Language Recognition by Generalized Quantum Finite Automata with Unbounded Error

Abuzer Yakaryılmaz and A. C. Cem Say
Boğaziçi University, Department of Computer Engineering,
Bebek 34342 Istanbul, Turkey
abuzer, say@boun.edu.tr

January 18, 2009

Abstract. We prove that the class of languages recognized by generalized quantum finite automata (GQFA’s) with unbounded error equals the class of stochastic languages. The capability of performing additional intermediate projective measurements does not increase the recognition power of GQFA’s over Kondacs-Watrous quantum finite automata in this setting. Unlike their probabilistic counterparts, allowing the tape head to stay put for some steps during its traversal of the input enlarges the class of languages recognized by GQFA’s with unbounded error.

1 Introduction

Research on theoretical models of practically implementable quantum computers has focused on variants of the quantum finite automaton (QFA). The bounded-error language recognition capabilities of various alternative QFA types have been analyzed and compared with their classical counterparts. In this paper, we examine the computational power of one of the most general read-only, one-way QFA models, the generalized quantum finite automaton (GQFA) [9], in the unbounded error setting. We prove that GQFA’s are equivalent in power to classical probabilistic finite automata in this setting. It turns out that the capability of performing additional intermediate projective measurements does not increase the recognition power of GQFA’s over Kondacs-Watrous quantum finite automata [5]. Unlike their probabilistic counterparts, allowing the tape head to stay put for some steps during its traversal of the input does enlarge the class of languages recognized by GQFA’s with unbounded error.

The rest of this paper is structured as follows: Section 2 contains the relevant background information. Section 3 presents our technique for

* This work was partially supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) with grant 108142 and the Boğaziçi University Research Fund with grant 08A102.
simulating a GQFA by a generalized probabilistic finite automaton [12]. Section 4 is a conclusion.

2 Preliminaries

This section covers the necessary background for the terminology and facts used in the rest of the paper.

Definition 1. A generalized probabilistic finite automaton (GPF A) with \( n \in \mathbb{Z}^+ \) states is a 5-tuple \( G = (S, \Sigma, \{A_\sigma \mid \sigma \in \Sigma\}, v_0, f) \), where

1. \( S = \{s_1, s_2, \ldots, s_n\} \) is the set of states,
2. \( \Sigma \) is the finite input alphabet,
3. \( A_\sigma \) is the \( n \times n \) real-valued transition matrix for symbol \( \sigma \), that is, \( A_\sigma(i, j) \) is the (possibly negative) “weight” of the transition from state \( s_i \) to state \( s_j \) when reading symbol \( \sigma \),
4. \( v_0 \) is the real-valued initial \( 1 \times n \) vector, and,
5. \( f \) is the real-valued final \( n \times 1 \) vector.

A GPFA \( G \) is associated by a function \( f_G : \Sigma^* \rightarrow \mathbb{R} \), in the following way: For an input string \( w \in \Sigma^* \),

\[
f_G(w) = v_0 A_{w_1} A_{w_2} \cdots A_{w_{|w|}} f, \tag{1}
\]

where \( w_i \) is the \( i^{th} \) symbol of \( w \).

The language \( L \subseteq \Sigma^* \) recognized by \( G \) with cutpoint \( \lambda \in \mathbb{R} \) is defined as

\[
L = \{ w \mid w \in \Sigma^*, f_G(w) > \lambda \}.
\]

It is known [12] that the class of languages recognized by GPFA’s with cutpoint is not changed when the transition matrices \( A_\sigma \) and the initial vector \( v_0 \) are constrained to be stochastic (i.e. with all entries in \([0, 1]\) and all rows adding up to 1), and the entries of the final vector \( f \) are restricted to be in \([0, 1]\). Such a machine is called a (one-way) probabilistic finite automaton (1PFA) [11]. Another generalization of the 1PFA is the two-way probabilistic automaton (2PFA) model [6], in which the input string is viewed as written on a tape flanked by two end-markers, and the machine has a tape head which can move to the left or stay put, as well as moving to the right. Interestingly, this additional capability does not increase the recognition power either [4]. The common family of languages recognized by GPFA’s, 1PFA’s and 2PFA’s with cutpoint is called the class of stochastic languages.
Definition 2. 1 A Generalized Quantum Finite Automaton (GQFA) with $n \in \mathbb{Z}^+$ states is a 6-tuple

