	1.0																			
q3																				
q4																				
q5																				
q6																				
q7																				
q8																				
q9																				
q10																				
q11																				
q12																				
q13																				
q14																				
q15																				
q16																				
q17																				
q18																				
q19																				
q20																				
q21		1.0																		
q22			1.0																	
q23				1.0																
q24					1.0															
q25						1.0														
q26							1.0													
q27								1.0												
q28									1.0											
q29										1.0										
q30											1.0									
q31												1.0								
q32													1.0							
q33														1.0						
q34															1.0					
q35																1.0				
q36																	1.0			
q37																		1.0		
q38																			1.0	
q39																				1.0

Table A.6. Machine 6: (Machine for the language $\{0^n10^n \mid n > 0\}$)

Transition Matrix for Symbol **#**

#	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15
q0	1.0															
q1		1.0														
q2			1.0													
q3				1.0												
q4						1.0										
q5					1.0											
q6							1.0									
q7								1.0								
q8										1.0						
q9									1.0							
q10											1.0					
q11												1.0				
q12														1.0		
q13													1.0			
q14															1.0	
q15																1.0

Transition Matrix for Symbol **0**

0	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15
q0	1.0															
q1		1.0														
q2			1.0													
q3				1.0												
q4					1.0											
q5						1.0										
q6							1.0									
q7								1.0								
q8									1.0							
q9										1.0						
q10											1.0					
q11												1.0				
q12													1.0			
q13														1.0		
q14															1.0	
q15																1.0

Transition Matrix for Symbol **1**

1	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15
q0		1.0														
q1			1.0													
q2	1.0															
q3					0.707	0.707										
q4							0.707	0.707								
q5							-0.707	0.707								
q6					-0.707	0.707										
q7									0.707	0.707						
q8											0.707	0.707				
q9											-0.707	0.707				
q10									-0.707	0.707						
q11													0.707	0.707		
q12															0.707	0.707
q13															-0.707	0.707
q14													-0.707	0.707		
q15				1.0												

Transition Matrix for Symbol **$**

$	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15
q0	1.0															
q1				1.0												
q2			1.0													
q3		1.0														
q4						1.0										
q5					1.0											
q6							1.0									
q7								1.0								
q8										1.0						
q9									1.0							
q10											1.0					
q11												1.0				
q12														1.0		
q13													1.0			
q14															1.0	
q15																1.0

Table A.7. Machine 7: (Machine for the language $\{0^n1^{2n}2^{3n} \mid n > 0\}$)

Transition Matrix for Symbol #

#	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20
q0	1.0																				
q1				1.0																	
q2			1.0																		
q3		1.0																			
q4										1.0											
q5									1.0												
q6																					
q7					-0.289+ i*0.5	-0.289 - i*0.5	0.577														
q8					-0.289 - i*0.5	-0.289 + i*0.5	0.577														
q9					0.577	0.577	0.577														
q10											1.0										
q11												1.0									
q12													1.0								
q13														1.0							
q14															1.0						
q15																1.0					
q16																	1.0				
q17																		1.0			
q18																			1.0		
q19																				1.0	
q20																					1.0
q21																					
q22																					
q23																					
q24																					
q25																					
q26																					
q27																					
q28																					
q29																					
q30																					
q31																					
q32								1.0													
q33																					
q34																					
q35																					
q36																					
q37																					
q38																					
q39																					
q40																					
q41																					

Continuation of Transition Matrix for Symbol #

#	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39	q40	q41
q0																					
q1																					
q2																					
q3																					
q4																					
q5																					
q6												1.0									
q7																					
q8																					
q9																					
q10																					
q11																					
q12																					
q13																					
q14																					
q15																					
q16																					
q17																					
q18																					
q19																					
q20																					
q21														0.577						-0.289+i*0.5	-0.289-i*0.5
q22														0.577						-0.289-i*0.5	-0.289+i*0.5
q23														0.577						0.577	0.577
q24				1.0																	
q25					1.0																
q26						1.0															
q27							1.0														
q28								1.0													
q29									1.0												
q30										1.0											
q31											1.0										
q32																					
q33													1.0								
q34			1.0																		
q35															1.0						
q36																1.0					
q37																	1.0				
q38																		1.0			
q39																			1.0		
q40	1.0																				
q41		1.0																			

Transition Matrix for Symbol **0**

0	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20	q21
q0	1.0																					
q1			1.0																			
q2				1.0																		
q3		1.0																				
q4					1.0																	
q5						1.0																
q6							1.0															
q7											1.0											
q8																	1.0					
q9																		1.0				
q10								1.0														
q11									1.0													
q12										1.0												
q13														1.0								
q14													1.0									
q15																1.0						
q16												1.0										
q17															1.0							
q18																			1.0			
q19																					1.0	
q20																				1.0		
q21																						
q22																						
q23																						
q24																						1.0
q25																						
q26																						
q27																						
q28																						
q29																						
q30																						
q31																						
q32																						
q33																						
q34																						
q35																						
q36																						
q37																						
q38																						
q39																						
q40																						
q41																						

Continuation of Transition Matrix for Symbol **0**

0	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39	q40	q41
q0																				
q1																				
q2																				
q3																				
q4																				
q5																				
q6																				
q7																				
q8																				
q9																				
q10																				
q11																				
q12																				
q13																				
q14																				
q15																				
q16																				
q17																				
q18																				
q19																				
q20																				
q21						1.0														
q22															1.0					
q23																		1.0		
q24																				
q25	1.0																			
q26		1.0																		
q27			1.0																	
q28					1.0															
q29								1.0												
q30				1.0																
q31							1.0													
q32											1.0									
q33												1.0								
q34													1.0							
q35									1.0											
q36														1.0						
q37										1.0										
q38																1.0				
q39																	1.0			
q40																			1.0	
q41																				1.0

Transition Matrix for Symbol **1**

1	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20
q0		**1.0**																			
q1	**1.0**																				
q2			**1.0**																		
q3				**1.0**																	
q4					**1.0**																
q5						**1.0**															
q6							**1.0**														
q7											**1.0**										
q8																	**1.0**				
q9																		**1.0**			
q10								**1.0**													
q11									**1.0**												
q12										**1.0**											
q13														**1.0**							
q14													**1.0**								
q15																**1.0**					
q16												**1.0**									
q17															**1.0**						
q18																			**1.0**		
q19																					**1.0**
q20																			**1.0**		
q21																					
q22																					
q23																					
q24																					
q25																					
q26																					
q27																					
q28																					
q29																					
q30																					
q31																					
q32																					
q33																					
q34																					
q35																					
q36																					
q37																					
q38																					
q39																					
q40																					
q41																					

Continuation of Transition Matrix for Symbol #

1	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39	q40	q41
q0																					
q1																					
q2																					
q3																					
q4																					
q5																					
q6																					
q7																					
q8																					
q9																					
q10																					
q11																					
q12																					
q13																					
q14																					
q15																					
q16																					
q17																					
q18																					
q19																					
q20																					
q21									1.0												
q22										1.0											
q23						1.0															
q24	1.0																				
q25		1.0																			
q26			1.0																		
q27				1.0																	
q28								1.0													
q29							1.0														
q30					1.0																
q31											1.0										
q32												1.0									
q33													1.0								
q34														1.0							
q35															1.0						
q36																1.0					
q37																	1.0				
q38																		1.0			
q39																			1.0		
q40																				1.0	
q41																					1.0

