Quantum Finite Automata using Ancilla Qubits

Kathrin Paschen
Lehrstuhl Informatik für Ingenieure und Naturwissenschaftler
Universität Karlsruhe

May 18, 2000

Abstract

We introduce a new model of quantum finite automata. By using ancilla qubits, it becomes possible to recognize any regular language with certainty. Some nonregular languages can be recognized with one–sided unbounded error. We analyze a class of languages that can be recognized in this model in terms of a cascade composition of automata. This also allows to treat the case of an automaton with both classical and quantum states.

1 Introduction

Quantum Computers, if they can be built, could be much more powerful than classical computers. However, their exact power is hard to estimate. This motivates the study of quantum finite automata: they are the quantum counterparts of a well–known classical model and may pave the way toward a better understanding of the theory behind quantum computing.

Several definitions of quantum finite automata (QFA) have been proposed ([3, 7, 8]). Some of them are considerably less powerful than classical finite automata (FA), others significantly more powerful. With the latter class, e.g. the 2QCFA in [3], it seems quite difficult to characterize the languages that they can recognize.

We introduce a new model of QFA. It uses ancilla qubits to satisfy unitarity conditions while simulating arbitrary FA. Ancilla QFA can recognize any regular language with certainty. Just like other types of QFA, they can also recognize certain nonregular languages with one–sided unbounded error. We suggest a characterization of a subset of these languages; to this end, we introduce a cascade composition of ancilla QFA and examine especially the case where every part of the cascade either simulates a classical FA or needs no ancilla qubits. This approach could be of interest also in the study of hybrid models like the 2QCFA in [3].

The paper is organized as follows: section 2 contains a brief introduction to QFA. Ancilla QFA are introduced and discussed in section 3, and the cascade composition is derived in section 4. The final section summarizes the results and points out some open questions.

2 Definitions

Generally, quantum finite automata (QFA), just like classical finite automata, have a finite set Q of states, a finite input alphabet Σ and a transition function δ that specifies how the automaton's state changes depend on the inputs.

Quantum finite automata are different from their classical counterparts in that they can be in a *superposition* of states. A superposition is a vector in a $|Q|$–dimensional complex vector space. We can choose a canonical basis of that space and a numbering of the states in Q and identify the state q_i with the ith basis vector. We use the Dirac notation and write $|q\rangle$ for a state q. A superposition of states is written as $\sum_{q_i \in Q} \alpha_i |q_i\rangle$; α_i is the *amplitude* of state q_i. All superpositions are required to have unit norm.

On reading an input, a quantum finite automaton changes its superposition of states. This change must preserve the unit norm; in finite–dimensional vector spaces, that is the case iff the corresponding operator is unitary.

Orthogonal measurements can be applied in order to determine the automaton's current state. They are given by an *observable* (a decomposition of $\mathbb{C}^{|Q|}$ into mutually orthogonal subspaces). When an observable is applied to a state, that state changes probabilistically to its projection onto one of the subspaces. The probability depends on the amplitudes; for example, measuring

$\alpha|q_0\rangle + \beta|q_1\rangle$ with the observable $\{q_0, q_1\}$ takes the system to state $|q_0\rangle$ with probability $|\alpha|^2$ and to state $|q_1\rangle$ with probability $|\beta|^2$.

Quantum systems can be composed using the tensor product. For example, if one part is in state $|q\rangle = \sum_i \alpha_i|q_i\rangle$, and the other is in state $|p\rangle = \sum_j \beta_j|p_j\rangle$, then together they are in state $|q\rangle \otimes |p\rangle = |q,p\rangle = \sum_{i,j} \alpha_i\beta_j|q_i,p_j\rangle$.

Several kinds of quantum finite automata have been proposed. We give a brief description of two of them before introducing the new model.

2.1 MO–QFA

Measure–once QFA (MO–QFA) [8] are 5–tuples $A = (Q, \Sigma, \delta, q_0, Q_F)$ where Q and Σ are defined as above and $Q_F \subseteq Q$ is the set of accepting states. The automaton is started in the configuration corresponding to its starting state q_0, on a tape containing an input word over Σ^+ to which $\#$ and $\$$ have been added as start and end markers, respectively ($\#, \$ \notin \Sigma$). This is done with all the QFA we discuss, even if it is not always explicitly mentioned.

