Quantum Information and Computation Report : 1-way quantum finite automata: strengths, weaknesses and generalizations

Octave Mariotti

Although it has been studied for a few decades, quantum computation is still full of unknown. A factor making this exploration difficult is the complexity of building quantum computers, as they require the ability to operate on a theoretically unbounded number of qubits. Of course, this is not possible in practice, as even classical computers do not have the infinite memory of Turing Machines. It remains nonetheless that effectively implementing quantum algorithms, such as Shor's factoring algorithm, is not likely to be done in the next decade.

However, there exists others models of quantum computation, derived from classical models, such as quantum push-down or finite automata. A study of the later reveals some interesting properties, hinting that the power of quantum computation is more complicated than simply being able to perform several computation in parallel by operating on a superposition of states. In this report, we will present the most important results of [1], and try to motivate the study of small quantum computing models. Since the main results are broken down into several technical proofs, related to each other in such a way that none could reasonably be evicted, we will mainly present ideas behind the proofs in order to convince a sceptical reader that these results are indeed provable rather than extensive demonstrations.

1 Background

The slow advancement towards the construction of a full quantum computer could mean that the first efficient quantum computer to be built will not be fully quantic, but instead only use small quantum devices as accelerators. Indeed, 1-way quantum automata only require a quantum system bounded in size to function : a classical device reads the input letter by letter, and applies the corresponding transformations to the quantum system. This prediction, stated in 1998 in [1],

still holds some truth today, despite a much more active research in quantum computing.

Quantums automata were simultaneously introduced by Kondacs and Watrous [4] and Moore and Crutchfield [5], with slightly different models. One year later, Ambainis and Freivalds further develop this model in "1-way quantum finite automata: strengths, weaknesses and generalizations"[1], and show several interesting properties as well as extensions of it. Since then, research on quantum automata did not undergo any breakthrough, and they most notably have become tools to introduce and demonstrate key concepts of quantum computing.

2 Definition

In order to formally define 1-way quantum finite automata, one must first define finite automata, reversible finite automata and probabilistic reversible finite automata.

Definition 1. A finite automaton (FA) is tuple $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ where :

- Q is a finite set of state.
- Σ is the input alphabet.
- $\#, \$ \notin \Sigma$ are respectively the left and right endmarkers of the tape
- $\Gamma = \Sigma \cup \{\#, \$\}$ is the working alphabet.
- $\delta : Q \times \Gamma \times Q \longrightarrow \mathbb{R}$ is the transition function, that gives the probability of a specific transition occurring.
- q_0 is the initial state of the automaton.
- Q_{acc} and Q_{rej} are respectively the set of accepting and rejecting states.

$Q_{non} = Q \backslash (Q_{acc} \cup Q_{rej})$ is the set of non-halting states.
Finally, we require that

$$\forall (q, a) \in Q \times \Gamma, \sum_{q' \in Q} \delta(q, a, q') = 1$$

This ensures that δ gives the probability to take a given transition.

When the computation begins, the automaton starts in q_o, and the left endmarker $\#$ is read. The automaton ends up in state q_1 with probability $\delta(q_o, \#, q_1)$. The computation continues

until the state leaves Q_{non}. Note that we can always restrict the termination of computation to this case by specifying $\delta(q, \$, q')$ in a smart way (that is, for any q, $q' \in Q_{acc} \cup Q_{rej}$).

A FA is deterministic when for any $(q, a) \in Q \times \Gamma$, there is exactly one $q' \in Q$ such that reading a in q leads to q', or more formally, such that $\delta(q, a, q') = 1$.

This definition of FAs differs a bit from classical ones, but it is easy to seee that they hold the same computational power while being closer to quantum automata in the way they work.