Transition Matrix for Symbol **2**

2	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20
q0				1.0																	
q1		1.0																			
q2																			1.0		
q3	1.0																				
q4					1.0																
q5						1.0															
q6							1.0														
q7																1.0					
q8																	1.0				
q9													1.0								
q10								1.0													
q11									1.0												
q12										1.0											
q13											1.0										
q14															1.0						
q15														1.0							
q16												1.0									
q17																		1.0			
q18			1.0																		
q19																				1.0	
q20																					1.0
q21																					
q22																					
q23																					
q24																					
q25																					
q26																					
q27																					
q28																					
q29																					
q30																					
q31																					
q32																					
q33																					
q34																					
q35																					
q36																					
q37																					
q38																					
q39																					
q40																					
q41																					

Continuation of Transition Matrix for Symbol #

2	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39	q40	q41
q0																					
q1																					
q2																					
q3																					
q4																					
q5																					
q6																					
q7																					
q8																					
q9																					
q10																					
q11																					
q12																					
q13																					
q14																					
q15																					
q16																					
q17																					
q18																					
q19																					
q20																					
q21	0.577	-0.577										0.577									
q22		0.577	-0.577									0.577									
q23	-0.577		0.577									0.577									
q24				1.0																	
q25					1.0																
q26						1.0															
q27							1.0														
q28								1.0													
q29									1.0												
q30										1.0											
q31											1.0										
q32													1.0								
q33	0.577	0.577	0.577																		
q34														1.0							
q35															1.0						
q36																1.0					
q37																	1.0				
q38																		1.0			
q39																			1.0		
q40																				1.0	
q41																					1.0

Transition Matrix for Symbol **$**

$	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20
q0				1.0																	
q1		1.0																			
q2																					1.0
q3	1.0																				
q4					1.0																
q5						1.0															
q6							1.0														
q7								0.577	-0.577											0.577	
q8									0.577	-0.577										0.577	
q9								-0.577		0.577										0.577	
q10											1.0										
q11												1.0									
q12													1.0								
q13														1.0							
q14															1.0						
q15																1.0					
q16																	1.0				
q17																		1.0			
q18																			1.0		
q19								0.577	0.577	0.577											
q20			1.0																		
q21																					
q22																					
q23																					
q24																					
q25																					
q26																					
q27																					
q28																					
q29																					
q30																					
q31																					
q32																					
q33																					
q34																					
q35																					
q36																					
q37																					
q38																					
q39																					
q40																					
q41																					

Continuation of Transition Matrix for Symbol #

$	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39	q40	q41
q0																					
q1																					
q2																					
q3																					
q4																					
q5																					
q6																					
q7																					
q8																					
q9																					
q10																					
q11																					
q12																					
q13																					
q14																					
q15																					
q16																					
q17																					
q18																					
q19																					
q20																					
q21	1.0																				
q22		1.0																			
q23			1.0																		
q24				1.0																	
q25					1.0																
q26						1.0															
q27							1.0														
q28								1.0													
q29									1.0												
q30										1.0											
q31											1.0										
q32												1.0									
q33													1.0								
q34														1.0							
q35															1.0						
q36																1.0					
q37																	1.0				
q38																		1.0			
q39																			1.0		
q40																				1.0	
q41																					1.0

APPENDIX B

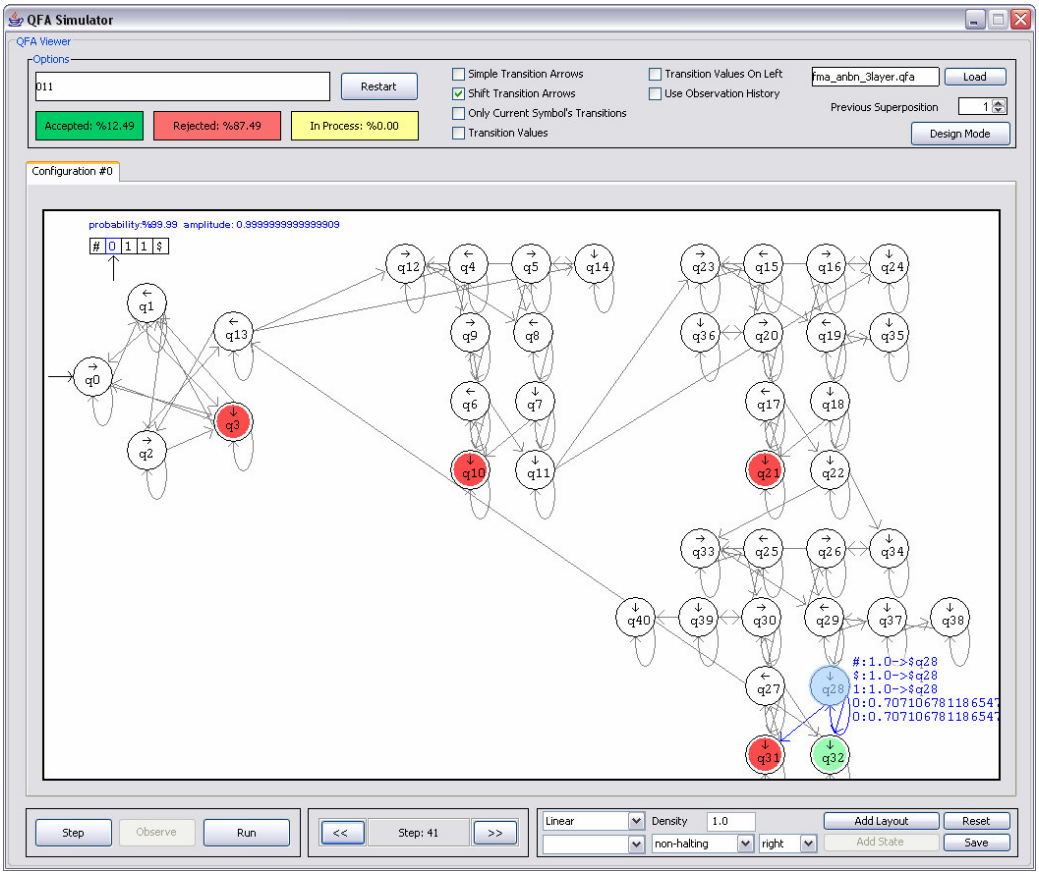

Figure B.1. State diagram of the 2QFA for language $\{0^n1^n \mid n > 0\}$

Table B.1. Transition Matrices of the 2QFA for language $\{0^n1^n \mid n > 0\}$:

Transition Matrix for Symbol #

#	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20
q0	1																				
q1				1																	
q2			1																		
q3		1																			
q4										1											
q5						1															
q6							1														
q7								1													
q8									1												
q9					1																
q10											1										
q11												1									
q12													1								
q13														1							
q14															1						
q15																					1
q16																	1				
q17																		1			
q18																			1		
q19																				1	
q20																1					
q21																					
q22																					
q23																					
q24																					
q25																					
q26																					
q27																					
q28																					
q29																					
q30																					
q31																					
q32																					
q33																					
q34																					
q35																					
q36																					
q37																					
q38																					
q39																					
q40																					