For each $\sigma \in \Sigma \cup \{\#, \$\}$, the transitions of the automaton on input σ can be written as a complex $|Q| \times |Q|$ matrix V_σ whose entry ij is the amplitude $\delta(q_i, \sigma, q_j)$ for going from state q_i to state q_j on input σ. The automaton processes σ by multiplying its current state vector with V_σ.

In order for A to be a well–formed quantum finite automaton, these matrices must be unitary. This is the case iff

$$\forall q_1, q_2 \in Q, \sigma \in \Sigma \cup \{\#, \$\} : \sum_{q \in Q} \overline{\delta(q_1, \sigma, q)}\delta(q_2, \sigma, q) = \begin{cases} 1 & \text{if } q_1 = q_2 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\overline{x}$ for $x \in \mathbb{C}$ is the complex conjugate of x.

MO–QFA perform their only measurement after processing the right end marker. They use the observable (E, E_F) where E is the subspace spanned by $\{|q\rangle : q \notin Q_F\}$ and E_F the subspace spanned by $\{|q\rangle : q \in Q_F\}$. The word is accepted if this projection leaves the automaton in an accepting state.

The power of MO–QFA depends on whether they are required to accept with bounded error. Brodsky and Pippenger have shown [5] that the class of languages recognized by MO–QFA with bounded error is exactly the class of group languages (i.e., languages whose syntactic semigroup is a finite group).

On the other hand, the language $\{w \in \{a,b\}^* | |w|_a = |w|_b\}$ can be accepted by a MO–QFA with unbounded error [5].

2.2 MM–QFA

For bounded error acceptance, the power of MO–QFA is quite limited. One way of adding a little irreversibility (and thus power) to QFA is by introducing intermediate measurements.

However, doing a measurement that causes the superposition to collapse to a single state is not useful—it would turn the QFA into a probabilistic finite automaton. Instead, one can partition the set of states in three subsets Q_{acc}, Q_{rej} and Q_{non}, called the accepting, rejecting and non–halting states, respectively, and use the spans of these sets as observable. A measurement is performed after every step.

Kondacs and Watrous [7] have shown that any language recognized by an MM–QFA with bounded error is regular. Moreover, there are some surprisingly simple languages that cannot be recognized by MM–QFA with bounded error: one example is the language $\{a,b\}^*a$.

3 Ancilla QFA

3.1 Definition

For finite automata, the restriction to unitary transitions is quite limiting. We'll show a way of getting around this restriction. The idea is to add ancilla qubits, so that each transition can be unitary. Formally, this is done by adding an output tape to the QFA.

Definition 3.1 (ancilla QFA) *An ancilla QFA is a 6–tuple* $(Q, \Sigma, \Omega, \delta, q_0, Q_F)$ *where* Q, Σ, q_0 *and* Q_F *are just as with MO–QFA. In addition, there is a finite output alphabet* Ω *and the transition function* $\delta : Q \times \Sigma \times Q \times \Omega \to \mathbb{C}[0,1]$.

Since the output tape may contain a superposition of words, we need a modified well–formedness constraint for δ.

Lemma 3.2 *The evolution specified by δ is well–formed if $\forall \sigma \in \Sigma, q_1, q_2 \in Q$:*

$$\sum_{q \in Q, \omega \in \Omega} \overline{\delta(q_1, \sigma, q, \omega)} \delta(q_2, \sigma, q, \omega) = \begin{cases} 1 & q_1 = q_2 \\ 0 & q_1 \neq q_2 \end{cases} \tag{2}$$

Proof: A configuration of an ancilla QFA is a pair (τ, q) where τ is the contents of the output tape, and q is the current state. Each input symbol $\sigma \in \Sigma$ specifies an operator V_σ as described before. These operators act on $\Omega^* \otimes Q$. We have to show that for every $\sigma \in \Sigma$: if $\tau \neq \tau'$ or $q \neq q'$, then $V_\sigma|\tau, q\rangle \perp V_\sigma|\tau', q'\rangle$.

Since the automaton moves right after every step and writes an output symbol every time, $V_\sigma|\tau, q\rangle \perp V_\sigma|\tau', q'\rangle$ whenever $\tau \neq \tau'$.

