Survey on Cellular Automata (1-Dimension and 2-Dimension CA)

Zeinab E. E. Mahmoud [1], Samiah J. Alzahrani, Lujain Natto[3], Wafa S. Alshehri[4], Mariam O. Alrashidy[5]

[1] Assistant professor, Faculty of Computing and Information Technology /King Abdulaziz University /Jeddah/ Saudi Arabia
Zmahmoud@kau.edu.sa

[2] Faculty of Computing and Information Technology /King Abdulaziz University /Jeddah/ Saudi Arabia
samiahalzahrani@hotmail.com

[3] Faculty of Computing and Information Technology /King Abdulaziz University /Jeddah/ Saudi Arabia
wa.sa.sh.2015@hotmail.com

[4] Faculty of Computing and Information Technology /King Abdulaziz University /Jeddah/ Saudi Arabia
nattoLd@live.com

[5] Faculty of Computing and Information Technology /King Abdulaziz University /Jeddah/ Saudi Arabia
mariamuoh23@yahoo.com

Abstract

Cellular automata are discrete organizations of cells in an n-dimensional grid growing in a synchronized way and reliant. The cells cooperate with each other over the use of a rule liable only on local features, which lead to some universal behavior where these rules are uncomplicated and can result in a very complex and complicated global behavior.

This paper introduces historical background about cellular automaton, deals with theory and applications of 1-dimensional cellular automata and 2-dimensional cellular automata,

Also, in this paper we will introduce a study of CA in a 1-D CA and 2-D CA, show some cellular automata phenomena in the natural life, provide 2-D CA application in image processing and finally, we will show issues and discussion that related to CA such as classification issue and majority problem in 1-D and 2-D also, it discuss a property of CA which a self-Reproducing with von-Neumann CA all these topics will give a general view about CA and its fields.

Keywords: *Cellular Automata, Survey on Cellular Automata, 1-D CA, 2-D CA.*

1. Introduction

A cellular automaton can be defined as a discrete dynamical system that contains a regular grid of finite state automata or cells. These cells can modify their states liable on their neighbor's states and according to rules that update locally. All these cells transform their state instantaneously, by the same update rule. This process is repetitive at distinct time steps. It seemed that simply updated rules astonishingly might produce very complex dynamics when applied in this manner. The most famous example of the cellular automaton is the Game-of-life by John Conway.

Cellular automata are discrete and homogeneous in both space and time; their interactions are locally. There are numerous processes in nature are ruled by local and homogeneous fundamental rules, which makes them agreeable to modeling and simulation by cellular automata, For example, in physics, they can model the fluid dynamics by moving point particles in a regular lattice. In this case, the local update rule is designed to mimic particle collisions. Some of the most expansively investigated concepts in cellular automata theory for instance reversibility and conservation laws driven by physics.

Cellular automata are also mathematical models for enormously parallel computation. The simple update rules can make the cellular automaton computationally universal, thus, qualified of performing random computation tasks.

A cellular automaton (CA) is a set of cells sorted in an N-Dimensional (N-D) lattice, each cell’s state changes as a function of time by set of rules that involves the states of the neighboring cells. The updating rule for the cells state is the same for each cell, and it is applied to the whole grid. The new state of each cell, at the next time, depends on the current state of the cell and the states of the neighboring cells. All cells on the lattice are updated synchronously. Thus, the state of the whole lattice advances in separated time steps.

Two most popular types of CA are: one dimensional CA (1D CA) and two dimensional CA (2D CA). If the grid is a linear array of cells, it is called 1D CA and if it is a rectangular or hexagonal grid of cells then it is called 2D CA. A CA with one central cell and four near neighborhood cells is called a von Neumann/Five neighborhood CA while a CA having one central cell and eight near neighborhood cells is called Moore/Nine neighborhood CA.

In this paper, we will show two kinds of Cellular automata. The rest of the paper is structured as follows: next section show historical background about Cellular automata; Section 3 gives overview of one dimensional cellular automata and describe some of its rules; Section 4 illustrates two dimensional cellular automata and its rules with details; In the last section of the paper we will present some of the important issues and we will present some of discussion concerning about the cellular automata, including classification of cellular automate as Wolfram classification which includes four styles that mainly based on a graphic analysis of the development of one-dimensional CA and Li and Packard have established a classification system depending on Wolfram's system. Then, we will discuss classification of cellular Automata rules based on their Properties in 1-Dim or 2-dim cellular automata. After that, we will describe some issues in CA the density classification problem in both 1-Dim and 2-Dim CA and cellular automata rules in 2-Dim include Toom’s Rule and Reynaga’s rule and discuss the properties of them and determine the most effective one. At the last parts of this paper , we will talk about the assumption that say “ the physical universe is, fundamentally, a discrete computational structure and Self-Reproducing property with von-Neunmann cellular automata. and finally we conclude the paper.

2. Historical background about cellular automaton

Cellular automata can be categorized as the simplest models of distributed processes. They contain an array of cells; each cell is allowed to be in a particular state from few states. All at once, each cell checks its neighbor's states. Then, each cell uses previous information and applies a simple rule to decide the state it must change to it. This elementary step is repetitive over the entire array, many times.

Cellular automata was invented in 1940 by two scientist of mathematicians John von Neumann and Stanislaw Ulam, through their work at Los Alamos National Laboratory in north-central New Mexico. Regardless of the easiness of the rules leading the changes of state as the automaton transfers from one generation to the next, the growth of such a system is complex really.

In the 1940s, Stanislaw Ulam worked at the Los Alamos National Laboratory to study the growth of crystals, by using a simple lattice network as his model. Simultaneously, John von Neumann worked on the problem of self-replicating systems. He found from the initial prototype that notion of one robot building another robot. This scheme is known as the kinematic model.

While he developed this design, von Neumann recognizes the excessive difficulty and cost of building a self-replicating robot. Ulam was the first one who recommended using a discrete system for producing a reductionist model of self-replication.

In the late 1950s, Ulam and von Neumann formed a method for computing liquid motion. They reach to the concept of the method by considering a liquid as a group of discrete units and compute the motion of each one depending on its neighbors' behaviors. Consequently was born the first structure of cellular automata.

In the 1960s, cellular automata were considered as a specific type of dynamical system and the linking with the mathematical field of symbolic dynamics was recognized for the first time. In 1969, Gustav A. Hedlund collected many results subsequent this opinion [2]. The most essential result is the classification in the Curtis–Hedlund–Lyndon theorem of the set of universal rules of cellular automata as the set of continuous endomorphisms of shift spaces.

In 1969, German computer pioneer Konrad Zuse infer "Zuse's Theory" and published his book Calculating Space, suggesting that the physical laws of the universe are discrete by nature, and that the whole universe is the production of a deterministic calculation on a single cellular automaton. " Zuse's Theory" became the basis of the area of study named digital physics.