Continuation of Transition Matrix for Symbol #

#	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39	q40
q0																				
q1																				
q2																				
q3																				
q4																				
q5																				
q6																				
q7																				
q8																				
q9																				
q10																				
q11																				
q12																				
q13																				
q14																				
q15																				
q16																				
q17																				
q18																				
q19																				
q20																				
q21	1																			
q22		1																		
q23			1																	
q24				1																
q25										1										
q26						1														
q27							1													
q28								1												
q29									1											
q30					1															
q31											1									
q32												1								
q33													1							
q34														1						
q35															1					
q36																1				
q37																	1			
q38																		1		
q39																			1	
q40																				1

Transition Matrix for Symbol 0

0	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20
q0	1																				
q1			1																		
q2				1																	
q3		1																			
q4					1																
q5													$-2^{-1/2}$		$2^{-1/2}$						
q6											$-2^{-1/2}$	$2^{-1/2}$									
q7											$2^{-1/2}$	$2^{-1/2}$									
q8								1													
q9										1											
q10							1														
q11																					
q12									1												
q13													$2^{-1/2}$		$2^{-1/2}$						
q14						1															
q15																1					
q16																					
q17																					
q18																					
q19																			1		
q20																					
q21																		1			
q22																					
q23																				1	
q24																	1				
q25																					
q26																					
q27																					
q28																					
q29																					
q30																					
q31																					
q32														1							
q33																					
q34																					
q35																					
q36																					1
q37																					
q38																					
q39																					
q40																					

Continuation of Transition Matrix for Symbol **0**

0	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39	q40
q0																				
q1																				
q2																				
q3																				
q4																				
q5																				
q6																				
q7																				
q8																				
q9																				
q10																				
q11			$2^{-1/2}$	$2^{-1/2}$																
q12																				
q13																				
q14																				
q15																				
q16			$-2^{-1/2}$	$2^{-1/2}$																
q17	$-2^{-1/2}$	$2^{-1/2}$																		
q18	$2^{-1/2}$	$2^{-1/2}$																		
q19																				
q20																1				
q21																				
q22													$2^{-1/2}$	$2^{-1/2}$						
q23																				
q24																				
q25					1															
q26													$-2^{-1/2}$	$2^{-1/2}$						
q27											$-2^{-1/2}$	$2^{-1/2}$								
q28											$2^{-1/2}$	$2^{-1/2}$								
q29								1												
q30																			1	
q31							1													
q32																				
q33									1											
q34						1														
q35															1					
q36																				
q37																	1			
q38																		1		
q39																				1
q40										1										

Transition Matrix for Symbol **1**

1	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20
q0		1																			
q1	1																				
q2			1																		
q3				1																	
q4													1								
q5						1															
q6										1											
q7								1													
q8									1												
q9							1														
q10											1										
q11												1									
q12					1																
q13														1							
q14															1						
q15																					
q16																	1				
q17																					1
q18																			1		
q19																					
q20																		1			
q21																					
q22																					
q23																1					
q24																					
q25																					
q26																					
q27																					
q28																					
q29																					
q30																					
q31																					
q32																					
q33																					
q34																					
q35																				1	
q36																					
q37																					
q38																					
q39																					
q40																					

Continuation of Transition Matrix for Symbol **1**

1	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39	q40
q0																				
q1																				
q2																				
q3																				
q4																				
q5																				
q6																				
q7																				
q8																				
q9																				
q10																				
q11																				
q12																				
q13																				
q14																				
q15			1																	
q16																				
q17																				
q18																				
q19															1					
q20																				
q21	1																			
q22		1																		
q23																				
q24				1																
q25													1							
q26						1														
q27										1										
q28								1												
q29																	1			
q30							1													
q31											1									
q32												1								
q33					1															
q34														1						
q35																				
q36																1				
q37																		1		
q38									1											
q39																			1	
q40																				1

Transition Matrix for Symbol $

$	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20
q0				1																	
q1		1																			
q2														1							
q3	1																				
q4					1																
q5									1												
q6							1														
q7								1													
q8						1															
q9										1											
q10											1										
q11												1									
q12													1								
q13			1																		
q14															1						
q15																1					
q16																				1	
q17																		1			
q18																			1		
q19																	1				
q20																					1
q21																					
q22																					
q23																					
q24																					
q25																					
q26																					
q27																					
q28																					
q29																					
q30																					
q31																					
q32																					
q33																					
q34																					
q35																					
q36																					
q37																					
q38																					
q39																					
q40																					

Continuation of Transition Matrix for Symbol **$**

$	q21	q22	q23	q24	q25	q26	q27	q28	q29	q30	q31	q32	q33	q34	q35	q36	q37	q38	q39	q40
q0																				
q1																				
q2																				
q3																				
q4																				
q5																				
q6																				
q7																				
q8																				
q9																				
q10																				
q11																				
q12																				
q13																				
q14																				
q15																				
q16																				
q17																				
q18																				
q19																				
q20																				
q21	1																			
q22		1																		
q23			1																	
q24				1																
q25					1															
q26									1											
q27							1													
q28								1												
q29						1														
q30										1										
q31											1									
q32												1								
q33													1							
q34														1						
q35															1					
q36																1				
q37																	1			
q38																		1		
q39																			1	
q40																				1

APPENDIX C

File format explanation:

File is composed of a series of nested blocks. Each block begins with an @ symbol following block name, then a pair of curly brackets marking the block body. There is a tree like hierarchy in the file. Each block may contain other blocks as well as single items. Ordering of the children in another block is not important, but names must be unique in any blocks scope. If it repeats, then the latter one will override the previous ones. Single items have a $ sign in front of their names, and they can not be split into multiple lines.

Example:

```
@block_name
{
      … // some data
}

$item_name item_data ...
```

For 2QFA design, we use a simple format. Main block is named qfa, and every other block resides in it.

Children of @qfa:

$symbols : It allows the designer to specify the tape alphabet.
@states : It describes the states
@layout : It describes the visual layout of the 2QFA
@transitions : It describes the transition function

Code for the machines:

