MODELS AND CHARACTERIZATIONS OF 1-WAY QUANTUM FINITE AUTOMATA

By

Alexander Brodsky

B.Math. (Computer Science with Physics Minor)

University of Waterloo

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTERS OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF COMPUTER SCIENCE

We accept this thesis as conforming

to the required standard

.....

.....

THE UNIVERSITY OF BRITISH COLUMBIA

1998

© Alexander Brodsky, December, 1998

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Computer Science
The University of British Columbia
2366 Main Mall
Vancouver, Canada
V6T 1Z4

Date:

Abstract

In the past year two different models of quantum finite automata have been proposed. The first model, introduced by Moore and Crutchfield[MC98], makes one measurement on its state at the end of its computation and is called a measure-once quantum finite automata. The second model, introduced by Kondacs and Watrous[KW97], makes a measurement of its state after every transition and is called a measure-many quantum finite automata. In this thesis we investigate the characteristics of the two models.

We characterize measure-once quantum finite automata when they are restricted to acceptance with bounded error, and we show that, when they are not so restricted, they can solve the word problem over the free group. We show that they can be simulated by probabilistic finite automata, which implies that they are no more powerful than probabilistic finite automata. We also describe an algorithm that determines if two automata are equivalent.

We show that the class of languages accepted by measure-many quantum finite automata is closed under quotient, complement, and inverse homomorphisms. We prove a necessary condition for a language to be accepted by a measure-many automaton with bounded error and we show that certain sets, including piecewise testable sets, can be accepted with bounded error by this automata.

Table of Contents

Abstract ii

Table of Contents iii

List of Figures v

Acknowledgement vi

1 Introduction 1

1.1 Quantum Finite Automata 1

1.2 Related Work 3

1.3 Organization 4

2 Background and Definitions 5

2.1 Quantum Mechanics 5

2.2 Notation 8

2.3 Deterministic Finite Automata 8

2.4 Reversible Finite Automata 9

2.5 Probabilistic Finite Automata 9

2.6 Closure Properties 11

2.7 Definition of Measure-Once Quantum Finite Automata 11

2.8 Definition of Measure-Many Quantum Finite Automata 13

2.9 Language Acceptance by a Quantum Finite Automaton 14

2.10 Equivalence of Quantum Finite Automata 14

2.11 Equivalence of MO-QFAs and Moore-Crutchfield QFAs 14

2.12 Equivalence of MM-QFAs and Kondacs-Watrous QFAs 17

3 Measure-Once Quantum Finite Automata **19**

3.1 Characterization of MO-QFAs That Accept with Bounded Error 19

3.2 Closure Properties 22

3.3 MO-QFAs Can Accept Non-Regular Sets 25

3.4 Equivalence of MO-QFAs 26

3.5 Simulation of MO-QFAs by PFAs 29

4 Measure-Many Quantum Finite Automata **31**

4.1 Closure Properties 31

4.2 Simulation of MO-QFAs 36

4.3 MO-QFAs Are Strictly Less Powerful than MM-QFAs 37

4.4 MM-QFAs That Accept with Bounded Error 38

4.5 MM-QFAs Can Accept Piecewise Testable Sets 42

5 Conclusions and Future Work **50**

5.1 Open Problems 51

Bibliography **52**

List of Figures

3.1 GFA M 24
3.2 RFA M' 24

Acknowledgement

I would like to thank my supervisor, Nick Pippenger, for the freedom he gave me in pursuing this topic, for the patience while I stumbled over definitions and for the guidance that narrowed my focus and made these results possible. I would also like to thank Kevin Hare for answering my endless math questions and Mike McAllister for the Herculean task of proof reading this thesis and suggesting many improvements.

I owe thanks to many members of the computer science department for making my stay here stimulating and enjoyable, specifically, my office mates Stephane and Dave for their friendship and for tolerating my monopolization of our office, Alan for invaluable advice on academic matters, and Mike for his never-ending encouragement.

Finally, I would like to thank my parents, my brother, Matt, and Paul for their overwhelming support in my academic life. Without it things would have turned out quite differently.

Chapter 1

Introduction

Models of 1-way quantum finite automata (QFA) have been introduced by Kondacs and Watrous[KW97] in 1997 and Moore and Crutchfield[MC98] in 1998. Unlike quantum Turing machines, which are conjectured to be more powerful than their classical, probabilistic analogs, quantum finite automata that accept with bounded error are more restricted than their classical analogs. In this thesis we define two different models of quantum finite automata and investigate their properties, limitations and relationships with classical probabilistic finite automata.

1.1 Quantum Finite Automata

Quantum finite automata are finite automata whose transitions are governed by the postulates of quantum mechanics. According to quantum mechanics the state of a system may be indeterminate until it is measured. This means that the system can be in a superposition of possible states and will remain so until it is collapsed into a single state by a measurement. Furthermore, the evolution, or progress, of the system must be reversible, until a measurement is performed. Thus, quantum finite automata make at least one measurement on their state during computation. Unless otherwise stated, "quantum finite automata" will mean 1-way quantum finite automata, which use a classical head that reads the string once.

We focus on two different models of 1-way quantum finite automata. The measure-once quantum finite automata (MO-QFA)was introduced by Moore and Crutchfield[MC98], and the measure-many quantum finite automata (MM-QFA) was introduced by Kondacs and Watrous[KW97]. The crucial difference between the two models is that an MM-QFA

makes a measurement of state after each symbol is read, allowing it to decide whether to accept, reject or continue, while an MO-QFA makes no measurements until the end of the computation. The ability accept or reject a string without having to read all of it makes the MM-QFA model significantly more powerful.

The power of a finite model of computation can refer to either the conciseness of one model over the other or to the set of languages that a model can accept. In this thesis the power of a machine refers to its acceptance capability.

Measure-once quantum finite automata are the simpler of the two models because they perform only one measurement and only at the end of their computation. Despite this limitation we show that the MO-QFA can accept non-regular languages and can solve the word-problem over the free group, but only with unbounded error. We prove that the class of languages accepted by these automata is closed under quotient and that this class is properly contained within the class of languages accepted by probabilistic finite automata. Two corollaries from this result are that probabilistic finite automata can solve the word problem over the free group and that there exists a computable method to determine if two measure-once quantum finite automata have the same distribution. It is well known that probabilistic finite automata can only accept regular languages if we require that they accept with bounded error[Rab63]. Similarly we show that measure-once quantum finite automata that accept with bounded error accept exactly the set of group languages, a proper subset of the set of regular languages.

Measure-many quantum finite automata perform a measurement of state after every transition and can halt and accept before reading the entire input. These automata can simulate the measure-once quantum finite automata exactly by using at most twice as many states as the measure-once automata. We show that the class of languages accepted by these automata is closed under complement, inverse homomorphisms, and quotient. Similar to the measure-once automata the measure-many automata that accept with bounded error can only accept a proper subset of the regular languages. In fact, we show that this class of

languages is not closed under homomorphisms. We state and prove a necessary condition for acceptance with bounded error and prove a lemma that relates the sufficiency of the condition to closure of the class under boolean operations. Finally, we show by construction that measure-many automata can accept piecewise testable sets with bounded error.

1.2 Related Work

Quantum finite automata were first introduced by Kondacs and Watrous [KW97]. The model is similar to the measure-many model and was derived from the model for 2-way quantum finite automata, which was introduced in the same paper. Using a technique similar to Rabin's[Rab63], Kondacs and Watrous proved that 1-way quantum finite automata that accept with bounded error are restricted to accepting a proper subset of the set of regular languages. Further investigation by Ambainis and Freivalds [AF98] showed that measure-many quantum finite automata could accept languages with probability higher than $\frac{7}{9}$ if and only if the language could be accepted by a reversible finite automata. They noted that any reversible finite automata can be simulated by a measure-many automata with certainty. It is not yet known whether there exists a language that cannot be accepted by a reversible finite automata but that can be accepted by a measure-many automata with probability greater than ≈ 0.68. Ambainis and Freivalds [AF98] also showed that, for some languages, measure-many automata can be exponentially more concise than deterministic automata. A later paper by Ambainis, Nayak and Vazirani[ANV98] showed that there also exist languages for which the deterministic automaton is exponentially more concise than any measure-many automaton.

Moore and Crutchfield[MC98] introduced a variant of the measure-once quantum finite automata in 1998. They showed that the class of languages accepted by this model is closed under inverse homomorphisms and boolean operations. They also described a technique for simulating measure-once automata by a generalized stochastic system with only a polynomial

increase in the number of states.

Probabilistic finite automata (PFA) are the classical analog of quantum finite automata and both models share many characteristics. Many results about probabilistic automata are adaptable to quantum automata. Rabin[Rab63] showed that PFAs that accept with bounded error can only accept regular languages. However, with unbounded error, PFAs can accept non-regular languages. In 1971, Paz[Paz71] described an algorithm to determine if two probabilistic automata have the same distributions. This was followed in 1989 by a result of Tzeng[Tze89] who showed that the equivalence problem could be solved in polynomial time.

Automata that allow a 2-way read head are closely related to 1-way automata, but can be surprisingly more powerful. In 1959, Sheperdson[She59] showed that 2-way deterministic automata are no more powerful than their 1-way counterparts. However, Freivalds[Fre81] showed that 2-way probabilistic finite automata can accept non-regular languages with bounded error and exponential run-time. Dwork and Stockmeyer[DS90] proved a complementary result in that showed that 2-way probabilistic automata must take exponential time for infinitely many inputs if they accept a non-regular language with bounded error. Kondacs and Watrous[KW97] showed that the quantum counterpart to 2-way probabilistic automata, 2-way quantum finite automata, can accept non-regular languages with bounded error in linear time.

1.3 Organization

This thesis is organized in the following way: chapter 2 defines quantum automata and explains background information, chapter 3 discusses measure-once quantum finite automata, chapter 4 discusses measure-many quantum finite automata, and chapter 5 summarizes our work and discusses some open problems.

Chapter 2

Background and Definitions

The computations of quantum finite automata are based on the postulates of quantum mechanics rather than classical mechanics. To facilitate the definition of QFAs, we briefly discuss the relevant points of quantum mechanics. This is followed by a discussion of various finite automata, including group finite automata, reversible finite automata and probabilistic finite automata. Finally, we define measure-once quantum finite automata and measure-many quantum finite automata, and discuss the differences between our definitions and those of Moore and Crutchfield[MC98] and Kondacs and Watrous[KW97].

2.1 Quantum Mechanics

The four concepts central to our discussion of finite quantum systems include: include: superposition, measurement, reversible time evolution, and representation. Superposition describes the configuration of a quantum system, measurement describes what we can determine about the system, reversible time evolution describes how the system evolves with time, and representation describes how we formally describe and manipulate the system.

According to quantum mechanics, a quantum system can be in a superposition of states, rather than a single state. For each physical state that the system can be in there is a corresponding complex probability density amplitude whose absolute square is the probability of the system being in the corresponding state after a measurement occurs. The sum of the squares of all the amplitudes sum to 1 to conserve probability. For example, at any point in time a classic 2-state system will either be in state $|0\rangle$ or state $|1\rangle$. A quantum system can be in a superposition of both states, e.g. $\alpha|0\rangle + \beta|1\rangle$. The probability of measuring the

system in state $|0\rangle$ is $|\alpha|^2$ and the probability of measuring the system in state $|1\rangle$ is $|\beta|^2$ where $|\alpha|^2 + |\beta|^2 = 1$.

The act of measurement collapses the superposition of states into a determinate state. A measurement changes the configuration of the system by reducing the superposition of the system to a single state. This is different from classical physics where a measurement is generally a passive operation on the system. Morrison[Mor90] discusses some of the theory of measurement, however most of this theory is unnecessary for our purposes. For our purposes a measurement causes the immediate collapse of the superposition of states.

Until a measurement occurs the system remains in a superposition of states and is indeterminate. This is unlike a classical system where a measurement of a system in some state implies that the system was in that state for some amount of time ϵ before the measurement. A measurement of the superposition of states determines what state the system is in, but destroys the superposition of states in the process. A partial measurement of the system is also possible. A partial measurement does not measure whether the system is in a particular state, but whether it is in a subset of states. The result of a partial measurement is a partial collapse of the superposition. After every measurement the surviving amplitudes are normalized to conserve probability. If the system evolves after a measurement, the system evolves from the state it was measured in.

Between measurements, the evolution of a quantum system, with respect to time, must be completely reversible. Furthermore, the evolution of the system must conserve probability. Since the amplitudes are complex numbers whose absolute squares must sum to 1, this implies that the evolution must be unitary. The system evolves through the application of operators, which perform operations on the quantum system. An operator is unitary if it causes the system to evolve in a time-reversible manner and conserves amplitude. Apart from measurement, the system must evolve in a unitary manner; hence, operators must be unitary. Since the operators are unitary, they can be composed to form more complicated unitary operators. The requirement of unitary evolution with respect to time is the most

limiting restriction on quantum mechanical systems, especially in the case of finite state systems.

