Applications of Algebraic Automata Theory to Quantum Finite Automata

Mark Mercer

Doctor of Philosophy

School of Computer Science

McGill University

Montreal, Quebec

2007-08-30

A thesis submitted to the Faculty of Graduate Studies
in partial fulfillment of the requirements
of the degree of Ph.D. Science

ACKNOWLEDGEMENTS

I would first like to thank my supervisor Denis Thérien for his patience, support, and guidance over the years. His intuition and depth of knowledge have touched every part of this thesis. Denis has provided me the funding and the opportunity to work and study at McGill. I have thoroughly enjoyed my experience, and I'm sure the lessons I've learned here will stay with me throughout my career.

I would also like to thank my co-supervisor Pascal Tesson, for reorienting me at the times when I felt the most lost, for regular invitations to Québec city, and for many rounds of thesis corrections.

I would like to thank my program committee members Claude Crépeau and Doina Precup for their advice and support. I also thank my external examiner, Carlo Mereghetti, for many helpful corrections and comments. I wish to thank Patrick Hayden, Alexei Miasnokov, and John Liede for participating on my defence committee.

I would like to thank my regular collaborator Martin Beaudry, for meetings in Sherbrooke and for his willingness to address the most technical of questions. I would like to thank Klaus-Jörn Lange for inviting me to visit his research group in Tübingen. It was an experience that I will never forget.

I would like to thank my parents for giving me such a great start in life. They instilled in me the value of education, and have supported me in countless ways. I would also like to thank my sisters, Ann-Marie and Christina, for their love and support.

Finally, I would like to thank my wife Masoumeh for her love and encouragement. Throughout this year, she has stood beside me to pick me up whenever I was stuck and to celebrate every success. She has made this the happiest year of my life.

ABSTRACT

The computational model of Quantum Finite Automata has been introduced by multiple authors (e.g. [38, 44]) with some variations in definition. The objective of this thesis is to understand what class of languages can be recognized by these different variations, and how many states are required.

We begin by showing that we can use algebraic automata theory to characterize the language recognition power of QFAs. Algebraic automata theory associates to each language a canonical *syntactic monoid*, and the algebraic structure of this monoid becomes a meaningful parameter in describing language classes. We show that the class of languages recognized by Latvian QFAs [3] corresponds exactly to boolean combinations of languages recognized by Brodsky and Pippenger's QFA model [20], which correspond exactly to those languages whose syntactic monoid is in the class **BG**. Known results give us a decision procedure for testing membership in this language class. We also use algebraic automata theory to give nearly tight upper and lower bounds on the class of languages recognized by Brodsky and Pippenger's QFAs.

We then extend a number of lower bound techniques known for Kondacs and Watrous' 1-way QFA model to Nayak's Generalized QFA. Both of these models are related in that they are permitted to halt before reading the entire input, allowing them to recognize certain languages whose syntactic monoid lies outside of **BG**.

Finally, we investigate the question of QFA succinctness. It is known that QFAs can recognize some languages using exponentially fewer states compared to deterministic finite automata. We extend results from [16] to show that the word problem over abelian groups has this property. We also give example of interesting noncommutative languages with this property.

ABRÉGÉ

Nous étudions dans cette thèse les automates quantiques finis (QFA), un modèle de calcul dont plusieurs définitions coexistent (e.g. [38, 44]). L'objectif central de leur étude est de comprendre quels sont les langages qui peuvent être reconnus par chacune des variantes et de déterminer le nombre d'états nécessaires pour ces calculs.

Nous montrons d'abord que la théorie algébrique des automates peut servir à caractériser la puissance de calcul des QFAs. La théorie algébrique des automates associe à chaque langage régulier un *monoïde syntactique*: plusieurs classes importantes de langages peuvent alors être mises en relation avec les propriétés algébriques de ces objets. En exploitant cet angle d'attaque, nous montrons que la classe de langages reconnus par les QFA dits "Lettons" [3] coïncide d'une part avec la classe des combinaisons booléennes de langages reconnus par les QFAs de Brodsky et Pippenger [20] et, d'autre part, à la classe de langages dont le monoïde syntactique appartient à la classe **BG**. Cette caractérisation démontre également l'existence d'un algorithme permettant de déterminer si un langage donné appartient à cette classe. L'approche algébrique nous permet aussi d'établir des bornes inférieures et supérieures très proches l'une de l'autre pour la classe de langages qui peuvent être reconnus par les QFAs de Brodsky et Pippenger.

Nous montrons ensuite que plusieurs des méthodes permettant de borner la puissance des QFAs "1-way" de Kondacs et Watrous peuvent être étendues aux QFAs généralisés de Nayak. Ces deux modèles peuvent tous les deux arrêter leur calcul avant d'avoir terminé la lecture de leur entrée et peuvent ainsi reconnaître des langages dont le monoïde syntactique n'est pas dans **BG**.

Finalement, nous étudions les QFAs succints. On sait que certains langages reconnus par des QFAs peuvent l'être par des automates quantiques qui utilisent un nombre d'états exponentiellement plus petit que leurs équivalents déterministes. Nous étendons les résultats de [16] et montrons que cela est le cas pour le problème du mot de tout groupe abélien. Nous donnons également la première construction de ce type pour une famille de groupes non commutatifs.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
Introduction

The central objective in theoretical computer science is to obtain a rigorous and formal understanding of computation. In particular, we would like to determine which problems can or cannot be solved with the use of a given set of computational resources. The fundamental resources of interest are time and space, but investigations outside of this framework have also produced considerable insight. Notable examples include the study of randomized computation [32], parallel computation [23], and nondeterminism [22].

In recent years there has been a push to better understand the power of computational devices which make nontrivial use of the principles of quantum mechanics. The excitement is driven by many interesting results, among them the discovery of a polynomial time quantum algorithm for factorization [59], a robust formal definition of a quantum computer [25, 13, 69] and the necessary error correction schemes [60], the discovery of quantum teleportation [11] and strong quantum cryptograpic schemes [12].

Much of the existing theoretical research into quantum computation has been in the pursuit of the 'quantum analogue' of established concepts in theoretical computer science and in related branches of mathematics. These include the quantum analogues of complexity classes [37, 68], formal grammars [44], random walks [1], and information [46].

A rich theory of finite automata has been developed to understand the power of computational devices which use finite memory. In this thesis, we study *Quantum Finite Automata* (QFAs), which are the analogue of finite automata in the sense that they model what computations can be performed by an online quantum machine with memory whose size does not change with the size of the input. Automata theory plays a foundational role in computer science, and it is hoped that some of this success can be transferred to the quantum case.

Quantum finite automata can be used to model the dynamics of finite quantum systems in the same way that deterministic finite automata model the dynamics of discrete finite systems. They are a simple model of quantum computation, and a good understanding of QFAs can produce results in related areas of quantum information science, for example as they did in the case of dense quantum coding [7]. Furthermore, it is important to understand the power of quantum computation in space restricted settings, as the best current implementations of quantum computers have only small constant-sized memory [67].

1.1 The Algebraic Approach

The class of finite automata and the class of *regular languages* which they recognize is an island of solid ground from which to launch theoretical investigations. For many years, researchers have sought to further strengthen our understanding of the regular languages by characterizing subclasses of the regular languages.

For a given automaton M with input alphabet Σ, each word in Σ^* will induce an operator on the set of states of M. These operators form forms a finite monoid under composition. It is well established that by taking this algebraic perspective on finite state machines, we obtain powerful insight into the structure of subclasses of the regular languages. This approach is known as *Algebraic Automata Theory*.

The ideal framework for the theory was developed by Eilenberg [27], who established a bijection between regular language classes which satisfy certain closure properties, called *varieties of languages*, with *varieties of monoids*. An extensive research program has uncovered a rich taxonomy of regular language classes with matching algebraic characterizations. Famous results include the algebraic characterization of *star-free* languages by Schützenberger [57] and of the piecewise testable languages by Simon [61]. These results has been applied to several areas in theoretical computer science, such as logic [42, 63] and circuit complexity [39, 8].

In this thesis, we apply algebraic automata theory to the study of quantum finite automata. We are able to prove a number of characterizations of the languages recognizable by these different QFA variations, and identify a number of surprising interrelations between them. These results are obtained using established knowledge of algebraic automata theory and results which are developed in the thesis.

1.2 Our Contributions

Several models of quantum finite automata have been proposed, and there is no clear consensus on which of these is the most appropriate. Each model allows for a different set of possible actions on reading an input letter. These differences correspond to different underlying physical assumptions. In this thesis we prove several properties of QFAs which, taken together, give us a much clearer picture of the interrelation between these different models.

We focus our attention on five QFA models from the literature. The simplest of these is the *Measure-Once QFA* (MOQFA), which are restricted to unitary transformations. MOQFA can only recognize languages which can in turn be recognized by permutation automata. However they can do so using much fewer states [5]. Several generalizations of this definition have been considered. In Kondacs and Watrous' definition (KWQFA), the machine is permitted to halt before reading the entire output. This is the most studied of the five variations. Another way of generalizing MOQFAs is to introduce some randomness through intermediate measurements. This corresponds to the definition of *Latvian QFA* (LQFA). The *Generalized QFA* (GQFA) definition simultaneously generalizes KWQFA and LQFA. The final definition, from Brodsky and Pippenger (BPQFA), corresponds to an important subclass of KWQFA.

We begin in Chapter 2 with an overview of the basic principles and formalism for quantum mechanics. We survey a number of computational models based on quantum mechanics, and we introduce the five variations of QFA which we consider. In the second half of the chapter, we give an introduction to algebraic automata theory.

In Chapter 3 we consider applications of Eilenberg's variety theory to characterize the class of languages recognized by different QFAs, including new results for LQFA and BPQFA. In Chapter 4 we consider a generalization of Eilenberg varieties to positive varieties in order to get a more refined characterization of the languages recognized by BPQFA. In Chapter 5 we extend some known techniques of Ambainis and Freivalds [4] as well as the LQFA characterization of Chapter 3 in order to prove impossibility results for GQFA. Finally in Chapter 6 we consider the question of constructing succinct QFAs. We provide new constructions for succinct QFAs, and we present some preliminary results which could be used to prove lower bounds on QFA size.

1.2.1 Characterizations of QFA

The objective of Chapter 3 is to apply Eilenberg's variety theorem to obtain algebraic characterizations of the computational power of QFA. We begin by considering which of the QFA variations have the necessary closure properties to form varieties of languages. A language variety is a class of languages which is closed under boolean operations, inverse morphisms, and word quotient. If for a particular QFA variation we can give constructions for all of these closure properties, this would immediately imply the existence of some exact algebraic characterization of this type of QFA. We see that this is true for MOQFA and LQFA.

The class of languages recognized by MOQFA is known [44] to correspond exactly to the class of languages recognized by permutation automata. We show that this result has a nice interpretation in terms of algebraic automata theory.

Next we consider the case of LQFA. We obtain an exact algebraic characterization: LQFA can recognize exactly those languages whose syntactic monoid is in the variety **BG** of block groups. The proof involves known properties of the class **BG** as well as several technical results regarding LQFAs. To obtain the characterization, we give a construction for LQFA to recognize the language $\Sigma^* a_1 \Sigma^* \dots a_k \Sigma^*$, which implies that LQFA can recognize all languages which are recognized by monoids in the class **J** of $\mathcal{J}$*-trivial* monoids. Then, we use algebraic tools to extend this to all languages which are recognized by monoids in **BG**. Finally, we show that LQFA cannot recognize the languages $a\Sigma^*$ or Σ^*a, and it turns out that this suffices to complete the argument.

There are a number of nice consequences of the LQFA characterization. Since membership in **BG** is decidable, this implies that recognizability by LQFA is also decidable. Furthermore, it implies the following characterization of languages recognized by LQFA: L is recognized by LQFA iff it is a boolean combination of languages of the form $L_0a_1L_1 \ldots a_kL_k$, where each L_i is a language recognized by a group.

A similar line of argument is then used to show that a language L is a boolean combination of languages recognized by BPQFA if and only if its syntactic monoid is in **BG**. This is a surprising connection between BPQFA and LQFA, since the types of permitted transformations for these two variations are quite different on the surface. The proof that boolean combinations of BPQFA is contained in **BG** relies on some existing lower bound techniques for KWQFAs.

1.2.2 BPQFA

In Chapter 4 we begin a finer investigation on the class of languages recognized by BPQFA. We begin with a proof that BPQFA cannot recognize the complement of the language $\Sigma^*a\Sigma^*b\Sigma^*$. The language $\Sigma^*a\Sigma^*b\Sigma^*$ can be recognized, however, so the class of languages recognized by BPQFA is not closed under complement. This language class does however form what is called a *positive variety*, which implies that there is some exact characterization for this class in terms of a structure called *ordered monoids*. We introduce the theory of ordered monoids and give some important examples of positive varieties of monoids.

We make a number of steps towards obtaining this exact characterization, obtaining nearly matching upper and lower bounds. On one hand, we present several constructions that provably extend the class of languages known to be recognizable by BPQFA. On the other hand, we develop an algebraic property that implies nonrecognizability by BPQFA. The results seem to point to the conjecture that L is recognized by BPQFA if and only if its ordered syntactic monoid is in $(\mathbf{Nil}^+ \textcircled{\scriptsize M} \mathbf{J_1}) * \mathbf{G}$.

1.2.3 GQFA

There are a number of impossibility results which exist for KWQFA [38, 4, 5, 6]. These results relate the recognizability of a language L by KWQFA to properties of the minimal automaton for L. In this chapter we investigate similar impossibility results for the case of GQFA.

Nearly all of the impossibility results for KWQFAs rely in part on a key lemma that separates the state space into two parts according to their behavior: the *ergodic* and the *transient* part. We show that this powerful lemma can be extended to the case of GQFA. This gives us a deeper understanding of the structure of GQFAs. Using this characterization, we can extend several KWQFA impossibility results to the GQFA case. These results highlight several key properties of GQFA, including the fact that the class of languages recognized by GQFA is not closed under complement, and that there are languages which can be recognized by GQFA with probability $p = 2/3$ but not with probability $p > 2/3$.

1.2.4 MOQFA Succinctness

An interesting property of quantum finite automata is that they can recognize certain languages using much fewer states than the smallest deterministic automaton, or even the smallest randomized automaton [4]. In an early result, it was shown that languages of the form $L_p = \{w : |w| \mod p = 0\}$ for prime p have MOQFA size $O(\log p)$. However, little is known regarding which languages can be recognized succinctly and which ones cannot.

Recently in [16] it was shown that the word problem over groups of the form $\mathbb{Z}_n^h$ can be recognized by MOQFA using $O(\log n)$ states. We show that this is true for all abelian groups. We show some new languages which can be recognized succinctly, including some interesting examples of succinctly recognizable noncommutative languages. We also present some ideas for normalizing QFA transitions in the hopes of proving lower bounds on QFA size.

CHAPTER 2
Background

In this chapter, we outline the necessary background for the discussion in the later chapters. In Section 2.1, we introduce quantum mechanics and quantum computation, and we give formal definitions for quantum finite automata. In Section 2.2 we introduce the fundamental concepts of algebraic automata theory, the main tool of our investigation.

2.1 Quantum Computation

The objective of this section is to introduce the mathematical foundations of quantum mechanics, and to give formal definitions to quantum finite automata. Before we begin, we give an informal introduction to fundamental concepts in quantum mechanics.

Quantum Mechanics is a mathematical framework for describing certain physical properties occurring at the atomic and sub-atomic level. This framework gives a good description of certain physical effects which are not adequately explained by classical mechanics.

Quantum mechanics resolves two seemingly contradictory observations about the nature of energy. The first of these observations is the apparent wave-like behaviour of energy, as seen for example in the diffraction and interference of light, and the interpretation of light as electromechanical waves. A concrete example of this can be seen in Young's double-slit experiment. In this experiment, a light is shone towards a filter consisting of two small slits, and then projected onto a screen. The resulting image exhibits an interference pattern that one would expect if light was composed of waves.

The second observation is that, for a fixed source and receiver of energy, the amount of energy received will often occur at discrete multiples of some fixed constant. This would suggest that, like atomic particles, energy should be distributed in discrete packets that take up a well-defined position in space. These energy packets are called *quanta*, and in the case of light they are called *photons.* This observation of discretized energy matches well with Bohr-Rutherford model of an atom, where electrons take discrete valence levels within an atom, and the energy level of an atom depends on the position of electrons within the different orbital levels. When an atom stores or releases energy by moving an electron between two orbitals, this will cause a photon of discrete magnitude to be absorbed or dissipated.

The two observations run into a conflict when we find that even single quanta can produce wave-like effects such as interference. This aspect of subatomic behavior is referred to as *wave-particle duality.* Quantum Mechanics reconciles the particle and wave nature of energy into a single theory. The quantum mechanical explanation of the wave-particle phenomena is that quanta behave as waves while they are propagating from sender to receiver, but they take on fixed positions as soon as their position is observed.

More formally, suppose we consider some measurable property of a particle, such as the position of a particle in space. Let X be a set of possible positions. While the particle is not being observed, the quantum state of the system is a vector consisting of complex values c_i called *amplitudes* associated to every $x_i \in X$, with these c_i's satisfying $\sum_i |c_i|^2 = 1$. The probability of observing the outcome x_i in this state is then $|c_i|^2$. As time passes, the state may change, and the c_i's are updated according to a linear transformation which preserves the condition $\sum_i |c_i|^2 = 1$. It is important to note that, as an independent observer of a quantum state, we do not have full knowledge of the amplitudes composing a quantum state. Rather, we obtain partial information about these amplitudes by performing measurements.

This is similar to the framework that is seen in a *random state*. For a random state, each possible position x_i has associated with it a positive real p_i satisfying $\sum_i p_i = 1$, and the evolution of the state over time is expressed as a linear transformation which preserves the condition $\sum_i p_i = 1$. In the quantum case, the negative and complex amplitudes introduce the possibility that two or more quantum potentialities may cancel each other. This suggests that a particle does not take a concrete position until the time that it is measured.

Quantum mechanics has a reputation for being very complicated due to the counter-intuitive consequences of the theory. The mathematics of quantum mechanics, however, is accessible to anyone with a background in linear algebra. We begin this section with a review of linear algebra, with particular attention to the concepts which arise when using quantum mechanics. In Section 2.1.2 we give the formal laws of quantum mechanics. Section 2.1.3 gives an overview of the history of quantum computation, outlining the important computational objects of study. In the last section we give the formal definition of the QFA models we consider and some discussion.

2.1.1 Preliminaries

We first review some linear algebraic concepts that arise frequently in quantum mechanics, and we introduce Dirac's vector notation. An expanded introduction can be found in [46].

Linear Algebra Primer

Recall that for any vector space V over field $\mathbb{F}$, or $\mathbb{F}$-vector space, there exists a set $B \subseteq V$ such that any vector $v \in V$ can be expressed uniquely as a linear combination $\sum_{b \in B} v_b b$, where $v_b \in \mathbb{F}$. Such a set is called a *basis* for V. It can be shown that any two bases for V must have the same cardinality, and this value is called the *dimension* of V. We will be chiefly concerned with finite dimensional V, in which case we can express a v as an n-dimensional column vector. The vectors in space V form an abelian group under addition, and we denote by $\mathbf{0}$ the identity.

For vector spaces V and W over $\mathbb{F}$, we say that $A : V \to W$ is a *linear transformation* if it satisfies $A(v_1 + v_2) = A(v_1) + A(v_2)$ and $A(\alpha v_1) = \alpha A(v_1)$ for arbitrary $v_1, v_2 \in V$, $\alpha \in \mathbb{F}$. Such a transformation can be expressed as a *matrix* of $m \times n$ coefficients, where m and n are the dimensions of W and V respectively. We often identify a linear transformation with its matrix representation.

We will denote by Av the image of v under A. As we are acting on the left, we will denote the composition of two linear transformations A and B as BA. BA is again a linear transformation. If $V = W$, we call A a *linear operator*. For each vector space there is a unique linear operator I such that $Iv = v$ for all $v \in V$. We call this the *identity operator* on V. If B is an operator such that $AB = BA = I$, then we call B an *inverse* of A. If A has an inverse, it is necessarily unique and we denote it as A^{-1}.

We say that $v \neq \mathbf{0}$ is an *eigenvector* of A if $Av = \lambda v$ for some $\lambda \in \mathbb{F}$. Then λ is the *eigenvalue* of A corresponding to v. The set of eigenvector-eigenvalue pairs of a linear operator characterize many important properties of that operator.

For a given eigenvalue λ, the set of all v such that $Av = \lambda v$ forms a subspace, which is called the *eigenspace* associated with λ. The dimension of this space is called the *multiplicity* of λ,

We define the *trace* $Tr(A)$ of a complex linear operator A to be the sum of the diagonal coefficients when A is represented as a matrix. This sum will be equal to the sum of the eigenvalues taken with their multiplicities, and thus is invariant under a change of basis. A useful property of the trace operation is that it satisfies $Tr(AB) = Tr(BA)$ for linear operators A and B.

Inner Product Spaces

We now restrict our attention to complex vector spaces. An *inner product* function on V is a function $(\cdot, \cdot) : V \times V \to \mathbb{C}$ that is linear in the second argument, is conjugate symmetric (i.e. $(v, w) = \overline{(w, v)})$, and is such that (v, v) is nonnegative real for all $v \in V$. We say that V equipped with such an inner product is called an *inner product space.*

The conditions on the inner product function allow us to introduce notions of lengths and angles to the vector space. First, we say that the *norm* $\|v\|$ of a vector v is the quantity $\sqrt{(v, v)}$. We say that vectors of unit length are *normal.* To *normalize* a vector is to scale a vector to unit length. Furthermore, we say that two nonzero vectors v and w are *orthogonal* if $(v, w) = 0$. We extend the concept of orthogonality to subspaces by saying that subspaces S and T of V are orthogonal if vectors in S and T are pairwise orthogonal.

It is often convenient to express vectors in terms of bases where the basis elements are pairwise orthogonal and normal, or *orthonormal*. In this case, the coefficient of basis element b for vector v is simply (b, v).

For any linear operator A, there is a unique linear operator $A^\dagger$ such that $(v, Aw) = (A^\dagger v, w)$ for all $v, w \in V$. This matrix is called the *adjoint* of A. In the finite dimensional case, the matrix representation of $A^\dagger$ can be obtained from A by taking the transpose of A and taking the conjugate of each element. The adjoint operator † satisfies $(AB)^\dagger = B^\dagger A^\dagger$.

Hilbert Spaces and Dirac Notation

A *Hilbert space* is a complex inner product space that is closed under taking limits. In the finite dimensional case, any complex vector space will be a Hilbert space.

Hilbert spaces have an interesting relationship with their duals. The *dual* V^* of an $\mathbb{F}$-vector space V is the vector space formed by the set of linear functions $f : V \to \mathbb{F}$. Hilbert spaces have the property that the map $\mu : V \to V^*$ defined by $\mu(v) = (v, \cdot)$ is an isomorphism. In particular, if $\{v_1, \ldots, v_n\}$ is a basis for V then $\{\mu(v_1), \ldots, \mu(v_n)\}$ is a basis for V^*.

Dirac introduced an unconventional notation for expressing vectors that highlights the relationship of Hilbert spaces with their duals, and is well suited to the Hilbert space manipulations that arise in quantum mechanics. The first convention is to denote vectors as $|\psi\rangle$ where ψ is the label of the vector. Second, the notation $\langle\psi|$ is used to denote the linear function $\langle\psi| : \mathbb{C}^n \to \mathbb{C}$ defined by $\langle\psi|(|\varphi\rangle) = (|\psi\rangle, |\varphi\rangle)$, i.e. $\langle\psi| = \mu(|\psi\rangle)$. The notation $\langle\psi|\varphi\rangle$ is used as a shorthand for $\langle\psi|(|\varphi\rangle)$, and thus $\langle\psi|\varphi\rangle$ is equal to the inner product. For the remainder of the thesis, we will use the notation $\langle\cdot|\cdot\rangle$ for inner products functions of Hilbert spaces. Observe that the matrix representation of $\langle\psi|$ is simply the conjugate transpose of $|\psi\rangle$. We define $|\psi\rangle^\dagger = \langle\psi|$ so that, for example, $(A|\psi\rangle)^\dagger = (|\psi\rangle)^\dagger A^\dagger = \langle\psi|A^\dagger$.

The following property, called the *completeness relation*, is often used to simplify expressions. Let $\{|i\rangle : 1 \leq i \leq n\}$ be an orthonormal basis for $\mathbb{C}^n$. Then for any vector $|\psi\rangle$ we have $|\psi\rangle = \sum_i \langle i|\psi\rangle|i\rangle = \sum_i |i\rangle\langle i|\psi\rangle = (\sum_i |i\rangle\langle i|)|\psi\rangle$. Thus $\sum_i |i\rangle\langle i| = I$.

Direct Sums and Tensor Products

Direct sums and *tensor products* are two composition operators on vector spaces. These operators allow us to compose two vector spaces to make a larger space, or conversely, it allows us to express structural decompositions of vector spaces.

We say that V is the direct sum of the subspaces S and T (we write $V = S \oplus T$) if every vector $|v\rangle \in V$ can be written uniquely as $|s\rangle + |t\rangle$ for some $|s\rangle \in S$ and $|t\rangle \in T$. In most cases of interest to us, S and T will be orthogonal subspaces, however this is not strictly required by the definition.

Let us write $(|s\rangle, |t\rangle) \in S \times T$ as $|s\rangle \otimes |t\rangle$, and let $\alpha \in \mathbb{C}$. The tensor product $S \otimes T$ of S and T is defined to be the quotient of the vector space $S \times T$ with respect to the following identities:

- $(|s_1\rangle + |s_2\rangle) \otimes |t\rangle = |s_1\rangle \otimes |t\rangle + |s_2\rangle \otimes |t\rangle$,

- $|s\rangle \otimes (|t_1\rangle + |t_2\rangle) = |s\rangle \otimes |t_1\rangle + |s\rangle \otimes |t_2\rangle$,
- $(\alpha|s\rangle) \otimes |t\rangle = \alpha(|s\rangle \otimes |t\rangle) = |s\rangle \otimes (\alpha|t\rangle)$.

If we identify vectors of $S \times T$ according to these rules, then the equivalence classes will form a vector space of dimension $m \cdot n$. In particular if $\{|s_1\rangle, \ldots, |s_m\rangle\}$ and $\{|t_1\rangle, \ldots, |t_n\rangle\}$ are bases for S and T respectively, then the set $\{|s_i\rangle \otimes |t_j\rangle : 1 \leq i \leq m, 1 \leq j \leq n\}$ forms a basis for $S \otimes T$. If S and T are Hilbert spaces, then there is a natural way to define a Hilbert space over $S \otimes T$ given the associated norms for S and T.

The notion of tensor product is related to bilinear forms. Let S and T be two $\mathbb{C}$-vector spaces of dimension m and n respectively. A *bilinear form* of S and T is a function $b : S \times T \rightarrow \mathbb{C}$ that is linear in S for each fixed $|t\rangle \in T$ and linear in T for each fixed $|s\rangle \in S$. The tensor product $S \otimes T$ is constructed so that the dual of $S \otimes T$ is isomorphic to the space of bilinear forms of S and T.

In many cases we will need to apply the tensor operation and direct sum operation several times, so it is important to note that both operations are associative. When working with a long sequence of tensors, it is common to discard the $\otimes$ symbol and just write $|s\rangle \otimes |t\rangle$ as $|s\rangle|t\rangle$ or even $|st\rangle$.

For operators A and B on S and T respectively, we define the operator $A \otimes B$ to be the unique linear operator on $S \otimes T$ satisfying $A \otimes B(|s\rangle \otimes |t\rangle) = A|s\rangle \otimes B|t\rangle$. The operator $A \oplus B$ is defined similarly.

Important Classes of Linear Operators

There are a number of special classes of linear transformations that arise in the study of quantum computation. We present some of the more important ones below. For this section we will assume that the operators are working on a Hilbert space.

Unitary operators: A operator U is called *unitary* if it satisfies $(|\psi\rangle, |\varphi\rangle) = (U|\psi\rangle, U|\varphi\rangle)$ for all vectors $|\psi\rangle$, $|\varphi\rangle$. In other words, $\langle\psi|U^\dagger U|\varphi\rangle = \langle\psi|\varphi\rangle$. Since this relation must hold in particular on any set of basis elements, by the completeness relation we must have $U^\dagger U = UU^\dagger = I$ and therefore $U^\dagger = U^{-1}$. Conversely, any U satisfying $U^\dagger = U^{-1}$ must be unitary.

Normal operators: An operator N is called *normal* if N commutes with $N^\dagger$. This is true, for example, in the case of unitary matrices. Normal matrices satisfy the following structure theorem:

Theorem 2.1 *(Spectral Decomposition Theorem) Let N be a normal matrix with eigenvectors $|\varphi_1\rangle, \ldots, |\varphi_k\rangle$ and corresponding eigenvalues $\lambda_1, \ldots \lambda_k$. Then N can be expressed as:*

$$N = \sum_i \lambda_i |\varphi_i\rangle\langle\varphi_i|.$$

This implies that normal matrices are diagonal with respect to any basis consisting of orthonormal eigenvectors.

It is convenient to define an extension of functions $f : \mathbb{C} \to \mathbb{C}$ to normal matrices. For normal N, we define $f(N)$ to be the matrix formed by applying the spectral decomposition and applying f termwise to the eigenvalues of N, i.e. $f(N) = \sum_i f(\lambda_i)|\varphi_i\rangle\langle\varphi_i|$. This is a well-defined operation in general. As an example, we define $\sqrt{N}$ to the operator formed by taking the square roots of the eigenvalues of N. This new operator satisfies $(\sqrt{N})(\sqrt{N}) = N$.

Hermitian operators: An operator H is *Hermitian* if it satisfies $H = H^\dagger$. We also call such operators *self-adjoint.* Hermitian operators are necessarily normal.

Positive operators: We say that A is a *positive operator* if $\langle\psi|A|\psi\rangle$ is a nonnegative real for all $\psi \in \mathbb{C}^n$. All positive operators are necessarily Hermitian and normal, which implies that the eigenvalues of a positive operator are nonnegative real.

Orthogonal Projectors: A *orthogonal projector* P is a Hermitian operator which satisfies $P^2 = P$. The set of all $|\psi\rangle$ such that $P|\psi\rangle = |\psi\rangle$ forms a subspace S, and if $B = \{|\phi_i\rangle\}$ is a basis for S, then $P = \sum_i |\phi_i\rangle\langle\phi_i|$.

2.1.2 Quantum Mechanics

We are now ready to review the mathematical formulation of quantum mechanics. We first give all of the basic definitions, and we follow this with some more advanced concepts that can be skipped on a first reading. All of the relevant terms can be found in the index.

We begin with a description of the system state. Quantum mechanics asserts that the state space of an isolated physical system corresponds to the set of norm one vectors in a Hilbert space. We call a vector of this form a *quantum state.*

In the cases we consider, we will assume that the dimension of this space is some finite n which is known to us, and that $\{|i\rangle : 0 \leq i < n\}$ is an orthonormal basis for the space.

The orthonormal basis vectors correspond to a set of properties of a state that are, in principle, perfectly distinguishable by measurements. For instance, the property could be the polarization of a photon, which could be distinguished by a set of polarizing filters. If the number of possible values for the property is two, then we call the system a *qubit*. Qubits have special significance since systems with less than two dimensions are trivial, and arbitrarily large finite-dimensional quantum state spaces can be built by tensoring qubits together.

We now describe how the system may behave while it is isolated. Let $|\psi_t\rangle$ be the state of a quantum state at some fixed time t, and let $|\psi_{t'}\rangle$ be the state of the same system at time $t' > t$. Then $|\psi_{t'}\rangle = U|\psi_t\rangle$ for some unitary U that does not depend on $|\psi_t\rangle$. We call U an *evolution operator*. Quantum mechanics conceivably permits any unitary U to be an evolution operator.

Without prior knowledge of a quantum state's preparation, we, as the external observers of a quantum state, must remove the state from isolation in order to obtain information about it. Quantum *measurement* is a formalization of this process. It is an inherently probabilistic process, in the sense that the outcome of a measurement is taken from a probability distribution. The randomness in measurement is not due to our ignorance of the system state, rather it reflects the true behaviour of the physical world in this circumstance. A quantum measurement can in general obtain only partial information about the quantum state, in the same way that a sample of a random variable gives only partial information about the exact distribution.

There are a number of ways to formally define measurements. The simplest and most relevant to our case is *projective measurements*. Let $S_1 \oplus \cdots \oplus S_k$ be a partition of the Hilbert space into orthogonal subspaces, and let P_i be the projector corresponding to S_i. Clearly, $\sum_i P_i = I$. Then the effect of measuring $|\psi\rangle$ with respect to $\{P_1, \ldots, P_k\}$ is twofold. First, the index i is communicated to the observer with probability equal to $\|P_i|\psi\rangle\|^2$. We call this index the *measurement outcome*. Secondly, the state changes (or *collapses*) to $P_i|\psi\rangle/\|P_i|\psi\rangle\|$, where i is the measurement outcome. Thus, a measurement will cause a change in the state unless $|\psi\rangle$ lies entirely within one of the S_i subspaces. For any such partition $S_1 \oplus \cdots \oplus S_k$ we can in principle construct an apparatus to implement the corresponding projective measurement.

Two quantum systems can be adjoined to make a single quantum system. Suppose wish to join two isolated systems over two Hilbert spaces V_1 and V_2. The Hilbert space associated with the combined system is the natural one induced by $V_1 \otimes V_2$. If the states of the two systems before the join is $|\psi_1\rangle$ and $|\psi_2\rangle$, the resulting state is $|\psi_1\rangle \otimes |\psi_2\rangle$.

Generalized Measurements and POVMs

Projective measurements characterize exactly the type of measurements which may be performed by interacting directly with a quantum system. However, it is possible to obtain more refined information about the closed system by first adjoining a second quantum system, applying a transformation to the combined system, and then removing the second system. These operations are exactly characterized by *generalized measurements.*

A generalized measurement is defined by a set of operators $\{N_i\}$ satisfying $\sum_i N_i^\dagger N_i = I$. On the application of such a measurement to state $|\psi\rangle$, the outcome of the measurement is i with probability $\|N_i|\psi\rangle\|^2 = \langle\psi|N_i^\dagger N_i|\psi\rangle$, in which case the state collapses to $N_i|\psi\rangle/\|N_i|\psi\rangle\|$. The special case of projective measurements corresponds to the case when the $N_i s$ are projection operators.

An alternative way to express generalized measurements is through *positive operator-valued measurements*, or *POVMs.* A POVM is expressed as a set of positive operators $\{E_i\}$ such that $\sum_i E_i = I$. The outcome of a POVM measurement on state ψ is the value i with probability $\langle\psi|E_i|\psi\rangle$. Observing that operators of the form $N_i^\dagger N_i$ are necessarily positive, we can convert a generalized measurement to a POVM by taking $E_i = N_i^\dagger N_i$ for all i. Conversely we can convert a POVM to a generalized measurement by taking $N_i = \sqrt{E_i}$.

Orthogonality and Perfect Distinguishability

If two state vectors $|\psi_1\rangle$ and $|\psi_2\rangle$ are orthogonal to each other, then they are *perfectly distinguishable* in the sense that there is exists a measurement which, when applied to state $|\psi_i\rangle$, will outputs i with probability 1. Clearly if $|\psi_1\rangle$ and $|\psi_2\rangle$ are orthogonal, then the measurement $\mathcal{M} = \{E_1, E_2\}$ defined by $E_1 = |\psi_1\rangle\langle\psi_1|$ and $E_2 = I - |\psi_2\rangle\langle\psi_2|$ is a distinguishing measurement.

