Characterization of Non-reachable States in Irreversible CA State Space*

Sukanta Das[1] and Biplab K. Sikdar[2]

[1] Department of Information Technology
sukanta@it.becs.ac.in
[2] Department of Computer Science & Technology
biplab@cs.becs.ac.in
Bengal Engineering & Science University, Shibpur, West Bengal, India, 711103

Abstract. This paper targets characterization of the non-reachable states of 1-dimensional irreversible cellular automata (CA). A theoretical framework has been developed to design algorithms for computing the number of non-reachable states as well as the number of single cycle attractors of such a CA.

Keywords: Irreversible CA, non-reachable states, reachability tree, attractor.

1 Introduction

In the early 1950s, von Neumann and Stan Ulam [6] initiated the concept of cellular automata (CA). Stephen Wolfram first studied a family of simple 3-neighborhood 1-dimensional cellular automata that could simulate complex behaviors [7]. This structure attracted a large section of researchers working in the diverse fields and a specialized class of 1-dimensional CA, called linear/additive CA, had gained the primary attention [1]. The matrix algebraic tool provided the framework for characterization of linear/ additive CA. However, characterization of 3-neighborhood nonlinear CA is yet to be explored. This motivates us to concentrate on the non-linear CA - its characterization and analysis of its state space. In this work, we target the special class of CA called irreversible CA. The *non-reachable* states and the *attractors* of irreversible CA are characterized. The theoretical framework thus developed leads to the design of algorithms for computing the number of non-reachable states as well as the number of single cycle attractors in an irreversible CA.

2 Cellular Automata Basics

A Cellular Automaton (CA) consists of a number of cells organized in the form of a lattice. It evolves in discrete space and time, and can be viewed as an

* This research work is supported by the Sponsored CA Research Projects, Bengal Engineering and Science University, Shibpur, West Bengal, India-711103.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 160–167, 2008.

autonomous finite state machine (FSM). Each cell stores a discrete variable at time t that refers to the present state of the cell. The next state of the cell at $(t+1)$ is affected by its state and the states of its *neighbors* at time t. In this work, we concentrate on such 3-neighborhood CA (self, left and right neighbors), where a CA cell is having two states - 0 or 1. Therefore, the next state S_i^{t+1} of the i^{th} CA cell is specified by the *next state function* f_i as

$$S_i^{t+1} = f_i(S_{i-1}^t, S_i^t, S_{i+1}^t) \tag{1}$$

where S_{i-1}^t, S_i^t and S_{i+1}^t are the present states of the neighbors at time t.

The $\mathcal{S}^t = (S_1^t, S_2^t, \cdots, S_n^t)$ is the present state of an $n-$cell CA and

$$\mathcal{S}^{t+1} = (f_1(S_0^t, S_1^t, S_2^t), f_2(S_1^t, S_2^t, S_3^t), \cdots, f_n(S_{n-1}^t, S_n^t, S_{n+1}^t)) \tag{2}$$

If $S_0^t = S_n^t$ and $S_{n+1}^t = S_1^t$ (that is, left neighbor of the left most cell is the right most cell and vice versa), then the CA is referred to as *periodic boundary CA*. On the other hand, if $S_0^t = S_{n+1}^t = 0$, the CA is *null boundary.*

If the next state function of the i^{th} cell is expressed in the form of a truth table, then the decimal equivalent of its output is conventionally referred to as the 'Rule' $\mathcal{R}_i$ [7]. In a two-state 3-neighborhood CA, there can be a total of 2^8 (256) rules. Three such rules 90, 150, and 75 are illustrated in *Table 1.* The first

Table 1. Truth table for rule 90, 150 and 75

Present state :	111	110	101	100	011	010	001	000	*Rule*
(RMT)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
(i) Next State :	0	1	0	1	1	0	1	0	90
(ii) Next State :	1	0	0	1	0	1	1	0	150
(iii) Next State :	0	1	0	0	1	0	1	1	75

Note: RMT stands for *Rule Min Term.* The value 0/1 noted in $3^{rd}/4^{th}/5^{th}$ row shows the output of the three variable switching function.

row of the table lists the possible 2^3 (8) combinations of the present states of $(i-1)^{th}$, i^{th} and $(i+1)^{th}$ cells at time t. The last three rows indicate the next states of the i^{th} cell at $(t+1)$ for the rules, 90, 150 and 75 respectively.