In order to reason about a quantum system we must be able to represent it. An n-state quantum finite system is represented by an n-dimension complex unit vector within an n-dimension complex Hilbert space $\mathcal{H}$. Each dimension of the $\mathcal{H}$ corresponds to one of the n states of the system and the set of vectors that form the standard basis of $\mathcal{H}$ correspond to the system being in each of its states with certainty. The configuration of the system at any time is a superposition of basis states denoted by a vector

$$|\Psi\rangle = \sum_{i=1}^{n} \alpha_i |q_i\rangle$$

where $|q_i\rangle$ is the ith vector of the standard basis, which corresponds to state q_i of the system, and α_i is the corresponding amplitude. The probability of a quantum system in superposition $|\Psi\rangle$ being in state q_i is

$$Pr[\text{system in state } \mathrm{q_i}] = |\langle \mathrm{q_i}|\Psi\rangle|^2 \tag{2.1}$$

where $\langle q_i|\Psi\rangle$ is the amplitude of state q_i. For the rest of the thesis we use vector notation to refer to states of the system. For example, instead of referring to state q_i we refer to the corresponding vector $|q_i\rangle$.

Operations performed on the system are represented by matrices. Unitary matrices represent unitary operators and projection matrices represent measurement operators. A unitary matrix is a complex matrix whose inverse is equal to the transpose of its complex-conjugate. A projection matrix is an idempotent matrix which projects a vector onto some subspace. We restrict our attention to diagonal projection matrices where entries in the diagonal are zero or one. This restriction does not affect the acceptance power of the models. If a system is in configuration $|\Psi_0\rangle$ and we apply the unitary operation U to it, then the new configuration of the system is $|\Psi_1\rangle = U|\Psi_0\rangle$. In terms of projection matrices Equation 2.1 can be rewritten as

$$Pr[\text{system in state } \mathrm{q_i}] = \langle \Psi|\mathrm{P_i}|\Psi\rangle$$

where P_i is a projection matrix whose ith diagonal entry is a one, and, $\langle\Psi|$ and $|\Psi\rangle$ are the row and column vectors representing the configuration of the system. A partial measurement projects the superposition represented by $|\Psi\rangle$ onto a subspace of $\mathcal{H}$, which may be spanned by many basis vectors. After a projection the vector is renormalized, but for our purposes this step is not necessary. The majority of our discussion will utilize matrix manipulations involving both unitary and projection matrices.

2.2 Notation

In this thesis the notation $\|v\|^2$ represents the norm of the vector. The operator $\otimes$ is the tensor product operator. The notation $[c]$ is the indicator of the condition; $[c] = 1$ if condition c is true and $[c] = 0$ otherwise. We usually use the bracket notation for vectors, $|v\rangle$ represents is a column vector v, while $\langle v|$ represents a row complex conjugate of v. The notation $\langle v|u\rangle$ represents the inner product of v and u. The over-line $\overline{c}$ denotes either the complement or the complex conjugate, we will state which if it is not clear from context. The $*$ denotes either the Kleene star operator or the complex conjugate of a vector or matrix, the meaning will be clear from the context.

2.3 Deterministic Finite Automata

Deterministic finite automata (DFAs), discussed by Rabin and Scott[RS59], are models of finite state computation and serve as the underlying model for automata theory. A DFA is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is a finite alphabet, and δ is a transition function of the form

$$\delta : Q \times \Sigma \to Q$$

that defines the new state of M given its state and the next symbol. The state q_0 is the initial state of M and F is the set of accepting states. If M reads a string x and ends

in an accepting state, then M is said to accept x, otherwise M is said to reject x. The configuration of a DFA during the computation is its instantaneous description that consists of its current state.

An important fact about DFAs is that all DFAs have a canonical form, also known as minimal form[HU79]. This fact is the result of the Myhill-Nerode theorem and will be used in some of our proofs.

2.4 Reversible Finite Automata

Quantum finite automata are quantum systems that usually operate in a unitary fashion. Since unitary operations are necessarily reversible, QFAs bear strong resemblance to various variants of reversible finite automata. A group finite automaton (GFA) is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ with the restriction that for every state $q \in Q$ and every input symbol $\sigma \in \Sigma$ there exists exactly one state $q' \in Q$ such that $\delta(q', \sigma) = q$, i.e. δ is a complete one-to-one function and the automata derived from M by reversing all transitions is deterministic.

A reversible finite automata (RFA) is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that for every state $q \in Q$ and for every symbol $\sigma \in \Sigma$ there is at most one state $q' \in Q$ such that $\delta(q', \sigma) = q$, or, if there exist states $q_1, q_2 \in Q$ and symbol $\sigma \in \Sigma$ such that $\delta(q_1, \sigma) = q = \delta(q_2, \sigma)$, then $\delta(q, \Sigma) = q$. This definition is equivalent to the one used by Ambainis and Freivalds[AF98] and is an extension of Pin's[Pin87] definition.

2.5 Probabilistic Finite Automata

Quantum finite automata are probabilistic models of computation, hence it is not surprising that they share many characteristics with probabilistic finite automata (PFAs) introduced by Rabin[Rab63] and discussed at length by Paz[Paz71]. A PFA is represented by a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where Q is the set of states, Σ is the alphabet, q_0 is the initial state, F

is the set of accepting states, and δ is the transition function

$$\delta : Q \times \Sigma \times Q \to [0,1]$$

such that $\delta(q,\sigma,q')$ denotes the probability of a transition from state q to state q' upon reading a symbol σ, such that for each state q and symbol σ it holds that $\sum_{q' \in Q} \delta(q,\sigma,q') = 1$.

It is convenient to describe PFAs using matrix notation. The configuration of an n-state PFA is represented by an n-dimensional stochastic vector. For each symbol in the alphabet there is a $|Q| \times |Q|$ stochastic matrix A_σ that determines the transitions that the PFA makes upon reading symbol σ. The initial configuration is represented by a stochastic row vector π, which is usually $(1,0,0,...,0)$, and the set of accepting states is represented by a zero-one column vector η where $\eta_i = [q_i \in F]$. The probability of a PFA M accepting a word $x \in \Sigma^*$ is

$$p_M(x) = \pi A(x) \eta$$

where $x = x_1 x_2 ... x_m$ and $A(x) = A_{x_1} A_{x_2} ... A_{x_m}$. A PFA M accepts a language $L \subseteq \Sigma^*$ with cut-point $\lambda \in [0,1]$ if for all $x \in L$ the probability of M accepting x is greater than λ and for all $x \notin L$ the probability of M accepting x is at most λ. A PFA M accepts L with bounded error if there exists a margin $\epsilon > 0$ such that $x \in L$ implies that $p_M(x) > \lambda + \epsilon$ and $x \notin L$ implies that $p_M(x) < \lambda - \epsilon$.

A language accepted by a PFA with some cut-point λ is known as a probabilistic cut-point event (PCE). The PFA corresponds to a stochastic system $S = (\pi, \{A_\sigma\}_{\sigma \in \Sigma}, \eta)$. An equivalent definition of a PCE is the set of words for which the stochastic system returns a value greater than the cut-point. If we allow arbitrary matrices instead of stochastic matrices we call the system a pseudo-stochastic system. A language accepted with cut-point λ by the pseudo-stochastic system is called a generalized cut-point event (GCE). A useful result of Paz[Paz71] is that the class of GCEs is equal to the class of PCEs.

Theorem 2.1 (Paz, 1971) *The class of PCEs is identical to the class of GCEs.*

2.6 Closure Properties

The set of operations under which a class of languages is closed can provide insight into the characteristics of the computation model. A class $\mathcal{C}$ is said to be closed under a particular operation if applying the operation to any language $L \in \mathcal{C}$ yields a language that is also in $\mathcal{C}$. In this thesis we focus on closure of boolean operations, homomorphisms, inverse homomorphisms, and word quotient.

The boolean operations are union, intersection and complement. The union of two languages L and L' over alphabet Σ is defined as $L \cup L' = \{x \in \Sigma^* \mid x \in L \ \vee \ x \in L'\}$, intersection is defined as $L \cap L' = \{x \in \Sigma^* \mid x \in L \wedge x \in L'\}$, and complement of a language is defined as $\overline{L} = \{x \in \Sigma^* \mid x \notin L\}$.

A homomorphism h is a one-to-one function of the form

$$h : \Sigma \to \Sigma^*$$

that maps symbols to strings and has the properties that $h(\epsilon) = \epsilon$ and $h(xy) = h(x)h(y)$. The homomorphism of a language L is defined as $h(L) = \{h(x) \in \Sigma^* \mid x \in L\}$. The inverse homomorphism of a language L is the set $h^{-1}(L) = \{x \in \Sigma^* \mid h(x) \in L\}$. The left word quotient of L for some word $w \in \Sigma^*$ is defined as $w^{-1}L = \{x \in \Sigma^* \mid wx \in L\}$. Symmetrically the right word quotient of L is defined as $Lw^{-1} = \{x \in \Sigma^* \mid xw \in L\}$. If we specified left or right end-markers as part of our computation models, then word quotients correspond to inverse homomorphisms on the end markers, a fact that will be useful later on.

2.7 Definition of Measure-Once Quantum Finite Automata

A measure-once quantum finite automaton is defined by a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is the input alphabet with an end-marker symbol \$, δ is the transition function of the form

$$\delta : Q \times \Sigma \times Q \to \mathbb{C}$$

such that for all $\sigma \in \Sigma$ and $q_1, q_2 \in Q$

$$\sum_{q' \in Q} \overline{\delta(q_1, \sigma, q')} \delta(q_2, \sigma, q') = \begin{cases} 1 & q_1 = q_2 \\ 0 & q_1 \neq q_2 \end{cases} \tag{2.2}$$

and $\delta(q, \sigma, q')$ denotes the amount of probability density amplitude that flows from state q to state q' upon reading symbol σ. The state q_0 is the initial state of the system and F is the set of accepting states. Equation 2.2 requires that the operators corresponding to the transition function δ be unitary.

The configuration of M is represented by an n-dimensional complex vector and is denoted by

$$|\Psi\rangle = \sum_{i=1}^{n} \alpha_i |q_i\rangle$$

where $n = |Q|$, $\{|q_i\rangle\}$ is the set of standard basis vectors corresponding to the states of M, α_i is the probability density amplitude of the machine being in state q_i, and $\sum_{i=1}^{n} |\alpha_i|^2 = 1$. Upon reading a symbol σ, M evolves in unitary fashion according to the equation

$$U_\sigma |\Psi\rangle = \sum_{q_i, q_j \in Q} \alpha_i \delta(q_i, \sigma, q_j) |q_j\rangle$$

where U_σ is a unitary operator represented by a unitary matrix.

If M is in configuration $|\Psi\rangle$ and reads string $x = x_1 x_2 ... x_m$, then the configuration of M, having read x is $U(x)|\Psi\rangle$ where $U(x) = U_{x_m} U_{x_{m-1}} ... U_{x_1}$. The initial configuration of M is denoted $|\Psi_0\rangle$ and usually corresponds to the machine starting out in state q_0. All input to M is in the form $x\$$ where $x \in (\Sigma \backslash \{\$\})^*$ and $\$$ is the end-marker. When M finishes reading the input string, it measures its configuration; if it is in an accepting state then it accepts. If M is in configuration $|\Psi_x\rangle$, having read x, then the probability of it accepting is given by

$$p_M(x) = \langle \Psi_x | P | \Psi_x \rangle = \| P | \Psi_x \rangle \|^2 = |\langle P | | \Psi_x \rangle|^2$$

where P is the projection operator onto the subspace spanned by the set $\{|q_i\rangle\}_{q_i \in F}$. The operator P is represented by a diagonal projection matrix where $P_{ii} = [q_i \in F]$ or the row vector $\langle P|$ where $P_i = [q_i \in F]$.

2.8 Definition of Measure-Many Quantum Finite Automata

A measure-many quantum finite automaton is defined by a 6-tuple $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ where Q is a finite set of states, Σ is the alphabet, δ is a unitary transition function of the same form as MO-QFAs, q_0 is the initial state, $Q_{acc} \subseteq Q$ is the set of accepting states, and $Q_{rej} \subseteq Q \backslash Q_{acc}$ is the set of rejecting states. The set Q is partitioned into three disjoint sets: Q_{acc}, Q_{rej} and $Q_{non} = Q \backslash (Q_{acc} \cup Q_{rej})$ where Q_{non} is the set of non-halting states.

The operation of an MM-QFA is similar to that of an MO-QFA except for one major difference. After a symbol is read, the MM-QFA measures whether or not it is in the subspace spanned by the set of accepting, rejecting, or non-halting states. Measuring the subspaces of the QFA collapses the superposition onto a particular subspace and leaves the superposition within the subspace unchanged except for a renormalization. If the measurement yields that M is in a halting subspace, i.e. accepting or rejecting, then M halts and accepts or rejects. Otherwise, the superposition in the non-halting subspace is renormalized and M proceeds to read the next symbol of its input. The transition for the end-marker \$ is required to transfer all amplitudes into the halting subspaces, guaranteeing that M will halt upon the measurement.

