IDENTIFICATION OF NON-UNIFORM PERIODIC BOUNDARY CELLULAR AUTOMATA HAVING ONLY POINT STATES

Nazma Naskar   Sumit Adak   Sukanta Das

Presented by

Sukanta Das

Department of Information Technology
Bengal Engineering and Science University, Shibpur
West Bengal, India. sukanta@it.becs.ac.in

OBJECTIVE

- Non-uniform 1D cellular automata under periodic boundary condition (PBCA).

- Identifying PBCA having only point state attractors.

APPROACH

- Count the point state attractors
- Count cyclic states.
- If both are same, we declare that the automaton is having only point states.

- To do these task, reachability tree, a discrete tool for characterizing cellular automata, has been utilized.

CELLULAR AUTOMATA

- In a 3-neighbourhood dependency, the next state function denoted as

$$S_i^{t+1} = f_i(S_{i-1}^t, S^t_i, S^t_{i+1}).$$

- The collection of states $S^t(S^t_1, S^t_2, \cdot \cdot \cdot, S^t_n)$ of cells at time t is the present state of CA having n cells

Periodic Boundary CA : $S^t_0 = S^t_n$ **and** $S^t_{n+1} = S^t_1$

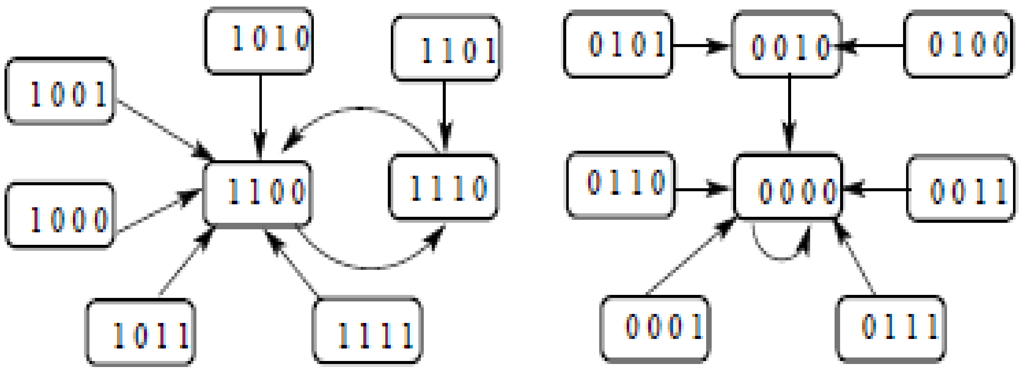
Present state: (RMT)	111 (7)	110 (6)	101 (5)	100 (4)	011 (3)	010 (2)	001 (1)	000 (0)	Rule
(i) Next State:	1	1	0	0	1	1	0	0	204
(ii) Next State:	1	1	1	1	0	0	0	0	240
(iii) Next State:	0	0	1	1	0	0	0	0	48
(iv) Next State:	0	0	0	0	0	0	0	0	0

Truth table for rule <204, 240, 48, 0>

- The decimal equivalent of the 8 outputs is traditionally called as Rule.

NON-UNIFORM CA

- The set of rules R=< R_1, R_2, …., R_i,……… R_n > where cell *i* acts with R_i is called rule vector.
- If $R_1 = R_2 = \cdot \cdot \cdot = R_n$, then the CA is uniform CA, otherwise non-uniform CA.