$$G = (\Sigma, Q, \{U_\sigma \mid \sigma \in \Gamma\}, \{M_\sigma \mid \sigma \in \Gamma\}, Q_{\text{acc}}, Q_{\text{rej}}),$$

where

1. $\Sigma$ is the input alphabet, not containing symbols $\epsilon$ and $\$, which are the left and right end-markers, respectively,
2. $\Gamma = \Sigma \cup \{\epsilon, \}$ is the tape alphabet,
3. $Q = \{q_1, q_2, \ldots, q_n\}$ is the set of states, and $q_1$ is the initial state,
4. $U_\sigma$ is the $n \times n$ complex-valued unitary transition matrix for symbol $\sigma \in \Gamma$ such that $U_\sigma(j, i) = \alpha$ if the amplitude of the transition from $q_i$ to $q_j$ is $\alpha$ when reading the symbol $\sigma$,
5. for each $\sigma \in \Gamma$, there exists a constant $c_\sigma$ such that $M_\sigma$ is a measurement that consists of a finite set of projections $\{M_{\sigma, i}\}$ satisfying $\sum_{i=1}^{c_\sigma} M_{\sigma, i} = I$, where $1 \leq i \leq c_\sigma$, and $I$ is the $n \times n$ identity matrix,
6. $Q_{\text{acc}}$ and $Q_{\text{rej}}$, disjoint subsets of $Q$, are the sets of accepting and rejecting states.

Additionally, $Q_{\text{non}} = Q \setminus (Q_{\text{acc}} \cup Q_{\text{rej}})$ is the set of non-halting states, containing $q_1$; $P_{\text{non}}$, $P_{\text{acc}}$, and $P_{\text{rej}}$ are diagonal zero-one projection matrices, which project the state vector onto the subspaces of non-halting, accepting, and rejecting states, such that $P_{\text{non}}(j, j) = 1$ if $q_j \in Q_{\text{non}}$, $P_{\text{acc}}(j, j) = 1$ if $q_j \in Q_{\text{acc}}$, and $P_{\text{rej}}(j, j) = 1$ if $q_j \in Q_{\text{rej}}$, respectively.

A GQFA with input $w \in \Sigma^*$ traverses its tape, which contains the string $\epsilon w \$, starting from the left end-marker, undergoing three operations in each step: First, its state vector evolves according to the unitary transformation dictated by the scanned symbol. Then, it undergoes the projective measurement associated with the same symbol. Finally, another measurement is performed to see whether the current state of the machine is in $Q_{\text{acc}}$, $Q_{\text{rej}}$, or $Q_{\text{non}}$. The computation continues with the next tape symbol only if this measurement yields the result that the machine has not halted yet.

We will find it convenient to analyze the computation of such a GQFA by tracing it separately for each possible sequence of outcomes for the measurements $M_\sigma$ performed during the traversal of $w$. The set of all such sequences is

$$T_w = \{t = (t(1), t(2), \ldots, t(l)) \mid 1 \leq t(i) \leq c_{w_i}\},$$

1 The GQFA was originally defined by Nayak [9]. We use an equivalent definition by Mercer [7].
where $l = |w| = |w| + 2,$ and $1 \leq i \leq l.$

Unlike the convention in the probabilistic case, the state vector of a quantum automaton is a column vector. The initial state vector of the machine is $|q_1\rangle$, containing 1 as the top element and 0 in every other position. For every $t \in T_w$, $|t^n_0\rangle = |q_1\rangle$. The computation along each $t$ is traced by

$$W_{\sigma,j} = M_{\sigma,j}U_{\sigma}, \quad (3)$$

$$|t_i\rangle = W_{w_i,t(i)}|t^n_{i-1}\rangle, \quad (4)$$

$$|t^n_i\rangle = P_{\text{non}}|t_i\rangle, \quad (5)$$

$$|t^a_i\rangle = P_{\text{acc}}|t_i\rangle, \quad (6)$$

$$|t^r_i\rangle = P_{\text{rej}}|t_i\rangle, \quad (7)$$

$P_{t,\text{acc}}(i) = P_{t,\text{acc}}(i - 1) + \langle t^a_i | t^a_i \rangle, \quad (8)$