Conversely, nonorthogonal states are not perfectly distinguishable. Suppose for the sake of contradiction that two nonorthogonal states $|\psi_1\rangle$ and $|\psi_2\rangle$ can be perfectly distinguished by some POVM $\{E_1, E_2\}$. Then $|\psi_2\rangle$ can be uniquely written as $|\psi_2\rangle = \alpha|\psi_1\rangle + \beta|\psi_2'\rangle$, where $|\alpha| > 0$ and $|\psi_2'\rangle$ is the projection of $|\psi_2\rangle$ onto the span of all vectors orthogonal to $|\psi_1\rangle$. The probability of obtaining outcome 1 on measuring $|\psi_2\rangle$ is:

$$\begin{aligned}
& \langle\psi_2|E_1|\psi_2\rangle \\
= \; & (\alpha^*\langle\psi_1| + \beta^*\langle\psi_2'|)E_1(\alpha|\psi_1\rangle + \beta|\psi_2'\rangle) \\
= \; & \alpha^*\alpha\langle\psi_1|E_1|\psi_1\rangle + \alpha^*\beta\langle\psi_2|E_1|\psi_1\rangle + \beta^*\alpha\langle\psi_1|E_1|\psi_2\rangle + \beta^*\beta\langle\psi_2|E_1|\psi_2\rangle \\
\geq \; & \alpha^*\alpha\langle\psi_1|E_1|\psi_1\rangle = \alpha^*\alpha > 0.
\end{aligned}$$

It is important to emphasize here that for two pairwise distinguishable states $|\psi_1\rangle$ and $|\psi_2\rangle$ it is possible for the system to arrive in some state $|\psi\rangle = \alpha|\psi_1\rangle + \beta|\psi_2\rangle$ for nontrivial α and β. This essential characteristic of quantum mechanics is called the *superposition principle*, and state $|\psi\rangle$ are said to be a *superposition* of $|\psi_1\rangle$ and $|\psi_2\rangle$.

Mixed States and Density Matrices

There are times when we would like to think of the state of the system as coming from some probability distribution $\mathcal{E} = \{(p_i, |\psi_i\rangle)\}$, where $|\psi_i\rangle$ occurs with probability p_i and $\sum_i p_i = 1$. We call $\mathcal{E}$ an *ensemble* of quantum states, or a *mixed state.*

There is a very useful formalism for representing mixed states via *density matrices.* For the ensemble $\mathcal{E}$ above, the corresponding density matrix ρ is:

$$\rho = \sum_i p_i|\psi_i\rangle\langle\psi_i|.$$

It is not hard to show that ρ is a positive operator with unit trace ($Tr(\rho) = 1$). Thus, ρ is normal and by the spectral decomposition it can be expressed as:

$$\rho = \sum_i \lambda_i |\phi_i\rangle\langle\phi_i|,$$

where $|\phi_i\rangle$ is the eigenvector corresponding to the eigenvalue λ_i and, since ρ is positive and has unit trace, it follows that the λ_i are nonnegative reals summing to 1. This observation has a number of interesting consequences. First, every ρ that is positive and has unit trace will correspond to some ensemble; in particular, the ensemble induced by the eigenvectors and eigenvalues of ρ. Thus, we can take the condition of being positive and unit trace as our definition for density matrices. Furthermore, since not all ensembles correspond to eigenvalue ensembles, it follows that two different ensembles may produce the same density matrix.

Density matrices allow us to succinctly describe the outcome of transformations and measurements on ensembles. If transformation U is applied to the ensemble $\mathcal{E}$, the resulting ensemble $\mathcal{E}' = \{p_i, U|\psi_i\rangle\}$ has density matrix $\rho' = \sum_i p_i U|\psi_i\rangle\langle\psi_i|U^\dagger = U(\sum_i p_i|\psi_i\rangle\langle\psi_i|)U^\dagger = U\rho U^\dagger$. In a similar way, one can show that a measurement $\{M_i\}$ on an ensemble with density matrix ρ yields the outcome i with probability $Tr(M_i\rho M_i^\dagger)$ and in this case the density matrix of the resulting ensemble is $M_i\rho M_i^\dagger/Tr(M_i\rho M_i^\dagger)$. Observe that measurement outcomes depend solely on the structure of the density matrix and not the initial ensemble. Any two ensembles having the same density matrix are therefore indistinguishable and so for many applications it is sufficient to identify an ensemble with its density matrix.

As a final note, we will sometimes need to refer to the space spanned by the eigenvalues of a density matrix ρ. This subspace is called the *support* of ρ, and we denote it $supp(\rho)$.

Von Neumann Entropy

Let X be a discrete random variable that takes values from $\{x_1, \ldots, x_n\}$, each i with probability p_i. Recall that the *Shannon entropy* of X is defined as:

$$H(X) = \sum_i -p_i \log p_i.$$

The Shannon entropy is a measure of the amount of uncertainty that we have in the value of a random variable. If X takes n possible values, then $H(X) \leq \log n$, with equality when X is uniformly distributed. For $p \in [0, 1]$ we use $H(p)$ as a shorthand to denote the entropy function of the Bernoulli random variable which takes value 1 with probability p and 0 otherwise.

Related important quantities include the *conditional Shannon entropy* $H(X|Y)$ of X given Y, which quantifies the uncertainty of X conditioned on knowing Y, and $I(X : Y)$, which is the *mutual information of* X *and* Y, which quantifies the amount of information that knowledge of X gives you towards knowledge of Y, or vice-versa. These quantities can be computed with the identities $H(X|Y) = H(X, Y) - H(Y)$ and $I(X : Y) = H(X, Y) - H(X) - H(Y)$, where $H(X, Y) = H(X \times Y)$.

The Von Neumann entropy measure is a generalization of this concept to mixed states:

Definition 2.1 *Suppose that ρ is a density matrix with spectral decomposition $\rho = \sum_{i=1} \lambda_k |\phi_i\rangle\langle\phi_i|$. Let X be a random variable with distribution $\{\lambda_1, \ldots, \lambda_k\}$. Then the Von Neumann entropy $S(\rho)$ of ρ is $H(X)$.*

In the case that an ensemble is made up of orthogonal vectors, from the Von Neumann entropy we get the Shannon entropy as a special case.

Von Neumann entropy, like Shannon entropy, is always nonnegative. Furthermore, for a mixed state over vectors in $\mathbb{C}^n$, the maximum possible entropy is $\log n$. The maximum is achieved when the density matrix can be expressed as $\rho = \frac{1}{n}\sum_{i=1}^{n} |\phi_i\rangle\langle\phi_i|$, where the $|\phi_i\rangle$'s are orthonormal eigenvectors of ρ. Such states are said to be *maximally mixed*.

Consider the application of a unitary U to the state ρ. Observe that this does not change the entropy of ρ since $\rho = \sum_i \lambda_i |\phi_i\rangle\langle\phi_i|$ and $U\rho U^\dagger = U\sum_i |\phi_i\rangle\langle\phi_i| U^\dagger = \sum_i \lambda_i |\phi_i\rangle\langle\phi_i|$. For a projective measurement $\{M_j\}$, the operation E defined by $\rho \mapsto \sum_j M_j \rho M_j^\dagger$ will satisfy $S(\rho) \leq S(E\rho)$.

The Von Neumann entropy is continuous with respect to natural distance measures for density matrices:

Lemma 2.2 *[46, Theorem 11.6] Let τ_0, τ_1 be two density matrices of dimension d and $\epsilon = \|\tau_0 - \tau_1\|_t = Tr(\tau_0 - \tau_1)$, $\epsilon < 1/3$. Then,*

$$|S(\tau_0) - S(\tau_1)| \leq \epsilon \log_2 d - \epsilon \log_2 \epsilon.$$

Completely Positive Superoperators

We have seen that the dynamics of a pure state is determined simply by unitary matrices. This transformation can be expressed in the density matrix formalism by the mapping:

$$\rho \mapsto U\rho U^\dagger. \tag{2.1}$$

In certain situations, we prefer a broader class of operations that include stochastic processes. Suppose, for example, that a measurement $\{M_j\}$ is made on a system in the mixed state ρ, and the outcome of the measurement is not known. Then the resulting state is transformed according to the rule:

$$\rho \mapsto \sum_j M_j \rho M_j^\dagger. \tag{2.2}$$

Such transformations may for example be used to model *decoherence*, which is the tendency of a pure quantum state to collapse (i.e. become measured) under pressure from the external environment. It may also be used to describe the future state of a system after a series of measurements, when the outcome of the intermediate measurements are not yet known.

Below we given a definition of *completely positive superoperators*, which capture the class of transformations that are permissible under the axioms of quantum mechanics:

Definition 2.2 *Let ρ be a $n \times n$ density matrix and let $E\rho$ be the density matrix of the state which results if we apply E. We say that E is a* completely positive superoperator *(CPSO) if:*

1. *The transformation $\rho \to E\rho$ is a linear transformation on the d^2-dimensional space of $d \times d$ matrices.*
2. *E is trace-preserving: $Tr(E\rho) = Tr(\rho)$.*
3. *E is* completely positive*, i.e. if H is the space on which E operates, then for any additional space H' the transformation $E \otimes I$ is a positive map on $H \otimes H'$.*

We will be particularly interested in CPSOs E corresponding to sequences of operations corresponding to alternating unitary transformations and measurements. It can be shown that such that CPSO of the form (2.1), or (2.2) in the case that the measurement is projective, imply $S(\rho) \leq S(E\rho)$. Conversely, any CPSO E satisfying $S(\rho) \leq S(E\rho)$ can be approximated using a series of unitary matrices and projective measurements.

Quantum Fourier Transform

The Fourier transform is a function which decomposes a periodic function into its frequency components. It is a fundamental analytical tool in many areas of mathematics, and has many applications in physics and engineering.

The Fourier transform also plays an important role in quantum computation. The efficient quantum algorithms for integer factoring and the discrete log problem both rely on the fact that we can use quantum computers to efficiently obtain certain useful information regarding the Fourier coefficients of a function. Both algorithms require that we can apply the Fourier transform in time polylogarithmic in the dimension of the space. Later in the thesis, we use the Fourier transform for a different purpose; namely to design of QFAs which use exponentially fewer states than the equivalent DFAs.

Let $\{|0\rangle, \ldots, |n-1\rangle\}$ be an orthonormal basis of $\mathbb{C}^n$. The *Quantum Fourier Transform* (QFT) of size n, denoted F_n, is the linear function which acts on basis vectors as:

$$F_n|j\rangle = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} e^{\frac{2\pi ijk}{n}} |k\rangle.$$

This is a unitary matrix and so it can be implemented as an evolution operator.

The QFT defined above is associated with the group $\mathbb{Z}_n$ of integers mod n with $+$ as the operation. It is a special case of the *Abelian Quantum Fourier Transform*, which we will describe below.

Let G be an abelian group, with $|G| = n$. We denote by $\mathbb{C}[G]$ the set of all functions $f : G \to \mathbb{C}$. By fixing an orthonormal basis $\{|g\rangle\}_{g\in G}$ of $\mathbb{C}^n$, We can naturally associate functions of this type to vectors in $\mathbb{C}^n$.

A *character* of G is a homomorphism $\chi : G \to \mathbb{C}$. Any two distinct characters χ_1, χ_2 will satisfy $\frac{1}{n}\sum_g \overline{\chi_1(g)}\chi_2(g) = 0$. Also, $\overline{\chi(g)}\chi(g) = 1$ for all χ, g, and $\frac{1}{n}\sum_g \overline{\chi(g)}\chi(g) = 1$. There will be n characters for the group G, thus if we scale each of them by $\frac{1}{\sqrt{n}}$ we get an orthonormal basis of $\mathbb{C}^n$.

The set of characters form a group $\hat{G}$ under product, which is called the *dual group* of G. This group will be isomorphic to G. Let χ_g be the character associated with g under an isomorphism. Then for functions $f \in \mathbb{C}[G]$, the abelian Fourier transform $\hat{f} : \hat{G} \to \mathbb{C}$ of f is $\hat{f}(\chi_g) = \sum_{g'} \chi_g(f(g'))$. In other words, $\hat{f}$ is an expression of the function f in the basis of characters. Furthermore, the transformation $f \mapsto \hat{f}$ can be inverted according to the formula $f(g) = \frac{1}{n}\sum_\chi \hat{f}(\chi)\chi(g^{-1})$

Up to the normalization factor, the transformation $f \mapsto \hat{f}$ corresponds to the function F_n above, where the group in this case is $\mathbb{Z}_n$ and $|j\rangle$ on the left hand side represents the basis vector χ_j. Likewise the abelian QFT F_G of G is defined as follows:

$$F_G|g\rangle = \frac{1}{\sqrt{n}}\sum_{g'} \chi_{g'}(g)|g'\rangle.$$

2.1.3 Historical Development of Quantum Computation

Whenever a mathematical model is made to formalize the behavior of a physical process, certain implicit assumptions are made about the underlying physical system. Turing, in his famous paper [65], argued at length that the assumptions made for his computation model were motivated by the real limitations of an automated process.

Although he was concerned only with computations, developments in the 1970's led researchers to formally consider notions of computability in resource-bounded settings. The class P of decision problems solvable in polynomial time on a Turing machine were found to be of central interest for a number of reasons. It is closed under composition, it contains many nontrivial problems, and polynomial time computability on a Turing machine corresponds exactly to polynomial time computability on random access machines, and thus on most physical implementations of computing devices. This cemented P as the *de facto* standard of efficient deterministic computation.

However, this is not the only reasonable model of efficient computation. For example, we may consider a machine which is permitted to make random choices based on the outcome of a sequence of unbiased coin tosses. The class BPP [47] of *bounded-error probabilistic polynomial time computable* languages consists of those languages L for which there exists a randomized algorithm and a probability bound $p > \frac{1}{2}$ such that all inputs are correctly classified by the randomized algorithm with probability at least p. Clearly P is contained in BPP, but there are languages in BPP for which no P algorithm is known. For example, the problem of testing whether a matrix of multivariate polynomials is nonsingular is in BPP but is not known to be in P.

The classes BPP and P may not be ideally suited to describe computation at a microscopic level. As we move from macroscopic to microscopic physical systems, the dominant physical rules begin to change. This has motivated researchers to reconsider the notion of efficient computation in different physical environments. An important early insight was provided by Landauer [41], who considered the thermodynamics of computing systems that work in a *closed* environment. A closed physical system is one which can exchange heat with the external environment but not matter. Landauer showed that implementing certain information-theoretic tasks requires a minimum amount of thermodynamic activity. In particular, he argued that the operation of erasing, i.e. setting a bit of arbitrary value to zero, would imply a decrease in entropy and thus require a dissipation of at least a fixed constant amount of heat energy. This is important considering the computational device involving many erasures would require some minimal interface for energy dissipation. Furthermore, it implies that *isolated* systems, such as those which are modeled by pure quantum states, are incapable of such erasures.

This motivated the investigation of *reversible computation*, which is the study of the power of computation devices that do not allow erasure. Bennett [10] showed that for every Turing machine computing a function, there is a *reversible* Turing machine computing an equivalent function with a linear factor overhead. Fredkin and Toffoli [29] demonstrated that there exists a simple reversible gate which is universal for computation, and showed that a circuit composed of binary AND, OR, and NOT gates can be converted into a reversible circuit with linear overhead in the depth. Later, Toffoli [64] completed the picture by showing that the reversible gates themselves can be implemented in such a way that state transitions can be achieved in a continuous reversible motion. These results were used by Benioff [9] to show that the transition function of a reversible Turing machine could be expressed as an evolution operator.

Researchers soon found evidence that quantum computers could potentially be superpolynomially faster at certain tasks when compared to Turing machines. Feynman [28] suggested that there were inherent limitations to simulations of quantum processes by classical machines, and thus quantum mechanics holds a power of computation that is not captured by existing devices. Bennett and Brassard [12] demonstrated the existence of provably secure public cryptography using quantum states.

Deutsch [25] was the first to formalize the notion of a quantum computer. His model of *quantum Turing machines*, or QTMs, is the natural extension of a Turing machine to quantum amplitudes. It is similar in this respect to a random Turing machine. Recall that a *configuration* of a Turing machine $T = (Q, q_0, \Sigma, \Gamma, \delta, F)$ is a tuple $(q, t, i) \in Q \times \gamma^* \times \mathbb{Z}^+$ which finitely describes the machine state, tape contents, and head position of an intermediate state of the machine. The state of a quantum Turing machine is a pure state over the basis of possible configurations of a regular Turing machine. While a random Turing machine makes one of several possible legal transitions, each with a certain probability, the quantum Turing machine replaces probability with amplitude. As in the random Turing machine model, the amplitude of the transition must depend only on the letter under the tape head and the value of the finite state. Thus, a set of such amplitudes for each possible legal transitions completely specifies the machine. The final condition is that the induced transition function must be unitary on the vector space of possible configurations.

The main accomplishment of Deutch's QTM is the definition of a quantum computational model which is universal for itself and which is sufficiently powerful to implement quantum informational tasks of interest, such as an EPR test or the implementation of a quantum cryptographic scheme. Bernstein and Vazirani [13] made a number of enhancements to this formalism. They first resolved several concerns regarding the QTM model. First, they showed that one can restrict the class of allowable transformations to those with coefficients taken from a fixed set. This is important because it is unreasonable to assume that machines using transitions with arbitrary coefficients can be constructed. They also showed that, to approximate the behavior of a given QTM M for T steps to within a factor ϵ, it is sufficient that the transitions be implemented to within $O(\log T)$ bits of accuracy. Furthermore, they showed that the universal simulation of a QTM could be implemented in a polynomial number of QTM steps. However, there are still significant drawbacks to the QTM model. Firstly, there is no known way to check whether a given QTM specification is well-formed, in the sense that the local transition function δ extends to a unitary transformation on the space of Turing machine configurations. Furthermore, primitives such as branching and looping which are fundamental in most other computational models can only be adapted to special cases of QTM, so much of our intuition about computation does not help us to understand the power of QTMs.

Yao later advocated a simpler model of quantum computation called *quantum circuits* [69]. The quantum circuit model is a special case of one considered earlier by Deutsch [26], which he called *quantum computational networks.* The more general model allowed feedback loops.

Recall that a qubit is a two-dimensional quantum system. We use $\{|0\rangle, |1\rangle\}$ as the basis of a qubit. In the quantum circuit model, operations are performed on a system of a finite set of qubits tensored together. For a quantum circuit, we fix a set of quantum *gates*, each of which are unary operators on a space of ℓ qubits for some fixed ℓ. The circuit is then specified by a sequence of quantum gates operating on each step, one can either apply either a unitary operation on a subset of the qubits, or perform the measurement $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ on a single qubit. A quantum circuit can be used to probabilistically compute a boolean function $f : \{0,1\}^n \to \{0,1\}$ by setting the initial state to $|x_1\rangle \cdots |x_n\rangle$, where x_i is the ith input bit, and setting the output of the machine to be the outcome of a qubit measurement.

Quantum circuits hold many advantages over quantum Turing machines. Unlike in the QTM case, any quantum circuit will be automatically well-formed. Furthermore almost all quantum algorithms and quantum information tasks of interest can be expressed naturally using quantum circuits. Quantum circuits can be simulated by a QTM, and a quantum circuit can be used to simulate the behavior of a QTM for a fixed number of steps.

2.1.4 Quantum Finite Automata

Quantum Finite Automata are abstract models of physical computation devices in the setting of online computation. An online computation device is one which receives its input as a series of input signals, where each input signal is taken from some finite set Σ. The machine is understood to have an internal state, and each input signal changes the state of the machine in a way that depends on the current input signal and the current state of the machine. In order to give such machines a mathematical treatment, certain assumptions have to be made regarding the underlying physical rules. We will call such a collection of assumptions a *model.* A central question in this framework is then: what languages $L \subseteq \Sigma^*$ can be recognized by these machines?

The most important model of online computation is the Deterministic Finite Automata, or DFA. Each DFA M has a finite set Q of possible internal states, and at all times during the computation M will be in some state $q \in Q$. When the input signal $\sigma \in \Sigma$ is received, the machine M will change its state according a fixed transition function $\delta : Q \times \Sigma \to Q$.

It is important to note that the definition DFAs do not fit all finite memory physical computation devices that we may wish to consider. The model of *randomized finite automata* is more appropriate, for example, if we wish to discuss a finite state machine which makes transition errors with some probability $\epsilon > 0$. In this case, the state of a machine at any given time is a random variable. It is in this same spirit that Quantum Finite Automata are defined.

The simplest model of QFAs is the one given by Crutchfield and Moore [44], which we call *Measure-Once QFAs*, or MOQFAs. This name refers to the fact that the state remains unobserved until the end of the computation, at which point a measurement is made to determine whether the given input word should be accepted or rejected.

Measure-Once QFA (MOQFA) An instance of an MOQFA is given by a tuple $M = (Q, q_0, \Sigma, \{U_\sigma\}, F)$, where Q is a finite set with $|Q| = n$, q_0 is a distinguished *start state*, $\{U_\sigma\}$ is a set of state transitions, and F is the set of *accepting states.*

The elements of Q correspond to a set of n physical states which are pairwise perfectly distinguishable as discussed in Section 2.1.2. For each $q \in Q$ we associate a vector $|q\rangle$ from an orthonormal basis $\{|q\rangle\}_{q\in Q}$ of $\mathbb{C}^n$. The state of a MOQFA at any time is a superposition of the $|q\rangle$'s.

The working alphabet will be $\Sigma \cup \{¢, \$\}$, where ¢ and \$ are distinguished start and end markers, respectively. For every $\sigma \in \Sigma \cup \{¢, \$\}$ we associate a unitary operator U_σ on $\mathbb{C}^n$. We use the set F to define a measurement $\{P_{acc}, P_{rej}\}$ with $P_{acc} = \sum_{q\in F} |q\rangle\langle q|$ and $P_{rej} = \sum_{q\notin F} |q\rangle\langle q|$. When a letter σ is read, the state is transformed from $|\psi\rangle$ to $U_\sigma|\psi\rangle$. The operators $U_{¢}$ and $U_{\$}$ correspond to preprocessing and postprocessing of the machine. The machine is initialized to the state $U_{¢}|q_0\rangle$ before the input word is read. On input $w = w_1 \dots w_m$ the machine moves to state $|\psi_w\rangle = U_{w_m} \cdots U_{w_1} U_{¢}|q_0\rangle$. When the final input character is read, the operator $U_{\$}$ is applied to the state and the resulting state is measured with respect to $\{P_{acc}, P_{rej}\}$. If the outcome of the measurement is *acc*, the machine accepts, otherwise it rejects.

Observe that the output of the machine on input w is, in general, a random variable. Thus we say that M *recognizes* L with probability p if every $w \in L$ ($w \notin L$) is accepted (rejected) with probability $p > \frac{1}{2}$. This will be the standard mode of recognition.

Let $\Sigma = \{a, b\}$. As an example, consider the language: $L_m = \{w : |w|_a \mod m = 0\}$, where $|w|_a$ denotes the number of occurrences of the letter a in w. Here is a simple MOQFA to recognize L_m. Let $M = (Q, q_0, \Sigma, \{U_\sigma\}, F)$, where $Q = \{0, 1, \ldots, m-1\}$, $q_0 = 0$, $F = \{0\}$, $U_{¢} = U_{\$} = U_b = I$ and U_a is the unique linear operator such that $U_a|i\rangle = |i + 1 \mod m\rangle$ for all i. It is easy to check by induction that U_a is unitary and $|\psi_w\rangle = ||w|_a \mod m\rangle$ for all w. Thus there is an m-state MOQFA recognizing L_m with probability $p = 1$.

Fix an MOQFA M over the alphabet Σ. We say that the state $|\psi\rangle$ is reachable if there exists a $w \in \Sigma^*$ such that $|\psi\rangle = |\psi_w\rangle$. The set of reachable states of an MOQFA are possibly infinite, so one might wonder to what extent this is indeed a 'finite' machine. Let us try to resolve this issue with a few relevant facts about these machines. When we insist that M recognizes a language with bounded probability $p > \frac{1}{2}$, there will exist vectors $|\psi\rangle$ and constants $\delta < 1$ such that there is no word w satisfying $\langle\psi_w|\psi\rangle > \delta$. Consider, for example, two states $|\psi_a\rangle$ and $|\psi_r\rangle$ such that $U_{\$}|\psi_a\rangle \in S_{acc}$ and $U_{\$}|\psi_r\rangle \in S_{rej}$. These states exist so long as P_{acc} and P_{rej} are nontrivial. Now consider the state $|\psi\rangle = \frac{1}{\sqrt{2}}(|\psi_a\rangle + |\psi_r\rangle)$. The neighborhood around this state is not reachable, otherwise there would be a word w which is accepted with probability p' such that $(1-p) < p' < p$, a contradiction. Furthermore, we can show:

Theorem 2.3 *If M recognizes L with probability $p > \frac{1}{2}$, then L is regular.*

Proof: The right equivalence relation for a language L is the relation $\sim_{L,r}$ on Σ^* defined by $x \sim_{L,r} y$ if for all $u \in \Sigma^*$ we have $xu \in L \Leftrightarrow yu \in L$. By the Myhill-Nerode Theorem [36], a language is regular if and only if the number of right equivalence classes is finite. Thus it is sufficient to show that if an MOQFA M recognizes L, then $\sim_{L,r}$ has finitely many such classes.

Let $U_w = U_{w_m} \cdots U_{w_1}$ for $w = w_1 \dots w_k$. Let n be the dimension of M's state space, and let $S_{acc} \oplus S_{rej}$ be the partition of $\mathbb{C}^n$ into the accepting and rejecting subspaces. For word w define $S_{w,acc} = U_w^\dagger S_{acc}$ and $S_{w,rej} = U_w^\dagger S_{rej}$. Then $S_{w,acc} \oplus S_{w,rej}$ is also an orthogonal decomposition of $\mathbb{C}^n$. Finally, define P_μ to be the projection operator into space S_μ.

Suppose $x \not\sim_{L,r} y$. Then without loss of generality there exists a word u such that $xu \in L$ but $yu \notin L$. We show that this implies a bounded distance between states $|x\rangle = U_x|q_0\rangle$ and $|y\rangle = U_y|q_0\rangle$. Since M recognizes L with probability p, we have:

$$\begin{aligned} \|P_{u,acc}|x\rangle\|^2 &\geq p \qquad && (\|P_{u,rej}|x\rangle\|^2 < (1-p)), \\ \|P_{u,rej}|y\rangle\|^2 &\geq p \qquad && (\|P_{u,acc}|y\rangle\|^2 < (1-p)). \end{aligned}$$

Define $|x'\rangle = |x\rangle - |y\rangle$. Then:

$$\begin{aligned} & p \leq \|P_{u,acc}|x\rangle\|^2 \leq \|P_{u,acc}|y\rangle\|^2 + \|P_{u,acc}|x'\rangle\|^2 \\ \Longrightarrow \quad & p \leq (1-p) + \|P_{u,acc}|x'\rangle\|^2 \\ \Longleftrightarrow \quad & \|P_{u,acc}|x'\rangle\|^2 \leq 2(p - \frac{1}{2}) \end{aligned}$$

Thus, states $|x\rangle$ and $|y\rangle$ must be distance at least $\sqrt{2(p-\frac{1}{2})}$ apart. But for any fixed integer n and distance $d > 0$, the number of pairwise distance d vectors of norm 1 in $\mathbb{C}^n$ is finite, and so the number of right congruence classes in L are finite and we are done. □

Kondacs Watrous QFA (KWQFA): A Kondacs-Watrous [38] QFA is defined by a tuple $M = (Q, \Sigma, \{A_\sigma\}, q_0, Q_{acc}, Q_{rej})$, where Q_{acc} and Q_{rej} are disjoint. Define $Q_{non} = Q - (Q_{acc} \cup Q_{rej})$. When a symbol σ is read, the machine applies the unitary transformation A_σ to the state, and then measures with respect to:

$$\left\{ P_{acc} = \sum_{q \in Q_{acc}} |q\rangle\langle q|, \quad P_{rej} = \sum_{q \in Q_{rej}} |q\rangle\langle q|, \quad P_{non} = \sum_{q \in Q_{non}} |q\rangle\langle q| \right\}.$$

If the measurements outputs acc (resp. rej), then the machine halts and accepts (resp. rejects) the input. Otherwise the machine continues. We require that the probability that the machine has not halted after processing the \$ symbol is 0.

Brodsky-Pippenger QFA (BPQFA): Brodsky and Pippenger [20] considered a number of different variations of QFAs, including this special case of KWQFA. A BPQFA is given by a tuple $M = (Q, \Sigma, \{A_\sigma\}, q_0, Q_{acc}, Q_{rej})$ as in the KWQFA model, with two changes. First, we additionally require that the machine does not transition to an accepting state until the endmarker is read. Second, we say that a BPQFA M recognizes L if each word $w \in L$ is accepted with probability $p > 0$, and each word $w \notin L$ is rejected with certainty.

Latvian QFA (LQFA): Defined by Ambainis et al [3], an LQFA is a tuple $M = (Q, \Sigma, \{A_\sigma\}, \{P_\sigma\}, q_0, Q_{acc})$, where the $\{A_\sigma\}$ are unitary matrices and $\{P_\sigma\}$ are measurements (each P_σ consists of a finite set of projections $\{P_{\sigma,i}\}$ satisfying $\sum_i P_{\sigma,i} = I$. We require that $P_\$$ is the measurement $\{P_{acc} = \sum_{q \in Q_{acc}} |q\rangle\langle q|, P_{rej} = \sum_{q \notin Q_{acc}} |q\rangle\langle q|\}$, and the machine accepts or rejects according to the outcome of this measurement. The mode of recognition for this machine is bounded error. LQFAs are permitted to perform arbitrary measurements before the end, however the machine cannot accept before reading the entire input.

Generalized QFA (GQFA): Introduced by Nayak [45], GQFAs generalize both the LQFA's ability to apply a different projective measurement for each letter, and the KWQFA's ability to halt before the end of the input. An instance of a GQFA is given by a tuple $M = (Q, \Sigma, \{A_\sigma\}, \{P_\sigma\}, q_0, Q_{acc}, Q_{rej})$. On input σ, the machine applies the unitary A_σ, then the measurement P_σ. Then, as in the case of KWQFAs, a measurement $\{P_{acc}, P_{rej}, P_{non}\}$ is made. If the output is *non*, then the machine reads the next letter. Otherwise, the machine halts and accepts or rejects accordingly.

We have made a slight change to the definition. The original definition allowed the machine to apply a sequence of ℓ alternating transformations and measurements for each letter. This does not effect the computational power of GQFAs since we can simulate a sequence of ℓ transformations and measurements by one transformation and measurement (Claim 1).

2.2 Algebraic Automata Theory

In this section we give a brief introduction to the central concepts in algebraic automata theory, and also present some known facts which are relevant to our investigation. For a more complete treatment, we recommend [52, 48].

2.2.1 Automata as Monoids

Let us first recall some fundamental definitions of semigroup theory. A *semigroup* is a set S equipped with a binary associative operation $\cdot_S$. A *monoid* M is a semigroup with a distinguished *identity element* 1 satisfying $1 \cdot_M m = m \cdot_M 1 = m$

for all $m \in M$. A simple example of a semigroup is the familiar set Σ^+ of all finite nonempty strings over alphabet Σ with concatenation as the binary operation. Likewise, the set Σ^* forms a monoid with the empty word as the identity element. To simplify notation, we often write the product of two elements m and n as mn.

A *subsemigroup* S' of S is a subset of S which is closed under the operation $\cdot_S$. A *submonoid* is a subset M' of a monoid M which is closed under $\cdot_M$ and forms a monoid under that operation. For semigroups S and T, the *direct product* of S and T is the semigroup with set $S \times T$ and operation $(s,t) \cdot_{S \times T} (s', t') = (s \cdot_S s', t \cdot_T t')$. The direct product of two monoids M and N naturally forms a monoid. For two semigroups S and T, a *morphism* is a function $\varphi : S \to T$ such that $\varphi(s \cdot_S s') = \varphi(s) \cdot_T \varphi(s')$. A *monoid morphism* is a semigroup morphism of monoids that additionally satisfies $\varphi(1_S) = 1_T$.

For a semigroup S we denote by S^1 the monoid formed from S by adding an identity element to S if no such element exists. Let S be a semigroup and let $\sim$ be an equivalence relation on monoid elements. We denote by $[s]$ the equivalence class of s. We say that $\sim$ is *stable* if for all $s, t \in S$ and $u, v \in S^1$ we have that $s \sim t$ implies $usv \sim utv$. In this case, the set of equivalence classes forms a semigroup under the operation $[s] \cdot [t] = [st]$. We call this semigroup the *quotient semigroup of* $\sim$ and it is denoted $S/\!\sim$.

For any set E, the set of all functions $f : E \to E$ form a monoid $T(E)$ under function composition, with the identity function as the identity element. We say that M is a *transformation monoid* if it is a submonoid of $T(E)$ for some E. There is a natural transformation monoid associated with the transition function of a DFA. Let

$A = (Q, q_0, \Sigma, \delta, F)$ be such an automaton. For every word w, the transition function δ induces a function $\delta|_w : Q \to Q$, and the set M_A of all such functions forms a monoid under composition. We call M_A the *transition monoid* of A. Furthermore, there is a natural morphism $\varphi : \Sigma^* \to M_A$ defined by $\varphi(w) = \delta|_w$ for all $w \in \Sigma^*$.

This motivates an algebraic reformulation of the notion of recognition by a finite automaton. We say that a language $L \subseteq \Sigma^*$ is *recognized by a monoid* M if there exists a morphism $\varphi : \Sigma^* \to M$ and a set $F \subseteq M$ such that $\varphi^{-1}(F) = L$. Let A be an automaton recognizing L. Then M_A recognizes L via the natural morphism and with $F = \{\delta|_w : \delta|_w(q_0) \in Q_{acc}\}$. Conversely, if L is recognized by a morphism $\varphi : \Sigma^* \to M$ for finite M, it is easy to construct a finite automaton that determines the value of $\varphi(w)$ for a given $w \in \Sigma^*$. Thus we have the following variation of Kleene's theorem:

Theorem 2.4 *A language $L \subseteq \Sigma^*$ is regular if and only if it is recognized by some finite monoid.*

We will see that this perspective allows us to parameterize Kleene's theorem by considering the class of languages recognized by subclasses of monoids.

We say that monoid N *divides* monoid M (denoted $N \preceq M$) if there exists a surjective morphism from some submonoid M' of M onto N. The division relation is a partial order on the isomorphism classes of the set of finite monoids. The significance of monoid division in this context is that it preserves language recognizability in the following sense: if N recognizes L and N divides M, it follows that M also recognizes L.

For every language L there is a monoid $M(L)$ recognizing L that is minimal with respect to the division relation; in other words, a monoid M recognizes L if and only if $M(L) \preceq M$. This monoid is called the *syntactic monoid* of L. It can be constructed as follows: For a language $L \subseteq \Sigma^*$ let $\equiv_L$ be the congruence on Σ^* defined by $x \equiv_L y$ if for all $u, v \in \Sigma^*$ we have $uxv \in L \Leftrightarrow uyv \in L$. We call $\equiv_L$ the *syntactic congruence* of L. Then the syntactic monoid is $\Sigma^*/\equiv_L$. Furthermore, if L is regular then the transition function of the minimal automaton for L is isomorphic to the syntactic monoid of L.

2.2.2 The Variety Theorem

In this section we introduce Eilenberg's variety theorem. The word *variety* is used here to denote a class of algebraic structures that are natural in the sense that they satisfy certain natural closure properties. This notion was originally used for classes of algebras over infinite sets by Birkhoff [18], but Eilenberg adapted this notion to the case of finite sets. He defines a *variety of finite monoids* to be a class of finite monoids closed under (finite) direct product and division. Many natural classes of finite monoids form varieties, for example the class $\mathbf{G}$ of monoids which are groups. Eilenberg's varieties are sometimes called *pseudovarieties* to distinguish them from the infinite case, but we will refer to them simply as varieties.

For a language $L \subseteq \Sigma^*$ and $w \in \Sigma^*$, the *left quotient* of w with L to be the language $w^{-1}L = \{x : wx \in L\}$. The *right quotient* Lw^{-1} is defined in a similar way.

We say that a class of languages $\mathcal{V}$ forms a *variety of languages* if it is closed under boolean operations, inverse homomorphisms, and word quotient[1] .

Let $\mathbf{V}$ be a variety of monoids. We write $\mathbf{V} \rightarrow \mathcal{V}$ if $\mathcal{V}$ is the class of all regular languages which can be recognized by some monoid in $\mathbf{V}$. This class will form a variety of languages. The *Variety Theorem [27]* establishes a strong relationship between these classes:

Theorem 2.5 *(Variety Theorem) The correspondence* $\mathbf{V} \rightarrow \mathcal{V}$ *is a one-to-one correspondence between varieties of monoids and varieties of languages.*

We mention at this point that there is a parallel theory of automata as semigroups, the distinction being that in the monoid theory the empty word is treated as a valid input but in the semigroup theory it is not. In other words, the monoid theory characterizes languages $L \subseteq \Sigma^*$ and the semigroup theory characterizes languages $L \subseteq \Sigma^+$. We will present only the monoid theory here, but from time to time we will point out the places in which the two theories diverge.

2.2.3 Structural Properties of Monoids

In this section we consider the main structural properties of finite monoids. We begin by considering an important type of substructure, which are the semigroups generated by a single element. Let x be the generator. Since the monoid is finite, there is a minimal k and p such that $x^{k+p} = x^k$. Thus the generated subsemigroup will have the shape of Figure 2–1.

