International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol. 4, No. 2, 2017, pp. 180-189.
ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

The study of cellular automata and compare them

Rahebeh Ghasemzadeh

Department of Computer Science, Shahindezh Branch, Islamic Azad University, Shahindezh, Iran.

Abstract

Cellular automaton provides an excellent environment for performing complex calculations using simple rules, getting help only from local information. Cellular automaton is divided into a number of cells, each of which can have one or more final state. The cells are affected by the neighbors with simple rules. The present article discusses the various types of cellular automata, and also provides a review of the existing literature of certain methods used by researchers to use cellular automata in the formatting of the purposes. One can use a combination of a variety of cellular automata with fuzzy logic in the fields of artificial intelligence.

Keywords: Cell, Cellular Automaton, Simple Rules.

Introduction:

The basic concept of cellular automata (CA) is certainly associated with the name of the scientist and mathematician, John von Neumann. Von Neumann was deeply interested in the relationships between biology and the new computing devices which is the result of new science. He had more in his mind the theory of automata and the biological phenomena of self-renewals, and he had raised a fundamental question, "What kind of a logical organization is sufficient for an automaton thus to be able to replicate itself?" The idea of using cellular automata as a framework was proposed by Elma so as to be a response to von Neumann's question. Therefore, the original concept of cellular automata can be attributed to Elma, while the initial development of the concept was done by von Neumann (Mitchell, 1998).

A cellular automaton (Wolfram, 1986) is a mathematical framework for various components that together they design simple and complex projects. The behavior of each component is defined by the behavior of its neighbors. Cellular automaton can produce a structure by starting a random state. A cellular automaton consists of two components: the first component is a cellular environment that consists of a regular grid of cells (pixels) and each cell (pixel) can be only in a finite number of possible states (Enayatifar, 2009).

Cells states (pixels) are defined by the previous states of the neighborhood surrounding cells (pixels) and are updated simultaneously in discrete stages of time.

The second component is a transition law $f(\eta_i^t)$ (cellular automaton law), which updates the state of $a_i^{(t+1)}$, similar to a function of η_i^t.

The local law "f "can be defined in the general state by the general rules, integrity and external integrity. In general rule, the next value of a cell is dependent upon the central cell value and the value of the neighboring cells to the current time step. For example

$$a_i^{(t+1)} = f[a_{i-1}^t, a_i^t, a_{i+1}^t]$$

Where $a_i^{(t)}$ indicates ith status of the cell in time tth. The function of "f" is the next state and is referred to as the automaton law (Ganguly, 2001).

In the rule of integrity, the next value of a cell depends upon the various states of the central cell and the neighboring cell. For example

$$a_i^{(t+1)} = f[a_{i-1}^t + a_i^t + a_{i+1}^t]$$

where the total value of the neighboring cells and the central cell are considered.

In the external integrity law, the next value of a cell depends on the current state of the central cell and the states of the neighboring cells. For example

$$a_i^{(t+1)} = f[a_i^t,\ a_{i-1}^t + a_{i+1}^t]$$

where the value of the central cell and the total values of the neighboring cells are considered.

There are various local structures for cellular automata. In general, we can assume any collection of sorted cells as neighbors. Two famous neighbor cellular automata are the von Neumann neighborhood and the Moore neighborhood (Ganguly, 2001), where the von Neumann neighborhood consists of 4 neighboring cells and for border neighbors with the north, south, east and, west, and Moore neighborhood constitute the cell itself and 8 neighbors (the von Neumann neighborhood plus diagonal cells). Fig. 1 shows these two neighborhoods along with two other common neighborhoods.

