The Complexity of Probabilistic versus Quantum Finite Automata

Gatis Midrijānis

University of Latvia, Raiņa bulvāris 19, Riga, Latvia. Email: mgatis@inbox.lv. Fax: +371-7820153.

Abstract. We present a language $L_n$ which is recognizable by a probabilistic finite automaton (PFA) with probability $1 - \epsilon$ for all $\epsilon > 0$ with $O(\log^2 n)$ states, with a deterministic finite automaton (DFA) with $O(n)$ states, but a quantum finite automaton (QFA) needs at least $2^{O(n/\log n)}$ states.

1 Introduction

A PFA is generalization of DFA. Many authors have tried to find out (2, 5, 7, 4 a. o.) the size advantages of PFA over DFA. On the other side it is known (3, 2) that the size of reversible finite automata (RFA) and the size of QFA exceed the size of the corresponding DFA almost exponentially for some regular languages (i.e. for languages recognizable by DFA). And so A. Ambainis, A. Nayak, A. Ta-Shma, U. Vazirani 3 wrote:

Another open problem involves the blow up in size while simulating a 1-way PFA by a 1-way QFA. The only known way for doing this is by simulating the PFA by a 1-way DFA and then simulating the DFA by a QFA. Both simulating a PFA by a DFA (7, 9, 8) and simulating DFA by a QFA (this paper) can involve exponential or nearly exponential increase in size. This means that the straightforward simulation of a probabilistic automaton by a QFA (described above) could result in a doubly-exponential increase in size. However, we do not know of any examples where both transforming a PFA into a DFA and transforming a DFA into a QFA cause big increases of size. Better simulations of PFA by QFAs may well be possible.

We will solve this problem.

2 Definitions and known results

We use the definition of 1-way QFA (further in text simply QFA) as in 2 and 3. This model was first introduced in 1 and is not the most general one, but is easy

* Research supported by Grant No.01.0.054 from the Latvian Council of Science, and Contract IST-1999-11234 (QAIP) from the European Commission.
to implement and deal with. A quantum finite automaton has a finite set of basis states $Q$, which consists of three parts: accepting states ($Q_{\text{acc}}$), rejecting states ($Q_{\text{rej}}$) and non-halting states ($Q_{\text{non}}$). One of the states, $q_{\text{ini}}$, is distinguished as the starting state.

Inputs to a QFA are words over a finite alphabet $\Sigma$. We shall also use the symbols $\varnothing$ and $\$ that do not belong to $\Sigma$ to denote the left and the right end marker, respectively. The set $I = \Sigma \cup \{\varnothing, \$\}$ denotes the working alphabet of the QFA. For each symbol $\sigma \in I$, a QFA has a corresponding unitary transformation $U_\sigma$ on the space $\mathbb{C}^Q$.

At any time, the state of a QFA is a superposition of basis states in $Q$.

The computation starts in the superposition $|q_{\text{ini}}\rangle$. Then the transformations corresponding to the left end marker $\varnothing$, the letters of the input word $x$ and the right end marker $\$ are applied in succession to the state of the automaton, unless a transformation results in acceptance or rejection of the input. A transformation consists of two steps:

1. First, $U_\sigma$ is applied to $|\psi\rangle$, the current state of the automaton, to obtain the new state $|\psi'\rangle$.
2. Then, $|\psi'\rangle$ is measured with respect to the observable $E_{\text{acc}} \oplus E_{\text{rej}} \oplus E_{\text{non}}$, where $E_{\text{acc}} = \text{span}\{|q\rangle \mid q \in Q_{\text{acc}}\}$, $E_{\text{rej}} = \text{span}\{|q\rangle \mid q \in Q_{\text{rej}}\}$, $E_{\text{non}} = \text{span}\{|q\rangle \mid q \in Q_{\text{non}}\}$. The probability of observing $E_i$ is equal to the squared norm of the projection of $|\psi'\rangle$ onto $E_i$. On measurement, the state of the automaton "collapses" to the projection onto the space observed, i.e., becomes equal to the projection, suitably normalized to a unit superposition.

If we observe $E_{\text{acc}}$ (or $E_{\text{rej}}$), the input is accepted (or rejected). Otherwise, the computation continues, and the next transformation, if any, is applied.

A QFA is said to accept (or recognize) a language $L$ with probability $p > \frac{1}{2}$ if it accepts every word in $L$ with probability at least $p$, and rejects every word not in $L$ with probability at least $p$.