Similar to an MO-QFA, the configuration of an MM-QFA is denoted by an n-dimensional complex vector where $n = |Q|$. However, because M can have a non-zero probability of halting part-way through the computation, it is useful to keep track of the cumulative accepting and rejecting probabilities. Because of this, we sometimes use the representation defined by Kondacs and Watrous[KW97], which uses a triple $(|\Psi\rangle, p_{acc}, p_{rej})$ to represent the configuration of M. The evolution of M is denoted by

$$(P_{non} U_\sigma |\Psi\rangle, p_{acc} + \| P_{acc} U_\sigma |\Psi\rangle \|^2, p_{rej} + \| P_{rej} U_\sigma |\Psi\rangle \|^2)$$

where p_{acc} and p_{rej} are the cumulative probabilities of accepting and rejecting, and P_{acc}, P_{rej} and P_{non} are the diagonal projection matrices that project the configuration onto the non-halting, accepting or rejecting subspaces.

2.9 Language Acceptance by a Quantum Finite Automaton

A QFA M is said to accept a language L with cut-point λ if for all $x \in L$ the probability of M accepting x is greater than λ and for all $x \notin L$ the probability of M accepting x is at most λ. A QFA M accepts L with bounded error if there exists an $\epsilon > 0$ such that for all $x \in L$ the probability of M accepting x is greater than $\lambda + \epsilon$ and for all $x \notin L$ the probability of M accepting x is less than $\lambda - \epsilon$. We call the ϵ the margin. Hence, if the margin is greater than zero, then M accepts with bounded error. We look at QFAs with both types of acceptance, but we pay more attention to QFAs that accept with bounded error.

2.10 Equivalence of Quantum Finite Automata

We say that two QFAs are equivalent if they have the same acceptance probability distribution on the set of words over an alphabet Σ. Formally, QFAs M_1 and M_2 are equivalent if for every word $x \in \Sigma^*$ the probability of M_1 accepting x is equal to the probability of M_2 accepting x, i.e. $p_{M_1}(x) = p_{M_2}(x)$ for all $x \in \Sigma^*$.

2.11 Equivalence of MO-QFAs and Moore-Crutchfield QFAs

The QFA introduced by Moore and Crutchfield[MC98] is similar to the MO-QFA. A Moore-Crutchfield QFA $M = (\mathcal{H}, \Sigma, \{U_\sigma\}_{\sigma \in \Sigma}, |s_{init}\rangle, \mathcal{H}_a)$ consists of an n-dimensional Hilbert space $\mathcal{H}$, an initial configuration vector $|s_{init}\rangle$, an accepting subspace $\mathcal{H}_a$ with a corresponding projection matrix P_a, an input alphabet Σ, and a unitary matrix U_σ for each symbol in Σ. The probability of a Moore-Crutchfield automaton accepting a string x is defined as

$$f(x) = \|P_a U(x)|s_{init}\rangle\|^2$$

is almost the same as an MO-QFA automaton except that $|s_{init}\rangle$ may be an arbitrary unit vector, P_a is not necessarily a diagonal projection matrix and there is no end-marker \$. However, there is a correspondence that preserves acceptance probability distribution between

MO-QFAs and Moore-Crutchfield QFAs.

Lemma 2.2 *There is a correspondence that preserves acceptance probability distribution between MO-QFAs that have left and right string end-markers and Moore-Crutchfield QFAs.*

Proof: Let $M = (\mathcal{H}, \Sigma, \{U_\sigma\}_{\sigma\in\Sigma}, |s_{init}\rangle, \mathcal{H}_a)$ be a Moore-Crutchfield QFA. We construct an equivalent MO-QFA $M' = (Q, \Sigma \cup \{\not\!\!\!

Proof: Let $M = (Q, \Sigma, \delta, q_0, F)$ be an MO-QFA with left and right end-markers, effectively allowing M to start in any possible configuration. We now define $M' = (Q, \Sigma, \delta', q_0, F)$ from M. Let δ be defined in terms of the transition matrices $\{U_\sigma\}_{\sigma \in \Sigma}$ and we define δ' from δ in the following way; for every $\sigma \in \Sigma$ let

$$U'_\sigma = U_{\not\!\!\not}^{-1} U_\sigma U_{\not\!\!\not}$$

and let

$$U'_\$ = U_\$ U_{\not\!\!\not}$$

Now consider what happens when M and M' read a string $x = x_1 ... x_n$. The probability of M accepting x is

$$\begin{aligned} p_M(x) &= \|PU(\not\!\!\not x\$)|q_0\rangle\|^2 \\ &= \|PU_\$ U_{x_n} ... U_{x_1} U_{\not\!\!\not}|q_0\rangle\|^2 \end{aligned}$$

and the probability that M' accepts x is

$$\begin{aligned} p_{M'}(x) &= \|PU'(x\$)|q_0\rangle\|^2 \\ &= \|PU'_\$ U'_{x_n} ... U'_{x_1}|q_0\rangle\|^2 \\ &= \|PU_\$ U_{\not\!\!\not} U_{\not\!\!\not}^{-1} U_{x_n} U_{\not\!\!\not} U_{\not\!\!\not}^{-1} U_{x_{n-1}} U_{\not\!\!\not} ... U_{\not\!\!\not}^{-1} U_{x_1} U_{\not\!\!\not}|q_0\rangle\|^2 \\ &= \|PU_\$ U_{x_n} ... U_{x_1} U_{\not\!\!\not}|q_0\rangle\|^2 \end{aligned}$$

is the same as M's. Thus one end-marker on the right suffices, and by symmetry one left end-marker would also suffice. Therefore, an MO-QFA starting in configuration $|q_0\rangle$ can simulate an MO-QFA starting in any arbitrary configuration. □

The converse of this follows by making the left end-marker transition matrix the identity matrix. This yields the corollary that there is a correspondence that preserves acceptance probability distribution between Moore-Crutchfield QFAs and MO-QFAs.

Corollary 2.4 *There is a correspondence that preserves acceptance probability distribution between Moore-Crutchfield QFAs and MO-QFAs.*

Proof: Follows directly from Lemma 2.2 and Theorem 2.3. □

In some of the proofs we assume that the MO-QFA starts in a configuration other than $|q_0\rangle$. However, by Theorem 2.3 we know that such automata are equivalent to ones that can only start in configuration $|q_0\rangle$.

2.12 Equivalence of MM-QFAs and Kondacs-Watrous QFAs

The definition of an MM-QFA is almost identical to the model defined by Kondacs and Watrous[KW97]. The only difference is that the Kondacs-Watrous QFA requires both a left end-marker ¢ and a right end-marker \$. Our definition of the MM-QFA only requires one end-marker at the right end of the tape. The left end-marker is unnecessary and is an artifact from their treatment of 2-way quantum finite automata.

Theorem 2.5 (End-Marker Theorem for MM-QFA) *Let M be an MM-QFA that has both left and right end-markers, then there exists an MM-QFA M' that uses only the right end-marker and is equivalent to M.*

Proof: Let $M = (Q, \Sigma, \delta, Q_{acc}, Q_{rej})$ be an MM-QFA that uses two end-markers and accepts L. Assume without loss of generality that

$$\begin{aligned} Q_{non} &= \{q_i \in Q \mid 0 \leq i < n_{non}\} \\ Q_{acc} &= \{q_i \in Q \mid n_{non} \leq i < n_{acc}\} \\ Q_{rej} &= \{q_i \in Q \mid n_{acc} \leq i < n_{rej} = n = |Q|\} \end{aligned}$$

which allows a simpler description of M'. We now construct $M' = (Q', \Sigma, \delta', Q'_{acc}, Q'_{rej})$ that accepts L with only the right end-marker. Let $Q' = Q \cup \{q_n, q_{n+1}, ..., q_{2n-n_{non}}\}$, $Q'_{acc} = \{q_{n+i-n_{non}} \in Q' \mid q_i \in Q_{acc}\}$ and $Q'_{rej} = \{q_{n+i-n_{non}} \in Q' \mid q_i \in Q_{rej}\}$. Assume that δ is

defined in terms of transition matrices $\{U_\sigma\}_{\sigma\in\Sigma}$. The construction of $\{U'_\sigma\}_{\sigma\in\Sigma}$ is similar to that in the proof of Theorem 2.3. Let I_l represent an identity matrix of size l and $m = n - n_{non}$. We now define δ' in terms of its unitary block matrices. For all $\sigma \in \Sigma$ let

$$U'_\sigma = \begin{bmatrix} U_{¢}^{-1} & & \\ & & \\ & & I_m \end{bmatrix} U_{sweep} \begin{bmatrix} U_\sigma & & \\ & & \\ & & I_m \end{bmatrix} \begin{bmatrix} U_{¢} & & \\ & & \\ & & I_m \end{bmatrix}$$

$$U'_\$ = U_{sweep} \begin{bmatrix} U_\$ & & \\ & & \\ & & I_m \end{bmatrix} \begin{bmatrix} U_{¢} & & \\ & & \\ & & I_m \end{bmatrix}$$

where

$$U_{sweep} = \begin{bmatrix} I_{n_{non}} & & \\ & & I_m \\ & I_m & \end{bmatrix}$$

transfers all probability amplitude from states in the old halting states to the new halting states. The old halting states, those in Q_{acc} and Q_{rej}, are no longer halting states in M'. The operation of M' is similar to the operation of the QFA constructed in Theorem 2.3, The 'sweeping' operation saves the amplitude that was in the old states, while it performs the $U_{¢}^{-1}$ operation in the new halting states, otherwise the $U_{¢}^{-1}$ corrupts the amplitude stored in the original halting states. □

The converse of this theorem follows from having the left end-marker transition be the identity. In some of our proofs we assume, for convenience, that MM-QFAs can start in an arbitrary configuration because we know that there exists an equivalent MM-QFA that starts in configuration $|q_0\rangle$.

Chapter 3

Measure-Once Quantum Finite Automata

Measure-once quantum finite automata can either be restricted to accepting with bounded error or unrestricted. Accordingly, for each model there is a corresponding class of languages that the model accepts. Define the $\mathbf{RMO}_\epsilon$ as the class of languages which can be accepted by an MO-QFA with a margin of at least ϵ. Define the 'unrestricted' class of languages as $\mathbf{UMO} = \mathbf{RMO_0}$ and define the 'restricted' class of languages $\mathbf{RMO} = \bigcup_{\epsilon>0} \mathbf{RMO}_\epsilon$.

In Section 3.1 we completely characterize the class of languages accepted by restricted MO-QFAs. In Section 3.2 we prove some general closure properties of the languages accepted by both the restricted and unrestricted model. Section 3.3 proves that the unrestricted model can accept non-regular sets. Section 3.4 describes an algorithm to determine if two MO-QFAs are equivalent and Section 3.5 proves that the set of languages accepted by MO-QFAs is a subset of the languages accepted by a PFA.

3.1 Characterization of MO-QFAs That Accept with Bounded Error

The restriction of bounded error on MO-QFAs is, not surprisingly, as limiting as in the case of PFAs[Rab63]. Kondacs and Watrous[KW97] showed that MM-QFAs can only accept a proper subset of regular languages if they are required to accept with bounded error. We show in Theorem 4.6 that every MO-QFA can be simulated exactly by a MM-QFA. This limits the class **RMO** to a proper subset of regular languages. The class **RMO** is exactly the class of languages accepted by GFAs, otherwise known as group languages. This completely characterizes MO-QFAs that accept with bounded error. We prove this result in the rest of this section, starting with Lemma 3.1.

Lemma 3.1 *Let U be a unitary matrix. For any $\epsilon > 0$ there exists an integer $n > 0$ such that for all vectors x, where $\|x\|^2 < 1$, it is true that $\|(I - U^n)x\|^2 < \epsilon$.*

Proof: Let $m = dim(U)$. Since U is a normal matrix and has eigenvalues that are of the form $e^{i\theta}$, U^n can be written as

$$U^n = PD^nP^{-1}$$

where P is a unitary matrix and D is the diagonal matrix of eigenvalues with the jth eigenvalue having the form $e^{i\pi r_j}$. If all eigenvalues in D are rotations through rational fractions of π, i.e. r_j is rational, then let $n = 2\prod_{j=1}^{m} q_j$ where q_j is the denominator of r_j. Thus $D^n = I$ and we are done.

Otherwise, at least one eigenvalue is a rotation of unity through an irrational fraction of π. Let $l \leq m$ be the number of these eigenvalues. For the other $m - l$ eigenvalues compute n, just as above, and let $D' = D^n$. The values on the diagonal of D' are either 1 or $e^{i\pi n r_j}$ where r_j is some irrational real number. Consider taking D' to some power $k \in \mathbb{Z}^+$. The values that are 1 do not change, but the other l values, which are of the form $e^{i\theta_j k}$ where $\theta_j = \pi n r_j$, represent a vector that rotates in an l-dimensional dense torus. Hence, there exists k such that the l-dimensional vector is arbitrarily close to $\vec{1}$. Thus for any $\epsilon' > 0$ there exists a $k > 0$ such that $\|(I - D'^k)\vec{1}\|^2 < \epsilon'$. Hence

$$\begin{aligned}
\|(I - U^{nk})x\|^2 &= \|(I - PD^{nk}P^{-1})x\|^2 \\
&= \|P(I - D'^k)P^{-1}x\|^2 \\
&\leq \|(I - D'^k)m\vec{1}\|^2 \\
&= m^2\|(I - D'^k)\vec{1}\|^2 \\
&\leq m^2\epsilon'
\end{aligned}$$

Select ϵ' such that $\epsilon' < \frac{\epsilon}{m^2}$ to complete the proof. □

If two configurations are close, then the differences in probability distributions of the configurations is also small. The following lemma, due to Bernstein and Vazirani[BV97],

allows us to relate the closeness of configurations to the variance in their probability distributions. This is necessary in order for us to partition the set of allowable configurations into equivalence classes.