So assume $\tau = \tau'$ and $q \neq q'$. Then $V_\sigma|\tau, q\rangle \perp V_\sigma|\tau, q'\rangle$ if the condition of the Lemma holds. □

The V_σ here is an infinite–dimensional matrix whose rows and columns are indexed by configurations. However, since the new state and output symbol do not depend on the tape contents, it is possible to write this matrix with $|\Omega||Q|$ columns and $|Q|$ rows.

Similar to simple 2–QFA [7], simple ancilla QFA are simple because the output symbol depends only on the state.

Definition 3.3 (Simple ancilla QFA) *An ancilla QFA is simple if there is a function $D : Q \to \Omega$ such that $\forall \sigma \in \Sigma, q, q' \in Q, \omega \in \Omega : \delta(q, \sigma, q', \omega) \neq 0 \Rightarrow D(q) = \omega$.*

Lemma 3.4 *For every ancilla QFA A, there is a simple ancilla QFA A' simulating A.*

Proof: A' needs enough output symbols so that D can be injective. □

It could be argued that adding a quantum output tape whose length is not constant means that ancilla QFA are not properly finite. On the other hand, that output tape is never read. Technically, we could use just enough ancilla qubits to hold a single output symbol. There are two ways of realizing this. The ancilla qubits could be reset after every step; this would be an irreversible operation. There is a more complicated solution which is completely reversible.

We use $|\Omega|$ qubits. On each qubit, we can perform two unitary operations U and V as well as their inverses. Choose for example $U = \widehat{U}_a, V = \widehat{U}_b$ where $\widehat{U}_a, \widehat{U}_b$ are as in the proof of Theorem 1 in [3]. Then to write output ω_i, apply U to the ith qubit and V to all other qubits.

With slight modifications, $\lceil \log_2 |\Omega| \rceil$ qubits would be enough. With more operations per qubit, that number could be reduced further. Of course, we could also use $|\Omega|$ pairwise prime numbers p_i and encode ω_i as $U^{p_i}V$; then a single qubit is enough.

3.2 Properties of ancilla QFA

Ancilla QFA can simulate any classical finite automaton.

Lemma 3.5 *For every regular language L, there is $k \in \mathbb{N}$ such that an ancilla QFA with k output symbols can recognize L exactly.*

Proof: Let L be a regular language, and A a minimal deterministic finite automaton recognizing L. For every input symbol σ, let G_σ be the directed graph whose vertices are the states of A with an edge from vertex s to vertex s' iff A, when in state s and reading input σ, goes to state s'. Now let $m(\sigma)$ be the maximum in–degree of all the vertices in G_σ, and $k = \max_\sigma m(\sigma)$. Then there is an ancilla QFA B with no more than k output symbols which recognizes L exactly.

B has one state for each state of A, say n states altogether. The transition function of A on input σ can be written as an $n \times n$ matrix M such that $M_{ij} = 1$ iff there is a transition from state i to state j on input σ and 0 otherwise. All row vectors of M are normalized because A is deterministic.

If M is unitary, then M can serve as transition matrix for B. If M is not unitary, there are states i, j, k of A with $i \neq j$ such that $\delta_A(i, \sigma, k) = 1$ and $\delta_A(j, \sigma, k) = 1$. We add outputs to obtain a finite transducer A' with transition function $\delta_{A'}$ such that $\delta_{A'}(i, \sigma, k, \omega_0) = 1$, $\delta_{A'}(i, \sigma, k, \omega_1) = 0$, $\delta_{A'}(j, \sigma, k, \omega_0) = 0$, and $\delta_{A'}(j, \sigma, k, \omega_1) = 1$. By repeating this, we can create a unitary transition matrix for B on input σ, using no more than $m(\sigma)$ output symbols. Since this can be applied for each input symbol, we do not need more than k output symbols altogether.

B recognizes L exactly because all amplitudes in the transition matrices of B are either 1 or 0. □

Corollary 3.6 *For every regular language L, there is $k \in \mathbb{N}$ such that a QFA using k ancilla qubits can recognize L exactly.*

Thus, Lemma 3.5 is a special case of the fact that any transformation on qubits can be realized by a superoperator in a suitably enlarged space [1, 6].

One obvious question is whether there is some constant $c \in \mathbb{N}$ such that any regular language can be recognized by some ancilla QFA using no more than c ancilla qubits. The answer is no.

