Cellular Automata:

Finest Problem Solving Approach for Complex Problems

Jyoti Chaturvedi

Researcher,
Department of Mathematics,
Faculty of Science,
Dayalbagh Educational Institute,
Dayalbagh, Agra -282005.

Gursaran

Supervisor,
Professor, Department of Mathematics,
Faculty of Science,
Dayalbagh Educational Institute,
Dayalbagh, Agra -282005.

***Abstract-* cellular automaton is the problem solving approach that works on simple rules to give solution in effective way. In this approach a cellular model is built for the problem space, several rules are implemented to predict the state of the system at some time instance.**

Cellular automata deals with only local information of the system cell and provides an excellent platform to solve complex problems. This paper provides a brief introduction to cellular automata (CA), its basic ingredients and application of CA in cryptography.

***Keyword-* Cellular automata, cellular system, ingredients of CA, application to cryptography.**

I. Introduction

This universe is full of complex problems and we humans always try to find the solution to the problems in front of us. Since the problems of the universe are so complex, therefore the approach to find the solution is also difficult. Humans have developed many different branches for investigation e.g. mathematics, physics, chemistry, art etc. to solve problems and predict the result in various manner. But the approach to get the solution for several problems using the old methods is very much time consuming as well as complex.

To solve the problems of this universe in less time and with minimum effort humans have developed computers which solves extremely complex problems in few second. In the direction of automation of the problem solving approach cellular automata has found to be really efficient.

A cellular automaton builds the model of the universe and treats it as a system (cellular system) to deal with. It is convenient to study some system by breaking it in small parts, study each part, find the influence of parts on each other and then predict the behavior of the system as a whole. With the influence of this concept cellular system is taken to be the system i.e. made up of collection of cells. These cells are made up of same material but the behavior of each is different. This is because of the dependence of current cell upon previous state of the cell as well as influence of the states of the other cells near it. Interaction of these cells produces a global behavior.

II. Motivation

The concept of CA was initiated in the early 1950's by J. Von Neumann and Stan Ulam. Ulam was interested in construction of graphics using simple rules. In his construction, the two dimensional space was divided into number of cells. A state is associated with each cell i.e. on or off and certain neighborhood rules were used to generate new state.

Neumann used Ulam concept to work on self – reproductive machine. Von Neumann showed that a cellular automaton can be universal. He devised a CA, each cell of which has a state space of 29 states, and showed that the devised CA can execute any computable operation.

However, due to its complexity, Von Neumann rules were never implemented on a computer.

IJSER

III. Basic ingredients of cellular system

Cellular automata works on cellular systems. Every system is associated with some information and identities. In cellular system this basic information is hidden in its various constitutes. These components (constitutes) are as follows:

A. Cellular space

Cellular system is a system that is broken in small pieces (cells).the space of the collection of these cells is called the cellular space. These cells can reside in a one dimensional or 2 dimensional or any n-dimensional space. We usually deal upto 3 dimensional space because we can easily visualize upto 3 dimensional spaces and cellular automata strongly deals with pictorial representation of the problem solution. Some of the cellular spaces are shown below in figure 1.

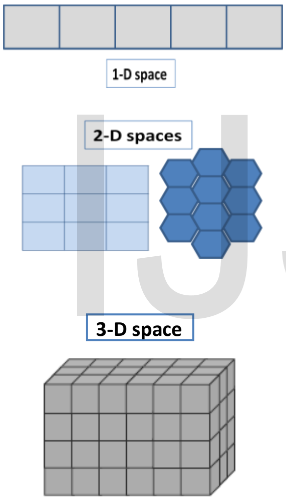


Figure 1. Different types of cellular spaces

B. State and State set

At certain time instance every cell in the system has some condition and some information associated. The above things specifies the state of the cell at any instance of time. And state set is the set of all possible values that can be taken by a state of the cell in the system. A special state that represents the resting or inactive condition of the cell is called quiescent state.

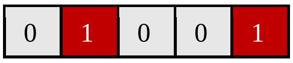


Figure 2. State of the cells in a system:
0 (grey) cell is dead, 1(red) cell is alive.

C. Time variable

Nothing is stationary in this world. Therefore the model of any universal object in cellular space i.e. cellular system changes with time , hence the state of the cells changes with every time instance. This time variable usually takes discrete values, so that we easily trace the system at every instance of time.

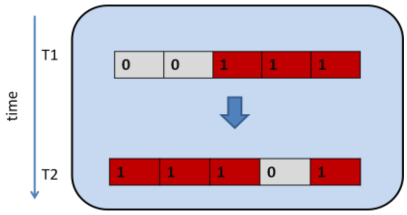
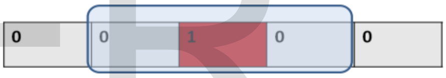


Figure 3. Change in cellular system from time T1 to T2.