Lemma 3.2 (Bernstein and Vazirani, 1997) *Let $|\psi\rangle$ and $|\varphi\rangle$ be two complex vector such that $\||\psi\rangle\|^2 = \||\varphi\rangle\|^2 = 1$ and $\||\psi\rangle - |\varphi\rangle\|^2 < \epsilon$. Then the total variation distance between the probability distributions resulting from measurement of $|\psi\rangle$ and $|\varphi\rangle$ is at most 4ϵ.*

Using these two lemmas, we prove our main result of this section.

Theorem 3.3 *A language L can be accepted by a MO-QFA with bounded error if and only if it can be accepted by a GFA.*

Proof: The first direction is easy. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a GFA that accepts L. Let δ be defined in terms of transition matrices. Since the transition matrices of a GFA are permutation matrices, and permutation matrices are unitary, the same matrices can be used in the definition of a MO-QFA. Since the matrices are permutation matrices, the MO-QFA will accept L with certainty.

For the other direction, by contradiction, assume that there exists a language L that can be accepted by a MO-QFA with bounded error but cannot be accepted by a GFA. Since the class **RMO** is a subset of the regular languages, L must be regular. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a MO-QFA that accepts L with bounded error, by the Myhill-Nerode theorem[HU79] the space of configurations of M's computation can be partitioned into a finite number of equivalence classes. We denote such an equivalence class $[x]$ where $x \in \Sigma^*$ and $[x] \sim_L [x']$ if for all $y \in \Sigma^*$, $xy \in L$ if and only if $x'y \in L$. Since L cannot be accepted by a GFA, there must exist two distinct equivalence classes $[y]$ and $[y']$, an equivalence class $[x]$, and a symbol σ, such that $[y\sigma] \sim_L [y'\sigma] \sim_L [x]$. If U_σ is the transition matrix for symbol σ, $|\psi\rangle \in [y]$ and $|\varphi\rangle \in [y']$ then $U_\sigma|\psi\rangle \in [x]$ and $U_\sigma|\varphi\rangle \in [x]$.

Since M accepts L with bounded error, let ϵ be the margin. By Lemma 3.1 there exists an integer $k > 0$ such that $\|(I - U_\sigma^k)|\psi\rangle\|^2 < \frac{\epsilon}{4}$ and $\|(I - U_\sigma^k)|\varphi\rangle\|^2 < \frac{\epsilon}{4}$. Hence, $U_\sigma^k|\psi\rangle \in [y]$

because if

$$\begin{aligned} \|(I - U_\sigma^k)|\psi\rangle\|^2 &= \||\psi\rangle - U_\sigma^k|\psi\rangle\|^2 \\ &= \|V(|\psi\rangle - U_\sigma^k|\psi\rangle)\|^2 \\ &< \frac{\epsilon}{4} \end{aligned}$$

where V is an arbitrary unitary matrix, then by Lemma 3.2 the probability of $VU_\sigma^k|\psi\rangle$ being measured in a particular state is within ϵ of $V|\psi\rangle$ being measured in the same state; this probability is less than the margin. Similarly $U_\sigma^k|\varphi\rangle \in [y']$. Hence $[y] \sim_L [y\sigma^k]$ and $[y'] \sim_L [y'\sigma^k]$.

We assumed $[x] \sim_L [y\sigma] \sim_L [y'\sigma]$ and showed that $[y] \sim_L [y\sigma^k]$ and $[y'] \sim_L [y'\sigma^k]$; therefore, $[y] \sim_L [x\sigma^{k-1}] \sim_L [y']$. Let z be the string that distinguishes $[y]$ and $[y']$. Therefore $\sigma^{k-1}z$ partitions $[x]$ into at least two distinct equivalence classes, but this is a contradiction. Hence, there cannot exist a language L that can be accepted by a MO-QFA with bounded error but not by a GFA. □

Theorem 3.3 implies that $\mathbf{RMO}_\epsilon = \mathbf{RMO}_{\epsilon'}$ for all $\epsilon > 0$, hence there are most two distinct classes of languages accepted by MO-QFAs, the restricted class **RMO**, which is equivalent to the class of languages accepted by a GFA, and the unrestricted class **UMO**.

3.2 Closure Properties

One important characteristic of a set of languages is its closure properties. A class is closed under an operation if applying the operation to any language in the class yields a language in the same class. The fact that a class is closed under a particular operation is called a closure property of that class.

The common operations on languages include boolean operations, homomorphisms, inverse homomorphisms, and word quotients. Moore and Crutchfield[MC98] showed that the unrestricted class of languages accepted by their model was closed under boolean operations and inverse homomorphisms. By Corollary 2.4 it follows that the class **UMO** is also closed

under boolean operations and inverse homomorphisms. Lemma 3.4 shows that their result also holds for the class **RMO**.

Lemma 3.4 *The class* **RMO** *is closed under boolean operations and inverse homomorphisms.*

Proof: By Theorem 3.3 the class **RMO** is the same as the class of group languages. Since the class of group languages is closed under boolean operations and inverse homomorphisms[Eil76], the result follows. □

Since both classes, **RMO** and **UMO**, are closed under inverse homomorphisms, it follows in Theorem 3.5 that they are also closed under left and right word quotients.

Theorem 3.5 *The classes* **RMO** *and* **UMO** *are closed under left and right quotient.*

Proof: Let $M = (Q, \Sigma, \delta, q_0, F)$ be a MO-QFA with a left and right marker that accepts L and let $w \in \Sigma^+$ be the quotient word. We show that there exists MO-QFAs M' and M'' that accept $w^{-1}L$ and Lw^{-1} with the same cut-point and margin as M. For the left quotient define the homomorphism

$$h(\sigma) = \begin{cases} \not\in w & \sigma = \not\in \\ \sigma & \text{otherwise} \end{cases}$$

The inverse homomorphism of the L is precisely $w^{-1}L$ and by Lemma 3.4 there exists a MO-QFA M' that accepts $w^{-1}L$ with the same cut-point and margin as M accepting L. For right quotient use the homomorphism

$$h(\sigma) = \begin{cases} \$w & \sigma = \$ \\ \sigma & \text{otherwise} \end{cases}$$

that yields Lw^{-1} in the same way. □

The class **RMO** is not closed under homomorphisms. We proved in Section 3.1 that the class **RMO** is exactly the set of languages accepted by a GFA. Using this result, we can construct a GFA whose language, under a simple homomorphism, is not a group language.

Theorem 3.6 *The class* **RMO** *is not closed under homomorphisms.*

Proof: Consider GFA M in Figure 3.1 and the language L that it accepts. Let h be a homomorphism

$$h(a) = a, \quad h(b) = b, \quad h(c) = b$$

let $L' = h(L)$, and let M' be the automaton that accepts L'.

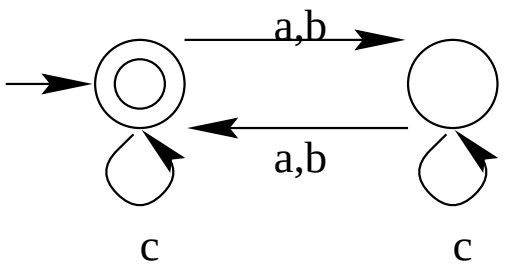

Figure 3.1: GFA M

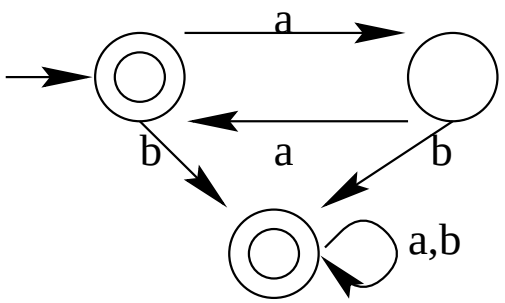

Figure 3.2: RFA M'

The minimal automaton that accepts L' is found in Figure 3.2 and is not a GFA but an RFA. Since the class **RMO** is exactly the class of languages accepted by a GFA, the language L is accepted by a GFA, but $h(L)$ is not. It follows that the class **RMO** is not closed under homomorphisms. □

3.3 MO-QFAs Can Accept Non-Regular Sets

Measure-once quantum finite automata that accept with bounded error can only accept a proper subset of regular languages, but using a technique similar to Rabin's[Rab63] we show that MO-QFAs can accept non-regular languages with unbounded error. This is similar to PFAs that can only accept regular languages when restricted to accepting with bounded error but can accept non-regular languages with unbounded error.

Lemma 3.7 *Let* $L = \{x \in \{a,b\}^* \mid |x|_a = |x|_b\}$, *there exists a 2 state MO-QFA* M *that accepts* x *with certainty if* $x \in L$ *and rejects* x *with a probability strictly greater than* 0 *if* $x \notin L$.

Proof: Let $M = (Q, \Sigma, \delta, q_0, F)$ where $Q = \{q_0, q_1\}$, $\Sigma = \{a, b\}$, $F = \{q_0\}$, and δ is defined by the transition matrices

$$U_a = U_b^{-1} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$

where α is an irrational fraction of π. Since U_a is a rotation matrix and α is an irrational fraction of π, then the orbit formed by applying U_a to $|q_0\rangle$ is a dense circle, and there exists only one k, such that $U_a^k|q_0\rangle = |q_0\rangle$, namely $k = 0$. This is also true for $U_a^{-1} = U_b$. Hence, $U(x)|q_0\rangle = |q_0\rangle$ if and only if the number of U_a rotations applied to $|q_0\rangle$ is equal to the number of U_b rotations. This only happens if the number of a's read is equal to the number of b's read. Otherwise, $|q_1\rangle$ will have a non-zero amplitude and a non-zero probability of rejection. In this case the end-marker is simply the identity operation because it is not necessary. Since L is a non-regular language, the result follows. □

Lemma 3.7 implies that the class **RMO** is properly contained within the class **UMO** and therefore the two classes are distinct.

A free group is a group in which there is no relation on the generators. The word problem over the free group is the problem of deciding whether a product of elements from the free group is equal to the identity. Concretely, let $\Sigma = \{1, a, a^{-1}, b, b^{-1}, ...\}$ be a finite alphabet

whose elements form a free group. Let $w \in \Sigma^*$ represent a composition of a finite number of elements. Define $L = \{w \in \Sigma^* \mid w = w_1 w_2 ... w_n = 1\}$ to be the set of words over Σ such that the product of the elements in the word is equal to the identity. If an automaton can accept L then it solves the word problem over the free group.

The MO-QFA in the above lemma solves the word problem for the free group. In this case the free group is generated by two elements $\{a, a^{-1} = b\}$. We can generalize this result to the general word problem for the free group.

Lemma 3.8 *The word problem for the free group language can be accepted by an MO-QFA.*

Proof: Construct a free group of rotation matrices from SO_3 as discussed by Wagon[Wag85]. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a 3-state MO-QFA where $\Sigma = \{a, a^{-1}, b, b^{-1}, ...\}$ such that $|\Sigma|$ is equal to the sum of the number of rotation matrices and their inverses, δ is defined by the rotation matrices and their inverses, and $F = \{q_0\}$. The MO-QFA will accept identity words with certainty and reject non-identity words with a strictly non-zero probability, hence solving the word problem for the free group. □

3.4 Equivalence of MO-QFAs

In classical automata theory there is an algorithm to determine if two automatons are equivalent. We face a similar problem with respect to MO-QFAs. Given two MO-QFAs we would like to determine if they are equivalent or not. Using a technique described by Paz[Paz71] and combining it with a preprocessing step developed by Moore and Crutchfield[MC98] we construct a method for testing the equivalence of two MO-QFAs. The key problem with MO-QFAs, unlike PFAs, is that measurement is a not bilinear operation. In PFAs the probability of some PFA M accepting string x is given by

$$p_M(x) = \pi A(x) \eta$$

and is bilinear. In the case of MO-QFAs the probability of a MO-QFA M' accepting string x

$$p_{M'}(x) = \|PU(x)|\Psi_0\rangle\|^2$$

is not bilinear.

The first step is to bilinearize the MO-QFA. Moore and Crutchfield[MC98] showed that any language accepted by a Moore-Crutchfield QFA is a GCE and provided a construction for the corresponding pseudo-stochastic system.

