A Study on Reversible Rules of Probabilistic Cellular Automata

Anupam Pattanayak
Department of Computer Science and Engineering
Indian Institute of Information Technology Guwahati
Guwahati, India
anupam.pk@gmail.com

Subhasish Dhal
Department of Computer Science and Engineering
Indian Institute of Information Technology Guwahati
Guwahati, India
subhasis.rahul@gmail.com

***Abstract*—Reversibility is an important phenomena in nature as well as in Computer Science. Obtaining plaintext back from ciphertext can be modeled as one kind of reversibility. Image restoration problem can be modeled as another kind of reversibility. Cellular automata (CA) are lattices and that are used as computation tools for modeling diverse complex dynamical systems. The CA evolve from one configuration to another over iterations using local transition rules. Number of cells that are allowed to undergo the local transition or update function in every time step varies from one kind of CA to another. In probabilistic CA (PCA), cells are selected randomly for update. Reversibility is one important issue in CA. Reversible CA are those CA which comes back to the initial state for any given inital state after some time steps. In this paper, we have studied the reversibility of a PCA where maximum two cells are selected randomly for possible updates in every time step. We have introduced a new tool, reachable state graph to understand the PCA reversibility dynamics and proposed a deterministic algorithm to find if a rule is reversible for PCA of arbitrary size.**

***Index Terms*—Cellular automata, Probabilistic cellular automata, Reversibility, Reachable state graph.**

I. Introduction

Cellular automata (CA) are computational tools whose dynamics, though defined through simple local rules, is capable of generating a rich set of global patterns and structures that emerge without being designed a priori [6]. CA are often applied to model real life systems involving a huge number of locally interacting elements such as particles in physics [22], molecules in chemistry [21], proteins in biology [3], and in image processing [4].

CA is a grid of cells. Each cell is in one of a finite set of possible states. A cell undergoes update function at every clock tick. Update function may change the cell state depending on states of the cell itself and it's neighboring cells. If all the cells of CA can change it's state at the same time then the CA is termed as synchronous CA. On the other hand, if only a set of cells can change state in a time step, then the CA is said to be asynchronous CA (ACA). In fully asynchronous CA, only one cell is chosen randomly for possible update in a time step. In $\alpha-$Asynchronous CA, all cells undergo a transition with a probability of α. In probabilistic CA (PCA), cells undergo the transition probabilistically. So in PCA, the cell update is done randomly [2]. We can view the fully asynchronous CA as a kind of PCA where at max one cell is chosen randomly for update. PCA are useful for modeling the systems where stochastic nature of the phenomenon exists in the system [5].

If each cell of a CA follows the same rule for transition, then the CA is referred as uniform CA. Otherwise, the CA is said to be non-uniform. In our present work, we have considered uniform PCA. If we consider two-way finite CA, then there is no left neighbor of leftmost cell and similarly there is no right neighbor of rightmost cell. If we consider these cells are 0, then that CA is referred as null boundary CA. If we consider that CA is circular, then the CA is referred as periodic CA.

Reversibility is an important feature found in nature and in Computer Science [3], [20]. Getting plaintext back from ciphertext can be modeled as one type of reversibility [18], [19]. Image restoration problem can be modeled as another type of reversibility [16], [17]. CA is reversible (or invertible), if its global map is reversible. In other words, given any particular CA configuration, if the CA evolves to this same configuration after one or more time steps, then the rule governing the CA is referred as reversible [13], [8], [14]. Reversibility of synchronous (and uniform) CA has been studied in [7]. Existing literature on reversibility of ACA only mentions the rules that can not be reversible [9]. There is no discussion on the reversible rules for ACA (and PCA) in the existing literature. This is due to the fact that PCA (and ACA) dynamics is difficult to characterize. The only existing results, that are available, is for fully asynchronous CA [11], [10]. This is the motivation behind our work.

Our Contribution: In this work, we have adventured a step further from fully asynchronous CA. We consider that any two cells are chosen randomly for possible update, and then we study the dynamics of this kind of PCA to understand it's reversibility. We have introduced a new tool, *reachable state graph*, to understand the PCA reversibility dynamics. We have proposed a deterministic algorithm to find if a rule is reversible for PCA of arbitrary size. We have also reported simulation results for PCA reversibility.

