Quantum Finite-State Machines

Amanda Young
Department of Mathematics
University of California, Davis
amyoung@math.ucdavis.edu

June 12, 2014

Abstract

Stochastic finite-state machines provide information about the internal dynamics of a system when viewed as measurement outputs. While these models are extremely accurate for describing the dynamics of classical systems, they are not sensitive to certain changes in quantum systems. This calls for the developement of quantum finite-state machines and a general theory for constructing such devices. In this report, we will develop the theory of both stochastic and quantum finite-state machines, and describe in detail how to 'quantize' a number of well known examples of stochastic finite-state machines as well as give key observations necessary for constructing such machines.

1 Introduction

The introduction of ϵ-machines gave way for optimal prediction of stochastic processes with a minimal amount of information [1]. In addition, the ϵ-machine construction is ideal as it is unique and we have methods for exact reconstruction of the machine from sufficient measurement output. One interpretation of ϵ-machines is that they are models reconstruct the hidden dynamics of a system viewed through a measurement device. The processes for which ϵ-machines are used, however, are all classical systems. This brings rise to the question of whether ϵ-machines are robust to accurately describe quantum systems and specifically, are they capable to adapt to changes in the measurement protocols? Unfortunately, but not surprisingly, the answer to these questions is no. This motivates the development of quantum finite-state machines.

To develop our working definition of a quantum finite-state machine, we will first go back to stochastic finite-state machines, which are the generalized class of models to which ϵ-machines belong. The processes inherent to stochastic finite-state machines are elementary to physics and computer science, as they provide a natural way to view natural phenomena as process languages and provide process languages with the structure of a dynamical system [2]. This indicates that quantum finite-state machines may be of vital use quantum computation, where quantum algorithms such as Shor's algorithm for factoring in polynomial time are celebrated [3].

In this report, we focus on a particular type of stochastic and quantum finite-state machines, specficially that of the stochastic and quantum finite-state generator. These models generate process languages as outputs of the internal dynamics of stochastic and quantum systems. We will first discuss the class of stochastic generators, the classical analog of the quantum generators and discuss some additional properties of stochastic generators that will be of use when constructing the quantum machines. Next, we will provide an in depth discussion of the beam splitter experiement, which provides motivation for the necessity of quantum finite-state machines. We follow this with our definition for a quantum finite-state machine as well as some key properties need to construct a quantized versions stochastic machines. We then give four examples of how to construct quantum finite-state machine: the quantum beam splitter, the Double-Zero Golden Mean Process, the Odd Process, and the RRXOR Process. We end with a discussion of some conjectures for the existence of quantum finite-state machines and future directions of research.

2 Stochastic Finite-State Machines

2.1 Definitions and Properties

The definitions we use for both stochastic and quantum finite-state machines will be the same as the definitions of stochastic and quantum generators in [4]. The class of stochastic finite-state machines is in actuality much larger than what is described in this work. However, since the focus in this paper will be on stochastic and quantum finite-state generators, we will use the phrase "finite-state machine" to mean "finite-state generator" throughout our exposition. We begin with the definition of a stochastic finite-state machine.

Definition 2.1. *A* ***stochastic finite-state machine*** *is a tuple* $\mathcal{M} = \{S, X, \{T^{(x)} : x \in X\}\}$ *where*

- S *is a finite set of states, including a start state.*
- X *is a finite alphabet.*
- $T^{(x)}$ *are substochastic matrices such that* $T = \sum_{x \in X} T^{(x)}$ *is a stochastic matrix. Furthermore,* $T^{(x)}_{ss'}$ *is the probability of transitioning from state* s *to state* s' *and emitting symbol* x*.*

A ***stochastic deterministic finite-state machine*** *is a stochastic finite-state machine such that for every* $x \in X$*, any row of* $T^{(x)}$ *has at most one nonzero entry (i.e. every state emits each symbol* $x \in X$ *at most once).*

The focus of this paper will be strictly on deterministic finite-state machines, both stochastic and quantum. For brevity, though, we will cease to use the term deterministic as it is always implied.

It is a property of stochastic matrices that there exists a unique left eigenvector $\langle \pi | = (\pi_1, \pi_2, \ldots, \pi_n)$, called the *stationary probability distribution* of eigenvalue one such that the components satisfy $\pi_i \geq 0$, and $\sum_i \pi_i = 1$. We use the stationary probability density to define the probability distribution $\Pr(w^L)$ of words of length L generated from the alphabet X for the stochastic finite-state machine $\mathcal{M}$.

Definition 2.2. *Let* $\mathcal{M} = \{S, X, \{T^{(x)} : x \in X\}\}$ *be a stochastic finite-state machine with stationary probability density* $\langle \pi |$*. Let* $w^L = w_1 w_2 \ldots w_L \in X^L$*. The probability distribution for words of length* L *is given by*

$$\Pr(w^L) = \langle \pi | T^{(w^L)} | \mathbf{1} \rangle \tag{2.1}$$

where $T^{(w^L)} = T^{(w_1)} T^{(w_2)} \ldots T^{(w_L)}$*, and* $|\mathbf{1}\rangle = (1, 1, \ldots, 1)^T$*.*

When constructing a quantum finite-state machine analog of a classical process, it will be of particular interest for us to first construct a stochastic finite-state machine for the process such that the transition matrix T is unistochastic.

Definition 2.3. *A matrix* T *is called* ***bistochastic****, if all rows and columns consist of non-negative numbers that sum to one. A bistochastic matrix* T *is called* ***unistochastic*** *if for all* i *and* j*,* $T_{ij} = |U_{ij}|^2$ *where* U *is a unitary matrix.*

It is trivially true that every unistochastic matrix is bistochastic, but the converse is in general not true. Our goal when constructing a quantum finite-state machine will be to construct a stochastic finite-state machine that has a unistochastic transition matrix. It is, in general, very difficult to immediately construct such a model. As an intermediary step, we often find it helpful to first construct a stochastic finite-state machine with a bistochastic transition matrix, and then modify the machine to produce a machine with a unistochastic transition matrix.

3 Motivation

3.1 Beam Splitter

We motivate the development of quantum finite-state machines with the example of the beam splitter experiment from [4]. In this experiment, an infinite number of beam splitters are defracting the beam of

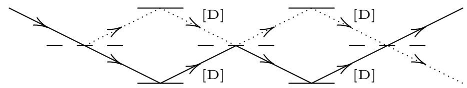
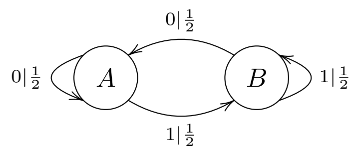


Figure 1: The infinite well of beam splitters with detectors after each splitter.

a laser. After passing through each beam splitter, the beam is either above or below the splitter which is measured by a dector, see Figure 1.