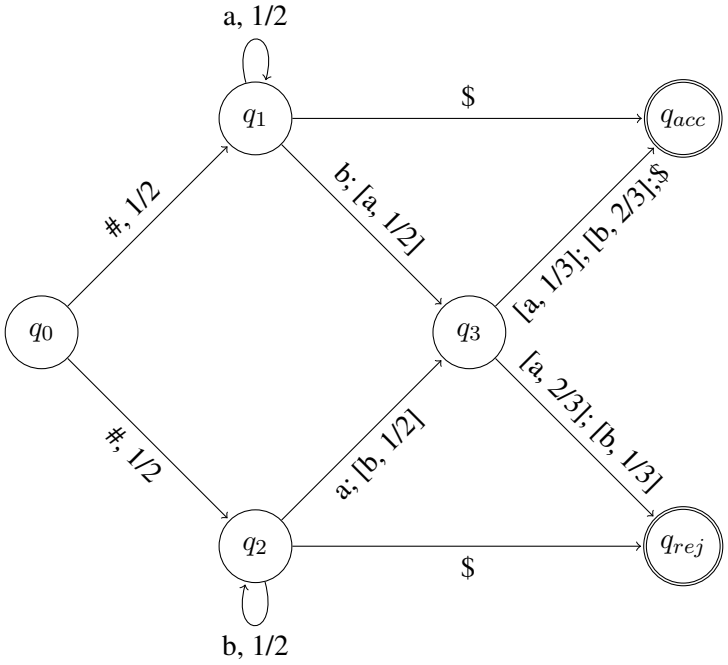

Figure 1: A finite automaton

Definition 2. A reversible finite automaton (RFA) is a deterministic finite automaton such that for any $(a, q') \in \Gamma \times Q$, there is at most one $q \in Q$ such that reading a in q leads to q', or more formally, such that $\delta(q, a, q') = 1$.

Intuitively, they are deterministic automata whose computations can be traced : given the last state and the input, there is only one way to go back to q_o following the letters of the input from the last to first.

Definition 3. A reversible automaton with probabilistic choices (PRFA) is a finite automaton such that for any $(a, q') \in \Gamma \times Q$, there is at most one $q \in Q$ such that reading a in q leads to q', or more formally, such that $\delta(q, a, q') \neq 0$.

It is the probabilistic analogue of the RFA. Of course, being reversible restrict the probabilistic choices that the automaton can make.

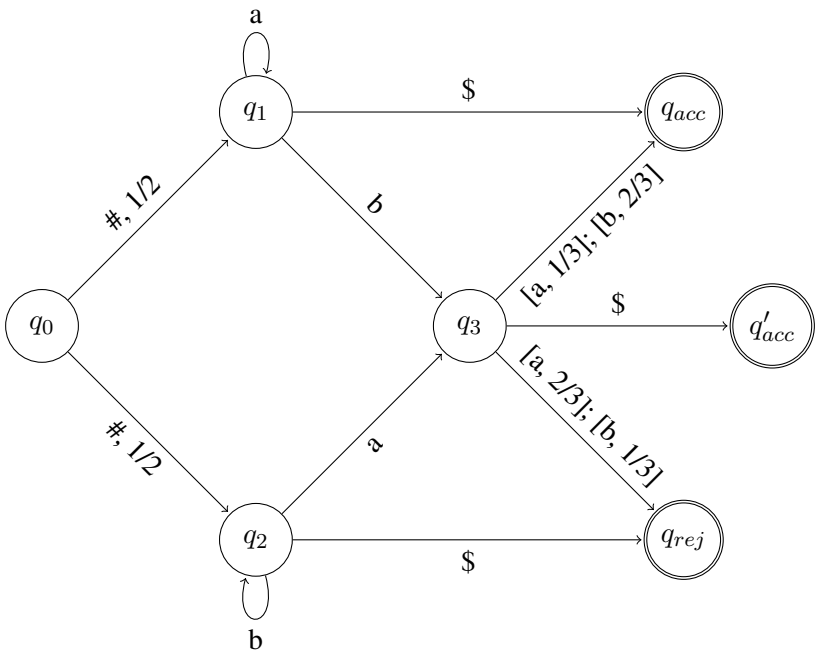

Figure 2: A reversible automaton with probabilistic choices

Although these models may seem unnatural, they can be viewed as classical analogues to quantum finite automata, as the reversibility is mandatory in quantum computing. PRFAs will in particular be used to asses the power of quantum automata by comparing what they can compute. Indeed, QFAs are probabilistic reversible automata, which make PRFAs the adequate model to determine what comes from quantum computation and what is simply probabilistic.

Definition 4. A 1-way quantum finite automaton (QFA) can be defined as a tuple $M = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$, with the same convention as finite automata, except that the transition function $\delta : Q \times \Gamma \times Q \longrightarrow \mathbb{C}$ changes codomain, and we require :

$$\forall (q, a) \in Q \times \Gamma, \sum_{q' \in Q} |\delta(q, a, q')|^2 = 1$$

We call an element of $l_2(Q)$ a superposition of states, and for $q \in Q$, we denote by $|q\rangle$ the superposition with value 1 at q and 0 anywhere else.