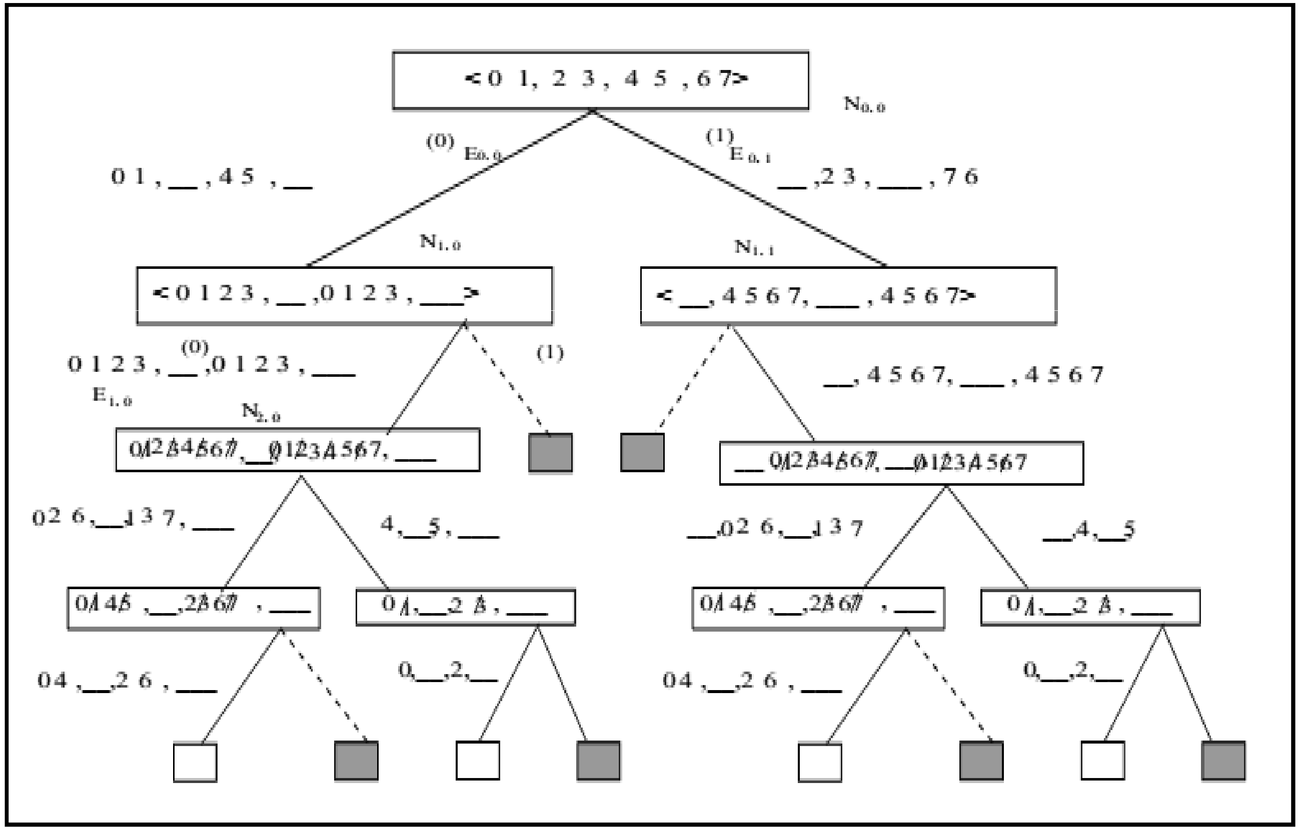


State transition diagram of CA <204, 240, 48, 0>

REACHABILITY TREE

- Reachability tree (RT) is a characterization tool for periodic boundary CA.
- It is a binary tree that represents the reachable states of a CA.
- It is a set of nodes (N) and edges (E) where each edge / node represents sets of RMTs.
- In case of periodic boundary CA, RMTs are divided into four groups.
- For periodic boundary CA we discard odd RMTs from first two sets and discard even RMTs from last two sets at $(n-2)^{th}$ level.
- In $(n-1)^{th}$ level, we allow those RMTs only in a set which is capable to produce RMTs at corresponding set number in edge of first level.

Reachability Tree for PBCA



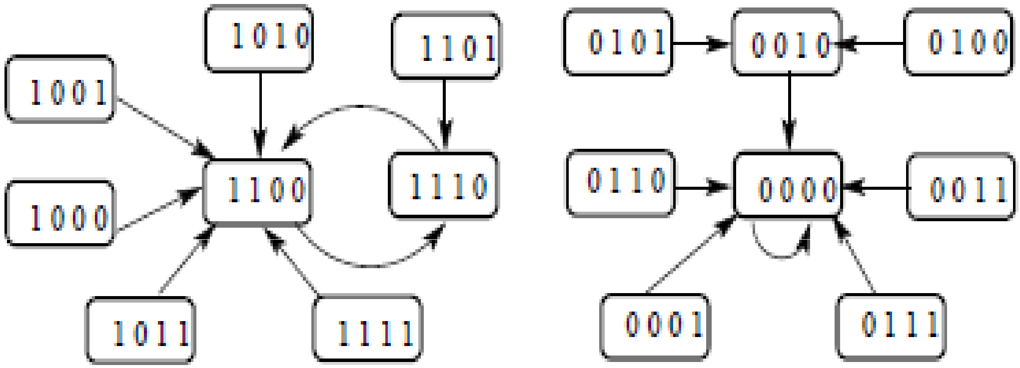
Reachability tree of CA <204, 240, 48, 0>

The left edge of tree represent 0-edge, where as right edge represent 1-edge.