D. Neighborhood

In the system of cells, cells interact with each other and affect the behavior, hence the state of the cell. collection of cells that directly influences the future state of the current cell (including current cell) is termed as neighborhood of the current cell.



IJSER

Figure 4. Neighbourhood of central cell: inside transparent rectangle.

There are several kinds of neighbourhoods defined below:

- Neumann neighbourhood- in this kind of neighbourhood the cells in neighbourhood are the cells in the north , south, east and west of the given cell as shown in figure 5.

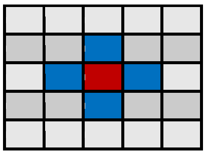


Figure5. Neumann neighbourhood

- Moore neighbourhood- here neighbourhood cells are the cells in north, north-east, east, south- east, south, south-west, west, north-west as shown in figure 6.

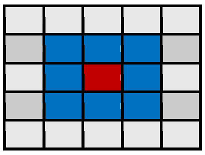


Figure 6. Moore neighbourhood.

There may be many other neighbourhoods such as extended Moore neighbourhood etc.

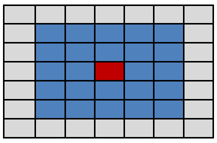


Figure 7. extended moore neighbourhood

E. *State transition function*

The rule according to which a cell acquire its future state on the basis of its own and neighbors' current state is called the state transition rule / function. In figure 8, one of the transition rules for 1 D space with 3 neighbors is shown.

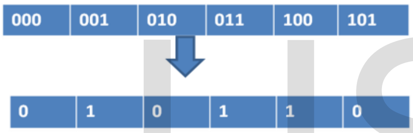


Figure 8. Transition rule for 1 D space

IJSER

On the basis of dependence of the current cell upon its neighbours in different manner, transition rules can fall in different categories that are described below.

- Standard rule- new state of the cell is related to the state of its neighborhood cells.
- Totalistic- new state depends only upon the sum of values of neighborhood states (including it). If state of the cells at time t be represented as $s_i(t)$ and $\phi_j(t)$ be the transition function for cell j, then mathematically we can write:

$$\phi_j(t) = \sum_{i=1}^{n} s_i(t)$$

Where i represents the number of the neighbour of the cell.

- Outer totalistic - New state depends upon the sum of values of neighborhood states (excluding it). If state of the cells at time t be represented as $s_i(t)$ and $\phi_j(t)$ be the transition function for cell j, then mathematically we can write:

$$\phi_j(t) = \sum_{\substack{i=1 \\ i \neq j}}^{n} s_i(t)$$

where i represents the number of the neighbour of the cell.

- Symmetric- rule is called symmetric with respect to permutation if it is not affected by the permutation of the neighborhood cells.

F. *Boundary conditions*

For a finite space boundary cells do not have enough neighbors as shown in figure 9, so that we can apply transition rule on them.

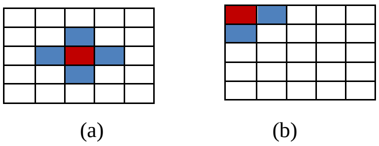


Figure 9. (a). Neumann neighbourhood with 4 surrounding neighbouring cells. (b). only two neighbouring cells of the boundary cell.

To solve this problem there are various boundary conditions described below and shown in figure 10.

- Periodic- Gluing opposite boundary cells to form a toroid.
- Assigned- A virtual cell is assumed as neighborhood with some state.
- Adiabatic- Copying boundary cell itself of cellular space to form neighborhood cell.
- Mirror- Copying the cell state of the cell next from the boundary cell.
- Absorbing- Simulate the finite space the behavior of the infinite space.

According to the problem one or more of these boundary conditions are used.

G. *Initialization and termination*

Updating process of space to produce new generation starts from some initial state. Assigning this initial state to the system is called initial condition. Also updating process should stop at some state. The condition to stop updating process is called termination condition.