Lemma 3.7 *For $k \in \mathbb{N}$, let $\mathcal{L}_k$ be the class of languages recognizable with bounded error by an ancilla QFA using no more than k output symbols. Then $\mathcal{L}_k \subsetneq \mathcal{L}_{k+1}$.*

Proof: We use Theorem 3.3 from [5] which says that a language can be recognized with bounded error by a MO–QFA iff the language can be recognized by a classical group finite automaton. Thus we only need to show that no constant number of output symbols can guarantee reversibility, where reversibility means that given the last output symbol, the current state, and the input symbol, the preceding state is uniquely determined.

For $k \in \mathbb{N}$, let $\Sigma_k = \{a_0, \ldots, a_k\}$, $L_k = \Sigma_k^* a_0 a_1 \ldots a_{k-1}$. L_k is recognized by the deterministic finite automaton A with $k+1$ states $s_0, \ldots, s_k$ and

$$\delta_A(s_i, a_j) = \begin{cases} s_1 & j = 0 \\ s_{i+1} & j = i \text{ and } i \leq k-1 \\ s_0 & \text{otherwise} \end{cases}$$

where s_0 is the starting state and s_k is the only accepting state. This automaton is minimal, and $\max_{a_i \in \Sigma_k} m(a_i) = m(a_0) = k+1$, so by Lemma 3.5, an ancilla QFA with $k+1$ output symbols can recognize L_k exactly. However, with fewer than $k+1$ output symbols, there have to be two states q_x and q_y

such that $\delta(q_x, a_0, q_1, a) \neq 0$ and $\delta(q_y, a_0, q_1, a) \neq 0$, so we do not have complete reversibility. □

Some people seem to consider proofs more aesthetic if they work with a two–letter alphabet. The proof of Lemma 3.7 can be modified to work with a two–letter alphabet by choosing a suitable encoding, e.g. encoding a_i as $a^i b$.

4 Cascade composition of ancilla QFA

Ancilla QFA can be quite powerful if we allow unbounded error recognition. An automaton A accepts a language L with one–sided unbounded error if A accepts all words from L with certainty and rejects words not from L with a probability > 0 (or vice versa). Because the error is one–sided, it is possible to run the automaton many times to obtain a better error bound.

Ambainis and Watrous [3] have shown that a 2–way automaton with quantum and classical states (2QCFA) can recognize the palindromes over the alphabet $\{a, b\}$ (L_{pal}) in exponential time and $\{a^n b^n | n \in \mathbb{N}\}$ (L_{eq}) in polynomial time. The state of a 2QCFA has a quantum and a classical part. When the automaton reads an input, it applies a quantum operation to the quantum part. This operation depends on the classical state and the input. The operation can be a unitary transformation or an orthogonal measurement. Afterwards, the automaton changes its classical state depending on the current classical state, the input symbol and the result of the measurement, if applicable. The head movements of the automaton are classical.

It seems difficult to characterize the class of languages that can be recognized by 2QCFA, so we use a restricted model that is still powerful.

For L_{pal} and L_{eq}, the ability to move left or take a measurement at any time is needed only to control the running time. A 1QCFA (i.e., a 1–way automaton with quantum and classical states; it works just like a 2QCFA except the head can only move right) could recognize both languages with one–sided unbounded error and make only a single measurement. Such a 1QCFA can be simulated by an ancilla QFA. This ancilla QFA has a special structure; it is equivalent to a cascade composition of an ancilla QFA simulating a classical FA and an MO–QFA.

4.1 Composition of QFA

It is known [8] that MO–QFA are closed under direct sum and tensor product. The direct sum $A \oplus B$ of two MO–QFA is a MO–QFA whose transition matrices have the form

$$U_\sigma = \begin{pmatrix} A_\sigma & 0 \\ 0 & B_\sigma \end{pmatrix},$$

where A_σ and B_σ are the transition matrices for A and B on input σ.

The tensor product $A \otimes B$ is a MO–QFA whose transition matrices are $U_\sigma = A_\sigma \otimes B_\sigma$.

These compositions do not allow A and B to interact. What about an analogue of the serial or cascade compositions of classical finite automata [4]? This requires adding outputs to one of the QFA and having them read by the second QFA, which could be difficult. However, it turns out that once we write down the transition matrices for the composite QFA, the outputs are no longer necessary.

