G53NSC Non-Standard Computation

Coursework

QCA – Quantum Cellular Automata

Bc. Jiří Kremser *jxk19u*
Bc. Ondřej Božek *oxb09u*

Academic Year 2009-2010

Abstract

Quantum Cellular Automata (QCA) is computational model based on classical cellular automata (CA), reversible computation and exploiting quantum phenomena. The aim of this paper is to give a reader insight to general concepts of QCA, as well as to show some models, especially Quantum dot QCA, however it does not exploit quantum phenomena. Since QCA is generalization of classical CA, the history and fundamentals of CA will be described. Some examples of emergent complex behaviour on the basis of a set of simple rules will be discussed on John Conway's Game of Life.

1 Introduction

The concept of Cellular Automata comes originally from John von Neumann and is spread across many fields. It can model any complex closed (rule-based) systems if we accept the constraint that the time and space are discretized to time steps and cells. These models are intuitive in contrast with the system of differential equations, which, however, preserve the continuity of the nature. Since these models can model physical laws and on the subatomic level we can observe quantum phenomena with all its weirdness, it is obvious to extend this concept to quantum world. Moore's law implies, amongst other, the computer circuits will get smaller and smaller. Rather than avoiding the quantum phenomena it is better to exploit them and make quantum exponential speed-up possible. QCA is theoretical computation model and this paper is summary of this problematic.

2 Classical Cellular Automata

Formally, CA is a 4-tuple (C, Σ, N, f), where C denotes an d-dimensional array of cells or lattice (cells are indexed by vectors from $\mathbb{Z}^N$), Σ denotes the alphabet, giving the possible states each cell may take, N denotes the neighbourhood (i.e. $N \subset Z^d$) and f denotes the transition function of type $\Sigma^N \rightarrow \Sigma$. The state of all cells in time is called configuration. Significant configuration is the starting configuration, since it has to be provided with the CA.

CA is discrete computational model, which is capable to provide the same computational power as Turing Machine ™, therefore it is Turing Complete. CA were probably firstly used by famous scientist Jon von Neumann in late 1940s when he was trying to describe self reproducing automaton. He succeeded by introducing two dimensional Von Neumann's cellular automata with rules (f function from definition) and starting configuration such that after certain amount of time steps there were two copies of the pattern from starting configuration and so on. Later on in 1980s Stephen Wolfram in his famous book *New Kind of Science*[14] defined four classes of cellular automata depending on complexity and predictability of their behaviour.

2.1 Wolfram Classes

Wolfram showed his classes on one dimensional CA, where neighbourhood of particular cell were the cell on the left hand side and the cell on the right hand side. There were only two states of cells – zero/one. Consequently there were only eight possible combinations of states of a cell and its neighbour cells. Generally it is a^b where a denotes the cardinality of Σ and b the size of the neighbourhood. By application transition function f on each of these eight triplets, it could make the cell either 0 or 1 in the next time step. This set of rules can be rewritten as an 8 bit string, where each bit denotes the return value for arguments 111, 110, ..., 000 as shown at *Figure 1*.

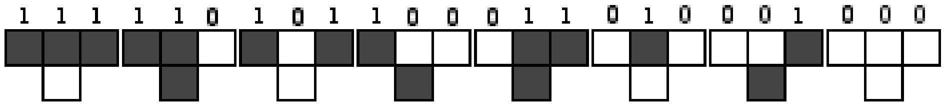


Figure 1: Rule 90 (90dec = 01011010bin)

Running CA with Rule 90 with starting configuration ...000000010000000... (one zero in the center) will “draw” a Sierpiński triangle as shown at *Figure 2*.
Rows represent the development of configurations in time. The starting configuration is on the top.