We know that after every beam spittler, the stream is detected above or below with equal probability, so can model this experiment with the following stochastic finite-state machine:



$$S = \{A,\, B\}, \quad X = \{0,\, 1\}, \quad T^{(0)} = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \quad T^{(1)} = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}. \tag{3.1}$$

The probability of any word with this stochastic finite-state machine is given by:

$$\langle \pi | T^{(w)} | \mathbf{1} \rangle, \quad \langle \pi | = \left(\tfrac{1}{2},\, \tfrac{1}{2} \right). \tag{3.2}$$

This produces a uniform word distribution on the words, as the photon stream is equally likely to be in either the upper or lower chamber after any beam splitter.

While this stochastic finite-state machine works well for describing this experiment, it is not capable of adapting to other protocols. For example, consider the protocol that removes the detectors from every other beam splitter, see Figure 2. We will refer to this as protocol II.

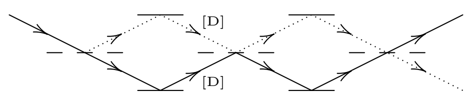


Figure 2: Beam splitter experiment with second protocol.

With this protocol, the photons enter the first beam splitter, and is detected as either being in the upper or lower chamber. Since their is no measurement after the second beam splitter, by the quantum mechanical nature of the photons they will interfere constructively along the path it was previously measured, and destructively along the other path. This means, for example, that if the stream is initially measured in the upper chamber that it will once again be measured in the upper chamber at the next beam splitter. Since these beams splitters can be thought of as 'independent' the same explaination holds for all successive beam splitters. Hence, the photons will always be measured in the upper chamber. The analogous argument holds for if the stream is first measured in the lower chamber. Thus, the stochastic finite-state machine for protocol II is given by:

Before any measurements occur, we know that the word distribution for words of length L, for protocol II is $\Pr(w^L) = \frac{1}{2}$ if $w^L = 0^L$ or 1^L, and $\Pr(w^L) = 0$ otherwise. One might wonder if we can adapt the stochastic finite-state machine $\mathcal{M} = \{S,\, X,\, T^{(0)}, T^{(1)}\}$ from the original experiment to correctly give the word distribution for protocol II. Let the symbol λ represent the chamber the photon is in after a beam splitter that has no detectors. We refer the λ as the null symbol. Since we do not know which chamber the photon is in, all we can say about the system is that some transition occurs. This means that the only possible choice for $T^{(\lambda)}$ is

$$T^{(\lambda)} = T^{(0)} + T^{(1)} = T.$$

Thus, words for protocol II are of the form $w = w_1 \lambda w_2 \lambda \ldots$. However, it can be show that, even with the null symbol the word distribution for words of length L is still uniformly distributed. Hence, the original stochastic finite-state machine cannot be adapted to represent the effects of measurement changes to the quantum mechanical system. It is cumbersome to use classical machines to model quantum mechanical systems as one would need to build a new model for each change to the measurement protocol. So a more flexible model that is sensitive to changes in measurement protocol is desired. We will see in Section 5.1 that quantum finite-state machines are exactly the class of robust models needed to deal with different measurement protocols.

4 Quantum Finite-State Machines

4.1 Definitions and Properties

We will now define the quantum version of a finite-state machine. Our goal is to minimally change the definition of a stochastic finite-state machine to construct a model that is sensitive to changes in measurement protocols for quantum systems.

Definition 4.1. *A* ***quantum finite-state machine*** *is a collection* $\mathcal{M}_q = \{Q,\, \langle\psi| \in \mathcal{H},\, X,\, \{T^{(x)} : x \in X\}\}$ *such that*

- $Q = \{q_1\, q_2, \ldots, q_n\}$ *is a set of* n *states.*
- *The state vector* $\psi \in \mathcal{H}$ *belongs to an* n*-dimensional Hilbert space* $\mathcal{H}$*.*
- X *is a finite alphabet of output symbols.*
- $U(x) = U \cdot P^{(x)}$ *is a* $n \times n$ *transition matrix that is a product of a unitary matrix* U *and an orthogonal projection operator* $P(x)$*.*
- *The projection operators are mutually orthogonal and satisfy* $\mathbf{1} = \sum_{x \in X} P(x)$*.*

A ***quantum deterministic finite-state machine*** *is a quantum finite-state machine in which each matrix* $U(x)$ *has at most one nonzero entry per row.*

To each quantum state $q_i \in Q$ we associate with it the canonical basis vector e_i, which is defined by having a single nonzero entry of a one in the ith coordinate. The collection $\mathcal{B} = \{e_i, i = 1, \ldots, n\}$ is a basis for the Hilbert space $\mathcal{H}$. The operators $P^{(x)}$ project onto the states q_i that have an incoming transition that outputs symbol x. Hence, if a state q_i has an incoming transition that outputs symbol x, by definition $P^{(x)} e_i = e_i$.

Proposition 4.2. *For any internal quantum state* q_i*, the incoming transitions are all labeled with the same output symbol.*

Proof. Let q_i be a quantum state that has an incoming transition with output symbol x. Then $P^{(x)} e_i = e_i$. Now assume that q_i has a second incoming transition that has output symbol $y \neq x$. Then, we also have $P^{(y)} e_i = e_i$. Since $y \neq x$, by the mutual orthogonality of the projections it follows that $P^{(x)} P^{(y)} = 0$. Thus

$$e_i = P^{(x)} e_i = P^{(x)} (P^{(y)} e_i) = (P^{(x)} P^{(y)}) e_i = 0,$$

a contradiction. Hence, it must be that every incoming transition to a state q_i is labeled with the same output symbol x. □

Corollary 4.3. *The number of output symbols* $|X|$ *is bounded above by the dimension of the Hilbert space, that is* $|X| \leq n$*.*

Proof. Since every state only has a single output symbol, the number of output symbols is bounded by $|Q|$. Since $|Q| = n = \dim \mathcal{H}$, it follows that $|X| \leq n$. □

One final property of quantum finite-state machines is that all states are recurrent. The sketch of this argument is now given.

Sketch. Every unitary matrix U generates a unistochastic matrix T where $T_{ij} = |U_{ij}|^2$. Since any unistochastic matrix can be decomposed into a linear combination of permutation matrices, this indicates that if a path exists from state q_i to state q_j, there must also be a path from q_j to q_i. $\square$

To calculate the word distributions for the quantum finite-state machines, we appeal to the density operator formalism of expectation used for quantum systems. Recall that a density opeator ρ on a collection of states $\{\psi_1, \ldots, \psi_k\}$ in finite dimensions is defined as

$$\rho = \sum_{i=1}^{k} p_i |\psi_i\rangle\langle\psi_i| \quad \text{where} \quad \sum_{i=1}^{k} p_i = 1.$$

As we used the stationary probability density vector for computing the word distributions for the stochastic finite-state machines, we will use the stationary density operator for computing word distributions for quantum finite-state machines.