A RFA is a QFA with elements only 0 and 1 in the matrices. A PFA is the same as a QFA but only instead of unitary matrices it has stochastic ones. A DFA is a PFA with only 0 and 1 in the matrices.

The size of a finite automaton is defined as the number of (basis) states in it. More exact definitions one can find, for example, in \cite{2}.

In \cite{2} there was given a language $L_n^\times$ consisting of one word $a^n$ in a single-letter alphabet and it was proved:

\textbf{Theorem 1}  

1. Any deterministic automaton that recognizes $L_n^\times$, has at least $n$ states.

2. For any $\epsilon > 0$, there is a probabilistic automaton with $O(\log^2 n)$ states recognizing $L_n^\times$ with probability $1 - \epsilon$.

\textit{Sketch of Proof.} The first part is evident. To prove the second part, Freivalds \cite{5} used the following construction. $O(\frac{\log n}{\log \log n})$ different primes are employed and $O(\log n)$ states are used for every employed prime. At first, the automaton randomly chooses a prime $p$, and the the remainder modulo $p$ of the length of
input word is found and compared with the standard. Additionally, once in every p steps a transition to a rejecting state is made with a ”small” probability \( \frac{\text{const}}{n} \).

The number of used primes suffices to assert that, for every input of length less than \( n \), most of primes \( p \) give remainders different from the remainder of \( n \) modulo \( p \). The ”small” probability is chosen to have the rejection high enough for every input length \( N \) such both \( N \neq n \) and \( \epsilon \)-fraction of all the primes used have the same remainders \( \mod p \).

In [3] was defined and theorem:

**Definition 2** \( f : \{0,1\}^m \times R \rightarrow \mathbb{C}^{2^n} \) serially encodes \( m \) classical bits into \( n \) qubits with success, if for any \( i \in [1..n] \) and \( b[i+1,n] = b[i+1...b_n] \in \{0,1\}^{n-i} \), there is a measurement \( \Theta_i,b[i+1,n] \) that returns 0 or 1 and has property that
\[
\forall b \in \{0,1\}^m : \text{Prob}(\Theta_i,b[i+1,n]|f(b,r)) = b_i \geq p.
\]

**Theorem 3** Any quantum serial encoding of \( m \) bits into \( n \) qubits with constant success probability \( p > 1/2 \) has \( n \geq \Omega(\sqrt{m}) \).

And also in [3] there was defined an \( r \)-restricted 1-way QFA for a language \( L \) as a 1-way QFA that recognizes the language with probability \( p > 1/2 \), and which halts with non-zero probability before seeing the right end marker only after it has read \( r \) letters of the input.

The following theorem was proved:

**Theorem 4** Let \( M \) be a 1-way QFA with \( S \) states recognizing a language \( L \) with probability \( p \). Then there is an \( r \)-restricted 1-way QFA \( M' \) with \( O(rS) \) states that recognizes \( L \) with probability \( p \).

### 3 Results

One of the components of the proof of Theorem 5 below is the following lemma:

**Lemma 1.** Language

\[ L_1 = \{ \omega \in \{0,1\}^*: \exists x, y \in \{0,1\}^* : \omega = x00y \} \]

is recognizable by a DFA.

**Sketch of Proof.** The automaton has five states: \( q_0, q_1, q_2, q_{\text{acc}} \) and \( q_{\text{rej}} \). Values of the transition function between states are:
\[
\begin{align*}
\delta(q_0, 0) &= q_1, & \delta(q_0, 1) &= q_0, & \delta(q_1, 0) &= q_2, & \delta(q_1, 1) &= q_0, & \delta(q_2, 0) &= q_2, & \delta(q_2, 1) &= q_2, & \delta(q_0, \$) &= q_{\text{rej}}, & \delta(q_1, \$) &= q_{\text{rej}}, & \delta(q_2, \$) &= q_{\text{acc}}.
\end{align*}
\]

**Theorem 5** For all \( k \geq 1, n = 2^k \), we define language

\[ L_n = \{ \omega \in \{0,1\}^n : \exists x, y \in \{0,1\}^* : \omega = x00y \}. \]

0. There is a RFA (so also a QFA, a PFA and a DFA) that recognize \( L_n \).

1. Any RFA that recognizes \( L_n \), has at least \( 2^{O(n)} \) states.
2. Any QFA that recognizes $L_n$ with probability $p > 1/2$, has at least $2^{O(n \log n)}$ states.

3. Any DFA that recognizes $L_n$, has at least $O(n)$ states.

4. For any $\epsilon > 0$, there is a PFA with $O(\log^2 n)$ states recognizing $L_n$ with probability $1 - \epsilon$.

Proof.