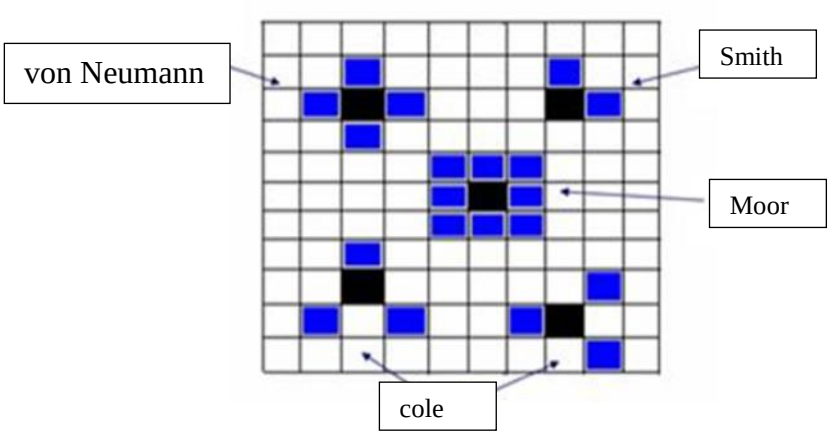


Figure 1: Some famous and common neighborhoods

For example, in a cell automaton, a global clock, provides an updating signal for all cells, in each step, at a moment, all the cells update their states at the same time given $f(\eta_i^t)$.

A one-dimensional cellular automaton k = 2 and Σ = {0, 1} are shown in Fig. 2 (where, Σ means that a set of states is finite in a state automaton cell and k = | Σ | means the number of states in each cell. Each cell is specified with an index of i and its state at time t, as s_i^t, wherein $s_i^t \in \Sigma$). At this point, the neighbors of the cell contain itself and its two closest neighbors and the border conditions are periodic (for example, the leftmost neighbor of the right is considered for the rightmost cell, and vice versa). The size of η_i neighboring for one dimensional cell automaton is often $|\eta i| = 2r + 1$ (Seredynski, 2004) where r is the radius of the cell automaton. In the case of one dimensional, it is $f = \Sigma^{2r+1} \to \Sigma$. In certain cases of the binary cellular automaton that the neighboring possible number is not very large, the cellular automaton rule is often shown like a look-up table (the table of rules), which lists every possible neighborhood along with the output line, and lists the value of the updating of the central cell in that neighborhood. One-dimensional cellular automaton, with 256 rules, names k=2 and r=1 as primary cellular automaton.

For the interpretation of the rules of the cellular automaton, a special contract (Fig. 3), which has been accepted here, will follow.

The central box (Fig. 3) indicates the current cell that the cell under investigation and all other boxes represent 8 nearest neighboring cells. Each of these cells can be considered as a variable. Thus, for the two dimensional cellular automaton, there must be considered 9 variables. Number in each box

determines the number of rule, which indicates the current cell dependence on its own and only specific neighborhood.

Rule 1 specifies the central cell dependence only to itself and dependence to all the neighbors will be determined by rule 510 (the Moore Neighborhood) and so on. These 9 rules are called the elementary rules (Fig. 3), in a state that cell would be dependent upon 2 or more neighboring cells; the number of the resulting rule is the sum of the numbers of the related cells. The number of the produced rules by the combination of 9 variables which consists of the description of the rules according to the independency. If the next state of a cell depends on its current state and its right neighbor, it is known as the Rule 3 (= Rule 1 + Rule 2); since depends on 2 cells. If the next state depends on its current state and all its Moore neighbors, it is known as Rule 511 (= Rule 1 + Rule 2+ Rule 4 + Rule 8 + Rule 16 + Rule 32 + Rule 64 + Rule 128 + Rule 256). All linear rules can be achieved in a similar way (uguz, 2015).

The primary cell automaton is the simplest type of cellular automaton architecture. This elementary architecture can be amended by many methods, such as changing the number of dimensions, changing the number of states of each cell, and changing the neighborhood size (Mitchell, 1998).

The present paper is thus organized as to address the concept of cellular automata, their performance and interpretation of the transition rules in the first part, and, in the second part, all the cellular automata are discussed and the third part would have the conclusions and new research areas for future work.