Machine 1:

```
@qfa
{
    $symbols 0 1

    @states
    {
        $q0 non-halting right
        $q1 reject stop
        $q2 reject stop
        $q3 accept stop
        $q4 non-halting right
        $q5 non-halting right
        $q6 non-halting right
        $q7 non-halting stop
        $q8 non-halting stop
        $q9 non-halting stop
        $q10 non-halting stop
        $q11 non-halting stop
        $q12 non-halting stop
        $q13 non-halting stop
        $q14 non-halting stop
    }

    @layout
    {
        @group0
        {
            $origin 200 220
            $type circular
            $direction ccw
            $startangle 180
            @states
            {
                $list $q0
            }
        }

        @group1
        {
            $type linear
            $origin 325 400
            @states
            {
                $list $q1 $q2 $q3
            }
        }

        @group2
        {
            $type linear
            $density 2.2
            $origin 550 150
            @states
            {
                $list $q4 $q7 $q10 $q12
            }
        }

        @group3
        {
            $type linear
            $density 2.2
            $origin 550 250
            @states
            {
                $list $q5 $q8 $q13
            }
        }
```

```
    @group4
    {
        $type linear
        $density 2.2
        $origin 550 350
        @states
        {
            $list $q6 $q9 $q11 $q14
        }
    }

}

@transitions
{
    @trans0
    {
        $symbol $
        $q0 $q0 1
        $q1 $q5 1
        $q2 $q6 1
        $q3 $q4 1
        $q4 $q1 "-0.2886751345948|0.5" $q2 "-0.2886751345948|-0.5" $q3 ir3
        $q5 $q1 "-0.2886751345948|-0.5" $q2 "-0.2886751345948|0.5" $q3 ir3
        $q6 $q1 ir3 $q2 ir3 $q3 ir3
        $q7 $q7 1
        $q8 $q8 1
        $q9 $q9 1
        $q10 $q10 1
        $q11 $q11 1
        $q12 $q12 1
        $q13 $q13 1
        $q14 $q14 1
    }

    @trans1
    {
        $symbol 0
        $q0 $q0 1
        $q1 $q1 1
        $q2 $q2 1
        $q3 $q3 1
        $q4 $q7 1
        $q5 $q13 1
        $q6 $q14 1
        $q7 $q4 1
        $q8 $q5 1
        $q9 $q6 1
        $q10 $q10 1
        $q11 $q9 1
        $q12 $q12 1
        $q13 $q8 1
        $q14 $q11 1
    }

    @trans2
    {
        $symbol 1
        $q0 $q0 1
        $q1 $q1 1
        $q2 $q2 1
        $q3 $q3 1
        $q4 $q12 1
        $q5 $q13 1
        $q6 $q9 1
        $q7 $q4 1
        $q8 $q5 1
        $q9 $q6 1
        $q10 $q7 1
        $q11 $q11 1
        $q12 $q10 1
        $q13 $q8 1
        $q14 $q14 1
    }
```

```
        @trans3
        {
            $symbol #
            $q0 $q4 ir3 $q5 ir3 $q6 ir3
            $q1 $q1 1
            $q2 $q2 1
            $q3 $q3 1
            $q4 $q0 ir3 $q4 ir3 $q5 -ir3
            $q5 $q0 ir3 $q5 ir3 $q6 -ir3
            $q6 $q0 ir3 $q6 ir3 $q4 -ir3
            $q7 $q7 1
            $q8 $q8 1
            $q9 $q9 1
            $q10 $q10 1
            $q11 $q11 1
            $q12 $q12 1
            $q13 $q13 1
            $q14 $q14 1
        }
    }

}
```

Machine 2:

```
@qfa
{
    $symbols 0 1

    @states
    {
        $q0 non-halting right
        $q1 non-halting left
        $q2 non-halting right
        $q3 reject stop
        $q4 reject stop
        $q5 reject stop
        $q6 accept stop
        $q7 non-halting left
        $q8 non-halting left
        $q9 non-halting left
        $q10 non-halting stop
        $q11 non-halting stop
        $q12 non-halting stop
        $q13 non-halting stop
        $q14 non-halting stop
        $q15 non-halting stop
        $q16 non-halting stop
        $q17 non-halting stop
    }


    @layout
    {
        @group0
        {
            $origin 200 220
            $type circular
            $direction ccw
            $startangle 180
            @states
            {
                $list $q0 $q1 $q2 $q3
            }
        }

        @group1
        {
            $type linear
            $origin 325 400
            @states
```

```
        {
            $list $q4 $q5 $q6
        }
    }

    @group2
    {
        $type linear
        $density 2.2
        $origin 550 150
        @states
        {
            $list $q7 $q10 $q13 $q15
        }
    }

    @group3
    {
        $type linear
        $density 2.2
        $origin 550 250
        @states
        {
            $list $q8 $q11 $q16
        }
    }

    @group4
    {
        $type linear
        $density 2.2
        $origin 550 350
        @states
        {
            $list $q9 $q12 $q14 $q17
        }
    }

}

@transitions
{
    @trans0
    {
        $symbol #
        $q1 $q3 1
        $q3 $q1 1
        $q4 $q8 1
        $q5 $q9 1
        $q6 $q7 1
        $q7 $q4 "-0.2886751345948|0.5" $q5 "-0.2886751345948|-0.5" $q6 ir3
        $q8 $q4 "-0.2886751345948|-0.5" $q5 "-0.2886751345948|0.5" $q6 ir3
        $q9 $q4 ir3 $q5 ir3 $q6 ir3
    }

    @trans1
    {
        $symbol 0
        $q1 $q2 1
        $q2 $q3 1
        $q3 $q1 1
        $q7 $q10 1
        $q8 $q16 1
        $q9 $q17 1
        $q10 $q7 1
        $q11 $q8 1
        $q12 $q9 1
        $q14 $q12 1
        $q16 $q11 1
        $q17 $q14 1
    }

    @trans2
    {
        $symbol 1
```

```
            $q0 $q1 1
            $q1 $q0 1
            $q7 $q15 1
            $q8 $q16 1
            $q9 $q12 1
            $q10 $q7 1
            $q11 $q8 1
            $q12 $q9 1
            $q13 $q10 1
            $q15 $q13 1
            $q16 $q11 1
        }

        @trans3
        {
            $symbol $
            $q0 $q3 1
            $q2 $q7 ir3 $q8 ir3 $q9 ir3
            $q3 $q0 1
            $q7 $q2 ir3 $q7 ir3 $q8 -ir3
            $q8 $q2 ir3 $q8 ir3 $q9 -ir3
            $q9 $q2 ir3 $q9 ir3 $q7 -ir3
        }
    }

}
```

