Algebraic Results on Quantum Automata

Andris Ambainis[1], Martin Beaudry[2], Marats Golovkins[1], Arnolds Ķikusts[1],
Mark Mercer[3], and Denis Thérien[3]

[1] Institute of Mathematics and Computer Science, University of Latvia,
Raiņa bulv. 29, Riga, Latvia *
ambainis@lanet.lv, marats@latnet.lv, arnolds@usa.com

[2] Département de Mathématiques et d'Informatique 2500, Boul. Université,
Sherbrooke (PQ) J1K 2R1, Canada **
beaudry@DMI.USherb.ca

[3] School of Computer Science, McGill University, 3480 rue University, Montréal
(PQ), H3A 2A7, Canada **
{jmerce1,denis}@cs.mcgill.ca

Abstract. We use tools from the algebraic theory of automata to investigate the class of languages recognized by two models of Quantum Finite Automata (QFA): Brodsky and Pippenger's end-decisive model, and a new QFA model whose definition is motivated by implementations of quantum computers using nucleo-magnetic resonance (NMR). In particular, we are interested in the new model since nucleo-magnetic resonance was used to construct the most powerful physical quantum machine to date. We give a complete characterization of the languages recognized by the new model and by Boolean combinations of the Brodsky-Pippenger model. Our results show a striking similarity in the class of languages recognized by the end-decisive QFAs and the new model, even though these machines are very different on the surface.

1 Introduction

In the classical theory of finite automata, it is unanimously recognized that the algebraic point of view is an essential ingredient in understanding and classifying computations that can be realized by finite state machines, i.e. the regular languages. It is well known that to each regular language L can be associated a canonical finite monoid (its syntactic monoid, $M(L)$) and unsurprisingly the algebraic structure of $M(L)$ strongly characterizes the combinatorial properties of L. The theory of pseudo-varieties of Eilenberg (which in this paper will be called **M**-varieties for short) provides an elegant abstract framework in which these correspondences between monoids and languages can be uniformly discussed.

* Research supported by Grant No. 01.0354 from the Latvian Council of Science; European Commission, contract IST-1999-11234. Also for the fourth and fifth authors, the University of Latvia, Kristaps Morbergs fellowship.

** Research supported by NSERC and FCAR.

Finite automata are a natural model for classical computing with finite memory, and likewise *quantum finite automata* (QFA) are a natural model for quantum computers that use a finite dimensional state space as memory. Quantum computing's more general model of *quantum circuits* [19] gives us an upper bound on the capability of quantum machines, but the fact that several years have passed without the construction of such circuits (despite the efforts of many scientists) suggests that the first quantum machines are not going to be this strong. Thus it is not only interesting but practical to study simpler models alongside of the more general quantum circuit model.

There are several models of QFA [17, 15, 8, 5, 9, 7] which differ in what quantum measurements are allowed. The most general model (independently [9] and [7]) allows any sequence of unitary transformations and measurements. The class of languages recognized by this model is all regular languages. In contrast, the model of [17] allows unitary transformations but only one measurement at the end of computation. The power of QFAs is then equal to that of permutation automata [17, 8] (i.e. they recognize exactly group languages). In intermediate models [15, 8, 5], more than one measurement is allowed but the form of those measurements is restricted. The power of those models is between [17] and [9, 7] but has not been characterized exactly, despite considerable effort [4, 2]. The most general model of QFAs describes what is achievable in principle according to laws of quantum mechanics while some of the more restricted models correspond to what is actually achieved by current implementations of quantum computers.

In view of the enduring success of the algebraic approach to analyze classical finite state devices, it is natural to ask if the framework can be used in the quantum context as well. The work that we present here answers the question in the affirmative. We will analyze two models of QFA: the model [8] and a new model whose definition is motivated by the properties of nucleo-magnetic resonance (NMR) quantum computing. Among various physical systems used to implement quantum computing, liquid state NMR has been the most successful so far, realizing quantum computers with up to 7 quantum bits [26]. Liquid state NMR imposes restrictions of what measurements can be performed, and the definition of the new model reflects this. In both cases we are able to provide an algebraic characterization for the languages that these models can recognize. It turns out that the class of languages recognized by these two models coincide almost exactly (that is, up to Boolean combinations), which is quite surprising considering the differences between the two models (for example, the NMR model allows mixed states while the [8] model does not). It is a pleasant fact that the **M**-variety that turns up in analyzing these QFA is a natural one that has been extensively studied by algebraists. Besides using algebra, our arguments are also based on providing new constructions to enlarge the class of languages previously known to be recognizable in these models, as well as proving new impossibility results using subspace techniques (as developed in [4]), information theory (as developed in [18]), and quantum Markov chains (as developed in [3]). In particular, we show that the Brodsky-Pippenger model cannot recognize the language $a\Sigma^*$ (it is already known [15] that $\Sigma^* a$ is not recognizable), and that our new quantum model cannot recognize $a\Sigma^*$ or $\Sigma^* a$.

The paper is organized as follows. In Section 2 we give an introduction to the algebraic theory of automata and we define the models. In the next two sections we give our results on the two models we introduced, and in the last section we outline some open problems.