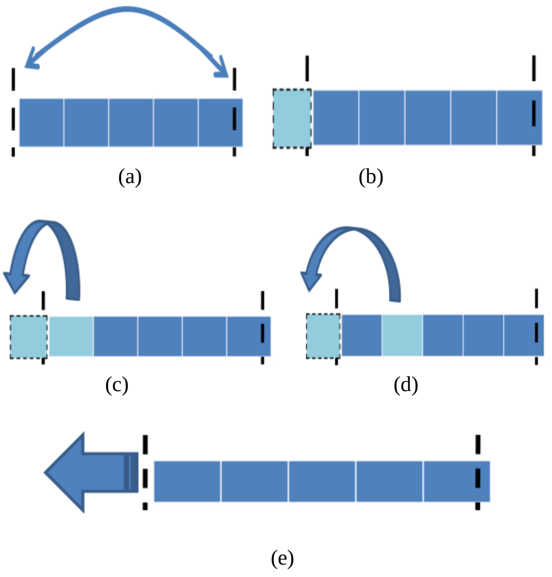


Figure 10. Boundary conditions (a) periodic, (b) assigned, (c) adiabatic, (d) mirror, (e) absorbing

IV. Definition of cellular automata

We make the model of the problem, we want to solve and then we predict the outcome of problem with the help of working of that model. Cellular automata (plural of automaton) is basically computer simulation that emulate law of nature. More formally we can say that

"Consider a space divided into cells, where each cell is repeatedly "updated" to a new state in an evolving sequence. A program of this nature is specifically called a cellular automaton when it is 1) parallel, 2) local, and 3) homogeneous (optional)."

In the above definition cellular automata has three properties that are described as follows:

- Parallelism- Cellular system constitutes evolve simultaneously and independently with time, therefore amendment of states said to take in parallel.
- Locality- New state only depends upon previous state and its neighbourhood. Hence updating process of the cell does not need complete system.
- Homogeneity- The rules that are applicable to a single cell of the space are same for the whole space.

V. Why CA over equations?

There are many reasons which shows that cellular automata is somehow better that dealing with mathematical equations. Some of the reasons are given below.

- Equations are difficult to deal with because:
  - So many if-then conditions.
  - Deal with derivative with respect to population size N. it is assumed that N takes large values, so bad model if N is small.
  - Number of equations is very large even for a small and simple model.
  - Number of variables (hence the memory space to be consumed) is large.
- It is easier as well as effective to deal with pictures than to solve the problem with pure mathematical equations.

VI. Classification of CA's

CAs are categorized according to the states of the neighborhood used, rules of transition used or patterns occur in various generation etc. some of these categories are described below and shown in figure 11.

IJSER

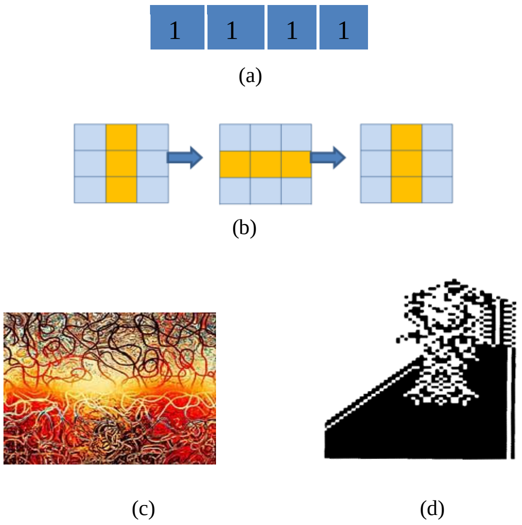
- Homogeneous state- cellular system in which state of all cells is same is called homogeneous state CA. E.g. the system in which all cells are dead or all cells are alive. The CA with above homogeneity is called homogenous with respect to space.

- Homogeneous transition- Cellular system in which transition rule for all the cells is same is said to be homogeneous transition CA. CA with this homogeneity is called homogenous with respect to time. If the system is just called homogeneous, then it is assumed to be homogenous with respect to both space and time.

- Periodic system- In this type of CAs pattern in a specific region periodically re-occurs. E.g. glider, eater etc.

- Chaotic- In this CA the pattern appeared in various generation are aperiodic random.

- Structured- Complex structure which may take very complex shapes but they do not remain forever. Here the complex structure appeared in the beginning and remain for some time (sometimes remain for longer duration), then first or third type of CAs are generated.



(a)

(b)

(c)

(d)

Figure 11. Types of CAs (a) homogenous state, (b) periodic CA with period 2, (c) random, (d) structured.

VII. Game of life

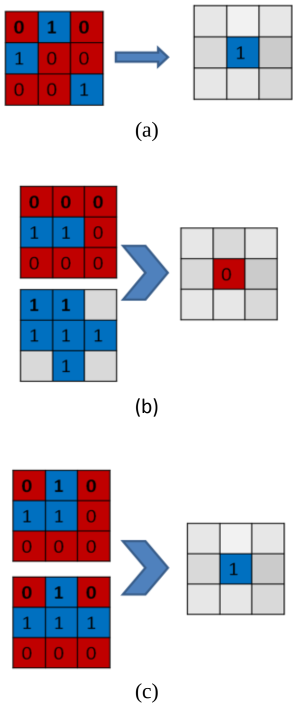
The most common and famous cellular automaton is game of life which was developed by John Conway in 1970.