Machine 3:

```
@qfa
{
    $symbols 0 1 2

    @states
    {
        $q0 non-halting right
        $q1 non-halting left
        $q2 non-halting right
        $q3 reject stop
        $q4 reject stop
        $q5 reject stop
        $q6 non-halting stop
        $q7 non-halting left
        $q8 non-halting left
        $q9 non-halting left
        $q10 non-halting stop
        $q11 non-halting stop
        $q12 non-halting stop
        $q13 non-halting stop
        $q14 non-halting stop
        $q15 non-halting stop
        $q16 non-halting stop
        $q17 non-halting stop
        $q18 non-halting left
        $q19 non-halting right
        $q20 reject stop

        $q21 non-halting left
        $q22 non-halting left
        $q23 non-halting left
        $q24 non-halting stop
        $q25 non-halting stop
        $q26 non-halting stop
        $q27 non-halting stop
        $q28 non-halting stop
        $q29 non-halting stop
        $q30 non-halting stop
        $q31 non-halting stop

        $q32 non-halting right
        $q33 non-halting stop
        $q34 accept stop

    }
```

```
    @layout
    {
        @group0
        {
            $origin 200 240
            $density 0.55
            $type circular
            $direction ccw
            $startangle 45
            @states
            {
                $list $q0 $q1 $q2 $q3 $q18 $q19 $q20
            }
        }

        @group1
        {
            $type linear
            $origin 400 290
//            $angle 0
            @states
            {
                $list $q4 $q5 $q6 $q34
            }
        }

        @group2
        {
            $type linear
            $density 2.2
            $origin 630 75
            @states
            {
                $list $q7 $q10 $q13 $q15
            }
        }

        @group3
        {
            $type linear
            $density 2.2
            $origin 630 150
            @states
            {
                $list $q8 $q11 $q16
            }
        }

        @group4
        {
            $type linear
            $density 2.2
            $origin 630 225
            @states
            {
                $list $q9 $q12 $q14 $q17
            }
        }

        @group5
        {
            $type linear
            $density 2.2
            $origin 630 350
            @states
            {
                $list $q21 $q24 $q27 $q29
            }
        }

        @group6
        {
            $type linear
            $density 2.2
```

```
            $origin 630 425
            @states
            {
                $list $q22 $q25 $q30
            }
        }

        @group7
        {
            $type linear
            $density 2.2
            $origin 630 500
            @states
            {
                $list $q23 $q26 $q28 $q31
            }
        }

        @group8
        {
            $type linear
            $density 2.2
            $origin 400 425
//            $angle 0
            @states
            {
                $list $q32 $q33
            }
        }


    }

    @transitions
    {
        @trans0
        {
            $symbol #
            $q0 $q0 1
            $q1 $q3 1
            $q2 $q2 1
            $q3 $q1 1

            $q4 $q21 1
            $q5 $q22 1
            $q34 $q23 1

            $q21 $q4 "-0.2886751345948|0.5" $q5 "-0.2886751345948|-0.5" $q34 ir3
            $q22 $q4 "-0.2886751345948|-0.5" $q5 "-0.2886751345948|0.5" $q34 ir3
            $q23 $q4 ir3 $q5 ir3 $q34 ir3


        }

        @trans1
        {
            $symbol 0
            $q0 $q0 1
            $q1 $q2 1
            $q2 $q3 1
            $q3 $q1 1

            $q4 $q9 1
            $q5 $q8 1
            $q6 $q32 1
            $q32 $q7 1
            $q7 $q4 "-0.2886751345948|0.5" $q5 "-0.2886751345948|-0.5" $q6 ir3
            $q8 $q4 "-0.2886751345948|-0.5" $q5 "-0.2886751345948|0.5" $q6 ir3
            $q9 $q4 ir3 $q5 ir3 $q6 ir3

            $q21 $q24 1
            $q22 $q30 1
            $q23 $q31 1
            $q24 $q21 1
            $q25 $q22 1
```

```
    $q26 $q23 1
    $q27 $q27 1
    $q28 $q26 1
    $q29 $q29 1
    $q30 $q25 1
    $q31 $q28 1


    $q19 $q20 1
    $q20 $q19 1

}

@trans2
{
    $symbol 1
    $q0 $q1 1
    $q1 $q0 1
    $q2 $q2 1
    $q3 $q3 1
    $q4 $q4 1
    $q5 $q5 1
    $q6 $q6 1

    $q7 $q10 1
    $q8 $q16 1
    $q9 $q17 1
    $q10 $q7 1
    $q11 $q8 1
    $q12 $q9 1
    $q13 $q13 1
    $q14 $q12 1
    $q15 $q15 1
    $q16 $q11 1
    $q17 $q14 1

    $q21 $q29 1
    $q22 $q30 1
    $q23 $q26 1
    $q24 $q21 1
    $q25 $q22 1
    $q26 $q23 1
    $q27 $q24 1
    $q28 $q28 1
    $q29 $q27 1
    $q30 $q25 1
    $q31 $q31 1


    $q18 $q19 1
    $q19 $q20 1
    $q20 $q18 1

    $q32 $q32 1
}

@trans3
{
    $symbol 2
    $q0 $q3 1
    $q1 $q1 1
    $q2 $q18 1
    $q3 $q0 1
    $q4 $q4 1
    $q5 $q5 1
    $q6 $q6 1


    $q7 $q15 1
    $q8 $q16 1
    $q9 $q12 1
    $q10 $q7 1
    $q11 $q8 1
    $q12 $q9 1
    $q13 $q10 1
```

```
            $q14 $q14 1
            $q15 $q13 1
            $q16 $q11 1
            $q17 $q17 1

            $q18 $q2 1
            $q19 $q19 1

            $q21 $q32 ir3 $q21 ir3 $q22 -ir3
            $q22 $q32 ir3 $q22 ir3 $q23 -ir3
            $q23 $q32 ir3 $q23 ir3 $q21 -ir3

            $q33 $q21 ir3 $q22 ir3 $q23 ir3


            $q32 $q33 1
        }

        @trans4
        {
            $symbol $
            $q0 $q3 1
            $q1 $q1 1
            $q3 $q0 1

            $q2 $q20 1
            $q20 $q2 1

            $q7 $q19 ir3 $q7 ir3 $q8 -ir3
            $q8 $q19 ir3 $q8 ir3 $q9 -ir3
            $q9 $q19 ir3 $q9 ir3 $q7 -ir3

            $q19 $q7 ir3 $q8 ir3 $q9 ir3

        }

    }

}
```

Machine 4:

```
@qfa
{
    $symbols 0 1

    @states
    {
        $q0 non-halting right
        $q1 non-halting left
        $q2 non-halting right
        $q3 reject stop
        $q4 reject stop
        $q5 reject stop
        $q6 accept stop
        $q7 non-halting left
        $q8 non-halting left
        $q9 non-halting left
        $q10 non-halting stop
        $q11 non-halting stop
        $q12 non-halting stop
        $q13 non-halting stop
        $q14 non-halting stop
        $q15 non-halting stop
        $q16 non-halting stop
        $q17 non-halting stop
        $q18 non-halting stop
        $q19 non-halting stop
        $q20 non-halting stop
        $q21 non-halting stop
        $q22 non-halting stop
    }

    @layout
```

```
    {
        @group0
        {
            $origin 160 220
            $density 0.5
            $type circular
            $direction ccw
            $startangle 180
            @states
            {
                $list $q0 $q1 $q2 $q3
            }
        }

        @group1
        {
            $type linear
            $origin 250 400
//            $angle 0
            @states
            {
                $list $q4 $q5 $q6
            }
        }

        @group2
        {
            $type linear
            $density 1.8
            $origin 450 150
            @states
            {
                $list $q7 $q10 $q13 $q15
            }
        }

        @group3
        {
            $type linear
            $density 1.8
            $origin 450 250
            @states
            {
                $list $q8 $q11 $q16 $q18 $q19
            }
        }

        @group4
        {
            $type linear
            $density 1.8
            $origin 450 350
            @states
            {
                $list $q9 $q12 $q14 $q17 $q20 $q21 $q22
            }
        }

    }

    @transitions
    {
        @trans0
        {
            $symbol #
            $q1 $q3 1
            $q2 $q2 1
            $q3 $q1 1
            $q4 $q8 1
            $q5 $q9 1
            $q6 $q7 1
            $q7 $q4 "-0.2886751345948|0.5" $q5 "-0.2886751345948|-0.5" $q6 ir3
            $q8 $q4 "-0.2886751345948|-0.5" $q5 "-0.2886751345948|0.5" $q6 ir3
            $q9 $q4 ir3 $q5 ir3 $q6 ir3
        }
```

```
        @trans1
        {
            $symbol 0
            $q1 $q2 1
            $q2 $q3 1
            $q3 $q1 1
            $q7 $q13 1
            $q8 $q19 1
            $q9 $q22 1
            $q10 $q7 1
            $q11 $q8 1
            $q12 $q9 1
            $q13 $q10 1
            $q14 $q12 1
            $q15 $q15 1
            $q16 $q11 1
            $q17 $q14 1
            $q18 $q16 1
            $q19 $q18 1
            $q20 $q17 1
            $q21 $q20 1
            $q22 $q21 1
        }

        @trans2
        {
            $symbol 1
            $q0 $q1 1
            $q1 $q0 1
            $q2 $q2 1
            $q3 $q3 1
            $q4 $q4 1
            $q5 $q5 1
            $q6 $q6 1
            $q7 $q15 1
            $q8 $q16 1
            $q9 $q12 1
            $q10 $q7 1
            $q11 $q8 1
            $q12 $q9 1
            $q13 $q10 1
            $q14 $q14 1
            $q15 $q13 1
            $q16 $q11 1
            $q17 $q17 1
        }

        @trans3
        {
            $symbol $
            $q0 $q3 1
            $q1 $q1 1
            $q2 $q7 ir3 $q8 ir3 $q9 ir3
            $q3 $q0 1
            $q4 $q4 1
            $q5 $q5 1
            $q6 $q6 1
            $q7 $q2 ir3 $q7 ir3 $q8 -ir3
            $q8 $q2 ir3 $q8 ir3 $q9 -ir3
            $q9 $q2 ir3 $q9 ir3 $q7 -ir3
            $q10 $q10 1
            $q11 $q11 1
            $q12 $q12 1
            $q13 $q13 1
            $q14 $q14 1
            $q15 $q15 1
            $q16 $q16 1
            $q17 $q17 1
        }
    }

}
```

