

Chapter 4

Quantum Automata with Open Time Evolution¹

Mika Hirvensalo
University of Turku, Finland

ABSTRACT

In this paper, a model for finite automaton with an open quantum evolution is introduced, and its basic properties are studied. It is shown that the (fuzzy) languages accepted by open evolution quantum automata obey various closure properties. More importantly, it is shown that major other models of finite automata, including probabilistic, measure once quantum, measure many quantum, and Latvian quantum automata can be simulated by the open quantum evolution automata without increasing the number of the states.

1. INTRODUCTION

1.1. Finite Automata

Finite automata are models of computation with a finite memory. For an exposition on finite automata, see (Eilenberg, 1974) or (Yu, 1997), for instance, but a brief description of finite automata consists of an input alphabet, the transition function, and a finite set of states, some of which are specified *initial*, some *final*. A finite automaton is *deterministic*, if there is only one initial state, and the transition function specifies always a single

state, not a set of states. There are many variants of nondeterministic finite automata, some of which we will later treat in more detail.

The computation of a finite automaton is *real time* computation, meaning that the *input*, which is a string (called *input word*) over the input alphabet, is given to the automaton one letter at time. Before the computation begins, the automaton is set in an initial state, and once a letter is received, the transition function specifies how the state changes. After reading the whole input word, the state of the automaton specifies the outcome of the computation. If the input word leaves the automaton in a final state (can be also called an *accepting* state), then we regard the input word as

DOI: 10.4018/978-1-4666-1574-8.ch004

accepted, otherwise *rejected*. The set of accepted strings is called the *language* accepted by the finite automaton. In general, any set of words is called a language, but languages which are accepted by deterministic finite automata, are called *regular*.

Hence finite automata are machines classifying words as accepted or rejected, or equivalently, machines answering to computational yes/no-questions. Restricting the computation to yes/no-problems is not a major restriction itself, as many computational problems demanding a larger output set can be relatively easily split into a sequence of yes/no-problems. On the other hand, further study (see e.g., (Eilenberg, 1974) or (Yu, 1997)) shows that the deterministic finite automata are far too simple to solve any complex problems.

Nevertheless, researchers have shown interest on finite automata for many decades, despite of their weak computational power. Indeed, finite automata are not interesting because they would resolve difficult computational problems (which they do not), but because the model is very elegant and simple, bearing notable closure properties (union, intersection, complement, Kleene star) and a variety of characterizations ranging over logic and algebra (Eilenberg, 1974). Furthermore, the theory of finite automata has numerous applications in numerous fields including pattern recognition, image compression, and DNA sequencing, for instance (see references in (Yu, 1997)).

The usual resource counted in finite automata computation is the number of automata states: the less, the better. In this article, we will show that the introduced variant can simulate many other models without increasing the number of states.

1.2. Quantum Computing

Any implementation of computation is a physical process anyway, and hence the nature of the physical system used for performing the computation may be of interest. In a sense, this was acknowledged already by von Neumann (Neumann, 1927), but introduced in the connection of computing as

late as in early 1980's by Paul Benioff (Benioff, The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines, 1980), (Benioff, 1982) (see also (Manin, 1980)). The relationship between computing with classical and quantum information computing was further reconsidered by Richard Feynman (Feynman, 1982), who presented a hypothesis that simulation of a *quantum system* may be impossible for a classical computer without an exponential slowdown.

The history of quantum computing, which begun roughly on 1980 contains several brilliant discoveries and a huge amount of detailed research work. To mention some outstanding achievements, Shor's polynomial time algorithm for factoring integers (Shor, 1994) and Grover's Algorithm (Grover, 1996) for unstructured quantum search quadratically faster than any classical search are among the most notable ones.

The early theoretical models for quantum computing included quantum Turing machines, first introduced by David Deutsch (Deutsch, 1985) and subsequently improved by Bernstein and Vazirani (Bernstein & Vazirani, 1997). Finite automata for quantum computing were introduced only several years later, and their history will be briefly introduced in the next section, where also formal specifications of finite automata will be represented. For a broader representation on quantum computing, we refer to (Hirvensalo, 2004).

2. PRELIMINARIES ON FINITE AUTOMATA

2.1. Classical Finite Automata

The computation of a finite automaton is described in the introduction, and here we give the formal definition. A *finite automaton* is a quintuple $\mathcal{F} = (Q, \Sigma, I, \delta, F)$, where Q is a finite set of *states*, Σ is the *alphabet*, $I \subseteq Q$ is the set of *initial states*,

14 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/quantum-automata-open-time-evolution1/66771

Related Content

Linking Interactome to Disease: A Network-Based Analysis of Metastatic Relapse in Breast Cancer

Maxime Garcia, Olivier Stahl, Pascal Finetti, Daniel Birnbaum, François Bertucci and Ghislain Bidaut (2011). *Handbook of Research on Computational and Systems Biology: Interdisciplinary Applications* (pp. 406-427).

www.irma-international.org/chapter/linking-interactome-disease/52327

Quantum Automata with Open Time Evolution

Mika Hirvensalo (2010). *International Journal of Natural Computing Research* (pp. 70-85).

www.irma-international.org/article/quantum-automata-open-time-evolution/41945

Automatic Test Data Generation Using Bio-Inspired Algorithms: A Travelogue

Madhumita Panda and Sujata Dash (2018). *Handbook of Research on Modeling, Analysis, and Application of Nature-Inspired Metaheuristic Algorithms* (pp. 140-159).

www.irma-international.org/chapter/automatic-test-data-generation-using-bio-inspired-algorithms/187685

A Synthesis Method of Gene Regulatory Networks based on Gene Expression by Network Learning

Yoshihiro Mori and Yasuaki Kuroe (2010). *Handbook of Research on Computational Methodologies in Gene Regulatory Networks* (pp. 266-288).

www.irma-international.org/chapter/synthesis-method-gene-regulatory-networks/38239

Optimising Prediction in Overlapping and Non-Overlapping Regions

Sumana B.V. and Punithavalli M. (2020). *International Journal of Natural Computing Research* (pp. 45-63).

www.irma-international.org/article/optimising-prediction-in-overlapping-and-non-overlapping-regions/241941