[1] Formally, $\mathcal{V}$ associates to each finite alphabet Σ a class of regular languages over Σ^*.

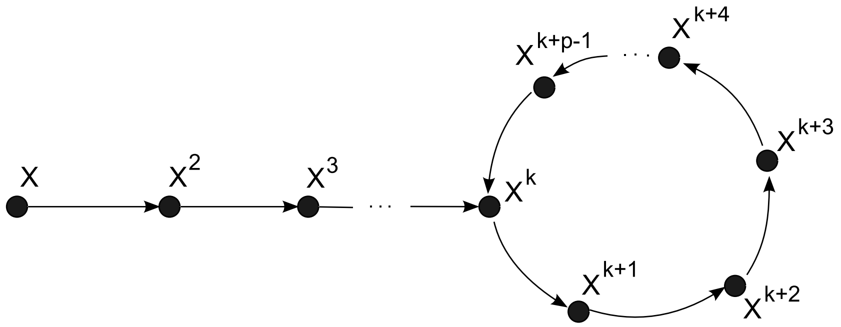


Figure 2–1: The structure of a finite semigroup with a single generator.

An *idempotent* is an monoid element e which satisfies $e^2 = e$. The elements $x_k, \ldots, x_{k+p-1}$ form a cyclic group, and the identity of this cyclic group is the unique idempotent in the subsemigroup generated by x. Conversely, every idempotent e in M forms a one-element subsemigroup (in fact, a submonoid).

We next consider the structural properties of monoid ideals. For a monoid M we say that the set $I \subseteq M$ forms an *ideal* of M if $MIM \subseteq I$. Likewise a right ideal (resp. left ideal) is a set $I \subseteq M$ such that $IM \subseteq I$ (resp. $MI \subseteq I$). For a subset S of M, it is easy to see that MSM is an ideal of M, and is in fact smallest ideal of M containing S. Similarly MS and SM are the minimal left and right ideals of S.

Green's relations are a series of equivalence relations on monoid elements defined in terms of the minimal ideals of these elements. They are given below:

- $x \mathcal{J} y$ if $MxM = MyM$,
- $x \mathcal{R} y$ if $xM = yM$,
- $x \mathcal{L} y$ if $Mx = My$,
- $x \mathcal{H} y$ if $x \mathcal{R} y$ and $x \mathcal{L} y$.

Green's relations are also defined for semigroups. For $s, t \in S$ we say that $s\,\mathcal{J}t$ if $S^1 s S^1 = S^1 t S^1$, and the other relations are defined in a similar way. All of the facts in this section also hold in the semigroup case.

A simple but useful fact about the relation $\mathcal{J}$ is that $x\,\mathcal{J}y$ if and only if there exists s, t, u, $v \in M$ such that $x = syt$ and $y = uxv$. A similar property holds for the other relations. The $\mathcal{R}$ and $\mathcal{L}$ relations are left and right compatible with multiplication respectively, i.e. $x\,\mathcal{R}y$ implies $mx\,\mathcal{R}my$ and $x\,\mathcal{L}y$ implies $xm\,\mathcal{R}ym$.

Corresponding to each of Green's relations, we also define the preorder relations $\leq_{\mathcal{J}}$, $\leq_{\mathcal{R}}$, $\leq_{\mathcal{L}}$, and $\leq_{\mathcal{H}}$. We say that $x \leq_{\mathcal{J}} y$ if $MxM \subseteq MyM$, and the other relations are likewise defined. Furthermore, we define $\mathcal{J}_x$ (resp. $\mathcal{R}_x, \mathcal{L}_x, \mathcal{H}_x$) to be the $\mathcal{J}$-equivalence (resp. $\mathcal{R}$, $\mathcal{L}$, $\mathcal{H}$- equivalence) class containing x.

Observe that the relation $\mathcal{H}$ refines $\mathcal{L}$ and $\mathcal{R}$, and the relations $\mathcal{R}$ and $\mathcal{L}$ each refine the $\mathcal{J}$ relation. The following lemma gives a natural bijective mapping between the $\mathcal{R}$, $\mathcal{L}$, and $\mathcal{H}$ equivalence classes within a $\mathcal{J}$-equivalence class:

Lemma 2.6 *(Green's Lemma) let s and t be such that $s\,\mathcal{R}t$, and let $u, v \in M$ be such that $su = t$ and $tv = s$. Define $\rho_u : M \to M$ by $\rho_u(x) = xu$ and likewise define $\rho_v : M \to M$ by $\rho_v(x) = xv$. Then ρ_u and ρ_v are bijections from $\mathcal{L}_s$ to $\mathcal{L}_t$ that preserve the $\mathcal{R}$ classes. The symmetric result holds for s and t satisfying $s\,\mathcal{L}t$.*

Let J be a set of elements corresponding to a $\mathcal{J}$ equivalence class of some monoid M. Consider a table with each row corresponding to a $\mathcal{R}$ class within J and each column as an $\mathcal{L}$ class, with $\mathcal{H}$ classes at the intersections. An example of such an 'egg-box picture' is shown in Figure 2–2. An immediate consequence of Green's

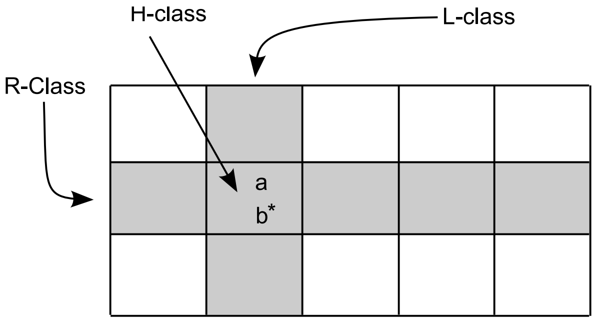


Figure 2–2: Diagram of a $\mathcal{J}$-equivalence class.

lemma is that there are an equal number of elements in each $\mathcal{H}$-class contained within a $\mathcal{J}$-class.

The properties of idempotent elements within $\mathcal{J}$-classes is of fundamental importance to the structure of $\mathcal{J}$-classes. In the remainder of this section we present a few such properties that will be used later in the thesis. The proofs are straightforward and can be found for example in [48], but we will give proofs for the first two lemmas to give a flavor of the technique.

Lemma 2.7 *For $x \in M$ and $e = e^2 \in M$, $x \leq_{\mathcal{R}} e$ if and only if $x = ex$ (resp. $x \leq_{\mathcal{L}} e$ if and only if $x = xe$).*

Proof: $x \leq_{\mathcal{R}} e$ implies that there exists a u such that $eu = x$. So then $x = eu = eeu = ex$. The converse is immediate, and the case of $\leq_{\mathcal{L}}$ is symmetric. □

Lemma 2.8 *If J is a $\mathcal{J}$-class that contains an idempotent, then every $\mathcal{R}$-class and $\mathcal{L}$-class in J contains an idempotent.*

Proof: Suppose $e \in J$ is an idempotent and let $e\mathcal{R}a$. Then there is a u such that $au = e$, and $ea = a$ by Lemma 2.7. Then $\mathcal{L}_a$ contains an idempotent uea since

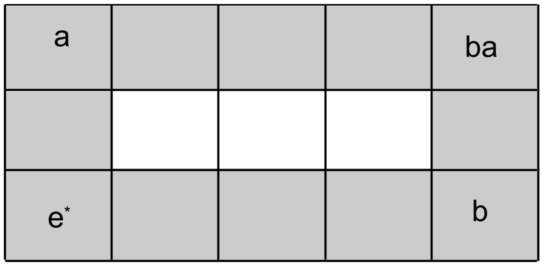


Figure 2–3: The relationship in Lemma 2.9.

$auea = a$ and $a \leq_{\mathcal{L}} uea \leq_{\mathcal{L}} auea = a$. Likewise every $b \,\mathcal{L} e$ is such that $\mathcal{R}_b$ contains an idempotent. □

Lemma 2.9 *Let a and b be monoid elements such that $b \in \mathcal{J}_a$. Then $ab \in \mathcal{J}_a$ (in particular, $ab \in \mathcal{R}_a \cap \mathcal{L}_b$) if and only if there is an idempotent in $\mathcal{R}_b \cap \mathcal{L}_a$.*

A diagram of this relationship is presented in Figure 2–3.

2.2.4 Identities

Let Γ be a infinite countable set and let $u, v \in \Gamma^*$. Then $u = v$ is a *monoid equation*. We interpret the words u and v as products of variables which take values from some monoid M. We say that M *satisfies* the equation $u = v$ if any valid substitution of the letters of u and v by elements in M leads to equality.

Several algebraic properties of monoids can be expressed succinctly in terms of monoid equations. For instance, a monoid is commutative if and only if the equation $xy = yx$ is satisfied, and a monoid is a *semilattice* if and only if it is idempotent $xx = x$ and commutative $xy = yx$. The class of commutative monoids and of semilattices form varieties, denoted **Com** and $\mathbf{J_1}$ respectively. We say that a defining equation of a variety is an *identity* for that variety.

Observe that equation varieties are preserved by the variety closure properties. For example, an equation satisfied by M and N will also be satisfied by $M \times N$. Conversely, all monoid varieties can be characterized equationally using a certain extension of the equational framework. We outline this extension below.

For strings $u, v \in \Sigma^*$ we define $r(u, v)$ to be the size of the smallest monoid that does not satisfy $u = v$, and we set $r(u, v) = \infty$ if u and v are equal. Then the function $d : \Sigma^* \times \Sigma^* \rightarrow [0, 1]$ defined by $d(u, v) = 2^{-r(u,v)}$ is a metric over strings in Σ^*. We denote by $\widehat{\Sigma}^*$ the set of limits of Cauchy sequences with respect to this metric. The set $\widehat{\Sigma}^*$ forms a monoid under concatenation. Now, for $u, v \in \widehat{\Sigma}^*$. we say that $u = v$ is a *(pseudo)*-equation, and we say that M satisfies the equation if there is a pair of sequences $u_1, u_2, \ldots$ and $v_1, v_2 \ldots$ converging to u and v such that M satisfies the equations $u_i = v_i$ in the limit. Monoid varieties can now be characterized as follows:

Theorem 2.10 *([19]) Every monoid variety can be defined by a set of pseudoequations.*

A similar characterization will hold in the case of semigroup varieties.

It is difficult to obtain a clear mental picture of an arbitrary element of $\widehat{\Sigma}^*$ from the definition. Fortunately we can already characterize most monoid varieties of interest by looking at a simple subclass of $\widehat{\Sigma}^*$. We consider the closure of finite strings in Σ^* under concatenation and the ω operator, which we define below.

Definition 2.3 *For $x \in \widehat{\Sigma}^*$, we denote by x^ω the limit* $\lim_{k\rightarrow\infty} x^{k!}$.

If x is a monoid element, then the limit x^ω is the idempotent element in the semigroup generated by x. This allows us to specify varieties equationally in terms of identities satisfied by the idempotents. For instance, the equation $x^\omega y = yx^\omega$

expresses the algebraic condition that all idempotents commute with all elements of the monoid.

2.2.5 Important Varieties

The power of the algebraic method follows not only from the basic framework, but from the taxonomy of varieties that has developed over many years of investigation. Varieties that are defined in terms of natural algebraic properties are often interrelated in meaningful ways.

Two central varieties are **G**, which is the variety of groups, and **A**, which is the variety of aperiodic monoids. The significance of these two varieties is exhibited by the celebrated Krohn-Rhodes Theorem [40], which states that all finite monoids divide an iterated semidirect product of groups and aperiodic monoids. In this sense, groups and aperiodic monoids are the building blocks of finite monoids.

In this section we highlight several important varieties of monoids, with particular focus to those which arise from our work. In many cases, there exists a combinatorial description of the class of languages recognized by these varieties.

Varieties of Groups

A *group* is a monoid for which all elements m there is an element m^{-1} satisfying $mm^{-1} = m^{-1}m = 1$. The variety **G** of all finite groups can be characterized by the equation $x^\omega y = yx^\omega = y$. To see this, on one hand idempotents within groups must act as the identity since $ee = e$ implies that $e = eee^{-1} = ee^{-1} = 1$. On the other hand, substituting y for 1 in the equation we get $x^\omega = 1$. Thus for any x there exists a suitable power k of x such that $x^k = 1$, and so $x^{k-1} = x^{-1}$.

The class of languages recognized by groups appears to be too complex to admit a useful combinatorial characterization. However, several important subvarieties of $\mathbf{G}$ have been characterized. A simple example is the variety $\mathbf{Ab}$ of *Abelian*, or commutative groups. Using the fact that any finite abelian group can be decomposed into a direct sum of cyclic groups, it can be shown that a language L is recognized by some finite abelian group if and only if there exists an m such that membership in L depends only on the number of occurrences of each letter modulo m.

There are also some nontrivial characterizations of subvarieties of $\mathbf{G}$, such as the class $\mathbf{G_{nil}}$ of *nilpotent* groups. For two subgroups H_1, H_2, let $[H_1\,,\,H_2]$ be the subgroup generated by group elements of the form $h_1h_2h_1^{-1}h_2^{-1}$, with $h_i \in H_i$. Let $G = G_0$. Then G is nilpotent if the series $G_1 = [G\,,\,G_0]$, $G_2 = [G\,,\,G_1]$,... ends in the trivial group. For $w, v \in \Sigma^*$, let $\binom{w}{v}$ be the number of distinct occurrences of v in w. A language L is recognized by a group in $\mathbf{G_{nil}}$ if and only if there exists positive integers m, k such that membership of w in L depends only on the value of $\binom{w}{v}$ mod m for all $v \in \Sigma^*$, $|v| \leq k$.

Varieties of Aperiodics

A monoid is called *aperiodic* if it does not contain a submonoid which forms a nontrivial group. The variety $\mathbf{A}$ of aperiodic monoids is characterized by the equation $x^\omega x = x^\omega$. To see this, note that if M is not aperiodic there will exist an x which is part of a group in M but is not the identity. But then x^ω must be the identity of the group, and so $x^\omega x = x \neq x^\omega$ and the equation is violated. On the other hand, if there is an x such that $x^\omega x \neq x^\omega$, then the set of elements generated by $x^\omega x$ form a nontrivial group with identity x^ω.

The class of languages recognized by aperiodic monoids was famously characterized by Schützenberger [57]. They correspond exactly to the "star-free" languages, which are those languages which are in the closure of finite languages under concatenation and boolean operations, but not star. This class of languages arises naturally in different contexts. For instance, it is equal to the class of languages which can be described by a formula in first-order logic over order [42].

Note that this class can include languages which are more simply expressed using a Kleene star. For example, the language L of words containing at least one a is star-free, since $L = \Sigma^* a \Sigma^* = \overline{\emptyset} a \overline{\emptyset}$.

The class **Nil** is a variety of semigroups which consists of only one idempotent which acts as a zero in the semigroup. This is naturally characterized by the equations $x^\omega y = x^\omega = y x^\omega$. It is not hard to show that the languages recognized by nilpotent monoids are exactly the languages $L \subseteq \Sigma^+$ such that either L or $\overline{L}$ is finite.

Several important aperiodic varieties arise from the definition of Green's relations. We say that a monoid M is *$\mathcal{J}$-trivial* (respectively $\mathcal{R}$, $\mathcal{L}$, or $\mathcal{H}$-trivial) if each of the $\mathcal{J}$-classes of M are singletons.

The class of $\mathcal{H}$-trivial monoids is exactly the variety $\mathbf{A}$. The variety $\mathbf{R}$ of $\mathcal{R}$-trivial monoids can be characterized by the equation $(xy)^\omega x = (xy)^\omega$. Similarly the variety $\mathbf{L}$ of $\mathcal{L}$-trivial monoids is characterized by the equation $y(xy)^\omega = (xy)^\omega$. The variety $\mathbf{J}$ of $\mathcal{J}$-trivial monoids satisfies $\mathbf{J} = \mathbf{L} \cap \mathbf{R}$, so $\mathbf{J}$ is characterized by the equations $(xy)^\omega x = (xy)^\omega = y(xy)^\omega$.

A beautiful combinatorial characterization of $\mathbf{J}$ was obtained by I. Simon [61]. We say that $a = a_1 \dots a_k$ is a *subword* of w if $w \in \Sigma^* a_1 \Sigma^* \dots a_k \Sigma^*$. We will call the language $\Sigma^* a_1 \Sigma^* \ldots a_k \Sigma^*$ a *subword test.* Let $\mathcal{J}$ be the language variety such that $\mathbf{J} \to \mathcal{J}$.

Theorem 2.11 *(Simon's Theorem) A regular language L is in $\mathcal{J}$ if and only if it is a finite boolean combination of languages of the form:*

$$\Sigma^* a_1 \Sigma^* \ldots a_k \Sigma^*.$$

A language is recognized by an $\mathcal{R}$-trivial monoid if and only if it is a boolean combination of languages of the form $\Sigma_0^* a_1 \Sigma_1^* \ldots a_k \Sigma_k^*$, where $a_i \notin \Sigma_{i-1}$ for all i. The $\mathcal{L}$-trivial monoids can recognize those language that are reversals of these languages.

2.2.6 Operations on Varieties

Several of the monoid varieties discussed in this thesis exhibit useful structural decompositions. We have mentioned for example that all finite monoids divide an iterated semidirect product of monoids in $\mathbf{G}$ and monoids in $\mathbf{J}_1$. These decomposition results can often give us a practical advantage, for example they may help to generate equations for a variety from the equations of a simpler variety, or to take advantage of powerful combinatorial results.

In order to formally introduce some of these decomposition results, we must first introduce some natural operations on varieties.

Wreath Product

Let $(M, +)$ and $(N, \cdot)$ be two monoids with identities 0 and 1 respectively. A *left action* $\cdot_L : N \times M \to M$ of a semigroup N on M is a function, denoted multiplicatively, that satisfies $n' \cdot_L (n \cdot_L m) = n'n \cdot_L m$. We say that $\cdot_L$ is *unitary* if $0 \cdot_L m = m$ and $n \cdot_L 1 = 1$.

A *semidirect product* of monoids M and N is a monoid over the ground set $M \times N$ with the product $(m,n)(m',n') = (m + n \cdot_L m', nn')$, with respect to some unitary left action $\cdot_L$. This is indeed an associative operation and $(0,1)$ is the identity. It is a partial direct product in the sense that the projection $\pi : M \times N \to N$ is a surjective morphism under this product. Semidirect products of semigroups are defined in a similar way, but without the condition of unitarity.

The wreath product $\mathbf{V} * \mathbf{W}$ of varieties $\mathbf{V}$ and $\mathbf{W}$ is the variety generated by the set of semidirect products of some $V \in \mathbf{V}$ and some $W \in \mathbf{W}$. Let $\mathbf{V} \to \mathcal{V}$ and $\mathbf{W} \to \mathcal{W}$. The class of languages recognized by monoids in $\mathbf{V} * \mathbf{W}$ has a useful combinatorial characterization in terms of $\mathcal{V}$, $\mathcal{W}$, and finite state transducers. This characterization is known as the *wreath product principle*.

Fix an alphabet Σ. Let $\varphi : \Sigma^* \to N$ be a morphism and let $\Gamma = N \times \Sigma$. Also, let $\sigma_\varphi : \Sigma^* \to \Gamma^*$ be the function defined by:

$$\sigma_\varphi(a_1 \dots a_k) = (1, a_1)(\varphi(a_1), a_2), \dots (\varphi(a_1 \dots a_{k-1}), a_k).$$

We call σ_φ the *sequential function associated with* φ. Such a sequential function can be computed by a finite state transducer whose transition monoid divides N.

Theorem 2.12 *(Wreath Product Principle) [62] If L is recognized by a monoid $M \in \mathbf{V} * \mathbf{W}$, then it is a finite boolean combination of languages of the form $X \cap \sigma_\varphi^{-1}(Y)$, for some alphabet Σ, monoid $W \in \mathbf{W}$, morphism $\varphi : \Sigma^* \to W$, and languages $X \subseteq \Sigma^*$, $Y \subseteq (W \times \Sigma)^*$ such that $Y \in \mathcal{V}$, $X \in \mathcal{W}$.*

Malcev Product

A *relational morphism of monoids* from M to N is a function $\varphi : M \to 2^N$ that satisfies $1_N \in \varphi(1)$, $\varphi(m_1)\varphi(m_2) \subseteq \varphi(m_1 m_2)$, and $\varphi(m)$ is nonempty for all $m \in M$. A relational morphism of semigroups is the same except that the $1_N \in \varphi(1)$ condition is not required. The *graph* of φ, denoted *graph*(φ), is the submonoid of $M \times N$ formed by the set $\{(m, n) \in M \times N : n \in \varphi(m)\}$. Let $\pi_1 : graph(\varphi) \to M$ and $\pi_2 : graph(\varphi) \to N$ be the projections on to the first and second coordinates respectively. Then $\varphi = \pi_2\pi_1^{-1}$. Note that π_1 and π_2 will be morphism, and π_1 will be surjective. We say that φ is *injective* if π_2 is injective. Injective relational morphisms have the property $(\varphi(m_1) \cap \varphi(m_2) = \emptyset$ for $m_1 \neq m_2)$. It can be shown that $M \preceq N$ if and only if there is an injective relational morphism from M to N.

The *Malcev product* $\mathbf{V} ⓜ \mathbf{W}$ of $\mathbf{V}$ and $\mathbf{W}$ is the set of all monoids M for which there exists a $W \in \mathbf{W}$ and a relational morphism $\varphi : M \to W$ such that for all idempotents $e \in W$ we have $\varphi^{-1}(e) \in \mathbf{V}$.

2.2.7 The Variety BG

The variety of *block groups* [50], denoted **BG**, is a natural and well-studied class of finite monoids which arises at several points in our investigation. A monoid M is a block group if every $\mathcal{L}$-class and every $\mathcal{R}$-class of M each contain at most one idempotent.

With this formulation, it is not hard to see that the **BG** is the largest monoid variety whose corresponding language variety does not contain the languages $a\Sigma^*$ or Σ^*a. On one hand, the syntactic monoids of $a\Sigma^*$ and Σ^*a respectively have two $\mathcal{R}$-related idempotents and two $\mathcal{L}$-related idempotents. On the other hand, if L is such that $M(L)$ has two $\mathcal{R}$-related idempotents x and y, then 1, x, and y form a

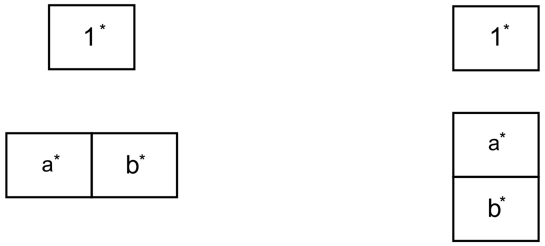


Figure 2–4: The syntactic monoids for Σ^*a and $a\Sigma^*$, respectively.

three-element submonoid which is isomorphic to the syntactic monoid of $a\Sigma^*$. A symmetric argument holds for Σ^*a when there are two $\mathcal{L}$-related idempotents.

The variety **BG** has several equivalent formulations. For instance, it is equivalent to:

- the variety $\mathbf{J} * \mathbf{G}$,
- the variety $\mathbf{J} \textcircled{M} \mathbf{G}$,
- the variety **EJ** of monoids M such that the set $E(M)$ of idempotents of M generates a **J**-trivial subgroup, and
- the variety **PG** of *power groups*, which are monoids formed by taking the powerset of a finite group G as monoid elements with product as the monoid operation.

These equivalence results give us several useful combinatorial characterizations of the variety **BG**. For example, a language L is recognized by a monoid in **PG** (and thus **BG**) if and only if it is a boolean combination of languages of the form $L_0a_1L_1 \ldots a_kL_k$, where each L_i is a language that is recognized by a group.

The variety **BG** can be characterized equationally as the class of all finite monoids which satisfy the equation:

$$(x^\omega y^\omega)^\omega = (y^\omega x^\omega)^\omega,$$

or equivalently those which satisfy the equation:

$$(x^\omega y^\omega)^\omega x^\omega = (x^\omega y^\omega)^\omega = y^\omega (x^\omega y^\omega)^\omega.$$

CHAPTER 3
Characterizations of QFA

The objective of this thesis is to understand the language recognition power of different models of QFA. An ideal but sometimes elusive goal in such an undertaking is to make broad statements about which languages can or cannot be recognized. In this chapter we apply algebraic automata theory to obtain characterizations of the class of languages recognized by QFA. Along the way, we introduce techniques for constructing QFAs, as well as techniques for showing nonrecognizability.

In Section 3.1.3 we investigate the extent to which the five models are closed under variety operations. In fact the language classes corresponding to MOQFA and LQFA satisfy all of the variety closure properties. Thus by Eilenberg's theorem it is immediate that there exists *some* algebraic characterization of these language classes: recognizability of a given language L by MOQFA or LQFA depends solely on algebraic properties of the syntactic monoid of L. In Sections 3.2 and 3.3 we obtain these characterizations. We also show in Section 3.4 that the boolean closure of languages recognized by BPQFA can also be characterized algebraically. For the case of MOQFA we find that this type of QFA can recognize exactly those languages whose syntactic monoids are groups.

Note that such an algebraic characterization requires both a recognizability result showing that each L with syntactic monoid as group can be recognized, and a matching proof that any L whose syntactic monoid is not a group cannot be recognized. To show a recognizability result on a variety of monoids, we can take advantage of combinatorial characterizations of the languages in the associated language varieties. The impossibility result is also assisted by the algebraic setting.

For LQFA, we show that these machines can recognize exactly those languages whose syntactic monoids are in the variety called **BG**, which we describe in Section 2.2.7. Known results about **BG** give us a nontrivial combinatorial characterization of these languages, which is essential for establishing the recognizability result. Basic facts about **BG** reduce the work in the impossibility part to proving the nonrecognizability of two canonical languages $a\Sigma^*$ and $\Sigma^* a$. This demonstrates the power of the algebraic perspective.

We find that the remaining models are closed under many but not all of the variety operations. However, we can still apply algebraic techniques to permit some partial characterization. In Section 3.4 we show that the boolean closure of languages recognized by BPQFA is exactly the language class corresponding to the monoid variety **BG**. In later chapters we will investigate generalizations of Eilenberg's framework to handle language classes which do not meet all of the variety closure properties.

3.1 Preliminaries

3.1.1 Criteria for Language Recognition

As the QFA models discussed here have a probabilistic output, we should specify the criteria for language recognition. We make note here of the distinction between bounded-error and unbounded error recognition. For bounded-error acceptance, we require that there is a lower bound p on the probability of accepting $w \in L$ and an upper bound p' on the probability of accepting $w \notin L$, such that p' is *strictly* less than p. Such a QFA can be easily converted into a QFA which gives the correct answer with probability $p > \frac{1}{2}$.

The weaker condition of unbounded-error acceptance would require that any word $w \in L$ is accepted with probability strictly greater than some upper bound p on the probability that any $w \notin L$ is accepted. Under this criterion, it is possible to construct QFAs that recognize nonregular languages. For example, consider a MOQFA over $\Sigma = \{a, b\}$ with $Q = \{q_0, q_1\}$, $A_\$ = U_{¢} = I$, initial state $|q_0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and

$$A_a = A_b^{-1} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix},$$

where α is an irrational multiple of π. Then $|\psi_w\rangle$ has nonzero amplitude in the second coordinate if and only if $|w|_a \neq |w|_b$. Thus by defining the accepting subspace to be *span*$\{|q_1\rangle\}$, M recognizes $L = \{w : |w|_a \neq |w_b|\}$ with unbounded error.

Unbounded-error recognition is considered impractical as there is no a priori way to build confidence in the answer given by the machine. In the bounded error case, where the correct answer is given with probability $p > 1/2$, it is well known (c.f. [47]) that we can build our confidence by running the algorithm several times on the same input and then taking the majority vote. A good bound on the minimum number of trials required for a desired accuracy is given by the Chernoff bound, given below. This is a special case of the Chernoff inequality [21]:

Theorem 3.1 *(The Chernoff Bound) Let $X_1, \ldots, X_n$ be i.i.d. random variables which take value 1 with probability $1/2 + \epsilon$ and value 0 otherwise. Then:*

$$\mathbf{P}\left[\sum X_i \leq n/2\right] \leq e^{-2\epsilon^2 n}.$$

Supposing we had a machine which, on any input, would produce the correct output with some fixed probability $p > \frac{1}{2}$. We run this machine on the same input n times, taking the X_i to be the outcome of the machine on the ith run of the machine. The Chernoff bound states that the probability that the machine is incorrect on a majority of these trials diminishes exponentially in n.

For MOQFA and LQFA, this simulation can be performed within the model itself. Specifically for a machine M and $n > 0$ we can construct M' that accepts w with probability equal to the probability that the majority of n trials of M on input w accept. For the MOQFA case we do this on a space with state set Q^n. The initial state is set to $\bigotimes_{i-1}^{n} |q_0\rangle$ and on input σ we apply the operation $\bigotimes_{i=1}^{n} U_\sigma$. This has the effect of simulating n copies of M in parallel. Let $[n] = \{1, \ldots, n\}$, and let S_{acc} and S_{rej} be the accepting and rejecting subspaces for one copy of M. Then we define:

$$S'_{acc} = \sum_{I \subseteq [n],\, |I| > n/2} \bigotimes_{i=1}^{n} S_{i,I},$$

where $S_{i,I} = S_{acc}$ if $i \in I$ and $S_{i,I} = S_{rej}$ otherwise. Lastly we define $S'_{rej} = \overline{S'_{acc}}$. M' will accept an input w with probability equal to the probability that a majority of the copies of M accept. The construction for LQFA is essentially the same. On the other hand, it is not possible in general to boost the probability of recognition for GQFA, KWQFA, or BPQFA, in fact for each of these models there are languages which QFAs can recognized with some probability $p > \frac{1}{2}$. but not with probability $1 - \varepsilon$ for all $\varepsilon > 0$.

3.1.2 Abstract State Descriptions

In Section 2.1.2, we described how density matrices can be used a succinct description of a mixed state, from which we can calculate the outcome of any sequence of measurement and evolution operations that are to follow. This is true despite the fact that the state is a random variable taken from an unbounded number of possible values.

For all of the QFA models that we discuss here, the processing of some input word w involves a number of probabilistic choices. The state of a MOQFA and LQFA behavior in an intermediate stage of processing can be suitably described using density matrices. It is useful to have a similar formalism to describe the states of the other machines.

For BPQFA and KWQFA, after reading some partial input $w = w_1 \dots w_k$ there is some probability p_{non} that the machine did not halt while w is being processed. Recall that a measurement is made after every input letter is read, and the machine will halt if the outcome of the measurement is acc or rej. Conditioned on machine M not halting while processing u, we know the outcome of every measurement and so the state of M is determined. This motivates the following description of the nonhalting part. For $\sigma \in \Sigma \cup \{¢, \$\}$, define A'_σ is the unitary operator $P_{non}A_\sigma$, where P_{non} is the projection into the nonhalting subspace. By induction, it is easy to show that the vector $|\psi_w\rangle = A'_{w_k} \dots A'_{w_1} A'_{¢} |q_0\rangle$ is the state of M conditioned on not halting, scaled by a factor of $\sqrt{p_{non}}$. Notice that for BPQFA, conditioned on M halting, M has rejected the input. Thus the vector $|\psi_w\rangle$ completely characterizes the behaviour of M on reading w over all probabilistic choices. For KWQFA, we must additionally keep track of the probability with which M has accepted or rejected while reading w, so the abstract state description is naturally a triple $(p_{acc}, p_{rej}, |\psi_w\rangle)$. Finally for GQFA, we use a triple $(p_{acc}, p_{rej}, \rho_w)$, where ρ_w is the density matrix corresponding to the mixed state of M, conditioned on not halting.

3.1.3 Closure Properties

All five models discussed in the thesis are closed under inverse morphisms. Suppose that $L \subseteq \Gamma^*$ is recognized by QFA M and $\varphi : \Sigma^* \to \Gamma^*$ is a morphism. Intuitively, we construct M' recognizing $\varphi^{-1}(L)$ by simulating, on input $\sigma \in \Sigma$, the behavior of M on the word $\varphi(\sigma) = \gamma_1 \ldots \gamma_k$. Thus on reading $w_1 \ldots w_n \in \Sigma^*$, the state of M' is equal to the state of M on reading $\varphi(w_1 \ldots w_n)$. Finally, M' accepts or rejects as M would, and so M' recognizes $\varphi^{-1}(L)$ with the same error bound for which M recognizes L. So it remains to show that the operations for a series of letters can be composed into the operation for a single letter. This is immediate in the case of MOQFAs, since each letter induces a unitary transformation in this case, and unitary matrices are closed under multiplication. The remaining four cases require an argument, since each letter induces a unitary transformation followed by a measurement and it is not clear that a series of such operations can be composed into one unitary operation followed by one measurement. For KWQFAs and BPQFAs, this can be done by adding extra halting states. To simulate a series of k transformations and measurements, we create k copies of each halting state. We send the halting part of the ith measurement into the ith copy of the halting state. This can be done with a single unitary matrix. For LQFAs we have the following theorem:

Claim 1 *([3]) Consider a sequence of l transformations and measurements operating on a finite space E. These operations can be simulated by one transformation and one measurement on a (possibly larger) finite space E'.*

The proof idea is based on *the principle of deferred measurement* [46]. Let $M = \{P_1, \ldots, P_k\}$ be a projective measurement and let U be unitary operator applied to some space A. We consider the operation M followed by U. We can simulate this operation on the space $A \otimes B$, where the B has dimension k. Define U_M to be an operator on A and B that satisfies:

$$U_M|\psi\rangle_A|0\rangle_B = \sum_i (P_i|\psi\rangle_A)|i\rangle_B.$$

This operation can be extended to a unitary matrix. Recalling the definition of the partial trace, if we apply U_M followed by U, the A subspace behaves exactly as if M and then U were applied to A. We can simulate the partial trace by just measuring with respect to $\{I_B \times |i\rangle\langle i|_B\}$, so if we make this measurement after the operation $U_M U$, we have the desired output in the subspace A. To complete the proof, we would need to show that subspace B can be reused without being initialized again.

All of the models we discuss are also closed under left and right quotient. Suppose M recognizes $w^{-1}L$. From M we can construct M' recognizing $w^{-1}L$ by initializing M' to the state of M on reading w. This initialization can be performed in all of the models, using the ideas from the inverse morphic closure. Likewise, to recognize Lw^{-1} we simulate M on L and then apply the operations for w.

Last, we cover the boolean operations. For the case of MOQFA, KWQFA, LQFA, and GQFA, from a QFA recognizing L we can construct an QFA for $\overline{L}$ just by swapping the accept and reject states. For the case of BPQFA we do not have this symmetry in the definition, and in fact we show in Chapter 4 that BPQFA are not closed under complement.

3.2 Characterization of MOQFA

In this section we present a theorem of [44] which characterizes the class of languages recognized by MOQFA. The theorem can be stated nicely in algebraic terms.

Theorem 3.2 *The language L is recognized by an MOQFA iff its syntactic monoid is in* $\mathbf{G}$.

In Section 3.1.3 it is shown that the class of languages recognized by MOQFA forms a variety. We need to show that the variety of monoids corresponding to MOQFA is exactly $\mathbf{G}$. For the upper bound it suffices to show:

Theorem 3.3 *Let L be a language such that $M(L) \in G$. Then there is an MOQFA recognizing L with probability of recognition 1.*

Note that all of the models in this paper include MOQFA as a special case, so this construction applies to all of the QFA models in the thesis.