Definition 4.4. *The* ***stationary density operator*** ρ *of a quantum finite state machine* $\mathcal{M}_q$ *is a density matrix that is invariant under unitary evolution, i.e.*

$$\rho = \sum_{x \in X} P^{(x)} U^* \rho \, U P^{(x)}. \tag{4.1}$$

For a quantum finite-state machine, the density operator is always of a particular form.

Theorem 4.5. *Let* $\mathcal{M}_q$ *be a quantum finite-state machine. Then the stationary density operator of* $\mathcal{M}_q$ *is* $\rho = |Q|^{-1} \cdot \mathbf{1}$.

Proof. We need only check that $\rho = |Q|^{-1} \cdot \mathbf{1}$ is invariant under the unitary evolution generated by $\mathcal{M}_q$.

$$\begin{aligned}
\sum_{x \in X} P^{(x)} U^* (|Q|^{-1} \cdot \mathbf{1}) U P^{(x)} &= |Q|^{-1} \sum_{x \in X} P^{(x)} U^* U P^{(x)} \\
&= |Q|^{-1} \sum_{x \in X} (P^{(x)})^2 \\
&= |Q|^{-1} \sum_{x \in X} P^{(x)} \\
&= |Q|^{-1} \cdot \mathbf{1}
\end{aligned}$$

$\square$

We are now able to define the probability distribution on words for quantum finite-state machines.

Definition 4.6. *Let* $\mathcal{M}_q$ *be a quantum finite-state machine and* $w^L = w_1 w_2 \ldots w_L \in X^L$ *a word of length* L. *Then the probability that* $\mathcal{M}_q$ *emits word* w^L *is given by*

$$\Pr(w^L) = \operatorname{Tr}(U^*(w^L) \rho U(w^L)) \tag{4.2}$$

where $\rho = |Q|^{-1} \cdot \mathbf{1}$ *is the stationary density operator of* $\mathcal{M}_q$ *and* $U(w^L) = U(w_1)U(w_2)\ldots U(w_L)$.

The following is an interesting and helpful property of the word distribution of single letter words.

Proposition 4.7. *The probability distribution of a single letter word* $x \in X$ *of a quantum finite-state machine* $\mathcal{M}_q$ *depends only on the number of quantum states and the rank of the operator* $P^{(x)}$. *Specifically,*

$$\Pr(x) = \frac{\operatorname{rank}(P^{(x)})}{|Q|}. \tag{4.3}$$

Proof. By calculation,

$$\Pr(x) = \operatorname{Tr}(P^{(x)} U^* \rho U P^{(x)}) = |Q|^{-1} \operatorname{Tr}(P^{(x)} U^* U P^{(x)}) = |Q|^{-1} \operatorname{Tr}(P^{(x)}) = \frac{\operatorname{rank}(P^{(x)})}{|Q|}.$$

$\square$

This will be particularly useful when constructing quantum finite-state machines from stochastic machines since the single letter probabilitiy $\Pr(x)$ is the ratio of the number of quantum states associated with output x to the total number of quantum states. This also gives us information on the number of quantum states necessary to construct the correct machine. We will discuss this further in Section 4.2.

Additionally, this indicates that single letter probabilities will always be rational for any quantum finite-state machine. Thus, there are stochastic finite-state machines for which quantum finite-state machines cannot be constructed. This indicates that our definition of quantum finite-state machines may be too narrow and that further investigation into the area is necessary. We will come back to this point in the conclusion.

4.2 Constructing Quantum from Stochastic

We now discuss some of the properties of stochastic and quantum finite-state machines that will be useful for constructing quantum versions of stochastic models. The main connection between quantum and stochastic finite-state machines is the fact that every quantum finite-state machine generates a stochastic finite-state machine with the same word distribution.

Theorem 4.8. *Let $\mathcal{M}_q$ be the quantum finite-state machine*

$$\mathcal{M}_q = \{Q,\, \langle\psi| \in \mathcal{H},\, X,\, \{U(x) : x \in X\}\}.$$

Then stocahstic finite-state machine

$$\mathcal{M} = \{S,\ X,\ \{T^{(x)} : x \in X\}\}$$

where $|S| = |Q|$ and $T^{(x)}_{ij} = |U(x)_{ij}|^2$ is a machine that generates the same process language and word distribution as $\mathcal{M}_q$.

Proof. Since $\mathcal{M}$ and $\mathcal{M}_q$ share the same alphabet, they can generate the same collection of words. Thus, it is only necessary to prove that the word probabilities are the same for both machines for every word. For the quantum machine,

$$\begin{aligned}
\Pr_{\mathcal{M}_q}(w^L) &= \mathrm{Tr}(U^*(w^L)\rho U(w^L)) \\
&= |Q|^{-1}\mathrm{Tr}(U^*(w^L)U(w^L)) \\
&= |Q|^{-1}\sum_{i=1}^{n}(U^*(w^L)U(w^L))_{ii} \\
&= |Q|^{-1}\sum_{i,j=1}^{n}U^*(w^L)_{ij}U(w^L))_{ij} \\
&= |Q|^{-1}\sum_{i,j=1}^{n}|U(w^L)_{ij}|^2.
\end{aligned}$$

For the stochastic machine,

$$\begin{aligned}
\Pr_{\mathcal{M}}(w^L) &= \langle\pi|T^{(w^L)}|\mathbf{1}\rangle \\
&= \sum_{i=1}^{n}\pi_i\sum_{j=1}^{n}T^{(w^L)}_{ij} \\
&= |S|^{-1}\sum_{i,j=1}^{n}T^{(w^L)}_{ij}.
\end{aligned}$$

By construction, $T^{(w^L)}_{ij} = |U(w^L)_{ij}|^2$ and $|S| = |Q|$, thus $\Pr_{\mathcal{M}}(w^L) = \Pr_{\mathcal{M}_q}(w^L)$ so the two machines generate equivalent process languages and word distributions. $\square$

Given a stochastic finite-state machine, our strategy to construct an equivalent quantum finite-state machine will be as follows: split states in the stochastic model until we obtain an equivalent stochastic finite-state machine with a unistochastic transition matrix $\tilde{T}$. Then construct a quantum finite-state machine with the same number of states as the new stochastic finite-state machine such that $\tilde{T}_{ij} = |U_{ij}|^2$. Finally, appeal to Theorem 4.8 to prove that the two machines generate the same process language and word distributions. The following observations will aid in constructing the desired stochastic finite-state machine with unistochastic transition matrix. Since quantum finite-state machines do not have transient states, it is sufficient to build the new stochastic finite-state machine from the recurrent componenet of the ϵ-machine.