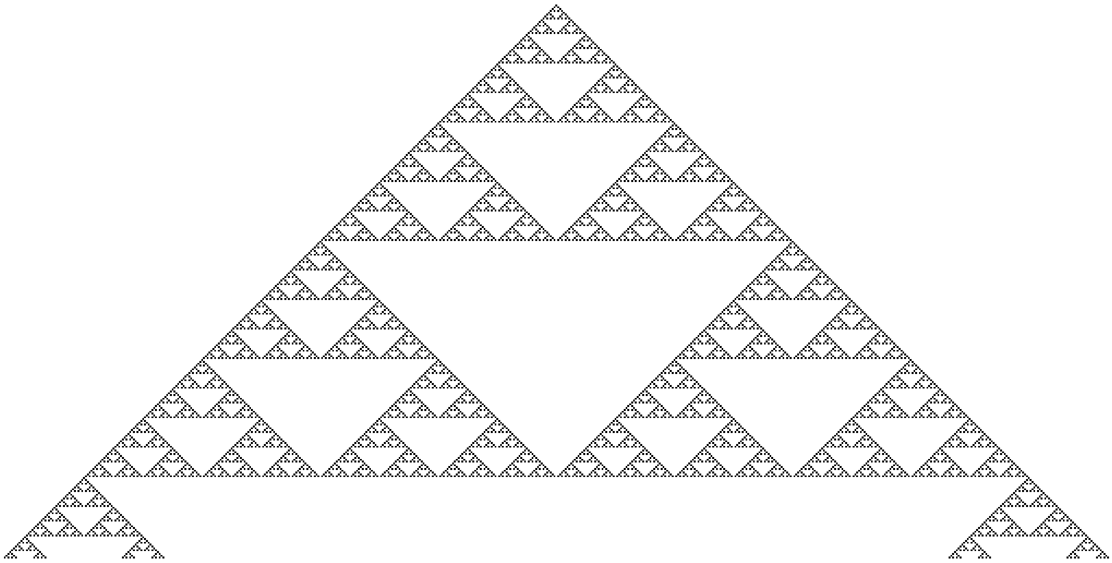


Figure 2: Progress after ~300 iterations.

a) Class 1

Evolution in time leads to some stable configuration. If there is no change between two configurations, this configuration will never change, because the transitional function f may depend only on cells and its neighbourhoods (not on time).

b) Class 2

Evolution in time leads to some periodic patterns. It is not stable, but still very well predictable (“Rule 90”).

c) Class 3

Chaos behaviour. Without simulating it is impossible to predict the future development. The laws of Chaos Theory can be applied here.

d) Class 4

On the edge of chaos. Characteristics are between class 2 and 3, some regions behave randomly, some are relatively stable or predictable.

Research assistant of Wolfram's Matthew Cook showed and proved[1][9] that one dimensional CA with “Rule 110” is universal computational model. The same result was achieved with John Conway's Game of Life[10]. Both these CA are members of Wolfram's class 4. Providing the Curch-Turing thesis holds, these models are equivalent and can simulate every computable function.

2.2 Game of Life

Introduced by British mathematician John Conway in 1970. It is probably the most known example of a CA. The Conway's Game of Life (GoL) is a 2-dimensional CA with neighbourhood made of eight adjacent (one cell horizontally, vertically or diagonally) cells. The state of cell can be either life (1) or dead (0). The evolution of each cell in time depends only on the states of its neighbours.

Schema for transition function:

- A dead cell comes to life if it has exactly three living cells in its neighbourhood.
- A cell remains living if it has two or three living cells in its neighbourhood.
- A cell dies otherwise. (on overpopulation or loneliness)

Conway discovered some interesting patterns and shapes in his game and divided them into these following categories.

a) stable or periodic repeating – stable patterns survive from generation to generation without changing, whilst periodical patterns (*oscillators*) repeats after certain number of time steps – period *p*. **Block** (p = 1) and **Blinker** (p = 2) are shown at *Figure 3.*

b) spaceships – they repeat its shape after certain number of time steps, but on the position next to original. The most famous spaceship and key pattern in *GoL* is the **Glider** shown at *Figure 3*.

c) guns – “fires” spaceships

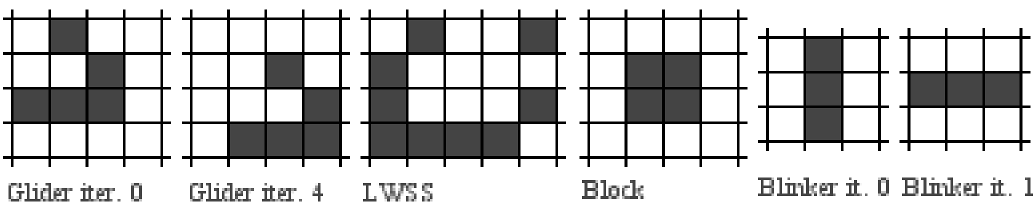


Figure 3: Some basic GoL patterns.