2 Preliminaries

2.1 Algebraic theory of automata

An **M**-variety is a class of finite monoids which is closed under taking submonoids, surjective homomorphisms, and direct products. Given an **M**-variety **V**, to each finite alphabet Σ we associate the class of regular languages $\mathcal{V}(\Sigma^*) = \{L \subseteq \Sigma^* : M(L) \in \mathbf{V}\}$. It can be shown that $\mathcal{V}(\Sigma^*)$ is a Boolean algebra closed under quotients (i.e. if $L \in \mathcal{V}(\Sigma^*)$ then for all $w \in \Sigma^*$ we have $w^{-1}L = \{x : wx \in L\} \in \mathcal{V}(\Sigma^*)$ and $Lw^{-1} = \{x : xw \in L\} \in \mathcal{V}(\Sigma^*)$) and inverse homomorphisms (i.e. if $\varphi : \Sigma^* \to \Sigma^*$ is a homomorphism and $L \in \mathcal{V}(\Sigma^*)$, then $\varphi^{-1}(L) \in \mathcal{V}(\Sigma^*)$). Any class of languages satisfying these closure properties is called a $*$-variety of languages. A theorem of Eilenberg [10] says that there is a 1-1 correspondence between **M**-varieties and $*$-varieties of languages: a driving theme of the research in automata theory has been to find explicit instantiations of this abstract correspondence.

The **M**-variety that plays the key role in our work is the so-called block groups [20], classically denoted **BG**. This variety is ubiquitous: it appears in topological analysis of languages [20], in questions arising in the study of non-associative algebras [6] and in constraint satisfaction problems [14]. It can be defined by the following algebraic condition: M is a block group iff for any $e = e^2$ and $f = f^2$ in M, $eM = fM$ or $Me = Mf$ implies $e = f$. For any language L, $M(L)$ is a block group iff L is a Boolean combination of languages of the form $L_0 a_1 L_1 \ldots a_k L_k$, where each $a_i \in \Sigma$ and each L_i is a language that can be recognized by a finite group: this class of languages is the largest $*$-variety that does not contain $a\Sigma^*$ or $\Sigma^* a$ for arbitrary alphabet satisfying $|\Sigma| \geq 2$ [20].

2.2 Models

We adopt the following conventions. Unless otherwise stated, for any machine M where these symbols are defined, Q is the set of classical states, Σ is the input alphabet, q_0 is the initial state, and $Q_{acc} \subseteq Q$ ($Q_{rej} \subseteq Q$) are accepting (rejecting) states. If Q_{acc} and Q_{rej} are defined then we require $Q_{acc} \cap Q_{rej} = \emptyset$. Also, each model in this paper uses distinct start and endmarkers, ¢ and \$ respectively. On input w, M processes the characters of ¢w\$ from left to right.

Let $|Q| = n$. For all QFA in this paper, the state of the machine M is a *superposition* of the n classical states. Superpositions can be expressed mathematically as vectors in $\mathbb{C}^n$. For each $q \in Q$ we uniquely associate an element of the canonical basis of $\mathbb{C}^n$, and we denote this element $|q\rangle$. Now the superposition can be written as the vector $\sum_{q_i \in Q} \alpha_i |q_i\rangle$. We say α_i is the *amplitude* with which we are in state q_i. We now require each such vector to have an l_2 norm of 1, where the l_2 norm $\| \sum \alpha_i |q_i\rangle \|_2$ of $\sum \alpha_i |q_i\rangle$ is $\sqrt{\sum |\alpha_i|^2}$. Superpositions are also sometimes called *pure states.* There are also cases where the quantum state of the machine is a random variable; in other words, the state is a 'classical' probability distribution of superpositions $\{(p_i, \psi_i)\}$, each ψ_i with probability p_i. In this case we say the system is in a *mixed state.* Mixed states can be expressed in terms of *density matrices* [19], and these are usually denoted ρ.

A *transformation* of a superposition is a linear transformation with respect to a unitary matrix. $A \in \mathbb{C}^{n \times n}$ is called unitary if $A^* = A^{-1}$, where A^* is the Hermitian conjugate of A and is obtained by taking the conjugate of every element in A^T. Unitary transformations are length preserving, and they are closed under product. A set $\{A_\sigma\}$ of transformations is defined for each machine, with one transformation for each $\sigma \in \Sigma \cup \{¢, \$\}$.

An outside observer cannot gain a priori information about the state of a quantum mechanical system except through a *measurement* operation. A measurement of a superposition ψ probabilistically projects ψ onto exactly one of j prespecified disjoint subspaces $E_1 \oplus \cdots \oplus E_j$ spanning $\mathbb{C}^n$. The index of the selected subspace is communicated to the outside observer. For all i, let P_i be the projection operator for E_i. Then the probability of projecting into E_i while measuring $E_1 \oplus \cdots \oplus E_j$ is $\|P_i \psi\|_2^2$.