Machine 5:

```
@qfa
{
    $symbols 0 1 2

    @states
    {
        $q0 non-halting right
        $q1 non-halting left
        $q2 non-halting right
        $q3 reject stop
        $q4 reject stop
        $q5 reject stop
        $q6 non-halting stop
        $q7 non-halting left
        $q8 non-halting left
        $q9 non-halting left
        $q10 non-halting stop
        $q11 non-halting stop
        $q12 non-halting stop
        $q13 non-halting stop
        $q14 non-halting stop
        $q15 non-halting stop
        $q16 non-halting stop
        $q17 non-halting stop
        $q18 non-halting left
        $q19 non-halting right
        $q20 reject stop

        $q21 non-halting left
        $q22 non-halting left
        $q23 non-halting left
        $q24 non-halting stop
        $q25 non-halting stop
        $q26 non-halting stop
        $q27 non-halting stop
        $q28 non-halting stop
        $q29 non-halting stop
        $q30 non-halting stop
        $q31 non-halting stop

        $q32 non-halting right
        $q33 non-halting stop
        $q34 accept stop

        $q35 non-halting stop
        $q36 non-halting stop

        $q37 non-halting stop
        $q38 non-halting stop
        $q39 non-halting stop

    }

    @layout
    {
        @group0
        {
            $origin 175 240
            $density 0.45
            $type circular
            $direction ccw
            $startangle 45
            @states
            {
                $list $q0 $q1 $q2 $q3 $q18 $q19 $q20
            }
        }

        @group1
        {
            $type linear
```

```
            $origin 350 290
//            $angle 0
            @states
            {
                $list $q4 $q5 $q6 $q34
            }
        }

        @group2
        {
            $type linear
            $density 1.3
            $origin 540 75
            @states
            {
                $list $q7 $q10 $q13 $q15
            }
        }

        @group3
        {
            $type linear
            $density 1.3
            $origin 540 150
            @states
            {
                $list $q8 $q11 $q16 $q35 $q36
            }
        }

        @group4
        {
            $type linear
            $density 1.3
            $origin 540 225
            @states
            {
                $list $q9 $q12 $q14 $q17 $q37 $q38 $q39
            }
        }

        @group5
        {
            $type linear
            $density 1.3
            $origin 540 350
            @states
            {
                $list $q21 $q24 $q27 $q29
            }
        }

        @group6
        {
            $type linear
            $density 1.3
            $origin 540 425
            @states
            {
                $list $q22 $q25 $q30
            }
        }

        @group7
        {
            $type linear
            $density 1.3
            $origin 540 500
            @states
            {
                $list $q23 $q26 $q28 $q31
            }
        }

        @group8
```

```
        {
            $type linear
            $density 1.3
            $origin 350 425
//            $angle 0
            @states
            {
                $list $q32 $q33
            }
        }


    }

    @transitions
    {
        @trans0
        {
            $symbol #
            $q0 $q0 1
            $q1 $q3 1
            $q2 $q2 1
            $q3 $q1 1

            $q4 $q21 1
            $q5 $q22 1
            $q34 $q23 1

            $q21 $q4 "-0.2886751345948|0.5" $q5 "-0.2886751345948|-0.5" $q34 ir3
            $q22 $q4 "-0.2886751345948|-0.5" $q5 "-0.2886751345948|0.5" $q34 ir3
            $q23 $q4 ir3 $q5 ir3 $q34 ir3


        }

        @trans1
        {
            $symbol 0
            $q0 $q0 1
            $q1 $q2 1
            $q2 $q3 1
            $q3 $q1 1

            $q4 $q9 1
            $q5 $q8 1
            $q6 $q32 1
            $q32 $q7 1
            $q7 $q4 "-0.2886751345948|0.5" $q5 "-0.2886751345948|-0.5" $q6 ir3
            $q8 $q4 "-0.2886751345948|-0.5" $q5 "-0.2886751345948|0.5" $q6 ir3
            $q9 $q4 ir3 $q5 ir3 $q6 ir3

            $q21 $q24 1
            $q22 $q30 1
            $q23 $q31 1
            $q24 $q21 1
            $q25 $q22 1
            $q26 $q23 1
            $q27 $q27 1
            $q28 $q26 1
            $q29 $q29 1
            $q30 $q25 1
            $q31 $q28 1


            $q19 $q20 1
            $q20 $q19 1

        }

        @trans2
        {
            $symbol 1
            $q0 $q1 1
            $q1 $q0 1
```

```
    $q7 $q13 1
    $q13 $q10 1
    $q10 $q7 1

    $q8 $q36 1
    $q36 $q35 1
    $q35 $q16 1
    $q16 $q11 1
    $q11 $q8 1

    $q9 $q39 1
    $q39 $q38 1
    $q38 $q37 1
    $q37 $q17 1
    $q17 $q14 1
    $q14 $q12 1
    $q12 $q9 1

    $q21 $q29 1
    $q22 $q30 1
    $q23 $q26 1
    $q24 $q21 1
    $q25 $q22 1
    $q26 $q23 1
    $q27 $q24 1

    $q29 $q27 1
    $q30 $q25 1



    $q18 $q19 1
    $q19 $q20 1
    $q20 $q18 1

}

@trans3
{
    $symbol 2
    $q0 $q3 1
    $q1 $q1 1
    $q2 $q18 1
    $q3 $q0 1
    $q4 $q4 1
    $q5 $q5 1
    $q6 $q6 1


    $q7 $q15 1
    $q8 $q16 1
    $q9 $q12 1
    $q10 $q7 1
    $q11 $q8 1
    $q12 $q9 1
    $q13 $q10 1
    $q14 $q14 1
    $q15 $q13 1
    $q16 $q11 1
    $q17 $q17 1

    $q18 $q2 1
    $q19 $q19 1

    $q21 $q32 ir3 $q21 ir3 $q22 -ir3
    $q22 $q32 ir3 $q22 ir3 $q23 -ir3
    $q23 $q32 ir3 $q23 ir3 $q21 -ir3

    $q33 $q21 ir3 $q22 ir3 $q23 ir3


    $q32 $q33 1
}

@trans4
```

```
        {
            $symbol $
            $q0 $q3 1
            $q1 $q1 1
            $q3 $q0 1

            $q2 $q20 1
            $q20 $q2 1

            $q7 $q19 ir3 $q7 ir3 $q8 -ir3
            $q8 $q19 ir3 $q8 ir3 $q9 -ir3
            $q9 $q19 ir3 $q9 ir3 $q7 -ir3

            $q19 $q7 ir3 $q8 ir3 $q9 ir3

        }

    }

}
```