1 He proved it by emulating another universal model – Tag System [8] on this 1-dim. CA.

Gliders are able to move diagonally across the 2D grid. LWSS (Lightweight spaceship) moves vertically or horizontally according to its rotation. Using gliders and other patterns it is possible to implement Minsky Machine or Turing Machine so it is universal as well as the 1-d CA with “Rule 110”. An example of simple TM implemented as GoL can be found at [15]. Furthermore, a configuration may contain a collection of guns that fire gliders in such a way as to construct new objects, including the copies of some original patterns. A "universal constructor"[11] can be built this way, which contains a Turing complete computer, and which can build many types of complex objects, including more copies of itself, like Von Neumann's self reproducing CA.

3 Reversible Cellular Automata

Global transition function F maps an arbitrary configuration to another one by applying the local transition function f to all cells.. A CA is said to be reversible cellular automaton (RCA), if and only if his global transition function F is bijective. In other words, each configuration has exactly one preimage according to F. Configuration which has no preimage[2], informally called Garden of Eden, appears when F is not surjective. John Myhill and Edward Forrest Moore proved[3], that if CA is injective then it is also bijective (surjectivity implies injectivity), thus reversible. There are 16 RCA out of 256 1-dimensional CA with neighbourhood size of one. For dimension grater or equals to two it is undecidable, whether an arbitrary CA is reversible. Luckily Tommaso Toffoli proved[12], that any n-dimensional CA can be simulated by n+1-dimensional RCA in 1977. In 1990 Toffoli and Norman H. Margolous introduced Partitioning Cellular Automata which can be build up from classical CA by partitioning technique[13]. The reversibility is important in the world of quantum mechanics, since Landauer’s principle can predict the energy consumption within closed systems.

4 QCA

It seem obvious that cellular automaton are themselves physics-like models of computations. These models are not continuous but represents space as simple lattice. In CA Everything is discrete including time, which jumps discontinuously. In the late 70s, Fredkin proposed that the world we live in is a huge cellular automaton. Fredkin s thought that all physical quantities can be seen as packets of information in a cellular automaton. Therefore it seems natural to study quantum extensions of CA as world from the most detailed view seems quantum-based at the moment. There is a big advantage in QCA models in comparison with other models of quantum computation. Principle of computation is in quantum cells interaction. Need for for environment interaction is reduced, and so is possibility of decoherence, the main obstacle for realization of a quantum computer. Cells are not required to be able to distinguish one neighbour from another. In cellular automaton uniform rules are applied in parallel across a whole lattice, therefore it is not needed to address each cell (qubit) separately.

2 They could not been created by application of F on any configuration.

3 Garden of Eden theorem.

This helps to eliminate errors resulting from cross talk on neighbouring cells (qubits) known from other models of quantum computational machines. Another benefit of CA framework is that many fabrication techniques naturally produce equally spaced units suitable as base for lattice used in cellular automata computation. Physical systems proposed as framework for QCA include quantum dot arrays and endohedral fullerenes.

4.1 History

- 1965 the first example of a Quantum Cellular Automata was Feyman's "quantum checkerboard" model of spinors in 2d space-time. Feynman invented the model in the 1940s while developing his space time approach to quantum mechanics. He did not publish the result until it appeared in a text on path-integrals co-authored by Albert Hibbs. It's interesting to note that Feynman's work on QCA was quite a long time before his seminal talk on "Simulating Physics with Computers" that is often cited as the beginnings of the field of quantum computation.

- 1988 Grossing and Zeilinger attempted to introduce the concept of quantum cellular automata, however their model has little in common with the models currently in use.

- 1990 Norm Marglous wrote Parallel Quantum Computation. Feynman and others have shown that the quantum formalism permits a closed, microscopic, and locally interacting system to perform deterministic serial computation. In this paper Marglous show that this formalism can also describe deterministic parallel computation

- 1995 the first successful model of one dimensional quantum cellular automata was due to Watrous.

4.2 What is QCA in fact?

There is no only generally accepted QCA model. There are many different definitions of QCA. Many of them seem to be computationally equal. But one unique, computationally powerful definition is missing. There is no such axiomatic definition, unlike its classical counterpart, that can immediately bring up way how to construct or enumerate all the instances of this model. Each set of authors defines QCA in their own particular way. The main common signs of various QCA definitions consists of a d-dimensional lattice of identical finite-dimensional quantum systems, a finite set of states, a finite neighbourhood scheme of single cell, and a set of local unitary transition rules. The states $s \in \Sigma$ are basis states spanning a finite-dimensional Hilbert space. Basis or basic states are initial states in which is each cell before computaiton starts. At each point in time a cell represents a finite-dimensional quantum system in a superposition of basis states. The global evolution function represents the discrete-time evolution of strictly finite cell lattice. There is demand on this evolution function to be unitary.