We will consider two modes of acceptance. For a probabilistic machine M, we say M recognizes L with *bounded (two-sided) error* if M accepts any $w \in L$ and rejects any $w \notin L$ with probability at least p, where $p > \frac{1}{2}$. We say M recognizes L with *bounded positive one-sided error* if any $w \in L$ is accepted with probability $p > 0$ and any $w \notin L$ is rejected with probability 1.

Liquid state NMR is the technique used to implement quantum computing on 7 quantum bits [26]. NMR uses nuclei of atoms as quantum bits, and the state of the machine is a molecule in which 7 different atoms can be individually adressed. One of features of NMR is that quantum transformations are simultaneously applied to a liquid containing 10^{21} molecules. Thus, we have the same quantum computation carried out by 10^{21} identical quantum computers. Applying a measurement is problematic, however. On different molecules, the measurement can have a different result. We can determine the fraction of molecules that produce each outcome, but we cannot separate the molecules by the measurement outcome. Because of that, the operations performed cannot be conditional on the outcome of a measurement. On the other hand, measurements which do not affect the next transformation are allowed. This situation is reflected in the definition of our new model, given below:

Latvian QFA (LQFA). A superset of this model has been studied in [18, 5]. A LQFA is a tuple $M = (Q, \Sigma, \{A_\sigma\}, \{P_\sigma\}, q_0, Q_{acc})$ such that $\{A_\sigma\}$ are unitary matrices, and $\{P_\sigma\}$ are measurements (each P_σ is defined as a set $\{E_1, \ldots, E_j\}$ of orthogonal subspaces). We define $Q_{rej} = Q \backslash Q_{acc}$ and we require that $P_\$$

is a measurement w.r.t. $E_{acc} \oplus E_{rej}$, where $E_{acc} = span\{Q_{acc}\}$ and $E_{rej} = span\{Q_{rej}\}$. Let ψ be the current state. On input σ, $\psi' = A_\sigma \psi$ is computed and then measured w.r.t. P_σ. After processing the \$, the state of M will be in either E_{acc} or E_{rej} and M accepts or rejects accordingly. The acceptance mode for LQFA is bounded error. This model is introduced as QRA-M-C in the classification of QFAs introduced in [12].

Also in [12], a probabilistic automata model related to LQFA was introduced, which they called '1-way probabilistic reversible C-automata' (we abbreviate this to PRA). A PRA is a tuple $M = (Q, \Sigma, \{A_\sigma\}, q_0, Q_{acc})$, where each A_σ is a *doubly stochastic* matrix. A matrix is doubly stochastic if the sum of the elements in each row and column is 1. The acceptance mode for PRA is bounded error. The two models are related in the following way: If M is a LQFA such that each P_σ measures with respect to $\bigoplus_{q\in Q} span\{|q\rangle\}$ for every $\sigma \in \Sigma$, then M can be simulated by a PRA. Conversely, a PRA can be simulated by a LQFA if each A_σ of the PRA has a *unitary prototype* [12]. A matrix $U = [u_{ij}]$ is a unitary prototype for $S = [s_{ij}]$ if for all i,j: $|u_{i,j}|^2 = s_{i,j}$. When S has a unitary prototype it is called unitary stochastic [16]. This relationship between LQFA and PRA is helpful in proving that certain languages are recognized by LQFA.

Brodsky-Pippenger QFA (BPQFA). The BPQFA model is a variation on the model introduced by Kondacs and Watrous [15] (we will call this model KWQFA). A KWQFA is defined by a tuple $M = (Q, \Sigma, \{A_\sigma\}, q_0, Q_{acc}, Q_{rej})$ where each A_σ is unitary. The state sets Q_{acc} and Q_{rej} will be halt/accept and halt/reject states, respectively. We also define $Q_{non} = Q \backslash (Q_{acc} \cup Q_{rej})$ to be the the set of nonhalting states. Lastly, for $\mu \in \{acc, rej, non\}$ we define $E_\mu = span\{Q_\mu\}$, and P_μ to be the projection onto E_μ. Let ψ be the current state of M. On input σ the state becomes $\psi' = A_\sigma \psi$ and then ψ' is measured w.r.t. $E_{acc} \oplus E_{rej} \oplus E_{non}$. If after the measurement the state is in E_{acc} or E_{rej}, M halts and accepts or rejects accordingly. Otherwise, ψ' was projected into E_{non} and M continues. We require that after reading \$ the state is in E_{non} with probability 0. The acceptance mode for KWQFA is bounded error.

The BPQFA model is one of several variations introduced by Brodsky and Pippenger in [8], which they called 'end-decisive with positive one-sided error'. A BPQFA M is a KWQFA where M is not permitted to halt in an accepting state until \$ is read, and the acceptance mode is changed to bounded positive one-sided error. Any BPQFA can be simulated by a KWQFA [8].

3 Latvian QFA

Our main result for this model is a complete characterization of the languages recognized by LQFA:

Theorem 1. *LQFA recognize exactly those languages whose syntactic monoid is in* **BG**.