Proof: First, suppose $M(L) \in \mathbf{G}$. Then there is a finite group G, a set $F \subseteq G$ and a homomorphism $\varphi : \Sigma^* \to G$ such that $\varphi^{-1}(F) = L$. We construct a MOQFA $M = (Q, q_0, \Sigma, \{U_\sigma\}, Q_{acc})$ that recognizes L by directly computing the value of $\varphi(w)$ on input w. We define $Q = G$, q_0 to be the identity of G, and $Q_{acc} = F$. Also define $U_{¢} = U_{\$} = I$, and for each $t \in \Sigma$ we define U_t to be the unique linear operator such that for all $s \in G$ we have $U_t|s\rangle = |s \cdot \varphi(t)\rangle$. This is just a permutation of the basis vectors and so each U_t is unitary. Observe that $U_tU_s|1\rangle = |st\rangle$. Thus by induction:

$$U_{w_k} \cdots U_{w_1}|1\rangle \quad = \quad |\varphi(w_1) \cdots \varphi(w_k)\rangle = |\varphi(w_1 \ldots w_k)\rangle$$

So the state of the machine upon reading w is $|\varphi(w_1 \ldots w_k)\rangle$, and by definition $\varphi(w_1, \ldots, w_k) \in F$ if and only if $w \in L$. □

To show that L cannot be recognized by MOQFA unless $M(L)$ is a group, it is sufficient to show that MOQFA cannot recognize $\Sigma^* a \Sigma^*$. To see this, suppose for example that there is an MOQFA that recognizes L such that $M(L)$ is not a group. Then there is some element x of $M(L)$ which does not have an inverse. Observe that 1 and x^ω form a two-element semilattice with multiplication $x^\omega x^\omega = 1 \cdot x^\omega = x^\omega \cdot 1 = x^\omega$. This is the monoid commonly denoted U_1, and it is easily seen to be isomorphic to the syntactic monoid of $\Sigma^* a \Sigma^*$. In particular, $U_1 \preceq M(L)$, so an MOQFA recognizing L would imply the existence of an MOQFA recognizing $\Sigma^* a \Sigma^*$.

The nonrecognizability of $\Sigma^* a \Sigma^*$ is based on the following important fact, which quantifies the intuitive notion that suitable large powers of a unitary matrix act almost as the identity.

Lemma 3.4 *For every unitary operator U on $\mathbb{C}^n$ and every $\varepsilon > 0$ there exists a k such that for all unit vectors $|\psi\rangle \in \mathbb{C}^n$ we have $|(|\psi\rangle, U^k|\psi\rangle)| \geq 1 - \varepsilon$.*

Proof: Unitary matrices satisfy $U^\dagger U = UU^\dagger$, and so they are normal. Recalling the spectral decomposition theorem, we can express U as $U = \sum_{i=1}^n \lambda_i |e_i\rangle\langle e_i|$, where each $|e_i\rangle$ is a normal eigenvector of U with $\langle e_i|e_j\rangle = 0$ for all $i \neq j$, and each λ_i is the eigenvalue corresponding to $|e_i\rangle$. Then:

$$\begin{aligned} U^2 &= (\sum_{i=1}^{n} \lambda_i |e_i\rangle\langle e_i|)(\sum_{j=1}^{n} \lambda_i |e_i\rangle\langle e_i|) \\ &= \sum_{i,j} \lambda_i\lambda_j |e_i\rangle\langle e_i|e_j\rangle\langle e_j| \\ &= \sum_i \lambda_i^2 |e_i\rangle\langle e_i|, \end{aligned}$$

and likewise, $U^k = \sum_i \lambda_i^k |e_i\rangle\langle e_i|$.

Recall that by definition of unitarity, we have $(|\varphi\rangle, |\psi\rangle) = (U|\varphi\rangle, U|\psi\rangle)$ for all $|\varphi\rangle$, $|\psi\rangle$. This implies that all of the eigenvalues λ_j of U satisfy $|\lambda_j| = 1$.

Since each λ_j lies on the unit circle, by Euler's formula it can be expressed uniquely as $e^{2\pi i\theta_j}$ for some $\theta_j \in [0, 1)$. When we take powers of the U, the eigenvalues cycle periodically around the unit circle. Let $\varepsilon' < \frac{\varepsilon}{n}$. By Theorem 201 of [34], for a suitable choice of k we have $|1 - \lambda_j^k| \leq \frac{\varepsilon}{n}$ for all j. Then, using the identity $\langle\psi|(\sum_i |e_i\rangle\langle e_i|)|\psi\rangle = \langle\psi|I|\psi\rangle = 1$, we obtain:

$$\begin{aligned} |(|\psi\rangle, U^k|\psi\rangle)| &= \left|1 - \sum_i (1 - \lambda_i^k)\langle\psi|e_i\rangle\langle e_i|\psi\rangle\right| \\ &\geq 1 - \sum_i |1 - \lambda_i^k| \\ &> 1 - \varepsilon, \end{aligned}$$

as desired. □

Theorem 3.5 *There is no MOQFA which recognizes the language $\Sigma^* a \Sigma^*$.*

Proof: Suppose MOQFA M recognizes $\Sigma^*a\Sigma^*$ with recognition probability $p > \frac{1}{2}$. Let A_a be the unitary matrix induced by the letter a. By Lemma 3.4, for any ε there is a k such that $\langle\psi|A_a^k|\psi\rangle \geq 1 - \varepsilon$ for all ψ. By a suitable choice of ε, we can ensure that $\||\psi\rangle - A_a|\psi\rangle\|_2^2 \leq \sqrt{2(p - \frac{1}{2})}$ for all $|\psi\rangle$, in particular for the vector reached after reading the preprocessing character ¢. Similarly to the argument of Theorem 2.3, the assumption of bounded error recognition implies that a^k and the empty string should both be accepted or both rejected. Either case will contradict the assumption that M is recognizing $\Sigma^*a\Sigma^*$ □.

3.3 Exact Characterizations of LQFA

We now prove the following algebraic characterization of LQFA:

Theorem 3.6 *A language L is recognized by an LQFA with bounded error if and only if $M(L) \in$* **BG**.

As discussed in the introduction to the chapter, the proof will be in two parts. In Section 3.3.1 we show that every language L such that $M(L) \in$ **BG** is recognized by LQFA. In Section 3.3.2, we show that LQFA cannot recognize the languages $a\Sigma^*$ or Σ^*a.

3.3.1 Recognizability Results for LQFA

The first step of the proof is to show that LQFA can recognize subword tests:

Theorem 3.7 *LQFA can recognize languages of the form $\Sigma^*a_1\Sigma^* \ldots a_k\Sigma^*$.*

Proof: In this proof, we write $u \in v$ for $u, v \in \Sigma^*$ if u occurs as a subword of v. We prove the result by induction on $a_1 \ldots a_k$. For the induction base, we construct an LQFA $M = (Q, \Sigma, q_0, \{A_\sigma\}, \{P_\sigma\}, F)$ for the language $\Sigma^*a\Sigma^*$. We set $Q = \{q_0, q_1, \ldots, q_{n-1}\}$, $A_{¢} = A_\$ = I$ and $P_{¢} = P_\$ = \{I\}$.

Let F_n be the Fourier transform over $\mathbb{Z}_n$. On input a, the transformation F_n is applied followed by measurement $M_a = \{|q_0\rangle\langle q_0|, \ldots, |q_{n-1}\rangle\langle q_{n-1}|\}$, and no action is performed for $\sigma \neq a$. The first a that is read will cause the machine to move to some state q_i uniformly at random. The second and subsequent applications will have the same effect. To see this, note that $F_n|q_0\rangle = \sum_j \sqrt{\frac{1}{n}}|q_j\rangle$ for the first a, and in general $F_n|q_i\rangle = \sum_j \frac{1}{\sqrt{n}}\chi_i(j)|q_j\rangle$, where χ_i is the character of i in the group $\mathbb{Z}_n$ and $|\chi_i(j)| = 1$. Thus after reading the input word, if we measure with respect to $\{P_{acc} = \sum_{i\neq 0}|i\rangle\langle i|, P_{rej} = |0\rangle\langle 0|\}$, we obtain the correct answer with probability at least $\left(\frac{n-1}{n}\right)$.

We now induct on this construction. The inductive hypothesis is that we have a machine $M^{(\ell)} = (Q^{(\ell)}, q_0, \Sigma, \{A_\sigma^{(\ell)}\}, \{P_\sigma^{(\ell)}\}, Q_{acc}^{(\ell)})$, such that the state of $M^{(\ell)}$ at any time is $|q\rangle\langle q|$ for some $q \in Q_{acc}^{(\ell)}$, and if w satisfies $a_1 \ldots a_\ell \in w$ then $q \in Q_{acc}^{(\ell)}$ with probability $\left(\frac{n-1}{n}\right)^i$, and $a_1 \ldots a_k \notin w$ implies $q \notin Q_{acc}^{(\ell)}$. Assume this is true for $\ell = i-1$. We augment the construction to make a machine for the case $\ell = i$.

Our augmentation will proceed as follows. First let $Q_{acc}^{(i)}$ be a set of $(n-1)^i$ new states, all of which are distinct from $Q^{(i-1)}$, and let $Q^{(i)} = Q^{(i-1)} \cup Q_{acc}^{(i)}$. For each $q \in Q_{acc}^{(i-1)}$ we uniquely associate $n-1$ states $q_2, \ldots, q_n \in Q_{acc}^{(i)}$. We leave q_0 unchanged.

It remains to define the $A_\sigma^{(i)}$ transitions. Define $\tilde{A}_\sigma^{(i-1)}$ (resp. $\tilde{P}_\sigma^{(i-1)}$) to be the transformation that acts as $A_\sigma^{(i-1)}$ on $Q^{(i-1)} \subset Q^{(i)}$ and as the identity elsewhere. Using Claim 1, we construct $P_\sigma^{(i)} A_\sigma^{(i)}$ so that they simulate the operation $\tilde{P}_\sigma^{(i-1)} \tilde{A}_\sigma^{(i-1)} B_\sigma^{(i)}$, where $B_\sigma^{(i)}$ is an additional operation (consisting of a unitary transformation and a measurement) which processes the a_i character. Note that the operations are applied from right to left. For all $\sigma \neq a_i$ we set $B_\sigma^{(i)}$ so as to perform no action. For $\sigma = a_i$, we define $B_\sigma^{(i)}$ so that, independently for each $q \in Q_{acc}^{(i-1)}$, the transformation F_n is applied to $Q_q = \{q, q_2, q_3, \ldots, q_n\}$, followed by the measurement $\{|q\rangle\langle q|, |q_2\rangle\langle q_2|, \ldots, |q_n\rangle\langle q_n|, \sum_{q' \notin Q_q} |q'\rangle\langle q'|\}$. At the end we have a machine $M = M^{(k)}$ that recognizes $\Sigma^* a_1 \Sigma^* \ldots a_k \Sigma^*$.

To simplify notation, we define $Q^{(0)} = Q_{acc}^{(0)} = \{q_0\}$ and $B_\sigma^{(1)} = A_\sigma^{(1)}$ for all σ. The correctness of the construction follows from this lemma:

Lemma 3.8 *Let w be any word. As we process w with M, for all $0 \leq i < k$ the total probability of M being in one of the states of $Q^{(i)}$ is nonincreasing.*

Proof: For any $S \subseteq Q$, denote by $P(S)$ the sum probability of being in one of the states of S. Every nontrivial A_σ matrix can be decomposed into a product of $B_{a_i}^{(i)}$ matrices operating on different parts of the state space. All of these matrices operate on the machine state in such a way that for all j and for any $\{q, q'\} \subseteq Q_{acc}^{(j)}$, at any time there is an equal probability of being in state q or q'. Thus the distribution of the state at any time can be completely specified by $P(Q_{acc}^{(0)}), \ldots, P(Q_{acc}^{(k)})$.

For all $0 \leq i < k$ the machine can only move from $Q^{(i)}$ to $Q \backslash Q^{(i)}$ when $B_{a_{i+1}}^{(i+1)}$ is applied, and this matrix has the effect of averaging the likelihood of being in any given state of $Q_{acc}^{(i)} \cup Q_{acc}^{(i+1)}$. Since $|Q_{acc}^{(i+1)}| = (n-1)|Q_{acc}^{(i)}|$, it follows that a $B_{a_{i+1}}^{(i+1)}$ operation will not increase $P(Q^{(i)})$ unless $P(Q_{acc}^{(i+1)}) > (n-1)P(Q_{acc}^{(i)})$. It can easily be shown by induction on the sequence of $B_{a_j}^{(j)}$ matrices forming the transitions of M that this condition is never satisfied. Thus $P(Q^{(i)})$ is nonincreasing for all i. □

We are now ready to prove that M recognizes $L = \Sigma^* a_1 \Sigma^* \cdots a_k \Sigma^*$. First we show that any $w \notin L$ is rejected with certainty. The transitions are constructed in such a way that M can only move from $Q^{(i-1)}$ to $Q^{(i)}$ upon reading a_i, and M cannot move from $Q^{(i-1)}$ to $Q^{(i+1)}$ in one step (even if $a_i = a_{i+1}$). Next we show that any $w \in L$ is accepted with probability $\left(\frac{n-1}{n}\right)^k$. After reading the first a_1, $P(Q^{(1)}_{acc}) \geq \left(\frac{n-1}{n}\right)$ and by Lemma 3.8 this remains satisfied until a_2 is read, at which point M satisfies $P(Q^{(2)}_{acc}) \geq \left(\frac{n-1}{n}\right)^2$. Inductively after reading subword a, M satisfies $P(Q_{acc}) \geq \left(\frac{n-1}{n}\right)^k$. Thus M recognizes $\Sigma^* a_1 \Sigma^* \ldots a_k \Sigma^*$ with probability $\left(\frac{n-1}{n}\right)^k$. □

We now return to the proof of Theorem 3.6. By Theorem 2.11, it follows that LQFA can recognize any language whose syntactic monoid is in $\mathbf{J}$. We can extend this construction to cover all languages whose syntactic monoid is in $\mathbf{J} * \mathbf{G}$ with the theorem below.

Lemma 3.9 *If LQFA can recognize every language whose syntactic monoid is in the variety* $\mathbf{V}$*, then they can recognize every language whose syntactic monoid is in the variety* $\mathbf{V} * \mathbf{G}$*.*

Proof: Recalling the wreath product principle in Section 2.2.6, It is sufficient to show that an arbitrary language of the form the $L = X \cap \sigma_\varphi^{-1}(Y)$, for arbitrary X such that $M(X) \in \mathbf{G}$, arbitrary Y such that $M(Y) \in \mathbf{V}$, and morphism $\varphi : \Sigma^* \rightarrow G$ for $G \in \mathbf{G}$, can be recognized by an LQFA.

Clearly X is recognized by an LQFA. By the assumption, there is an LQFA $M = (Q, q_0, G \times \Sigma, \{A_{g \times \sigma}\}, F)$ that recognizes Y. We will construct a machine $M' = (Q', q'_0, \Sigma, \{A_\sigma\}, F')$ that applies the transduction $\sigma_\varphi : \Sigma^* \rightarrow (G \times \Sigma)^*$ and sends this input into M. We begin by defining $Q' = G \times Q$ and $q'_0 = (1, q_0)$. For $\sigma \in \Sigma$, the transformation A_σ is applied in two steps. First, for each $g \in G$ the operation $A_{g \times \sigma}$ is applied to the space $g \times Q$. Second, the basis states are permuted by $P_{\varphi(\sigma)}$ defined by $P_{\varphi(\sigma)}|g, q\rangle = |g\varphi(\sigma), q\rangle$. This is indeed a permutation since φ is a group morphism. We set $F' = G \times F$.

Initially, all of the amplitude is in the subspace corresponding to $\{1\} \times Q$. Consider the behaviour of M after reading partial input $a_1, \dots a_{i-1}$. All of the machine's amplitude is in the subspace corresponding to the states $\varphi(a_1 \dots a_{i-1}) \times Q$. Thus on input a_i the operation corresponding to $(\varphi(a_1 \dots a_{i-1}), a_i)$ is applied to M, so M' recognizes $\sigma_\varphi^{-1}(Y)$ as required. □

Thus, LQFA can recognize every language whose syntactic monoid is in $\mathbf{J} * \mathbf{G}$. Noting the fact that $\mathbf{J} * \mathbf{G} = \mathbf{BG}$ completes the theorem. □

3.3.2 Impossibility Results

In this section we show that LQFA cannot recognize $a\Sigma^*$ or $\Sigma^* a$. Both proofs depend on the properties of Von Neumann entropy, outlined in Section 2.1.2. The first result is implied by the results of Nayak.

Theorem 3.10 *[45] There is no LQFA that recognizes the language $\Sigma^* a$ with probability $p > \frac{1}{2}$.*

Proof Sketch: This proof was originally given for GQFAs, of which LQFAs are a special case. The proof idea is to show that the dynamics of an n-state LQFA recognizing $\Sigma^* a$ would imply the existence of series of mixed states over the working space of the machine whose entropy grows unbounded, contradicting the maximal entropy bound of $\log n$. Recall that we denote by $I(X : Y)$ the mutual information between two random variables X and Y. The entropy lower bounds are implied by the famous Holevo theorem:

Theorem 3.11 *(Holevo Theorem [35]) Let X be a random variable taking value x with probability p_x, let $\{\rho_x\}_{x \in X}$ be a set of density matrices, and let $\rho = \sum_x \rho_x$. Then for any measurement of ρ, if Y is the classical random variable corresponding to the outcome of the measurement, then:*

$$I(X : Y) \leq S(\rho) - \sum_x p_x S(\rho_x).$$

This theorem has many immediate consequences regarding the relation between classical information and quantum information, for instance the fact that no more than n classical bits of information can be stored accurately in n quantum bits. In this particular case, it implies the following:

Theorem 3.12 *If ρ_0 and ρ_1 are two density matrices which can be distinguished correctly by a measurement with probability p, then for $\rho = \frac{1}{2}(\rho_0 + \rho_1)$ we have:*

$$S(\rho) \geq \frac{1}{2}(S(\rho_0) + S(\rho_1)) + 1 - H(p).$$

Suppose that X is a uniform boolean variable. Then $\rho_X = \frac{1}{2}(\rho_0 + \rho_1) = \rho$. Taking Y to be the outcome of a measurement on $\rho_X = \rho$, we see that $I(X : Y) \geq 1 - H(p)$. Applying this inequation with the Holevo theorem gives the claimed inequality. □

Suppose M is an LQFA recognizing $\Sigma^* a$ with probability $p > \frac{1}{2}$. Let ρ_w be the state of an LQFA on input w. The last step of the proof is to build a series of density matrices whose entropy grows unbounded. We start with the machine states ρ_a and ρ_b corresponding to reading the words a or b. Since M recognizes $\Sigma^* a$, it follows that ρ_a and ρ_b are distinguishable with probability at least p, so the state $\rho' = \frac{1}{2}(\rho_a + \rho_b)$ has entropy at least $1 - H(p)$, which is a positive constant for fixed $p > \frac{1}{2}$.

For finite L, let ρ_L correspond to the mixed state $\frac{1}{|L|}\sum_{w\in L}\rho_w$, and for $\sigma \in \{a,b\}$, let E_σ be the operation applied when reading σ. Then $\rho_{L\sigma} = \frac{1}{|L\sigma|}\sum_{w\in L\sigma}\rho_w = E_\sigma \rho_L$. Let $L_k = \{a,b\}^k$. Note that $\rho_{L_{k+1}} = \frac{1}{2}(\rho_{L_k a} + \rho_{L_k b})$. Words in $L_k a$ are distinguishable from words in $L_k b$ with probability p, and so $S(\rho_{L_{k+1}}) \geq S(\rho_{L_k}) + 1 - H(p)$. For sufficiently large k this becomes larger than $\log n$, a contradiction. Therefore, there can be no LQFA recognizing $\Sigma^* a$. □

The next theorem was proven in [3]:

Theorem 3.13 *There is no LQFA that recognizes the language $a\Sigma^*$ with probability $p > \frac{1}{2}$.*

Proof: Again in this case we will apply basic properties of Von Neumann entropy, but the proof of this theorem is considerably more involved than the previous. Suppose M is an LQFA that recognizes the $a\Sigma^*$ with probability $p > \frac{1}{2}$, with $\Sigma = \{a,b\}$. Let A and B be the operations corresponding to reading a and b respectively. Let $\varphi : \Sigma^* \to M(a\Sigma^*)$. Now $\varphi(a)$ is idempotent, so for all $i, j > 0$ if ρ_w is the state reached on reading w, the states $A^i\rho_w$ and $A^j\rho_w$ should both be accepted with probability at least p or rejected with probability at least p. The same statement holds also for B.

The proof involves constructing a normalized transformation A_{lim} which can replace A and still recognize $a\Sigma^*$ with probability at least p, but in addition A_{lim} will satisfy $A_{\text{lim}}A = AA_{\text{lim}} = A_{\text{lim}}$. We also construct a normalized B_{lim} with the same property. Note that $\varphi(ab)$ and $\varphi(ba)$ are also idempotent. Using A_{lim} and B_{lim}, normalized versions of the transformations AB and BA are constructed. The normalized transformations for AB and BA are called C_{lim} and D_{lim} respectively. The proof will be in two parts. First, we show that applying the final measurement to $C_{\text{lim}}\rho$ ($D_{\text{lim}}\rho$), where ρ is the initial state, causes the machine to accept (reject) with probability p. Next, we conclude by showing $C_{\text{lim}} = D_{\text{lim}}$, thus contradicting the assumption that M recognizes $a\Sigma^*$.

The proof uses the *trace distance*. The trace distance between matrices ρ_0 and ρ_1 is defined as $\|\rho_0 - \rho_1\|_t$, where $\|\rho\|_t = \|Tr(\sqrt{\rho\rho^\dagger})\|_t$ is the *trace norm* of ρ. The trace distance is a metric on the space of $n \times n$ density matrices. If ρ_0 and ρ_1 are such that $\delta = \|\rho_0 - \rho_1\|_t \geq 0$, then there is a measurement that distinguishes ρ_0 from ρ_1 with probability at least $\frac{1}{2} + \delta/4$. Furthermore, for any CPSO E we have $\|E\rho_0 - E\rho_1\|_t \leq \|\rho_0 - \rho_1\|_t$.

Let E be a CPSO. Define E' to be the operation that applies E with probability $\frac{1}{2}$, and applies the identity operation otherwise.

Lemma 3.14
1. *For any CPSO E such that $S(E\sigma) \geq S(\sigma)$ for all mixed states σ, we have that for any mixed state ρ the sequence $E'\rho$, $(E')^2\rho$, ..., $(E')^i\rho$, ... converges.*
2. *Let E_{lim} be the map $\rho \rightarrow lim_{i\rightarrow\infty}(E')^i\rho$. Then, E_{lim} is a CPSO and $S(E_{lim}\rho) \geq S(\rho)$ for any density matrix ρ.*

Proof: Let $\rho_i = (E')^i\rho$. The sequence $S(\rho_1), S(\rho_2), \ldots$ is nondecreasing and is bounded above by $\log n$, thus for fixed n it will converge to some fixed value s_{lim}. Now consider the sequence $\rho_1, \rho_2, \ldots$. The space of $n \times n$ density matrices forms a compact space with respect to the trace metric, and so the sequence must have some limit point ρ_{lim}, i.e. for all ε there exists an i such that $\|\rho_{lim} - \rho_i\|_t < \varepsilon$. By the continuity of S, $S(\rho_{lim}) = s_{lim}$.

It remains to show $E'\rho_{lim} = \rho_{lim}$. It is sufficient to show $E\rho_{lim} = \rho_{lim}$. Suppose instead that $\|\rho_{lim} - E\rho_{lim}\|_t = \delta > 0$. Let $\rho'_{lim} = \frac{1}{2}(\rho_{lim} + E\rho_{lim})$. By Theorem 3.12 and the fact that trace distance implies distinguishability, we have $S(\rho'_{lim}) > S(\rho_{lim}) + 1 - H(\frac{1}{2} + \frac{\delta}{4})$. We show that this implies that there exists i such that $S(\rho_i) > s_{lim}$, contradicting the fact that s_{lim} is an upper bound on $S(\rho_j)$ for all j. Let $\epsilon > 0$ be chosen such that $\epsilon \log_2 d - \epsilon \log_2 \epsilon < H(\frac{1}{2} + \frac{\delta}{4})$. Since ρ is a limit point, there exists i such that $\|\rho_{lim} - \rho_i\|_t \leq \epsilon$. CPSOs do not increase the trace distance, so $\|\rho'_{lim} - \rho_{i+1}\|_t \leq \epsilon$. By Lemma 2.2, $S(\rho_{i+1}) \geq S(\rho'_{lim}) - \epsilon \log_2 d + \epsilon \log_2 \epsilon > s_{lim}$, a contradiction.

To see the second part, notice that the limit of a sequence of linear maps on $d \times d$ matrices is a linear map on $d \times d$ matrices. Furthermore, if each map is trace-preserving and positive, the limit is trace preserving and positive. Finally, $S(E_{\mathrm{lim}}\rho) = s_{lim} \geq S(\rho_i)$. □

Let us take a moment to consider the meaning of the operator E_{lim}. Suppose that LQFA M recognizes some language L with bounded probability. and suppose that E is the transition performed within the LQFA whenever some letter σ is read. Let $\varphi : \Sigma^* \to M(L)$ be the syntactic morphism. If $\varphi(\sigma)$ is idempotent, then we can replace the operation E in M with E^k for any k and still recognize L correctly. In the same way, we can replace E with E_{lim} in M and recognize L correctly. The E_{lim} operation is a normalized version of E which has the nice property $E_{lim}E = E_{lim}E$.

Let M be a LQFA recognizing $a\Sigma^*$ with probability p, and let A and B be the operations performed when reading an a (b), We define $C = A_{\mathrm{lim}}B_{\mathrm{lim}}$, $D = B_{\mathrm{lim}}A_{\mathrm{lim}}$. Informally the operations C_{lim} and D_{lim} correspond to the transitions invoked on reading the profinite strings $(a^\omega b^\omega)^\omega$ and $(b^\omega a^\omega)^\omega$ respectively. Let ρ be the state after reading the start marker ¢, and let ρ_x be the state reached after reading x. We want C_{lim} (D_{lim}) to map ρ a state which is accepted (rejected) with probability p. Define Q_a (Q_b) to be the set of all probabilistic combinations of states ρ_{ax} (ρ_{bx}). Also, let $\overline{Q}_a$ and $\overline{Q}_b$ be the closures of Q_a and Q_b.

Lemma 3.15 *(Lemma 7 in [3]).* $C_{\mathrm{lim}}\rho \in \overline{Q}_a$ *and* $D_{\mathrm{lim}}\rho \in \overline{Q}_b$.

In particular, this lemma implies that if ρ is the initial state, the final measurement of M distinguishes $C_{\mathrm{lim}}\rho$ from $D_{\mathrm{lim}}\rho$ with probability p.

Proposition 3.16 *For a mixed state* ρ, $C_{lim}\rho = \rho$ *if and only if* $D_{lim}\rho = \rho$.

Proof: Suppose $C_{lim}\rho = \rho$. This implies $C\rho = \rho$. Otherwise, by Theorem 3.12, $S(C'\rho) > S(\rho)$ and, since $S((C')^i\rho) \geq S(C'\rho)$, we have $S(C_{lim}\rho) > S(\rho)$ and $C_{lim}\rho \neq \rho$. Likewise, since $C = A_{lim}B_{lim}$, this implies $B\rho = \rho$ and $A\rho = \rho$. The other direction is similar. □

We are now ready to show:

Lemma 3.17 $C_{lim} = D_{lim}$.

Proof: Note that $C_{lim}\rho = C_{lim}C_{lim}\rho$. By Proposition 3.16, This implies $C_{lim}\rho = D_{lim}C_{lim}\rho$. To show that $D_{lim} = C_{lim}$, take an arbitrary ρ and consider $\|C_{lim}\rho - D_{lim}\rho\|_t$. Let $\rho_{diff} = C_{lim}\rho - D_{lim}\rho$. We need to show that $Tr(\rho_{diff}) = 0$.

Note that $C_{lim}\rho_{diff} = \rho_{diff} = D_{lim}\rho_{diff}$. We can decompose ρ_{diff} as $\rho_{diff} = \rho_+ - \rho_-$, where ρ_+ and ρ_- are both positive. We show that $Tr(\rho_+) = 0$ and $Tr(\rho_-) = 0$. We will need the following proposition, which is easy to check:

Proposition 3.18 *Let* A *be an arbitrary CPSO. Assume that* ρ *is such that* $A\rho = \rho$. *Let* H *be the support of* ρ. *Then* $A(H) \subseteq H$.

Observe that C_{lim}, $D_{lim}\rho_+ = \rho_+$, and likewise for ρ_-. $Tr(\rho_{diff}) = Tr(\rho_+) - Tr(\rho_-)$. We can obtain the values $Tr(\rho_+)$ and $Tr(\rho_-)$ using the following proposition:

Proposition 3.19 *(Proposition 3 of [3]) Let E be a CPSO such that $S(E\rho) \geq S(\rho)$. Let H be such that $E(H) \subseteq H$. Then, for any ρ, $TrP_H\rho = TrP_HE\rho$*

In particular, note that $C_{lim}\rho_+ = \rho_+ = D_{lim}\rho_+$, and likewise for ρ_-. Let H_+ and H_- be the projection onto the support of ρ_+ and ρ_- respectively. Taking $E = C_{lim}$ or $E = D_{lim}$, we see that $Tr(P_{H_+}C_{lim}\rho) = Tr(P_{H_+}\rho = Tr(P_{H_+}D_{lim}\rho)$, and thus $Tr(\rho_+) = Tr(P_{H_+}\rho_{diff}) = Tr(P_{H_+}(C_{lim}\rho - D_{lim}\rho)) = 0$. Similarly $Tr(\rho_-) = 0$, and therefore $Tr(\rho_{diff}) = 0$ and therefore $C_{lim}\rho = D_{lim}\rho$. □

Now $C_{lim} = D_{lim}$, but the final measurement distinguishes $C_{lim}\rho$ from $D_{lim}\rho$ with probability p. This contradicts the assumption that M recognizes $a\Sigma^*$ with probability p. This completes the proof of Theorem 3.13. □

3.4 Characterization of Boolean Closure of BPQFA

In this section we present the following algebraic characterization of the boolean closure of BPQFA.

Theorem 3.20 *L can be recognized by a Boolean combination of BPQFA if and only if its syntactic monoid is in* **BG**.

The outline of the proof is similar to the case of LQFA. We start with the recognizability results.

3.4.1 Recognizability Results

Brodsky and Pippenger gave a BPQFA construction for subword tests. The key to their construction is what they call a trigger chain. We review this in Section 4.1.

Thus, boolean combinations of BPQFA can recognize every piecewise testable language, i.e. every language whose syntactic monoid is in $\mathbf{J}$. We can extend this to the following:

Theorem 3.21 *Boolean combinations of BPQFA can recognize every language whose syntactic monoid is in* $\mathbf{J} * \mathbf{G}$.

Proof: Recalling the wreath product principle in Section 2.2.6, It is sufficient to show that an arbitrary language of the form the $L = X \cap \sigma_\varphi^{-1}(Y)$, where X is a language such that $M(X) \in \mathbf{G}$, Y is a language such that $M(Y) \in \mathbf{J}$. We can use the MOQFA construction to recognize X, so it is sufficient that we can recognize $\sigma_\varphi^{-1}(Y)$. Since $M(Y) \in \mathbf{J}$, $Y = \cup_i \cap_j Y_{ij}$ where Y_{ij} is a subword test or the complement of a subword test. Then $\sigma_\varphi^{-1}(Y) = \sigma_\varphi^{-1}(\bigcup_i \bigcap_j Y_{ij}) = \bigcup_i \bigcap_j \sigma_\varphi^{-1}(Y_{ij})$. It is now sufficient to show that BPQFA can recognize an arbitrary language of the form $\sigma_\varphi^{-1}(Y_{ij})$. If Y_{ij} is a subword test, then we can compute $\sigma_\varphi^{-1}(Y_{ij})$ with the same strategy as in LQFA for Theorem 3.9. Otherwise, we do the same on $\sigma_\varphi^{-1}(\overline{Y_{ij}})$ and use the fact that $\sigma_\varphi^{-1}(Y_{ij}) = \overline{\sigma_\varphi^{-1}(\overline{Y_{ij}})}$ and the closure under complement. □

3.4.2 Impossibility Results

We now show that neither of the languages $a\Sigma^*$ and $\Sigma^* a$ are boolean combinations of languages recognized by BPQFA. First, we will introduce two lemmas which have been used extensively to prove impossibility results for KWQFAs and BPQFAs. These lemmas describe the state of the machine after reading a long sequence of letters. We will use the notation introduced in Section 3.1.2.

Lemma 3.22 *Let M be a KWQFA or BPQFA, then for all $w \in \Sigma^*$ there exists a partition $E_1^w \oplus E_2^w$ of the state space into orthogonal subspaces such that*

1. *$|\psi\rangle \in E_1^w$ implies $A'_w|\psi\rangle = A_w|\psi\rangle$, $A'_w|\psi\rangle \in E_1^w$, and*
2. *$|\psi\rangle \in E_2^w$ implies $\lim_{k\to\infty} A'_{w^k}|\psi\rangle = \vec{0}$.*

We will refer to the subspaces E_1^w and E_2^w as the *ergodic* and *transient* subspaces, respectively. Note that the behavior of a state $|\psi\rangle \in E_1$ behaves exactly as a state in an MOQFA. The next lemma describes the behavior of the machine while reading a string of the form $(x \cup y)^*$.

Lemma 3.23 *Let M be a KWQFA or BPQFA. Then for all pairs $x, y \in \Sigma^*$ there exists a partition $E_1^{x,y} \oplus E_2^{x,y}$ into orthogonal subspaces such that:*

1. *$|\psi\rangle \in E_1^{x,y}$ implies $A'_x|\psi\rangle = A_x|\psi\rangle$, $A'_x|\psi\rangle \in E_1^{x,y}$, $A'_y|\psi\rangle = A_y|\psi\rangle$, $A'_y|\psi\rangle \in E_1^{x,y}$*
2. *$|\psi\rangle \in E_2^{x,y}$ implies that for any $\varepsilon > 0$ there exists $w \in (x \cup y)^*$ such that $\|A'_t|\psi\rangle\| \leq \varepsilon$*

Let $|\psi\rangle$ be the state of a BPQFA after reading the input symbol. Then $|\psi\rangle$ can be split into $|\psi\rangle = |\psi_1\rangle + |\psi_2\rangle$, with $|\psi_i\rangle \in E_i^{x,y}$. Since the operations A'_x and A'_y are linear, we can think of the $E_1^{x,y}$ and $E_2^{x,y}$ component of the machine as acting independently. The transitions A'_x and A'_y act unitarily on the subspace $E_1^{x,y}$, so the $|\psi_1\rangle$ component has a periodic behaviour. The $|\psi_2\rangle$ component behaves in an essentially aperiodic fashion.

Intuitively, any language L recognized by a BPQFA machine using only the $E_1^{x,y}$ component will be such that $M(L)$ satisfies $x^\omega = y^\omega$, since the $E_1^{x,y}$ component behaves as an MOQFAs. Now let us consider the E_2 part. By the lemma, for words x and y and for all ε there exists w_x and w_y such that length of the $E_2^{x,y}$ part of the state after reading xw_x or yw_x is of length at most ε. Likewise there exists a word w such that the length of the $E_2^{x,y}$ part of the state after reading wx or wy is at most

ε. This suggests that transition monoid of a language recognized by the $E_2^{x,y}$ part will be $\mathcal{R}$-trivial and $\mathcal{L}$-trivial, and thus $\mathcal{J}$-trivial. Taken together, this suggests an upper bound of **EJ**, or in other words **BG**, on the monoid variety corresponding to boolean combinations of languages recognized by BPQFA. This is formally proven below:

Theorem 3.24 *The languages $a\Sigma^*$ and $\Sigma^* a$ cannot be expressed as Boolean combinations of languages recognized by BPQFA.*

Proof: We begin with the $a\Sigma^*$ result. By closure under inverse homomorphisms, it is sufficient to show this for $\Sigma = \{a, b\}$. Let M be a BPQFA that recognizes L with probability p, and let $|\psi\rangle = A'_{¢}(|q_0\rangle)$. The first step is to show that for any two prefixes $v, w \in \{a, b\}^+$ and any $\varepsilon > 0$, there exists $v', w' \in \{a, b\}^*$ such that $\|A'_{vv'}|\psi\rangle - A'_{ww'}|\psi\rangle\|_2 < \varepsilon$. Thus if $\varepsilon \leq \sqrt{p}$, it follows that M cannot distinguish between vv' and ww' with sufficiently large probability. As in Lemma 3.22, separate E_{non} into two subspaces E_1 and E_2 with respect to the words $x = a$ and $y = b$. Then we can rewrite $|\psi\rangle$ as $|\psi\rangle = |\psi_1\rangle + |\psi_2\rangle$, where $|\psi_i\rangle \in E_i$. By the lemma, and since A'_a and A'_b act unitarily on E_1, for any ε' there exists v' and w' such that $\|A'_{vv'}|\psi\rangle - |\psi_1\rangle\|_2^2 < \varepsilon'$ and $\|A'_{ww'}|\psi\rangle - |\psi_1\rangle\|_2^2 < \varepsilon'$. For any fixed ε, we can find a sufficiently small ε' so that $\|A'_{vv'}|\psi\rangle - A'_{ww'}|\psi\rangle\|_2^2 < \varepsilon$.

