A Surface Level Dive into the Fundamentals of Quantum Finite Automata (QFA)

Authored By Tyler Rimaldi and Murray Coueslant

Motivation **Tyler**

Before QFAs were discovered, there did not exist a framework to computationally demonstrate properties of quantum mechanics. As Richard Feynman (Yuling Tian and Zhou) suggests, there simply wasn't an efficient way of simulating a quantum mechanical system on a classical computer, further mentioning if computing machines supported the underlying physical rules, then it may be possible to perform simulations efficiently. Shortly after Feynman's remarks, researchers revisited the Church-Turing Principle and defined quantum Turing machines, inspiring other researchers to join, leading to further discoveries such as demonstrating equivalence between quantum Turing machines and quantum circuits, founding polynomial-time algorithms for factoring prime numbers on quantum machines, and implementing square root time database searches (Yuling Tian and Zhou). Since these discoveries, much more progress has been made in the field of study. But how do these things work? Well there are many different varieties and flavors of QFA, but let's touch the surface and explore some of the elementary aspects.

Quantum Finite Automaton **Murray**

A QFA is not analogous to a regular finite automaton, and there is really no way to equate the two. (Kondacs and Watrous (1997)) They are able to recognise a family of languages called the quantum languages, which are not regular or stochastic. If we think of a QFA as a series of matrices, and understand their transition function to produce a probability of the automata to be in an accepting state, then we can begin to see their usefulness. A QFA can be somewhat described using a triplet containing a complex projective space, an alphabet, and the set of transition matrices for each symbol in that alphabet. In this way, one can describe how the automata would transition through the space according to some input string. Alas, quantum states are not trivial to measure, and as such bring with them an element of probabilism. A QFA may only describe the probability of accepting or rejecting a string, and cannot say outright that it will or will not. Pairs of states in a QFA may also be orthogonal or orthonormal. An orthogonal pair means that upon measurement in one state, there is zero probability that the automata will be in the other state upon further measurement. For two states to be orthonormal, they must also be orthogonal and their state vector must be of unit length. At a quantum level, systems like QFAs are extremely susceptible to the smallest amount of interference, even observing the system causes changes due to Heisenberg's uncertainty principle, and as such the real world applications are often messy and imprecise, but usable enough that many applications of these systems have indeed been found.

Quantum Computing **Tyler**

The QFA, the quantum version of a DFA, is created by changing the basic elements of a DFA to the quantum components as mentioned in the previous paragraph. In the paper titled "Experimental demonstration of quantum finite automaton", researchers expose benefits of using a QFA over DFA with a promise problem (input is promised to belong to a particular subset of all possible inputs) (Benioff (1980)). The task assumes a input string of length k*P or k*P+R, where P is a prime number, R is a positive integer less than P, and k is an arbitrary integer greater than or equal to 0. The promise problem addresses the following: Can it be determined that the length of an input string is divisible by P with no remainder or with a remainder of R?

By using QFAs, this problem can be solved by using three orthonormal bases, which represents one qutrit. "Inside the QFA is a three-dimensional quantum state, with $|0\rangle\,|1\rangle\,|2\rangle$ being its 3 orthonormal basis states, where $|0\rangle$ is both the start state and the only accepted state. The quantum state will go through n copies of unitary operator U, where n is the length of the input string." So what's the big deal? QFAs and DFAs are highly similar, in fact they have the same time complexity because each transition will happen n times, namely the size of the input string. But what is novel about the QFA is it's space complexity. Given the quantum states, by definition these states can exist in superposition and need to not be orthogonal–where as a DFA must have orthogonal states. With the setup illustrated in the experiment, the DFA solves this same problem in P orthonormal states, whereas a QFA does it in just 3 (Benioff (1980)).

Optimized Database Lookup Time **Murray**

There exist quantum algorithms which can achieve database lookup in square root time complexity. Algorithms, such as Grover's algorithm (Grover (1996)), provide us the ability to find an element in an n element database in $O(\sqrt{n})$. The algorithm produces a correct 'result' with a probability of less than one that it is the actual item you were searching for. On top of that, the algorithm is unusable in actual database systems because of the idea of an 'oracle', which is used in modelling decision problems like this and requires at least linear time complexity. However, it was shown that any quantum algorithm aiming to solve problems like Grover's search problem are strongly bound to square root time, and as such Grover's is an optimal algorithm for this task.

Conclusion **Tyler**

It is evident that QFAs are critical to providing a framework in exploring quantum mechanics and quantum computations.There are a number of different theoretical definitions and implementations of QFAs that are based on the underlying principles of finite automata, deterministic finite automata, nondeterministic finite automata, and more closely related, probabilistic finite automata. We have provided merely a surface level exploration of QFAs and their fundamental properties. Nonetheless, It is clear that as classical computers helped with exploring classical theory of computation, quantum computing plays a pivotal role in exploring the theory of quantum computation. The discovery of QFAs has played a catalytic role in enabling researchers from across disciplines of physical and theoretical sciences to develop new questions and definitions with respect to quantum mechanics research.

References

P. Benioff. *The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines*, volume 22. Springer, 1980.

L. K. Grover. A fast quantum mechanical algorithm for database search. In *STOC '96*, 1996.

A. Kondacs and J. Watrous. On the power of quantum finite state automata. In *Proceedings of the 38th IEEE Conference on Foundations of Computer Science*, pages 66–75, 1997.

M. L. S. Z. Yuling Tian, Tianfeng Feng and X. Zhou. Experimental demonstration of quantum finite automaton. volume 5. URL