Cellular Automata: Elementary Cellular Automata

Rupali Bhardwaj, School of Mathematics and Computer Applications, Thapar University, Patiala, India
Anil Upadhyay, Department of Mathematics, Bipin Chandra Tripathi Kumaon Engineering College, Dwarahat, India

ABSTRACT

Cellular automata (CA) are discrete dynamical systems consist of a regular finite grid of cell; each cell encapsulating an equal portion of the state, and arranged spatially in a regular fashion to form an n-dimensional lattice. A cellular automata is like computers, data represented by initial configurations which is processed by time evolution to produce output. This paper is an empirical study of elementary cellular automata which includes concepts of rule equivalence, evolution of cellular automata and classification of cellular automata. In addition, explanation of behaviour of cellular automata is revealed through example.

KEYWORDS

Cellular Automata, Characteristic Matrix, Elementary Cellular Automata, Rule Equivalence, State Transition Diagram

1. INTRODUCTION

The term cellular automata is a discrete model studied in mathematics, physics, computability theory, complexity science and theoretical biology. Cellular automata consist of a regular finite grid of cell; each cell encapsulating an equal portion of the state, and arranged spatially in a regular fashion to form an n-dimensional lattice. For each cell, a set of cells called its neighbourhood (usually including the cell itself) is defined relative to the specified cell. An initial state (time $t=0$) is given by assigning a state for each cell; according to some fixed rule (generally, a mathematical function) next state is created (time $t= t+1$) that determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighbourhood. For example, the rule might be that the cell is “on” in the next generation if and only if exactly one of the cells in the neighbourhood is “on” in the present generation otherwise the cell is “off” in the next generation. If same set of rules are used to update the stage of every cell in the lattice then cellular automata is called uniform otherwise it is called nonuniform cellular automata. A cellular automaton is a sextuple mathematical structure $\{ L, S, f, R_0, \eta, \delta_0 \}$, whereas:

- $L \in \mathbb{N}$, where $L$ is total number of cells, length of the cellular automaton.
- $S$ is the finite set of states alphabet, from which the generation of $c^t_k$ cell take their value, $c^t_k : \mathbb{Z} \rightarrow S$
• \( f \) is the evolution function of CA, \( f : S^0 → S \)
• \( R_\theta \) is the local rule where \( \theta ∈ 2^{S^0} \) is the rule number, \( R_\theta : S^0 → S^1 \)
• \( \eta ∈ N \) is the size of neighbourhood defined by \( \eta = 2 \times r + 1 \) where \( r \) is radius.
• \( \delta_0 = \{ c_i^0 \}, k = 0,1,2,...L \) is seed of the cellular automaton.

After a brief discussion of cellular automata, this paper will survey finite-width elementary cellular automata. Section 2 will explain construction of elementary cellular automata (ECA) and explain Wolfram’s naming conventions for rules used in evolution of CA. Section 3 documents mirrored equivalence, inversion equivalence, and the combination of both. Section 4 will demonstrate how state transition diagrams assists the study of finite-width elementary CA. Section 5 concludes the paper.

2. ELEMENTARY CELLULAR AUTOMATA

In their most general form, cellular automata are dynamical systems and their behaviour is illustrated using “space-time diagrams” in which the configuration of states in the \( d \)-dimensional lattice is plotted as a function of time. This lattice evolves through time in harmony with some type of rule. Cell’s next state is determined using a particular rule by the state that precedes it in lattice.

For a one-dimensional cellular automaton, the lattice \( L \) is an array of cells, and the transition rule \( \theta \) updates a cell value according to the values of a neighbourhood of \( \eta = 2 \times r + 1 \) cells around it, that means:

\[
f : S^{2r+1} → S
\]

\[
c_i^{t+1} = f(c_{i-r}^t, ..., c_{i-1}^t, c_i^t, c_{i+1}^t, ..., c_{i+r}^t)
\]

\[
c_j^t \in S, j = i - r, ..., i + r
\]

where \( t \) means the evolution time, also taking discrete values, and \( c_i^t \) means the value of the cell \( i \) at time \( t \) [1, 2]. Therefore, given a configuration of cell value, \( c_i^t \) at time \( t \), it will be updated by the application of the transition rule, \( R_\theta \) to generate the new configuration of cell value, \( c_i^{t+1} \). The simplest one-dimensional automata are one in which state space is binary \( \{ 0, 1 \} \) and the local neighbourhood is limited to nearest-neighbours \( (r = 1) \). This type of cellular automata will be henceforth referred as Elementary Cellular Automaton (ECA). Elementary CA rules use a “nearest neighbour” scheme \( (r = 3) \). That is, a cell at position \( k \) of space time diagram at time step \( t + 1 \) is given by the values of the cells at positions \( k - 1, k, k + 1 \) of the space time diagram at time step \( t \).

Rules take into account all possible three-cell combinations, hence there are \( 2^3 = 8 \) different configurations of neighbours and total of \( 2^{23} = 256 \) rules. For instance, the rule mentioned above (figure 1) would be-

\[
t: 111 110 101 100 011 010 001 000
\]
\[
t + 1: 0 1 0 1 1 0 1 0
\]
Related Content

Exploring the Effects of Hardware Performance, Application Design and Cognitive Demands on User Productivity and Perceptions
[www.igi-global.com/chapter/exploring-effects-hardware-performance-application/4467?camid=4v1a](www.igi-global.com/chapter/exploring-effects-hardware-performance-application/4467?camid=4v1a)

Critical LAN Success Factors: An Implementation Model
[www.igi-global.com/article/critical-lan-success-factors/55675?camid=4v1a](www.igi-global.com/article/critical-lan-success-factors/55675?camid=4v1a)
The Impact of Computer Processor Speed on End-User Productivity
[www.igi-global.com/article/impact-computer-processor-speed-end/3766?camid=4v1a](www.igi-global.com/article/impact-computer-processor-speed-end/3766?camid=4v1a)

Modeling Method for Assessing Privacy Technologies
[www.igi-global.com/chapter/modeling-method-assessing-privacy-technologies/18222?camid=4v1a](www.igi-global.com/chapter/modeling-method-assessing-privacy-technologies/18222?camid=4v1a)