Suppose L is a Boolean combination of m languages $L_1, \ldots, L_m$, with each L_i recognized by a BPQFA M_i. As above, we can construct inductively on the languages L_i, two words $v = v_1 v_2 \ldots v_m \in \{a, b\}^*$ and $w = w_1 w_2 \ldots w_m \in \{a, b\}^*$ such that av and bw are indistinguishable for every M_i. Thus we must have either $\{av, bw\} \subseteq L$ or $L \cap \{av, bw\} = \emptyset$, and in either case $L \neq a\Sigma^*$. For the construction, we first choose

v_1 and w_1 so that, for all v' and w', av_1v' and bw_1w' are indistinguishable by M_1. Given that, for all v' and w', $av_1 \dots v_{i-1}v'$ and $bw_1 \dots w_{i-1}w'$ are not distinguishable by any of $M_1, \dots, M_{i-1}$; we construct v_i and w_i so that, for all v' and w', $av_1 \dots v_iv'$ and $bw_1 \dots w_iw'$ are indistinguishable by M_i.

A similar analysis can be used to show the Σ^*a result. Consider a single BPQFA M recognizing L with probability p. Let $|\psi\rangle = A'_{¢}|q_0\rangle$ be the initial state. Let $b \in \Sigma\backslash\{a\}$, and let E_1 and E_2 be as in Lemma 3.22 with $x = a$ and $y = b$. We can uniquely split $|\psi\rangle$ into $|\psi_1\rangle + |\psi_2\rangle$, where $|\psi_1\rangle \in E_1$ and $|\psi_2\rangle \in E_2$.

Suppose L is a boolean combination of m languages $L_1, \dots, L_m$ where each L_i is recognized by some BPQFA M_i with probability p_i. For any ε, we can construct a word $w = w_1 \dots w_m$ such that, for all w', the condition $\|A'_{ww'}|\psi\rangle - A'_{ww'}|\psi_1\rangle\|_2 < \varepsilon$ is met by each M_i. If we choose $\varepsilon < \sqrt{\min\{p_i\}}$, then there is an k such that for all i, machine M_i satisfies $\|A'_{ww'ab^k}|\psi\rangle - A'_{ww'a}\|_2 < p_i$. Thus we must have either $\{ww'ab^k, ww'a\} \subseteq L$ or $\{ww'ab^k, ww'a\} \cap L = \emptyset$, and in either case $L \neq \Sigma^*a$. □

It is somewhat surprising that the **BG** variety turns up in both the characterization of languages recognized by LQFA and of the boolean closure of languages recognized by BPQFA. But if we look at the proofs of both cases we see some similarities between the two. For example, in the recognizability part of both characterizations, we use the fact that **J** is a lower bound on the monoid variety and the fact that this can be extended to $\mathbf{J} * \mathbf{G}$ by a simulation of a transducer via vthe wreath product principle.

There are also similarities in the irreversibility part. We have seen that the class of languages recognized by LQFA and the class of languages recognized by

BPQFA can trade off randomness to allow some nonreversible behaviour. Also in both cases there is a finite bound on the number of irreversible transitions which may be invoked. This prevents either model from recognizing $\Sigma^* a$ or $a\Sigma^*$, which would require an unbounded number of irreversible transitions to recognize.

On the BPQFA side, this comes from the fact that irreversible steps can only be made by halting and rejecting, whereas if we want to accept w there has to be a nonzero bound on the probability that the machine has to read the entirety of a word w. In the LQFA case, the limit on the number of irreversible steps comes from the finite upper bound on the entropy of the state.

An unexpected consequence of these results is that the class of languages recognized by LQFA and the boolean closure of languages recognized by BPQFA are both closed under reversal. This is surprising considering the fact that there is no obvious way to construct an LQFA or BPQFA for L^R given an LQFA or BPQFA for L.

CHAPTER 4
BPQFA

In the previous chapter, we saw that we can use Eilenberg's variety theorem to characterize the boolean closure of languages recognized by BPQFA. We shall see in Section 4.2 that the class of languages recognized by BPQFA is in fact not closed under complement, and thus does not form a variety. This implies that our characterization of BPQFA is not complete. Fortunately, we can take advantage of an extension of Eilenberg's theorem to the case of *ordered monoids* [51]. This extension can be used to characterize language classes which are closed under all of the variety operations except complement. In this chapter, we use these algebraic methods to make several steps toward an exact characterization of the languages recognized by BPQFA.

The BPQFA model corresponds to an important subclass of KWQFA. There is a subtle power that arises from being able to halt before the end: it allows the state transition, conditioned on not halting, to be nonunitary. BPQFAs capture exactly what nonunitary behavior can be produced by halting before the end of the input. We formalize this idea in Section 4.1. It is exactly this power that is used to show that BPQFA and hence KWQFA can recognize the language $\Sigma^* a_1 \Sigma^* \dots a_k \Sigma^*$.

In Section 4.3, we give an overview of the theory of ordered monoids. In the remaining sections, we outline our progress towards an exact characterization of

BPQFA. In Section 4.4, we show several new BPQFA constructions, and in Section 4.5 we give some algebraic conditions for recognition by BPQFA.

4.1 Preliminaries

Recall that in Section 3.1.2 we described a simple and useful characterization of the behavior of BPQFAs, over all probabilistic choices, using scaled pure states. We adopt this point of view for the remainder of this chapter. In particular, we will not assume that $|\psi\rangle$ vectors have unit length. Also, we define S_{acc} (resp. S_{rej}, S_{non}) to be the space on which P_{acc} (resp. P_{rej}, P_{non}) projects.

We first present a geometric interpretation of the type of transformations which can be performed by BPQFA. It highlights the fact that by halting before the end, we can implement a projection operator on the space of scaled pure states.

Lemma 4.1 *Let M be a BPQFA (or KWQFA). For $a \in \Sigma$, there is a subspace S_a of S_{non} such that the operation $A'_a = P_{non}A_a$ can be decomposed as $A'_a = V_aP_a$, where P_a is a projection onto S_a and V_a is an operator which acts unitarily on the subspace S_{non}. Conversely, for every V_a that acts unitarily on S_{non} and every P_a that projects into a subspace of P_{non}, there is a unitary A_a such that $V_aP_a = P_{non}A_a$.*

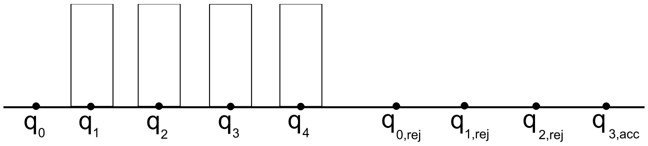
Proof: Let $A'_a = P_{non}A_a$, let $S_a = \{|\psi\rangle : |\psi\rangle \in S_{non} \wedge A_a|\psi\rangle \in S_{halt}\}$, and let $S'_a = \{|\psi\rangle : |\psi\rangle \in S_{halt} \wedge A_a|\psi\rangle \in S_{non}\}$. The sets S_a and S'_a are orthogonal subspaces and, by unitarity of A_a, they have the same dimension. Let R_a be a unitary matrix which swaps these two subspaces. We define $V_a = R_aA_a$ and we define P_a to be the projection onto S_a. Now $A'_a = V_aP_a$ as required. The matrix V_a is clearly unitary and it maps vectors in S_{non} to vectors in S_{non}. The construction of A_a in the converse is similar. □

We have mentioned that the intermediate stochastic state of a KWQFA after reading $w \in \Sigma^*$ is characterized by a triple $\psi_w = (p_{w,acc}, p_{w,rej}, |\psi_w\rangle)$, where $p_{w,acc}$ ($p_{w,rej}$) is the probability of accepting (rejecting) while reading w, and $|\psi_w\rangle$ is the state conditioned on not halting. Thus we can think of the output of a KWQFA as coming from two sources, from the values of $p_{w,acc}$ and $p_{w,rej}$, and from the outcome of the measurement on $|\psi_w\rangle$. The lemma above implies that BPQFA describe exactly what information can be gained just from measuring $|\psi_w\rangle$.

We now present the construction given by Brodsky and Pippenger for subword tests. We will make a few adjustments to the presentation to make it easier to present the generalized constructions that we describe in later sections.

Theorem 4.2 *([20]) BPQFA can recognize $L = \Sigma^* a_1 \Sigma^* \dots a_k \Sigma^*$.*

Proof: Brodsky and Pippenger called their construction a *trigger chain.* The machine M we construct for L consists of $k+2$ nonhalting states $q_0, \dots q_{k+1}$, k halt-and-reject states $q_{0,rej}, q_{1,rej}, \dots, q_{k-1,rej}$, and a single accept state $q_{k,acc}$. Reading the initial character ¢ sets the initial state to $\frac{1}{\sqrt{k+1}} \sum_{i=1}^{k+1} |q_i\rangle$, giving the following picture for the case $k = 3$:



A *trigger* is an operation which averages the amplitudes in two adjacent states q_i and q_{i+1}. This is achieved by applying the following operation on the states q_i,

q_{i+1}, and $q_{i,rej}$ respectively:

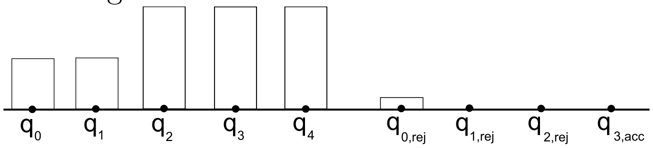
$$T = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{\sqrt{2}} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix}.$$

Before each application, the Q_{rej} coordinate will always be reset to 0. In this case, T has the following effect:

$$T \begin{bmatrix} \alpha & \beta & 0 \end{bmatrix}^T = \begin{bmatrix} \frac{\alpha}{2} + \frac{\beta}{2} & \frac{\alpha}{2} + \frac{\beta}{2} & \frac{\alpha}{\sqrt{2}} - \frac{\beta}{\sqrt{2}} \end{bmatrix}^T.$$

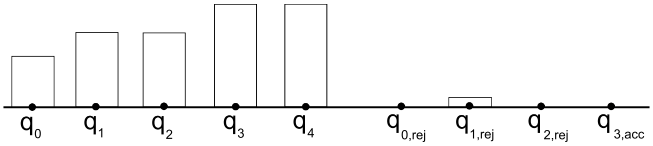
If α and β are positive reals, then T averages the amplitude between the first two states, and places any excess amplitude into Q_{rej}. After projecting the resulting vector into S_{non}, the trigger has the effect of projecting the initial vector onto the subspace spanned by $[\ 1\ \ 1\ \ 0\ ]^T$. If $\alpha = \beta$, then the trigger will have no effect. Let T_i for $i \leq k$ be the operation that applies the T operation to states q_i, q_{i+1}, and $q_{i,rej}$, and let T_k be the operation which applies the T operation to q_k, q_{k+1}, and $q_{k,acc}$.

Suppose for instance that the trigger T_0 is applied to the initial state. For $k = 3$, this produces the following state:



If this action is followed by a measurement with respect to $\{P_{acc}, P_{rej}, P_{non}\}$ then the machine will halt and reject with probability corresponding to the squared magnitude of the amplitude in $q_{0,rej}$, otherwise the contents of $q_{0,rej}$ will be emptied and we will be left with a balanced weight between q_0 and q_1. Note that applying T_1, T_2, or T_3 to the initial state would have no effect.

Now if we were to follow the T_0 action with a T_1 action, we would arrive at the following picture:



The operation T_3 puts a bounded amount of amplitude in q_3 if and only if q_3 has been lowered from its initial value, which happens only if T_0, T_1, T_2 have been applied in sequence. This is the idea that will be used to recognize subwords.

We now define the machine M. The transformation associated with each letter will involve a sequence T_i operations. Reading the initial character ¢ sets the initial state to $\frac{1}{\sqrt{k+1}} \sum_{i=1}^{k+1} |q_i\rangle$. On input $\sigma \in \Sigma$, M applies the transformation $A_\sigma = A_{\sigma,1} \cdots A_{\sigma,k}$, where

$$A_{\sigma,i} = \begin{cases} T_{i-1} & \text{if } a_i = \sigma, \\ I & \text{otherwise.} \end{cases}$$

Note in particular that the triggers will be applied from right to left.

Finally, let $U_\$$ be the operation that applies the operation T_k, while the remaining amplitude is sent to the rejecting states.

Let us establish some properties of the machine M.

Claim 2 *Let $\alpha_0, \ldots, \alpha_k$ be the amplitudes of $q_0, \ldots, q_k$ upon reading $w \in \Sigma^*$. Then the α_i's are positive real and satisfy $\alpha_0 \leq \alpha_1 \leq \cdots \leq \alpha_k$.*

To see this, note that the property holds for the initial state, and the T_i operators preserve this property. Also:

Claim 3 *Let A and B be sequences of T_i operations with $i \leq k$, and let BA be the composition of these sequences, applied from right to left. Then the amplitude of q_k after applying BA is at most the amplitude of q_k after applying A.*

Claim 3 follows by Claim 2 and an inductive argument on k. This can be extended to the following:

Claim 4 *Let $A_1, B_1, \ldots, A_m, B_m$ be sequences of $T_{i-1,i}$ operations with $i \leq k$. Then the amplitude of q_k after the operation $B_m A_m \cdots B_1 A_1$ is at most the amplitude of q_k after the operation $A_m \cdots A_1$.*

We now return to proving the correctness of M. Note that M will accept with nonzero probability if and only if the amount of amplitude in q_k has changed from $\frac{1}{\sqrt{k+1}}$ while reading the input sequence. For all i, q_i will have amplitude $\frac{1}{\sqrt{k+1}}$ as long as the subword $a_1 \ldots a_i$ has not been read. This can be shown by induction on i. For $i = 1$, note that the definitions of A_σ are such that for all $w \notin \Sigma^* a \Sigma^*$, A'_w does not contain the trigger T_0, while A'_{a_1} does contain the trigger T_0. Assume the claim is true for all $i' < i$. Thus q_{i-1} does not change its amplitude unless $a_1 \ldots a_{i-1}$ is read. When a_{i-1} is read, observe that the $A_{a_{i-1},i} \cdots A_{a_{i-1},k}$ operation is applied to states which have already equal in amplitude so the amplitude in q_i will not be change until a_i is read in sequence (note that this holds even if $a_{i-1} = a_i$). Thus if $a_1 \ldots a_k \notin w$, on reading w the amplitude in q_k will remain at its initial value and the machine will reject with probability 1.

It remains to show that every $w \in L$ is accepted with some probability $p > 0$. If $w = a_1 \dots a_k$ is read, the amount of amplitude in q_k will have decreased by a factor of $\frac{1}{2^k}$. By Claim 4, the amplitude of q_k on reading some word in L is at most $\frac{1}{2^k}$, and so there is a positive lower bound on the probability that a given word in L is accepted. This completes the proof of correctness. □

This construction demonstrates a surprising computational power of BPQFA. In the next section we prove an important limit on this power.

4.2 Impossibility Results

In this section we show that BPQFA cannot recognize the language $\overline{\Sigma^* b \Sigma^* a \Sigma^*}$. Since there is a BPQFA construction for the language $\Sigma^* b \Sigma^* a \Sigma^*$, this implies that BPQFA are not closed under complement.

Theorem 4.3 *([3]) For any $a \neq b$ and for any Σ satisfying $\{a, b\} \subseteq \Sigma$, BPQFA cannot recognize $\overline{\Sigma^* b \Sigma^* a \Sigma^*}$.*

Proof: Without loss of generality we let $\Sigma = \{a, b\}$. In this case, $\overline{\Sigma^* b \Sigma^* a \Sigma^*} = a^* b^*$. Our proof makes frequent use of the following corollary to the ergodic-transient lemma:

Corollary 4.1 *([6]) For any KWQFA (or BPQFA) M and word w we can define subspaces $E_1^w \oplus E_2^w = E_{non}$ such that $|\psi_1\rangle \in E_1^w$ implies $(A'_w)^i(|\psi_1\rangle) = (A_w)^i(|\psi_1\rangle)$ for all i, and $|\psi_2\rangle \in E_2^w$ implies $\lim_{i\to\infty} \|(A'_w)^i|\psi_2\rangle\| = 0$.*

We now establish a relationship between projection operations and idempotents. Let us define the following subclass of operations:

Definition 4.1 *A projection operator P is an S_{non}-*projection *if $P(S_{non}) \subseteq S_{non}$.*

Thus if we restrict our attention to vectors in E_{non}, then P will behave exactly as a projection. This is not true of all projections, e.g. it is not true of a projection onto a line that is not in S_{non} or perpendicular to S_{non}. This definition is relevant to our situation since the state $|\psi\rangle$ of M after reading some partial input must satisfy $|\psi\rangle \in E_{non}$.

Claim 5 *Any S_{non}-projection P can be simulated by a unitary transformation U and the BPQFA measurement.*

This is just a special case of Lemma 4.1.

Let L be a language recognized by a BPQFA M with probability p, and let $\varphi : \Sigma^* \to M(L)$ be the syntactic morphism. Clearly, if A'_a is an E_{non}-projection, then $\varphi(a)$ must be idempotent (i.e. $\varphi(a) = e = e^2$). We claim that the following converse is also true:

Claim 6 *Let L, M, p, and φ be as above, and let $\varphi(a)$ be an idempotent. Let M' be the machine constructed by replacing each A'_a with an E_{non}-projection onto E_1^a. Then M' also recognizes L with probability p.*

Proof: Suppose that M' does not recognize L with probability p. Thus, either M' accepts some word $w \in L$ with probability $p_w < p$, or M' accepts some word $w \notin L$ with probability $p_w > 0$. We consider the former case, the latter is similar.

Define ε so that $\sqrt{p} = \sqrt{p_w} + \varepsilon$. Let k be the number of occurrences of a in w. Note that $k > 0$, otherwise M and M' would accept w with the same probability. Let $w = w_0 a w_1 \dots w_{k-1} a w_k$ with $w_i \in (\Sigma \backslash \{a\})^*$. Let U be a unitary matrix such that U' is the E_{non}-projection onto E_1^a. We set j to be such that $||(A'_a)^j|\phi\rangle - U'|\phi\rangle||_2 = \varepsilon' < \frac{\varepsilon}{k}$

for all $|\phi\rangle \in E_{non}$ (we know by Corollary 4.1 that such a j exists). Now consider:

$$w' = w_0 a^j w_1 \ldots w_{k-1} a^j w_k.$$

We have $w' \in L$ since $\varphi(a)$ is idempotent. Let $|\psi\rangle = |q_0\rangle$ be the initial state of M. Note that, for all $|\phi\rangle$, $(A'_a)^j A'_{w_0}|\phi\rangle = U' A'_{w_0}|\phi\rangle + |\xi\rangle$ for some $|\xi\rangle$ satisfying $|||\xi\rangle|| < \varepsilon'$. So there exists a vector $|\xi_1\rangle$ such that $|||\xi_1\rangle|| < \varepsilon'$ and:

$$\begin{aligned} A'_{w_k}(A'_a)^j \cdots A'_{w_1}(A'_a)^j A'_{w_0}|\psi\rangle &= A'_{w_k}(A'_a)^j \cdots A'_{w_1}(U'A'_{w_0}|\psi\rangle + |\xi\rangle) \\ &= A'_{w_k}(A'_a)^j \cdots A'_{w_1}U'A'_{w_0}|\psi\rangle + |\xi_1\rangle. \end{aligned}$$

In general there exists vectors $|\xi_i\rangle$, $1 \le i \le k$, such that $|||\xi_i\rangle||_2 \le \varepsilon'$ for all i, and:

$$A'_{w_k}(A'_a)^j \cdots A'_{w_1}(A'_a)^j A'_{w_0}|\psi\rangle = A'_{w_k}U' \cdots A'_{w_1}U'A'_{w_0}|\psi\rangle + \sum_{i=1}^{k} |\xi_i\rangle$$

and so:

$$\begin{aligned} \mathbf{P}[M \text{ accepts } w'] &= \left\| P_{acc}A'_{\$}\left(A'_{w_k}U' \cdots A'_{w_1}U'A'_{w_0}|\psi\rangle + \sum |\xi_i\rangle\right)\right\|_2^2 \\ &\le \left(\left\|P_{acc}A'_{\$}\left(A'_{w_k}U' \cdots A'_{w_1}U'A'_{w_0}|\psi\rangle\right)\right\|_2 + \sum \||\xi_i\rangle\|_2\right)^2 \\ &< \left(\sqrt{p_w} + \varepsilon\right)^2 = p. \end{aligned}$$

The original M accepts w' with probability strictly less than p, a contradiction. □

Proof: [Theorem 4.3] Suppose for contradiction that M is a BPQFA which recognizes $\overline{\Sigma^* b \Sigma^* a \Sigma^*}$ with probability p. By Claim 6, we can assume without loss of generality that A'_a and A'_b are E_{non}-projections.

For any M and w, we can define $S_{w,rej}$ to be the set of all vectors $|\psi\rangle \in S_{non}$ such that $A'_w|\psi\rangle \in S_{rej}$ (if M halts with certainty before w is processed then $A'_w|\psi\rangle = \vec{0} \in S_{rej}$). It is easy to show by linearity that $S_{w,rej}$ is a subspace. For shorthand, define:

$$S_\alpha = \bigcap_{w,x,y\in\Sigma^*} S_{wbxay\$,rej}, \quad S_\beta = \bigcap_{x,y\in\Sigma^*} S_{xay\$,rej}, \quad S_\gamma = \bigcap_{y\in\Sigma^*} S_{y\$,rej}.$$

Observe that $S_\alpha \supseteq S_\beta \supseteq S_\gamma$. At all times, the state vector of M must be contained in the subspace S_α in order to recognize the language $\overline{\Sigma^*b\Sigma^*a\Sigma^*}$, since all words containing the subword ba must be rejected with certainty. When the first b is read, the state vector must fall into the subspace S_β, since by definition $|\phi\rangle \in S_\alpha$ implies $A'_b|\phi\rangle \in S_\beta$. If an a is read while the state vector is in the subspace S_β, the state vector must fall into the subspace S_γ, and the state vector must remain here until the end of the computation. We argue that any vector in $|\psi\rangle \in S_\alpha$ will fall into S_γ reading an a followed by a b, thus the word ab is rejected with certainty, a contradiction.

Define $\overline{S_\beta}$ be the subspace such that $S_\beta \oplus \overline{S_\beta} = \mathbb{C}^n$. The vector $|\psi_a\rangle = A'_a|\psi\rangle$ can be uniquely decomposed into $|\psi_\alpha\rangle + |\psi_\beta\rangle$, where $|\psi_\alpha\rangle \in S_\alpha \cap \overline{S_\beta}$ and $|\psi_\beta\rangle \in S_\beta$. We claim that $|\psi_\beta\rangle \in S_\gamma$. Observe that $A'_a|\psi\rangle = A'_aA'_a|\psi\rangle$, so $|\psi_\alpha\rangle + |\psi_\beta\rangle = A'_a(|\psi_\alpha\rangle + |\psi_\beta\rangle)$. Let $P_{\overline{\beta}}$ be the projection operator onto $\overline{E_\beta}$. Now:

$$\begin{aligned}|\psi_\alpha\rangle + |\psi_\beta\rangle = A'_a(|\psi_\alpha\rangle + |\psi_\beta\rangle) &\implies P_{\overline{\beta}}(|\psi_\alpha\rangle + |\psi_\beta\rangle) = P_{\overline{\beta}}(A'_a(|\psi_\alpha\rangle + |\psi_\beta\rangle)) \\ &\iff \psi_\alpha = P_{\overline{\beta}}(A'_a|\psi_\alpha\rangle) \\ &\implies |\psi_\alpha\rangle = A'_a|\psi_\alpha\rangle\end{aligned}$$

and so:

$$\begin{aligned}|\psi_\alpha\rangle + |\psi_\beta\rangle = A'_a(|\psi_\alpha\rangle + |\psi_\beta\rangle) &\iff |\psi_\alpha\rangle + |\psi_\beta\rangle = |\psi_\alpha\rangle + A'_a|\psi_\beta\rangle \\ &\iff |\psi_\beta\rangle = A'_a|\psi_\beta\rangle.\end{aligned}$$

From $|\psi_\beta\rangle \in S_\beta$ it follows that $A'_a|\psi_\beta\rangle \in S_\gamma$, and thus $|\psi_\beta\rangle \in S_\gamma$. Now consider $|\psi_{ab}\rangle = A'_b(|\psi_\alpha\rangle + |\psi_\beta\rangle)$. Since $A'_b(|\psi_\alpha\rangle + |\psi_\beta\rangle) \in S_\beta$ and $A'_b|\psi_\beta\rangle \in S_\beta$, we must have $A'_b|\psi_\alpha\rangle \in S_\beta$. But $|\psi_\alpha\rangle \perp S_\beta$ and A'_b is an S_{non}-projection, so we must have $A'_b|\psi_\alpha\rangle = \vec{0}$. Thus $|\psi_{ab}\rangle = A'_b|\psi_\alpha\rangle + A'_b|\psi_\beta\rangle = A'_b|\psi_\beta\rangle \in S_\gamma$. Thus, ab is rejected with certainty, as we wanted to show. □

4.3 Syntactic Ordered Monoids

In this section, we briefly describe how Eilenberg's framework can be extended to ordered monoids. As before, there is also a parallel theory of *ordered semigroups*. For a fuller treatment, see Pin [52].

We say that a relation $\leq$ on a monoid M is an *order* if it is reflexive, antisymmetric, and transitive, and *stable* if multiplication on the left and right preserves the relation. An *ordered monoid* is a monoid M equipped with a stable ordering $\leq_M$. If the ordering is implicit we omit the relation sign and say that M is an ordered monoid. Note that for any monoid M we can associate a natural ordered monoid by equipping it with the ordering defined by $x \leq y$ iff $x = y$. We call this the *trivial* ordering, and it is the one we associate by default to the free monoid Σ^* over Σ.

The standard definitions regarding monoids have natural extensions to the ordered case. The *direct product* of M and N is the ordered monoid with element set $M \times N$, and componentwise multiplication and order relation. An *ordered submonoid* is a subset M' of M which forms an ordered monoid with respect to the operation and ordering of M. For ordered monoids M and N, we say that the morphism $\varphi : M \to N$ is a *morphism of ordered monoids* if it is a monoid morphism that respects the ordering of M and N (i.e. $m_1 \leq_M m_2 \Rightarrow \varphi(m_1) \leq_N \varphi(m_2)$). We say that M *divides* N (writing $M \preceq N$) if there is an ordered submonoid N' of N and a surjective ordered morphism $\varphi : N' \to M$.

An *order ideal* of M is a set $F \subseteq M$ such that $(\forall y \in F)(\forall x \leq_M y) x \in F$. For an element $y \in M$ we define the *order ideal generated by* y, denoted $\downarrow y$, as $\{x : x \leq_M y\}$. Any order ideal is a union of the order ideals of its maximal elements.

We say that a language $L \subseteq \Sigma^*$ is *recognized by the ordered monoid* M if there exists an order ideal $F \subseteq M$ and a morphism of ordered monoids $\varphi : \Sigma^* \to M$ such that $L = \varphi^{-1}(F)$. Observe that this is the same as in the unordered case except for the restriction on the choice of F. If a monoid M is equipped with the trivial ordering, then there is no restriction on the choice of F and we obtain the original notion of recognition as a special case.

For any language $L \subseteq \Sigma^*$, there is a canonical ordered monoid $M(L)$ recognizing L, called the *ordered syntactic monoid* of L. It is canonical in the sense that it is the smallest ordered monoid recognizing L in the sense of the division relation and is unique up to isomorphism. In other words, the ordered monoid M recognizes L iff $M(L) \preceq M$.

The ordered syntactic monoid of L can be constructed from the minimal automaton for L. We outline this process below. A *congruential order* is a stable quasi-ordering on Σ^*. Each congruential order $\leq$ has a natural ordered monoid associated with it, which is called the *quotient monoid* and is denoted $\Sigma^*/\leq$, formed by taking the equivalence classes of $\leq$ as monoid elements with the natural ordering and multiplication. Let $\leq_L$ be a congruential order on Σ^* defined by $x \leq_L y$ if for all $u, v \in \Sigma^*$ we have $uyv \in L \Rightarrow uxv \in L$. Then $M(L) = \Sigma^*/\leq_L$. This monoid is finite if and only if L is regular.

A positive variety of languages is a class of regular languages closed under union and intersection (we say *positive boolean combinations*), inverse morphisms, and word quotient. A variety of ordered monoids is a set of finite monoids closed under taking submonoids, direct products, and surjective homomorphisms. If $\mathbf{V}$ is a variety of ordered monoids, then let $\mathcal{V}$ be the class of regular languages L such that $M(L) \in \mathcal{V}$.

For a variety of ordered monoids $\mathbf{V}$, we write $\mathbf{V} \rightarrow \mathcal{V}$ if $\mathcal{V}$ is the set of all languages recognized by some ordered monoid in $\mathbf{V}$. In this case set $\mathcal{V}$ forms a positive variety of languages. Furthermore, we have the following generalization of the variety theorem:

Theorem 4.4 *(Positive Variety Theorem) [51] The correspondence* $\mathbf{V} \rightarrow \mathcal{V}$ *defines a one-to-one correspondence between varieties of ordered monoids and the positive varieties of languages.*

All of the definitions and results so far can be applied to the semigroup case by replacing submonoids with subsemigroups, monoid morphisms with semigroup morphisms and Σ^+ for Σ^* in all places except for the definition of $\leq_L$.

In the remainder of this section, we give some examples varieties of ordered semgroups and monoids.

- Any variety $\mathbf{V}$ or monoids (resp. semigroups) can be treated as a variety of ordered monoids by equipping each monoid with the trivial ordering.
- The variety $\mathbf{J}^+$ consists of those monoids M such that the identity 1 of M is maximal in the ordering. A language is recognized by an ordered $\mathbf{J}^+$ monoid if and only if it is a positive boolean combination of languages of the form $\Sigma^* a_1 \Sigma^* \dots a_k \Sigma^*$. $\mathbf{J}^+$ is an ordered subvariety of $\mathbf{J}$.
- The variety $\mathbf{J_1^+}$ consists of all idempotent and commutative ordered monoids in $\mathbf{J}^+$. The class of languages recognized by ordered monoids in $\mathbf{J_1^+}$ are exactly those which can be expressed as positive boolean combinations of languages of the form $\Sigma^* a \Sigma^*$.
- The variety of semigroups $\mathbf{Nil}^+$ consists of those semigroups S which contain exactly one idempotent, and this idempotent is smaller than any element in the ordering. In particular, the unique idempotent must be a zero. A language L is recognized by a semigroup in $\mathbf{Nil}^+$ if and only if there exists a positive integer k such that $|x| > k \Rightarrow x \in L$. Semigroups in $\mathbf{Nil}^+$ are necessarily $\mathcal{J}$ - trivial.

For any variety $\mathbf{V}$ of ordered monoids, the set of ordered monoids $\overline{\mathbf{V}}$ obtained by reversing the ordering is again a variety. Let $\mathbf{J_1^-} = \overline{\mathbf{J_1}}$, and likewise define $\mathbf{J}^-$ and $\mathbf{Nil}^-$. A language L satisfies $M(L) \in \mathbf{V}$ if and only if $M(\overline{L}) \in \overline{\mathbf{V}}$.

Algebraic properties of ordered monoids can be expressed by inequations. For instance, the property of that the identity of M is maximal in the partial ordering (and thus $M \in \mathbf{J}^+$) can be expressed by the inequation $x \leq 1$. Likewise, membership in $\mathbf{J_1^+}$ can be expressed by the inequation $x \leq 1$ and the equations $xy = yx$ and $x = xx$. Analogous to the unordered case, it can be shown that every variety of ordered monoids can be characterized in terms of inequalities over $\hat{\Sigma}^*$ [54]. With a slight abuse of terminology, we will refer to such inequalities as identities. The variety $\mathbf{Nil}^+$, for example is characterized by the identity $x^\omega \leq y$.

4.3.1 Positive Varieties Defined by Composition

Semidirect products and relational morphisms are also extended to ordered monoids [55]. Let S be an ordered semigroup and let T be an ordered monoid. Let $\cdot$ be a left action of T on S , and suppose that $\cdot$ additionally satisfies:

- $t \leq t' \Rightarrow t \cdot s \leq t' \cdot s$, and
- $s \leq s' \Rightarrow t \cdot s \leq t \cdot s'$,

Then the set $S \times T$ forms an ordered monoid with the operation defined by $(s,t)(s',t') = (s + t \cdot s', tt')$ and the order defined by $(s,t) \leq (s' \leq t')$ iff $s \leq s'$ and $t \leq t'$.

For a morphism $\varphi : \Sigma^* \to W$. Let $\sigma_\varphi : \Sigma^* \to (W \times \Sigma^*)$ be the function defined by:

$$\sigma_\varphi(a_1 \dots a_n) = (\varphi(a_1), a_1)(\varphi(a_1a_2), a_2) \dots (\varphi(a_1 \dots a_{n-1}), a_n)$$

We call σ_φ the *sequential function associated with* φ.

Theorem 4.5 *(Wreath Product Principle of Ordered Monoids) [56] A language is in* $\mathbf{V} * \mathbf{W}$ *if and only if it is a positive boolean combination of languages of the form* $\sigma_\varphi^{-1}(V)$ *for some morphism* $\varphi : \Sigma^* \to W$*, with* $W \in \mathbf{W}$ *and* $M(V) \in \mathbf{V}$*.*

An *ordered relational morphism* $\phi : (S, \leq) \to (T, \leq)$ is a relational morphism from S to T such that the graph of the relation is an ordered monoid. For ordered variety $\mathbf{V}$, Such a relational morphism is said $\mathbf{V}$-relational if, for idempotents $e \in T$, we have that $\phi^{-1}(e) \in \mathbf{V}$. The *Mal'cev* product $\mathbf{V} \circledM \mathbf{W}$ of $\mathbf{V}$ and $\mathbf{W}$ is the variety of ordered monoids generated by the set of ordered monoids M for which there exists an ordered $\mathbf{V}$-relational morphism from $\mathbf{V}$ to $\mathbf{W}$.

4.3.2 Examples

In this section we consider a number of examples of ordered varieties of monoids related to our discussion.

ECOM$^-$

The variety $\mathbf{ECOM}^-$ of monoids M is the class of monoids such that the submonoid generated by the idempotents $E(M)$ of M forms a commutative monoid in $\mathbf{J}^-$. In other words, $\mathbf{ECOM}^-$ is characterized by the equations $x^\omega y^\omega = y^\omega x^\omega$ and $x^\omega \geq 1$.

This variety is equivalent to the variety $\mathbf{INV}^-$ generated by naturally ordered inverse monoids [55]. A monoid element m is said to be *inverse* if there exists a unique $\overline{m} \in M$ such that $m\overline{m}m = m$ and $\overline{m}m\overline{m} = \overline{m}$. A monoid M is a naturally ordered inverse monoid if every element of M is inverse and M is ordered by the rule $x \leq y$ iff there exists an idempotent e such that $x = ye$. In [55] it is further shown that $\mathbf{ECOM}^-$ is equivalent to $\mathbf{J}_1^- * \mathbf{G}$ and $\mathbf{J}_1^- \circledM \mathbf{G}$ [55].

In Pin [49], it was shown that a language L is in $\mathbf{ECOM}^-$ if and only if it can be recognized by a *reversible finite automata* (RFA), which we define below. A partial DFA M is a generalization of DFA where we permit the transition function δ to be incomplete function. We say that M recognizes the language $L(M) = \{w | \delta(w, q_0) \text{ is defined}, \delta(w, q_0) \in F\}$. A reversible automata is a partial DFA such that the transition function δ is injective.

$\mathbf{BG}^+$

The variety $\mathbf{BG}^+$ is a natural ordered counterpart to the variety $\mathbf{BG}$. It is defined as the class of all monoids in $\mathbf{BG}$ which satisfy the ordering condition $x^\omega \leq 1$. Conversely, it can be shown that monoids which satisfy $x^\omega \leq 1$ are in $\mathbf{BG}$, so this single inequation exactly characterizes $\mathbf{BG}^+$.

As with $\mathbf{BG}$, this variety has equivalent formulations. In particular:

$$\mathbf{BG}^+ = \mathbf{J}^+ * \mathbf{G} = \mathbf{J}^+ \circledM \mathbf{G}.$$

By applying the wreath product principle, we can show that a language L is recognized by a monoid in $\mathbf{BG}^+$ if and only if it is a positive boolean combination of languages of the form $L_0 a_1 L_1 a_2 \ldots a_k L_k$, where each L_i is a language whose syntactic monoid is a group.