Zero part follows from fact that all finite languages are recognizable by some RFA and $L_n$ is finite language.

First part: We give to automaton word $a_1 a_2 a_3 a_4 a_5 a_6 1...a_k 1$, where $a_i \in \{0, 1\}$. It is obvious that then automaton cannot decide what to answer till the end of word. We prove that automaton always has to branch at every $a_i$. Suppose contrary, there is $a_i$ where automaton goes to the same state whether it read $a_i = 0$ or $a_i = 1$. Then forward we give the next symbols $01^{n-2i}$ and automaton cannot decide what to answer. So it must branch for every $a_i$, we can say it "remembers" this bit. But maybe it can merge ("forget") afterwards? No, because constructions

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig1}
\caption{From different states go to one with the same input symbol.}
\end{figure}

are forbidden by reversibility, but construction

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig2}
\caption{From different states go to one with different input symbols.}
\end{figure}

by the same reason as branching must occur (for all states $q_i$, $q_j$, $q_m$, $i \neq j$). Then it follows that automaton "remembers" all bits and the total number of states is at least $2^k$.

Second part: We use technique introduced by [3]. Let $M$ be any $n$-restricted QFA accepting $L_n$ with probability $p > 1/2$. The following claim formalizes the
intuition that the state of \( M \) after \( n \) symbols in form \( a_1 a_2 1 a_3 1 a_4 a_5 1 a_6 1 \ldots a_k 1 \) have been read is an "encoding" (in case of RFA, so deterministic, we said "remember") of the \( \{a_i\} \).

**Claim.** There is a serial encoding of \( k \) bits into \( \mathbb{C}^Q \), and hence into \( \lceil \log |Q| \rceil \) qubits, where \( Q \) is the set of basis states of the M.

**Proof.** Let \( Q_{\text{acc}} \) and \( Q_{\text{rej}} \) be the set of accepting and rejecting states respectively. Let \( U_{\sigma} \) be a unitary operator of \( M \) corresponding to the symbol \( \sigma \in \{0, 1, \$, \$\} \).

We define an encoding \( f : \{0, 1\}^k \rightarrow \mathbb{C}^Q \) of \( k \)-bit strings into unit superpositions over the basis states of the QFA \( M \) by letting \( |f(x)| \) be the state of the automaton \( M \) after the input string \( a_1 a_2 1 a_3 1 a_4 a_5 1 a_6 1 \ldots a_k 1 \) where \( a_i \in \{0, 1\} \) has been read. We assert that \( f \) is a serial encoding.

To show that indeed \( f \) is such an encoding, we exhibit a suitable measurement for the \( a_i \)-th bit for every \( i \in [1..k] \). Let, for \( y \in \{0, 1\}^{n-2^{i+1}} \), \( V_i(y) = U_3 U_1^{n-2^{i+1}} U_0 U_1^{y_1} U_2^{-1} \ldots U_1^{y_{n-2^{i+1}}} U_0 U_1^{y_{n-2^{i+1}}} U_2^{-1} \). The \( i \)-th measurement then consists of first applying the unitary transformation \( V_i(1 a_{i+1} .. a_k 1) \) to \( |f(x)| \), and then measuring the resulting superposition with respect to \( E_{\text{acc}} \otimes E_{\text{rej}} \otimes E_{\text{non}} \). Since for words with form \( a_1 a_2 1 \ldots a_k 01^{n-2^{i+1}} \), containment in \( L_n \) is decided by the \( a_i \), and because such words are accepted or rejected by then \( n \)-restricted QFA \( M \) with probability at least \( p \) only after the entire input has been read, the probability of observing \( E_{\text{acc}} \) if \( a_i = 0 \), or \( E_{\text{rej}} \) if \( a_i = 1 \), is at least \( p \). Thus, \( f \) defines a serial encoding.

Then it follows from Theorem \( \text{3} \) that \( \lceil \log |Q| \rceil = \Omega(\frac{k}{\log k}) \), but since \( k = \frac{n}{2} \), we have \( |Q| = 2^{\Omega(\frac{n}{\log n})} \). From Theorem \( \text{4} \) it follows that any quantum automaton that recognizes \( L_n \) also require \( 2^{\Omega(\frac{n}{\log n})} \) states.

Third part: Easy.