Theorem 3.9 (Linearization Theorem, Moore and Crutchfield, 1998) *If L is a language accepted by an n-state MO-QFA M with some cut-point λ then L is a GCE and can be accepted by a pseudo-stochastic system with the same cut-point and with $2n^2$ states.*

Proof: Let $M = (Q, \Sigma, \delta, q_0, F)$ be a MO-QFA with n-states that accepts L. Let $M' = (Q', \Sigma, \delta', q_0', F')$ be an automaton where $Q' = Q \times Q$, $q_0' = (q_0, q_0)$, $F' = \{(q_i, q_j) \in Q' \mid q_i \in F \wedge q_j \in F\}$ and δ' is defined in terms transition matrices $\{U'_\sigma\}_{\sigma\in\Sigma}$ where $U'_\sigma = (U^*_\sigma)^{-1} \otimes U_\sigma^{-1}$ and U_σ are the transition matrices defining δ. Let $\langle P|$ be the row vector corresponding to the accepting states of M, let π be the initial row vector representing $|(q_0, q_0)\rangle$, and let η be the final state vector where $\eta_i = [q_i' \in F']$. The probability of M accepting string x is

$$\begin{aligned}
p_M(x) &= \|PU(x)|q_0\rangle\|^2 \\
&= |\langle P|U(x)|q_0\rangle|^2 \\
&= |\langle q_0|U^{-1}(x)|P\rangle|^2 \\
&= \langle q_0^* \otimes q_0|(U^{-1})^* \otimes U^{-1}|P^* \otimes P\rangle \\
&= \pi U'(x)\eta \\
&= p_{M'}(x)
\end{aligned}$$

and $p_{M'} = \pi U'(x)\eta$ is bilinear. The new automaton has n^2 states but also might have non-real entries in its transition function. Since any complex number $a + ib$ can be represented by the

matrix

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix}$$

we can construct a $2n^2$-state pseudo-stochastic system that is equivalent to M', is bilinear and uses real entries in its transition matrices. Let $\pi' = (\pi, \vec{0})$, let A_σ correspond to U'_σ where each entry in U'_σ is replaced by a 2×2 real matrix defined above, and let η' be the acceptance vector such that $\eta'_{2i} = [q'_i \in F']$. This is a pseudo-stochastic system $S = (\pi', \{A_\sigma\}_{\sigma \in \Sigma}, \eta')$ that accepts L with the same cut-point as M. Hence, L is a GCE and S has $2n^2$ states. □

Given two MO-QFAs M and M' with m and n states respectively we use the method of Moore and Crutchfield[MC98] to convert them into pseudo-stochastic systems $N = (\pi, \{A_\sigma\}, \eta)$ and $N' = (\pi', \{A'_\sigma\}, \eta')$ with $2m^2$ and $2n^2$ states respectively. We now construct a $2m^2 + 2n^2$ state pseudo-stochastic system $N'' = (\pi'', \{A''_\sigma\}, \eta'')$ where

$$\pi'' = [\pi, \pi'],\ \eta'' = \begin{bmatrix} \eta \\ -\eta' \end{bmatrix},\quad A''_\sigma = \begin{bmatrix} A_\sigma & 0 \\ 0 & A'_\sigma \end{bmatrix}$$

The two automata M and M' are equivalent if for all $x \in \Sigma^*$ the system N'' always returns 0, i.e.

$$p_{N''}(\Sigma^*) = \pi'' A''(\Sigma^*) \eta'' = 0$$

This reduces the MO-QFA equivalence problem to the initial distribution equivalence problem, which is, given two initial distributions and a pseudo-stochastic system are the two distributions equivalent, i.e. do they yield the same probability for each word in Σ^*. To solve this problem we look at all vectors, ordered lexicographically with respect to $x \in \Sigma^*$, of the form $A''(x)\eta''$ where $|x| \leq d = 2m^2 + 2n^2$ and select the maximal independent set in lexicographic order. Paz[Paz71] showed that looking at all vectors generated by words of length d or less is sufficient to find a basis B for the smallest space that contains the vector set $\{A''(\Sigma^*)\eta''\}$. We now get the following theorem, which is a modification of a corollary of Paz[Paz71].

Theorem 3.10 *Let π and ρ be two initial distributions, let $\{A_\sigma\}$ the set of matrices, let η be the final state vector, and let B be the basis for the minimal space that contains $\{A(\Sigma^*)\}$ generated by the above method. If $\pi v = \rho v$ for all $v \in B$ then $\pi A(x)\eta = \rho A(x)\eta$ for all $x \in \Sigma^*$.*

Proof:

$$\pi A(x)\eta = \pi \sum_{v_i \in B} a_i v_i = \sum_{v_i \in B} a_i \pi v_i = \sum_{v_i \in B} a_i \rho v_i = \rho \sum_{v_i \in B} a_i v_i = \rho A(x)\eta$$

□

This leads us to an algorithm to determine the equivalence of two MO-QFAs.

Theorem 3.11 *Given two MO-QFAs M and M' we can determine if they are equivalent.*

Proof: Using the Moore-Crutchfield method we convert M and M' into pseudo stochastic systems N and N'. We construct a new pseudo stochastic system N'' as detailed above. We compare the initial distribution π'' to the zero distribution $\vec{0}$. If the two are equivalent then we know that N and N' have the same distribution because N'' subtracts the second from the first. This yields the constant 0 if and only if the two are the same. Thus, we can determine if two MO-QFAs are equivalent. □

3.5 Simulation of MO-QFAs by PFAs

Most classical computation is either deterministic or probabilistic, hence it is useful to ask how probabilistic automata compare to their quantum analogs. In the case of MO-QFAs, any language accepted by an MO-QFA can also be accepted by a PFA. If L can be accepted by an MO-QFA with bounded error, then it be accepted by a PFA with bounded error.

Theorem 3.12 *Let M be a MO-QFA that accepts L with cut-point λ then*

1. *There exists a PFA that accepts L with some cut-point λ'.*
2. *If M accepts L with bounded error, then there exists a PFA that accepts L with bounded error.*

Proof: The second result follows from Theorem 3.3 that states that that the set of languages accepted by an MO-QFA with bounded error is a proper subset of the regular language. Since PFAs can accept regular sets with zero error, the second result follows.

By Theorem 3.9 if L is accepted by an MO-QFA then L is a GCE. By Theorem 2.1 the class of GCEs is equal to the class of PCEs[Paz71], which are accepted by PFAs. Hence there exists a PFA that can accept the same language as M with some cut-point λ'. □

This theorem along with Lemma 3.8 shows that PFAs can solve the word problem for the free group, we get the following result.

Corollary 3.13 *The word problem for the free group language can be solved by a PFA.*

Chapter 4

Measure-Many Quantum Finite Automata

Measure-many quantum finite automata are more powerful than MO-QFAs because a measurement is performed after every transition. This allows the machine to terminate before reading the entire string and simulate spin states. Just as in the case of MO-QFAs, MM-QFAS can accept without or with bounded error. Let $\mathbf{RMM}_\epsilon$ be the class of languages that can be accepted by an MM-QFA where the allowed margin is at least ϵ. Let $\mathbf{UMM} = \mathbf{RMM_0}$ be the 'unrestricted' class of languages accepted by an MM-QFA and let $\mathbf{RMM}$ be the class of 'restricted' languages where $\mathbf{RMM} = \bigcup_{\epsilon>0} \mathbf{RMM}_\epsilon$.

In Section 4.1 we prove several closure properties of both classes of languages. Section 4.2 shows how MM-QFAs can simulate MO-QFAs exactly. In Section 4.3 we show that MM-QFAs are strictly more powerful then MO-QFAs. We then focus our attention on MM-QFAs that accept with bounded error. Section 4.4 proves a necessary condition for a language to be in the class $\mathbf{RMM}$. In Section 4.5 we prove that piecewise testable sets are members of the class $\mathbf{RMM}$. In the same section we introduce a couple of novel techniques for constructing MM-QFAs and note some special characteristics of the construction that could be applicable elsewhere.

4.1 Closure Properties

The languages accepted by MM-QFAs belong to either $\mathbf{RMM}$ or $\mathbf{UMM}$, just like MO-QFAs. However, it is unknown whether these classes are closed under the same operations as the classes $\mathbf{RMO}$ and $\mathbf{UMO}$ which contain the languages accepted by MO-QFAs. In this section we show that the classes $\mathbf{RMM}$ and $\mathbf{UMM}$ are closed under complement, inverse

homomorphisms, and word quotient. However, it is still unknown whether the classes are closed under intersection, and hence, under boolean operations. We do know however that the restricted class **RMM** is not closed under homomorphisms, which is similar to the restricted class of languages accepted by MO-QFAs. We prove closure for the class $\mathbf{RMM}_\epsilon$ for all $\epsilon \geq 0$, under complement, inverse homomorphisms, and word quotient in Theorem 4.1. This implies that the classes **RMM** and **UMM** are also closed under the same operations

Theorem 4.1 *The class* $\mathbf{RMM}_\epsilon$ *is closed under complement, inverse homomorphisms, and word quotient.*

Proof: Let M be an MM-QFA that accepts L with cut-point λ and margin ϵ. Let $\overline{M}$ be the same MM-QFA except that the accept and reject states are transposed, i.e. all accept states become reject states and all reject states become accept states. Hence, the probability of $\overline{M}$ accepting a string $x \in L$ is

$$p_{\overline{M}}(x) \leq 1 - (\lambda + \epsilon) = \overline{\lambda} - \epsilon$$

and the probability of $\overline{M}$ accepting $x \notin L$ is

$$p_{\overline{M}}(x) > 1 - (\lambda - \epsilon) = \overline{\lambda} + \epsilon$$

where $\overline{\lambda} = 1 - \lambda$. Therefore $\overline{M}$ accepts $\overline{L}$ with cut-point $\overline{\lambda}$ and a margin of ϵ. This implies that the class $\mathbf{RMM}_\epsilon$ is closed under complement.

Given M and a homomorphism h we construct a machine M' that accepts $h^{-1}(L)$. Let $M = (Q, \Sigma, \delta, Q_{acc}, Q_{rej})$ and $M' = (Q', \Sigma, \delta', Q'_{acc}, Q'_{rej})$. Assume that δ and δ' are defined in terms of matrices $\{U_\sigma\}_{\sigma\in\Sigma}$ and $\{U'_\sigma\}_{\sigma\in\Sigma}$. Unlike the proof for MO-QFAs in [MC98], the direct construction of

$$U'_\sigma = U(h(\sigma))$$

will not work unless h only maps symbols to symbols, instead of the general case of symbols to strings. This is because a measurement occurs between transitions, and combining transitions without taking this into account could produce incorrect configurations. Specifically after every

transition some amount of probability amplitude is placed in the halting states. This probability amplitude is not allowed to interact with the non-halting states in the following transitions. However, because the transitions are unitary, interaction could occur. To solve this problem we use extra states to store the amplitude while transition combination $U(h(\sigma))$ is executed so that it does not interact with the amplitude in the non-halting states. Assume without loss of generality that

$$\begin{aligned} Q_{non} &= \{q_i \in Q \mid 0 \leq i < n_{non}\} \\ Q_{halt} &= \{q_i \in Q \mid n_{non} \leq i < n\} \end{aligned}$$

where $n = |Q|$ and $n_{non} = |Q_{non}|$. Let $m = \max_{\sigma \in \Sigma}\{|h(\sigma)|\}$, i.e the maximum length of all strings that h maps to. Let

$$Q' = Q \cup Q'_{halt}$$

where

$$\begin{aligned} Q'_{halt} &= \{q_i\}_{i=n+1}^{n+m(n-n_{non})} \\ Q'_{acc} &= Q_{acc} \cup \{q_{n+j(i-n_{non})} \in Q'_{halt} \mid q_i \in Q_{acc}, 1 \leq j \leq m\} \\ Q'_{rej} &= Q_{rej} \cup \{q_{n+j(i-n_{non})} \in Q'_{halt} \mid q_i \in Q_{rej}, 1 \leq j \leq m\} \end{aligned}$$

Intuitively, we clone the halting states m times. Hence, M' has $m+1$ copies of the halting sets, starting with set 0, the original set, followed by m copies. To complete the construction, we now define δ' in terms of the matrices that define δ. Let V_σ be a unitary block matrix

$$V_\sigma = U_{shift} \begin{bmatrix} U_\sigma & \\ & I_{m(n-n_{non})} \end{bmatrix}$$

where

$$U_{shift} = \begin{bmatrix} I_{n_{non}} & & \\ & & I_{n-n_{non}} \\ & I_{m(n-n_{non})} & \end{bmatrix}$$

The matrix U_{shift} is a unitary matrix that shifts the amplitudes in the halting set i to the halting set $i+1$ and the amplitude in halting set m to halting set 0. We are now ready to construct U'_σ. In analogy to the MO-QFA case where $U'_\sigma = U(h(\sigma))$, for MM-QFAs where $h(\sigma) = x = x_1x_2...x_l$ and $l \leq m$, let

$$U'_\sigma = V(h(\sigma)) = V(x) = V(x_l)V(x_{l-1})...V(x_1)$$

After every x_i sub-transition the halting amplitude is shifted and stored in the $m+1$ halting sets of states. When the x_{i+1} sub-transition is done, the amplitude in halt state set 0 is zero, which means that no undesired interaction between the halting and non-halting states occurs. A minimum of m transitions must occur before halting set m contains non-zero amplitude, however m is the maximum length of the string that h maps to, which is sufficient to guarantee that halting set 0 will never receive non-zero amplitude from halting set m.