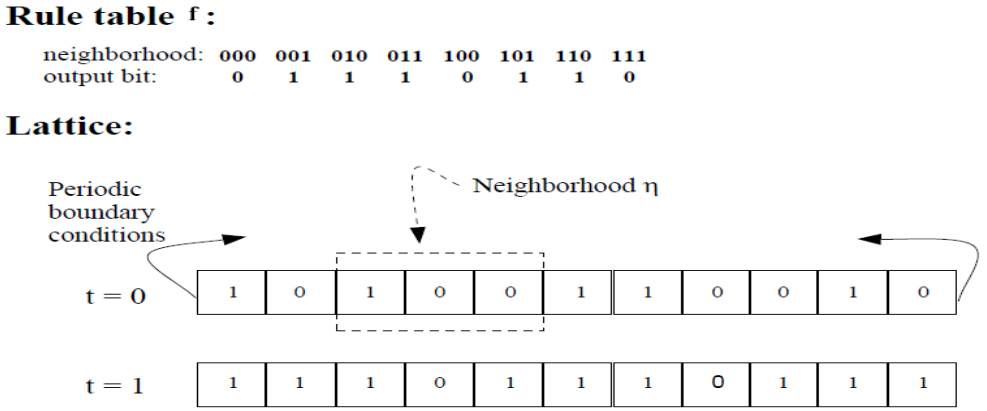


Fig. 2: An example of a one-dimensional cell automaton with r=1 and k=2 for a shown time period

64	128	256
32	1	2
16	8	4

Fig. 3: The central box of the two dimensional cellular automaton

Different types of cellular automata

This section discusses a different types of cellular automata, which have been proposed by various authors thus to simplify the design and formatting of complex systems.

Linear cellular automaton

If the rule of the cellular automaton applies only to the XOR logic, it is then known as the linear rules. A cellular automaton that has all the cells of that linear rule, it is known as linear cellular automaton (Maji, 2003), which is shown Table 1.

Table 1: The rules of linear cellular automaton

Rule No.	Next State Logic Function
Rule 60	$q_i(t+1) = q_{i-1}(t) \oplus q_i(t)$
Rule 90	$q_i(t+1) = q_{i-1}(t) \oplus q_{i+1}(t)$
Rule 102	$q_i(t+1) = q_i(t) \oplus q_{i+1}(t)$
Rule 150	$q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$
Rule 170	$q_i(t+1) = q_{i+1}(t)$
Rule 204	$q_i(t+1) = q_i(t)$
Rule 240	$q_i(t+1) = q_{i-1}(t)$

Where $\oplus$ is the function of XOR.

Learning automaton

Learning automaton (Norendra, 1974) is an automaton that can do a number of things. Methods based on learning automaton have two main stages:

Learning stage: At this stage there is no algorithm to be run, but only the act of learning is supposed to be done. Such as voice recognition of male and female with a high precision, and without recognizing the color of skin, if white, black or red. The more you see, the more you learn.

Test stage: if we make sure that our work result is correct, we give the already-known inputs to the system so as to see whether it has diagnosed correctly or incorrectly.

The extracted knowledge in learning automaton is in the limits of our data, so it can only do a few practices. Any selective action of learning automaton is evaluated in a probabilistic environment and the result is used for the learning automaton with a positive or negative reinforcement signal. The mentioned

signal would affect the next state, which is used by the learning automaton. The final goal of a learning automaton is the learning of the most important action among all actions. The best action is the one that gets the positive signals with the maximum probability from its environment. Fig. 4 depicts the interaction of learning automaton about the environment. Environment can be used as an example of the triple E = {α, β, c} that α={α1, α2, ..., αr} is the set of inputs and β = {β1, β2, … , βm} is the set of outputs and c = {c1, c2 ... , cr} is the set of the likely penalties that in it, ci is the probable that αi takes the penalty. In static environments, the ci values will remain unchanged, while these values in non-static environments during the development of learning automaton will change. Due to the nature of the input, the environment can be divided into three types of P, Q and, S:

In the environment type P: β is a set of {0,1} that β1=1 and β2=0 are the signals of reward and penalty, respectively.

In the environment type Q: β is an arbitrary finite set of distinct signs.

In the environment type S: β is a random value in the range [0,1] of real numbers.