Definition 4.1 (Cascade composition of QFA) *Let $A = (Q, \Sigma, \delta, q_0, Q_F)$, $B = (P, \Omega, \gamma, p_0, P_F)$ be two QFA, and $h : Q \times \Sigma \to \Omega$ a mapping. The cascade composition of A and B using h is an automaton $A \odot_h B = (Q \times P, \Sigma, \vartheta, (q_0, p_0), Q_F \times P_F)$ where*

$$\vartheta((q,p), \sigma, (q', p')) = \delta(q, \sigma, q') \cdot \gamma(p, h(q, \sigma), p').$$

Serial and parallel composition are special cases of cascades: for serial composition, h depends on q only; parallel composition is obtained when $\Omega = \Sigma$ and $h(q, \sigma) = \sigma$ for all $q \in Q$.

Let A_σ (B_ω) be the transition matrix of A (B) on input σ (ω), and let $A_\sigma^{i,j}$ stand for the entry in the ith row and jth column of A_σ. Then

$$(A \odot_h B)_\sigma = \begin{pmatrix} A_\sigma^{0,0} \cdot B_{h(q_0,\sigma)} & A_\sigma^{0,1} \cdot B_{h(q_0,\sigma)} & \dots \\ A_\sigma^{1,0} \cdot B_{h(q_1,\sigma)} & A_\sigma^{1,1} \cdot B_{h(q_1,\sigma)} & \dots \\ \vdots & & \end{pmatrix} \tag{3}$$

Thus, the action of σ on $A \odot_h B$ can be written as

$$(q, p) \stackrel{\sigma}{\mapsto} (A_\sigma(q), B_{h(q,\sigma)}(p)). \tag{4}$$

Lemma 4.2 *If A and B are QFA, then so is $A \odot_h B$, for any h.*

Proof: We show that, if A_σ and all the $B_{h(q_i,\sigma)}$ are unitary, then so is $(A \odot_h B)_\sigma$. All rows are normal, since for all k, l

$$\sum_{i=1}^{|Q|} \sum_{j=1}^{|P|} |A_\sigma^{k,i}|^2 |B_{h(q_k,\sigma)}^{l,j}|^2 = \sum_{i=1}^{|Q|} |A_\sigma^{k,i}|^2 = 1,$$

because the rows of A_σ and all $B_{h(q_i,\sigma)}$ are normal. The rows are also pairwise orthogonal (by a similar argument). □

Lemma 4.3 *Let A be an ancilla QFA with $\delta(q_i, \sigma, q_j, \omega_k) = \alpha_{i,j,k}$. Let A' be an automaton with $\delta'(q_i', \sigma, q_j') = \sum_k \alpha_{i,j,k}$. A' need not be well–formed. Define B, B' and $(A \odot_h B)'$ similarly. Then*

$$(A \odot_h B)' = A' \odot_h B'.$$

Proof: Let the transition Matrix for B on input $h(q_i, \sigma)$ be B_i, and define B_i' similarly for B'. Then

$$A \odot_h B = \begin{pmatrix} \alpha_{0,0,0} B_0 & \alpha_{0,0,1} B_0 & \cdots \\ \alpha_{1,0,0} B_1 & \alpha_{1,0,1} B_1 & \cdots \\ \vdots & & \end{pmatrix}.$$

$$(A \odot_h B)' = \begin{pmatrix} B_0' \sum_k \alpha_{0,0,k} & B_0' \sum_k \alpha_{0,1,k} & \cdots \\ B_1' \sum_k \alpha_{1,0,k} & B_1' \sum_k \alpha_{1,1,k} & \cdots \\ \vdots & & \end{pmatrix}.$$

With $A' = \begin{pmatrix} \sum_k \alpha_{0,0,k} & \sum_k \alpha_{0,1,k} & \cdots \\ \sum_k \alpha_{1,0,k} & \sum_k \alpha_{1,1,k} & \cdots \\ \vdots & & \end{pmatrix}$, the Lemma follows. □

The $\odot$ composition is associative, but not in general commutative. It is well–defined on ancilla QFA: Let A be an ancilla QFA with $\delta(q_i, \sigma, q_j, \omega_k) =$

$\alpha_{i,j,k}$. For the cascade, we use A' with $\delta(q_i', \sigma, q_j') = \sum_k \alpha_{i,j,k}$. A' need not be a QFA. However, $(A \odot_h B)' = A' \odot_h B'$ by Lemma 4.3, so in cascading ancilla QFA we can first eliminate the ancilla qubits, then cascade and then add ancilla qubits to the result until it is a well–formed QFA.