5 QCA types

5.1 Grössing-Zeilinger QCA

Grőssing and Zeilinger introduced the term "quantum cellular automata" and they were also first which attempted to create appropriate model. This pioneering definition of QCA, however, has not been studied much further, mostly because the "non-local" behaviour makes the Grőssing-Zeilinger definition non-physical. In addition, it has little in common with the concepts developed in quantum computation later on. The Grőssing-Zeilinger definition really concerns what is called today a quantum random walk.

5.2 LQCA - Watrous QCA

The first model of QCA researched in depth was that introduced by Watrous. A Watrous-QCA is defined over an infinite 1-dimensional lattice, a finite set of states including a quiescent state. The transition function maps a neighbourhood of cells to a single quantum state simultaneously over whole lattice. This model of Quantum Cellular Automaton is in literature also referred to as 1d-QCA or Linear Quantum Cellular Automaton (LQCA). Following definition is adapted from [3].

Definition (LQCA). A linear quantum cellular automaton is a 4-tuple $A = (\Sigma, q, N, \delta)$, where (with $q\Sigma = \{q\} \cup \Sigma$):

- Σ is a finite set of symbols (i.e. "the alphabet", giving the possible basic states each cell may take);
- q is a symbol such that $q \notin \Sigma$ (i.e. "the quiescent symbol", which may be thought as a special state for empty cells);
- N is a set of n successive signed integers (i.e. "the neighbourhood", telling which cell is next to whom);
- $\delta : \acute{H}_{(q\Sigma)n} \rightarrow \acute{H}_{q\Sigma}$ is a function from super-positions of n symbols words to super-positions of one symbol words (i.e. "the local transition function", describing the way a cell interacts with its neighbours).

Moreover δ must verify:

- the quiescent stability condition: $[\delta|q^n\rangle) = |q\rangle]$;

- the no-nullity condition: $\forall w \in (q\Sigma)^n$, $[\delta|w\rangle \neq 0]$.

In this definition H_Σ denotes the Hilbert space filled with the cell states Σ. Set of possible configurations of the CA is extended to include all linear super-positions of the classical cell configurations. Similarly to their classical counterparts Watrous Quantum Cellular Automata, or LQCA in other words, consist of a row of identical, finite dimensional, quantum systems. One cell is labeled "accept" cell. The quiescent[4] state of allows only a finite number of cells to be active and thus makes the lattice finite. Finite lattice is essential to avoid an infinite product of global evolution (more thereinafter) and, to obtain a well-defined QCA. These rows evolve in discrete time steps by means of local transition function. Transition function maps the cell configurations of a given neighbourhood to a quantum state. Homogenous and synchronous application of transition function gives rise to global evolution Δ.

In order to make LQCA physically acceptable model of computation, it must be confident that the global evolution Δ is physically acceptable in a quantum theoretical setting, it must be certain that Δ is unitary. It is possible to define transition functions that do not represent unitary evolution of the configuration. There are two properties that transition function must not have in order to construct unitary evolution. Firstly it must not produce super-positions of configurations which do not preserve the norm, global evolution preserves the sum of probabilities squared to 1. Secondly they must not include a global transition function which is not unitary. This leads to non-physical properties such as super-luminal[5] signalling. Unfortunately this requirement for global evolution function to be unitary is really non-trivially related to the definition of the transition function δ. (The set of LQCA is not closed under composition and inverse.)

It would help us a lot if there would exist a function which can efficiently decide if given transition function δ constructs global evolution function which is unitary. Such a function could be than applied to whichever local transition function. This would help us to construct particular LQCA, suitable for simulation of physics, much easier. Number of local transition functions which do induce a unitary global evolution is likely to be rather scarce.

There is tendency in computer science to decide if program is valid, by means of syntactical correctness. Its obvious that once programing language is universal, adding more expressiveness does not mean adding more computational power, but it only allows us to express something easier and in more than one way. There is a catch, if we enrich syntax of the language so much that it allows us to create non-valid (unrealistic, non-physical) programs. So the user needs to performs non-trivial (non-syntactic) decisions to recognize and exclude those non-realistic variants and the process of program creation is significantly more complicated. Such a programing language can be considered too lose. This is analogous to current state of LQCA formalism.