Learning cellular automaton

Learning cellular automaton (Beigy, 2004) is a combination of learning automaton (Narendra, 1974) and cellular automaton (Wolfram, 1986). Learning cellular automaton method is a combination of the ability of learning automaton with the help of cellular automaton window. Learning cellular automaton method is a tool for formatting the simple component systems (Enayarifar, 2009). The behavior of each component in learning cellular automaton is determined from the behavior of the neighbors and the previous experiences of its own. Learning cellular automaton shows the complex behavior through simple interactions of the components. Therefore, the learning cellular automaton is appropriate for formatting the real-world problems.

As can be seen in Fig. 5, the learning cellular automaton is a cellular automaton that each of its cells is equipped with one or more cellular automaton, which determines the state of the related cell. Similar to cellular automaton, in learning cellular automaton the local rule is dominant in the environment and determines the way of action selecting by giving reward and penalty signals.

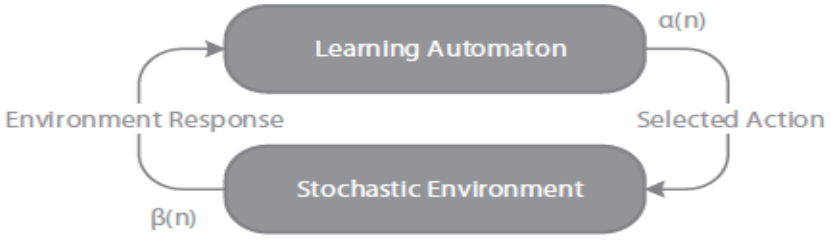


Figure 4: Interactions of learning automaton with the environment

Fuzzy logic

Fuzzy logic is another method of artificial intelligence. The key thought about fuzzy logic is a theory that allows a thing to exist to some extent, instead of having all this or all that. The degree of attachment to a set or a category can be in numerical way, by the value of membership, between 1 and 0. The variables become fuzzy by the use of membership function, which determines the degree of membership to the fuzzy set. These variables are called linguistic variables. Membership functions are curves that determine how a point in the input space to be mapped to a value membership in the interval [0, 1]. These curves can be in different shapes including triangular, trapezoidal, Gaussian curve, and so on. Fuzzy rule-based format is implemented on the base of IF-THEN, where IF is a vector of explanatory variables phase of the area (input) and a THEN is a result of the fuzzy od a dependent variable (output). The user can understand the uncertainty of the data via the fuzzy logic (Uguz, 2015).

Fuzzy cellular automaton (FCA)

Given the ability and the use of fuzzy logic in processing unknown data, fuzzy cellular automaton defines a structure, wherein instead of using the known values in the cells and transfer functions, there are used unknown and fuzzy values. In other words, fuzzy cellular automaton is a cellular automaton, in which there is used the fuzzy logic for the states of the cells and the rules of transfer state (Anvarinejhad, 2004). All states of a cell and local transfer function (rules) in the fuzzy cellular automaton are fuzzy. In fact, all modes of a cell, linguistic variables (understandable) are based on our knowledge of the issue. The next state of a cell depends on its current state and its neighbors. These state changes, according to the transfer function, are local, which is a fuzzy function and it is the same for all cells. This function makes use of the value of the current membership of each of the neighbors so as to calculate its value in the next step. In order to illustrate the development of fuzzy cellular automaton, there is used the value of linguistic variables of cell at any time and the range is the same for all neighboring cells and remain unchanged over time. Fuzzy cellular automaton can be defined as a foursome {z, s, r, f} where Z is a regular network of cells with n dimensions, S is a set of states of each cell in the range [0,1] and the range (r ... N) is the local radius and f: S^{2r+1} is the fuzzy transfer function.