There are ancilla QFA with four or more states which cannot be decomposed as a cascade composition of nontrivial ancilla QFA.

Note that in order for an automaton to be an $\odot$–product, all its transition matrices have to decompose simultaneously in the manner of equation 3. There are ancilla QFA for which this is not the case.

On the one hand, the number of states in $A \odot B$ is the product of the numbers of states in A and B, so if an ancilla QFA has a prime number of states, decomposition is not straightforward. It is possible to add dummy states so that the automaton can be decomposed, but then one of the factors may have as many states as the original.

4.2 A cascade of classical FA and MO–QFA

We show how a 1QCFA can be simulated by a cascade composition of ancilla QFA. Our definition of 1QCFA is almost identical to the definition of 2QCFA in [3].

Definition 4.4 *A 1QCFA is a 10–tuple* $(Q, S, \Sigma, \Omega, \delta, h, q_0, s_0, Q_F, S_F)$ *with quantum states* Q*, classical states* S*, input alphabet* Σ*, starting states* q_0, s_0 *and accepting states* Q_F, S_F*. The transition function* $\delta : S \times \Sigma \to S$ *specifies the classical state changes. The function* $h : S \times \Sigma \to \Omega$ *specifies the unitary transformation* $\omega \in \Omega$ *to be performed on the quantum state.*

When the automaton is in state (s, q) and reads input σ, it changes its classical state to $\delta(s, \sigma)$ and performs the operation $h(s, \sigma)$ on its quantum state. The automaton accepts if, when it reads the right end marker, its classical state is in S_F and a measurement with the observable $(Q \setminus Q_F, Q_F)$ leaves the quantum state in a state from Q_F.

Lemma 4.5 *Every 1QCFA can be simulated by a cascade composition* $A \odot_h B$ *of two ancilla QFA.* A *simulates a classical FA and* B *simulates a MO–QFA.*

Corollary 4.6 *A language can be recognized with one–sided unbounded error by a 1QCFA iff it is the inverse image under some finite transduction of a language that can be recognized with one–sided unbounded error by a MO–QFA.*

5 Conclusions and open questions

We have introduced a new model of QFA and shown some of what it can do as well as suggesting a more general framework for analysis.

Ancilla QFA can simulate classical FA, and the simulation is straightforward. This is an advantage compared to the simulation using 2–QFA [7]. The cascade composition allows to study hybrid models, which are interesting because an automaton that is mostly classical and uses just a few quantum bits might be one of the earliest types of quantum computers that can be built.

Several extensions of the model are possible. For example, 2–way ancilla QFA would be interesting, as well as ancilla QFA with intermediate measurements. In order to strengthen the result of Corollary 4.6, it would be desirable to characterize the class of languages that can be recognized with one–sided unbounded error by MO–QFA. Equation 4 suggests that it is possible to describe the semigroups of languages recognized by 1QCFA (and generally by longer cascades of this type) in terms of a wreath product of finite semigroups and certain types of infinite semigroups.

References

[1] D. Aharonov, A. Kitaev, and N. Nisan. Quantum circuits with mixed states. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 20–30. ACM Press, 1998.

[2] A. Ambainis and R. Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations. *FOCS*, (39):332–341, 1998.

[3] A. Ambainis and J. Watrous. Two–way finite automata with quantum and classical states. Technical Report cs.CC/9911009, 1999.

[4] Michael A. Arbib. *Theories of Abstract Automata.* Prentice Hall, Inc., New York, 1969.

[5] A. Brodsky and N. Pippenger. Characterizations of 1-way quantum finite automata. Technical Report TR-99-03, University of British Columbia, 1999.

[6] J. Gruska. *Quantum Computing.* McGraw–Hill, 1999.

[7] A. Kondacs and J. Watrous. On the power of quantum finite state automata. *FOCS*, (38):66–75, 1997.

[8] C. Moore and J. P. Crutchfield. Quantum automata and quantum grammars. Technical report, Santa Fe Institute, 1997.

[9] J. Preskill. *Lecture Notes for Physics 229: Quantum Information and Computation.* http://www.theory.caltech.edu/people/preskill/ph229/, 1998.