Machine 6:

```
@qfa
{
    $symbols 0 1

    @states
    {
        $q0 non-halting right
        $q1 non-halting right
        $q2 reject stop
        $q3 non-halting left
        $q4 non-halting left
        $q5 non-halting right
        $q6 reject stop
        $q7 non-halting stop
        $q8 non-halting left
        $q9 non-halting right
        $q10 reject stop
        $q11 non-halting stop

        $q12 non-halting left
        $q13 non-halting right
        $q14 reject stop
        $q15 accept stop
    }

    @layout
    {
        @group0
        {
            $origin 145 195
            $density 0.5
            $startangle 180
            $type circular
            @states
            {
                $list $q0 $q1 $q3 $q2
            }
        }

        @group1
        {
            $origin 430 93
            $density 2.0
            $type linear
            @states
            {
                $list $q4 $q5
            }
        }

        @group2
```

```
    {
        $origin 429 176
        $density 2.0
        $type linear
        @states
        {
            $list $q6 $q7
        }
    }

    @group3
    {
        $origin 430 263
        $density 2.0
        $type linear
        @states
        {
            $list $q8 $q9
        }
    }

    @group4
    {
        $origin 430 351
        $density 2.0
        $type linear
        @states
        {
            $list $q10 $q11
        }
    }

    @group5
    {
        $origin 430 440
        $density 2.0
        $type linear
        @states
        {
            $list $q12 $q13
        }
    }

    @group6
    {
        $origin 433 520
        $density 2.0
        $type linear
        @states
        {
            $list $q14 $q15
        }
    }

}


@transitions
{
    @trans0
    {
        $symbol #

        $q4 $q5 1
        $q5 $q4 1

        $q8 $q9 1
        $q9 $q8 1

        $q12 $q13 1
        $q13 $q12 1

    }
    @trans1
    {
```

```
            $symbol 0
        }
        @trans2
        {
            $symbol 1
            $q0 $q1 1
            $q1 $q2 1
            $q2 $q0 1

            $q3 $q4 ir2 $q5 ir2
            $q6 $q4 -ir2 $q5 ir2

            $q4 $q6 ir2 $q7 ir2
            $q5 $q6 -ir2 $q7 ir2

            $q7 $q8 ir2 $q9 ir2
            $q10 $q8 -ir2 $q9 ir2

            $q8 $q10 ir2 $q11 ir2
            $q9 $q10 -ir2 $q11 ir2

            $q11 $q12 ir2 $q13 ir2
            $q14 $q12 -ir2 $q13 ir2

            $q12 $q14 ir2 $q15 ir2
            $q13 $q14 -ir2 $q15 ir2

            $q15 $q3 1
        }
        @trans3
        {
            $symbol $
            $q1 $q3 1
            $q3 $q1 1

            $q4 $q5 1
            $q5 $q4 1

            $q8 $q9 1
            $q9 $q8 1

            $q12 $q13 1
            $q13 $q12 1

        }

    }


}
```

Machine 7:

```
@qfa
{
    $symbols 0 1 2

    @states
    {
        $q0 non-halting right
        $q1 non-halting left
        $q2 non-halting right
        $q3 reject stop
        $q4 reject stop
        $q5 reject stop
        $q6 non-halting stop
        $q7 non-halting left
        $q8 non-halting left
        $q9 non-halting left
        $q10 non-halting stop
        $q11 non-halting stop
        $q12 non-halting stop
        $q13 non-halting stop
        $q14 non-halting stop
```

```
        $q15 non-halting stop
        $q16 non-halting stop
        $q17 non-halting stop
        $q18 non-halting left
        $q19 non-halting right
        $q20 reject stop

        $q21 non-halting left
        $q22 non-halting left
        $q23 non-halting left
        $q24 non-halting stop
        $q25 non-halting stop
        $q26 non-halting stop
        $q27 non-halting stop
        $q28 non-halting stop
        $q29 non-halting stop
        $q30 non-halting stop
        $q31 non-halting stop

        $q32 non-halting right
        $q33 non-halting stop
        $q34 accept stop

        $q35 non-halting stop
        $q36 non-halting stop

        $q37 non-halting stop
        $q38 non-halting stop
        $q39 non-halting stop

        $q40 reject stop
        $q41 reject stop

    }

    @layout
    {
        @group0
        {
            $origin 175 240
            $density 0.45
            $type circular
            $direction ccw
            $startangle 45
            @states
            {
                $list $q0 $q1 $q2 $q3 $q18 $q19 $q20
            }
        }

        @group1
        {
            $type linear
            $origin 350 290
//            $angle 0
            @states
            {
                $list $q4 $q5 $q6
            }
        }

        @group2
        {
            $type linear
            $origin 350 540
//            $angle 0
            @states
            {
                $list $q40 $q41 $q34
            }
        }

        @group3
        {
            $type linear
```

```
        $density 1.3
        $origin 540 75
        @states
        {
            $list $q7 $q10 $q13 $q15
        }
    }

    @group4
    {
        $type linear
        $density 1.3
        $origin 540 150
        @states
        {
            $list $q8 $q11 $q16
        }
    }

    @group5
    {
        $type linear
        $density 1.3
        $origin 540 225
        @states
        {
            $list $q9 $q12 $q14 $q17
        }
    }

    @group6
    {
        $type linear
        $density 1.3
        $origin 540 350
        @states
        {
            $list $q21 $q24 $q27 $q29
        }
    }

    @group7
    {
        $type linear
        $density 1.3
        $origin 540 425
        @states
        {
            $list $q22 $q25 $q30 $q35 $q36
        }
    }

    @group8
    {
        $type linear
        $density 1.3
        $origin 540 500
        @states
        {
            $list $q23 $q26 $q28 $q31 $q37 $q38 $q39
        }
    }

    @group9
    {
        $type linear
        $density 1.3
        $origin 350 425
//          $angle 0
        @states
        {
            $list $q32 $q33
        }
    }
```

```
}

@transitions
{
    @trans0
    {
        $symbol #
        $q0 $q0 1
        $q1 $q3 1
        $q2 $q2 1
        $q3 $q1 1

        $q40 $q21 1
        $q41 $q22 1
        $q34 $q23 1

        $q21 $q40 "-0.2886751345948|0.5" $q41 "-0.2886751345948|-0.5" $q34 ir3
        $q22 $q40 "-0.2886751345948|-0.5" $q41 "-0.2886751345948|0.5" $q34 ir3
        $q23 $q40 ir3 $q41 ir3 $q34 ir3


        $q4 $q9 1
        $q5 $q8 1
        $q6 $q32 1
        $q32 $q7 1
        $q7 $q4 "-0.2886751345948|0.5" $q5 "-0.2886751345948|-0.5" $q6 ir3
        $q8 $q4 "-0.2886751345948|-0.5" $q5 "-0.2886751345948|0.5" $q6 ir3
        $q9 $q4 ir3 $q5 ir3 $q6 ir3

    }

    @trans1
    {
        $symbol 0
        $q0 $q0 1
        $q1 $q2 1
        $q2 $q3 1
        $q3 $q1 1

        $q7 $q10 1
        $q8 $q16 1
        $q9 $q17 1
        $q10 $q7 1
        $q11 $q8 1
        $q12 $q9 1

        $q14 $q12 1

        $q16 $q11 1
        $q17 $q14 1


        $q21 $q27 1
        $q27 $q24 1
        $q24 $q21 1

        $q22 $q36 1
        $q36 $q35 1
        $q35 $q30 1
        $q30 $q25 1
        $q25 $q22 1

        $q23 $q39 1
        $q39 $q38 1
        $q38 $q37 1
        $q37 $q31 1
        $q31 $q28 1
        $q28 $q26 1
        $q26 $q23 1

        $q19 $q20 1
        $q20 $q19 1

    }
```

```
@trans2
{
    $symbol 1
    $q0 $q1 1
    $q1 $q0 1

    $q7 $q10 1
    $q8 $q16 1
    $q9 $q17 1
    $q10 $q7 1
    $q11 $q8 1
    $q12 $q9 1

    $q14 $q12 1

    $q16 $q11 1
    $q17 $q14 1

    $q21 $q29 1
    $q22 $q30 1
    $q23 $q26 1
    $q24 $q21 1
    $q25 $q22 1
    $q26 $q23 1
    $q27 $q24 1

    $q29 $q27 1
    $q30 $q25 1



    $q18 $q19 1
    $q19 $q20 1
    $q20 $q18 1

}

@trans3
{
    $symbol 2
    $q0 $q3 1
    $q1 $q1 1
    $q2 $q18 1
    $q3 $q0 1
    $q4 $q4 1
    $q5 $q5 1
    $q6 $q6 1


    $q7 $q15 1
    $q8 $q16 1
    $q9 $q12 1
    $q10 $q7 1
    $q11 $q8 1
    $q12 $q9 1
    $q13 $q10 1
    $q14 $q14 1
    $q15 $q13 1
    $q16 $q11 1
    $q17 $q17 1

    $q18 $q2 1
    $q19 $q19 1

    $q21 $q32 ir3 $q21 ir3 $q22 -ir3
    $q22 $q32 ir3 $q22 ir3 $q23 -ir3
    $q23 $q32 ir3 $q23 ir3 $q21 -ir3

    $q33 $q21 ir3 $q22 ir3 $q23 ir3


    $q32 $q33 1
}
```

```
        @trans4
        {
            $symbol $
            $q0 $q3 1
            $q1 $q1 1
            $q3 $q0 1

            $q2 $q20 1
            $q20 $q2 1

            $q7 $q19 ir3 $q7 ir3 $q8 -ir3
            $q8 $q19 ir3 $q8 ir3 $q9 -ir3
            $q9 $q19 ir3 $q9 ir3 $q7 -ir3

            $q19 $q7 ir3 $q8 ir3 $q9 ir3

        }

    }

}
```