Looking at the single letter word probabilities, we can determine the number of states, up to a multiple necessary for our quantum model by finding the lowest common denominator of the single letter word probabilities, which I will denote by $\mathrm{LCD}(\Pr(x))$. Given $\mathrm{LCD}(\Pr(x))$, any quantum finite-state machine will satisfy $|Q| = n \cdot \mathrm{LCD}(\Pr(x))$ where $n \in \mathbb{N}$. Furthermore, the rank of the projection operator $P^{(x)}$ must satisfy $\mathrm{rank}(P^{(x)}) = |Q| \cdot \Pr(x)$. This determines the number of states which have symbol x emitted from each incoming transition.

Since quantum finite-state machines have states that only have a single emitted symbol for incoming transitions, the associate stochastic finite-state machine described in Theorem 4.8 will also satisfy this property. Furthermore, by the unistochasticity of the transition matrix, the sum of all probabilities from incoming transitions must sum to one for any state. Looking at the recurrent component of the ϵ-machine, we will begin constructing our new stochastic finite-state machine by identifying states that violate the above properties. We will split these states as many times as necessary until we produce a model such that all incoming transitions emit a single symbol and the total incoming probability sums to one. This will produce a stochastic finite-state machine with a bistochastic transition matrix. All that is left to check is if the resulting transition matrix is actually unistochastic. If not, we attempt to rework some of the transitions in the model so that the transition matrix is unistochastic. It is trivial from there to construct the quantum finite-state machine. We now use this strategy to compute several examples of quantum-finite state machines.

5 Examples

In this section we describe how to construct quantized versions of several well known stochastic finite-state machines.

5.1 Quantized Beam Splitter

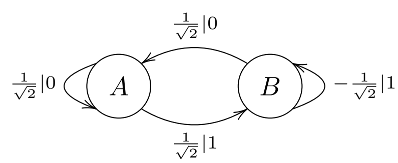
In Section 3 we motivated the construction of quantum finite-state machines with the example of a beam splitter. To prove that the above construction of a quantum finite-state machine correctly models quantum systems we construct the quantum finite-state machine for the beam splitter and show that it is sensitive to changes in the measurement protocol. Note that the stochastic matrix T from the stochastic finite-state machine in Section 3 is, in fact, unistochastic. Furthermore, each state from the stochastic model has only a single incoming emitted symbol. These are the two key features in designing quantum finite-state machines from a classical machine. Thus, we need only find a unitary matrix U for which $T_{ij} = |U_{ij}|^2$. Any such unitary will work. We choose the following quantum finite-state machine for the original beam splitter experiment.

$$S = \{A, B\}, \quad X = \{0, 1\}, \quad U = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad P^{(0)} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad P^{(1)} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{5.1}$$

We recall that the matrices $U(x) = UP^{(x)}$ are the quantum analogy of the symbol labeled transition matrices $T^{(x)}$ for the stochastic machines. The quantum machine can be summmarized by the following quantum version of a hidden Markov diagram.

Word, w	Probability, $\Pr(w)$
0	$\frac{1}{2}$
1	$\frac{1}{2}$
00	$\frac{1}{4}$
01	$\frac{1}{4}$
10	$\frac{1}{4}$
11	$\frac{1}{4}$
000	$\frac{1}{8}$
001	$\frac{1}{8}$
010	$\frac{1}{8}$
011	$\frac{1}{8}$
100	$\frac{1}{8}$
101	$\frac{1}{8}$
110	$\frac{1}{8}$
111	$\frac{1}{8}$

Table 1: Word distribution generated by quantum finite-state machine for original experiment.



Recall that the probability of observing a word $w = w_1w_2 \dots w_n$ generated by a quantum finite-state machine is given by

$$\Pr(w) = \mathrm{Tr}(U(w)^* \rho\, U(w)), \quad U(w) = U(w_1)U(w_2)\dots U(w_n).$$

Since the quantum finite-state machine for the beam splitter experiment has two states, it follows that $\rho = \frac{1}{2} \cdot \mathbf{1}$. We use the Matlab function WordDist(U, $P0$, $P1$, L) (see Appendix A for code) to calculate the word distributions of all words up to length L. For example, when $L = 3$ the words for the above quantum finite-state machine satisfy the distribution found in Table 1. Computing the word distributions for higher values of L verify that, in fact, this quantum finite-state machine has a uniform distribution over words of a set length, as in the original beam splitter experiment.

We now wish to determine if this quantum machine is robust under different measurement schemes. Once again, we consider protocol II in which the set of detectors is removed after every other beam splitter. When the beam hits a splitter that does not have a detector, it is unknown if the beam is found in the upper or lower chamber. We denote the unknown outcome with the null symbol λ. Since we would either see a '1' or '0' it follows that

$$U(\lambda) = U(P^{(0)} + P^{(1)}) = U. \tag{5.2}$$

Therefore, for a the word $w = w_1w_2 \dots w_n$ observed with the second protocol, we have

$$\Pr(w) = \Pr(w_1\lambda w_2\lambda \dots w_n\lambda) = \mathrm{Tr}(U(w)^* \rho\, U(w)) \tag{5.3}$$

where

$$U(w) = U(w_1)U(\lambda)U(w_2)U(\lambda)\dots U(w_n)U(\lambda). \tag{5.4}$$

Using the Matlab function WordDist2(U, $P0$, $P1$, L) (code in Appendix A), which is a slight modification of WordDist to model protocol II, we can determine the word distribution for all words of length less than or equal to L. The words up to length $L = 5$ with nonzero probability can be found in Table 2. It follows that with the introduction of $U(\lambda)$, the quantum finite-state machine for the beam splitter experiment produces the correct word distribution for protocol II. Hence, the quantum finite-state machine we constructed for the beam splitter is robust under the change of measurement protocol.

Word, w	Probability, $\Pr(w)$
0	$\frac{1}{2}$
1	$\frac{1}{2}$
00	$\frac{1}{2}$
11	$\frac{1}{2}$
000	$\frac{1}{2}$
111	$\frac{1}{2}$
0000	$\frac{1}{2}$
1111	$\frac{1}{2}$
00000	$\frac{1}{2}$
11111	$\frac{1}{2}$

Table 2: Word distribution generated by quantum finite-state machine for beam splitter with protocol II.