$P_{t,\text{rej}}(i) = P_{t,\text{rej}}(i - 1) + \langle t^r_i | t^r_i \rangle, \quad (9)$

where $\sigma \in \Gamma$, $1 \leq j \leq c_\sigma$, and $1 \leq i \leq l$; $W_{\sigma,j}$ represents the combined effect of the unitary transformation $U_{\sigma}$, followed by the measurement $M_{\sigma,j}$, (Equation 3); $|t_i\rangle$ is the vector containing the amplitudes of the states along the branch $t$ after the application of $W_{w_i,t(i)}$ in the $i$th step (Equation 4); $P_{t,\text{acc}}$ and $P_{t,\text{rej}}$ are finite sequences that trace the acceptance and rejection probabilities that have accumulated so far during the computation along branch $t$, with initial values $P_{t,\text{acc}}(0) = 0$ and $P_{t,\text{rej}}(0) = 0$, respectively. As seen in Equations 6–9, the acceptance and rejection probabilities are calculated using the amplitudes of the relevant states. Halting states “drop out” of the state vector, and computation continues with only the non-halting states having nonzero amplitude (Equation 5).

The overall probability that $G$ will accept $w$ is

$$f_G(w) = \sum_{t \in T_w} P_{t,\text{acc}}(|w| + 2), \quad (10)$$

and the language $L \subseteq \Sigma^*$ recognized by $G$ with cutpoint $\lambda \in [0,1)$ is defined as

$$L = \{ w \mid w \in \Sigma^*, f_G(w) > \lambda \}.$$

A (one-way) Kondacs-Watrous quantum finite automaton (1KWQFA) [5] is a restricted GQFA where all the $M_{\sigma} = \{I\}$, and so $T_w$ contains a single sequence. It is known [13] that the class of languages recognized with cutpoint (i.e. with unbounded error) by 1KWQFA’s equals the class of stochastic languages. Like the 1PFA, the 1KWQFA can be generalized by allowing the tape head more freedom of movement on the input.
Yakaryılmaz and Say [13] have shown that “1.5-way” KWQF A’s [1], where
the head is allowed to stay put, and does not have to move right in all
steps of the computation, can recognize some nonstochastic languages
with unbounded error, meaning that they are strictly more powerful than
1KWQFA’s, unlike their probabilistic counterparts.

3 Simulation of GQFA’s by Probabilistic Finite Automata

Any technique of simulating a GQFA by a GPFA will have to address
the following differences between the two models: Firstly, GQFA transi-
tions are specified using complex numbers, whereas GPFA’s are restricted
to using real numbers. Secondly, GQFA’s can halt before the entire in-
put string is read, whereas GPFA’s must traverse the string completely.
Finally, the “squaring” that occurs (Equation 8) during acceptance by
the GQFA has no counterpart in the GPFA. The simulation method we
present in Theorem 1 handles these issues with a quadratic overhead in
the number of states.

Theorem 1. Let \( G_1 \) be a GQFA with \( n \) states and \( f_{G_1} : \Sigma \rightarrow [0,1] \) be its
acceptance probability function. Then, there exists a GPFA \( G_2 \) with \( O(n^2) \)
states such that \( f_{G_1}(w) = f_{G_2}(w) \) for all \( w \in \Sigma^* \).

Proof. Let \( G_1 = (\Sigma, Q, \{U_\sigma \mid \sigma \in \Gamma\}, \{M_\sigma \mid \sigma \in \Gamma\}, Q_{acc}, Q_{rej}) \) and
\( |Q_{non}| = k_1, |Q_{acc}| = k_2, |Q_{rej}| = k_3 \), where \( k_1 > 0, k_2, k_3 \geq 0 \) and
\( k_1 + k_2 + k_3 = n \). We assume that the states in \( Q \) are indexed as

- \( q_i \in Q_{non} \) for \( 1 \leq i \leq k_1 \),
- \( q_i \in Q_{acc} \) for \( k_1 < i \leq k_1 + k_2 \),
- \( q_i \in Q_{rej} \) for \( k_1 + k_2 < i \leq n \).