Proof: We begin by showing that the languages recognized by LQFA forms a $*$-variety of languages. It is straightforward to show:

Theorem 2. *The class of languages recognized by LQFA is closed under union, complement, inverse homomorphisms, and word quotient.*

Next, to prove that LQFA cannot recognize any language whose syntactic monoid is not in **BG**, we need to show that LQFA cannot recognize Σ^*a or $a\Sigma^*$. We note that LQFA are a special case of Nayak's EQFA model [18], and EQFAs cannot recognize Σ^*a. We sketch the proof that $a\Sigma^*$ is not recognizable below.

Theorem 3. *LQFAs cannot recognize $a\Sigma^*$.*

Finally, we prove the following theorem below:

Theorem 4. *LQFAs recognize any language whose syntactic monoid is in* **BG**.

This will compete the characterization. □

Proof of Theorem 3 (sketch) Suppose the LQFA M recognized $a\Sigma^*$. Let ρ_w be the state of M on reading w as a density matrix. Suppose σ and τ are of the form $\sigma = \sum_{w \in S \subseteq a\Sigma^*} p_w \rho_w$, $\tau = \sum_{w \in T \subseteq b\Sigma^*} p_w \rho_w$ with $\sum p_w = 1$. By linearity we can distinguish between σ and τ using $P_\$$ with some fixed probability $p > 1/2$. We show that a sequence $\sigma_1, \sigma_2, \ldots$ of σ matrices and a sequence $\tau_1, \tau_2, \ldots$ of τ matrices converge to the same limit, causing a contradiction.

We will need some notions from quantum information theory [19]. A *completely positive superoperator* is a linear operation that is a completely positive map on the space of $d \times d$ (particularly, density) matrices. For any density matrix ρ, the *Von Neumann entropy* $S(\rho)$ of ρ is $\sum -\lambda_i \log \lambda_i$, where the λ_is are the eigenvalues of ρ. It can be shown that any sequence of unitary transformations and measurements forms a CPSO E satisfying $S(E\rho) \geq S(\rho)$ for any ρ.

For all CPSOs E, we define E' to be the (CPSO) operation that performs the operation E with probability $1/2$, and the identity otherwise.

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

Lemma 2. *Let A, B be two sequences of unitary transformations and measurements. Let $C = A_{lim}B_{lim}$ and $D = B_{lim}A_{lim}$. Then, $C_{lim} = D_{lim}$.*

Let A, B, be the operations corresponding to reading a, b. We also consider A_{lim}, B_{lim}, $C = A_{lim}B_{lim}$, $D = B_{lim}A_{lim}$, C_{lim} and D_{lim}. Let Q_a (Q_b) be the set of density matrices corresponding to all probabilistic combinations of states ρ_{ax} (ρ_{bx}). Let $\overline{Q_a}$ and $\overline{Q_b}$ be the closures of Q_a and Q_b.

Lemma 3. *Let ρ be the state after reading the start marker ¢. Then, $C_{lim}\rho \in \overline{Q_a}$ and $D_{lim}\rho \in \overline{Q_b}$.*

By Lemmas 2 and 3, there exists sequences corresponding to $C_{lim}\rho$ and $D_{lim}\rho$, that are respectively probabilistic combinations of ρ_{ax} and ρ_{bx}, and they converge to the same limit. □

The next theorem will assist in our proof of Theorem 4.

Theorem 5. *LQFA can recognize languages of the form $\Sigma^* a_1 \Sigma^* \dots a_k \Sigma^*$.*

Proof: We start with construction of a PRA that recognizes $\Sigma^* a_1 \Sigma^* \dots a_k \Sigma^*$ with probability $(\frac{n\text{-}1}{n})^k$, where n is any natural number. We construct our PRA inductively on the length of the subword. For $k = 1$ we construct $M^{(1)} = (Q^{(1)}, q_0, \Sigma, \{A_\sigma^{(1)}\}, Q_{acc}^{(1)})$ as follows. Let $Q^{(1)} = \{q_0, q_2, \dots, q_n\}$, $A_{a_1}^{(1)} = \frac{1}{n}\mathbf{1}$ (where $\mathbf{1}$ is a $n \times n$ matrix of all ones), $A_\sigma^{(1)} = I$ for all $\sigma \neq a_1$, and $Q_{acc}^{(1)} = Q^{(1)} \backslash \{q_0\}$. It is easy to check that this machine accepts any $w \in \Sigma^* a_1 \Sigma^*$ with probability $(\frac{n\text{-}1}{n})$ and rejects any $w \notin \Sigma^* a \Sigma^*$ with probability 1.

Assume we have a machine $M^{(i\text{-}1)} = (Q^{(i\text{-}1)}, q_0, \Sigma, \{A_\sigma^{(i\text{-}1)}\}, Q_{acc}^{(i\text{-}1)})$ recognizing inputs containing the subword $a_1 \dots a_{i\text{-}1}$ with probability $(\frac{n\text{-}1}{n})^{i\text{-}1}$, we construct $M^{(i)} = (Q^{(i)}, q_0, \Sigma, \{A_\sigma^{(i)}\}, Q_{acc}^{(i)})$ recognizing inputs containing the subword $a_1 \dots a_i$ with probability $(\frac{n\text{-}1}{n})^i$. Our augmentation will proceed as follows. First let $Q_{acc}^{(i)}$ be a set of $(n\text{-}1)^i$ new states all distinct from $Q^{(i\text{-}1)}$, and let $Q^{(i)} = Q^{(i\text{-}1)} \cup Q_{acc}^{(i)}$. For each of the states $q \in Q_{acc}^{(i\text{-}1)}$ we uniquely associate n-1 states $q_2, \dots, q_n \in Q_{acc}^{(i)}$. We leave q_0 unchanged.