REFERENCES

1. Kondacs A. 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. Gruska, J., *Quantum Computing*. McGraw Hill, 1999.

3. Denning P. J., J. B. Dennis and J. E. Qualitz, *Machines, Languages, and Computation*. Prentice-Hall, 1978.

4. Nielsen, M. A. and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

5. Hardy, Y. and W. Steeb, *Classical and Quantum Computing. Birkhäuser*, 2001.

6. Say, C., *Quantum Computing Lecture Notes*, http://www.cmpe.boun.edu.tr/~say/quantum.doc

7. Ömer, B., *A Procedural Formalism for Quantum Computing*, M.S. Thesis, Technical University of Vienna, http://tph.tuwien.ac.at/~oemer/doc/qcldoc.pdf, 1998.

8. Teutenberg, J., QUASI Simulator. http://www.cs.auckland.ac.nz/~jteu004/projects/QUASI.htm , 2001.

9. Rohde, P., *Quack! Matlab Based Quantum Computer Simulator*, http://www.physics.uq.edu.au/people/rohde/blog/?page_id=20 , 2005.

10. Butscher, B. and H. Weimer, *"libquantum" The C Library For Quantum Computing*. http://www.enyo.de/libquantum/ , 2005.

11. Ömer, B., *QCL:A Programming Language for Quantum Computers.* http://tph.tuwien.ac.at/~oemer/qcl.html, 2006.

12. Gough, A., *"Quantum Entanglement", A Quantum Simulator Module for Perl*. http://search.cpan.org/dist/Quantum-Entanglement/, 2002.

13. *Fraunhofer Quantum Computing Simulator*, http://www.qc.fraunhofer.de/

14. Diaz, B. J., J. M. Burdis and F. Tabakin, QDENSITY: Simulation of A Quantum Computer (Mathematica Module). http://www.pitt.edu/~tabakin/QDENSITY/ index.htm, 2005.

15. Rodríguez, F. L., *Quantum Information Suite*, http://www.fernandolucas.info/QCS/, 2005.

16. Moore, C. and J. P. Crutchfield, *Quantum Automata and Quantum Grammars*. Theoretical Computer Science pages 275 – 306, 2000.

17. Bertoni, A., C. Mereghetti and B. Palano, *Some Formal Tools for Analyzing Quantum Automata*. Theoretical Computer Science pages 14 – 25, 2006.

18. Shepherdson, J. C., *The Reduction of Two-way Automata To One-way Automata*. IBM Journal pages 198 – 200, 1959.

19. Vardi, M. Y., *A Note on The Reduction of Two-Way Automata to One-Way Automata*. Information Processing Letters 30 pages 261-264, 1989.

20. Vaidman, L., *On Schizophrenic Experiences of The Neutron Or Why We Should Believe In The Many-Worlds Interpretation of Quantum Theory*. arXiv:quant-ph/9609006, 1996.

21. M-fun for QC Progs, http://www.ar-tiste.com/m-fun/m-fun-index.html

22. Qubit4matlab, http://bird.szfki.kfki.hu/~toth/qubit4matlab.html

23. Quantum Octave, http://quantum-octave.sf.net/

24. Open Qubit, http://www.ennui.net/~quantum/index.shtml

25. Qubiter, http://www.ar-tiste.com/qubiter.html

26. QuaSi 1 / 2 (Quantum Circuit Simulator), http://iaks-www.ira.uka.de/QIV/QuaSi/aboutquasi.html

27. Quantum Algorithm Designer, http://www-users.cs.york.ac.uk/~sok/QAD/

28. jQuantum - Quantum Computer Simulator, http://jquantum.sourceforge.net/

29. Quantum Computer Emulator, http://rugth30.phys.rug.nl/compphys0/qce.htm

30. List of QC Simulators, http://www.quantiki.org/wiki/index.php/List_of_QC_simulators