In the 1970s, a two-dimensional cellular automaton called Game of Life became broadly well-known, mostly amongst the primary computing community.[4] Game of Life was invented by John Conway and propagated by Martin Gardner in a Scientific American article, [5] its rules are as follows: If there is a cell has two black neighbors, it stays in its current state. If it has three black neighbors, it converts black. In entirely other states it converts white. Even with its simplicity, the system accomplishes an impressive variety of behavior, changing between seeming randomness and order. One of the greatest features of the Game of Life is the recurrent occurrence of gliders, preparations of cells that basically move themselves across the grid.

In mid of 1981, Stephen Wolfram independently started working on cellular automata after considering how complex patterns seemed molded in nature in invasion the Second Law of Thermodynamics. His first paper in Reviews of Modern Physics studying fundamental cellular automata (Rule 30 in particular) published in June 1983. [6]

In 1990s, Wolfram's research assistant Matthew Cook proved that rule 110 universal.

In 2002, Wolfram published his book A New Kind of Science, which expansively argues that the findings about cellular automata are strong and have importance for all disciplines of science. [7]

3. One- Dimensional Cellular Automata

3.1 Definition

The cellular automata can be one, two or more dimensions; our concern in this paper is one and two dimensions cellular automata. First of all one dimension cellular automata has several definitions the most suitable one is " a uniform linear array of identical cells of infinite or finite extent with discrete variable at each cell"[7]. The general five-parameter that define any cellular automata is:

1. Number of states.
2. Neighborhood size.
3. Length of the cellular automata.
4. Guiding Rules.
5. Number of times the generation takes.

3.2 Neighborhood Structure

We can imagine the one dimension cellular as a row of ordered, infinite any shape of cells usually in squares. Every cell in the row can contain different finite number of values called states equal or greater than two. Two-state automata is the simplest case, which every cell can be ether zero or one, on or off. Graphically cells can be black or any other color and in the greater than two-state automata more colors are used. As seen in Fig. 1 a two-state automata is shown which cells either zero/off/ dead or one/on/alive.

Fig. 1 Two-state automata

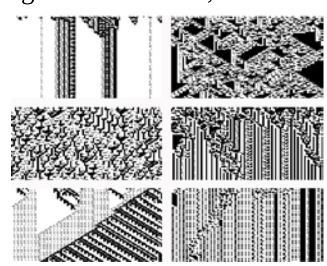
1	0	1	1	0	1	0	0	1	1	1	0

3.3 Rules Description Of 1-D Ca

A cellular automata doesn't update cells state over and over never stops. Starting with the updating of the cells of the cellular automata we must specify the initial state of the cells, the initial conditions or seed. Then to produce a new generations of cell and update the cells state we have to specify a rule that control this updating, which include the current cell state and the state of the adjacent cells called neighborhood. The rule is being applied to all cells at same time step. When time come to cells to change their states each cell look around and get their neighborhoods states. Depending on the cell current state and the neighborhood state and the rule being applied, the cell state updated. There are 256 different rules for one dimension, two-state, three-neighborhood classified based on its complexity into four classes by Wolfram[9]:

- Class I
  Cells generation leads to states that are homogeneous in which all cells have the similar states.
- Class II
  Cells generation leads to a set of periodical, settled structures that are separated and simple.
- Class III
  Cells generation leads to disorganized patterns.
- Class IV
  Cells generation leads to complex patterns, sometimes long-lived.

In cellular automata the next generation is drawn under the past generation and so on producing a possible lovely two-dimensional patterns It is impressive that simple rules can lead to such complex and interesting pictures such as Fig. 3. And as seen in Fig. 2 one row is the current generation filled in with live/dead cells; the other row is the next generation used as a suitable place for computing the updated states. The gray cell the left end of the first row lake of neighbor on its right and the grey cell on the right end lake of neighbors on its left, so both are wrap-around from the other side of the array. The left end cell is considered as a neighbor to the cell on the right end.



1	0	1	1	0	1	0	0	1	1	1	0

Fig. 2 sample set of generations

Fig. 3 One dimensional cellular automata evolution

3.4 Special Rules

ISSN 2348 – 7968

every rule has a number range from its equivalent 00000000b to 1111111b and its decimal equivalent from 0 to 255. There are some famous special rules we will listed below:

- Rule 30 (00011110b)
  This rule is used in mathematics to generates random integers.
- Rule 184 (10111000b)
  This rule used in traffic flow model in one corridor of a high way described in Fig. 4.

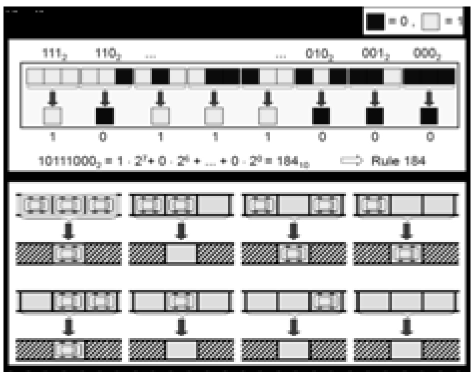


Fig. 4 The traffic rule

- Rule 90 (01011010b)
  This rule called Sierpiński automaton. The updated state of the cell is the XOR function of its neighborhood.
- Rule 110 (01101110b)
  The updated state of the cell is the OR function of its neighborhood and the cell current state.
- Rule 51 (00110011b)
  This rule gives the complementary of the cell state.
- Rule 170 (10101010b)
  This rule perform a left shift to cellular automata row.
- Rule 240 (11110000b)
  This rule perform a right shift to cellular automata row.
- Rule 204 (11001100b)
  This rule called the identity rule.

3.5 In Natural Life

A lots of cellular automata phenomena in the natural life exists as shown if Fig. 5 such as gene regulation when one gene prevent or activate other gene, and that one also prevent or activate a third one and so on. Another phenomena is multi-cellular organisms at the heart of the growth of a multi-cellular organism is the process of cellular division. Also the chess-board is a clear example, image processing, parallel computation and many phenomena in the real world. Shell patterns, super-organisms, image processing, parallel computation etc all are examples of cellular automata phenomena in the natural life.

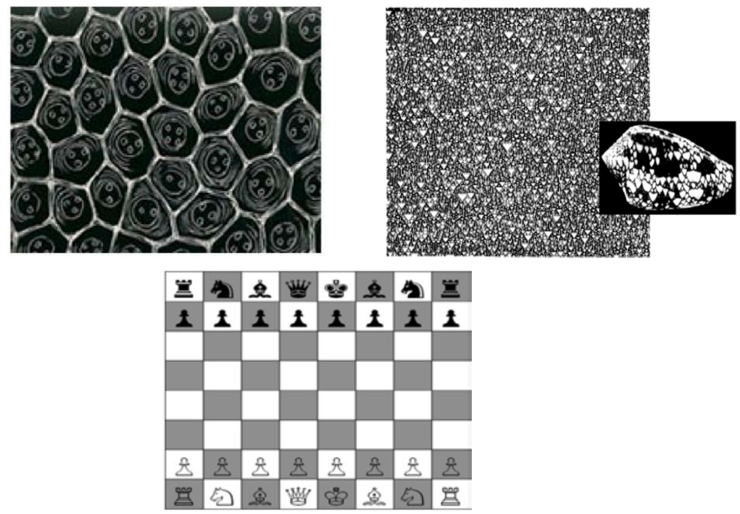


Fig.5 Some examples of one dimensional cellular automata phenomena.