For any symbol \( \sigma \in \Gamma \), and \( 1 \leq i \leq c_\sigma \), \( W_{\sigma,i} \) (Equation 3) can be partitioned into nine blocks as

\[
W_{\sigma,i} = \begin{pmatrix}
W_{\sigma,i,n-r} & W_{\sigma,i,a-n} & W_{\sigma,i,r-n} \\
W_{\sigma,i,n-a} & W_{\sigma,i,a} & W_{\sigma,i,r-a} \\
W_{\sigma,i,n-r} & W_{\sigma,i,a-r} & W_{\sigma,i,r-r}
\end{pmatrix},
\]
Lemma 1. Let $W_{\sigma,i,a-b}$ represent the transitions from the states in $Q_a$ to the states in $Q_b$, $a, b \in \{n : non, a : acc, r : rej\}$. We define $A_{\sigma,i}$, $v_0$, and $f$ as

$$A_{\sigma,i} = \left( \begin{array}{c|c} W_{\sigma,i,n-n} \otimes W_{\sigma,i,n-n}^* & \mathbf{0}_{k_1^2 \times k_2} \\ \hline W_{\sigma,i,n-a(1)} \otimes W_{\sigma,i,n-a(1)}^* & I_{k_2^2 \times k_2} \\ \vdots & \vdots \\ W_{\sigma,i,n-a(k_2)} \otimes W_{\sigma,i,n-a(k_2)}^* & \mathbf{0}_{1 \times k_2} \end{array} \right),$$

$v_0 = (1, 0, \ldots, 0)^T,$

$f = (0, \ldots, 0, 1_{1 \times k_2}),$

where

- $A_{\sigma,i}$ is a $(k_1^2 + k_2) \times (k_1^2 + k_2)$ square matrix;
- $v_0$ and $f$ are $k_1^2 + k_2$ dimensional column and row vectors, respectively;
- $W_{\sigma,i,n-n}^*$ denotes the conjugate of $W_{\sigma,i,n-n}$;
- $\mathbf{0}_{k_1^2 \times k_2}$ represents the $k_1^2 \times k_2$ zero matrix;
- $I_{k_2^2 \times k_2}$ is the $k_2 \times k_2$ identity matrix;
- $W_{\sigma,i,n-a}(i)$ denotes the $i$th row of $W_{\sigma,i,n-a}$, and $W_{\sigma,i,n-a}(i)^*$ denotes its conjugate, $1 \leq i \leq k_2$; and
- $1_{1 \times k_2}$ is a $1 \times k_2$ row vector whose entries are all 1’s.

Let $w \in \Sigma^*$ be an input string, $w = \varepsilon w \$, and $l = |w| = |w| + 2$. For each $t \in T_w$,

$$P_{t,acc}(|w| + 2) = f A_{w_1,t(l)} A_{w_1,t(l-1)} \cdots A_{w_1,t(1)} v_0.$$  

Proof. Let $v_i$ be the column state vector such that $v_i = A_{w_i,t(i)} v_{i-1}$, where $1 \leq i \leq l$. We will prove that

$$v_i = \left( \begin{array}{c} (|t_i^n(1..k_1) \otimes |t_i^n(1..k_1)^* \otimes P_{t,acc}^{\sigma_{k_1+1}}(i), \ldots, P_{t,acc}^{\sigma_{k_1+k_2}}(i))^T, \end{array} \right)$$

$k_1^2$ entries $k_2$ entries

where

- $0 \leq i \leq l$;
- $|t_i^n(1..k_1)$ is a $k_1$-dimensional vector formed of the top $k_1$ entries of the non-halting states vector $|t_i^n)$, and $|t_i^n(1..k_1)^*$ is its conjugate;
- $P_{t,acc}^{\sigma_{k_1+k_2}}(i)$ is the probability that the computation halts by reaching the accepting state $q_{k_1+k}$ in the first $i$ steps of branch $t$, $P_{t,acc}^{\sigma_{k_1+k}}(0) = 0$, where $1 \leq k \leq k_2$. 
For $i = 0$, 
\[ v_0 = \left( (1, 0, \ldots, 0) \otimes (1, 0, \ldots, 0) \right)_{k_1 \text{ entries}} \otimes \left( (1, 0, \ldots, 0) \right)_{k_2 \text{ entries}}^T, \]

\[ = \left( (|t^n_i\rangle (1..k_1) \otimes |t^n_i\rangle^* (1..k_1)) \right)^T P_{t,acc}^{q_{k_1+1}} (0), \ldots, P_{t,acc}^{q_{k_1+k_2}} (0))^T. \]

For $1 \leq i \leq l$, given that Equation 11 holds for $i - 1$, we calculate 
\[ v_i = A_{w_i,t(i)} v_{i-1}. \]