$\mathbf{Nil}^+ \textcircled{M} \mathbf{J_1}$

In this section we give some basic properties of the variety $\mathbf{Nil}^+ \textcircled{M} \mathbf{J_1}$ of ordered monoids, as it is a central point in our investigation of BPQFA.

Theorem 4.6 *The variety $\mathbf{Nil}^+ \textcircled{M} \mathbf{J_1}$ is characterized by the inequations $(xy)^\omega x = (xy)^\omega = y(xy)^\omega$ and $x^\omega \leq x$.*

Proof: Suppose $M \in \mathbf{Nil}^+ \textcircled{M} \mathbf{J_1}$. Then by definition there is a monoid $N \in \mathbf{J_1}$ and an ordered relational morphism $\varphi : M \to N$ such that $\varphi^{-1}(e) \in \mathbf{Nil}^+$ for all idempotents e. Take any $x \in M$. By definition of φ, there is at least one element $e \in \varphi(x)$, and it is idempotent since N is idempotent. Then $e \in \varphi(x^\omega)$, and since $\varphi^{-1}(e) \in \mathbf{Nil}^+$ we have $x^\omega \leq x$. Furthermore $\mathbf{Nil}^+ \textcircled{M} \mathbf{J_1} \subseteq \mathbf{Nil} \textcircled{M} \mathbf{J}_1 = \mathbf{J}$ [48] so M satisfies $(xy)^\omega x = (xy)^\omega = y(xy)^\omega$

On the other hand, let M be a monoid that satisfies the equations above, and let $E(M)$ be the set of idempotents of M. Let N be the monoid with element set $2^{E(M)}$ with set intersection as the binary operation. It is easy to check that N is idempotent and commutative. Define the morphism $\varphi : M \to N$ by $\varphi(m) = \{e \in E(M) : em = e = me\}$. We check that φ is indeed a morphism. Clearly $e \in \varphi(x) \cap \varphi(y)$ implies $e \in \varphi(xy)$. Suppose $e \in \varphi(xy)$. Then $e = exy \leq_{\mathcal{R}} ex \leq_{\mathcal{R}} e$ implies $ex \, \mathcal{R} e$, and so $ex = e$ since M is $\mathcal{J}$-trivial. Likewise $xe = e$ and therefore $e \in \varphi(x)$, and in the same way $e \in \varphi(y)$.

Now suppose that $\varphi(x) = \varphi(y)$. M is $\mathbf{J}$-trivial so it satisfies the equation $x^\omega x = x^\omega = x x^\omega$. Thus $x^\omega \in \{e \in E(M) : ex = e = xe\} = \varphi(x) = \varphi(y) = \varphi(y^\omega)$. Likewise, $y^\omega \in \varphi(x^\omega)$. This implies that $x^\omega = x^\omega y^\omega = y^\omega$. Finally M satisfies $x^\omega \leq x$, so $y^\omega = x^\omega \leq x$ and therefore $\varphi^{-1}(x)$ satisfies the $\mathbf{Nil}^+$ equation. □

4.4 Recognizability results

We say that a BPQFA recognizes a language $L \subseteq \Sigma^*$ with certainty if it correctly distinguishes all $w \in \Sigma^*$ with probability 1. It is possible to completely characterize the class of languages recognized by BPQFA in this way.

Theorem 4.7 *The language L is recognizable by BPQFA with certainty if and only if it is recognized by a reversible finite automaton.*

Proof: This fact was implicitly stated in [33]. In one direction, it is easy to simulate a reversible finite automaton with a BPQFA with probability of acceptance 1. Let $R = (Q, \Sigma, q_0, \delta, F)$ be an RFA, we construct a BPQFA M' recognizing $L(R)$. Choose $Q' = Q \times \{0, 1\}$ and let $Q'_{acc} = F \times 1$, $Q'_{rej} = (Q - F) \times 1$, $Q'_{non} = Q \times 0$. Let $\delta_w : Q \to Q$ be the partial function defined as $\delta_w(q) = \delta(q, w)$. For every $\sigma \in \Sigma^*$ we simulate δ_σ on the $Q \times 0$ subspace, except that we map undefined transitions to $Q \times 1$. There is always some way to complete the partial function δ_w so that it forms permutation. Choose an arbitrary completion and let A_σ be the permutation matrix associated with that permutation.

Now let M be a BPQFA that recognizes L with certainty. Without loss of generality assume that $L \neq \emptyset$. The first step will be to construct a normalized machine M' which recognizes L with certainty. Denote by $|\psi_w\rangle$ the unnormalized state of the BPQFA upon reading w. For the empty string ε we can assume that $\||\psi_\varepsilon\rangle\| = 1$, otherwise the machine has halted with some nonzero probability, contradicting $L \neq \emptyset$.

Suppose wa is such that $\||\psi_w\rangle\| = 1$ but $\||\psi_{wa}\rangle\| < 1$. This implies that $|\psi_u\rangle \perp |\psi_w\rangle$ for any prefix u of w. By definition there is some nontrivial probability of rejecting wa. Since now we must ultimately reject all strings in $wa\Sigma^*$ with probability

1, we can create a new reject state $q_{w,r}$ and change the operator A_a so that it sends $|\psi_w\rangle$ directly into $|q_{w,r}\rangle$. We claim that for any w, w' such that w is not right congruent to w' and $\| |\psi_w\rangle \| = \| |\psi_{w'}\rangle \|$, we must have $|\psi_w\rangle \perp |\psi_{w'}\rangle$. Suppose not. Then w.l.o.g. there is a string x such that $wx \in L$ and $w'x \notin L$. We can write $|\psi_{w'}\rangle$ as $\alpha|\psi_w\rangle + \beta|\psi_w^\perp\rangle$ for $|\alpha| \neq 0$. Then $w'x$ is accepted with nonzero probability, a contradiction.

Finally, since any pair of states corresponding to different right congruence classes must be orthogonal, the set of reachable states for each congruence classes is contained within a subspace which is disjoint from the subspaces of the other classes. Since the transformations are unitary, we can keep track of the subspaces with a partial injective automaton. □

Theorem 4.8 *Let* $\mathbf{V}$ *be an ordered variety of semigroups or monoids. If BPQFA can recognize every language whose ordered syntactic monoid is in* $\mathbf{V}$*, then BPQFA can recognize every language whose ordered syntactic monoid is in* $\mathbf{V} * \mathbf{G}$*.*

Proof: As in Theorem 3.9, this is just a matter of applying the wreath product principle and simulating the transduction in the monoid. In this case, we need the ordered wreath product principle (Theorem 4.5). □

Combining this result, the fact that $\mathbf{BG}^+ = \mathbf{J}^+ * \mathbf{G}$, and the trigger chain construction, we get:

Theorem 4.9 *BPQFA can recognize every language whose syntactic monoid is in* $\mathbf{BG}^+$ *or* $\mathbf{J}_1^- * \mathbf{G}$*.*

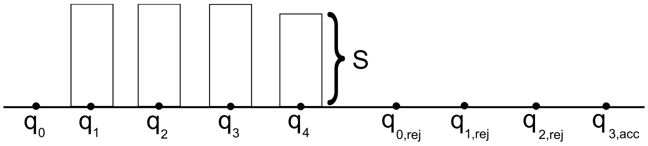


Figure 4–1: The initial state of the machine M_s.

There are, however, generalizations of the trigger chain idea that can be used to recognize a strictly larger class of languages. We consider some of these extensions in the remainder of the section.

Let M be a BPQFA that implements the trigger chain to recognize the language $\Sigma^* a_1 \Sigma^* \dots a_k \Sigma^*$. We modify this machine to recognize a different language. Recall that the last step of M is to apply the trigger T_k on states q_k and q_{k+1}. Let α_k and α_{k+1} be the amplitudes of these states after reading the input word. The machine accepts with probability $\frac{|\alpha_k - \alpha_{k+1}|^2}{2}$, which by construction will be at least some value $\delta > 0$ when $\alpha_k \neq \alpha_{k+1}$.

For $s \in [0, \frac{1}{\sqrt{k+1}}]$ we define M_s to be a BPQFA machine that behaves exactly as M above except that the amplitude of q_{k+1} is initialized to s instead of $\frac{1}{\sqrt{k+1}}$. A picture for the case $k = 3$ is given in Figure 4.4.

Under a certain formal condition, which we describe below, M_s will recognize a language with bounded probability. Let $f : \Sigma^* \to [0, 1]$ be the final amplitude of q_{k+1} as a function of the input word, let $S = \{f(w) | w \in \Sigma^*\}$, and for $s \in S$ let $L_s = \{w | f(w) = s\}$.

Lemma 4.10 *If there is a $\delta > 0$ such that $|s' - s| \geq \delta$ for all $s' \in S - \{s\}$, then M_s recognizes the language $\overline{L_s}$. Otherwise, M_s does not recognize any language with bounded probability.*

Proof: Suppose $\delta > 0$ exists. When $w \notin L_s$, w is accepted with probability at least $\frac{\delta^2}{2}$. When $w \in L_s$, the amplitude of q_k after reading the input will be the same as that of q_{k+1}, so w will be accepted with probability 0. On the other hand, if no such δ exists then for all $\varepsilon > 0$ there exists a word w that is accepted with probability ε' such that $0 \leq \varepsilon' \leq \varepsilon$. □

Here is a simple example of a language L which can be recognized by an application of Lemma 4.10:

$$L = \overline{\Sigma^* a \Sigma^* b \Sigma^*} \cup \Sigma^* a \Sigma^* b \Sigma^* a \Sigma^* b \Sigma^*.$$

The syntactic monoid of L is not in $\mathbf{BG}^+$ or in $\mathbf{J}_1^- * \mathbf{G}$, and so it falls outside of the constructions that we have presented thus far. To see this, we consider the syntactic monoid $M(L)$ and the syntactic morphism $\varphi : \Sigma^* \to M(L)$. Observe that $\varphi(a)$ and $\varphi(b)$ are idempotents, and $\varphi(ab) \neq \varphi(ba)$, and therefore $M(L) \notin \mathbf{J}_1^- * \mathbf{G}$. Now suppose $\varphi(a) \leq 1$. This would imply $\varphi(a)\varphi(b) \leq \varphi(b)$, contradicting $b \in L$ and $ab \notin L$.

Let $t = \frac{1}{\sqrt{3}}$, and let $s = \frac{3}{4}t$. Let M be the BPQFA with the trigger chain for the language $\Sigma^* a \Sigma^* b \Sigma^*$. We claim that M_s recognizes the language L. We apply Lemma 4.10. We first show that $L_s = \Sigma^* a \Sigma^* b \Sigma^* \cap \overline{\Sigma^* a \Sigma^* b \Sigma^* a \Sigma^* b \Sigma^*}$. The amplitudes α_0, α_1, α_2 of q_0, q_1, q_2 are initialized to $\begin{bmatrix} 0 & t & t \end{bmatrix}^T$ and will remain at these values until the first a is read. At this point, the amplitudes shift to $\begin{bmatrix} \frac{1}{2}t & \frac{1}{2}t & t \end{bmatrix}$ until the next b is read, setting the amplitudes to $\begin{bmatrix} \frac{1}{2}t & \frac{3}{4}t & \frac{3}{4}t = s \end{bmatrix}^T$. We can similarly check that q_2 will stay at value s until the subword *abab* is seen, at which point α_3 will become $\frac{11}{16}t$. Thus taking $\delta = \frac{1}{16}$ in Lemma 4.10 we get the desired result.

We can generalize the construction above to the following language, which seems to capture the class of languages which can be recognized by the technique of Lemma 4.10.

Theorem 4.11 *For $a_1 \ldots a_k \in \Sigma^*$ let $L(a_1 \ldots a_k) = \Sigma^* a_1 \Sigma^* \ldots a_k \Sigma^*$. Then for all $a_1 \ldots a_k$ the language*

$$L = \overline{L(a_1 \ldots a_k)} \cup \left(\bigcup_i L(a_1 \ldots a_i a_{i-1} a_i \ldots a_k) \right)$$

is recognized by a BPQFA.

Proof: We apply the trigger extension to the machine M recognizing $L(a_1 \ldots a_k)$. Let $t = \frac{1}{\sqrt{k+1}}$, the initial weights of $q_1, \ldots, q_k$, and let $s = (1 - \frac{1}{2^k})t$.

First, suppose that $w \in L$. There are two cases. If $a_1 \ldots a_k \notin w$ then $f(w) = t > s$. Otherwise, suppose that $a_1 \ldots a_k \in w$ and there is some i such that

$$a_1 \ldots a_i a_{i-1} a_i \ldots a_k \in w.$$

Check that $f(a_1 \ldots a_i a_{i-1} a_i \ldots a_k) < s$, and thus by Claim 4,

$$f(w) \leq f(a_1 \ldots a_i a_{i-1} a_i \ldots a_k) < s.$$

Let α_j be the amplitude of q_j and suppose $w \notin L$. Then w can be decomposed as $w = w_0 a_1 w_1 \ldots a_k w_k$, with $a_j \notin w_{j-1}$. After reading $w_0 a_1$ the amplitude in q_1 becomes $\frac{1}{2}t$, and after reading $w_0 a_1 w_1 a_2$ the amplitude in q_2 becomes $\frac{3}{4}t$. Suppose that after reading $w_0 a_1 \ldots w_{i-1} a_i$ we have $\alpha_i = (1 - \frac{1}{2^i})t$ and $\alpha_j = t$ for $j > i$. Now since $a_{i+1} \notin w_i$, α_i will not change while reading w_i unless $a_{i-1}a_i \in w_i$, which is not possible assuming $w \notin L$. □

By taking unions and intersections of languages of the form in Theorem 4.11, we can recognize e.g. the language $\overline{L(ab)} \cup L((ab)^n)$. We give a general construction below.

Theorem 4.12 *For all $a_1 \dots a_k$, $n \in \mathbb{N}$ the language*

$$\overline{L(a_1 \dots a_k)} \cup \left(\bigcup_i L(a_1 \dots a_i (a_i a_{i-1})^n a_{i+1} \dots a_k) \right)$$

is recognized by a BPQFA.

Proof: We show that this language is a positive boolean combination of the languages in Theorem 4.11. We explicitly construct this boolean combination in an iterative fashion. At every step we will have constructed a language of the form $L' = \overline{L(a_1 \dots a_k)} \cup \bigcup_{w \in W} L(w)$ for some set $W \subseteq \Sigma^*$. We initially take $W = \{a_1 \dots a_i a_{i-1} a_i \dots a_k : i \leq k\}$ so that the language is exactly the one obtained from Theorem 4.11. We wish to grow the words in W so that each of them contain at least one subword from the set $\{a_1 \dots a_i (a_{i-1} a_i)^n a_{i+1} \dots a_k : i \leq k\}$. At each step we remove a word $w = w_1 \dots w_m$ of minimal length and then take the intersection $L' \leftarrow L' \cap (\overline{L(w)} \bigcup_i L(w_1 \dots w_i w_{i-1} w_i \dots w_m))$. This effectively changes W to

$$W \leftarrow (W - \{w\}) \cup \{w_1 \dots w_i w_{i-1} w_i \dots w_{k_w} : i \leq m\}.$$

We continue this process until the length of each string in W is at least kn. At this point, L' will be such that every word in W contains at least one subword from the set $\{a_1 \dots a_i (a_{i-1} a_i)^n a_{i+1} \dots a_k : i \leq k\}$ as a subword. Finally,

$$\begin{aligned} L' &\leftarrow L' \cup \bigcup_i L(a_1 \dots a_i (a_{i-1} a_i)^n a_{i+1} \dots a_k) \\ &= \overline{L(a_1 \dots a_k)} \cup \bigcup_i L(a_1 \dots a_i (a_{i-1} a_i)^n a_{i+1} \dots a_k), \end{aligned}$$

which completes the construction. □.

The trigger chain can be used in a more subtle fashion to recognize languages which seem to fall out of the scope of Theorem 4.12. We give an example below.

Theorem 4.13 *BPQFA can recognize the language* $(L(cb) \cap \overline{L(ca)}) \cup L(cab)$.

Proof: The construction of M recognizing this language contains two trigger chains running in parallel. We divide the amplitude equally among both chains, so that if $p_{i,acc,w}$ is the probability that the ith chain accepts, so that the probability that M accepts is $\frac{1}{2}(p_{1,acc,w} + p_{2,acc,w})$. We use the letter q to denote states from the first chain and we use r for the second chain.

The first chain accepts words in $w \in L(cab)$ with bounded positive probability, and words $w \notin L(cab)$ with probability 0. The second chain accepts words in $L(cb) \cap \overline{L(ca)}$ with bounded positive probability, and words $w \notin (L(cb) \cap \overline{L(ca)}) \cup L(cab)$ with probability 0. Under these two conditions, all words in the language are accepted with bounded probability, and words not in the language are accepted with probability 0.

To implement the first chain we simply run the original trigger chain construction on the q states. The second chain uses only four states: $r_0, \dots, r_3$. The amplitudes in r_1, r_2, r_3 will be initialized to $\frac{1}{\sqrt{6}}$. On reading c the operation T_0 is applied to r_0, r_1, and likewise on reading b or a the operations T_1 or T_2 are applied. When the $

operator is read, we apply a trigger operation on r_2 and r_3, but the excess amplitude is sent to an accept state. If the amplitudes in r_2 and r_3 are different, then the second chain accepts with nonzero probability. Furthermore, if a word containing cb but not ca is read, there is a positive lower bound on the probability of acceptance. Thus the two chains meet the required conditions, and M recognizes the language as desired. □

Yet another modification of the trigger chain construction can recognize the reversal of this language: $(L(bc) \cap \overline{L(ac)}) \cup L(bac)$. All of these languages that we have introduced are examples of languages whose ordered syntactic monoid is in $\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}$. Consider for example the language $L = (L(cb) \cap \overline{L(ca)}) \cup L(cab)$. As this is a boolean combination of subword languages, the syntactic monoid $M(L)$ satisfies the equations $(xy)^\omega x = (xy)^\omega = y(xy)^\omega$ for the variety $\mathbf{J}$. Furthermore, we claim that $M(L)$ satisfies $x^\omega \leq x$. Let $w = w_1 w_2 w_3$ be any word contained in L. We need to show that for sufficiently large i and for all $j \geq 0$ $w_1(w_2)^{i+j}w_3 \in L$ This is clearly true if $w \in L(cab)$, suppose that this is not the case. Then clearly $w \in (L(cb) \cap \overline{L(ca)})$, and powering w_2 will put a c to the right of an a. Furthermore w_1 must not contain the letter c, otherwise w would contain the subword ca. Thus, $w_1 w_2^2 w_3 \in L(cab)$ and $M(L)$ does indeed satisfy the equations of $\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}$.

Our investigations seem to suggest the following:

Conjecture 4.1 *Every language whose ordered syntactic monoid is in* $\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}$ *can be recognized by BPQFA.*

If we can show this, then by Theorem 4.5 this can immediately be extended to the class of languages whose ordered syntactic monoid is in $(\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}) * \mathbf{G}$.

In the next section we will see that we can use algebraic techniques to extend the impossibility result in Section 4.2 to all languages whose ordered syntactic monoid is not in $(\mathbf{Nil}^{+} \textcircled{\scriptsize M} \mathbf{J_1}) \textcircled{\scriptsize M} \mathbf{G}$.

4.5 More Impossibility Results

Our first result is that can we can generate an algebraic condition for recognizability by BPQFA from the fact that BPQFA cannot recognize $\overline{\Sigma^* a \Sigma^* b \Sigma^*}$.

Theorem 4.14 *Let $\mathcal{V}$ be the positive variety of languages recognized by BPQFA, and let $\mathbf{V}$ be the corresponding variety of ordered monoids. Then $\mathbf{V}$ satisfies the equation $(x^\omega y^\omega)^\omega \leq x^\omega y^\omega$.*

Proof: Suppose that $(M, \leq) \in \mathbf{V}$ does not satisfy this inequation. We use this to show that $\mathbf{V}$ contains the ordered syntactic monoid of $\overline{\Sigma^* a \Sigma^* b \Sigma^*}$, causing a contradiction. Since M does not satisfy the equation there are two idempotent elements m_x and m_y of M such that $(m_x m_y)^\omega \not\leq m_x m_y$. Let $(M', \leq)$ be the submonoid generated by m_x and m_y. By Theorem 3.20 The ordered monoid M' is contained in $\mathbf{BG}$ and so it satisfies $(x^\omega y^\omega)^\omega x^\omega = (x^\omega y^\omega)^\omega = y^\omega (x^\omega y^\omega)^\omega$. This implies that $(m_x m_y)^\omega \not\leq m_x$ and $(m_x m_y)^\omega \not\leq m_y$.

Now consider the sequence of monoid elements defined by $m_1 = m_x m_y$, $m_2 = m_1 m_x$, $m_{2i+1} = m_{2i} m_y$, $m_{2(i+1)} = m_{2i+1} m_x$. Let n be the minimum index such that $m_n = m_{n+1} = (m_x m_y)^\omega$. Furthermore let j be the maximal index such that $j < n$ and $(m_x m_y)^\omega \not\leq m_j$. Suppose j is even, the other case is similar. Then we can express m_j as a product $m_j = m_L m_R$ with $m_L = (m_x m_y)^i$ and $m_R = m_x$. Then we can define an ordered congruence $\preceq$ on the elements of M' according to the rule $m \preceq m'$ if for all $u, v \in M'$, $m_L u m v m_R \leq m_L u m' v m_R$ Define M'' to be the quotient of that ordered congruence. Then we can recognizes $\overline{\Sigma^* a \Sigma^* b \Sigma^*}$ by the morphism $\varphi : \{a, b\}^* \to M''$ defined by $\varphi(a) = [m_x]$,$\varphi(b) = [m_y]$, which contradicts the assumption that $\mathcal{V}$ is the class of languages recognized by BPQFA. $\square$

We can furthermore show the following:

Theorem 4.15 *Let $\mathcal{V}$ be the positive variety of languages recognized by BPQFA, and let $\mathbf{V}$ be the corresponding variety of ordered monoids. Then $\mathbf{V} \subseteq (\mathbf{Nil}^+ \circledM \mathbf{J_1}) \circledM \mathbf{G}$*

Proof: It is sufficient to show:

$$[[(x^\omega y^\omega)^\omega x^\omega = (x^\omega y^\omega)^\omega = y^\omega (x^\omega y^\omega)^\omega, (x^\omega y^\omega)^\omega \leq x^\omega y^\omega]] \subseteq (\mathbf{Nil}^+ \circledM \mathbf{J_1}) \circledM \mathbf{G}.$$

For this, we use a consequence of Ash's Type II theorem that was formulated in [55]. First, we need a few definitions. We say that $\overline{x}$ is a weak inverse of x if $\overline{x} x \overline{x} = \overline{x}$. We say that a monoid M is closed under weak conjugation if for all $m \in M$ and all $x, \overline{x} \in M$ satisfying $\overline{x} x \overline{x} = \overline{x}$ we have $x m \overline{x} \in M$ and $\overline{x} m x \in M$. For a monoid M, define $D(M)$ to be the smallest monoid closed under weak conjugation.

Theorem 4.16 *([55]) Let $\mathbf{V}$ be a variety of ordered monoids. Then $M \in \mathbf{V} \circledM \mathbf{G}$ if and only if $D(M) \in \mathbf{V}$.*

Since $M \in \mathbf{BG} = \mathbf{J} \circledM \mathbf{G}$, we know that $D(M)$ is $\mathbf{J}$-trivial (i.e. it satisfies $(xy)^\omega x = (xy)^\omega = y(xy)^\omega$), so it is sufficient to show that $D(M)$ satisfies $x^\omega \leq x$.

By definition we know that $D(M)$ is generated by taking all products of monoid elements formed by the following context-free grammar G with rules $S \to 1$, $S \to eS$ and $S \to eS$ for all $e = e^2$, and $S \to x S \overline{x} | \overline{x} S x$ for all x satisfying $\overline{x} = \overline{x} x \overline{x}$. We argue that for all $w \in D(M)^*$, $eval(w)^\omega \leq eval(w)$ by induction on length of w. To simplify notation, for the remainder of the proof we drop the $eval(w)$ function.

For $|w| = 1$, w contains just one idempotent so we clearly have $w^\omega = w$ The case $|w| = 2$ was shown above. For the inductive case $|w| = k > 2$, w must be of the form $w'e$ or ew' for idempotent e, or of the form $xw'\overline{x}$ or $\overline{x}w'x$.

For the first case, by the induction hypothesis we have $w'e \geq (w')^\omega e \geq (w'^\omega e)^\omega \geq (w'e)^\omega$, where the last inequality follows from the $\mathcal{J}$-trivial equations. The second case is similar. For the third case, observe that $\overline{x}x = (\overline{x}x)^2$. $w = xw'\overline{x}$. We then have: $xe\overline{x} = xe\overline{x}x\overline{x} \geq x(e\overline{x}x)^\omega\overline{x} = x(e\overline{x}x)^\omega e\overline{x} = (xe\overline{x})^\omega$. Thus, $D(M)$ satisfies the equations for $\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}$ and we are done. $\square$

To summarize, our current results seem to point to the following conjecture:

Conjecture 4.2 *Let $\mathcal{V}$ be the positive variety of languages recognized by BPQFAs, and let* $\mathbf{V}$ *be the corresponding variety of ordered monoids. Then:*

$$\mathbf{V} = (\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}) * \mathbf{G} = (\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}) \mathbin{\textcircled{M}} \mathbf{G}.$$

There are two components to this conjecture. The first component is the conjecture $(\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}) * \mathbf{G} \subseteq \mathbf{V}$. To prove this, it would be sufficient to show that $\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1} \subseteq \mathbf{V}$. The issue here is that we do not seem to have a sufficient combinatorial understanding of the class of languages recognized by ordered monoids in $\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}$ in order to prove this result. However, we have identified several non-trivial examples of languages we have found to have ordered syntactic monoid in $\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}$, and our understanding of this language class continues to develop.

The second component is the conjecture $(\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}) * \mathbf{G} = (\mathbf{Nil}^+ \mathbin{\textcircled{M}} \mathbf{J_1}) \mathbin{\textcircled{M}} \mathbf{G}$. While it is not true in general that $\mathbf{V} * \mathbf{G} = \mathbf{V} \mathbin{\textcircled{M}} \mathbf{G}$, in the background sections we referred to several examples of this phenomena, including the case that V is equal to $\mathbf{J}$, to $\mathbf{J}^+$, or $\mathbf{J}_1^-$. It is also true whenever $\mathbf{V}$ is *local.* We believe that $\mathbf{V} * \mathbf{G} = \mathbf{V} \textcircled{M} \mathbf{G}$ holds for the case of $\mathbf{V} = \mathbf{Nil}^+ \textcircled{M} \mathbf{J_1}$. If so, then we will be very close to proving an exact characterization of the languages recognized by BPQFA.

CHAPTER 5
GQFA

We have seen that MOQFAs, which apply only a unitary transformation for each input symbol, can recognize only those languages whose syntactic monoid are groups. This is due to the inherent reversibility of unitary transformations. The set of states $\{U^k|\psi\rangle : k > 0\}$ contains vectors which are arbitrarily close to $|\psi\rangle$, so for arbitrary values of k it is impossible to tell with bounded precision whether the matrix U was applied at all.

In this thesis we have considered two types of generalized transformations: one is to allow measurements that introduce randomness to the state, and the other is to allow the machine to halt before reading the entire input. LQFA can use only the first type of generalization, KWQFA can use only the second, and generalized QFA (GQFA) can use both. We know exactly the limits of LQFA, and a number of important lower bounds are known for KWQFA. In this section, we combine our knowledge of LQFA and KWQFA to prove lower bounds on GQFA.

KWQFAs are significantly more powerful than MOQFAs in that they are able to perform subword tests. However, there are still quantifiable limitations on the power of KWQFAs. The key issue is that reversible transformations can only be achieved by halting before reading the full input word. In order to distinguish between prefixes of arbitrary length with bounded probability, it is necessary that the entire input word be read with bounded probability. Thus, it is important to understand the trade-off between reversibility and the probability of reading the entire input word. A key result in this regard, which we refer to as the *ergodic-transient lemma*, was given by Ambainis and Freivalds [4] (Lemma 3.22) and extended by Ambainis, Ķikusts and Valdats [6]. The ergodic-transient lemma led to a series of results which gave lower bounds on the probability of KWQFA recognizing certain languages L, in terms of the minimal automaton for L.

In this chapter, we show a parallel result for the case of GQFA. As the probability of halting while active tends to zero, in the limit a GQFA will behave exactly as an LQFA. Combined with our recent characterization of the languages recognized by LQFA, this give us a much clearer picture of the power of GQFA. In Section 5.1 we review the known results for KWQFA, and compare them to the known results for GQFA. In Section 5.2, we prove the generalization of the ergodic-transient lemma to the case of GQFAs. In Section 5.3, we review the main consequences of the lemma.

5.1 Review of KWQFA Impossibility Results

In this section we review the impossibility results that have been obtained for KWQFA. Since the languages recognized by KWQFA are closed under inverse morphisms and word quotient, a proof that a single language L cannot be recognized by KWQFA can immediately be extended to a class of languages. Furthermore, the condition for the impossibility result can be stated succinctly in terms of structural properties of the minimal automaton. Several results of this kind were demonstrated in the literature [4, 5, 6, 20]. The conditions in this case are called *forbidden constructions*. We formalize this idea below.

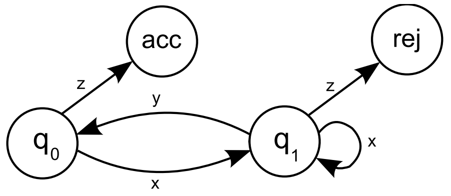


Figure 5–1: The forbidden construction of Theorem 5.1.

We will define a *partially specified automaton* P to be a tuple of the form $P = (Q, q_0, \Sigma, \delta, Q_{acc}, Q_{rej})$ such that Q is a finite set of states, $q_0 \in Q$ is the initial state, Σ is the input alphabet, $\delta : Q \times \Sigma \rightarrow Q$ is a partial transition function, and Q_{acc} and Q_{rej} are disjoint subsets of Q. Let $D = (Q', q'_0, \Gamma, \delta', F')$ be a DFA. We say that P *occurs within* D if there is a mapping $\eta : Q \rightarrow Q'$ and $\eta : \Sigma \rightarrow \Gamma^*$ such that $\eta(Q) \subseteq Q'$, $q'_0 \in \eta(Q)$, $\eta(Q_{acc}) \subseteq F'$, $\eta(Q_{rej}) \subseteq Q' - F'$, and for all defined transitions $\delta(q, \sigma) = q'$ of D we have $\delta'(\eta(q), \eta(\sigma)) = \eta(q')$. We call P a *forbidden construction* if the occurrence of P within the minimal automaton for L implies an impossibility result for L. The first result of this form was given by Brodsky and Pippenger:

Theorem 5.1 *([20]) If L is such that the forbidden construction in Figure 5.1 occurs within the minimal automaton for L, then L is not recognizable by a KWQFA with probability $\frac{1}{2} + \varepsilon$ for any $\varepsilon > 0$.*

Proof Sketch: This is an immediate consequence of the proof that KWQFA cannot recognize $\Sigma^* a$ and the KWQFA closure properties. Let w be a word such that the transition function δ of the minimal automaton for L satisfies $\delta(q_0, w) = q_1$. If we consider a morphism $\varphi : \{a, b\}^* \rightarrow \Sigma^*$ defined by $\varphi(a) = xy$ and $\varphi(b) = x$, then the quotient $w^{-1}\varphi^{-1}(L)z^{-1}$ will equal $\Sigma^* a$, which cannot be recognized by KWQFA. The same line of reasoning will also hold for GQFAs.

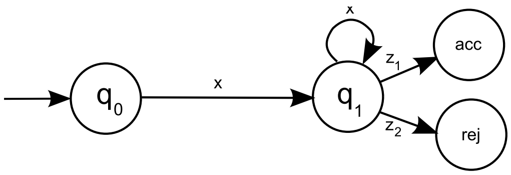


Figure 5–2: The forbidden construction of Theorem 5.2.

We mentioned in Section 3.1.1 that there is no general strategy to a priori boost the probability of recognizing a language L for KWQFA. The impossibility of such a boosting construction follows from the theorem below. The proof of this theorem relies on ergodic-transient lemma for KWQFA.

Theorem 5.2 *([4]) If L contains the forbidden construction given in Figure 5–2, then L cannot be recognized by a KWQFA with probability $p > 7/9$.*

There are languages whose minimal automaton contain this forbidden construction yet can be recognized by KWQFA with bounded error, for instance the language $\Sigma^* a \Sigma^* b \Sigma^*$. The bound on p was improved in [5] to $(54 + 4\sqrt{7})/81 \approx 0.7726$, and this was shown to be tight. This result does not generalize to GQFAs since GQFAs can implement the LQFA construction of Theorem 3.7 to recognize $\Sigma^* a \Sigma^* b \Sigma^*$ with probability $1 - \varepsilon$ for any $\varepsilon > 0$.

The next theorem is also a consequence of the ergodic-transient theorem. It was key to proving that the class of languages recognized by KWQFA is not closed under union.

Theorem 5.3 *([6]) If the minimal automaton for L contains states q, q_1, q_2 and words x, y, z_1, z_2 such that*

- *reading x while in q brings you to q_1,*
- *reading y while in q brings you to q_2,*

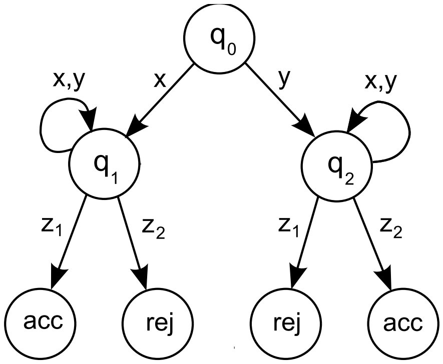


Figure 5–3: The forbidden construction of Theorem 5.3.

- *reading x or y while in q_1 or q_2 brings you back to the same state,*
- *reading z_1 (z_2) while in q_1 brings you to a(n) accept (reject) state, and*
- *reading z_1 (z_2) while in q_2 brings you to a reject (accept) state*

(i.e. if the minimal automaton contains the forbidden construction of Figure 5.1), then L cannot be recognized by KWQFA with probability $1/2 + \epsilon$.

We will now show how this result is used to prove that KWQFA are not closed under union. Later in Section 5.3 we will see that a similar argument applies to GQFA.

Theorem 5.4 *([6]) KWQFA are not closed under union.*

Proof: We consider the languages L_1 and L_2 corresponding to the languages recognized by automata D_1 and D_2 in Figure 5–4. We give a construction for L_1; the construction for L_2 is similar. Our machine M will have six nonhalting states $s_1, \ldots, s_6$. The transformation for the ¢ symbol will initialize the state to $\frac{1}{\sqrt{3}}(|s_1\rangle + |s_2\rangle + |s_3\rangle)$. Reading an a will swap the amplitudes in states s_1, s_2, and s_3 for that of s_4, s_5, and s_6 respectively. When a b is read, the amplitude in s_2, s_3, and s_5 will be sent to the

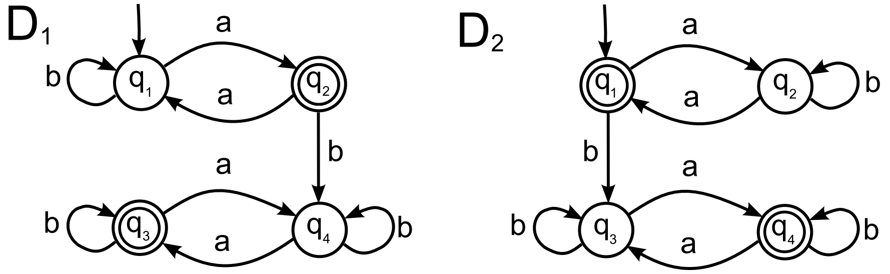


Figure 5–4: The minimal automata for L_1 and L_2 in Theorem 5.4.

reject state, and s_6 will be sent to the accept state. Finally, at the end of the input, the amplitude in s_1, s_5, and s_6 will sent to accept and the amplitude of s_2, s_3, and s_4 will be sent to reject. These transitions are injective so this swapping of states can be implemented unitarily.