Definition 1. *A rule is* **Balanced** *if it contains equal number of 1s and 0s in its* $8-$*bit binary representation; otherwise it is an* **Unbalanced** *rule.*

Definition 2. *The set of rules* $\mathcal{R} =< \mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_i, \cdots, \mathcal{R}_n >$ *that configures the cells of a* CA *is called the* **rule vector**.

The sequence of states generated (state transitions) during its evolution with time directs the CA behavior. The state transition diagram of an irreversible CA may contain *cyclic* (lies in a cycle) and *non-cyclic* states (*Fig.1*). Further, in an *irreversible* CA there are some states that are not reachable (*non-reachable* states) from the other state and some states are having more than one predecessor [4,5]. For example, the states marked as 5 and 13 of *Fig.1* are the non-reachable states. Whereas 15 and 7 have more than one predecessor.

An irreversible CA contains one or more cycles, called attractors. *Fig.1* contains two cycles – one of length 3 ($7 \rightarrow 3 \rightarrow 11 \rightarrow 7$) and other is of length 1 (15). This paper concentrates on the characterization of non-reachable states and the attractors of length 1 (single cycle attractors) in an irreversible CA. Here we refer such single cycle attractors as simply attractors. The next section introduces the concept of Reachability tree to formalize the characterization.

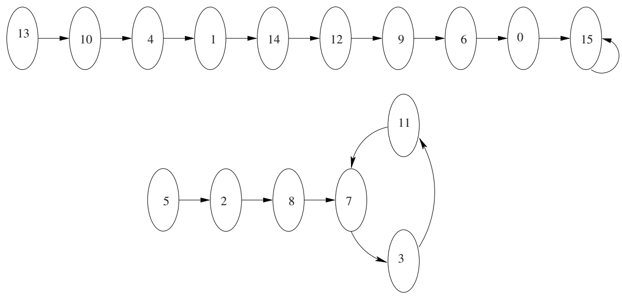

Fig. 1. State transitions of an irreversible $CA < 105, 177, 171, 75 >$

3 Reachability Tree

Reachability Tree, proposed in [2,3], is a binary tree that represents the reachable states of a CA. Each node of the tree is constructed with RMT(s) of a rule (*Section 2*). The left edge of a node of the tree is considered as the 0-edge and the right edge is as the 1-edge (*Fig.2*). The number of levels of the reachability tree for an $n-$cell CA is $(n + 1)$. Root node is at Level 0 and the leaf nodes are at Level n. The nodes of Level i are constructed following the selected RMTs of $(i + 1)^{th}$ CA cell rule $\mathcal{R}_{i+1}$, while computing the next state.

The number of leaf nodes in the reachability tree denotes the number of reachable states of a CA and a sequence of edges from the root to a leaf node, representing an $n-$bit binary string, is the reachable state. The binary string is formed assuming that the 0-edge and 1-edge represent 0 and 1 respectively.

During next state computation of a CA cell, the RMTs of the rule configuring the CA cell take the leading role. However, the RMTs of two consecutive cell rules $\mathcal{R}_i$ and $\mathcal{R}_{i+1}$ are related while the CA changes its state. Since the CA is in 3-neighborhood, the RMTs are of 3-bit. So, a three bit window can be considered that slides over the present state, from left to right, to get the next state [2]. If the RMT window for i^{th} cell is $(b_{i-1}b_ib_{i+1})$, $b_i = 0/1$, then the RMT window for $(i+1)^{th}$ cell will be either $(b_ib_{i+1}0)$ or $(b_ib_{i+1}1)$. In other words, if the i^{th} CA cell changes its state following the RMT k (decimal equivalent of $b_{i-1}b_ib_{i+1}$) of rule $\mathcal{R}_i$, then the $(i + 1)^{th}$ cell will generate its next state following the RMT

Table 2. Relationship between RMTs of cell i and cell $(i+1)$ for next state computation

RMT at i^{th} rule	RMTs at $(i+1)^{th}$ rule
0	0, 1
1	2, 3
2	4, 5
3	6, 7
4	0, 1
5	2, 3
6	4, 5
7	6, 7