To ease comprehension and manipulation, we define, for any $a \in \Gamma$, $V_a : l_2(Q) \longrightarrow l_2(Q)$

by

$$V_a(|q\rangle) = \sum_{q' \in Q} \delta(q, a, q') \, |q'\rangle$$

It maps the current state to the next when reading a, and can easily be extended to superpositions. Previous assumptions implies that $V_a$is unitary for any a.

There are two different models of computations for QFAs.

- **Measure-once**[5] In the measure-once computation, an automaton reads its input and applies the corresponding unitaries to its superposition letter after letter. Then, when the right endmarker is read, a measurement is performed, and the superposition collapses either to a state of Q_{acc} or Q_{rej}
- **Measure-many**[4] In the measure many model, the state of the automaton is observed after each transformation by a unitary, in the measurement basis $E_{non} \oplus E_{acc} \oplus E_{rej}$, which are the subspaces spanned by the non-halting, accepting and rejecting states respectively. The computation ends when a measurement makes the superposition collapse to a halting subspace.

It was shown in [4] that measure-many automata actually contain measure-once automata. Therefore, we will only consider the second model. A working example of quantum finite automata can be found in the proof of claim 1.

3 Computational properties of QFAs

The first results concerning QFAs were quite pessimistic. In fact, they are only able to recognize a strict subset of regular languages [4]. We will see that they have the surprising property that their power depends on their probability of acceptance. If we require it to be higher than $7/9$, then their power is similar to that of RFAs. Conversely, those which accept with a smaller probability are slightly more powerful.

Theorem 1. Let L be a language and M be its minimal automaton. Assume that there is a word x and two states $q_1, q_2 \in Q$ satisfying:

- $q_1 \neq q_2$,
- If M starts in state q_1 and reads x, it passes to q_2,

- If M starts in state q_2 and reads x, it passes to q_2,
- q_2 is neither "all-accepting" state, nor "all-rejecting" state.

Then L cannot be recognized by a QFA with probability at least $7/9 + \epsilon$ for any fixed $\epsilon > 0$.

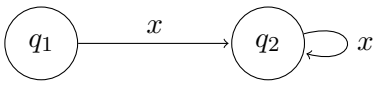

Figure 3: The "forbidden construction"

Proof. The proof recalls many concepts of Markov chains. Given a QFA M recognizing L, we show that for some word y, the probability of M giving the correct answer on y is less than $7/9 + \epsilon$. It relies on decomposing the non-halting subspace of states in two subspaces : transient and recurrent. Reading x will decrease the probability of being in the transient subspace, up to 0 when reading it infinitely many times.

Then, we show that the probability of halting relies only on the part of the decomposition belonging to the transient subspace, and we bound this probability. Finally, we show that deciding whether $y \in L$ is inconsistent with a $7/9 + \epsilon$ bound, as M cannot distinguish y from $x^i y$ with enough certainty. □

Theorem 2. Let L be a language and M be its minimal automaton. If M does not contain the "forbidden construction" of theorem 1, then L can be recognized by a reversible finite automaton.

Proof. We define non-reversibility as a tuple (q_1, q_2, q, a) where $q_1, q_2, q \in Q$, $a \in \Sigma$, $q_1 \neq q_2$, and reading a in q_1 or q_2 leads to q. Such tuples can be partially ordered with the relation : $(q_1, q_2, q, a) < (q_1', q_2', q', a')$ if and only if q_1' or q_2' is reachable from q. We can verify that < is transitive and anti-reflexive.

Hence, given an automaton M verifying the hypothesis of the theorem, we can turn it into a reversible automaton. Indeed, if it is not reversible, it must contain a tuple (q_1, q_2, q, a) maximal with respect to <. We can duplicate q and all states reachable from q, such that reading a in q_1 leads to the first copy, and reading a in q_2 leads to the second copy. This decrease the finite number non-reversibility tuples of M by one. Applying this construction recursively turns M into a reversible automaton. □

Corollary 1. A language can be recognized by a quantum finite automaton with probability $7/9 + \epsilon$ if and only if it can be recognized by a reversible finite automaton.

Proof. Follows from theorem 1 and 2. □

Claim 1. The language $L = a^*b^*$ can be recognized by a QFA with the probability of correct answer $p = 0.68\ldots$ where p is the root of $p^3 + p = 1$.