The first $k^2_i$ entries of $v_i$ make up the vector 
\[ (W_{w_i,t(i),n-n} \otimes W_{w_i,t(i),n-n}^*)(|t^n_{i-1}\rangle (1..k_1) \otimes |t^n_{i-1}\rangle^* (1..k_1)) \]

\[ = (W_{w_i,t(i),n-n} |t^n_{i-1}\rangle (1..k_1)) \otimes (W_{w_i,t(i),n-n}^* |t^n_{i-1}\rangle^* (1..k_1)) \]

\[ = (|t^n_{i}\rangle (1..k_1) \otimes |t^n_{i}\rangle^* (1..k_1)). \]

The $k^{th}$ one of the remaining entries of $v_i$, i.e., $v_i(k^2_i+k)$, where $1 \leq k \leq k_2$, is 
\[ (W_{w_i,t(i),n-a}(k) \otimes W_{w_i,t(i),n-a}(k)^*)(|t^n_{i-1}\rangle (1..k_1) \otimes |t^n_{i-1}\rangle^* (1..k_1)) \]

\[ = (W_{w_i,t(i),n-a}(k) |t^n_{i-1}\rangle (1..k_1)) \otimes (W_{w_i,t(i),n-a}(k)^* |t^n_{i-1}\rangle^* (1..k_1)) \]

\[ = (|t^n_{i}\rangle (k_1 + k)) \otimes (|t^n_{i}\rangle^* (k_1 + k)) + P_{t,acc}^{q_{k_1+k}} (i - 1) \]

\[ = (|t^n_{i}\rangle (k_1 + k)) (|t^n_{i}\rangle^* (k_1 + k)) + P_{t,acc}^{q_{k_1+k}} (i - 1) \]

\[ = P_{t,acc}^{q_{k_1+k}} (i). \]

So, it is easily followed that 
\[ v_l = \left( (|t^n_{l}\rangle (1..k_1) \otimes |t^n_{l}\rangle^* (1..k_1)) \right)^T \]

\[ = P_{t,acc}^{q_{k_1+1}} (l), \ldots, P_{t,acc}^{q_{k_1+k_2}} (l))^T, \]

and 
\[ f_{w_l} = \sum_{k=1}^{k_2} P_{t,acc}^{q_{k_1+k}} (l) \]

\[ = P_{t,acc} (l) \]

\[ = P_{t,acc} (|w| + 2). \]

By using Lemma 1, Equation 10 can be rewritten as 
\[ f_{\bar{G}_1}(w) = \sum_{t \in T_w} f_{A_{w_1,t(l)} A_{w_1,t(l-1)} \cdots A_{w_1,t(1)}} v_0. \]
which equals, by Equation 2,
\[
f(A_{w_1,c_{w_j}})(A_{w_{i-1},1}+\cdots+A_{w_{i-1},c_{w_{i-1}}})\cdots(A_{w_1,1}+\cdots+A_{w_1,c_{w_1}})v_0.
\]
By defining \(A_\sigma = \sum_{i=1}^{c_{\sigma}} A_{\sigma,i}\), we obtain
\[
f_{G_1}(w) = fA_{w_i}A_{w_{i-1}}\cdots A_{w_1}v_0.
\]
Since \(f_{G_1}(w) = (f_{G_1}(w))^T\),
\[
f_{G_1}(w) = v_0^TA_{w_1}^T\cdots A_{w_i}^Tf^T,
\]
and by defining \(v'_0 = v_0^TA_{\sigma_0}^T, A'_\sigma = A_{\sigma}^T (\sigma \in \Sigma)\), and \(f' = A_g^Tf^T\), we obtain
\[
f_{G_1}(w) = v'_0A'_{w_1}\cdots A'_{w_i}f'.
\]
Equation 12 has the same form as Equation 1, but the entries of \(v'_0, f', \) and \(A'_{\sigma \in \Sigma}\) are complex-valued. Using the fact that any complex number \(c = a + bi\) can be represented by the \(2 \times 2\) real matrix
\[
c = \begin{pmatrix} a & b \\ -b & a \end{pmatrix},
\]
we can simulate these [8, 13] by real-valued matrices \(v''_0, f'', \) and \(A''_{\sigma \in \Sigma}\) at the cost of doubling their dimensions. That is, the state set, say \(S\), of the machine for which the \(A''_{\sigma \in \Sigma}\) are transition matrices must contain \(2(k_1^2 + k_2)\) states.