This construction delays the need for a measurement by storing the amplitude in an auxiliary set of states. If M is in configuration $(|\Psi\rangle, p_{acc}, p_{rej})$ and M' is in a corresponding configuration $(|\Psi'\rangle, p_{acc}, p_{rej})$ where $|\Psi'\rangle = |\Psi\rangle|\vec{0}\rangle$. The result of M' reading σ is equivalent to M reading $h(\sigma) = x = x_1x_2...x_l$. The resulting configuration of M is

$$\left(P_{non}\widehat{U}(l)|\Psi\rangle, p_{acc} + \sum_{i=1}^{l} \|P_{acc}\widehat{U}(i)|\Psi\rangle\|^2, p_{rej} + \sum_{i=1}^{l} \|P_{rej}\widehat{U}(i)|\Psi\rangle\|^2\right)$$

while the resulting configuration of M' is

$$\begin{aligned}
& \left(P'_{non}U'_\sigma|\Psi'\rangle, p_{acc} + \|P'_{acc}U'_\sigma|\Psi'\rangle\|^2, p_{rej} + \|P'_{rej}U'_\sigma|\Psi'\rangle\|^2\right) \\
= & \left(P'_{non}V(h(\sigma))|\Psi'\rangle, p_{acc} + \|P'_{acc}V(h(\sigma))|\Psi'\rangle\|^2, p_{rej} + \|P'_{rej}V(h(\sigma))|\Psi'\rangle\|^2\right) \\
= & \left(P'_{non}V(x)|\Psi'\rangle, p_{acc} + \|P'_{acc}V(x)|\Psi'\rangle\|^2, p_{rej} + \|P'_{rej}V(x)|\Psi'\rangle\|^2\right) \\
= & \left(P'_{non}V_{x_l}...V_{x_1}|\Psi'\rangle, p_{acc} + \|P'_{acc}V_{x_l}...V_{x_1}|\Psi'\rangle\|^2, p_{rej} + \|P'_{rej}V_{x_l}...V_{x_1}|\Psi'\rangle\|^2\right) \\
= & \left(P'_{non}\widehat{V}(l)|\Psi'\rangle, p_{acc} + \sum_{i=1}^{l} \|P'_{acc}\widehat{V}(i)|\Psi'\rangle\|^2, p_{rej} + \sum_{i=1}^{l} \|P'_{rej}\widehat{V}(i)|\Psi'\rangle\|^2\right) \\
= & \left(P'_{non}\widehat{V}(l)|\Psi\rangle|\vec{0}\rangle, p_{acc} + \sum_{i=1}^{l} \|P'_{acc}\widehat{V}(i)|\Psi\rangle|\vec{0}\rangle\|^2, p_{rej} + \sum_{i=1}^{l} \|P'_{rej}\widehat{V}(i)|\Psi\rangle|\vec{0}\rangle\|^2\right) \\
= & \left((P_{non}\widehat{U}(l)|\Psi\rangle)|\vec{0}\rangle, p_{acc} + \sum_{i=1}^{l} \|(P_{acc}\widehat{U}(i)|\Psi\rangle)|\vec{0}\rangle\|^2, p_{rej} + \sum_{i=1}^{l} \|(P_{rej}\widehat{U}(i)|\Psi\rangle)|\vec{0}\rangle\|^2\right)
\end{aligned}$$

where $\widehat{U}(i) = \prod_{j=1}^{i}(U_{x_{i+1-j}}P_{non})$ and $\widehat{V}(i) = \prod_{j=1}^{i}(V_{x_{i+1-j}}P'_{non})$. The new configuration of M' corresponds to the new configuration of M in exactly the same way as the original configurations. Therefore M' can accept $h^{-1}(L)$ with the same distribution as M, which means that $\mathbf{RMM}_{\epsilon}$ is closed under inverse homomorphisms.

The closure under word quotient follows from closure under inverse homomorphism, because of the presence of end-markers. For left quotient use the homomorphism

$$\begin{aligned} h(\not\epsilon) &= \not\epsilon w \\ h(\sigma) &= \sigma \\ h(\$) &= \$ \end{aligned}$$

to accept $w^{-1}L$ and for right quotient use the homomorphism,

$$\begin{aligned} h(\not\epsilon) &= \not\epsilon \\ h(\sigma) &= \sigma \\ h(\$) &= w\$ \end{aligned}$$

to accept the language Lw^{-1}. This is identical to the argument used in Theorem 3.5. □

Corollary 4.2 *The classes* **RMM** *and* **UMM** *are closed under complement, inverse homomorphisms, and word quotients.*

Just like the class **RMO**, the restricted class **RMM** is not closed under homomorphisms. This is proved in Theorem 4.5 and uses Lemma 4.3 and Lemma 4.4.

Lemma 4.3 (Kondacs and Watrous, 1997) *The language $L = \{a,b\}^*b$ cannot be accepted by an MM-QFA with bounded error.*

Lemma 4.4 (Ambainis and Freivalds, 1998) *If L can be accepted by a RFA, it can be accepted by an MM-QFA with certainty.*

Theorem 4.5 *The restricted class of languages accepted by MM-QFAs is not closed under homomorphisms.*

Proof: Let $L = \{a, b\}^* c$. Since this language can be accepted by an RFA, by Lemma 4.4 it can be accepted by an MM-QFA with bounded error. Define the homomorphism h as

$$\begin{aligned} h(a) &= a \\ h(b) &= b \\ h(c) &= b \end{aligned}$$

Applying this homomorphism to L we get $L' = h(L) = \{a, b\}^* b$. By Lemma 4.3 L' cannot be accepted by an MM-QFA with bounded error. The result follows. □

A more interesting question is whether the classes **RMM** and **UMM** are closed under boolean operations. Unlike MO-QFAs that have two types of states, accept and reject, MM-QFAs have three types of states, accept, reject, and non-halt. Because of this, the standard procedure of taking the tensor product of two automata to compose their intersection or union does not work. A general method of intersecting two MM-QFAs is not known. Thus, it is not known whether **RMM** and **UMM** are closed under boolean operations.

4.2 Simulation of MO-QFAs

The transition mechanism of an MM-QFA is more complex than that of an MO-QFA, therefore, intuitively, MM-QFAs should be more powerful than MO-QFAs. Ambainis and Freivalds[AF98] mentioned that MM-QFAs are a more general model than MO-QFAs but no proof was given. In Theorem 4.6 we show that an MO-QFA with n states can be simulated exactly by an MM-QFA with $2n$ states. To simulate exactly means that the distributions of both automata are identical.

Theorem 4.6 *For every MO-QFA M there exists an MM-QFA M' that has the same distribution as M.*

Proof: Let $M = (Q, \Sigma, \delta, q_0, F)$ be an MO-QFA and let $M' = (Q', \Sigma, \delta', q_0, Q'_{acc}, Q'_{rej})$ be an MM-QFA where $n = |Q|$, $Q' = Q \cup \{q_n, q_{n+1}, ..., q_{2n-1}\}$ is the set of states and, Q'_{acc} and Q'_{rej} are defined as

$$\begin{aligned} Q'_{acc} &= \{q_{n+i} \in Q' \mid q_i \in F\} \\ Q'_{rej} &= \{q_{n+i} \in Q' \mid q_i \notin F\} \end{aligned}$$

The set Q' consist of the original set of states, plus an additional state for each original state. Let δ be represented by the set of unitary matrices $\{U_\sigma\}_{\sigma \in \Sigma}$ and define δ' by the set of matrices $\{U'_\sigma\}_{\sigma \in \Sigma}$ where the matrix U'_σ is

$$U'_\sigma = \begin{bmatrix} U_\sigma & \\ & I_n \end{bmatrix}$$

and

$$U'_\$ = \begin{bmatrix} U_\$ & \\ & I_n \end{bmatrix} U_{flip}$$

where

$$U_{flip} = \begin{bmatrix} & I_n \\ I_n & \end{bmatrix}$$

The matrix U_{flip} exchanges all amplitude between the states in Q and their corresponding states in $Q' \backslash Q$. Thus, M' has a 0 probability of halting until the end-marker is read and has an identical distribution to M when it does halt. Thus M' simulates M exactly. □

4.3 MO-QFAs Are Strictly Less Powerful than MM-QFAs

By Theorem 4.6 it follows that the class **RMO**, the class of languages accepted with bounded error by MO-QFAs, is contained within the class **RMM**. Similarly the class **UMO**, the class of languages accepted with unbounded error by MO-QFAs, is contained within the class **UMM**. While we do not know if the latter containment is proper, Theorem 4.7 proves that the former containment is proper; this implies that MM-QFAs are strictly more powerful than MO-QFAs.

Theorem 4.7 *The class* **RMO** *is properly contained within the class* **RMM**.

Proof: Let L be the finite language $L = \{a\}$. Since L is a finite language it can be accepted by an RFA. By Lemma 4.4 L can be accepted by a MM-QFA with bounded error. Hence L is in the class **RMM**. By Theorem 3.3 the class **RMO** is identical to the class of languages accepted by a GFA. Since GFAs cannot accept finite languages, the result follows. □

4.4 MM-QFAs That Accept with Bounded Error

The restriction of bounded error acceptance reduces the class of languages that an MM-QFA can accept to a proper subclass of the regular languages[KW97]. To study the class of languages accepted by MM-QFAs with bounded error we look at the corresponding minimal automata for the languages. As we mentioned earlier, Ambainis and Freivalds[AF98] showed that if the minimal DFA $M = (Q, \Sigma, \delta, q_0, F)$ for language L contains an irreversible construction, defined by two distinct states $q_1, q_2 \in Q$ and strings $x, y, z \in \Sigma^+$ such that $\delta(q_1, x) = \delta(q_2, x) = q_2$, $\delta(q_2, y) \in F$ and $\delta(q_2, z) \notin F$, then an RFA cannot accept L and an MM-QFA cannot accept it with a probability greater than $\frac{7}{9}$. In fact, we have not been able to construct any MM-QFA that could accept a language not accepted by an RFA with probability greater than ≈ 0.68. Ambainis and Freivalds[AF98] showed that it is a necessary and sufficient condition that the minimal DFA corresponding to a language L cannot contain the above construction for L to be accepted by an RFA or by an MM-QFA with a probability greater than $\frac{7}{9}$.

We derive a similar necessary condition for a language L to be a member of the class **RMM**. This condition, called the partial order condition, is a relaxed version of a condition defined by Meyer and Thompson[MT69]. A language L is said to satisfy the partial order condition if the minimal DFA for L satisfies the partial order condition. A DFA satisfies the partial order condition if it does not contain two distinguishable states $q_1, q_2 \in Q$ such that there exists strings $x, y \in \Sigma^+$ such that $\delta(q_1, x) = \delta(q_2, x) = q_2$, and $\delta(q_2, y) = q_1$. States q_1

and q_2 are said to be distinguishable if there exists a string $z \in \Sigma^*$ such that $\delta(q_1, z) \in F$ and $\delta(q_2, z) \notin F$ or vice versa[HU79]. Theorem 4.8 proves that the partial order condition is necessary for an MM-QFA to accept L with bounded error.

Theorem 4.8 *If $M = (Q, \Sigma, \delta, q_0, F)$ is a minimal DFA for language L that does not satisfy the partial order condition then $L \notin \mathbf{RMM}$.*

Proof: By contradiction assume that $L \in \mathbf{RMM}$. Let $L_b = \{a, b\}^* b$. Since states q_1 and q_2 are distinguishable states in the minimal DFA, the Myhill-Nerode theorem tells us that there exists a string z that distinguishes the two states, i.e. $\delta(q_1, z) \notin F$ and $\delta(q_2, z) \in F$ or vice versa. Without loss of generality assume that $\delta(q_1, z) \notin F$ and $\delta(q_2, z) \in F$. (If the opposite is true then we can simply take the complement of L by flipping all the non-accepting states to accepting and all accepting states to non-accepting. This does not change the structure of M except for the accepting set.)

Since the minimal DFA for L does not satisfy the partial order condition there exist strings $x, y \in \Sigma^+$ as defined above and a distinguishing string $z \in \Sigma^*$. Let s be the shortest string such that $\delta(q_0, s) = q_1$. Let $L' = s^{-1} L z^{-1}$. By Theorem 4.1 $\mathbf{RMM}$ is closed under word quotient. It follows that $L' \in \mathbf{RMM}$. Define the homomorphism h as

$$\begin{aligned} h(a) &= xy \\ h(b) &= x \\ h(\Sigma - \{a, b\}) &= xy \end{aligned}$$

where the last definition is for completeness. Let $L'' = h^{-1}(L')$. By Theorem 4.1 $\mathbf{RMM}$ is closed under inverse homomorphisms, therefore the language $L'' \in \mathbf{RMM}$. But $L'' = L_b$, which by Lemma 4.3, is not in $\mathbf{RMM}$. Hence $L \notin \mathbf{RMM}$. □

The partial order condition is so named because once the state q_2 is visited, there is no path back to state q_1. Thus, there exists a partial order on the states of the DFA. At

the current time, we do not know whether or not this condition is sufficient for MM-QFA acceptance with bounded error.

As mentioned before, we do not know whether the class **RMM** is closed under boolean operations. However, Theorem 4.11 relates closure under intersection to the partial order condition. In order to prove it we first need a couple of lemmas.