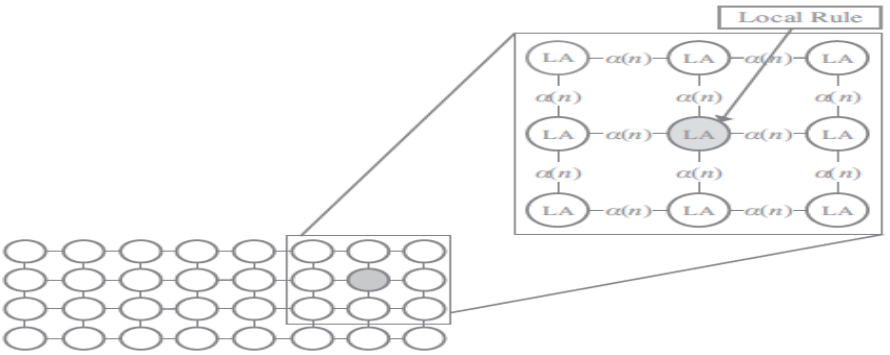


Figure 5: Learning cellular automaton in a cellular environment with Moore neighborhood

Complementary Learning automaton

If the cellular automaton rule contains only XNOR logic, then it is called complementary rule. A cellular automaton with all cells that has complementary rules is called the complimentary cellular automaton (Krishna, 2012).

Additive cellular automaton

Cellular automata that have a combination of XOR and XNOR rules are called additive cellular automata. Of new features of this cellular automaton are the tool that is based on additive cellular automaton, and is used for error diagnosis, and a diverse range of practical applications that can be used to solve real-life problems (Krishna, 2012).

Reversible cellular automaton

A cellular automaton is called a reversible cellular automaton from this aspect that it always returns to its original state. An interesting feature of being reversible is not that it just goes backward, but the inverted repeat is also possible. Using reversible rules makes this possibility to recover from any point, anywhere. One rule is for forward repetitions and another rule is for backward repetitions. This type of cellular automaton is used for encryption.

If all the cells follow different rules, then the cellular automaton is called hybrid. If all the cells follow one rule, then the cellular automaton is called uniform (Krishna, 2012).

Cellular automata have the capability of performing complex calculations with a high degree of efficiency and power, as well as formatting the behavior of complex systems in nature. For this reason, the cellular automata and the related architectures are widely studied in natural sciences, mathematics, and computer sciences. They are used for formatting the physical and biological phenomena, such as floods, earthquakes, and the formation of galaxies. They can be considered as mathematical tools. They are used as parallel computing devices for high-speed simulation and computing tasks, such as image processing (Farmer, 1984) some of which are discussed in the following.

Cellular automaton for parallel computing

In the design of the automaton, certain applications are suggested for the cellular automata, such as the making of the parallel multipliers, prime numbers sieving, and parallel processing of computers, as well as for sorting machines. Cellular automaton can be used for fault-tolerant computing automata (Toffoli, 1984). Two-dimensional cellular automaton is widely used in image processing and map recognition. Researchers have evaluated the cellular automata as a computing device, which has been introduced as a common computer in nanometer scale.

Cellular automaton for formatting games

Cellular automaton is used for formatting various games, the most popular of which was suggested by Berlekamp et al. They have shown how the very simple rules of cellular automata can describe the behavior of the very complex systems, such as the Game of Life (Gardner, 1970). The game originally was proposed by Conway and was popularized by Martin Gardner. In addition to the Game of Life, there are certain other games that have been formatted through cellular automata.

Conclusion

Cellular automata are decentralized spatially extended systems consisting of large numbers of simple identical components with local connectivity. in the cellular automata, computations are done in a parallel manner. In each step, the cell considers the adjacent neighbors and using transmission state rules takes its next discrete state. Besides every cell can process independently from the other cells. The cellular automaton is used for computational algorithm design, image processing, graphics production, simulation of physical and biological phenomena, etc. in second section different kinds of Cellular automata has been mentioned. By examining the different formatting of cellular automaton, it seems that the fuzzy cellular automaton, in terms of accuracy, is better than others. Because considering the uncertainty and ambiguity in digital images, fuzzy logic can have a suitable behavior in such situations. Also, cellular automata fuzzy regard to the remainder to diagnose edge detection, image enhancement and noise removal in digital images has better operation. The reversible cellular automata better operation has in cryptography.