3.6 Applications

One dimensional Cellular automata is active and valuable computation machine for applications like cryptography, algorithms, modeling physical systems and Designing fractal pattern. The most recent published research about One dimensional Cellular automata application is done by K. J. Jegadish Kumar and V. Karthick about “AES S-Box Construction using One Dimensional Cellular Automata Rules” [8]. In this paper the authors has proposed an Reversible Cellular Automata (RCA) rule-based function used for Advanced Encryption Standard(AES) S-Box shown in Fig.6. The Substitution Box (S-Box) is the main core aspect of AES algorithm that specify the non-linearity of the hole algorithm. after the experimental analysis has been done, result shows that the S-Box implemented by using RCA rule-30 have a better performance than using the other rules and traditional AES S-Box as shown in Fig.7.

$$\begin{pmatrix} n_{0,0} & n_{0,1} & n_{0,2} & n_{0,3} \\ n_{1,0} & n_{1,1} & n_{1,2} & n_{1,3} \\ n_{2,0} & n_{2,1} & n_{2,2} & n_{2,3} \\ n_{3,0} & n_{3,1} & n_{3,2} & n_{3,3} \end{pmatrix} = f\left(\begin{bmatrix} m_{0,0} & m_{0,1} & m_{0,2} & m_{0,3} \\ m_{1,0} & m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,0} & m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,0} & m_{3,1} & m_{3,2} & m_{3,3} \end{bmatrix}, \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix}\right)$$

Fig.6 The proposed approach is depicted in matrix form

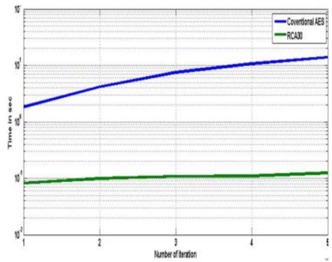


Fig.7 Computational Time analysis of S-Box

4. Two Dimensional Cellular Automata

4.1 Definition

A two dimensional cellular automaton (2D-CA) is a collection of cells arranged in an 2-Dimensional (2-D) lattice, each cell's state changes according to set of rules that includes the states of the neighboring cells.

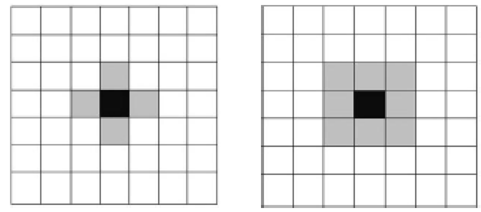
2D CA is a rectangular or hexagonal grid of cells. A CA with one central cell and four near neighborhood cells is called a von Neumann/Five neighborhood CA while a CA having one central cell and eight near neighborhood cells is called Moore/Nine neighborhood CA.

4.2 Ca Representation

A CA can be represented with six- tuple, C= {L,D,N,Q,δ,$q0$}; where L is the regular lattice of cells, the number of cells is finite. D is dimensional organization of the lattice 1-D, 2-D, 3-D, etc… ,Q is the finite set of states, $q0$ is called the initial state such that $q0 \in Q$,N is a finite set neighbor cells and δ: $Q^n \rightarrow Q$ is the transition function, the state transitions starts from an initial state qi, cells change their state based on their current state and the states of neighbor cells.

4.3 Neighborhood Structure

The neighborhood of a cell, called the central cell, made up of the central cell and neighbor cells. The states of the central cell and neighbor cells determine the next state of the central cell. There are several neighborhood structures for cellular automata. The two most common structures are Von Neumann and Moore neighborhood, as shown in Fig. 8.



(a) (b)

Fig. 8 Neighborhood structure (a) Von Neumann, (b) Moore

Von Neumann neighborhood has five cells, it consists of the central cell and its four immediate non-diagonal neighbors and has a radius of 1. The radius of a neighborhood is the maximum distance from the central cell to cells in the neighborhood, horizontally or vertically. Moore neighborhood has nine cells, it consists of the central cell and its eight neighbors and has a radius of 1. Extended Moore neighborhood consisting of the same cells as in the Moore neighborhood structure, but the radius of neighborhood is increased to 2.

4.4 Rules Description Of 2-D Ca

In 2D Nine Neighborhood CA the next state of a specified cell is affected by the current state and eight cells in its nearest neighborhood. Such dependencies are computed by various rules. A particular rule convention which is mentioned in [10] is shown in Table 2.

Table 1: Two dimensional CA rule convention

64	128	256
32	1	2
16	8	4

The central box represents the current cell and the other boxes represent the eight nearest neighbors of the current cell. The number in each box shows the rule number describing the dependency of the current cell on that certain neighbor only. Rule 1 describes dependency of the central cell on itself alone while rule 128 describes dependency of its top neighbor, and so on.

These nine rules are the main rules. When the cell has dependency on two or more neighboring cells, the rule number is the sum of the numbers of the related cells. For example, the 2D CA rule 171 (128+32+8+2+1) indicates

to the five-neighborhood dependency of the central cell on (itself, top, left, bottom and right). The number of such rules is ${}^9C_0 + {}^9C_1 + \ldots + {}^9C_9 = 512$ which involves rule describing no dependency.
These rules are listed into nine lists in the following way. List-N for N=1, 2…. 9, involves the rules that indicate to the dependency of current cell on the N neighboring cells among itself, top, left, bottom, right, top-left, top-right, bottom-left and bottom-right. The 2D CA rule 171 which is above-mentioned belongs to list-5. So list-1 includes 1, 2, 4, 8, 16, 32, 64, 128, and 256. List-2 includes 3, 5, 6, 9, 10, 12, 17, 18, 20, 24, 33, 34, 36, 40, 48, 65, 66, 68, 72, 80, 96, 129, 130, 132, 136, 144, 160, 192, 257, 258, 260, 264, 272, 288, 320 and 384. Similarly rules belonging to other lists can be produced. It is noted that the number of 1's in the binary sequence of a rule is same as its list number.
For two state nine neighborhood CA, there are 2^{2^9} possible rules exist. Out of them only $2^9 = 512$ are linear rules that is, the rules which can be realized by EX-OR operations only and the rest of the $2^{2^9} - 2^9$ rules are non-linear which can be realized by all possible operations of CA. Taking XOR operation (/s) among nine basic rules, we get other 502 linear rules (excluding Rule0).
The example shown in the following explains how a composite Rule449 is calculated using basic rules and XOR operations.
Example: Rule449 is expressed in terms of basic rule matrices as follows: $Rule449 = Rule256 \oplus Rule128 \oplus Rule64 \oplus Rule1$.