Finally, we construct each $A_\sigma^{(i)}$ from $A_\sigma^{(i\text{-}1)}$. Define $\tilde{A}_\sigma^{(i\text{-}1)}$ to be the transformation that acts as $A_\sigma^{(i\text{-}1)}$ on $Q^{(i\text{-}1)} \subset Q^{(i)}$ and as the identity elsewhere. We let $A_\sigma^{(i)} = \tilde{A}_\sigma^{(i\text{-}1)} B_\sigma^{(i)}$, where $B_\sigma^{(i)}$ is an additional transformation that will process the a_i character (note that the matrices are applied from right to left). For all $\sigma \neq a_i$ we define $B_\sigma^{(i)} = I$. For $\sigma = a_i$ and we define $B_\sigma^{(i)}$ so that, independently for each $q \in Q_{acc}^{(i\text{-}1)}$, the transformation $\frac{1}{n}\mathbf{1}$ is applied to $\{q, q_2, q_3, \dots, q_n\}$. At the end we have a machine $M = M^{(k)}$ that recognizes $\Sigma^* a_1 \Sigma^*, \dots, a_k \Sigma^*$.

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 4. *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 of Lemma 4: 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 any $\{q, q'\} \subseteq Q_{acc}^{(j)}$, at any time there is an equal probability of being in state q or q'. Thus it is sufficient to keep track of the total probability of being in $Q_{acc}^{(j)}$. For any $S \subseteq Q$, denote by $P(S)$ the sum probability of being in one of the states of S.

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 $Q_{acc}^{(i)} \cup Q_{acc}^{(i+1)}$. Since $|Q_{acc}^{(i+1)}| = (n\text{-}1)|Q_{acc}^{(i)}|$, it follows that a $B_{a_{i+1}}^{(i+1)}$ operation will not increase $P(Q^{(i)})$ unless $P(Q_{acc}^{(i+1)}) > (n\text{-}1)P(Q_{acc}^{(i)})$. 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. □

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 $Q^{(i\text{-}1)}$ 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 4 this remains satisfied until a_2 is read, at which point M will satisfy $P(Q^{(2)}_{acc}) \geq \left(\frac{n-1}{n}\right)^2$. Inductively after reading subword a, M satisfies $P(Q_{acc}) \geq \left(\frac{n-1}{n}\right)^k$. Thus M indeed recognizes $\Sigma^* a_1 \Sigma^* \dots a_k \Sigma^*$.

All that remains is to show that we can simulate each A_σ using LQFA transformations. Recall that each A_σ is a product of $B^{(i)}_{a_i}$ matrices operating on different parts of the state space. If each $B^{(i)}_{a_i}$ has a unitary prototype, then each A_σ could be simulated using the series of l transformations and measurements. We first show that we can collapse this operation into one transformation and one measurement. Assume we have a sequence of l unitaries U_i on a space E, each of them followed by a measurement $E_{i1} \oplus \cdots \oplus E_{ik_i}$. Define a new space E' of dimension $(\dim E) \cdot \prod_i k_i$. It is spanned by states $|\psi\rangle|j_1\rangle \dots |j_l\rangle$, $|\psi\rangle \in E$, $j_i \in \{0, \dots, (k_i\text{-}1)\}$. Each U_i can be viewed as a transformation on E' that acts only on the $|\psi\rangle$ part of the state. Replace the measurements by unitary transformations V_i defined by:

$$V_i|\psi\rangle|j_1\rangle \dots |j_i\rangle \dots |j_l\rangle = |\psi\rangle|j_1\rangle \dots |(j_i + j) \bmod k_i\rangle \dots |j_l\rangle$$

for $|\psi\rangle \in E_{ij}$. Consider a sequence of l unitaries and l measurements on E. Starting from $|\psi\rangle$, it produces a mixed state $\{(p_i, |\psi_i\rangle)\}$, where each $(p_i, |\psi_i\rangle)$ corresponds to a specific sequence of measurement outcomes. Then, if we start $|\psi\rangle|j_1\rangle \dots |j_l\rangle$ and perform U_1, V_1, …, U_l, V_l and then measure all of j_1, …, j_l, the final state is $|\psi_i\rangle|j_1'\rangle \dots |j_l'\rangle$ for some j_1', …, j_l' with probability p_i. Thus, when we restrict to $|\psi\rangle$ part of the state, the two sequences of transformations are effectively equivalent. Finally, composing the U_i and V_i transformations gives one unitary U and we get one unitary followed by one measurement. It is now sufficient to prove that each $B^{(i)}_{a_i}$ has a unitary prototype.