Conway wanted to develop a game that cannot be predicted at any instance. So the objective of this particular application of CA is "*To make a game unpredictable as much as possible using simple rules*". To attain this objective he looked towards cellular automata.

In this game, we have a two dimensional space that is broken into number of cells. Each cell has a state i.e. a cell is taken either as alive or dead. The future state of the cell depends upon the state of its neighborhood according to the following rules:

- If cell is dead:
  - — Reproduction- It becomes alive if it has exactly 3 alive neighbors among eight surrounding neighbors.
  - — Remain dead otherwise.

- If cell is alive:
  - — Loneliness- It Dies if it has less than 2 alive neighbors.
  - — Happiness - Remains alive if it has 2 or 3 alive neighbors
  - — Overcrowding- Dies if it has more than 3 alive neighbors.



(a)

(b)

(c)

Figure 12. Rules: (a) reproduction, (b) loneliness and overcrowding, (c) happiness.

CA produces beautiful patterns on these rules for game of life, some of which are as shown in figure 13.

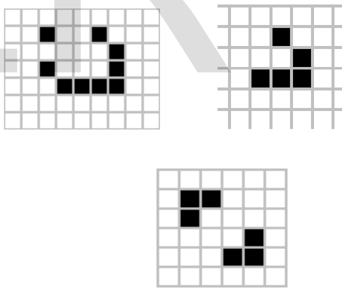


Figure 13. Patterns produced in game of life

VIII. Cellular automata and Cryptography

Cryptography is the branch of communication, which was developed to build some rules in order make secured communication between two or more parties sitting at a long distance apart.

To make communication secure, in cryptography there are two types of cryptosystem: one that deal with only one private key (private key cryptography) and the other deals with one private and one public key (public key cryptography). In general public key cryptography is more expensive than private key

IJSER

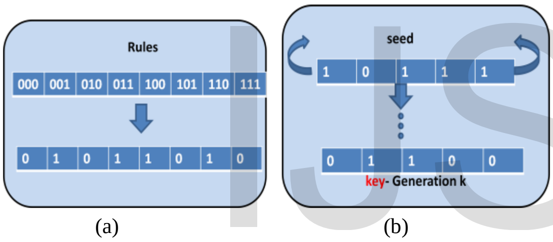
cryptography. Therefore we try to use private key scheme.

In private key cryptography we encrypt the message (that is to be secured) with one secret key and send to the receiving end. At the receiving end the message is decrypted using same secret key.

So the most important issue here to design the secret key. The key should be random (unpredictable) and used only once to make the scheme powerful. To achieve this goal CA is used to generate the key for one session.

Generation of the key- Sending as well as receiving end agree with a seed for CA. then take the key that is the output of the kth generation, then again take this key or some other seed to generate the key for next session in a similar manner and so on. Key generation for one session is shown in figure 14.

Use of CA makes the key i. e. generated a random one.



IJSER

Figure 14. (a) rules of CA, (b) key generation at kth generation.

IX. Complexity of CAs

The complexity of cellular automata directly depends upon the states of the cells used and the number of neighboring cells used. In fact, increment in these numbers increases the complexity, hence the time taken in execution increases.

Let we have k states that can be taken by a cell and n neighboring cell for a given cell. Then

Possible number of neighborhood states (say ns)= k^n.

Number of possible rules r = k^{kn}.

Hence the complexity of CA is $O(k^{kn})$.

X. Conclusion

A cellular automaton is the perfect platform to explain working of complex phenomenon with the limited information of local environment.

Modeling and analysis of the model of real world episodes is very simple and effective using CAs.

References

[1] Dario Floreano and Claudio Mattiussi, *Bio-Inspired Artificial Intelligence THEORIES, METHODS, AND TECHNOLOGIES,* The MIT Press Cambridge, Massachusetts London, England, 2008.

[2] Niloy Ganguly, Biplab K Sikdar, Andreas Deutsch, Georey Canright, P Pal Chaudhuri, "A Survey on Cellular Automata," Dresden University of technologies, Technical report- center for high performance computing, December 2003.

[3] Gould Tobochnik , Tapio Rantala, *lecture notes on Cellular automata*, Basics of Monte Carlo simulations, Kai Nordlund 2006.

[4] Andy Witkin and Michael Kass, *Lecture P4: Cellular Automata*, Princeton University, COS 126 , General Computer Science , Fall 2002 .

[5] M Phani Krishna Kishore, S Kanthi Kiran, B Bangaru Bhavya, S Harsha Chaitanya S, "A Novel Encryption System using Layered Cellular Automata", *Proceedings of the World Congress on Engineering* ,Vol I WCE, July 6 - 8, 2011, London, U.K.