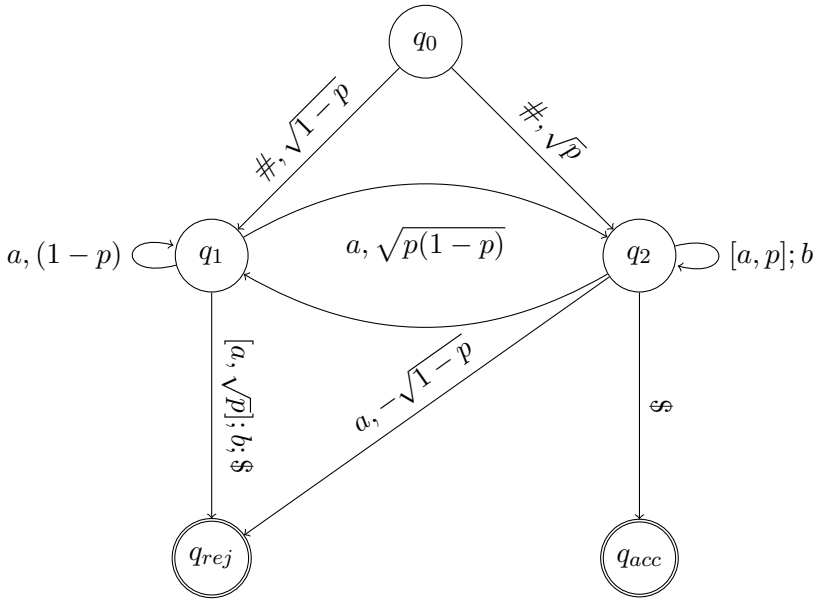

Figure 4: The QFA M recognizing a^*b^*

Proof. We will see how M acts depending on its input

- Case 1 : a^*

  The state after reading # is $|\psi\rangle = \sqrt{1-p}\,|q_1\rangle + \sqrt{p}\,|q_2\rangle$. Then :

$$\begin{aligned} V_a(|\psi\rangle) &= \sqrt{1-p}[(1-p)\,|q_1\rangle + \sqrt{p(1-p)}\,|q_2\rangle + \sqrt{p}\,|q_r ej\rangle] \\ &\quad + \sqrt{p}[(p)\,|q_2\rangle + \sqrt{p(1-p)}\,|q_1\rangle - \sqrt{1-p}\,|q_r ej\rangle] \\ &= (1-p+p)\sqrt{1-p}\,|q_1\rangle + (1-p+p)\sqrt{p}\,|q_2\rangle \\ &= |\psi\rangle \end{aligned}$$

  Thus, the state remains $|\psi\rangle$ while reading a's, and upon reading the right endmarker, M accepts with probability p.

- Case 2 : a^*b^+

  When the first b is read, the state becomes $V_b(|\psi\rangle) = \sqrt{1-p}\,|q_{rej}\rangle + \sqrt{p}\,|q_2\rangle$. Performing a measurement in this state has probability $1-p$ to end up in q_{rej}, in which case the computation ends, and probability p to end up in q_2, in which case M will end up accepting the input with probability 1. Thus, the acceptance probability is p.

- Case 3 : $x \notin a^*b^*$

  The initial segment of x is $a^*b^+a^+$. Reading the first b makes M reject with probability $(1-p)$. Then, reading a makes it reject with probability $p(1-p)$. Then, another b or the right endmarker follows, making once again M reject with probability $p^2(1-p)$.

  This yields

$$\begin{aligned} P_{rej} &\geq 1 - p + p(1-p) + p^2(1-p) \\ &\geq (1 + p + p^2)(1-p) \\ &\geq 1 - p^3 \\ &\geq p \end{aligned}$$

□

Corollary 2. Some languages can be recognized by QFAs with probability p, but not with probability greater than $7/9$.

Proof. The minimal automaton of a^*b^* contains the forbidden construction of theorem 1. □

This shows that QFAs are a bit more powerful than PRFAs, meaning that quantum phenomena can help computation, even on such restricted models.

4 Complexity

Although QFAs lack computational capabilities compared to their classical counterparts, they have the ability to be more space-efficient than finite automata on some computations[1].

Since the time needed for a computation on finite automata is always linearly related to its input, a natural complexity measure for such models is the number of states of the automata.

Let us first state a simple conversion bound.

Claim 2. Let L be a language recognized by a QFA with n states. Then it can be recognized by a deterministic automaton with $2^{\mathcal{O}(n)}$ states.