For i^{th} level, each edge is constructed according to the RMTs of Rule R_{i+1}.

Nodes of ith level is constructed by successor RMTs of corresponding edge at $(i-1)^{th}$ level. Edge (node) of RT is denoted by $E_{i.j}$ ($N_{i.j}$) where i is level index and j is the j^{th} edge (node) of i^{th} level, value of j varies from 0 to 2^i-1.

Processing of State Transition Diagram



State transition diagram of CA <204, 240, 48, 0>

1. Process the state space of the CA and remove the non-reachable states.

2. Identify new non-reachable (which are originally acyclic but reachable) states in the processed state space.

3. Repeat (1) and (2) until no new non-reachable states can be identified.

Processing of Reachability Tree

Reachability tree and state transition diagram:

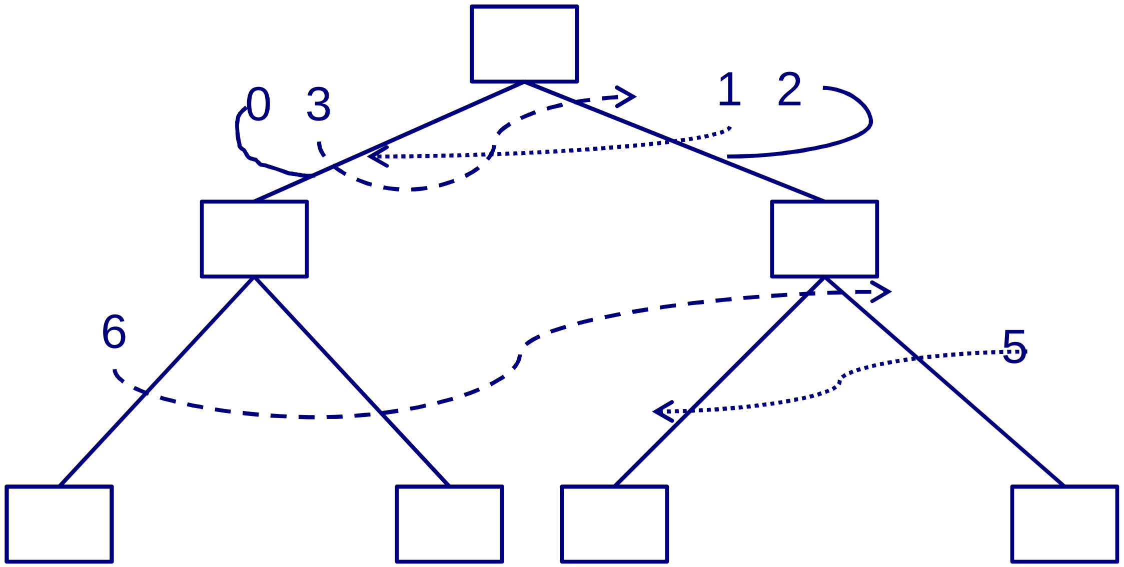
Our task is :

- To identify non-reachable states.
- To remove non-reachable states from the state transition diagram.

a) A sequence of edges from root to a leaf node associates a reachable state .
b) At least one RMT sequence which corresponds to predecessor of the state.
c) A relationship among the states can be traced.
d) Since the RSs and the states, both of an automaton can be traced in the tree, which RS corresponds to what state can be identified.