4.5 Application In Image Processing

4.5.1 Relationship of 2D CA with image

The image can be described as a 2-dimensional function I.
$I = f(x,y)$
Where x and y are spatial coordinates. Amplitude of f at any pair of coordinates (x,y) is called intensity I or gray value of the image. When spatial coordinates and amplitude values are all finite, discrete quantities, the image is called digital image. The digital image I is represented by a single 2-dimensional integer array for a gray scale image and a chain of three 2-dimensional arrays for each color (Red, Green and Blue). As the digital image is a 2-dimensional array of m×n pixels, so we are interested in two- dimensional CA model. An image is viewed as a two dimensional CA where each cell represents a pixel in the image and the intensity of the pixel is represented by the state of that cell. The color values of the pixels are updated synchronously at a discrete time step. So the time to make any image processing task is very less. Due to this type of manner of CA model effects a large application in image processing such as image restoration, enhancement, segmentation, compression, feature extraction and pattern recognition.
Two dimensional CA algorithms are mostly used in image processing because its structure is very similar to an image. The following section shows a review of some published works by different researchers in the field of cellular automata with application in image processing. The review is labeled into the following sub-headings.

4.5.2 Translation of images

Translation of image means moving the image in all directions. Generally, the image can be translated in x (left, right) and y (up, down) directions. Translation of images using CA includes diagonal movement of images. Choudhury et al. have applied eight fundamental 2-dimensional CA rules to any images for some iterations and found that these rules will translate the images in all directions [10], these rules which belong to list-1 are Rule2, Rule4, Rule8, Rule16, Rule32, Rule64, Rule128, Rule256. Table 3 summarize their works.
The range of shift is based on the number of repetitions of the application of such rules as explained in the following figures (Fig. 9 and Fig. 10).

Table 2: Translation of images using basic 2D CA rules

Rule of List -1 to be applied	**Direction of Translation of Images**
Rule2	Left
Rule32	Right
Rule8	Top
Rule128	Bottom
Rule4	Top- Left (Diagonal)
Rule16	Top- Right (Diagonal)
Rule64	Bottom- Right (Diagonal)
Rule256	Bottom- Left (Diagonal)

ISSN 2348 – 7968

Qadir et al have done an extension of the previous work[11]. They used twenty five neighborhood instead of nine neighborhood for images translation. They have applied twenty four fundamental rules to any images and obtained satisfying results. The idea of their work is to move the images in more directions in addition to the previous work so it can be used in games applications. The review on images translation using CA infers that the number of iteration required for nine neighborhood structure are more than twenty five neighborhood structure.

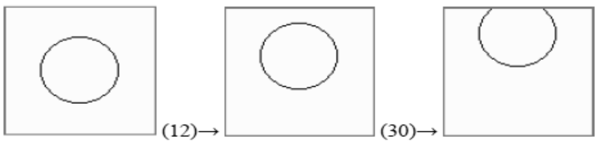


Fig. 9 Application Rule-8 (To a circle of radius – 26 pixels)

Fig. 10 Application Rule-256

4.5.3 Multiple copies of any arbitrary image

Choudhury et al. have used the rules other than the fundamental rules to generate multiple copies of the particular image, the number of copies is the list number to which the applied rule belongs. So the maximum number of copies an image can have on application of such rules is 9, because the maximum list number is 9. It is noted that the multiple copies occur only when number of repetition is 2^n (n=1, 2, 3,…). The following figures clarify the above assertion for n=5.

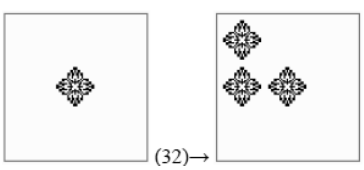


Fig. 11 Application Rule-7, List -3

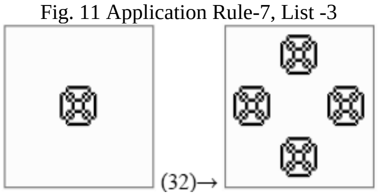


Fig. 12 Application Rule-170, List -4

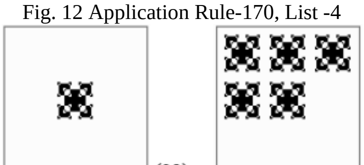


Fig. 13 Application Rule-31, List – 5

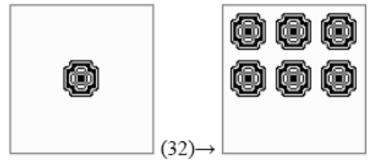


Fig. 14 Application Rule-63, List – 6

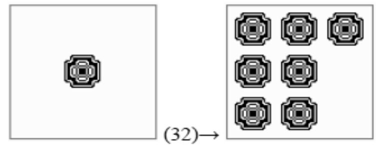


Fig. 15 Application Rule - 415, List – 7

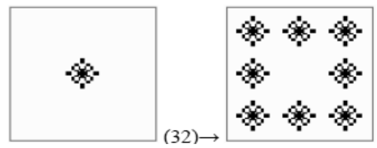


Fig. 16 Application Rule - 510, List – 8

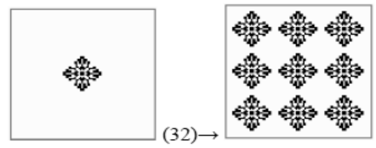


Fig. 17 Application Rule - 511, List – 9

It is also noted that the rules belonging to same group create equal number of copies, but the distributions of these copies are different. For example after applying rule-5 of list-2 on an image (centered), the two copies of that image are located in the center and upper-left corner of the matrix (matrix indicates to the bounded area, where image is shown on the screen). But when rule- 68 of the same list is applied on the same image (centered), the two copies of original image are found in upper-left and lower-right corners of the matrix.

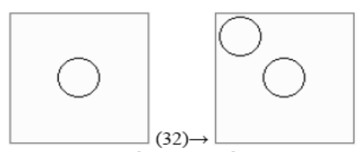


Fig. 18 Application Rule – 5, List-2

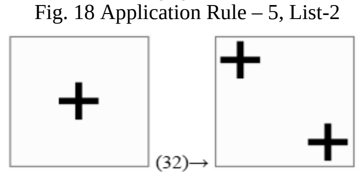


Fig. 19 Application Rule-68, List-2

These copies of the image are created within the display matrix shown that maximum length of the image in all directions, it is not more than 31% of the length of row or column of the display square matrix (Here it is 100).

4.5.4 Zooming in, zooming out and thickening, thinning of symmetric images

In zooming there are two operations, zooming in and zooming out. Chaudhry et al. used hybrid CA concept for zooming of symmetric images [10]. For zooming in they

ISSN 2348 – 7968

have applied Rule2, Rule32, Rule8 and Rule128 in four different regions of images respectively and for zooming out Rule32, Rule2, Rule128, Rule8 are applied respectively.

Thinning is a special state of scaling and it is a significant step in image analysis. Thinning horizontal and vertical blocks using 2D hybrid CA has been explained by Chaudhry et al. [10]. They applied Rule32, Rule2, Rule1 and Rule1 in four different regions of images respectively for horizontal thinning and, Rule1, Rule1, Rule128, Rule8 for vertical thinning.

The idea of their work is to apply different rules to different pixels of a symmetric image, the shape of the image changes; zooming in, zooming out, thickening and thinning of the image can be shown. The following figures explain these concepts.