4 Quiescent means inactive, passive or not participating.

5 Super-luminal signalling is passing of signals faster than is speed of light.

Therefore there is a necessity to tighten the definition of linear quantum cellular automata. We need to find more restrictive definition whose unitarity may be checked algebraically or syntactically, but computational power must remain same. Pablo Arrighi managed to algebraically characterize unitary LQCA by adding constraints into the model which do not change the quantum cellular automata computational power. This is step towards possibility of algebraical verification of local transition function. But there is still a long way ahead.

5.3 Partitioned Quantum Cellular Automata (PQCA)

As mentioned above in order to have physically acceptable model of computation we need to fairly uneasily decide the right local transition function. Transition function has to construct unitary global evolution function. If we want to avoid this nontrivial decision process, some restrictions on LQCA need to be introduced. Partitioned Quantum Cellular Automata is such kind of restricted LQCA that allows us to construct well-formed QCA easily.

One-dimensional Partitioned Quantum Cellular Automata is 1d-QCA where each cell in whole lattice is partitioned in to three sub-cells. The next state of any cell now depends only on the states of the left sub-cell of the right neighbour, the middle sub-cell of the cell itself, and the right sub-cell of the left neighbour.

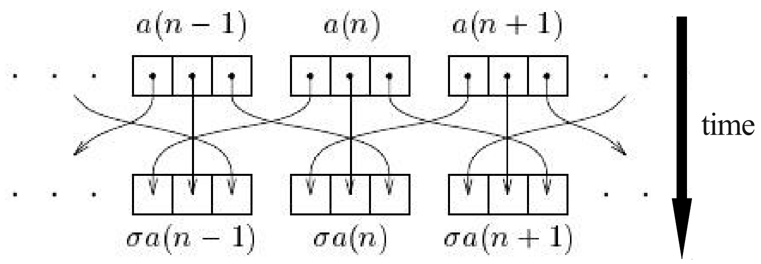


Figure 4: evolution of cells in 1d-PQCA [2]

It was shown that any Quantum Turing Machine can be simulated by PQCA with constant slowdown and every PQCA can be simulated by QTM with linear slowdown.[2] This is also proof of computational universality of PQCA. Thence this restriction does not reduce computational power of plain LQCA more than acceptable.

5.4 Quantum-Dot Cellular Automata

Quantum-Dot Cellular Automata (QdCA), often called only QCA, are classical CA implemented in quantum mechanical structures. They do not exploit quantum effects for the actual computation. QdCA is a hybrid of a quantum circuit with individual qubit control and with constant nearest-neighbour interaction. The cell in this automaton consists of four quantum dots forming a square. Electrons can tunnel between dots, but cannot leave the cell. If two excess electrons are in the same cell, Coulomb repulsion will force them to dots on opposite corners. As a corollary, there are two ground states representing zero and one as shown at *Figure 5*.

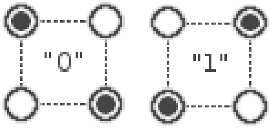
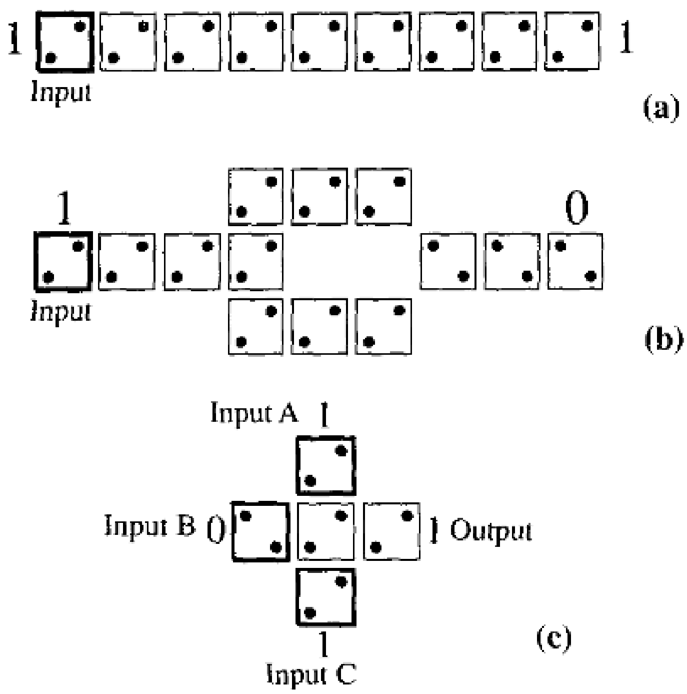


Figure 5: Q-dot ground states.