Table 3. RMTs of the $CA < 8, 112, 44, 68 >$

RMT	111 (7)	110 (6)	101 (5)	100 (4)	011 (3)	010 (2)	001 (1)	000 (0)	Rule
First cell	d	d	d	d	1	0	0	0	8
Second cell	0	1	1	1	0	0	0	0	112
Third cell	0	0	1	0	1	1	0	0	44
Fourth cell	d	1	d	0	d	1	d	0	68

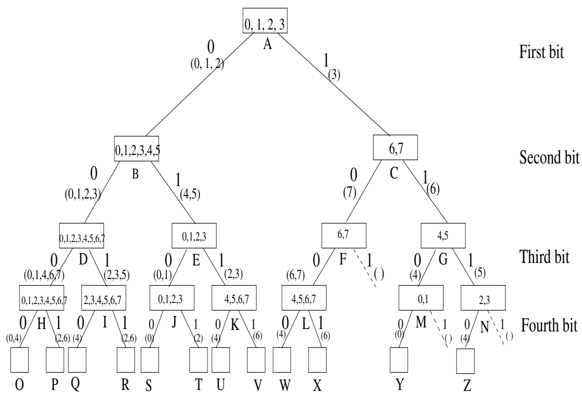

Fig. 2. Reachability Tree for the $CA < 8, 112, 44, 68 >$

$2k \ mod \ 8 \ (b_i b_{i+1} 0)$ or $(2k+1) \ mod \ 8 \ (b_i b_{i+1} 1)$ of rule $\mathcal{R}_{i+1}$. This relationship between the RMTs of $\mathcal{R}_i$ and $\mathcal{R}_{i+1}$, while computing the next state of a CA, is shown in *Table 2*. The relation, noted in the table, plays an important role for characterizing the CA behavior configured with different cell rules.

Fig.2 is the reachability tree for a $CA < 8, 112, 44, 68 >$. The RMTs of the CA rules are noted in *Table 3*. The decimal numbers within a node at level i represent the RMTs of the CA cell rule $\mathcal{R}_{i+1}$ following which the cell $(i+1)$ may change its state. The RMTs of a rule for which we follow 0-edge or 1-edge are noted in the bracket. For example, the root node (level 0) of *Fig.2* is constructed with RMTs 0, 1, 2 and 3 as cell 1 (rule 00001000) can change its state following any one of the RMTs 0, 1, 2, and 3. As the state of its left neighbor is always 0, the RMTs 4, 5, 6 & 7 are the $don't \ care$s for cell 1. It is obvious from *Fig.2* that there are 12 possible sequences of edges in the tree. That is, 12, out of 16, CA states are reachable and the rests are non-reachable. Based on the theory of

Reachability tree, we next report the proposed characterization of non-reachable states and the attractors of a CA.

4 Characterization of CA Targeting Non-reachable States and the Attractors

This section presents a scheme to characterize the irreversible CA states. It identifies the non-reachable states and also computes the number of non-reachable states of an irreversible CA in linear time. It also finds the number of single cycle attractors of a CA. The theoretical aspects of such characterization are formulated in the following theorems.

Theorem 1. *An $n-$cell irreversible CA contains at least 2^{n-3} non-reachable states.*

Proof. In 3-neighborhood, $\frac{1}{8}$ of the total CA states are to be determined by each of the 8 RMTs of i^{th} CA cell rule $\mathcal{R}_i$. Since the CA is irreversible, there is at least one RMT of $\mathcal{R}_i$ that causes an unbalanced reachability tree for the CA. Therefore, $\frac{1}{8}$ of total states are obviously non-reachable. Hence, the number of non-reachable states is at least $\frac{2^n}{8} = 2^{n-3}$.

Theorem 2. *An $n-$cell irreversible CA constructed only with the balanced rules contains at least 2^{n-2} non-reachable states.*

Proof. Let us consider the reachability tree for an $n-$cell irreversible CA, configured only with balanced rules, is balanced up to the i^{th} level and rule $\mathcal{R}_i$ is responsible for that. Since $\mathcal{R}_i$ is balanced, therefore, there exist at least 2 RMTs that cause the tree as unbalanced. As $\frac{1}{8}$ of total states are determined by an RMT, total number of non-reachable states for such CA is $(\frac{1}{8} + \frac{1}{8} = \frac{1}{4})$ of the total states. Hence an $n-$cell irreversible CA, configured with balanced rules, contains at least 2^{n-2} non-reachable states. Hence the proof.