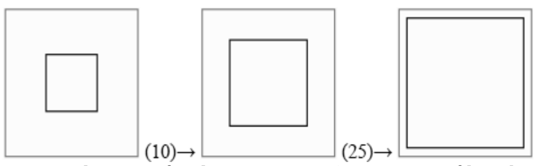


Fig. 20 Application of Rules–2,32,8,128 (To a square of length - 40 pixels, transformation is zooming in)

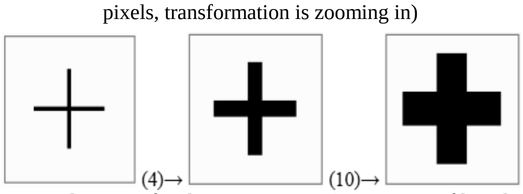


Fig. 21 Application of Rules – 2,32,8,128 (To a + sign of length - 55 pixels and breadth–3 pixels, transformation is zooming in)

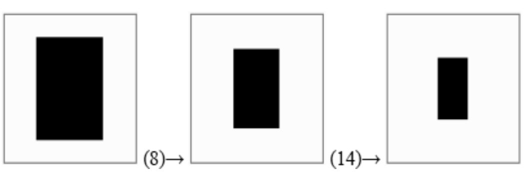


Fig. 22 Application of Rules-32, 2, 128,8 (To a rectangle of length - 50 & breadth – 70 pixels, transformation is zooming out)

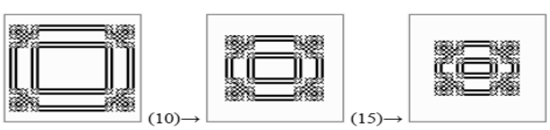


Fig. 23 Application of Rules-32, 2, 128,8 (To a complex pattern, transformation is zooming out)

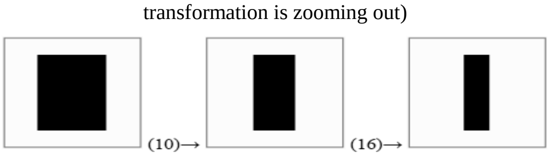


Fig. 24 Application of Rules-32, 2,1,1 (To a rectangle of length-50 & breadth-70 pixels, transformation is horizontal thinning)

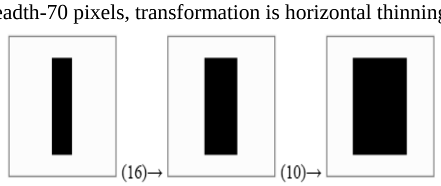


Fig. 25 Application of Rules – 2,32,1,1 (To a rectangle of length - 50 & breadth – 70 pixels, transformation is horizontally thickening, reverse of the transformations in Fig. 25)

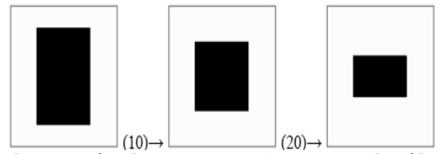


Fig. 26 Application of Rules – 1,1,128,8 (To a rectangle of length - 50 & breadth – 70 pixels, transformation is vertically thinning)

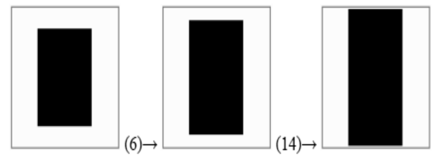


Fig. 27 Application of Rules–1, 1, 8, 128 (To a rectangle of length - 50 & breadth – 70 pixels, transformation is vertically thickening)

In this application, the display matrix is divided into four regions as shown in Fig. 28.

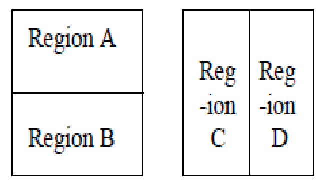


Fig. 28 Division of (100x100) 2 D CA region for hybrid transform

Two rules (may be different) are applied to two regions Region A and Region B followed by application of another two rules (may be different) to the regions Region C and Region D. These two applications called a single hybrid step, indicated as Hybrid (a, b, c, d), where a, b, c, d are four rules applied respectively to those four regions Region A, Region B, Region C and Region D. It is observed that Hybrid (2,32,8,128) makes Zooming in (Fig. 20 and Fig. 21). Hybrid (32,2,128,8) makes Zooming out (Fig. 22 and Fig. 23). Hybrid (32,2,1,1) makes horizontal thinning. Hybrid (2,32,1,1) makes horizontal thickening (Fig. 24 and Fig. 25). Hybrid (1,1,128,8) makes vertical thinning. Hybrid (1,1,8,128) makes vertical thickening (Fig. 26 and Fig. 27).

5. Issues and Discussion

A CA has three major characteristics, dimension represented by d, states for each cell k, and radius r. The dimension identifies the composition of cells, a one dimensional line, two dimensional etc. The radius expresses the number of cells in each direction that will have an influence the updating of a cell.

In one-dimensional cellular automata, a radius of r consequences in a neighborhood of size referred by m where, m = 2r + 1. In case of cellular automata with higher dimension it must be indicated whether the radius refers only to directly

IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 7, July 2015.
www.ijiset.com
ISSN 2348 – 7968

adjacent cells only or involves diagonally adjacent cells too.

The local rule which used to update each cell in cellular automata space is often represented as a rule table. This table indicates what value the cell should take on based on the possible set of states the neighborhood can have. The number of potential groups of states the neighborhood can have is k^m which give k^{k^m} rule tables.

Cellular automata had appeared in a huge number of distinct scientific areas since their introduction by John von Neumann in the 1950's. A modern history of this society of work is offered by Sarkar which breakup the field into three major types: classical, games, and modern. These same three categories also shows by McIntosh describes these three classes by their significant works, i.e. von Neumann's self-reproducing machines for classical which discussed in later sections where standard research is constructed on von Neumann's cellular automata as a tool for developing genetic self-duplicate, Conway's Game of Life for games, and Wolfram's classification scheme for modern type. From other side Wolfram in the 1980's observed very different dynamical performances in simple CA and classed them into four types, showing a range of simple, complex, and chaotic behavior. There is some techniques and methodology that aims to classify cellular automata (CA) depending on their behavior and its dynamics performance of the automata over all possible starting states. A lot of CA will look to be in a number of different classes for certain special starting states, but for most usual primary conditions will be steady.

5.1 Wolfram classification

One of classification techniques was offered by Wolfram. The Wolfram classification system includes four styles which are mainly based on a graphic analysis of the development of one-dimensional CA.

- Class I: development result in a homogeneous situation in which all cells have the same value.
- Class II: advancement prompts an arrangement of steady or intermittent structures that are isolated and simple.
- Class III: development result in disordered patterns.
- Class IV: development result in complex forms.

Where, classes III and IV are especially hard to distinguish between.