Lemma 4.9 *Let M be a DFA that satisfies the partial order condition. The minimal DFA M' that accepts $L(M)$ satisfies the partial order condition.*

Proof: Let $M = (Q, \Sigma, \delta, q_0, F)$ and $M' = (Q', \Sigma, \delta', q_0', F')$. Assume by contradiction that M' does not satisfy the partial order condition. Hence, M' has two states that correspond to the equivalence classes $[q_1']$ and $[q_2']$ such that $[q_1']x \sim_L [q_2']x \sim_L [q_2']$ and $[q_2']y \sim_L [q_1']$. By the Myhill-Nerode theorem the equivalence classes partition the set Q. Hence, for each equivalence class $[q_i']$ there is a corresponding subset of Q. Let Q_1 and Q_2 denote the subsets of Q corresponding to the equivalence classes $[q_1']$ and $[q_2']$ and assign an arbitrary order on the subsets. Select the first state, say $p_1 \in Q_1$, and define the set $R = \{q \in Q_2 \mid \exists n, m \in \mathbb{Z}^+,\ \delta(p_1, x^m) = \delta(q, x^n) = q\}$. If there exists a state $r \in R$ and string $y \in \Sigma^+$ such that $\delta(r, y) = p_1$, then M does not satisfy the partial order condition, and this is a contradiction. Otherwise, there does not exist a $y \in \Sigma^+$ such that $\delta(r, y) = p_1$ for all $r \in R$. In this case there is a partial order on p_1 and on $Q_1 \backslash \{p_1\}$ because p_1 will never be visited again if M reads a sufficient number of xs. Remove p_1 from Q_1 and repeat the procedure on $p_2 \in Q_1$. After a finite number of iterations we will either find a p_i that satisfies our requirements, which means that M does not satisfy the partial order condition and is a contradiction, or none of the states in Q_1 will have the required characteristics, in which case M' satisfies the partial order condition. Hence, if M satisfies the partial order condition, so will its minimal equivalent M'. □

Lemma 4.10 *Let M' and M'' be minimal automata that satisfy the partial order condition and accept the languages L' and L'' respectively. The minimal automaton M that accepts $L = L' \cap L''$ also satisfies the partial order condition.*

Proof: Let $M' = (Q', \Sigma, \delta', q_0', F')$ and $M'' = (Q'', \Sigma, \delta'', q_0'', F'')$. We first construct an automaton M that accepts $L' \cap L''$ by combining M' and M'' using a direct product. Define $M = (Q, \Sigma, \delta, q_{00}, F)$ where $Q = Q' \times Q''$, $q_{00} = (q_0', q_0'')$, $F = \{(q', q'') \in Q \mid q' \in F' \wedge q'' \in F''\}$ and $\delta((q', q''), \sigma) = (\delta'(q', \sigma), \delta''(q'', \sigma))$.

We argue that if M' and M'' satisfy the partial order condition, then so will M. Assume, by contradiction, that M does not satisfy the partial order condition. Then there exists two states $q_{ij} = (q_i', q_j'')$ and $q_{kl} = (q_k', q_l'')$ and strings $x, y, z \in \Sigma^+$ such that $\delta(q_{ij}, x) = \delta(q_{kl}, x) = q_{kl}$, $\delta(q_{kl}, y) = q_{ij}$ and $\delta(q_{ij}, z) \in F$ if and only if $\delta(q_{kl}, z) \notin F$. In the first case assume that either $i \neq k$ or $j \neq l$, and without loss of generality, assume the former. Then there exists state $q_i' \in Q'$ and state $q_k' \in Q'$ such that $\delta'(q_i', x) = \delta'(q_k', x) = q_k'$, $\delta_1(q_k', y) = q_i'$. But this means that M' does not satisfy the partial order condition, a contradiction. In the second case assume that $i = k$ and $j = l$. This implies that $q_{ij} = q_{kl}$ and hence there cannot exist a string z that distinguishes the two states, also a contradiction. Therefore M must satisfy the condition.

Since M satisfies the partial order condition and accepts L, Lemma 4.9 proves that the minimal automaton that accepts L also satisfies the partial order condition. □

Theorem 4.11 *If the partial order condition is sufficient for acceptance with bounded error by MM-QFAs then the class* **RMM** *is closed under intersection.*

Proof: By Lemma 4.10 the intersection of two languages that satisfy the partial order condition is a language that satisfies the partial order condition. The result follows. □

A possible attempt to prove that the class **RMM** is not closed under intersection would involve intersecting two languages in **RMM** and showing that the resulting language is not. Theorem 4.11 shows that this method will not work unless it is shown that the partial order condition is insufficient. To study whether the partial order condition is sufficient, as well as necessary, we describe a naturally defined class of languages that can be accepted by an MM-QFA with bounded error.

4.5 MM-QFAs Can Accept Piecewise Testable Sets

A piecewise testable set is a boolean combination of sets of the form

$$L_x = \Sigma^* x_1 \Sigma^* x_2 \Sigma^* ... \Sigma^* x_n \Sigma_*$$

where $x_i \in \Sigma$ [Per94]. Intuitively, L_x is the language of strings that contain the subsequence x. We define a partial piecewise testable set as the language of strings that have a common subsequence; hence, a piecewise testable set is then a boolean combination of partial piecewise testable sets.

We show, by construction, that MM-QFAs can accept partial piecewise testable sets with bounded error. Furthermore, the MM-QFAs we construct are of a special kind we call 'end-decisive' and the languages accepted by end-decisive MM-QFAs with bounded error are closed under intersection, which implies that MM-QFAs can accept piecewise testable sets with bounded error. To construct the MM-QFAs we introduce several useful concepts: junk states, trigger chains and end-decisiveness.

A junk state is a halting state that does not contribute to the probability of accepting or rejecting a particular string. In general, any probability amplitude flowing into the junk state is split, usually evenly, between an accepting and rejecting state. That is, if α amplitude flows into a junk state, then $\frac{\alpha}{\sqrt{2}}$ amplitude flows into an accept state, and the same amount will flow into a reject state.

A machine is end-decisive if the probability of the machine accepting a string before reading the end-marker is the same irrespective of the string. The machine may halt before reading the full string, but it may only do so in a junk state because junk states do not contribute to the probability of accepting or rejecting a string. An end-decisive or decisive state is a halting state that contributes to the probability of a string being accepted or rejected. An end-decisive MM-QFA will only place amplitude in a decisive state upon reading the end-marker.

A trigger chain is a construction of junk states and transition matrices that causes a reduction in amplitude of a particular state only if the amplitude of another state is changed, presumably by some previous transition. Trigger chains correspond directly to partial piecewise testable sets. Consider the matrix

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

This matrix is a special case of one of the transition matrices introduced by Ambainis and Freivalds[AF98]. This matrix operates on three states and is a triggering mechanism of the chain. Consider the vectors

$$|\psi\rangle = (\alpha, 0, \beta)^T$$

and

$$U|\psi\rangle = \left(\frac{\alpha}{2} + \frac{\beta}{2}, \frac{\alpha}{\sqrt{2}} - \frac{\beta}{\sqrt{2}}, \frac{\alpha}{2} + \frac{\beta}{2}\right)^T$$

The two are equal if and only if $\alpha = \beta$. If $\alpha \neq \beta$ then the result is that the amplitudes of the first and third state are averaged with the remainder of the amplitude in the second state. We define a generalized version of U by embedding it into a larger identity block matrix. Define U_i to be

$$U_i = \begin{bmatrix} I_i & & \\ & U & \\ & & I_{n-i-3} \end{bmatrix}$$

where I_m is an $m \times m$ identity matrix, U is defined as above, and n is the number of states or, equivalently, the size of U_i. Usually the second state of the state triple is a halting state; this implies that at the start of every transition it is going to have 0 amplitude. The matrix U_i operates on a triple of states q_i through to q_{i+2}. We assume that state q_{i+1}, the second state, is a junk state unless otherwise noted. Each junk state consists of two states, an accepting and a rejecting one. However, for the purposes of clarity we do not describe the accepting

and rejecting states that implement a junk state. Theorem 4.12 proves, by construction, that partial piecewise testable sets can be accepted by MM-QFAs with bounded error.

Theorem 4.12 *Let L_x be a partial piecewise testable language. There exists an end-decisive MM-QFA that accepts L_x with bounded error.*

Proof: We construct an MM-QFA that accepts L_x. Let $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ where $x = x_0 x_1 ... x_n$, $m = 4n + 8$ and $Q = \{q_0, ..., q_m\}$. Define

$$\begin{aligned} Q_{junk} &= \{q_i \in Q \mid 0 < i < 2n \ \wedge \ i \equiv 1 \pmod 2\} \cup \{q_{2n+3}, q_{2n+4}\} \\ Q_{acc} &= \{q_{2n+1}\} \cup \{q_i \in Q \mid 2n + 4 < i \leq 3n + 6\} \\ Q_{rej} &= \{q_i \in Q \mid 3n + 6 < i \leq 4n + 8\} \end{aligned}$$

where the last $2n + 4$ states are used to implement the junk states.

We now construct the transition matrices for M. Let δ be defined by the transition matrices $\{U_\sigma\}_{\sigma \in \Sigma}$. Each transition matrix U_σ will consist of a product of n matrices. Let

$$U_\sigma = J U_n^\sigma U_{n-1}^\sigma ... U_0^\sigma$$

where

$$U_i^\sigma = \begin{cases} S & i = 0 \ \wedge \ x_0 = \sigma \\ U_{2i-2} & 1 \leq i \leq n \ \wedge \ x_i = \sigma \\ I_m & \text{otherwise} \end{cases}$$

and

$$S = \begin{bmatrix} 0 & 1 & & \\ 1 & 0 & & \\ & & & I_{m-2} \end{bmatrix}$$

The matrix S shifts the amplitude of q_0 to the junk state q_1. This is the first trigger that is activated when x_0 is read. The initial configuration of the machine is $|\psi_{init}\rangle = (\alpha_0, \alpha_1, ..., \alpha_{2n+4})^T$ where

$$\alpha_i = \begin{cases} \frac{1}{\sqrt{n+2}} & 0 \leq i \leq 2n + 2 \ \wedge \ i \equiv 0 \pmod 2 \\ 0 & \text{otherwise} \end{cases}$$

This means that the amplitude is evenly distributed among all non-halting states. The matrix J implements the junk states; it transfers the amplitude from the junk states to the accepting and rejecting states, i.e. flushes the junk states. Finally we define the transition matrix for the end-marker $\$$ as

$$U^{\$} = JFU_{2n}$$

where

$$F = \begin{bmatrix} R & & & & & & \\ & \ddots & & & & & \\ & & R & & & & \\ & & & 0 & 0 & 0 & 0 & 1 \\ & & & 0 & 1 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 1 & 0 \\ & & & 0 & 0 & 1 & 0 & 0 \\ & & & 1 & 0 & 0 & 0 & 0 \\ & & & & & & & I_{2n+4} \end{bmatrix}$$

and

$$R = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

sends all amplitude into the junk states because junk states have 0 amplitude at the start of every transition. The matrix U_{2n} sends some minimum amount of amplitude, into an accept state if the amplitudes of states q_{2n} and q_{2n+2} are unequal.

We now describe the operation of the machine. The only decisive accepting state in the machine is q_{2n+1}, and amplitude only flows into it when the end-marker is read. In order for it to get a non-zero amplitude, the amplitudes for states q_{2n} and q_{2n+2} must differ. Since all non-halting states start with the same amplitude, and since the amplitude of state q_{2n+2} will not change during the execution of the machine, until the end-marker is read, it follows that in order for the amplitude of state q_{2n} to change, the amplitude of state q_{2n-2} must change. Following

the same argument, state q_{2i} will not change in amplitude, until state q_{2i-2} changes in amplitude. Furthermore the change in amplitude of state q_{2i} is governed by the matrix components U_{2i-2} and U_{2i}. Hence, the initial change of amplitude of state q_{2i} depends exclusively on a change in amplitude of state q_{2i-2} and is governed by component U_{2i-2} that is located in the transition matrix U_{x_i}. It also follows that if any other transition matrix is applied, then the amplitude of state q_{2i} will not change since the corresponding component is not part of the transition matrix. Hence M can read $(\Sigma - \{x_i\})^*$ without changing the amplitude of state q_{2i}, but, as soon as x_i is read, component U_{2i-2} will be applied and q_{2i} will have a decreased amplitude, provided that state q_{2i-2} already had a decrease in its amplitude. Finally, the amplitude of any state q_{2i} will never increase beyond its initial value, and once the amplitude of state q_{2i} decreases, it will never increase beyond $\frac{1}{\sqrt{n+2}} - (\frac{1}{2})^{n+2}$. The main observation is that the amplitude of state q_{2i} will not decrease until the amplitude of state q_{2i-2} decreases, and until symbol x_i is read. Reading any other symbol instead will not change the amplitude of state q_{2i} and if state q_{2i-2} retains its original amplitude, then reading symbol x_i will not change the amplitude of state q_{2i}. For the case of symbol x_0, the amplitude of state q_0 is changed by matrix S to 0 and is the starting trigger. When the end-marker is read, some amount of amplitude, a minimum of $\frac{1}{\sqrt{n+2}}(\frac{1}{2})^{n+3}$, placed into the accepting state only if the amplitudes of states q_{2n} and q_{2n+2} differ. The rest of the amplitude, from all $n+2$ non-halting states is channeled into junk states. If the amplitudes of q_{2n} and q_{2n+2} do not differ then all amplitude is channeled into junk states.