Coulombic interaction between electrons cause the cells of QdCA to take the same polarization, if two cells are close enough together. Using this principle, it is easy to build circuits and logic gates from these uniform cells.

At the figure bellow there are shown a) line of cells representing a wire, b) inverter representing operation NOT and c) majority gate.



a) When some ground state is set to the input of the line (wire) on the left hand side, the other cells tend to take the same polarization to decrease energy. In the state, when all cells have the same polarization, the electrons are as widely separated as possible.

b) The input is split into two lines and then other line is connected in the angle of 45°. Again since the cells are capacitively coupled to their neighbour cells, the output line will take opposite polarization than the input one. Thus this circuit is realizing logical function **NOT**.

c) The majority gate sets output to one, if most of inputs are set to one, zero otherwise. If input C (control input) is set to one, then output is equal to A **OR** B. If the C is set to zero, then output is equal to A **AND** B.

This set of logic operators (even without OR or AND) is universal set. Each other logic function can be build up from this basic set.

6 Conclusion

The concept of Quantum Cellular Automata is quiet young. It is currently in its early phase. Many of important definitions appeared most recently. As obvious from this paper many accurate definitions are also still missing. Many branches of this young discipline are waiting for exploration. For example multi-dimensional QCA were researched very slightly, although they for sure hide many interesting and beneficial findings. Now we can see less than tip of an iceberg. Many interesting discoveries can be expected in near future.

QCA formalism seems as promising framework for building robust and scalable computers suitable for quantum computations. They overcome some technical obstacles known from other approaches. Difficulties with manipulation of individual quantum registers and related decoherence of adjacent qubits induced by cross-talks are surpassed by global evolution of QCA without need for addressing of individual cells.

CMOS technology is approved industry standard for VLSI devices for past decades. In years to come CMOS will impose its fundamental limits. Quantum Dot Cellular Automata is one of the many proposed replacements. QCA resolves all problems of current CMOS technology, but also brings its own.

Only feasible implementation method for mass production of QCA devices is molecular QCA with inter-dot distance of 2 nm and an inter-cell distance of 6 nm. However this technology is in the presence limited by availability of its practical fabrication methods[16].

7 Resources

[1] *Simulating Physics with Computers*, Richard P. Feynman, 1981

[2] *On One-Dimensional Quantum Cellular Automata*, John Watrous, 1995?

[3] *Algebraic Characterizations of Unitary Linear Quantum Cellular Automata*, Pablo Arrighi, 2006

[4] *Parallel Quantum Computation*, Norman Margolus, 1994

[5] *Quantum Cellular Automata*, Bassam Aoun, Mohamad Tarifi

[6] *Quantum Mechanics and Path Integrals*, Feynman and Hibbs, New York: McGraw-Hill, Problem 2-6, pp. 34-36, 1965

[7] *Quantum Cellular Automata*, Karoline Wiesner, 2008

[8] *Universality of Tag Systems with P=2*, Cocke, J. and Minsky, M. J. Assoc. Comput. Mach. 11, 15-20, 1964.

[9] *Universal Cellular Automata and Class 4*, A. Dhar, P. Lakdawala, G. Mandal, S. R. Wadia, 1994.

[10] *The fantastic combinations of John Conway's new solitaire game "life"* , Martin Gardner , Scientific American 223 (October 1970): 120-123.

[11] *Construction Theory, Self-Replication, and the Halting Problem*, Hiroki Sayama, Journal-ref: Complexity 13(5):16-22, 2008.

[12] *Computation and construction universality of reversible cellular automata,* Tommaso Toffoli, J. Comp. Syst. Sci. 15 (1977), 213-231.

[13] *Invertible cellular automata: A review.* Tommaso Toffoli and Norman H. Margolus. Physica D: Nonlinear Phenomena, 45:229–253, September 1990.

[14] *A New Kind of Science*, Wolfram, Stephen, 1st edn.. (Wolfram Media, 2002).

[15] *A Turing Machine in Conway's Game of Life, extendable to Universal Turing Machine*, Paul Rendell, 2000, URL: http://rendell-attic.org/gol/tm.htm

[16] *Quantum dot cellular automaton*, Wikipedia.org, 2010, URL: http://en.wikipedia.org/wiki/Quantum_dot_cellular_automaton