After that, Li and Packard have established a classification system depending on Wolfram's system; the modern version has six styles and there have been several tries to classify cellular automata in correctly and precise classes, from the original classification by the Wolfram's. For instance, Li-Packard system which basically break down Wolfram's Class II into three extra classes: fixed point, two-cycle, and periodic. [15]

By Indian Statistical Institute help and authors of Classification of Cellular Automata Rules Based on Their Properties at the nth iteration they generate a code to get the nth iteration for arbitrary 1-D or 2-D CA rules. Where cell value is updated based on the updating rule, which includes the cell value as well as other cell values in a specific neighborhood nearest to the cell itself. The cellular automata is totally well-defined with the support of five factors number of States, amount of Neighborhood, Size of the cellular automata, Rules either its Uniform or hybrid and Number of times the development requires. These five factors affect number of rules in CA.

Where potential number of Uniform CA rules defines via equation 1:

$$State^{(state)\ \wedge\ Nighborhood} \qquad (1)$$

And the potential number of hybrid CA rule via equation 2:

$$\left(State^{state\ power\ Nighborhood}\right)^{length}\ of\ the\ CA \qquad (2)$$

Potential number of Hybrid CA rules modifying over diverse time throughout the development via equation 3:

$$\left(\left(State^{State^{Neighborhood}}\right)^{length\ of\ the\ CA}\right)^{time} \qquad (3)$$

The number of uniform cellular automata rules increases hugely with the growth in neighborhoods such as if neighborhoods was 4 consequences to that the uniform CA rules will be 2^16 in the same way if neighborhood was 5 uniform CA rules will be 2^32. Produce a significant question as: How to achieve the exact CA rule(s), which will demonstrate a specific application? The solution of this question is classification of automata. In Wolfram’s methodology the space-time schema of 1-D CA after determinate number of repetitions, the patterns produced for each repetition and think about it in a 2-D plane extended it to the 2-D area results in the assembling of each of the plane forms produced in continuous

IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 7, July 2015.
www.ijiset.com
ISSN 2348 – 7968

iterations one on upper of the other growth to a three-dimensional form, which demonstrations of nested construction. At this point, Wolfram classed one-dim CA rules depending on their complexity. While, classification of cellular automata rules based on their properties theory based on the behavior of the rule in the n number of iteration. [16]

5.2 Application of 2-dim CA

We will discuss in this section one of the most famous example in 2 dimension cellular automata in games field is Conway's Game of Life. The Game of Life was first spread in year 1970 by Gardner where this game belongs under the Mathematical Games category which two-dimension cellular automata. It work with two states only that define via two colors white and black and its update rule to its cells depend on neighbors which eight in this case an update rule as follows:

If two neighbors are black, then the cell stays the same color;
If three neighbors are black the cell becomes black;
If any other number of neighbors is black the cell becomes white.

Cells are told to be alive or dead if they are in black or white colors, respectively. [17]

One of the problems in cellular automata (CAs) is the density classification problem. It depend on finding a cellular automata such that, given any initial configuration of 0's and 1's and if the fraction of 1's density is greater than 1/2 and it converges to the all-0 stable point configuration otherwise. Solving the density classification task using CA 184 with memory propose was to solve this problem by designing a CA brilliant by two mechanisms universal in nature: diffusion and non-linear sigmoidal response. This CA, which is unlike the classical ones because it has many states, has a success ratio of 100% operates for any system amount, any dimension and any density. [18]

5.3 Majority problem

5.3.1 Majority problem in 1-Dimanstion:

In one-dimensional binary cellular automata is starting with a random initial configuration and repeated for a maximum number of steps *I* or until a fixed point is reached. If the initial configuration holds more ones than zeros, the CA is estimated to have solved the job if a fixed point of all ones is reached and vice versa. The fraction of ones in a CA configuration is denoted as P, so the problem solution P *f* for an initial configuration of P_0 as in equation 4:

$$P_f = \begin{cases} 0 : & P_0 < 0.5 \\ 1 : & \text{otherwise.} \end{cases} \quad (4)$$

The condition where the initial configuration has an equivalent amount of ones and zeros (p= 0.5) is usually prevented by using an odd number of cells which is difficult as it need organizing the global state of the system using local communication between space cells . [19]

5.3.2 Majority problem in 2-dim:

Majority problem is more complicated to solve it in case of 2-dim instead of 1-dim and how to achieve rule that solve the arbitrary order of particles or cells in automata.
As cellular automata consist of group of cells in space denoted by ℓ which subset of Z^d where d represent dimension of automata $\ell = (\frac{Z}{LZ})^2$ where L is automata size and every cells in automata will have one of these state represent by $E = \{0,1\}^{\ell}$.
To make it simple we will consider configuration as function from ℓ to {0, 1} follow:
$x^t(c)$ Where state $c \in \ell$ at time denoted by t.
Each cell has connection to set of cells called neighborhood k cells, define with local transitions obtained with k vectors that form a set:
$N = (n_{1,\dots,n_k})$ Those elements of set called neighbored.

Cells can update them state according a state of neighbored with local rule of CA with function f from set $\{0,1\}^k$ to {0, 1}
When cells update synchronously via this equation:
$x^{t+1}(c) = f(x^t(c+n_1),\dots,(x^t(c+n_k))$ As $c \in \ell$
The density classification problem involves in a CA that would analysis to constant configurations $0 = 0^{\ell}$ or $1 = 1^{\ell}$ depend on the density of states 0 and 1 in automata in case that one state in the start of configuration is it less than or greater than ½ of automata space where the initial configuration $x \in E$.
According to the previous description we able to find the quality of a good classification (Q) via equation 5:

$$Q = \sum_{x \in E} PGC\ (x)/2^n \quad (5)$$

5.4 Cellular automata rules

In 1978 , toom rule structured by Andrei Toom for 2-dimension CA which strong and simple than other rules

ISSN 2348 – 7968

such as Reynaga's rule that we will discuss in next part.[20]

(a) t =0 (b) t=5 (c) t=15 (d) t= 30

Fig.29 Evolution of Toom's rule with L = 21. Blue/dark and white cells represent cells in state 0 and 1, respectively. [20]

- Toom's Rule [21]

In 2-dimension CA with toom's rule on space of squares/cells in a space we able to place with +1 or -1 on the top of cell then, use update rule via majority vote of cell state and state of neighborhood at the north and east of the current state in figure 1 describe experiment achieved by author Nazim *Fat`es* which describe different generation according toom's rule where +1 in specify via certain color and -1 with another one.

We can summarize Toom's Rule a deterministic version by the following steps:

1) At time t=0 the cells initialize to some values from set {+1,-1}.
2) At time>0 we apply Toom's rule at the same time for each cell in automata.
   - If center site + neighbored spin in North + neighboring spin at East > 0 then, current spin will be +1.
   - If center site + neighbored spin in North + neighboring spin at East < 0 then, current spin will be -1.