Fourth part: The PFA \( Q \) in Theorem \( \text{1} \) has one rejecting \( (q_{\text{rej}}) \), one accepting \( (q_{\text{acc}}) \), one initial \( (q_{\text{ini}}) \) state and many non-halting states \( q_i \). We build PFA \( Q' \) recognizing language \( L_n \) with one rejecting \( (q'_{\text{rej}}) \), one accepting \( (q'_{\text{acc}}) \), one starting \( (q'_{\text{ini}}) \) state and several non-halting states \( q'_{i,0}; q'_{i,1} \) and \( q'_{i,2} \) where \( i \) is from set of states’ indexes from automaton \( Q \). For every transition from state \( q_i \) to state \( q_j \) with probability \( p \) for the input symbol \( a \) (we denote this by \( f(q_i, a, q_j, p) \)) there are 6 transitions in \( Q' \) (we denote it by \( \Gamma' \)):

1. \( f'(q'_{i,0}, 1, q'_{i,0}, p) \)
2. \( f'(q'_{i,0}, 0, q'_{i,1}, p) \)
3. \( f'(q'_{i,1}, 1, q'_{i,0}, p) \)
4. \( f'(q'_{i,1}, 0, q'_{i,2}, p) \)
5. \( f'(q'_{i,2}, 1, q'_{i,1}, p) \)
6. \( f'(q'_{i,2}, 0, q'_{i,2}, p) \)

For every transformation \( f(q_{\text{ini}}, \$, q_i, p) \), there is a transformation \( f'(q'_{\text{ini}}, \$, q'_{i,0}, p) \). For every \( f(q_i, a, q_{\text{rej}}, p) \) there is \( f'(q'_{i,k}, x, q'_{\text{rej}}, p) \) such that for all \( k \in \{0, 1, 2\}, x \in \{0, 1\} \), and for every \( f(q_i, \$, q_{\text{rej}}, p) \) there is \( f'(q'_{i,k}, \$, q'_{\text{rej}}, p) \).
p) for all $k \in \{0, 1, 2\}$, and for any $f(q_i, \$, q_{acc}, p)$ there are $f'(q_i', 2, \$, q_{acc}', p)$,
$f'(q_i', 0, \$, q_{rej}', p)$, $f'(q_i', 1, \$, q_{rej}', p)$.

Informally, we make 3 copies from states in $Q$ and their meaning is similar than for states of automaton from Lemma 1. Automata computes parallel two things: is length of input word $n$ and is there any adjacent zeroes in it. It is obviously that the accepted words are those whose length is $n$ and there are two adjacent 0 in them.

□

4 Conclusion

We have shown that sometimes quantum automata must be almost doubly exponential larger than classical automaton. But there still remains open the other question. As follows from result of Ambainis and Freivalds [2], any language accepted by a QFA with high enough probability can be accepted by a RFA which is at most exponentially bigger that minimal DFA accepting the language. Thus follows that Theorem 5 is close to maximal gap between probabilistic and quantum automaton with high enough (this was precisely computed by Ambainis and Ūkūns [8] - greater than $\frac{524+\sqrt{7}}{81} = 0.7726...$) probability of success. But it is not clear how it is when we allow smaller probability of correctness. Author do not know any lower or upper bound in this case.

Acknowledgements

I would like to thank Rūsiņš Freivalds for suggesting the problem and help during research.

References

1. A. Kondacs, J. Watrous, “On the power of quantum finite state automata”, in Proc. 38th FOCS, 1997, pp. 66–75.
2. A. Ambainis, R. Freivalds, “1–way quantum finite automata: strengths, weaknesses, and generalizations”, in Proc. 39th FOCS, 1998, pp. 332–341.
3. A. Ambainis, A. Nayak, A. Ta-Shma, U. Vazirani: “Dense Quantum Coding and a Lower Bound for 1-way Quantum Automata”, quant-ph/9804043, 1998.
4. Z. Rasčevskis, “The Complexity of Probabilistic versus Deterministic Finite Automata”, http://www.ima.mdh.se/personal/rbr/courses/sundbyholmproc/zigmars.ps
5. R. Freivalds, “On the growth of the number of states of determinization of probabilistic automata,”, Avtomatika i Vychislitel’naja Tehnika (in Russian), 1982, N.3, pp. 39–42.
6. M. O. Rabin, “Probabilistic Automata”, Information and Control, 6(1963), pp. 230–245.
7. A. Ambainis, “The complexity of probabilistic versus deterministic finite automata”, Proceedings of ISAAC’96, Lecture Notes in Computer Science, 1178(1996), pp. 233–237.
8. A. Ambainis, A. Ūkūns, “Exact results for accepting probabilities of quantum automata”, quant-ph/0109136, 2001.