Corollary 1. *An $n-$cell linear/additive irreversible CA contains at least 2^{n-2} non-reachable states.*

Proof. Since a linear/additive rule is balanced [2,3], the result is directly followed from *Theorem 2*.

We next propose an algorithm that calculates the number of non-reachable states of an irreversible CA utilizing the concept of reachability tree.

4.1 Computing the Number of Non-reachable States

The algorithm (*CalNonReachableStates*) assumes the variables S, an array of sets, and nos (the number of sets in S). The arrays $oldWeight$ and $newWeight$ are used to store the number of states that may be reachable and to store the number of states that are non-reachable respectively. The number of non-reachable states are finally stored in the variable *NS*.

Algorithm 1: **CalNonReachableStates**
Input: n (CA size), $Rule[n][8]$ (CA).
Output: number of non-reachable states.
Step 1: Find $S[1] = \{j\}$, where $Rule[1][j] = 0$ and $1 \leq j \leq 3$,
and $S[2] = \{j\}$, where $Rule[1][j] = 1$ and $1 \leq j \leq 3$.
If $S[i] = \phi$ (i=1/2), set NS := 2^{n-1}, oldWeight[1] := 2^{n-1} and $nos := 1$.
Otherwise, set oldWeight[1] := 2^{n-1}, oldWeight[2] := 2^{n-1} and $nos := 2$.
Step 2: For $i = 2$ to $n - 1$ do 2.1 to 2.4
2.1 For $j = 1$ to nos
Determine RMTs for the next level nodes from $S[j]$ following *Table 2*.
Distribute these RMTs of i^{th} rule with value 0 into $S'[2j-1]$ and 1 into $S'[2j]$.
Set newWeight[2j-1] := oldWeight[j]/2 and newWeight[2j] := oldWeight[j]/2.
If $S'[k] = \phi$, set NS := NS + newWeight[k], where $k = 2j - 1$, $2j$.
2.2 Replace RMTs 4, 5, 6 and 7 by equivalent RMTs 0, 1, 2 and 3 respectively for each $S'[k]$.
2.3If $S'[k] = S'[k']$ for any k', set oldWeight[k] := newWeight[k] + newWeight[k'];
otherwise, set oldWeight[k] := newWeight[k].
2.4 Assign unique sets of S' to S, and nos := number of sets in S.
Step 3: For $j = 1$ to nos
Determine next RMTs of $S[j]$, of which 2 are invalid since it is the last rule.
Distribute these RMTs of last rule with value 0 into $S'[2j-1]$ and 1 into $S'[2j]$.
If $S'[k] = \phi$, then set NS := NS +oldWeight[k]/2, where $k = 2j - 1, 2j$.
Step 4: Report the value of NS as the number of non-reachable states of the CA.

Complexity: Since *Algorithm 1* uses a loop in *Step 2* that depends on n, and the maximum value of nos. nos is constant. Therefore, the time complexity of the algorithm is $O(n)$.

4.2 Computing the Number of Attractors

Since the next state of a single cycle attractor is the attractor itself (attractor 15 of *Fig.1*), there should be at least one RMT (*Table 1*) of each cell rule ($\mathcal{R}_i$) for which the CA $\mathcal{R}$ cell (i) does not change its state. For example, the RMT $x0x$ ($x = 0/1$) of a rule is considered to find the next state of cell i when the current states of its left neighbor ($(i-1)^{th}$ cell), self (i^{th}) and right neighbor ((($i+1)^{th}$ cell) are x, 0 and x respectively. It implies, if the RMT is '0', the state change of the cell (i) is $0 \to 0$. That is, for the rule $\mathcal{R}_i$, if the RMT 0 (000), 1 (001), 4 (100) or 5 (101) is 0, the CA cell i configured with $\mathcal{R}_i$ does not change its state. Similarly, if the RMTs 2 (010), 3 (011), 6 (110) or 7 (111) is 1 in $\mathcal{R}_i$, the cell configured with $\mathcal{R}_i$ can stick to its current state in the next time step. For

RMTs	111 (7)	110 (6)	101 (5)	100 (4)	011 (3)	010 (2)	001 (1)	000 (0)	
	1	1	0	0	1	1	0	0	Rule for cell i

Fig. 3. RMTs of rule 204

example, if a CA cell is configured with the rule 204 (*Fig.3*), all RMTs of the rule help formation of attractors.