Observe that any block diagonal matrix such that all of the blocks have unitary prototypes is itself a unitary prototype, and that unitary prototypes are trivially closed under permutations. Each $B^{(i)}_{a_i}$ can be written as a block diagonal matrix, where each block is the 1×1 identity matrix or the $\frac{1}{n}\mathbf{1}$ matrix, so it remains to show that there is a unitary prototype for $\frac{1}{n}\mathbf{1}$ matrices. Coincidentally the quantum Fourier transform matrix [19], which is the basis for most efficient quantum algorithms, is a unitary prototype for $\frac{1}{n}\mathbf{1}$. This completes the proof that A_σ can be simulated by an LQFA, and the proof of the theorem. □

Proof of Theorem 4: We give a PRA construction recognizing the language L defined by $w \in L$ if and only if $w = w_0 a_1 w_1 \dots a_k w_k$ where for each i, $w_0 a_1 w_1 \dots w_i \in L_i$ for some prespecified group languages $L_0, \dots, L_k$. By the cancellative law of groups, this is sufficient to show that PRA recognize any language of the form $L_0 a_1 L_1 \dots a_k L_k$. We will see that each transition matrix has a unitary prototype, thus there is an LQFA recognizing this language as well.

This along with the closure properties of LQFA is sufficient to prove that any language whose syntactic monoid is in **BG** is recognized by an LQFA.

For all i let $G_i = M(L_i)$. Also let $\varphi_i : \Sigma^* \to G_i$ and F_i be such that $\varphi_i^{-1}(F_i) = L_i$. We compose these groups into a single group $G = G_0 \times \cdots \times G_k$ with identity $1 = (1, 1, \ldots, 1)$.

Let $M = (Q, q_0, \Sigma, \{A_\sigma\}, Q_{acc})$ be a PRA recognizing the subword $a_1 \ldots a_k$ constructed as in Theorem 5. From M we construct $M' = (Q', q_0', \Sigma, \{A_\sigma'\}, Q_{acc}')$ recognizing L. We set $Q' = Q \times G$, $q_0' = (q_0, 1)$, $Q_{acc}' = Q_{acc} \times F_k$, and $A_{¢} = A_{\$} = I$. For each $\sigma \in \Sigma$ define A_σ' as follows. Let P_σ be the permutation matrix that maps (q, g) to $(q, g\sigma)$ for each $q \in Q$ and $g \in G$. For each $1 \leq i \leq k$ let $A_{i\sigma}'$ be the matrix that, for each $f \in F_{i-1}$, acts as the transformation $B_\sigma^{(i)}$ on $Q^{(i)} \times \{f\}$ and as the identity everywhere else. Finally, $A_\sigma' = P_\sigma A_{\sigma 1}' \ldots A_{\sigma k}'$.

The A_σ' are constructed so that M' keeps track of the current group element at every step. If M is in state (q, g), then after applying $A_1', \ldots, A_k'$ it remains in $Q \times \{g\}$ with probability 1. The P_σ matrix 'translates' all of the transition probabilities from $Q \times \{g\}$ to $Q \times \{g\sigma\}$. Initially M is in $Q \times \{1\}$, so after reading any partial input w, M will be in $Q \times \{1w\}$ with probability 1. In this way M will always keep track of the current group element.

Each A_σ' matrix refines A_σ from the $\Sigma^* a_1 \Sigma^* a_2 \ldots a_k \Sigma^*$ construction in such a way that, on input σ after reading w, we do not move from $Q^{(i-1)}$ to $Q^{(i)}$ (the action performed by $B_{a_i}^{(i)}$) unless $\sigma = a_i$ and $w \in F_{i-1}$. This is exactly what we need to recognize L. The transition matrices can be simulated by LQFA by the same argument as in Theorem 5.

Lemma 5. *Let w be any word. As we process the characters of w in M, for all $0 \leq i < k$ the total probability of being in one of the states of $Q^{(i)} \times G$ is nondecreasing.*

Proof: Same argument as in Lemma 4 holds.

Proof of correctness: It is easy to see that M will reject any word not in L. We do not move out of $Q^{(0)} \times G$ unless we read a_1 in the correct context. Inductively, we do not move into Q_{acc} unless we have read each subword letter on the correct context and the current state corresponds to a group element $f \in F_k$. Now suppose $w \in L$. Rewrite w as $w_0 a_1 \cdots a_k w_k$. Clearly M does not move out of $Q^{(0)} \times G$ while reading w_0. The character a_1 is now read, and M moves to $(Q^{(1)} \times G) \backslash (Q^{(0)} \times G)$ with probability $\frac{n-1}{n}$. By the previous lemma, this probability does not decrease while reading w_1. So now after reading $w_0 a_1 w_1$ we will be in $Q_{acc}^{(1)} \times G$ with probability $\frac{n-1}{n}$. If a_2 is read we move to $Q^{(2)}$ with probability $(\frac{n-1}{n})^2$. By induction after reading $w_0 a_1 \ldots w_{k-1} a_k$ we move to $(Q^{(k)} \times G) \backslash (Q^{(k-1)} \times G)$ with total probability at least $(\frac{n-1}{n})^k$. Finally, after reading w_k we move to Q_{acc}' with total probability at least $(\frac{n-1}{n})^k$, and so we accept any $w \in L$ with this probability. By choosing a suitable n we can recognize L with arbitrarily high probability. □