To trace the relation, we merge the nodes/edges of the tree (TREE MERGING)

Tree Merging Procedure

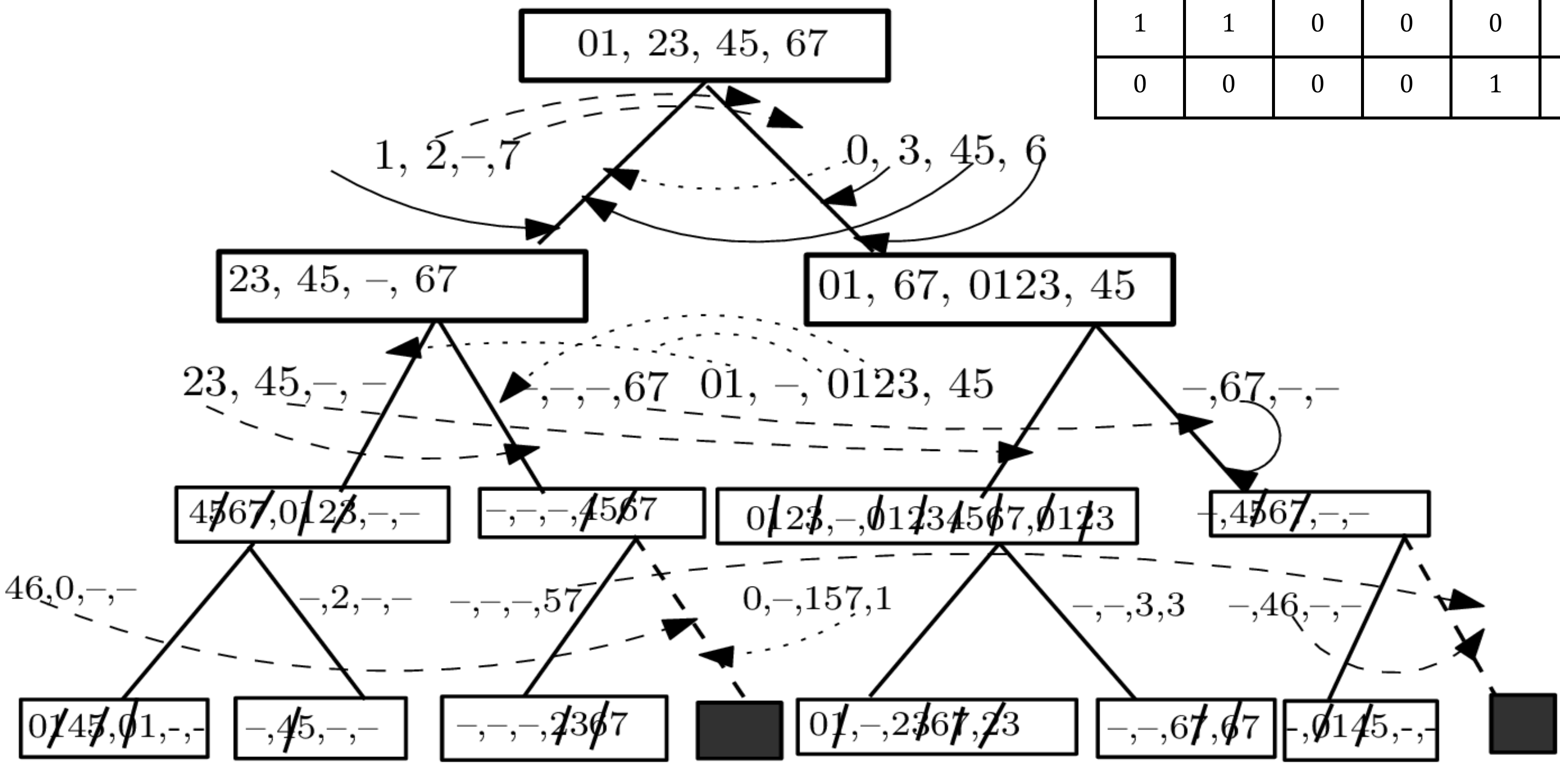


Merging Type:

1. Self merging
2. Forward merging
3. Backward merging
4. Cross merging

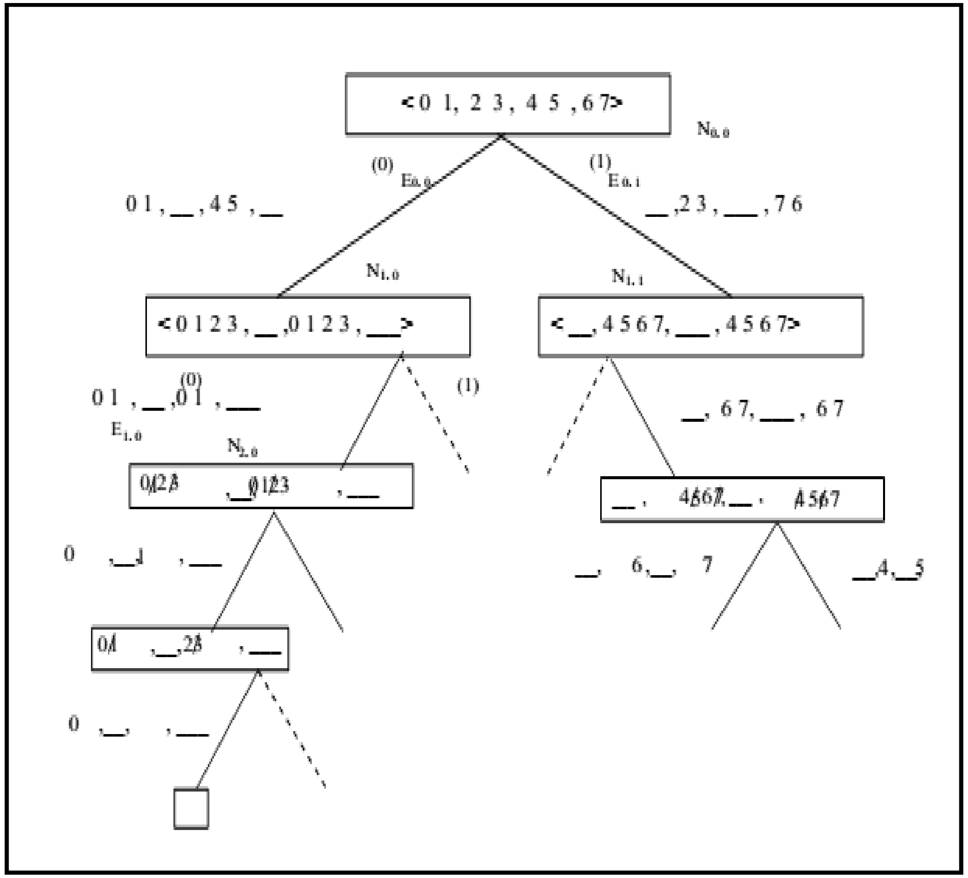
Tree Merging

111	110	101	100	011	010	001	000	Rule
0	1	1	1	1	0	0	1	121
1	1	0	0	0	0	0	0	192
0	0	0	0	1	1	0	0	12