5.2 The Double-Zero Golden Mean

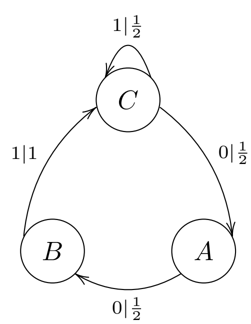
The Golden Mean process can be described as generating sequences of the form:

$$\ldots 1^{n_1} 0 1^{n_2} 0 1^{n_3} 0 \ldots$$

where $n_i \in \mathbb{Z}_{\geq 0}$. It consists of all sequences in which a zero must be followed by a one. The Double-Zero Golden Mean Process is the set of all sequences of the form

$$\ldots 1^{n_1} 00 1^{n_2} 00 1^{n_3} 00 \ldots$$

where $n_i \in \mathbb{Z}_{\geq 0}$. These are the sequences where zeros must come in pairs, and a pair of zeros must be followed by a one. When building a quantum finite-state machine, we begin by looking at the recurrent component of its ϵ-machine. For the Double-Zero Golden Mean Process this is:



$$S = \{A, B, C\}, \quad X = \{0, 1\}, \quad T^{(0)} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix}, \quad T^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}. \tag{5.5}$$

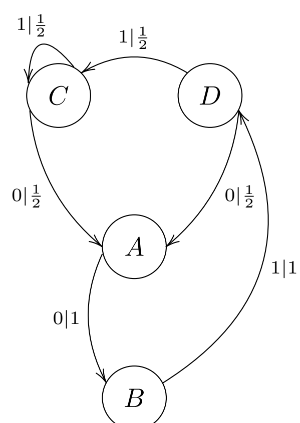
One can check that the Double-Zero Golden Mean Process has the word distribution for words length up to $L = 4$ shown in Table 3.

We wish to construct a quantized version of the Double-Zero Golden Mean Process. As stated in Section 4, the number of states in our quantum machine $|Q|$ must be divisible by $\mathrm{LCD}(\Pr(0), \Pr(1)) = 2$ where LCD stands for lowest common denominator. Additionally, for any state $q_i \in Q$, the total sum of incoming probabilities may not exceed one. Hence, it must be that state C splits into two states. Therefore, $|Q| \geq 4$. Furthermore, the ratio $\mathrm{rank}(P^{(0)}) : \mathrm{rank}(P^{(1)})$ is $1 : 1$. With these constraints in mind, we aim to produce a four-state stochastic deterministic finite-state machine which generates the same word distributions as the Double-Zero Golden Mean Process and such that the transition matrix is unistochastic. Once we find such a stochastic finite-state machine, we will then produce a 'quantized' version. For example, stochastic

Word, w	Probability, $\Pr(w)$
0	$\frac{1}{2}$
1	$\frac{1}{2}$
00	$\frac{1}{4}$
01	$\frac{1}{4}$
10	$\frac{1}{4}$
11	$\frac{1}{4}$
001	$\frac{1}{4}$
010	$\frac{1}{8}$
011	$\frac{1}{8}$
100	$\frac{1}{4}$
110	$\frac{1}{8}$
111	$\frac{1}{8}$
0010	$\frac{1}{8}$
0011	$\frac{1}{8}$
0100	$\frac{1}{8}$
0110	$\frac{1}{16}$
0111	$\frac{1}{16}$
1001	$\frac{1}{4}$
1100	$\frac{1}{8}$
1110	$\frac{1}{16}$
1111	$\frac{1}{16}$

Table 3: Word Distribution for Double-Zero Golden Mean Process

finite-state machine

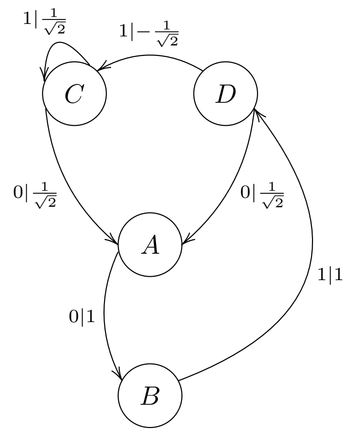


$$S = \{A, B, C, D\}, \quad X = \{0, 1\}, \quad T^{(0)} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \end{pmatrix}, \quad T^{(1)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 \end{pmatrix} \tag{5.6}$$

generates the same word distribution as the Double-Zero Golden Mean Process and has a unistochastic matrix $T = T^{(0)} + T^{(1)}$. The matrix T satisfies $T_{ij} = |U_{ij}|^2$ where U is the unitary matrix

$$U = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \end{pmatrix}. \tag{5.7}$$

Thus, the quantized version of the Double-Zero Golden Mean Process is given by



$$Q = \{A,\, B,\, C,\, D\}, \quad X = \{0,\, 1\}, \quad U = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \end{pmatrix} \tag{5.8}$$

$$P^{(0)} = |e_1\rangle\langle e_1| + |e_2\rangle\langle e_2|, \quad P^{(1)} = |e_3\rangle\langle e_3| + |e_4\rangle\langle e_4|. \tag{5.9}$$

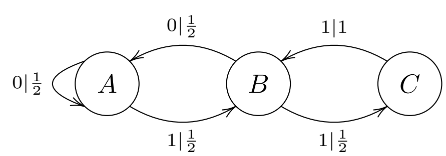
Running the program WordDist(U, $P^{(0)}$, $P^{(1)}, L$) one can verify that the word distributions generated by the above quantum finite-state machine are the same as the word distributions generated by the Double-Zero Golden Mean Process. Hence it is a quantized version of the process.

5.3 The Odd Process

The Odd Process is a stochastic finite-state machine that generates words in which the appearance of ones must occur in blocks of odd length. Hence, the Odd Process generates all bi-infinite sequences of the form:

$$\ldots 0^{n_1} 1^{2m_1+1} 0^{n_2} 1^{2m_2+1} 0^{n_3} \ldots$$

where $n_i,\, m_i \in \mathbb{Z}_{\geq 0}$. The Odd Process has the following recurrent ϵ-machine component:



$$S = \{A,\, B,\, C\}, \quad X = \{0,\, 1\}, \quad T^{(0)} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad T^{(1)} = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{pmatrix}. \tag{5.10}$$

The word distribution for the Odd Process with words up to length $L = 3$ can be found in Table 4.