We will show that for some languages, the converse holds, that is QFAs can be exponentially more space-efficient than classical finite automata, even probabilistic.

Theorem 3. Let p be a prime. Define the language $L_p = \{a^i, p \text{ divides } i\}$. Any finite automaton recognizing L_p has at least p states. However, for any $\epsilon > 0$, there exists a QFA of size $\mathcal{O}(log(p))$ recognising L_p with probability $1 - \epsilon$.

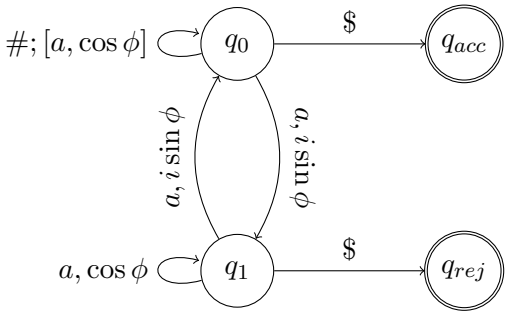

Figure 5: shape of the QFA U_k

Proof. We will construct an automaton accepting words in L_p with probability 1, and rejecting the others with probability at least 7/8. Then, we increase the probability of correct answer to $1 - \epsilon$.

For $k \in \{1, \ldots, p-1\}$, we define the automaton U_k as in figure 5, where $\phi = \frac{2\pi k}{p}$. Then, reading a^j leads to the state $\cos(j\phi)\,|q_0\rangle + i\sin(j\phi)\,|q_1\rangle$

If p divides j, then reading the right endmarker leads to acceptance with probability 1. Thus every U_k accepts words of L_p with probability 1. For $a^j \notin L_p$, we say that U_k is good if it rejects it with probability greater than $1/2$.

Then we show that there exists a subset of size $\lceil 8\ln(p) \rceil$ of the U_k such that sufficiently many are good so that the automaton built with one starting state leading to a uniform superposition of such U_k rejects with high enough probability. We can the transform each u_k to an automaton containing 2^d non-halting states, where d depends on ϵ in order to increase the probability of correct answer. The modified automaton is simply d copies of U_k, such that its state is the tensor products of all the copies. It has $2^d - 1$ rejecting states and 1 accepting states, and only the state corresponding to all copies in the state q_0 leads to acceptance, whose amplitude is $\cos^d(j\phi)$. □

Theorem 4. Any probabilistic finite automaton recognizing L_p with probability $1/2 + \epsilon$, for a fixed $\epsilon > 0$, has at least p states.

Proof. Suppose that such an automaton exists. Since it uses a single-letter alphabet, it can be viewed as a Markov chain. Splitting ergodic and transient states, we reduce our study to a Markov chain containing only one ergodic set, as two or more can be studied separately.

Then, we can bound the probability distribution on the accepting states, contradicting the law of large numbers for Markov chains stated in [3] (Theorem 4.2.1). □

However, it was also shown that QFAs are not always space efficient. In particular, [2] contains a proof that a specific language takes almost exponentially more states to be recognized on a QFA than on a DFA.

5 Conclusion

In contrast to more general models, such as quantum circuits, the benefit of quantum automata over their classical counterparts is very limited, and they will most certainly not yield a revolution in computing. However, the properties presented in [1] are peculiar enough to hint that they are still complex objects whose behaviour can surprise us, as the results of theorems 1 and 2. As such, they are interesting to study as "limited quantum computers", to precisely analyse the contribution of quantum mechanics to computation. In particular, theorems 3 and 4 show that in spite of Holevo's bound, that states that we can only retrieve one classical bit from a qubit, the infinite space that it can span can still be used to increase efficiency. As stated by the authors, such results could help us devise quantum algorithms by extending QFAs or using them as subroutines.

References

[1] A. Ambainis and R. Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations, 1998.

[2] A. Ambainis, A. Nayak, A. Ta-Shma, and U. Vazirani. Dense quantum coding and a lower bound for 1-way quantum automata, 1998.

[3] John G. Kemeny and James Laurie Snell. *Finite Markov chains*. Undergraduate texts in mathematics. Springer, New York, 1976. Reprint of the 1960 ed. published by Van Nostrand, Princeton, N.J., in the University series in undergraduate mathematics.

[4] A. Kondacs and J. Watrous. On the power of quantum finite state automata. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 66–75, Oct 1997.

[5] C. Moore and J. Crutchfield. Quantum automata and quantum grammars. 1997.