In time where there are 3 cells the sum will never be a zero this rule called also NEC rule according it's depending on North, East and center cells to specify the value of next generation of automata.
Toom's probabilistic rule depending on probability and can done via:
Applying deterministic rule of Toom's rule in case that it's give +1 to a cell with probability q change it to -1 and vice versa with probability p. Toom's rule use majority rule on neighbored in north and east of cell and a cell itself we can formally describe the local rule of majority (maj) where, majority in $\{0,1\}^{2m+1}$ maj value of $(q_{1,\dots,}q_{2m+1}) = 0$ if $q_i < m$ and $(q_{1,\dots,}q_{2m+1}) = 1$ otherwise on neighbored which of set $N_3 = \{(0,0),(0,1),(1,0)\}$. Marco et al. proof that a rule is suitable form infinite automata for classifying density while, Nazim ,Fat proof the inverse via study the property of rule which give author deprived result as classifier and the more worst the its fall even with small space automata as he/she work with L =9 and using 10^5 random models and get quality result Q=0.536 from that there is 0.7% are not classified well the bad quality here is result of bad classification while studied automata with L=21 where half of cells was misclassified.[15]

- Reynaga's rule

The rule is originally stated on von Neumann neighborhood $N = \{(0,0),(0,1),(1,0),(0,-1),(-1,0)\}$
Rey rule break up the neighborhood into 2 pieces and applying majority rule according to central cell state whether it in 0 or 1 state with local function as in equation 6:

$$Rey(C,N,E,S,W) = \begin{cases} \text{maj}(C,N,E) \text{ if } C = 1 \\ \text{maj}(C,S,W) \text{ otherwise.} \end{cases} \quad (6)$$

Where C, N, E, S, W represent central cell, of the North, East, South and West cells of the von Neumann neighborhood. This rule is better than toom's rule but it still weak if we compare it with other rules in 1-dim or 2-dim automata statistical study of density classification/ majority problem by Nazim with 10^5 model for L=21 as in Figure 30 which record quality to 0.61 which sure better than previous one . [20]

(a) t = 0 (b) t = 10 (c) t = 20 (d) t = 40

Fig.30 Evolution of Reynaga's rule with L = 21. [20]

5.5 CA and real world

In the start of 1982 until 2002 scientist such as Zuse, Fredkin and Wolfram give an assumption that say " the physical universe is, fundamentally, a discrete computational structure. Everything in our world—quarks, trees, human beings, remote galaxies—are just a pattern in a cellular automaton, much like a glider in *Life*" Betros study this assumption on them encyclopedia to proof the truth of this assumption in 2010, for two reasons:

- They study this assumption because of them interest in philosophy science.
- It may useful to study CA for world and study its computation property with the current one.

At the end of the study they find its unpredictable assumption because, computational complexity of the world. In my opinion it's may be correct assumption, but it will be very complex because of amount of cells from that how many rule will be there! ,if we talk about person as cells in cellular automata then, how many person in world space and how many possible rule in this case! [22]

5.6 Self-Reproducing with von-Neunmann cellular automata

General purpose digital computers offer an excellent chance for studies of this kind of science and von-Neunmann constructed a theory on them. He desired this theory to deal with the logical sides of both man made automata such as digital computers and natural systems like nervous system, brain and so on.Von Neumann's view of automata theory was very close to Wiener's view of robotics and each shaped the other. But von Neumann's automata theory placed more importance on logical and digital computers, while Wiener's cybernetics was focused physiology and control engineering.

One puzzle von-Neumann presented and in effect solved was: what style of logical organization is enough for an automaton to control itself in such a method that it reproduces itself?

He first expressed this question by using a kinematic automaton system, and later re-expressed and explained it in terms of cellular automaton system. Let's explain the kinematic system quickly and then advance the cellular system sufficiently to see how self-reproducing is achieved in it. Take into account a digital computer or automata which works synchronously and which is organized completely of switches (and, or and not) and delays (which delay pulses). We will denote these elements as computing elements. To denote an input capability and active output capability we will refer to five kinds of original elements: a kinematic (muscle like) element which can move elements in the region when signaled to do so by computing element; a cutting element which will separate two elements when signaled to do so by a computing element; fusing (joining or linking) element, which will connect two elements when signaled to do so by a computing element. A rigid element (e.g., girder or bar), which will provide rigid or structural support to assemblies of elements and a detecting element capable of recognizing each kind of elements and connecting this information to a computing element. We called an automaton created of these parts a kinematic automaton.

To model self-reproducing von Neumann used idealized neurons as computing elements. These neurons joined the function of switching and delay and were inactive in that they make no output except they are simulated. This category of neuron was operated by von Neumann in working on logical plan of the first stored program electronic computer, the EDVAC (Electronic Discrete Variable Automatic Computer).

Von Neumann's designed in the 1940s, self-replicating machine without the use of a computer description completed in 1966 by Arthur W. Burks after von Neumann's death defined the system as using 29 states, these states representing ways of signal carriage and logical operation, and acting upon signals represented as bit streams. A 'tape' of cells encrypts the sequence of actions to be executed by the machine. Using a writing head (termed a construction arm) the machine can construct a new form of cells, permitting it to make a full copy of itself, and the tape.

Self-reproducing system must able to perform it is task with the following conditions:

- Universal computation which the ability to work as a Universal Turing Machine and be able to work out any task.
- Build universality, in other words the ability to construct any structure in cellular space starting from a given description.

Von Neumann's -universal automaton was efficient for reading the input tape (made of cells of the cellular space), decoding the records on the tape, and using a constructing arm to build the configuration defined on the tape in an empty part of the cellular area. His machine was also able of backspacing the tape, making a copy of the tape, adding the copy to the structure just constructed; give a sign to the configuration that the construction process had ended, and take back the constructing arm to stop the operation.

The basic components of von-Neumann automata were a universal constructor A, which can build another automaton according to instruction I .Second element is a copier B, which can make copy according to the instruction tape I. Third, controller C, which links A and B and permits A to construct a new automaton according to I, B to copy instructions from I and glue them to the newly created automaton and splits the new automaton from the system A + B + C. Fourth, an automaton D, consisting of A, B and C. it's also have instruction tape ID explaining how to construct automaton D. Finally, an automaton called E containing D + ID as in figure 31.

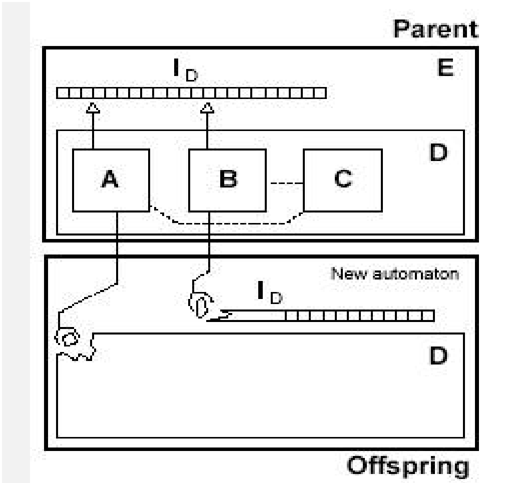


Figure.31 Neumann's self-reproducing automata [23]

The figure above illustrates Neumann's theory of self-reproduction in cellular automata. He developed a Universal Turing Machine with an input tape of instructions, ID, a Constructor A, a Copier B and a Controller C as we describe them in above paragraph. The figure shows that a parent automata E, reproduces its offspring where, the instruction tape instructions are replicated, along with the automaton D where offspring is a precise copy of the parent. [23]

6. CONCLUSION

At the beginning of this paper we describe the overall history of CA. after that, we discuss one dimension CA, definition, structure, rules and applications and similarly for 2-dimension CA where One dimensional Cellular automata is valuable computation machine for applications like cryptography, algorithms, modeling physical systems and Designing fractal pattern and two dimensional CA algorithms are used in different applications. It is mostly used in image processing because its structure is very similar to an image. There are some published works in the field of cellular automata with application in image processing; these works apply some rules on image to make different modifications.