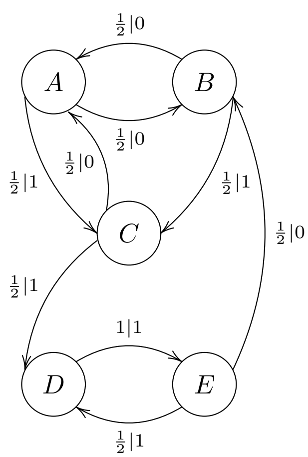
Once again, we would like to construct a stochastic finite-state machine for the Odd Process that has a unistochastic transition matrix. Since $\Pr(0) = \frac{2}{5}$, and $\Pr(1) = \frac{3}{5}$ we know that the quantum finite-state machine has a multiple of five states and that the ratio $\mathrm{rank}(P^{(0)}) : \mathrm{rank}(P^{(1)})$ is 2:3. Therefore, a minimal model will have exactly five states. We work towards finding such a quantum machine.

The method we use in this process is to construct a stochastic model with a bistochastic matrix and check if the bistochastic matrix is, in fact, unistochastic. In general, it is easier to construct models with bistochastic matrices than it is straight away construct a model with a unistochastic matrix. However, it is not guaranteed that if we find a stochastic model with a bistochastic matrix that it will actually be unistochastic matrix. To illustrate that this is more challenging than first expected, in Figure 3 we give example of a

Word, w	Probability, $\Pr(w)$
0	$\frac{2}{5}$
1	$\frac{3}{5}$
00	$\frac{1}{5}$
01	$\frac{1}{5}$
10	$\frac{1}{5}$
11	$\frac{2}{5}$
000	$\frac{1}{10}$
001	$\frac{1}{10}$
010	$\frac{1}{10}$
011	$\frac{1}{10}$
100	$\frac{1}{10}$
101	$\frac{1}{10}$
110	$\frac{1}{10}$
111	$\frac{3}{10}$

Table 4: Word Distribution for the Odd Process

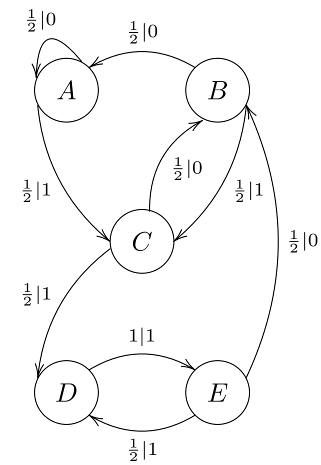
stochastic model for the Odd Process with a bistochastic transition matrix that is not unistochastic. We can see that both the rows and columns of T sum to one, but since there are rows that have a single zero entry in common, we see that it is not possible for this matrix to satisfy $T_{ij} = |U_{ij}|^2$.



$$T = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix}$$

Figure 3: A stochastic finite-state machine for the Odd Process with bistochastic, but not unistochastic, transition matrix T.

However, the example from Figure 3 gives more information about how to construct a model with a unistochastic transition matrix. We see that, for example, one issue is that both A and B transition to state C but share no other transition state. We modify the previous example to get the following stochastic finite-state machine for the Odd Process that does have a unistochastic finite-state machine.

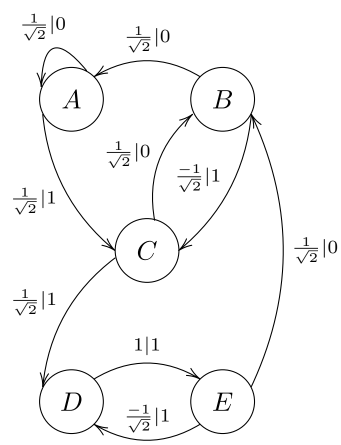


$$S = \{A,\, B,\, C,\, D,\, E\}, \quad X = \{0,\, 1\}, \quad T^{(0)} = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 \end{pmatrix}, \quad T^{(1)} = \begin{pmatrix} 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{1}{2} & 0 \end{pmatrix} \tag{5.11}$$

The unistochastic matrix $T = T^{(0)} + T^{(1)}$ satisfies $T_{ij} = |U_{ij}|^2$ where

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} & 0 \end{pmatrix}.$$

Thus, a quantum finite-state machine for the Odd Process is given by:



$$Q = \{A,\, B,\, C,\, D,\, E\}, \quad X = \{0,\, 1\}, \quad U = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} & 0 \end{pmatrix} \tag{5.12}$$

$$P^{(0)} = |e_1\rangle\langle e_1| + |e_2\rangle\langle e_2|, \quad P^{(1)} = |e_3\rangle\langle e_3| + |e_4\rangle\langle e_4| + |e_5\rangle\langle e_5| \tag{5.13}$$

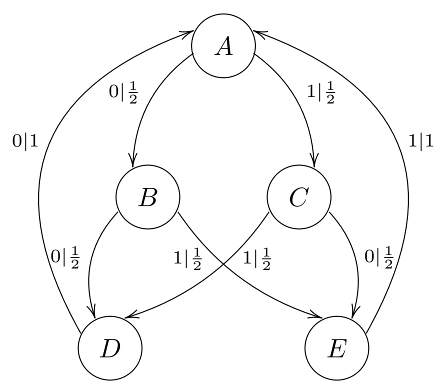
Word, w	Probability, $\Pr(w)$
0	$\frac{1}{2}$
1	$\frac{1}{2}$
00	$\frac{1}{4}$
01	$\frac{1}{4}$
10	$\frac{1}{4}$
11	$\frac{1}{4}$
000	$\frac{1}{6}$
001	$\frac{1}{12}$
010	$\frac{1}{12}$
011	$\frac{1}{6}$
100	$\frac{1}{12}$
101	$\frac{1}{6}$
110	$\frac{1}{6}$
111	$\frac{1}{12}$

Table 5: Word Distribution for the RRXOR Process

Once again, the function WordDist(U, $P0$, $P1$, L) can be used to verify that the word distribution $\Pr(w) = \mathrm{Tr}(U^*(w)\,\rho\,U(w))$ generated by the above quantum finite-state machine is indeed the same as the Odd Process, where the stationary density matrix ρ is given by $\rho = \frac{1}{5}\mathbf{1}$.

5.4 Quantum RRXOR

As our final example, we consider the RRXOR Process. The RRXOR process is described by flipping a fair coin twice, then taking the XOR operation of the two coin flips. The ϵ-machine for the RRXOR Process has the following recurrent component.



$$S = \{A,\, B,\, C,\, D,\, E\}, \quad X = \{0,\, 1\}, \quad T^{(0)} = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad T^{(1)} = \begin{pmatrix} 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.14}$$

The word distribution for the RRXOR Process for words of length $L = 3$ or less is found in Table 5. We note that since $\Pr(0) = \frac{1}{2} = \Pr(1)$, any quantum finite-state machine that generates the RRXOR Process will have an even number of states, half of which are '0' states, and half of which are '1' states.