4 Results for BPQFA

Our main result for BPQFA is given below:

Theorem 6. *The language L has its syntactic monoid in* **BG** *iff it is a Boolean combination of languages recognized by BPQFA.*

Proof: Similar to the LQFA case, we first show that this class of languages forms a $*$-variety. BPQFAs have been shown to be closed under inverse homomorphisms and word quotient [8], and we get Boolean combinations by definition. Next, we give the lower bounds. It is known that BPQFA cannot recognize $\Sigma^* a$, since KWQFA cannot recognize $\Sigma^* a$ [15] and any BPQFA can be simulated by a KWQFA. This proof can be easily extended to Boolean combinations of BPQFA. We prove the following theorem later in the section:

Theorem 7. *The language $a\Sigma^*$ is not a Boolean combination of languages recognized by BPQFA.*

Thus L is a Boolean combination of languages recognized by BPQFA only if $M(L)$ is in **BG**. Finally, we prove the following upper bound, by extending a construction of [8] in a manner similar to Theorem 4:

Theorem 8. *Any language whose syntactic monoid is in* **BG** *is a Boolean combination of languages recognized by BPQFA.*

This completes the proof of the main result. □

Proof of Theorem 7: We use a technique introduced in [15] to analyze BPQFAs. Let ψ be an unnormalized state vector of M. Define $A'_\sigma = P_{non} A_\sigma$, and for any word $w = w_1 \dots w_k$ let $A'_w = A'_{w_k} \cdots A'_{w_1}$. Then if ψ is the start vector, the vector $\psi_w = A'_w \psi$ completely describes the probabilistic behaviour of M, since M halts while reading w with probability $1 - \|\psi_w\|_2^2$, and continues in state $\frac{\psi_w}{\|\psi_w\|_2}$ with probability $\|\psi_w\|_2^2$. We also use the following lemma from [4]:

Lemma 6. *[4] Let $\{x, y\} \subseteq \Sigma^+$. Then there are subspaces E_1, E_2 s.t. $E_{non} = E_1 \oplus E_2$ and (1) if $\psi \in E_1$, then $A'_x(\psi) \in E_1$ and $A'_y(\psi) \in E_1$ and $\|A'_x(\psi)\| = \|\psi\|$ and $\|A'_y(\psi)\| = \|\psi\|$; (2) if $\psi \in E_2$, then for any $\epsilon > 0$, and for any word $t \in (x|y)^*$ there exists a word $t' \in (x|y)^*$ such that $\|A_{tt'}(\psi)\| < \varepsilon$.*

We first show that, for any BPQFA M, any $\varepsilon > 0$, and for any two prefixes $v, w \in \{a, b\}^+$, there exists $v', w' \in \{a, b\}^*$ such that $\|A'_{vv'}\psi - A'_{ww'}\psi\|_2^2 < \varepsilon$. In other words, any input with prefix vv' is indistinguishable from an input with prefix ww' by M. Let $\psi = A'_{¢}(|q_0\rangle)$, and let b be some letter in $\Sigma \backslash \{a\}$. As in Lemma 6, 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 ψ as $\psi = \psi_1 + \psi_2$, where $\psi_i \in E_i$. 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 $\|A'_{vv'}\psi - \psi_1\|_2^2 < \varepsilon'$ and $\|A'_{ww'}\psi - \psi_1\|_2^2 < \varepsilon'$. For sufficiently small ε' we have $\|A'_{vv'}\psi - A'_{ww'}\psi\|_2^2 < \varepsilon$.

Suppose we have a language L that is a Boolean combination of m languages $L_1, \ldots, L_m$ recognized by BPQFA. As above, we can construct inductively on the L_i languages two words $v = v_1 v_2 \cdots v_m \in \{a,b\}^*$ and $w = w_1 w_2 \ldots w_m \in \{a,b\}^*$ such that av and bw are indistinguishable for every L_i. Thus we must have either $\{av, bw\} \subseteq L$ or $L \cap \{av, bw\} = \emptyset$. Either way, the Boolean combination of BPQFAs does not recognize $a\Sigma^*$. □

Note that in our characterization we have to take 'Boolean combinations' because BPQFA are not closed under complement. This follows from the theorem below, which we will prove in the full version:

Theorem 9. *Over any Σ s.t. $\{a,b\} \subseteq \Sigma$, BPQFA cannot recognize $\overline{\Sigma^* b \Sigma^* a \Sigma^*}$.*

5 Conclusion

In this paper we have produced algebraic characterizations for the languages that can be recognized by Brodsky-Pippenger Quantum Finite Automata and by a new model which we called Latvian Quantum Finite Automata. A somewhat surprising consequence of our results is that the two models are equivalent in power, up to Boolean combinations. It has been shown that a language L is recognizable by an LQFA iff its syntactic monoid is a block group; hence membership in the class is decidable. The situation is more complicated for BPQFA since the corresponding class of languages is not closed under complement. The good news is that we have shown that the class forms what is known as a *positive* $*$-variety and thus is amenable to algebraic description through the mechanism of *ordered monoids* [22]. We know that this positive $*$-variety strictly contains the regular languages that are open in the group topology and a precise characterization seems to be within reach.