If \(\mathbf{v}\) is the first row of \(v''_0\) and \(\mathbf{f}\) is the first column of \(f''\), then
\[
f_{G_1}(w) = \mathbf{v}A''_{w_1}\cdots A''_{w_i}\mathbf{f}.
\]
Therefore, for any given input string \(w \in \Sigma^*\),
\[
f_{G_1}(w) = f_{G_2}(w),
\]
where \(G_2 = (S, \Sigma, \mathbf{v}, \{A''_{\sigma} \mid \sigma \in \Sigma\}, \mathbf{f})\) is a GPFA with \(O(n^2)\) states.

Theorem 1 establishes that every language recognized by cutpoint by a GQFA is stochastic. Combining this with the fact, mentioned in Section 2, that every stochastic language can be recognized by a 1KWQFA, we obtain

**Corollary 1.** The class of languages recognized with unbounded error by GQFA’s equals the class of stochastic languages.
4 Concluding Remarks

In the bounded error case, it is an open problem to determine whether or not there exist languages which can be recognized by GQFA’s but not by 1KWQFA’s [7]. Our results establish that the capability of performing additional intermediate projective measurements does not increase the recognition power of GQFA’s over 1KWQFA’s in the unbounded error setting.

To our knowledge, no generalizations of GQFA’s which allow 1.5- or 2-way head movement, à la KWQFA’s [5] have been formally defined yet. For classical probabilistic finite automata, this generalization does not add language recognition power to the machines in the unbounded error case. Regardless of the specifics of its definition, any “1.5GQFA” model will have to contain 1.5KWQFA’s as specimens, and the fact [13] that 1.5KWQFA’s can recognize nonstochastic languages, combined with our Theorem 1, show that such an additional capability would enlarge the class of languages recognized by GQFA’s with unbounded error.

There exist one-way QFA models which are more powerful than the GQFA in the bounded error setting [2, 3, 10]. The relationship between the classes of languages recognized by these machines with unbounded error and the class of stochastic languages is an open question.

References

1. Masami Amano and Kazuo Iwama. Undecidability on quantum finite automata. In STOC ’99: Proceedings of the thirty-first annual ACM symposium on Theory of computing, pages 368–375, New York, NY, USA, 1999. ACM.
2. Alberto Bertoni, Carlo Mereghetti, and Beatrice Palano. Quantum computing: 1-way quantum automata. In Zoltán Ésik and Zoltán Fülöp, editors, Developments in Language Theory, volume 2710 of LNCS, pages 1–20. Springer Berlin / Heidelberg, 2003.
3. Massimo Pica Ciamarra. Quantum reversibility and a new model of quantum automaton. In FCT ’01: Proceedings of the 13th International Symposium on Fundamentals of Computation Theory, pages 376–379, London, UK, 2001. Springer-Verlag.
4. Jānis Kaneps. Stochasticity of the languages acceptable by two-way finite probabilistic automata. Diskretnaya Matematika, 1:63–67, 1989. (Russian).
5. Attila Kondacs and John Watrous. On the power of quantum finite state automata. In FOCS’97: Proceedings of the 38th Annual Symposium on Foundations of Computer Science, pages 66–75, Miami, Florida, 1997.
6. Yu. I. Kuklin. Two-way probabilistic automata. Avtomatika i vyčistitel’naja tehnika, 5:36–36, 1973. (Russian).
7. Mark Mercer. Lower bounds for generalized quantum finite automata. In Language and Automata Theory and Applications, volume 5196 of LNCS, pages 373–384. Springer Berlin/Heidelberg, 2008.
8. Cristopher Moore and James P. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 237(1-2):275–306, 2000.

9. Ashwin Nayak. Optimal lower bounds for quantum automata and random access codes. In *FOCS ’99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 369–376, Washington, DC, USA, 1999. IEEE Computer Society.

10. Kathrin Paschen. Quantum finite automata using ancilla qubits. Technical report, University of Karlsruhe, 2000.

11. Michael O. Rabin. Probabilistic automata. *Information and Control*, 6:230–243, 1963.

12. Paavo Turakainen. Generalized automata and stochastic languages. *Proceedings of the American Mathematical Society*, 21:303–309, 1969.

13. Abuzer Yakaryılmaz and A. C. Cem Say. Languages recognized with unbounded error by quantum finite automata. Technical report, 2008. arXiv:0809.0073v2.