Building a quantum state machine for the RRXOR process is an interesting example as it is the first example for which the restrictions on the incoming state symbols and probabilities are especially important. For both the Double-Zero Golden Mean Process and Odd Process, the number of quantum states was the minimum number of states larger than the number of states in the recurrent ϵ-machine component. For example, the quantum model for the Double-Zero Golden Mean Process needed an even number of states.

The recurrent component of the ϵ-machine for this process has three states and we constructed a quantum finite-state machine for the process with four states, the smallest even number greater than three. Similarly for the Odd Process, we showed that any quantum machine must have a multiple of five states. The recurrent component of the ϵ-machine for this process had three states, and the quantum finite-state machine had five states, the lowest multiple of five greater than three. If the RRXOR Process was to follow suit, we would expect that its corresponding quantum finite-state machine would have six states. However, as we will show, the quantum model must have twelve states.

We discuss the states in the ϵ-machine that must split in the quantum finite-state machine. Consider first, the bottom row of states (i.e. states D and E). Recall that quantum states may only have one incoming symbol. However, both D and E have two incoming symbol. Hence, each state must split into two states for a total of four states in the bottom row. Now, if no other states split in the model, then each of the four bottom states would transition with probability one to state A. However, a quantum state may only have incoming probabilities which sum to one. Hence, we see that state A must split into four separate states, and so the top row of states will also have four states. If states B and C do not split, then there are four states that will transition to both B and C. Once again, the allowed incoming probability would be exceeded and two additional states are needed. Hence, there is a total of four states in the middle row as well. Thus, the minimum quantum finite-state machine for the RRXOR process will have at least twelve states.

The question now becomes, is there a stochastic finite-state machine with twelve states that models the RRXOR process with a unistochastic matrix? As shown below, there is indeed such a model. See Figure 4. From this we construct a quantum finite-state machine for the RRXOR Process, as shown in Figure 5. Once again we use the function WordDist(U, $P0$, $P1$, L) to verify that quantum RRXOR Process generates the same word distributions as the stochastic model.

6 Conclusion

The above procedure has worked rather well for constructing quantum finite-state machines from deterministic stochastic finite-state machines. This leaves the question of if there will always be a quantized version of a deterministic stochastic finite state machine given that $\Pr(x) \in \mathbb{Q}$ for all $x \in X$, and if so, what is the minimal such quantum finite-state machine? I give the following conjecture to the answer of these questions.

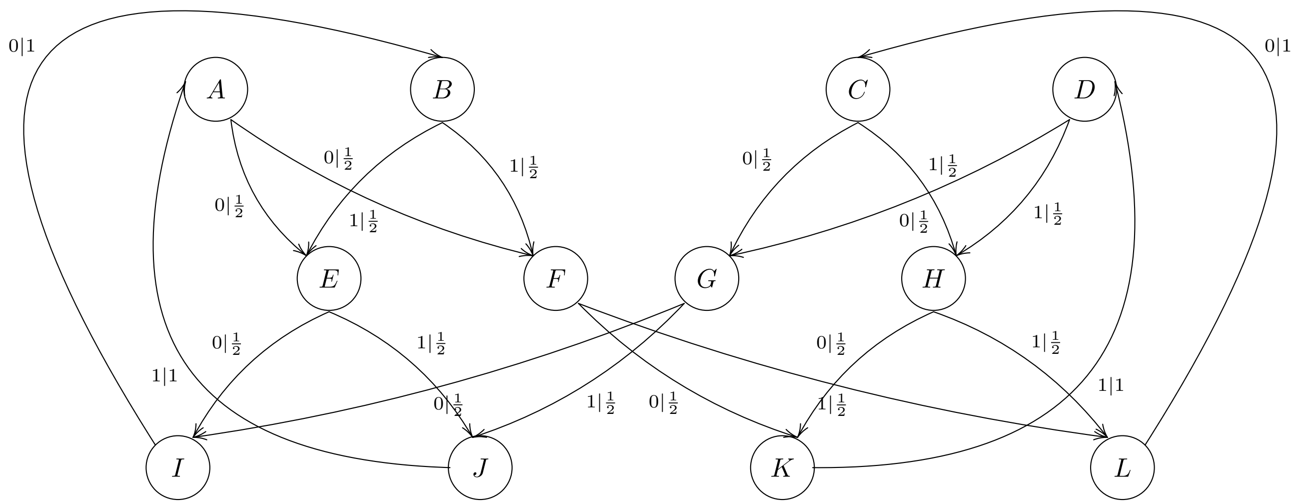
Conjecture 6.1. *Let $\mathcal{M} = \{S, \{0, 1\}, \{T^{(x)} : x \in X\}\}$ be a stochastic finite-state machine such that $\Pr(x) \in \mathbb{Q}$ for $x = 0, 1$ and let n be the least common denominator of $\{\Pr(x) : x = 0, 1\}$. Furthermore, let $m \geq |S|$ be the minimum number of states necessary so that no state has more than one incoming emitted symbol and that the incoming probabilities do not exceed a sum of one. Then there is a quantum finite-state machine $\mathcal{M}_q$ that generates the same process language and word distribution as $\mathcal{M}$ such that $|Q| = k \cdot n$ where $k \cdot n = \inf\{c \cdot n : c \cdot n \geq m\}$.*

The difficulty in proving this conjecture comes from showing the existence of a unistochastic transition matrix for the intermediate stochastic finite-state machine. As we outlined earlier, it is straightforward to show to existence of a bistochastic transition matrix, but this does not imply the existence of a unistochastic transition matrix.

One property of ϵ-machines that is not shared with quantum finite-state machines is the uniqueness of the model. It is a tragic fact that there are an infinite number of unitary matrices that produce the same unistochastic matrix. For any unitary U, any element in the collection $\mathcal{U} = \{e^{i\theta}U : 0 \leq \theta < 2\pi\}$ (as well as others) will be a unitary that produces the same unistochastic matrix. Perhaps the answer, though, to this travesty is to determine sufficient conditions on which two unitaries will produce the same unistochastic matrix. Since it does not matter what unitary we choose to construct a quantum finite-state machine, if we understand the class of unitaries that produce a given unistochastic transition matrix, we can say that the model is unique up to an equivalence class.

A more troubling problem with our current construction of quantum finite state machines is their inability to model deterministic stochastic finite-state machines that have non-rational transition probabilities. This indicates that our definition of quantum finite-state machine is not robust enough and that a more general definition of quantum finite-state machine is needed. Future work would include developing such a definition.