Another open problem is to characterize algebraically the Kondacs-Watrous model. It is an easy consequence of our results on BPQFA that KWQFA can recognize any language whose syntactic monoid is in **BG**. However, outside of **BG** the question of language recognition is still unresolved.

The class of languages recognized by KWQFA is known not be closed under union [4], hence does not form a $*$-variety. It is nevertheless meaningful to ask for an algebraic description of the $*$-variety generated by those languages. We conjecture that the right answer involves replacing block groups by a 1-sided version $\mathbf{V}$ of this $\mathbf{M}$-variety defined by the following condition: for any $e = e^2$ and $f = f^2$ in M, $eM = fM$ imply $e = f$. The corresponding variety of languages can be described as largest variety that does not contain $\Sigma^* a$ for $|\Sigma| \geq 2$.

References

1. A. Ambainis and R. Freivalds. 1-way Quantum Finite Automata: Strengths, Weaknesses, and Generalizations. *Proceedings of the 39th IEEE Symposium on Foundations of Computer Science*, pp. 332–341. 1998.

2. A. Ambainis, A. Ķikusts: Exact Results for Accepting Probabilities of Quantum Automata. *Theoretical Computer Science*, 295, pp. 3–25, 2003.
3. D. Aharonov, A. Ambainis, Julia Kempe, Umesh Vazirani. Quantum walks on graphs. *Proceedings of STOC'01*, pp. 50–59.
4. A. Ambainis, A. Ķikusts, and M. Valdats. On the class of Languages Recognized by 1-way Quantum Finite Automata. *Proceedings of STACS 2001*, pp. 75–86. 2001.
5. A. Ambainis, A. Nayak, A. Ta-Shma and U. Vazirani. Quantum dense coding and quantum finite automata. *Journal of ACM*, 49, pp. 496–511, 2002.
6. M. Beaudry, F. Lemieux, and D. Thérien. Finite loops recognize exactly the regular open languages, *Proceedings of the 24th ICALP Colloquium on Automata, Languages and Programming*, LNCS 1256, Springer-Verlag. 1997.
7. A. Bertoni, C. Mereghetti, B. Palano. Quantum computing: 1- way quantum finite automata. *Proceedings of DLT'2003*, LNCS 2730, pp. 1–20. 2003.
8. A. Brodsky and N. Pippenger. Characterizations of 1-Way Quantum Finite Automata", *SIAM Journal on Computing*, 31(5), pp. 1456–1478, 2002.
9. M. P. Ciamarra. Quantum Reversibility and a New Model of Quantum Automaton. *FCT 2001*, pp. 376–379.
10. S. Eilenberg. Automata, Languages and Machines Vol B. *Academic Press.* 1976.
11. C. Fuchs, J. van de Graaf. Cryptographic distinguishability measures for quantum mechanical states. *IEEE Transactions on Information Theory*, 45(4), pp. 1216–1227, 1999.
12. M. Golovkins and M. Kravtsev. Probabilistic Reversible Automata and Quantum Automata. *COCOON 2002.* pp. 574–583. 2002.
13. J. Gruska. Quantum Computing. *McGraw-Hill*, p. 160. 1999.
14. P. Jeavons, D. Cohen and M. Gyssens. *Closure Properties of Constraint Satisfaction Problems.* JACM, 44, 4. pp. 527–548. 1997.
15. A. Kondacs and J. Watrous. On the power of Quantum Finite State Automata. *FOCS 1997*, pp. 66–75. 1997.
16. A. Marshall, I. Olkin. Inequalities: Theory of Majorization and Its Applications. *Academic Press*, 1979.
17. C Moore, J. Crutchfield. Quantum Automata and Quantum Grammars. *Theoretical Computer Science*, 237(1-2), pp. 275–306, 2000.
18. A. Nayak. Optimal Lower Bounds for Quantum Automata and Random Access Codes. *Proc. 40th FOCS*, pp 369–377. 1997.
19. M. Nielsen, I. Chuang. *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
20. J. E. Pin. BG=PG: A success story. *NATO Advanced Study Institute Semigroups, Formal Languages and Groups*, pp. 33-47. 1995.
21. J. E. Pin. On languages accepted by finite reversible automata. *Proceedings of the 14th International Colloquium on Automata, Languages, and Programming.* LNCS 267, pp. 237–249. 1987.
22. J. E. Pin, A variety theorem without complementation, *Russian Mathematics (Izvestija vuzov.Matematika)* 39, pp. 80–90. 1995.
23. J. E. Pin. Varieties of Formal Languages. *North Oxford Academic Publishers, Ltd, London.* 1986.
24. M. Rabin. Probabilistic Automata. *Information and Control*, 6(3), pp. 230–245. September 1963.
25. I. Simon. Piecewise Testable Events. *Proc. 2nd GI Conf.*, pp. 214–222, 1975.
26. L. Vandersypen, M. Steffen, G. Breyta, C. Yannoni, M. Sherwood, I Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414, 883–887, 2001.