The probability of M accepting a string not in the language is $\frac{1}{2}$, while the probability of M accepting a string in the language is at least $\frac{1}{2} + \frac{1}{n+2}(\frac{1}{2})^{2n+6}$. Therefore M accepts with bounded error. □

The MM-QFA construction in Theorem 4.12 is end-decisive. This means that we can take the intersection of partial piecewise testable languages and their complement. We prove these two results in the next two lemmas.

Lemma 4.13 *Let M be an end-decisive MM-QFA that accepts with bounded error. The*

complement of M is also an end-decisive MM-QFA that accepts with bounded error.

Proof: Leave the junk states as they are, and flip all the deciding accepting and rejecting states of M. The new automaton is also end-decisive and has the same margin as M and therefore accepts with bounded error. □

Lemma 4.14 *Let M and M' be end-decisive MM-QFAs that accept with bounded error. There exists an end-decisive MM-QFA M'' that accepts $L = L(M) \cap L(M')$ with bounded error.*

Proof: To facilitate the proof we use a different formalism from the one defined in Section 2.8 for MM-QFAs. Let $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej}, Q_{junk})$ and $M' = (Q', \Sigma, \delta', q'_0, Q'_{acc}, Q'_{rej}, Q'_{junk})$ be end-decisive MM-QFAs that accept $L(M)$ and $L(M')$. We construct a MM-QFA $M'' = (Q'', \Sigma, \delta'', q''_0, Q''_{acc}, Q''_{rej}, Q''_{junk})$ that accepts $L = L(M) \cap L(M')$. In this case Q_{acc} and Q_{rej} are sets that contain deciding accept and reject states, while Q_{junk} contains junk halting states, each of which can be implemented by an accepting and rejecting state. The two formalisms are equivalent.

Let $Q'' = Q \times Q'$ and $q''_0 = (q_0, q'_0)$. The three sets of halting states are defined as

$$\begin{aligned} Q''_{junk} &= \{(q_i, q'_j) \in Q'' \mid q_i \in Q_{junk} \ \vee \ q'_j \in Q'_{junk}\} \\ Q''_{acc} &= \{(q_i, q'_j) \in Q'' \mid q_i \in Q_{acc} \ \wedge \ q'_j \in Q'_{acc}\} \\ Q''_{rej} &= \{(q_i, q'_j) \in Q'' \mid (q_i, q'_j) \notin Q''_{junk} \ \wedge \ (q_i \in Q_{rej} \ \vee \ q'_j \in Q'_{rej})\} \end{aligned}$$

and the transition function δ'' is a tensor product of the transition functions δ and δ'.

Since M and M' are end-decisive, the decisive states will only have non-zero amplitude when the end-marker is read. Since the junk states do not contribute to the probability of acceptance or rejection of a string, only the decisive states play a role and these states will only have a non-zero amplitude at the end of the computation. Thus, the only measurement that will measure any amplitude in the decisive states is the one after the end-marker is read. Hence, we return to the situation where we have two types of states, accepting and rejecting. We handle this in the

same way as MO-QFAs. The fact that M'' accepts with bounded error follows from the fact that taking the tensor product and defining accepting and rejecting states as above implies that the probability of M'' accepting a string x is

$$P[M''(x) = accept] = P[M(x) = accept]P[M'(x) = accept]$$

Let M and M' accept with respective cut-points λ and λ' and margins ϵ and ϵ'. If $x \in L$ then

$$\begin{aligned} P[M''(x) = accept] &\geq \lambda\lambda' + \lambda\epsilon' + \lambda'\epsilon + \epsilon\epsilon' \\ &= (\lambda\lambda' + \epsilon\epsilon') + (\lambda\epsilon' + \lambda'\epsilon) \\ &\geq \lambda'' + \epsilon'' \end{aligned}$$

where $\lambda'' = \lambda\lambda' + \epsilon\epsilon'$ and $\epsilon'' = \min\{\lambda\epsilon', \lambda'\epsilon\}$. If $x \notin L$ then without loss of generality assume that $x \notin L(M)$. It follows that

$$\begin{aligned} P[M''(x) = accept] &\leq \lambda\lambda' - \lambda\epsilon' + \lambda'\epsilon - \epsilon\epsilon' \\ &\leq \lambda\lambda' - \lambda\epsilon' + \lambda'\epsilon - \epsilon\epsilon' + 2\epsilon\epsilon' \\ &= (\lambda\lambda' + \epsilon\epsilon') + (\lambda\epsilon' - \lambda'\epsilon) \\ &\leq \lambda'' - \epsilon'' \end{aligned}$$

If both languages $L(M)$ and $L(M')$ are in $\mathbf{RMM}_{\epsilon}$ then the language L is in $\mathbf{RMM}_{\epsilon''} \subseteq \mathbf{RMM}$. The MM-QFA M'' accepts L with bounded error and the decisive states are only used when the end-marker is read, even though junk states may occur throughout the computation. Thus, M'' is an end-decisive MM-QFA that accepts with bounded error. □

Corollary 4.15 *Let M and M' be end-decisive MM-QFAs that accept with bounded error. There exists an end-decisive MM-QFA M'' that accepts $L = L(M) \cup L(M')$ with bounded error.*

Since the class of languages that are accepted by end-decisive MM-QFAs with bounded error is closed under boolean operations, and since MM-QFAs that accept partial piecewise

testable sets with bounded error can be constructed such that they are end-decisive, it follows that we can construct MM-QFAs that accept piecewise testable sets with bounded error.

Theorem 4.16 *MM-QFAs can accept piecewise testable sets with bounded error.*

Chapter 5

Conclusions and Future Work

We have defined two different models of 1-way quantum finite automata. These computation models, termed measure-once and measure-many, are probabilistic and are similar to probabilistic finite automata. The measure-once automaton operates by performing unitary operations, on its configuration, corresponding to the symbols that it reads. At the end of its computation it measures its state and accepts if it is in an accepting state. The measure-many model also performs unitary operations on its configuration, which correspond to the symbols that it reads, but also performs a partial measurement after every transition collapsing its superposition to a subspace spanned by the non-halting, accepting or rejecting states. This allows measure-many automata to halt part way through the computation.

We showed that the class of languages accepted by measure-once automata with bounded error (**RMO**) is exactly the class of languages accepted by group finite automata. This implies that the class **RMO** is closed under boolean operations, inverse homomorphisms, and word quotients. This complements the results of Moore and Crutchfield[MC98] that proved that the class of languages accepted by a measure-once automaton with unbounded error (**UMO**) is also closed under these operations. Measure-once automata that accept with unbounded error can accept non-regular languages and, in particular, can solve the word problem over the free group. We showed that any language accepted by a measure-once automaton can also be accepted by a probabilistic finite automaton, which means that probabilistic finite automata can also solve the word problem over the free group. Finally, we presented an algorithm for determining if two measure-once automata are equivalent, i.e. have the same distribution.

We defined the class of languages accepted by measure-many automata with bounded error (**RMM**) and the class of languages accepted by measure-many automata with unbounded error (**UMM**). We showed that both classes are closed under complement, inverse homomorphism, and word quotient. We showed by construction that the class **RMO** is properly contained within **RMM** and that the class **UMO** is contained within **UMM**. We proved a necessary condition for a language to be in the class **RMM**, which we call the partial ordering condition. We have shown by construction that piecewise testable sets are contained within the class **RMM** and, in the process, introduced the concepts of junk states, trigger chains, and end-decisiveness. We have also shown that the class of languages accepted by end-decisive measure-many automata are closed under boolean operations.

5.1 Open Problems

The partial ordering condition is necessary for a language to be a member of the class **RMM**. However it is still unknown whether the condition is sufficient. We conjecture that the partial ordering condition is also sufficient for membership in the class **RMM**.

Conjecture 5.1 *The partial ordering condition is necessary and sufficient for a language L to be a member of the class* **RMM**.

We showed that if the partial ordering condition was sufficient for membership within the class **RMM** then the class **RMM** is closed under boolean operations. An easier open problem is to determine if the class **RMM** is closed under boolean operations. A negative result would prove the above conjecture false.

Finally, Ambainis and Freivalds[AF98] state any language that has irreversible transitions into non-spin states cannot be accepted by a measure-many automaton with a probability greater than $\frac{7}{9}$. We have not been able to construct an automaton that can accept such a language with a probability greater than ≈ 0.68. Narrowing this gap is also an open problem.

Bibliography

[AF98] Andris Ambainis and Rūsiņš Freivalds. 1-way quantum finite automata: Strengths, weaknesses and generalizations. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 332–342, November 1998.

[AKN98] Dorit Aharonov, Alexei Kitaev, and Noam Nisan. Quantum circuits with mixed states. In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*, pages 20–30, 1998.

[Ang82] Dana Angluin. Inference of reversible languages. *Journal of the Association for Computing Machinery*, 29(3):741–765, July 1982.

[ANV98] Andris Ambainis, Ashwin Nayak, and Umesh Vazirani. On the space-efficiency of 1-way quantum finite automata. *Preprint, To be submitted to Information Processing Letters*, (quant-ph/9804043), 1998.

[Ben73] Charles Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:198–200, November 1973.

[Ber76] S. Berberian. *Introduction to Hilbert Spaces*. Chelsea Publishing Company, New York, 1976.

[BH97] Gilles Brassard and Peter Høyer. An exact quantum polynomial-time algorithm for Simon's problem. In *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*, 1997.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal of Computing*, pages 1411–1473, October 1997.

[Deu85] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London*, A400:97–117, 1985.

[Deu89] D. Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London*, A425:73–90, 1989.

[DS89] Cynthia Dwork and Larry Stockmeyer. On the power of 2-way probabilisitic finite state automata. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 480–485, 1989.

[DS90] Cynthia Dwork and Larry Stockmeyer. A time complexity gap for two-way probabilisitic finite-state automata. *SIAM Journal of Computing*, pages 1474–1483, December 1990.

[Eil76] S. Eilenberg. *Automata, Languages and Machines*, volume B. Academic Press, New York, 1976.

[Fre81] Rūsiņš Freivalds. Probabilistic two-way machines. In *Proceedings of 10th International Symposium on Mathematical Foundations of Computer Science*, volume 118 of *Lecture Notes on Computer Science*, pages 33–45, September 1981.

[Gas96] Stephen Gasiorowiz. *Quantum Physics.* John Wiley and Sons, Inc., New York, 1996.

[GKP94] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics.* Addison-Wesley Publishing Company, New York, New York, 1994.

[Gos92] Amit Goswami. *Quantum Mechanics.* Wm. C. Brown Publishers, New York, 1992.

[Gro96] Lov Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pages 212–219, 1996.

[HU79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley Publishers, Reading, Massachusetts, 1979.

[KW97] Attila Kondacs and John Watrous. On the power of quantum finite state automata. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 66–75, 1997.

[LMT97] Klaus-Jörn Lange, Pierre McKenzie, and Alain Tapp. Reversible space equals deterministic space. In *Proceedings of the 12th IEEE Conference on Computational Complexity*, pages 45–50, 1997.

[MC98] Cristopher Moore and James P. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 1998.

[Mor90] Michael A. Morrison. *Understanding Quantum Physics.* Prentice Hall, Englewood Cliffs, NJ, 1990.

[MT69] A. R. Meyer and C. Thompson. Remarks on algebraic decomposition of automata. *Mathematical Systems Theory*, 3(2):110–118, 1969.

[Paz71] Azaria Paz. *Introduction to Probabilistic Automata.* Academic Press, New York, New York, 1971.

[Per94] Dominique Perrin. Finite automata. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 1. Elsvier Science Publisher, 1994.

[Pin87] J. Pin. On languages accepted by finite reversible automata. In *Proceedings of the 14th International Colloquium on Automata, Languages and Programming*, volume 267 of *Lecture Notes on Computer Science*, pages 237–249, 1987.

[Rab63] M. Rabin. Probabilistic automata. *Information and Control*, 6:230–245, 1963.

[RS59] M. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3:114–125, April 1959.

[She59] J. Shepherdson. The reduction of two-way automata to one-way automata. *IBM Journal of Research and Development*, 3:198–200, April 1959.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[Sim97] Dan Simon. On the power of quantum computation. *SIAM Journal of Computing*, pages 1474–1483, October 1997.

[Tze89] Wen-Guey Tzeng. The equivalence and learning of probabilistic automata. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 268–273, 1989.

[Tze92] Wen-Guey Tzeng. A polynomial-time algorithm for the equivalence of probabilistic automata. *SIAM Journal on Computing*, 21(2):1484–1509, August 1992.

[Wag85] Stan Wagon. *The Banach-Tarski Paradox*. Cambridge University Press, New York, New York, 1985.

[Yao93] Andrew Chi-Chi Yao. Quantum circuit complexity. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pages 352–361, 1993.