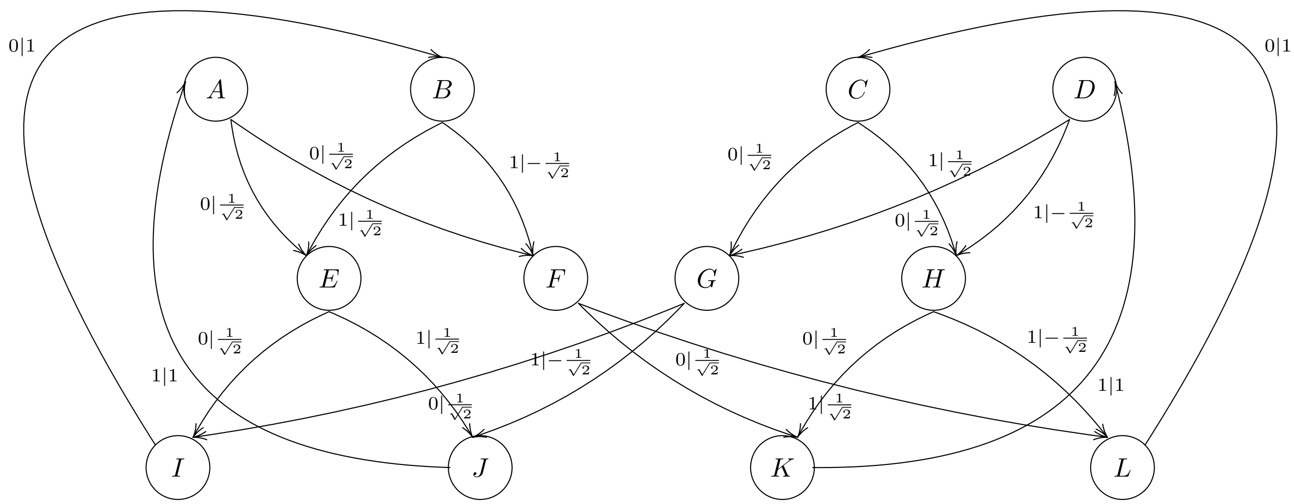
Other future research into quantum finite-state machines is the consider their relation to quantum spin systems. It has been postulated that the quantum finite-state machines could be equivalent to Matrix Product



$$T^{(0)} = \begin{pmatrix}
0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{5.15}$$

$$T^{(1)} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{5.16}$$

Figure 4: The 12 state stochastic finite-state machine for the RRXOR Process with unistochastic transition matrix.



$$U = \begin{pmatrix}
0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{\sqrt{2}} \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{5.17}$$

$$P^{(0)} = |e_1\rangle\langle e_1| + |e_4\rangle\langle e_4| + |e_5\rangle\langle e_5| + |e_7\rangle\langle e_7| + |e_9\rangle\langle e_9| + |e_{11}\rangle\langle e_{11}| \tag{5.18}$$

$$P^{(1)} = |e_2\rangle\langle e_2| + |e_3\rangle\langle e_3| + |e_6\rangle\langle e_6| + |e_8\rangle\langle e_8| + |e_{10}\rangle\langle e_{10}| + |e_{12}\rangle\langle e_{12}| \tag{5.19}$$

Figure 5: The 12 state quantum finite-state machine for the RRXOR Process.

State (MPS) or Projected Entangled Pair States (PEPS) representations of ground states of quantum spin systems. Proving an MPS or PEPS represnetation of a ground state space in the study of quantum spin systems as they are extremely useful in proving the existence of spectral gaps in the thermodynamic limit as well as stability results. Future research would be to investigate this relationship and see if we could use quantum finite-state machines to construct MPS and PEPS representations for ground state spaces.

A Matlab Code

First, is the code for the fuction WordDist(U, $P^{(0)}$, $P^{(1)}$, L).

```
%This function returns the word probabilities for all words of length L of a
%quantum finite-state machine.

function Output = WordDist(U, P0, P1, L)

[n,m] = size(U);
rho = 1/n*eye(n);
U0 = U*P0;
U1 = U*P1;

Output = cell(2^(L+1)-2, L+3);

for i = 1:2^(L+1)-2
Output{i, 1} = 'Prob of';
Output{i, L+2} = 'is';
end

for i = 1:L

%Constructing all binary words of length i.
D = [0:2^i-1]';
B = rem(floor(D*pow2(-(i-1):0)), 2);

%Fill in binary words to cell
for j = 1:2^i
for l = L+2-i:L+1
Output{2^i-2+j,l} = B(j, l-L-1+i);
end
end

%Determine probabilities for each word.
for j = 1:2^i
T = eye(n);
for k = 1:i
if B(j,k) == 0
T = T*U0;
else
T = T*U1;
end
end

  Output{j+2^i-2, L+3} = trace(T'*rho*T);
end
clear B
```

```
clear D

end
```

Second, is the code for the function WordDist(U, $P^{(0)}$, $P^{(1)}$, L). This is specifically used for calculating the word probabilities for the quantum beam splitter for protocol II. The only differences in the two codes are in the calcuation of the matrix T.

```
%This function returns the word probabilities for all words of length L of
%quantum finite-state machine with a null output after each measurement.
%Specifically, this is used to measure the word distributions for the beam
%splitter with protocol II.

function Output = WordDist2(U, P0, P1, L)

[n,m] = size(U);
rho = 1/n*eye(n);
U0 = U*P0;
U1 = U*P1;

Output = cell(2^(L+1)-2, L+3);

for i = 1:2^(L+1)-2
Output{i, 1} = 'Prob of';
Output{i, L+2} = 'is';
end

for i = 1:L

%Constructing all binary words of length i.
D = [0:2^i-1]';
B = rem(floor(D*pow2(-(i-1):0)), 2);

%Fill in binary words to cell
for j = 1:2^i
for m = L+2-i:L+1
Output{2^i-2+j,m} = B(j, m-L-1+i);
end
end

%Determine probabilities for each word.
for j = 1:2^i
T = eye(n);
for k = 1:i
if B(j,k) == 0
T = T*U0*U;
else
T = T*U1*U;
end
end

Output{j+2^i-2, L+3} = trace(T'*rho*T);
end
clear B
clear D
```

```
end
```

References

[1] C. Shalizi, J. Crutchfield "Computational mechanics: pattern and prediction, structure and simplicity," *Journal of Statistical Physics* 104 (2001), 816–879

[2] C. Moore, J. Crutchfield, "Quantum automata and quantum grammars," arXiv:quant-ph/9707031 (1997).

[3] P.W. Shor, "Algorithms for quantum computation: discrete logaritms and factoring," *Proc. 35th Symp. on Foundations of Computer Science* (1994), 56-65.

[4] K. Wiesner, J. Crutchfield, "Computation in finitary stochastic and quantum procecsses," arXiv:quant-ph/0608206v4 (2007).