Reachability tree with merging of CA <121, 192, 12>

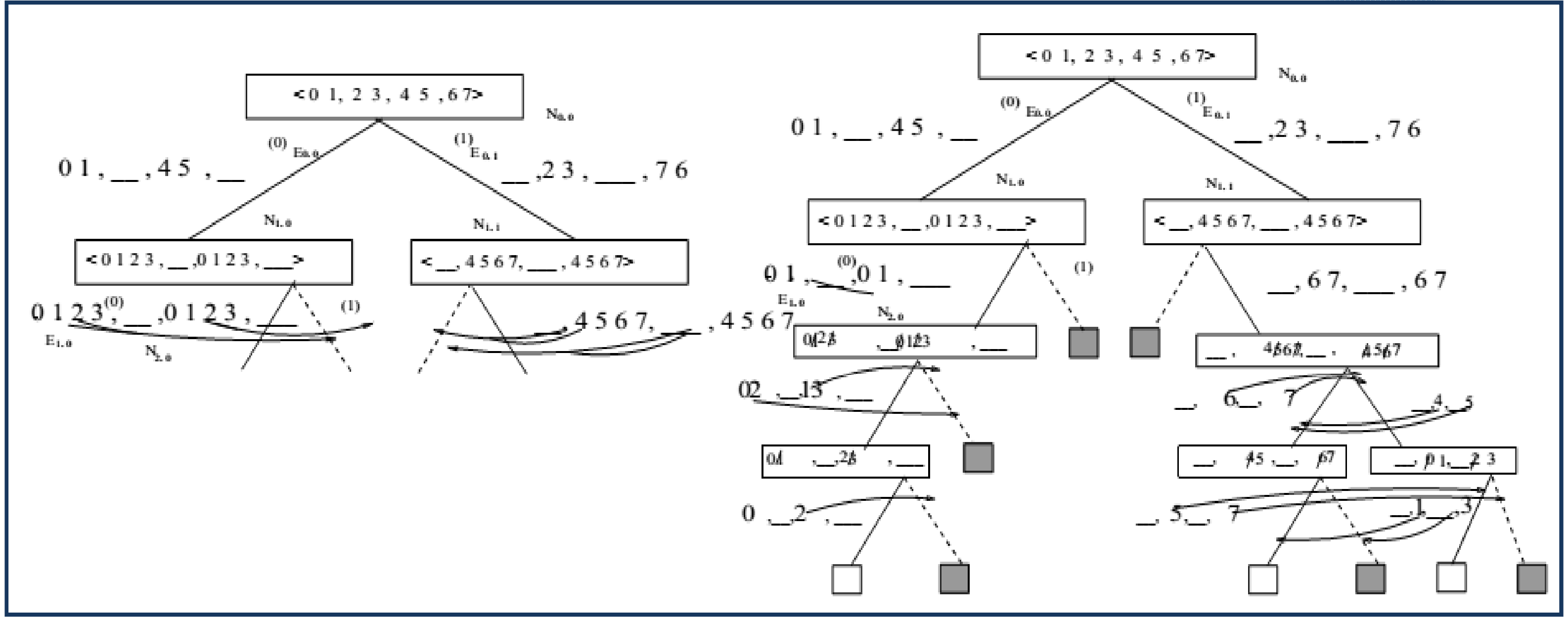
Algorithm 1: Counting Point States in PBCA



1. Perform merging.
2. Preserve only those RMT, which are self merged.
3. For (n-2)th and (n-1)th levels node, discard the RMTs as per rule
4. Indentify the state left at level.
5. Report the nstates as point state attractor

State 0000 is point state

Algorithm 2: Counting Cyclic States in PBCA



1. Perform merging.
2. Discard those RMT, which are fall on self merged RMT and non reachable edges.
3. For (n-2)th and (n-1)th levels node, discard the RMTs as per rule
4. Count number of RMTs left in leaf level.
5. Report the number as cyclic states.

Three States are cyclic state

VERIFICATION

Algorithm1: Counting of point state attractors

Rules	Point state attractors
< 204, 240, 48, 0 >	1
< 5, 73, 200, 80 >	3

Algorithm2: Counting of cyclic states

Rules	Cyclic states
< 204, 240, 48, 0 >	3
< 5, 73, 200, 80 >	3

PBCA $<5, 73, 200, 80>$ is having only point states.

CONCLUSION

- Here we have covered some aspects of non-uniform periodic boundary CA having only point states.
- To do this we take help of tree merging technique.
- Using the concept of tree merging, an algorithm for identifying point states attractors from CA state space is reported.
- Another algorithm is also reported to count cyclic states in CA state space.
- Using these two algorithm we can identify a PBCA having only point states

REFERENCES

[1] P Pal Chaudhuri, D Roy Chowdhury, S Nandi, and S Chatterjee. Additive Cellular Automata - Theory and Applications, volume 1. IEEE Computer Society Press, USA, ISBN 0-8186-7717-1, 1997.

[2] Sukanta Das and Biplab K. Sikdar. Characterization of 1-d periodic boundary reversible CA. Electr. Notes Theor. Comput. Sci., 252:205_227,2009.

[3] Nazma Naskar, Avik Chakraborty, Pradipta Maji, and Sukanta Das. Analysis of reachability tree for identification of cyclic and acyclic CA states. In ACRI, pages 63-72, 2012.

[4] Nazma Naskar, Sukanta Das, and Biplab K. Sikdar. Characterization of nonlinear cellular automata having only single length cycle attractors. J.Cellular Automata, 7(5-6):431-453, 2012.

[5] Stephen Wolfram. Cellular Automata and Complexity, chapter 4, pages 159-202. Westview Press, 2002.


Thank You