If no b's are read, then it is easy to see that M will accept with probability 2/3 on reading an odd number of a's. If a b is read after an even number of a's, then the machine rejects with probability at least 2/3. Finally, if b is read after an odd number of a's, then M rejects with probability at most 1/3 if the total number of a's is even, and with probability 2/3 if the total number of a's is odd. On the other hand, consider $L_3 = L_1 \cup L_2$. L_3 consists of the set of strings containing no b or an odd number of a's after the first b. If we take the strings $x = ab$ and $y = ba$, we see that the minimal automaton for L_3 contains the forbidden construction of Theorem 5.3, and so L_3 cannot be recognized. □

In the next section, we extend the ergodic-transient lemma to GQFAs.

5.2 Ergodic-Transient Lemma

We will adopt the following notation. Let $M = (Q, q_0, \Sigma, \{U_a\}, \{P_a\}, Q_{acc}, Q_{rej})$ be a GQFA, where P_a is the measurement $\{P_{a,i} : 1 \leq i \leq m_a\}$. Let S_{acc}, S_{rej}, S_{halt},

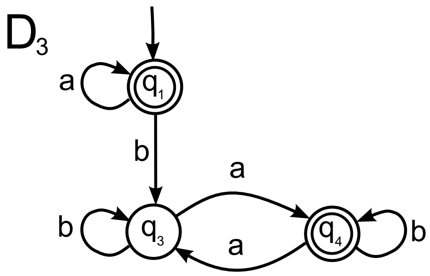


Figure 5–5: The minimal automaton for $L_3 = L_1 \cup L_2$ in Theorem 5.4.

S_{non} be the space spanned by states Q_{acc}, Q_{rej}, $Q_{acc} \cup Q_{rej}$, and $Q - Q_{acc} \cup Q_{rej}$ respectively.

Recall from chapter 3 that we require the density matrix formalism to describe the behavior of a GQFA machine taken over all probabilistic choices induced by the intermediate measurements. We will be using *weighted density matrices*,which are density matrices scaled by a factor $p \in [0, 1]$, to describe the GQFA state ρ_w on reading some input prefix w. The factor p in this case will represent the probability that the machine has not halted while processing the current prefix.

Let A_a be the mapping $\rho \mapsto \sum_i P_{a,i} U_a \rho U_a^\dagger P_{a,i}$, and let $A'_a = P_{non}(A_a\rho)P_{non}$. Then $A'_a\rho$ is a weighted density matrix such that $Tr(A'_a\rho) = p_a Tr(\rho)$ where p_a is the probability of not halting while reading a. Furthermore for $w = w_1 \dots w_n \in \Sigma^*$, we define $A'_w = A'_{w_n} \cdots A'_{w_1}$.

Lemma 5.5 *For every $w \in \Sigma^*$ there exists a pair E_1, E_2 of orthonormal subspaces of $\mathbb{C}^n$ such that $\mathbb{C}^n = E_1 \oplus E_2$ and for all weighted density matrices ρ over $\mathbb{C}^n$ we have:*

1. *If $supp(\rho) \subseteq E_1$, then $supp(A'_w\rho) \subseteq E_1$ and $Tr(A'_w\rho) = Tr(\rho)$.*
2. *If $supp(\rho) \subseteq E_2$, then $supp(A'_w\rho) \subseteq E_2$ and $lim_{k\to\infty} Tr((A'_w)^k\rho) = 0$.*

Proof: The proof proceeds as in [4]. We first show how to do this for the case that $\|w\| = 1$, and then we sketch how to extend it to arbitrary length words. Let $w = a$. We first construct the subspace E_1 of $\mathbb{C}^n$. E_2 will be the orthogonal complement of E_1. Let

$$E_1^1 = span(\{|\psi\rangle : Tr(A'_a|\psi\rangle\langle\psi|) = Tr(|\psi\rangle\langle\psi|)\}).$$

Equivalently, $E_1^1 = span\{|\psi\rangle : supp(A_a(|\psi\rangle\langle\psi|)) \subseteq S_{non}\}$ where S_{non} is the nonhalting subspace. We claim that $supp(\rho) \in E_1^1$ implies that $supp(A_a(\rho)) \in S_{non}$. By linearity it is sufficient to show this for $\rho = |\psi\rangle\langle\psi|$. Essentially, we need to show that the condition of $|\psi\rangle$ satisfying $Tr(A'|\psi\rangle\langle\psi|) = Tr(|\psi\rangle\langle\psi|)$ is closed under linear combinations. Suppose that $|\psi\rangle = \sum_j \alpha_j|\psi_j\rangle$, with $|\psi_j\rangle$ satisfying $supp(A_a(|\psi_j\rangle\langle\psi_j|)) \in S_{non}$ and $\sum_j |\alpha_j|^2 = 1$. Then:

$$\|\sum_i P_{halt}P_{a,i}U_a(\sum_j \alpha_j|\psi_j\rangle)\|^2 \leq \sum_{i,j} \|\alpha_j P_{halt}P_{a,i}U_a|\psi_j\rangle\|^2 = 0,$$

and thus $supp(A_a|\psi\rangle\langle\psi|) \in S_{non}$. Therefore, for mixed states ρ we have $supp(A_a\rho) \in S_{non}$ if and only if $supp(\rho) \in E_1^1$. For general $i > 2$, let:

$$E_1^i = span(\{|\psi\rangle : supp(A_a|\psi\rangle\langle\psi|) \in E_1^{i-1} \wedge Tr(A'_a|\psi\rangle\langle\psi|) = Tr(|\psi\rangle\langle\psi|)\}).$$

As before, for weighted density matrices ρ, we can interchange the condition $Tr(A'_a\rho) = Tr(\rho)$ for $supp(A_a\rho) \subseteq S_{non}$.

Observe that $E_1^i \subseteq E_1^{i+1}$ for all i. Since the dimension of each of these spaces is finite, there must be an i_0 such that $E_1^{i_0} = E_1^{i_0+j}$ for all $j > 0$. We define $E_1 = E_1^{i_0}$, and set E_2 to be the orthogonal complement of E_1.

It is clear that the first condition of the lemma is true for mixed states with support in E_1. For the second part, it will be sufficient to show the following claim:

Claim 7 *Let $j \in \{1, \ldots, i_0\}$. There is a constant $\delta_j > 0$ such that for any $|\psi\rangle \in E_2^j$ there is an $l \in \{0, \ldots, j-1\}$ such that $Tr(P_{halt} A_a (A'_a)^l (|\psi\rangle\langle\psi|)) \geq \delta_j$.*

Proof: We proceed by induction on j. Let $\mathcal{H} = \bigoplus_{k=1}^{m_a} \mathbb{C}^n$. Let $P_k : E_2^1 \to \mathcal{H}$ be the projector into the kth component of $\mathcal{H}$, and let $T_1 : E_2^1 \to \mathcal{H}$ be the function $T_1|\psi\rangle = \sum_k P_k P_{halt} P_{a,k} A_a |\psi\rangle$. Observe that $\|T_1|\psi\rangle\|^2$ is the probability of halting when a is read while the machine is in state $|\psi\rangle\langle\psi|$. By the previous discussion, $Tr(A'_a|\psi\rangle\langle\psi|) = 1 - \|T_1|\psi\rangle\|^2$. Define $\|T_1\| = min_{\| |\psi\rangle \|=1} \|T_1|\psi\rangle\|$. Note that the minimum exists since the set of unit vectors in $\mathbb{C}^n$ is a compact space. Also, let $\delta_1 = \|T_1\|^2$. Then $\delta_1 > 0$, otherwise there would be a vector $|\psi\rangle \in E_2^1$ such that $supp(A_a|\psi\rangle\langle\psi|) \in S_{non}$, a contradiction.

Now assume that δ_{j-1} has been found. We need to show that, for $|\psi\rangle \in E_2^j$, either a constant sized portion of $|\psi\rangle$ is sent into the halting subspace, or it is mapped to a vector on which we can apply the inductive assumption. We construct two functions $T_{j,halt}, T_{j,non} : E_2^j \to \mathcal{H}$ defined by:

$$\begin{aligned} T_{j,halt}|\psi\rangle &= \sum_{k=1}^{m_a} P_k P_{halt} P_{a,k} A_a |\psi\rangle, \\ T_{j,non}|\psi\rangle &= \sum_{k=1}^{m_a} P_k P_{E_2^{j-1}} P_{non} P_{a,k} A_a |\psi\rangle. \end{aligned}$$

Then the quantity $\|T_{j,halt}|\psi\rangle\|^2$ is the probability of halting while reading a, and $\|T_{j,non}|\psi\rangle\|^2 = Tr(P_{E_2^{j-1}} A'_a |\psi\rangle\langle\psi|)$. Note that for all vectors $|\psi\rangle \in E_2^j$ we must have either $\|T_{j,halt}|\psi\rangle\| \neq 0$ or $\|T_{j,non}|\psi\rangle\| \neq 0$, otherwise $|\psi\rangle$ is in E_1^j, a contradiction. This implies that $\|T_{j,non} \oplus T_{j,halt}\| > 0$. Note also that $\|T_{j,non} \oplus T_{j,halt}\| \leq 1$.

Define $\delta_j = \delta_{j-1}\frac{\|T_{j,non} \oplus T_{j,halt}\|^2}{2m_a}$. Take any unit vector $|\psi\rangle \in E_2^j$. Then $\|(T_{j,non} \oplus T_{j,halt})|\psi\rangle\| \geq \|T_{j,non} \oplus T_{j,halt}\|$. Recall that the range of $T_{j,non} \oplus T_{j,halt}$ is $\bigoplus_{k=1}^{m_a} \mathbb{C}^n \oplus \bigoplus_{k=1}^{m_a} \mathbb{C}^n$. In one of these subspaces, $(T_{j,non} \oplus T_{j,halt})|\psi\rangle$ has size at least $\frac{1}{\sqrt{2 \cdot m_a}}$. If it is in one of the last m_a subspaces, corresponding to $T_{j,halt}$ part, then there is nothing further to prove. Otherwise, assume that this component is in one of the subspaces corresponding to the $T_{j,non}$ part. In particular, there is a k such that $|\phi\rangle = P_{non}P_{a,k}A_a|\psi\rangle$ satisfies

$$\|P_{E_2^{j-1}}|\phi\rangle\|^2 \geq \frac{1}{2 \cdot m_a}.$$

We can split $|\phi\rangle$ into $|\phi_1\rangle + |\phi_2\rangle$, with $|\phi_i\rangle \in E_i^{j-1}$. By the inductive hypothesis, there is an $l < j-1$ such that $Tr(P_{halt}A_a(A'_a)^l(|\phi_2\rangle\langle\phi_2|)) \geq \delta_{j-1}Tr(|\phi_2\rangle\langle\phi_2|)$. Furthermore, the first condition of the lemma implies that for every choice of $(k_1, \ldots, k_l) \in [m^a]^l$,

$$P_{halt}P_{a,k_l}U_aP_{a,k_{l-1}}U_a \cdots P_{a,k_1}U_a|\phi_1\rangle = \vec{0}.$$

This implies that $Tr(P_{halt}A_a(A'_a)^l(|\phi_1\rangle\langle\phi_1|)) = 0$ and $Tr(P_{halt}A_a(A'_a)^l(|\phi_1\rangle\langle\phi_2|)) = Tr(P_{halt}A_a(A'_a)^l(|\phi_2\rangle\langle\phi_1|)) = 0$. Together, we obtain:

$$
\begin{aligned}
& Tr(P_{halt}A_a(A'_a)^l|\phi\rangle\langle\phi|) \\
= \; & Tr(P_{halt}(A'_a)^l(|\phi_1\rangle\langle\phi_1| + |\phi_1\rangle\langle\phi_2| + |\phi_2\rangle\langle\phi_1| + |\phi_2\rangle\langle\phi_2|)) \\
= \; & Tr(P_{halt}A_a(A'_a)^l(|\phi_1\rangle\langle\phi_1|)) + Tr(P_{halt}A_a(A'_a)^l(|\phi_1\rangle\langle\phi_2|)) \\
& \quad +Tr(P_{halt}A_a(A'_a)^l(|\phi_2\rangle\langle\phi_1|)) + Tr(P_{halt}A_a(A'_a)^l(|\phi_2\rangle\langle\phi_2|)) \\
= \; & Tr(P_{halt}A_a(A'_a)^l(|\phi_2\rangle\langle\phi_2|)) \geq \delta_{j-1}\frac{\|T_{j,non} \oplus T_{j,halt}\|^2}{2m_a}.
\end{aligned}
$$

This concludes the proof of the claim. □

Proposition 5.6 *Let U_a be the unitary transformation that is applied when a is read. Then $U_a = U_a^1 \oplus U_a^2$, where U_a^i acts unitarily on subspace E_i.*

Proof: By the unitarity of U_a, it is sufficient to show that $|\psi\rangle \in E_1$ implies $U_a|\psi\rangle \in E_1$. By definition of E_1, $|\psi\rangle \in E_1$ implies that all of the vectors $P_{a,i}U_a|\psi\rangle$ are in E_1. But $U_a|\psi\rangle = \sum_i P_{a,i}U_a|\psi\rangle$, and thus $U_a|\psi\rangle \in E_1$ since E_1 is a subspace. □

We are now ready to prove the second part of the lemma. We first show that $|\psi\rangle \in E_2$ implies $supp(A_a|\psi\rangle\langle\psi|) \subseteq E_2$. Let $|\psi'\rangle = U_a|\psi\rangle$. Then $A_a|\psi\rangle\langle\psi| = \sum_i |\psi_i\rangle\langle\psi_i|$, where $|\psi_i\rangle = P_{a,i}U_a|\psi\rangle$. Split $|\psi_i\rangle$ into vectors $|\psi_{i,1}\rangle + |\psi_{i,2}\rangle$, with $|\psi_{i,1}\rangle \in E_1$ and $|\psi_{i,2}\rangle \in E_2$. We claim that either $|\psi_{i,1}\rangle$ or $|\psi_{i,2}\rangle$ are trivial vectors. Suppose $\||\psi_{i,1}\rangle\| \neq 0$, and consider the intersection of the image of $P_{a,i}$ in the space spanned by $|\psi_{i,1}\rangle$ and $|\psi_{i,2}\rangle$. Now $|\psi_{i,1}\rangle$ implies that $U_a^{-1}|\psi_{i,1}\rangle \in E_1$ and thus $P_{a,i}|\psi_{i,1}\rangle \in E_1$, which implies $|\psi_i\rangle \in E_1$.

Now since each $|\psi_i\rangle$ satisfies $|\psi_i\rangle \in E_1$ or $|\psi_i\rangle \in E_2$, then we are done since the fact that the $|\psi_i\rangle$'s are orthonormal and sum to $U_a|\psi\rangle \in E_2$ implies that $|\psi_i\rangle \in E_2$ for all i. Thus, $|\psi\rangle \in E_2$ implies $span(A_a|\psi\rangle\langle\psi|) \subseteq E_2$.

To complete the proof of the second part of the lemma, for any ρ with $supp(\rho) \in E_2$, we can repeatedly apply Claim 7 to show that $Tr((A'_a)^k(\rho)) \to 0$ as $k \to \infty$.

To construct E_1 and E_2 for $w = w_1 \dots w_n$, we define $E_1^0 = S_{non}$ and E_1^k to be the set of all vectors $|\psi\rangle$ such that $Tr(A'_a|\psi\rangle\langle\psi|) = 1$ and $supp(A'_a|\psi\rangle\langle\psi|) \in E_1^{k-1}$, where $a = w_{k \mod n+1}$. We can then follow the proof as above. The proof of the first part of the theorem and of Claim 7 will generalize since the proof does not make use of the fact that the transformation and measurement defining E_1^j is the same as that of E_1^{j+1}. Proposition 5.6 will apply to w_i for all i. □

The ergodic-transient lemma can be extended in the following way:

Lemma 5.7 *Let M be an n-state GQFA over alphabet Σ, and let $x, y \in \Sigma^*$. Then there exists a pair E_1, E_2 of orthonormal subspaces of $\mathbb{C}^n$ such that $\mathbb{C}^n = E_1 \oplus E_2$ and for all weighted density matrices ρ over $\mathbb{C}^n$ we have:*

1. *If $supp(\rho) \subseteq E_1$, then for all $w \in (x \cup y)^*$, $supp(A'_w\rho) \subseteq E_1$, and $Tr(A'_w\rho) = Tr(\rho)$.*
2. *If $supp(\rho) \subseteq E_2$, then $supp(A'_w\rho) \subseteq E_2$ and for all $\varepsilon > 0$ there exists a word $w \in (x \cup y)^*$ such that $Tr(A'_w\rho) \leq \varepsilon$.*

Proof: Let E_1^w be the subspace constructed as in Lemma 5.5. Let $E_1 = \cap_{w\in(x\cup y)^*} E_1^w$, and let E_2 to be the orthogonal complement of E_1.

Suppose that $supp(\rho) \subseteq E_2$. If there is a $w \in (x \cup y)^*$ such that $supp(\rho) \subseteq E_2^w$, we can directly apply the argument from the previous lemma to show that $Tr((A'_w)^j\rho) \to$

0 as $j \to \infty$. However such a w may not exist so a stronger argument is necessary. As the application of an A'_w transformation can only decrease the trace of ρ, for any ε there exists a $t \in (x \cup y)^*$ such that for all $w \in (x \cup y)^*$, $Tr(A'_t\rho) - Tr(A'_{tw}) \leq \varepsilon$. For all i let t_i be a such a string for $\varepsilon = \frac{1}{2^i}$. Consider the sequence $\rho_1, \rho_2, \ldots$ defined by $\rho_i = A'_{t_i}\rho$. The set of weighted density matrices form a compact, closed space with respect to the trace metric, and so this sequence of must have a limit point ρ.

We claim that $Tr(\rho) = 0$. Suppose not. The support of ρ is in E_2, so there must be some word $w \in (x \cup y)^*$ such that $Tr(A'_w\rho) < Tr(\rho)$. This contradicts the assumption that ρ is a limit point. □

Finally we note a very simple fact that will allow us to extend impossibility results for LQFA to GQFA:

Fact 1 *Let M be a GQFA. Let E_1 be the subspace defined as in Lemma 5.7, and suppose that the state of the machine ρ on reading the ¢ character satisfies supp$(\rho) \in E_1$. Then there is an LQFA M' such that, for all $w \in (x \cup y)^*$ the state of M on reading w is isomorphic to the state of M' on reading w.*

We are now ready to apply these technical results to prove several fundamental properties of GQFAs.

5.3 Results

Recall that the nonclosure of KWQFA under union was shown in [6]. Using the results of the previous section, we can follow a similar proof outline to show nonclosure of GQFA under union. Our first result gives a necessary condition for recognition by GQFAs.

Theorem 5.8 *If the minimal automaton for L contains the forbidden construction for Theorem 5.3, then L cannot be recognized by GQFA with probability $p > \frac{1}{2}$.*

Proof: Suppose that M recognizes L containing the forbidden construction of Theorem 5.3 with probability $p > \frac{1}{2}$. By closure under left quotient, we can assume that the state q_0 in the forbidden construction is also the initial state of the minimal automaton for L.

Let ρ be the initial state of the machine and let $\rho_w = A'_w \rho$. The basic outline of the proof is that we will use Lemma 5.7 and Theorem 3.13 to find two words $w_1 \in x(x \cup y)^*$, $w_2 \in y(x \cup y)^*$ such that ρ_{w_1} and ρ_{w_2} have similar output behavior. We then analyze the acceptance probabilities of the words $w_1 z_1$, $w_1 z_2$, $w_2 z_1$, and $w_2 z_2$ to arrive at a contradiction.

Let E_1 and E_2 be subspaces which meet the conditions of Lemma 5.7 with respect to x and y. We claim that for all $\varepsilon > 0$ there exists $u, v \in (x \cup y)^*$ such that $\|Tr(P_{E_1}\rho_{xu} - P_{E_1}\rho_{yv})\|_t \leq \varepsilon$. Suppose to the contrary that there exists $\varepsilon > 0$ such that $\|Tr(P_{E_1}\rho_{xu} - P_{E_1}\rho_{yv})\|_t > \varepsilon$ for all u, v. Then by Fact 1, there exists an LQFA that can recognize the language $x(x \cup y)^*$ with bounded error, contradicting Theorem 3.13. Let $\delta = p - \frac{1}{2}$, and choose $\varepsilon = \frac{\delta}{4}$.

By Lemma 5.7, for all ε' we can find $u' \in (x \cup y)^*$ such that $Tr(P_{E_2}\rho_{xuu'}) < \varepsilon'$. Furthermore we can find $v' \in (x \cup y)^*$ such that $Tr(P_{E_2}\rho_{xuu'v'}) < \varepsilon'$ and $Tr(P_{E_2}\rho_{yvu'v'}) < \varepsilon'$. Let $w_1 = xuu'v'$ and $w_2 = yvu'v'$, and choose $\varepsilon' = \frac{\delta}{4}$

Let $p_{i,acc}$ ($p_{i,rej}$) be the probability with which M accepts (rejects) while reading w_i. Furthermore let $q_{ij,acc}$ (resp $q_{ij,rej}$) be the probability that M accepts if the state of the machine is ρ_{w_1} and the string $z_j\$$ is read. Since $\|\rho_{w_1} - \rho_{w_2}\|_t \leq \|\rho_{xu} - \rho_{yv}\|_t =$

$\frac{\delta}{2} \leq \varepsilon$, $q_{1j,acc}$ (and likewise $q_{1j,rej}$) can be different from $q_{2j,acc}$ by a factor of at most $\frac{\delta}{2}$. As a consequence, one of the words w_1z_1, w_1z_2, w_2z_1, or w_2z_2 must not be classified correctly. Suppose, for instance that w_1z_1, w_1z_2, and w_2z_1 are classified correctly. Since $q_{11,rej}$ differs from $q_{21,rej}$ by a factor of at most $\frac{\delta}{2}$, the fact that w_1z_1 is accepted and w_2z_1 is rejected implies that $p_{2,rej} > p_{1,rej} + \delta$. since $q_{12,rej}$ differs from $q_{22,rej}$ by at most a factor of $\frac{\delta}{2}$, will be rejected with probability greater than $1 - p$, a contradiction. The other cases are similar. □

Corollary 5.1 *GQFA are not closed under union.*

Proof: The languages L_1 and L_2 in Theorem 5.4 were shown to be recognizable by KWQFAs, thus they can also be recognized by GQFA. On the other hand, the minimal automaton of $L_1 \cup L_2$, contains the forbidden construction of Theorem 5.8. □

By the argument of Theorem 3.3 in [6], the nonclosure of GQFAs under union immediately implies that there exists languages L which are recognized with probability $\frac{1}{2} \leq p < 1$ but not with probability arbitrarily close to 1. Using a technique similar to that of Theorem 5.8, we can give a forbidden construction which implies an upper bound of 2/3 on the acceptance probability. The constructions of language L_1 and L_2 in Theorem 5.4 shows that this value is tight.

Theorem 5.9 *If the minimal DFA M_L for L contains states q_1, q_2, q_3, words x, y, z_1, z_2 such that $\delta(x, q_1) = \delta(x, q_2) = q_2$, $\delta(y, q_1) = \delta(y, q_2) = q_3$.,*

- *reading x while in q_1 brings you to q_2,*
- *reading y while in q_1 brings you to q_3,*
- *reading x or y while in q_2 or q_3 brings you back to the same state,*

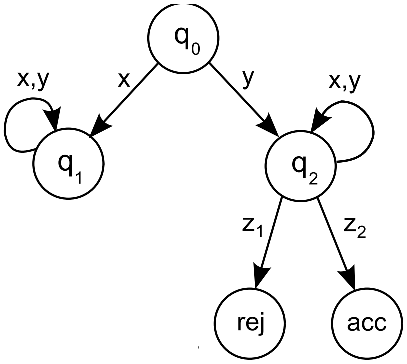


Figure 5–6: The forbidden construction of Theorem 5.9.

- *q_3 is not all accepting or all-rejecting.*

Then L cannot be recognized by GQFA with probability $p > \frac{2}{3}$.

Proof: Since $q_2 \neq q_3$ and by closure under complement, without loss of generality, there exists a word z_3 such that $xz_3 \in L$ and $yz_3 \notin L$. We also assume that q_1 is the initial state. Let ρ be the state of M after reading the ¢ symbol. As in Lemma 5.7, split $\mathbb{C}^n$ into subspaces E_1 and E_2 with respect to x and y.

For all ε, we can find $w_1 \in x(x \cup y)^*$ and $w_2 \in x(x \cup y)^*$ such that $\|\rho_{w_1} - \rho_{w_2}\|_t \leq \varepsilon$, $Tr(P_{E_2}\rho_w) < \varepsilon$, $Tr(P_{E_2}\rho_w) < \varepsilon$. let p_i be the probability that M rejects while reading w_i and w_i respectively, and let q_{i3} be the probability of rejecting when M is in state w_i and reads z_3. By setting ε, the difference between q_{13} and q_{23} can be made arbitrarily small, so $p_1 + q_{13} \leq 1/3$ and $p_1 + q_{23} \geq 1/3$ imply that $p_3 - p_1 > 1/3$. This further implies that M accepted while reading w_1 with probability greater than $1/3$, contradicting the assumption that $w_1 z_1$ is rejected with probability at least p. □

Comparing KWQFA and GQFA, we see that both types of QFA can recognize any language whose syntactic monoid is in **BG**. This comes from the fact that KWQFA and hence GQFA can recognize boolean combinations of languages recognized by BPQFA [3]. However GQFA can use the LQFA constructions to achieve a much higher probability of correctness. While GQFAs can recognize L with $M(L) \in \mathbf{BG}$ with probability $1 - \varepsilon$ for all $\varepsilon > 0$, the maximum probability of correctness for recognizing L by a KWQFA construction diminishes rapidly as the complexity of L increases. This shows that allowing arbitrary intermediate measurements can make QFAs more expressive.

However, we see a different picture if we consider languages L such that $M(L) \notin \mathbf{BG}$. In this case we see that the known lower bounds for KWQFA recognizing languages with syntactic monoid outside of **BG** apply in almost exactly the same way to GQFA. This suggests that the arbitrary intermediate measurements of GQFA do not help for recognizing these languages. It is an interesting open problem to determine whether or not there exists languages which can be recognized by GQFA but not KWQFA. The results of this chapter bring us much closer to answering this question.

CHAPTER 6
MOQFA Succinctness

So far in this thesis, we have seen many results which emphasize the fact that quantum finite automata are less expressive than deterministic finite automata. In this chapter, we will see that QFAs can outperform DFAs in the sense that they can recognize languages using much fewer states than the equivalent DFA. The literature contains several results regarding the asymptotic behavior of the size of MOQFA versus the size of the minimal deterministic automata for certain language classes. For example, an early QFA result [4] showed that, for languages of the form $L_p = \{w : |w| \mod p = 0\}$ for prime p, the size of the smallest MOQFA is $O(\log p)$.

In this chapter, we will be considering the asymptotic behaviour, for a given class $\mathcal{W}$ of languages (not necessarily a variety), the size of the smallest MOQFA recognizing $L \in \mathcal{W}$ with probability p as a function of the size of the minimal automaton and p. While we know exactly which languages can be recognized by MOQFA, it is not yet understood which classes of languages can be recognized succinctly. We will give an overview of known results in this area and outline our recent work.

It is important to note that, in many relevant computational settings, it is not possible to obtain storage improvements by moving from classical to quantum memory. As a consequence of Holevo's Theorem, we know that the outcome of any measurement on an n-qubit quantum state will have entropy at most n, which implies that at most n bits of information can be reliably stored in a system of

n qubits. Ambainis et al. [7] showed that even in the simpler case of encoding a classical bit string so that a single but arbitrarily chosen bit can be extracted with high probability, quantum bits are only modestly more efficient than classical bits.

We will focus our attention on the size of MOQFAs recognizing the *word problem* L_G over a group G. Let G^* be the free monoid with alphabet G. For $w \in G^*$ we define $eval(w)$ to be the product of the group elements. Then we define $L_G = \{w : eval(w) = 1\}$. The word problem is good candidate for the exponential succinctness property in light of the following results. Bertoni, Mereghetti, and Palano [16] showed that the word problem over $\mathbb{Z}_n^h$ can be recognized by an MOQFA with probability $p = \frac{1}{3}(2 - \delta)$ using $O(\frac{1}{\delta^2} \log |\mathbb{Z}_n^h|) = O(\frac{1}{\delta^2} h \log n)$ states. On the other hand, there exists a class of languages L such that each $M(L)$ is a cyclic, and the size of the smallest MOQFA is $\Omega(\sqrt{\frac{|M(L)|}{\log |M(L)|}})$ [14].

The succinct constructions in [4, 16] involve Fourier techniques. In Section 6.1, we suggest a strategy for generalizing these constructions by moving to general abelian and nonabelian Fourier transforms. In Section 6.1.3 we highlight a key obstacle to generalizing Fourier techniques beyond abelian groups. Nevertheless, we show that there are exponentially succinct constructions for some interesting classes of nonabelian groups, such as the class of dihedral groups.

Let us denote by $D(L)$ the size of the minimal automaton for L, and by $D_Q(L, \delta)$ the size of the smallest MOQFA recognizing L with probability $\frac{1}{2} + \delta$. We denote by L_G the word problem over a group G. In this case, we have $D(L_G) = |M(L_G)| = |G|$. Known results suggest that $D_Q(L, \delta)$ depends upon the algebraic properties of G. To prove lower bounds on $D_Q(L_G)$ for groups with specific algebraic properties, we need to argue that algebraic properties of G imply particular structural properties of any MOQFA M recognizing L_G. One way to do this is to consider the algebraic properties of the mapping $\mu_M : \Sigma^* \to GL(n)$ from input words to state transition operators. In Section 6.2, we look at techniques for normalizing the transformations of a given MOQFA so that they have desirable properties (in particular, so that $\mu_M(\Sigma^*)$ is a finite group with similar algebraic properties to G).

In related work, the question of automata size has been considered for randomized finite automata, and exponentially succinct probabilistic automata have been constructed for certain languages [30, 2]. It has been shown that some of these constructions can be adapted to the case of KWQFA [4]. On the other hand, the L_p languages mentioned earlier require $\Omega(\sqrt{p})$ states to be recognized by probabilistic automata.

Some work has been done to understand the space complexity of MOQFA in the special case of languages such that $M(L)$ is a cyclic group. It was shown that $D_Q(L) = O(\sqrt{|D(L)|})$ for this class of languages [43]. Furthermore, there is an infinite sequence of these languages which satisfy $D_Q(L_n) = \Omega(\sqrt{\frac{n}{\log n}})$ [14], whereas the word problem over $\mathbb{Z}_n$ requires only $O(\log n)$ states.

The literature contains a number of papers which use algebra to obtain results about MOQFAs. These papers use the fact that there is a natural metric on the set of unitary matrices for which the closure $\overline{\mu(\Sigma^*)}$ of $\mu(\Sigma^*)$ forms a *compact group* [24]. This property has been used to show the decidability of certain questions regarding MOQFA [66, 17], and to give a simple proof of the fact that languages recognized by MOQFA with bounded error are regular [17].

6.1 Succinct Constructions

Before we begin, we note that QFAs are only able to recognize languages more succinctly than DFAs when the probability p of correctness is strictly less than 1. Suppose for example that M recognizes the word problem over G with probability 1, and let $|\psi_g\rangle$ be the state reached on reading g. For all $h \neq g$, there would a measurement which would distinguish $|\psi_h\rangle$ from $|\psi_g\rangle$ with certainty, which further implies that $|\psi_h\rangle \perp |\psi_g\rangle$. Since there are $|G|$ pairwise orthogonal states, the dimension of the state space and hence the number of QFA states must be at least $|G|$. So, for the case of $p = 1$, the natural construction is optimal.

It is helpful to adopt the following point of view on MOQFAs. For an MOQFA M over Σ^*, define $\eta_M : \Sigma^* \to [0, 1]$ so that $\eta_M(w)$ is the probability that M accepts w. We call η_M the *characteristic function* of M. In general we will call any function $f : \Sigma^* \to [0, 1]$ a *probability assignment*.

All of the known exponentially succinct constructions for MOQFAs have a similar basic construction. The goal is to approximate a target probability assignment. First, a collection of small MOQFAs inducing probability assignments are defined, with each of these small MOQFAs tracking a different aspect of the syntactic group. Then a probabilistic argument shows that there is a small subset of this set such that, for all inputs words, a significant fraction of the probability assignments from the subset will agree with the target function. Finally, a MOQFA is built from this subset to recognize the desired language.

The following well-known construction can be used to take a weighted sum of the outputs of several MOQFA:

Lemma 6.1 *Let $M_1, \ldots, M_k$ be MOQFA such that M_i is of size n_i and has characteristic function η_i. Then for any set $\{p_1, \ldots, p_n\}$ of positive reals satisfying $\sum_i p_i = 1$, there is a MOQFA M of size $\sum_i n_i$ with characteristic function given by:*

$$\eta(w) = \sum_i p_i \eta_i(w).$$

Proof: The strategy will be to run the k machines in parallel, and choose the final measurement so that the output of machine i is given with probability p_i. We define the state set of M to be the disjoint union of the state sets Q_i of the machines M_i. This naturally partitions the state space into the direct sum $\bigoplus_i \mathbb{C}^{n_i}$. Let us denote by $U_{\sigma,i}$ the unitary operator corresponding to reading the letter σ on the ith machine. We define $A_{¢}$ of M to be the matrix that, for all i, places $\sqrt{p_i}$ amplitude in the state corresponding to the initial state of M_i, and then applies the operation $\bigoplus_i U_{¢,i}$. For $\sigma \in \Sigma \cup \{\$\}$, we define the operator $A_\sigma = \bigoplus_i A_{\sigma,i}$. We choose the set F of final states to be the disjoint union of the final sets F_i. Let $|\psi_{w\$}\rangle$ be the state of M after reading w and the \$ symbol. The total length of the component of $|\psi_{w\$}\rangle$ in the accepting subspaces is $(\sum_i(\sqrt{p_i}\sqrt{\eta_i(w)})^2)^{1/2}$, so the characteristic of M is $\sum_i p_i \eta_i(w)$ as required. □

We will be interested in approximating probability assignments in the following sense:

Definition 6.1 *We say that probability assignment f δ-approximates the assignment g if $|f(w) - g(w)| \leq \delta$ for all $w \in W$. If M is such that $\eta_M = f$ we say M δ-approximates g.*

If the characteristic function η of M 0-approximates the assignment g, we say that M *simulates* g.

6.1.1 Abelian Case

We adopt the following notation for complex numbers. We define $\Theta(z) = \frac{z}{|z|}$. Also, let $real[z]$ ($comp[z]$) be the real (complex) component of z. Finally, let $\phi(z) = \frac{real[z]}{|z|}$ be the *phase* of z. Recall that $z = |z|e^{i\phi(z)}$, and $z = |z|(\cos(\phi(z)) + i\sin(\phi(z)))$ by Euler's formula.

Theorem 6.2 *For $G \in \mathbf{Ab}$ and $\delta > 0$, the word problem over G is recognizable by an MOQFA with probability $p = \frac{1}{3}(2-\delta)$ using $O(\frac{1}{\delta^2}\log|G|)$ states.*

Proof: This is a straightforward generalization of the construction for recognizing $\mathbb{Z}_n^h$ in [16]. The key difference is that we show that we can replace the Fourier transform over $\mathbb{Z}_n^h$ with the general abelian Fourier transform. The construction in [16] is in turn a natural generalization of the succinctness proof for $\mathbb{Z}_p$ in [4].

We would like to construct an MOQFA which succinctly approximates the probability assignment $\eta_G : \Sigma^* \to [0,1]$ defined by:

$$\eta_G(w) = \begin{cases} 1 & \text{if } eval(w) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

To show that L_G is recognized with probability $p = \frac{1}{3}(2-\delta)$, it is sufficient to show the following:

Lemma 6.3 *For $G \in \mathbf{Ab}$, there exists a MOQFA construction M such that the probability assignment $\eta' = \frac{1}{2} + \frac{1}{2}\eta$ can be δ-approximated by an $O(\frac{1}{\delta^2}\log|G|)$ state QFA.*

This construction can be modified to recognize the word problem over G with probability $p = \frac{1}{3}(2 - \delta)$ as follows. Let η_0 be the characteristic function defined by $\eta_0(w) = 0$ for all $w \in \Sigma^*$. This is simply the characteristic function of an MOQFA which always rejects its input. We can construct a 1-state MOQFA with this property. Using Lemma 6.1 we can construct a machine M' using $|M| + 1$ states with characteristic function $\frac{2}{3}\eta_M + \frac{1}{3}\eta_0$. One can easily check that a machine with this characteristic will recognize the word problem over G with probability p. It remains to prove the lemma.