Future research:

1) Combination of learning cellular automaton and fuzzy logic for learning in artificial intelligence.

2) Increase the rules of reversible cellular automata and combine it with additive cellular automata for eliminating specific problems.

References

Anvarinejhad T. Meybodi M. R.(2004). Fuzzy cellular automata. Proceedings of the 5th Iranian Conference on Fuzzy Systems. P 57–65.

Beigy H. Meybodi M. (2004). A mathematical framework for cellular learning automata, Advances in Complex Systems , Vol 7, Issue 03n04. 295–320.

Enayatifar R. Meybodi M. R. (2009).Adaptive Edge Detection via Image Statistic Features and Hybrid Model of Fuzzy Cellular Automata and Cellular Learning Automata, Information and Multimedia Technology International Conference on , 273 – 278.

Farmer D. Toffoli T. Wolfram S. (1984). Cellular Automata, Proceedings of an Interdisciplinary Workshop, VOL 10, pages 247.

Ganguly N, et. al (2001).Theory of Additive Cellular Automata, Fundamenta Informaticae XXI , IOS Press, 1001–1021.

Gardner M . (1970). The Fantastic Combinations of John Conways New Solitaire Game of Life. Scientific American,VOL 223, 120–123

Krishna P.V, et. al. (2012).A Survey on Cellular Automata and Its Applications, Communications in Computer and Information Science, Vol 269, 753-762.

Maji P, et. al. (2003). Theory and Application of Cellular Automata For Pattern Classification, Fundamenta Informaticae, VOL 58, 321–354.

Mitchell M, et. al. (1998). Computation in Cellular Automata:A Selected Review, Nonstandard Computation, pp. 95–140.

Narendra K. S. Thathachar M. (1974). Learning automata a survey. IEEE Trans Syst ManCybern, VOL 4, 323–34.

Seredynski. F. et. al. (2004). Cellular automata computations and secret key cryptography, parallel computing, VOL 30, Issues 5-6 ,753-766.

Toffoli T. (1984). CAM: A High-Performance Cellular Automata Machine Physica D: Nonlinear Phenomena ,vol 10, Issues 1-2, 195–204.

Uguz s. et. al. (2015). Edge detection with fuzzy cellular automata transition function optimized by PSO, Computers & Electrical Engineering, Vol 43, 180–192.

Enayatifar R. Meybodi M. R. (2009).Adaptive Edge Detection via Image Statistic Features and Hybrid Model of Fuzzy Cellular Automata and Cellular Learning Automata, Information and Multimedia Technology International Conference on , 273 – 278.

Farmer D. Toffoli T. Wolfram S. (1984). Cellular Automata, Proceedings of an Interdisciplinary Workshop, VOL 10, pages 247.

Ganguly N, et. al (2001).Theory of Additive Cellular Automata, Fundamenta Informaticae XXI , IOS Press, 1001–1021.

Gardner M . (1970). The Fantastic Combinations of John Conways New Solitaire Game of Life. Scientific American,VOL 223, 120–123

Krishna P.V, et. al. (2012).A Survey on Cellular Automata and Its Applications, Communications in Computer and Information Science, Vol 269, 753-762.

Maji P, et. al. (2003). Theory and Application of Cellular Automata For Pattern Classification, Fundamenta Informaticae, VOL 58, 321–354.

Mitchell M, et. al. (1998). Computation in Cellular Automata:A Selected Review, Nonstandard Computation, pp. 95–140.

Narendra K. S. Thathachar M. (1974). Learning automata a survey. IEEE Trans Syst ManCybern, VOL 4, 323–34.

Seredynski. F. et. al. (2004). Cellular automata computations and secret key cryptography, parallel computing, VOL 30, Issues 5-6 ,753-766.

Toffoli T. (1984). CAM: A High-Performance Cellular Automata Machine Physica D: Nonlinear Phenomena ,vol 10, Issues 1-2, 195–204.

Uguz s. et. al. (2015). Edge detection with fuzzy cellular automata transition function optimized by PSO, Computers & Electrical Engineering, Vol 43, 180–192.