At the last section of this paper, we illustrated the three major characteristics of CA; dimension represented by d, states for each cell k, and radius r.The dimension n identifies the composition of cells, a one dimensional line, two dimensional etc. The radius expresses the number of cells in each direction that will have an influence the updating of a cell.In case of cellular automata with higher dimension it must be indicated whether the radius refers only to directly adjacent cells only or involves diagonally adjacent cells too. As we discuss classification in 1-Dim CA one of them was Wolfram classification which a classification techniques was offered by Wolfram having four classes discussed above the second classification type was Li and Packard have established a classification system depending on Wolfram's system with six styles. Another study show that one dimension CA can be classified based on its properties with support of five factors number of States, amount of Neighborhood, Size of the cellular automata, Rules either its Uniform or hybrid and Number of times the development requires where these five factors affect number of rules in CA. Wolfram classed one-dim CA rules depending on their complexity. While, classification of cellular automata rules based on their properties theory based on the behavior of the rule in the n number of iteration. After that we talked about one of the most famous examples in the cellular automata in games field is Conway's Game of Life and its states and updating rules. Also, one of the important issues in CA calls majority problem/density classification with one dimension and two dimension cellular automata was discussed. We mentioned the concept of Toom rule developed by Andrei Toom for 2-dimension CA which strong and simple than other rules such as Reynaga's rule but, via study the property of rule which give author deprived result as classifier and the more worst the its fall even with small space automata as he/she work with L =9 and using 10^5 random models and get quality result Q=0.536 from that there is 0.7% are not classified well the bad quality here is result of bad classification while studied automata with L=21 where half of cells was misclassified while, Reynaga's rule was better than toom's rule but it still weak if we compare it with other rules in 1-dim or 2-dim automata statistical study of density classification/ majority problem by Nazim with 10^5 model for L=21 which record quality to 0.61. Everything in our world— just a pattern in a cellular automaton, Betros study that to proof the truth of this assumption in 2010 was viewed in the paper. At the end of discussing the most important issues in cellular automata the self-reproducing property was discussed.

References

[1] Wolfram. S. (1986). Theory and applications of cellular automata (Vol. 1). Singapore: World Scientific.

[2] Hedlund, G. A. (1969). Endomorphisms and automorphisms of the shift dynamical system. Theory of computing systems, 3(4), 320-375.

[3] Adamatzky, A. (Ed.). (2010). Game of life cellular automata (p. 579). London:: Springer.

[4] Gardner, M. (1970). Mathematical games: The fantastic combinations of John Conway's new solitaire game "life". Scientific American, 223(4), 120-123.

[5] Wolfram, S. (1983). Statistical mechanics of cellular automata. Reviews of modern physics, 55(3), 601.

[6] Wolfram, S. (2002). A new kind of science (Vol. 5). Champaign: Wolfram media.

[7] N. Pitsianis, PH. Tsalides, G.L. Bleris, A. Thanailakis, and H. C. Card, “Deterministic One-Dimensional Cellular Automata”, Journal of Statistical Physics, Vol. 56, 1989

[8] K. J. Jegadish Kumar and V. Karthick, “AES S-Box Construction using One Dimensional Cellular Automata Rules”, International Journal of Computer Applications, Volume 110, No. 12, January 2015.

[9] Adam Clarridge, “Cellular Automata: Algorithms and Applications” Queen’s University, March 2009.

[10] Choudhury, P. P., Nayak, B. K., Sahoo, S., Rath, S. P. 2008. Theory and Applications of Two-dimensional, Null-boundary, Nine-Neighborhood, Cellular Automata Linear rules. arXiv: 0804.2346. cs.DM;cs.CC; cs.CV.

[11] Qadir F., Shah, J., Peer, M. A., Khan, K. A. 2012. Replacement of graphics translations with two dimensional cellular automata, twenty five neighborhood model, International Journal of Computational Engineering and Management, Vol. 15, Issue 4, pp. 33-39.

[12] CHIRA, CAMELIA, ANCA GOG, and RODICA IOANA LUNG. "COMPLEX SYSTEMS AND CELLULAR AUTOMATA.".

[13] Nayak, Deepak Ranjan, Prashanta Kumar Patra, and Amitav Mahapatra. "A Survey on Two Dimensional Cellular Automata and Its Application in Image Processing.".

[14] Ganguly, Niloy, Biplab K. Sikdar, Andreas Deutsch, Geo Rey Canright, and P. Pal Chaudhuri. "A Survey on Cellular Automata.".

[15] Kunkle, Daniel R. Automatic Classification of One-dimensional Cellular Automata, 2003.

[16] Choudhury, Pabitra, and Sudhakar Sahoo. "Classification of Cellular Automata Rules Based on Their Properties.", pp. 1-4.

[17] Briceño, Raimundo, Pablo Moisset De Espanés, Axel Osses, and Ivan Rapaport. "Solving the Density Classification Problem with a Large Diffusion and Small Amplification Cellular Automaton." Physica D: Nonlinear Phenomena 261 (2013): 70-80.

[18] Stone, Christopher, and Larry Bull. "Evolution of Cellular Automata with Memory: The Density Classification Task." *Biosystems* 97.2 (2009): 108-16.

[19] Nazim Fates. A note on the Density Problem in Two Dimensions. Automata 2012 - 18th International Workshop on Cellular Automata and Discrete Complex Systems, Sep 2012, La Marana, Corse, France.

[20] Praeteritorum, P. http://dabacon.org/pontiff/?p=7099,(2013, May 11). A Paradox of Toom's Rule Visited,May 18, 2015.

[21] Reggia, J., Chou, H., & Lohn, J. (n.d.). Cellular Automata Models of Self-replicating Systems. Advances in Computers, 141-183.

[22] Parihar, S. (n.d.). 1.0 Introduction. Retrieved May 17, 2015,http://sjsu.rudyrucker.com/~shruti.parihar/paper/.

Samiah J. Alzahrani received B. S. degree in computer science in 2009 from Taif University. Currently, Computer training supervisor at Technical and Vocational Training Corporation, And Master student in King Abdul-Aziz University.

Lujain Natto received B. S. degree in computer science in 2013 from King Abdul-Aziz University. Currently, Master student in King Abdul-Aziz University.

Wafa S. Alshehri received B. S. degree in computer science in 2006 from Taif University. Currently, Teacher Assistant at Taif University, and Master Student in King Abdul-Aziz University.

Mariam O. Alrashidy received B. S. degree in Software Engineering in 2012 from Hail University. Currently, study Master in King Abdul-Aziz University.