Proof: As in [15], the first step of the succinct QFA construction is to express the function η in terms of the Fourier basis. We define $f : G \to [0, 1]$ as:

$$f(g) = \begin{cases} 1 & \text{if } g = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Let $\hat{f}$ be the Fourier transform of f. Also define $\|\hat{f}\|_1 = \sum_\chi |\hat{f}(\chi)|$. Let $w \in \Sigma^*$ and let $eval(w) = g$. Finally, let $n = |G|$. Then we can express $\eta(w)$ as:

$$\begin{aligned}
\eta(w) = f(g) &= \frac{1}{n}\sum_{\chi} \hat{f}(\chi)\cdot\chi(g^{-1}) \\
&= \frac{1}{n}\sum_{\chi} real[\hat{f}(\chi)\cdot\chi(g^{-1})] \\
&= \frac{1}{n}\sum_{\chi} \cos(\phi(\hat{f}(\chi)\cdot\chi(g^{-1}))) \\
&= \frac{1}{n}\sum_{\chi} \left(2\cos^2\left(\phi(\hat{f}(\chi)\cdot\chi(g^{-1}))/2\right)-1\right) \\
&= \frac{1}{n}\sum_{\chi} 2\cos^2\left(\phi(\hat{f}(\chi)\cdot\chi(g^{-1}))/2\right)-1.
\end{aligned}$$

And so:

$$\frac{1}{n}\sum_{\chi}\cos^2\left(\phi(\hat{f}(\chi)\cdot\chi(g^{-1}))/2\right) = \frac{1}{2}+\frac{1}{2}\eta(w).$$

We claim that the left hand side is a weighted sum of probability assignment which can each be computed by MOQFAs of small size. For a fixed $\chi \in \hat{G}$ let $\eta_\chi(w) = \cos^2(\phi(\hat{f}(\chi)\cdot\chi(g^{-1}))/2)$, where $g = eval(w)$. This assignment can be simulated by a QFA M_χ of size 2. Recall that complex numbers of unit magnitude correspond exactly to 2×2 matrices of the form:

$$A(z) = \begin{bmatrix} real[z] & -comp[z] \\ comp[z] & real[z] \end{bmatrix} = \begin{bmatrix} \cos(\phi(z)) & -\sin(\phi(z)) \\ \sin(\phi(z)) & \cos(\phi(z)) \end{bmatrix}.$$

We set the initial state of M_χ to be $\left[\cos(\phi(\hat{f}(\chi))/2), \sin(\phi(\hat{f}(\chi)/2)\right]$, and for all $g \in G$ we define $A_g = A(\phi(\chi(g^{-1}))/2)$. Then if we choose the accepting state to be the space spanned by the first coordinate, M_χ will accept w with probability $\eta_\chi(w)$.

In the remainder of the proof, we show that the assignment $\eta' = \frac{1}{n}\sum_\chi \eta_\chi$ can be δ-approximated by the assignment induced by a MOQFA with $O(\log n/\delta^2)$ states. We exactly follow the proof of Theorem 3 in [15]. In particular, this shows that the theorem is not restricted to groups of the form $\mathbb{Z}_n^h$.

We will need Hoeffding's inequality, which states that for set S of *i.i.d.* random variables over $[0, 1]$ with expected value μ:

$$P\left[\left|\frac{1}{|S|}\sum_{s\in S} s - \mu\right| \geq \delta\right] \leq 2e^{-2\delta^2|S|}.$$

Let $S \subseteq \hat{G}$ and let η_S be the assignment $\frac{1}{|S|}\sum_{s\in S}\eta_s$. By Lemma 6.1, this assignment can be simulated by an MOQFA using $O(|S|)$ states. Note that the value of $\eta'(w)$ and $\eta_S(w)$ depend only on $eval(w)$. Using Hoeffding's inequality we find that:

$$\begin{aligned} & P\left[\sup_{w\in G^*}\{\eta_S(w) - \eta'(w)\} \geq \delta\right] \\ \leq\; & n \cdot P\left[\sup_{g\in G}\{\eta_S(g) - \eta'(g)\} \geq \delta\right] \\ \leq\; & n2e^{-2\delta^2|S|}. \end{aligned}$$

We select $|S|$ to be $O(\log n/\delta^2)$, so that $n2e^{-2\delta^2|S|} < 1$ for suitably large $|S|$ and so there exists a suitable choice of S so that η_S will δ-approximate η'. □

In the hope of extending these results to nonabelian groups, it is natural to look at *representations*, which are the analogous structure to abelian group characters.

6.1.2 Representation Theory

We will now recall some basic facts about the representation theory of finite groups. A fuller treatment can be found in [31, 58]. The *general linear group* of dimension n over field $\mathbb{F}$ is the group of invertible linear operators on an $\mathbb{F}$-vector space of dimension n. We will denote by $GL(n)$ the general linear group over $\mathbb{C}^n$.

A *representation* of a group G is a group morphism $\rho : G \to GL(n)$. The morphism ρ will satisfy $\rho(1) = I$ and there exists an inner product of $\mathbb{C}^n$ for which each $\rho(g)$ will be unitary. A subspace W of $\mathbb{C}^n$ is said to be *stable* under ρ if $\rho(g)W \subseteq W$ for all g. If W is a strict subset of $\mathbb{C}^n$, it can be shown that the orthogonal complement $\overline{W}$ of W will also be stable under ρ, and so ρ can be decomposed into a direct sum $\rho_1 \oplus \rho_2$ of two representations acting independently on subspaces of $\mathbb{C}^n$. In this case we say that ρ is *reducible.* If no such decomposition is possible we say that ρ is an *irreducible representation* or an *irrep.* The *dimension of an irrep* is the dimension of the space on which the irrep operates. We say that two representations ρ_1 and ρ_2 are isomorphic if there exists a linear transformation A such that $A\rho_1(g) = \rho_2(g)A$ for all $g \in G$.

For a representation ρ, the *character of* ρ is the function $\chi_\rho : G \to \mathbb{C}$ defined by $\chi_\rho(g) = Tr(\rho(g))$. For functions of the form $f : G \to \mathbb{C}$ let us define an inner product $(f, f') = \frac{1}{|G|} \sum_g f(g)\overline{f'(g)}$.

Theorem 6.4 *Let τ be a representation and let $\tau = \tau_1 \oplus \cdots \oplus \tau_m$ be a decomposition of τ into irreducible representations. If the irreducible representation ρ occurs k times within the decomposition of τ, then $(\chi_\tau, \chi_\rho) = k$.*

In particular, the number of occurrences of an irrep within a representation is independent of the choice of decomposition.

Representations occur naturally in the construction of MOQFAs. For instance, in the construction of Theorem 3.3, the mapping $\varphi : G \rightarrow GL(n)$ defined by $\varphi(g) = A_g$ is a representation. Likewise, the construction in Section 6.1.1 induces a representation. In fact the construction Theorem 3.3 corresponds to a special canonical representation called the *regular representation.* The regular representation is of particular interest since all of the information about the structure of a finite group's irreducible representations can be extracted from the regular representation and it's associated character.

The Regular Representation

Fix a group G with $|G| = n$. Suppose that we define an orthonormal basis $\{e_g\}_{g \in G}$ of $\mathbb{C}^n$. let ρ be the representation over $GL(n)$ defined by $\rho(g)e_{g'} = e_{g'g}$. We call this the *regular representation* of G. By considering the character of the regular representation we can obtain some fundamental properties of representations.

Theorem 6.5 *If ρ is an irreducible representation of dimension d and τ is the regular representation then $(\chi_\tau, \chi_\rho) = d$.*

Since the regular representation contains an occurrence of each irreducible representation, it follows that there are at most n irreducible representations of a group of size n. Furthermore if $\rho_1, \ldots, \rho_k$ are a complete set of irreducible representations and the dimension of ρ_i is d_i, then $\sum_i d_i^2 = n$. With respect to an appropriately chosen basis, the regular representation will be a block diagonal matrix with submatrices of size $d_1, \ldots, d_k$. In fact we can think of the Fourier transform of a group G as a function which maps the canonical basis $e_1, \ldots, e_n$ of the regular representation to one in which the regular representation is block diagonal.

For the special case of abelian groups, all of the representations have dimension one. This implies that there are exactly n representations of an abelian group of size n, and in this case the characters are equal to the representations.

6.1.3 Nonabelian Case

In this section, we will discuss a few ideas for generalizing Theorem 6.2 to the nonabelian case. We first recall a few facts regarding the discrete Fourier transform for nonabelian groups.

Let G be a finite group, and let us denote by $\hat{G} = \{\rho_1, \ldots, \rho_k\}$ be the set of irreducible representations of G, and for all i let d_i be the dimension of representation ρ_i. Let $f : G \to \mathbb{C}$. Then the Fourier transform of f is a function which maps each irrep ρ_i to $GL(i)$ as follows:

$$\hat{f}(\rho_i) = \sum_g f(g)\rho_i(g).$$

In the case of abelian groups the ρ_i matrices are dimension 1 and we obtain the abelian Fourier transform as a special case. Fix an arbitrary basis for the matrices and let us denote by $[M]_{j\ell}$ the $j\ell$th coordinate of a matrix. It can be shown by an application of Schur's lemma [58] that for all i, j, ℓ the set of functions $h_{i,j,\ell} : G \to \mathbb{C}$ defined by $h_{i,j,l}(g) = [\rho_i(g)]_{j\ell}$ are linearly independent and form an orthogonal basis with respect to the inner product $(h, h') = \sum_{g\in G} h(g)h'(g^{-1})$. Furthermore, the following inversion formula holds:

$$f(g) = \sum_i \frac{d_i}{n} Tr(\hat{f}(\rho_i)\rho_i(g^{-1})).$$

Let us apply this idea in order to express η_G (c.f. Section 6.1.1) as a linear combination of probability assignments.

$$\begin{aligned}
\eta_G(w) = f(g) &= \sum_i \frac{d_i}{n} Tr(\hat{f}(\rho_i)\rho_i(g^{-1})) \\
&= \sum_{i,j,\ell} \frac{d_i}{n} [\hat{f}(\rho_i)]_{j\ell}[\rho_i(g^{-1})]_{\ell j} \\
&= \sum_{i,j,\ell} \frac{d_i}{n} real[[\hat{f}(\rho_i)]_{j\ell}[\rho_i(g^{-1})]_{\ell j}].
\end{aligned}$$

As in the previous case, this gives us a linear combination of functions $f_{i,j,\ell} : G \to \mathbb{C}$. It is tempting to think that these functions can be computed by a small sized QFA. In this case, we could prove the existence of a small MOQFA recognizing G.

It is possible to construct an MOQFA of size $2 \cdot i$ for which one of the coordinates of the final state vector has value $real[[\hat{f}(\rho_i)]_{j\ell}[\rho_i(g^{-1})]_{\ell j}]$. This is a simple matter of maintaining the vector $\rho_i(w)e_j$ as the state of the machine (e_j being the basis vector for coordinate e_j), and then updating according to the rule $\rho_i(wa)e_j = \rho_i(a)\rho_i(w)e_j$. The factor two is required to separate the real component of each vector from the complex coordinate. At the end, the amplitude in the ℓth coordinate is equal to the desired value.

This does not solve the problem however, as the measurement outcome will be the norm of this value squared. This was overcome in the abelian case by taking the square root of each $\chi_i(g)$ individually. This cannot work in the nonabelian setup as the square root function on unitary matrices does not distribute.

While the technique of [16] cannot be immediately applied to nonabelian groups, there are indeed classes of nonabelian groups which can be recognized succinctly. Certainly, it is immediate from [16] that the set of all groups of the form $\mathbb{Z}_n \times S_3$, where S_3 is the group of permutations of three elements, can be recognized using $O(\log n)$ states. The next lemma can be used to show that there are more interesting classes of nonabelian groups which can be recognized succinctly:

Lemma 6.6 *Suppose that G is a semidirect product of S and T, with $|T| = t$, and suppose that there exists an MOQFA $M = (Q, q_0, \Sigma, \{U_\sigma\}, F)$ recognizing the word problem over S with probability $p \geq \frac{1}{2}$ using m states. Then there exists an MOQFA M' recognizing the word problem over G with probability p using $m \cdot |T|$ states.*

Proof: Let $*$ be the left action associated with the semidirect product of S and T, so that $(s_1, t_1)(s_2, t_2) = (s_1 \cdot_S (t_1 * s_2), t_1 \cdot_T t_2)$.

Let $Q \times T$ be the set of states of M'. We split the state space is into $|T|$ subspaces of size m, each corresponding to the subset of states $Q \times \{t\}$ for some fixed state. The initial state of M' will be $|q_0, 1\rangle$. For $t \in T$, let R_t be the unitary matrix that, that maps $|q_i, t'\rangle$ to $|q_i, t't\rangle$. This is just a permutation of the basis elements, so R_t is unitary. On input (s, t), M' will apply the matrix $R_t \bigoplus_{t' \in T} U_{t' \cdot_G s}$, where each $U_{t' \cdot_G s}$ acts on the subspace $Q \times \{t'\}$. The set of final states of M' is $F \times \{1\}$.

The transitions are constructed so that all of the amplitude will be contained in one of the $|T|$ subspaces. After $w = (s_1, t_1) \ldots (s_n, t_n)$ is read, the amplitude will be in the subspace corresponding to $eval(t_1 \ldots t_n)$. Furthermore, it is easy to show by induction that the state within that subspace will be exactly the that would be reached by M upon reading a string of elements of S that evaluates to

$$s_1 \cdot_S (t_1 * s_2) \cdot_S \cdots \cdot_S ((t_1 \cdot_T \cdots \cdot_T t_{n-1}) * s_n).$$

Suppose that M' has read w such that $eval(w) = (s, t)$. If $t \neq 1$, then the state after reading w will be orthogonal to the accepting subspace of M'. Now consider the case that $t = 1$. The state within the subspace $Q \times t$ corresponds to a state reached by M after reading a sequence of elements that evaluates to s, and thus M' will accept w with probability p if $s = 0$, otherwise reject with probability at least p. □

Let us now give some examples of nonabelian groups which can be recognized succinctly using this lemma. First, we consider the dihedral group D_n, which is the set of reflections and rotations of an n-gon. This is isomorphic to a semidirect product of $S = \mathbb{Z}_n$ with $T = \mathbb{Z}_2$, where the left action defined by:

$$t \cdot s = \begin{cases} s & \text{if t = 0} \\ \text{-s} & \text{if t= 1.} \end{cases}$$

Note that this is indeed an abelian group, since $(0,1)(s,0) = (-s,1)$, while $(s,0)(0,1) = (s,1)$. In this case S has an MOQFA of size $O(\log n)$ and $|T| = 2$, so D_n also has a QFA of size $O(\log n)$. In general.

Corollary 6.1 *For any fixed group T, let S_T be the set of all semidirect products of A and T such that $A \in \mathbf{Ab}$. Then for any $\delta > 0$, the word problem over group $G \in S_T$ can be recognized by an MOQFA with probability of correctness $p = \frac{1}{3}(2-\delta)$ using $O(\frac{1}{\delta^2}\log|A|)$ states.*

This includes the set of groups of the form $A \times T$ as a special case.

6.2 Algebraic Structure of MOQFAs

It is likely that, for many groups, there are nontrivial lower bounds on size of MOQFAs recognizing the word problem. We would like to begin an investigation into lower bound results for these machines.

Suppose that MOQFA M recognizes L_G. Without any further condition, we can say very little about the algebraic structure of $\mu_M(G^*)$. However, we can say that the metric closure of $\mu_M(G^*)$ forms a *compact group* [24]. A natural metric d can be defined on $\mu_M(G^*)$ as $d(X,Y) = \min_{|\psi\rangle} \|(X-Y)|\psi\rangle\|$. Then the closure $\overline{\mu_M(G^*)}$ of $\mu_M(G^*)$ with respect to this metric is a compact group. Using characterizations of compact groups, this fact has been used to construct algorithms for several decision problems related to MOQFA [66, 17].

However, we will need stronger properties to prove lower bounds on specific groups, since properties of G do not immediately extend to $\overline{\mu(G^*)}$. For at least some cases, this is possible. The following result was implicitly stated in [17]. We prove it here in order to discuss generalizations to less restricted classes of groups.

Theorem 6.7 *If MOQFA M recognizes the word problem over $\mathbb{Z}_n$, then there is an MOQFA M' of the same size as M such that $\mu_{M'}(\Sigma^*)$ is a finite cyclic group which has $\mathbb{Z}_n$ as a divisor.*

Proof: Let M be an MOQFA recognizing $L_{\mathbb{Z}_n}$ with probability $p > \frac{1}{2}$. Let $A = \mu(1)$, where 1 is the generator of $\mathbb{Z}_n$. We can assume for $i \in \mathbb{Z}_n$ that $\mu(i) = A^i$, for if not we can construct a machine M' with this property with the same size as M, and necessarily M' will also recognize $L_{\mathbb{Z}_n}$ with probability p.

Recall again that $A = \sum_j \lambda_j |\phi_j\rangle\langle\phi_j|$ by the spectral theorem. Let $\theta_j \in [0, 1)$ be the unique number such that $e^{2\pi i\theta_j} = \lambda_j$. If θ_j is rational we will say that λ_j is rational, otherwise we say λ_j is irrational. We view each θ_j as an element of $\mathbb{R}\backslash\mathbb{Z}$.

A collection of reals $\xi_1, \ldots, \xi_k$ is linearly independent in $\mathbb{Q}$ if there are no set of rationals $q_1, \ldots, q_k$ such that $\sum_i q_i\xi_i = 0$. We now recall Kronecker's theorem.

Theorem 6.8 *Let $(\xi_1, \ldots, \xi_k) \in \mathbb{R}^k/\mathbb{Z}^k = T$ (T is the k-dimensional torus), and let $T' = \{j(\xi_1, \ldots, \xi_k) : j \in \mathbb{N}\}$. If $\xi_1, \ldots, \xi_k$ are rational then T' is a finite set. Otherwise, T' forms a dense subset of T whose metric closure is a subtorus of T. Finally, if $\xi_1, \ldots, \xi_k$ are irrational and linearly independent then the closure of T' is T.*

Consider the case where all eigenvalues are irrational. Then the set $R = \{\mu(w)|q_0\rangle : w \in Z_n^*\}$ of *reachable* points will be such that, for every pair $|\psi_1\rangle, |\psi_2\rangle \in R$ and for all $\epsilon > 0$, there are a sequences of vectors $|v_1\rangle, \ldots, |v_m\rangle$ such that $|v_1\rangle = |\psi_1\rangle$, $|v_m\rangle = |\psi_2\rangle$, and $\langle v_i|v_{i+1}\rangle \geq 1 - \varepsilon$. This implies that M cannot recognize $\mathbb{Z}_m$ with bounded probability.

Now suppose that there are some rational eigenvalues. Then there is some n' such $\lambda_i^{n'} = 1$ for every rational λ_i. Let us consider the output probabilities on input $1^{z+bn'}$ for fixed $z < n'$. Let $P_{acc} = \sum_i |p_i\rangle\langle p_i|$. Let $|\psi_\ell\rangle = P_{acc}A^\ell|q_0\rangle$.

$$\begin{aligned} PA^\ell|q_0\rangle &= (\sum_i |p_i\rangle\langle p_i|)(\sum_j \lambda_j^\ell|\phi_j\rangle\langle\phi_j|)|q_0\rangle \\ &= \sum_{i,j} \lambda_j^\ell|p_i\rangle\langle p_i|\phi_j\rangle\langle\phi_j|q_0\rangle \\ &= \sum_{i,j} \lambda_j^\ell c_{ij}|p_i\rangle, \end{aligned}$$

Where c_{ij} is the complex number $\langle p_i|\phi_j\rangle\langle\phi_j|q_0\rangle$ So then:

$$\langle\psi_\ell|\psi_\ell\rangle = \sum_{i,j,j'} \overline{\lambda_{j'}^\ell}\,\overline{c_{ij'}}\lambda_j^\ell c_{ij}.$$

Now consider this sum for $\ell = z + bn'$ for growing b. For the rational λ_j's, the quantity $\lambda_j^{z+bn'}$ will be constant. I claim that for all fixed z either $\{1^{z+bn'} : b \in \mathbb{N}\}$ are each accepted with probability at least p or rejected with probability at least p. Suppose not. Then A must contain irrational eigenvalues, otherwise $\{\langle\psi_\ell|\psi_\ell\rangle : \ell \in z + bn'\}$ would be a singleton, implying that all of $\{1^{z+bn'} : b \in \mathbb{N}\}$ are accepted with the same probability. By Kronecker's theorem the metric closure of the set $T' = \{j(\theta_1, \ldots, \theta_m)\}$ is a connected torus which implies that there is some b for which $1 - p \leq |\{\langle\psi_\ell|\psi_\ell\rangle : \ell \in z + bn'\}|^2 \leq p$, a contradiction. If we replace the irrational θ_is with 0 we obtain a limit point of T', so we can replace the irrational λ_js with the value 1 and still get a machine which recognizes $\mathbb{Z}_n$ with bounded error.

This machine M' will be such that $\mu(\mathbb{Z}_n^*)$ forms a finite cyclic group whose order divides n. □

This naturally raises the question of what other kinds of normalization results we can obtain. The proof suggests that the irrational eigenvalues in a transformation are not useful in recognizing the word problem, so they may be eliminated. However, this intuition has yet to be made formal for the word problem over general groups. It seems that, at the least Theorem 6.7 should be extendable to abelian groups in the following sense:

Conjecture 6.1 *If MOQFA M recognizes the word problem over an abelian group G, then there is an MOQFA M' of the same size as M such that $\mu_{M'}(\Sigma^*)$ is a finite commutative group which has G as a divisor.*

It would be sufficient to show that for two noncommuting operations A and B we can construct modified operations A' B' which do commute. We may be able to do this by looking at the commutator of A and B. Suppose that w is a word which is accepted by M with probability at least p. Then for all i, measuring $(ABA^{-1}B^{-1})^i|\psi_w\rangle$ will cause the machine to accept with probability at least p. This seems to suggest that there is no space advantage to be gained allowing A and B to be noncommutative.

Suppose we could show in general that for the word problem over any group G we can normalize $\mu_M(\Sigma^*)$ to a finite $\mu_{M'}(\Sigma^*)$. This would mean that $\mu_{M'}(\Sigma^*)$ would be a representation of some group G' for which G is a divisor, and so the structure of $\mu_{M'}(\Sigma^*)$ would depend on the structure of the irreducible representations of G. It seems likely that the size of the smallest MOQFA recognizing the word problem over

G is related to the size of the largest irreducible representations. Supporting this idea is the fact that, for every class of groups which is known to have $O(\log |G|)$ sized MOQFAs, there is a constant upper bound on the size of the irreducible representations for this class.

It would also be interesting to determine whether we can tighten the normalization result. While we have shown that we can normalize the transformations in the cyclic case so that $\mu_M(\Sigma^*)$ is finite and cyclic, it is possible that $\mu_M(\Sigma^*)$ is a much larger group than $\mathbb{Z}_n$. We wonder if this is an artifact of the proof, or if we can construct smaller MOQFAs by choosing $\mu_M(\Sigma^*)$ to be a larger group.

CHAPTER 7
Conclusion

In this thesis, we have shown that one can successfully apply techniques from algebraic automata theory to prove meaningful results about QFAs. Our investigation has identified the languages whose syntactic monoid is in **BG** as central. We have seen that this class of languages corresponds exactly to the class of languages recognized by LQFA, and to the boolean closure of languages recognized by BPQFA. This implies that both KWQFA and GQFA can also recognize every language with syntactic monoid in **BG**. Furthermore, we have shown that for KWQFA and GQFA there are strong impossibility results for languages whose syntactic monoid is outside of **BG**.

We have made considerable progress in characterizing the class of languages recognized by BPQFA. Again, the algebraic perspective has helped us to identify the most relevant possibility and impossibility results to work on. We have left open the problem of the exact characterization of BPQFA. The key missing link seems to be in our incomplete understanding of the language variety corresponding to $\mathbf{Nil}^+ \circledM \mathbf{J_1}$. It is an interesting open problem to obtain a combinatorial characterization of this class.

We have developed a number of technical results regarding QFAs which highlight the structure which can be found in QFAs on the condition that they recognize certain languages. We have shown that in the case of BPQFAs, the transformations for letters which map to idempotents in the syntactic monoid can be ‘normalized’ so that they have special structure, and we have shown that cyclic languages recognized by MOQFAs can be normalized so that they generate a finite group. We have seen that the transition functions for a GQFA, as in the case of KWQFA, can be decomposed into an ergodic and transient component, and this result identifies a key limit to the power granted by halting before the end.

Our difficulty in characterizing KWQFA and GQFA may be due to the lack of good tools for characterizing language classes which are not closed under union. In the thesis we recalled how the Eilenberg theory can be generalized to deal with nonclosure under complement. There has been recent work successfully extending the Eilenberg theory to language classes which are closed under a restricted class of inverse morphisms [53], so perhaps there is an algebraic way to deal with language classes which are not closed under boolean operations.

Finally, there is much more work to do in the area of MOQFA succinctness. On one hand, the lower bounds on MOQFA size for recognizing the word problem are very limited. While it is not expected, it is still possible that, for the class of all finite groups $\mathbf{G}$, we can recognize the word problem over $G \in \mathbf{G}$ in $O(\log |G|)$ states. On the other hand, there are classes of ‘nearly’-abelian groups for which we are not known to have space efficient constructions, such as the class of nilpotent groups of class two. In recent work we have identified the nonabelian groups of order pq for prime p, q, $p < q$, as being good candidates for proving lower bounds. These groups have a relatively simple structure, having a semidirect product decomposition and having only two generators, yet these groups have large representations and they are non-nilpotent. A nontrivial lower bound for these groups would give us considerable insight into the succinctness question.

REFERENCES

[1] Dorit Aharonov, Andris Ambainis, Julia Kempe, and Umesh Vazirani. Quantum walks on graphs. In ACM, editor, *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing: Hersonissos, Crete, Greece, July 6–8, 2001*, pages 50–59, New York, NY, USA, 2001. ACM Press.

[2] Andris Ambainis. The complexity of probabilistic versus deterministic finite automata. In *Proceedings of the 7th International Symposium on Algorithms and Computation*, 1996.

[3] Andris Ambainis, Martin Beaudry, Marats Golovkins, Arnolds Kikusts, Mark Mercer, and Denis Thérien. Algebraic results on quantum automata. *Theory of Computing Systems*, 38:165–188, 2006.

[4] Andris Ambainis and Rusins Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations. In *39th Annual Symposium on Foundations of Computer Science*, pages 332–341. IEEE Computer Society Press, 1998.

[5] Andris Ambainis and Arnolds Ķikusts. Exact results for accepting probabilities of quantum automata. *Theoretical Computer Science*, 295(1–3):3–25, February 2003.

[6] Andris Ambainis, Arnolds Ķikusts, and Māris Valdats. On the class of languages recognizable by 1-way quantum finite automata. In *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science*, volume 2010 of *Lecture Notes in Computer Science*, pages 75–86, 2001.

[7] Andris Ambainis, Ashwin Nayak, Amnon Ta-Shma, and Umesh Vazirani. Dense quantum coding and quantum finite automata. *Journal of the ACM*, 49(4):496–511, July 2002.

[8] D. Barrington and D. Thérien. Finite monoids and the fine structure of NC^1. In *Proceedings of the 19th ACM STOC*, pages 101–109, 1987.

[9] P. Benioff. Quantum mechanical Hamiltonian models of discrete processes that erase their own histories: Application to turing machines. *International Journal of Theoretical Physica*, 21(3/4):177–201, 1982.

[10] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and development 6*, pages 525–532, 1973.

[11] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.*, 70, 1993.

[12] C.H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers Systems and Signal Processing, Bangalore India*, pages 175–179, December 1984.

[13] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal of Computing*, 26(5):111–1473, 1997.

[14] Alberto Bertoni, Carlo Mereghetti, and Beatrice Palano. Lower bounds on the size of quantum automata accepting unary languages. In *8th Italian Conference on Theoretical computer science ICTCS. Bertinoro, Italy,*, 2003.

[15] Alberto Bertoni, Carlo Mereghetti, and Beatrice Palano. Quantum computing: 1-way quantum automata. In *Developments in Language Theory*, volume 2710 of *Lecture Notes in Computer Science*. Springer, 2003.

[16] Alberto Bertoni, Carlo Mereghetti, and Beatrice Palano. Small size quantum automata recognizing some regular languages. *Theoretical Computer Science*, 340:394–407, 2005.

[17] Alberto Bertoni, Carlo Mereghetti, and Beatrice Palano. Some formal tools for analyzing quantum automata. *Theoretical Computer Science*, 356:14–25, 2006.

[18] Garrett Birkhoff. On the structure of abstract algebras. In *Proceedings of the Cambridge Philosophical Society*, volume 31, pages 433–454, 1935.

[19] S. L. Bloom. Varieties of ordered algebras. *Journal of Computer and Systems Sciences*, 51, 1976.

[20] Alex Brodsky and Nicholas Pippenger. Characterizations of 1-way quantum finite automata. *SIAM Journal on Computing*, 31(5):1456–1478, October 2002.

[21] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

[22] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of ACM STOC'71*, pages 151–158, 1971.

[23] S. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64(1–3):2–21, 1985.

[24] H. Derksen, E. Jeandel, and P. Koiran. Quantum automata and algebraic groups. *Journal of Symbolic Computation (special issue on the occasion of MEGA 2003)*, 39(3-4):357–371, 2005.

[25] David Deutsch. Quantum, theory, the Church-Turing principle, and the universal quantum computer. *Proceedings of the Royal Society of London Series A*, 400:97–117, 1985.

[26] David Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London Series A*, 425:73–90, 1989.

[27] Samuel Eilenberg. *Automata, Languages, and Machines.* Academic Press, New York, 1 edition, 1976.

[28] Richard Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 1982.

[29] Ed Fredkin and Tommaso Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21:219–253, 1982.

[30] R. Freivalds. On the growth of the number of states in result of determinization of probabilistic finite automata. *Automatic Control and Computer Sciences*, 3:39–42, 1982.

[31] William Fulton and Joe Harris. *Representation Theory: A First Course.* Springer, 1991.

[32] J. Gill. Computational complexity of probabilistic turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.

[33] Marats Golovkins and Jean-Éric Pin. Varieties generated by certain models of reversible finite automata. In *Proceedings of COCOON 2006*, volume 4112 of *Lecture Notes in Computer Science*, pages 83–93, 2006.

[34] G.H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers.* Oxford University Press, 1960.

[35] A. S. Holevo. Statistical problems in quantum physics. In *proceedings of the second Japan-USSR Symposium on Probability Theory*, volume 330 of *Lecture Notes in Mathematics*, pages 104–119. Springer-Verlag, 1973.

[36] John Hopcroft and Jeffrey Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley Publishing, Reading Massachusetts, 1979.

[37] Alexi Kitaev and John Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 608–617, 2000.

[38] Attila Kondacs and John Watrous. On the power of quantum finite state automata. In *38th Annual Symposium on Foundations of Computer Science*, pages 66–75. IEEE, IEEE Computer Society Press, 20–22 October 1997.

[39] M. Koucký, P. Pudlak, and D. Thérien. Bounded-depth circuits: Separating gates from wires. In *Proceedings of the 37th ACM STOC Converence*, pages 257–265, 2005.

[40] K. Krohn and J. Rhodes. The algebraic theory of machines. *Transactions of the American Mathematical Society*, pages 450–464, 1965.

[41] Rolf Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(183–191), 1961.

[42] R. McNaughton and S. Papert. *Counter-Free Automata.* MIT Press, Cambridge, Mass., 1971.

[43] Carlo Mereghetti and Beatrice Palano. On the size of one-way quantum finite automata with periodic behaviors. *Theoret. Inf. Appl.*, 36:277–291, 2002.

[44] Cris Moore and Jim Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 237(1-2):275–306, 2000.

[45] Ashwin Nayak. Optimal lower bounds for quantum automata and random access codes. In *40th Annual Symposium on Foundations of Computer Science (FOCS '99)*, pages 369–377, Washington - Brussels - Tokyo, October 1999. IEEE.

[46] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information.* CUP, 2000.

[47] Christos Papadimitriou. *Computational Complexity.* Addison Wesley, 1994.

[48] Jean-Éric Pin. *Varieties of Formal Languages.* North Oxford Academic Publishers, United Kingdom, 1986.

[49] Jean-Éric Pin. On the language accepted by finite reversible automata. In Thomas Ottmann, editor, *Automata, Languages and Programming, 14th International Colloquium*, volume 267 of *Lecture Notes in Computer Science*, pages 237–249, Karlsruhe, Germany, 13–17 July 1987. Springer-Verlag.

[50] Jean-Éric Pin. BG=PG, a success story. In John Fountain, editor, *NATO Advanced Study Institute Semigroups, Formal Languages, and Groups*, pages 33–47. Kluwer Academic Publishers, 1995.

[51] Jean-Éric Pin. A variety theorem without complementation. *Russian Mathematics*, 39:80–90, 1995.

[52] Jean-Éric Pin. *Handbook of formal languages*, volume I, pages 679–746. Springer, 1996.

[53] Jean-Éric Pin and Howard Straubing. Some results on C-varieties. *Theoret. Informatics Appl.*, 39:239–262, 2005.

[54] Jean-Éric Pin and Pascal Weil. A reiterman theorem for pseudovarieties of finite first-order structures. *Algebra Universalis*, 35:577–595, 1996.

[55] Jean-Éric Pin and Pascal Weil. Semidirect products of ordered semigroups. *Communications in Algebra*, 30:146–149, 2002.

[56] Jean-Éric Pin and Pascal Weil. The wreath product principle for ordered semigroups. *Communications in Algebra*, 30:5677–5713, 2002.

[57] Marcel-Paul Schützenberger. On finite monoids having only trivial subgroups. *Information and control*, 8:190–194, 1965.

[58] Jean-Pierre Serre. *Representation Theory of Finite Groups.* Springer-Verlag, 1977.

[59] Peter W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*. IEEE press, 1994.

[60] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:2493–2496, 1995.

[61] Imre Simon. Piecewise testable events. In *2nd GI Conference on Automata Theory and Formal Languages*, pages 214–222, 1975.

[62] Howard Straubing. The wreath product and its applications. In Jean-Éric Pin, editor, *Formal properties of finite automata and applications*, volume 386 of *Lecture Notes in Computer Science*, pages 15–24, 1989.

[63] Pascal Tesson and Denis Thérien. Logic meets algebra: the case of regular languages. *Logical Methods in Computer Science*, 3(1):1–37, 2007.

[64] Tommaso Toffoli. Bicontinuous extensions of invertible combinatorial functions. *Mathematical Systems Theory*, 14:13–23, 1981.

[65] Alan Turing. On computable numbers, with an application to the entscheidungsproblem. In *Proceedings of the London Mathematical Society*, volume 42 of *2*, pages 230–265, 1936.

[66] P. Koiran V. Blondel, E. Jeandel and N. Portier. Decidable and undecidable problems about quantum automata. *SIAM Journal on Computing*, 34(6):1464–1473, 2005.

[67] Lieven Vandersypen, Matthias Steffan, Gregory Breyta, Costantino Yannoni, Mark Sherwood, and Isaac Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883–887, 2001.

[68] John Watrous. Succinct quantum proofs for properties of finite groups. In *Proceedings of the IEEE FOCS*, pages 537–546, 2000.

[69] Andy Yao. Quantum circuit complexity. In *Proceedings of the 36th annual FOCS*, pages 352–361, 1993.

Index

Index

Appendix A

$\mathbf{G}$	Groups
$\mathbf{Ab}$	Abelian groups
$\mathbf{A}$	Aperiodic monoids
$\mathbf{J}$	$\mathcal{J}$-trivial monoids
$\mathbf{R}$	$\mathcal{R}$-trivial monoids
$\mathbf{Nil}$	Nilpotent semigroups
$\mathbf{J}_1$	Semilattices
$\mathbf{BG}$	Block groups
$\mathbf{V} * \mathbf{W}$	Wreath product
$\mathbf{V} \mathbin{\textcircled{M}} \mathbf{W}$	Malcev product

Other Symbols

$\mathbb{R}, \mathbb{C}$	real, complex numbers
$supp(\rho)$	Support of ρ
$Tr(\rho)$	Trace of ρ
$S(\rho)$	Von Neumann Entropy of ρ
$H(p), H(X)$	Shannon entropy
$\equiv_L$	Syntactic congruence
$M(L)$	Syntactic monoid
$\preceq$	Division (of monoids)
$\mathcal{J},\ \mathcal{H},\ \mathcal{R},\ \mathcal{L}$	Green's relations