Property 1: A rule $\mathcal{R}_i$ can contribute to the formation of single cycle attractor(s) if at least one of the RMTs 0, 1, 4 or 5 is 0, or the RMT 2, 3, 6 or 7 is 1.
If any rule does not maintain *Property 1*, the CA can not have single cycle attractors. The following algorithm *CalNoOfAttractors* scans a CA rule vector $\mathcal{R}$ from left to right and explores all the attractors of the CA.

Algorithm 2: **CalNoOfAttractors**
Input: n (CA size), $Rule[n][8]$ (CA).
Output: NoA (number of attractors).
Step 1: If any rule does not maintain *Property 1*, return $NoA = 0$.
Step 2: If RMT j (j =0, 1, 2, 3) is capable of generating the attractors, assign $S[1] = \{j\}$, where $Rule[1][j] = 0$, $S[2] = \{j\}$, and $Rule[1][j] = 1$.
If $S[i] = \phi$, then set oldWeight[1] := 2^{n-1} and $nos := 1$, where $i = 1$ or 2.
Otherwise, set oldWeight[1] := 2^{n-1}, oldWeight[2] := 2^{n-1} and $nos := 2$.
Step 3: For $i = 2$ to $n - 1$ do 3.1 to 3.4
3.1 For $j = 1$ to nos
Determine RMTs for the next level nodes from $S[j]$ following *Table 2*.
Remove the RMTs that are not capable of generating attractors.
Distribute these RMTs of i^{th} rule with value 0 into $S'[2j - 1]$ and 1 into $S'[2j]$.
Set newWeight[2j-1] := oldWeight[j]/2 and newWeight[2j] := oldWeight[j]/2.
3.2 Replace RMTs 4, 5, 6 and 7 by equivalent RMTs 0, 1, 2 and 3 respectively for each $S'[k]$.
3.3If $S'[k] = S'[k']$ for any k', set oldWeight[k] := newWeight[k] + newWeight[k'];
Otherwise, set oldWeight[k] := newWeight[k].
3.4 Assign unique sets of S' to S, and nos := number of sets in S.
Step 4: For $j = 1$ to nos
Determine next RMTs of $S[j]$, of which 2 are invalid since it is the last rule.
Remove the RMTs that are not capable of generating attractors.
Distribute these RMTs of last rule with value 0 into $S'[2j - 1]$ and 1 into $S'[2j]$.
If $S'[k] = \phi$, then set NoA := NoA +oldWeight[k]/2, where $k = 2j - 1, 2j$.
Step 5: Report NoA as the number of attractors.

Complexity: The complexity of *Algorithm 2* is also $O(n)$ as it is for *Algorithm 1*.

5 Conclusion

This paper presents an efficient scheme to calculate the number of non-reachable states of an irreversible CA in linear time. An algorithm is also proposed that computes the number of single cycle attractors of such a CA. A theoretical framework has been reported to characterize the non-reachable states as well as the attractors of the CA.

References

1. Pal Chaudhuri, P., Roy Chowdhury, D., Nandi, S., Chatterjee, S.: Additive Cellular Automata - Theory and Applications, vol. 1. IEEE Computer Society Press, California (1997)
2. Das, S., Sikdar, B.K.: Classification of CA Rules Targeting Synthesis of Reversible Cellular Automata. In: Proceedings of International Conference on Cellular Automata for Research and Industry, ACRI, France, pp. 68–77 (September 2006)
3. Das, S., Sikdar, B.K., Chaudhuri, P.P.: Characterization of Reachable/Nonreachable Cellular Automata States. In: Proceedings of Sixth International Conference on Cellular Automata for Research and Industry, ACRI, The Netherlands, pp. 813–822 (October 2004)
4. Moore, E.F.: Machine models of self reproduction. In: Burks, A.W. (ed.) Essays on Cellular Automata, University of Illinois Press, Urbana (1970)
5. Myhill, J.: The converse of moore's garden of eden theorem. In: Proceedings of American Mathematical Society, vol. 14, pp. 685–686 (1963)
6. von Neumann, J.: The theory of self-reproducing Automata. In: Burks, A.W. (ed.), Univ. of Illinois Press, Urbana (1966)
7. Wolfram, S.: Cellular Automata and Complexity - Collected Papers. Addison-Wesley, Reading (1994)