The rest of this paper has been organized as follows. We discuss the preliminaries and introduce the notations used in this paper in section II. Existing works on irreversibility of CA are discussed in the section III. Next, in section IV, we construct the all possible states that a PCA state can evolve to after a state transition. Thereafter, we introduce the reachable state graph, a tool used to analyze the dynamics of the PCA in section V. The algorithm to check if the rule is reversible is also described in this section. The simulation-based results on reversible rules for PCA is reported in section VI. Finally, we conclude the paper in section VII.

II. Preliminaries

CA of length n is often denoted by $n - CA$. Elementary CA (ECA, or sometimes just CA) is a 1-D CA where every cell is in a binary state and a cell has just two neighbors - left neighbor and right neighbor [1], [7]. So, in ECA the local cell update function is a mapping $f : \{0,1\}^3 \rightarrow \{0,1\}$. This function f is a rule that defines the local transformations:

$c \rightarrow f(l, c, r)$, where $l, c, r \in \{0, 1\}$, and l, c, r denote left neighbor, cell itself, and right neighbor respectively.

This (possible) change of cell is called a transition. We say that a transition is active if $f(l, c, r) = \bar{c}$ and transition is passive if $f(l, c, r) = c$. If we see the *truth table* of the left neighbor, the cell, and right neighbor, then the output column can be one of the total 256 possibilities. We refer each of these 256 combinations of CA output as a rule. The rules are numbered according to the decimal equivalent of 8-bit binary, in reverse, of CA output column of the truth table. Hereafter, we use the term CA to mean ECA with periodic boundary.

For CA, we can think of left neighbor cell, The local cell itself, and right neighbor cell values together as a *min term*. When this min term is combined with a CA rule − how does the cell change or not − we refer that as *rule min term* (RMT) [9]. This RMT is the key to CA dynamics insight. For any CA rule, some RMTs are active (the cell value flips) and remaining RMTs are passive (the cell value remains unchanged). For example, the passive and active RMTs of rule 20 is shown in Table I.

TABLE I
ACTIVE AND PASSIVE RMTS FOR RULE 20

Left neighbor cell l	Cell c	Right neighbor cell r	Update function $f(l,c,r)$	Remark
0	0	0	0	RMT 0: passive
0	0	1	0	RMT 1: passive
0	1	0	1	RMT 2: passive
0	1	1	0	RMT 3: active
1	0	0	1	RMT 4: active
1	0	1	0	RMT 5: passive
1	1	0	0	RMT 6: active
1	1	1	0	RMT 7: active

To characterize all possible state transitions in subsequent time steps, from any given state for a given rule, any CA state can be represented as a set of different RMTs. For this, we need to find out all the passive RMTs and active RMTs for rules 0 - 255. For example, we show the active and passive RMTs for CA rules 0-15 in the Table II.

TABLE II
ACTIVE AND PASSIVE RMTS FOR RULES 0-15

Rule	Active RMTs	Passive RMTs
0	2, 3, 6, 7	0, 1, 4, 5
1	0, 2, 3, 6, 7	1, 4, 5
2	1, 2, 3, 6, 7	0, 4, 5
3	0, 1, 2, 3, 6, 7	4, 5
4	3, 6, 7	0, 1, 2, 4, 5
5	0, 3, 6, 7	1, 2, 4, 5
6	1, 3, 6, 7	0, 2, 4, 5
7	0, 1, 3, 6, 7	2, 4, 5
8	2, 6,7	0, 1, 3, 4, 5
9	0, 2, 6, 7	1, 3, 4, 5
10	1, 2, 6,7	0, 3, 4, 5
11	0, 1, 2, 6,7	3, 4, 5
12	6, 7	0, 1, 2, 3, 4, 5
13	0, 6, 7	1, 2, 3, 4, 5
14	1, 6, 7	0, 2, 3, 4, 5
15	0, 1, 6, 7	2, 3, 4, 5

Next, we analyze any given CA state in terms of RMTs present in that state. For example, given a periodic CA of size four, any CA state can be characterized as a sequence of different RMTs (or cyclic shifts of this sequence), and these RMT sequence sets are listed in Table III.

TABLE III
4-CELL CA STATES AND ITS RMT SEQUENCE SET

CA configuration (left, cell, right, fourth cell)	RMT sequence set of CA state
(0,0,0,0) or it's cyclic shift	{0}
(0,1,0,0) or it's cyclic shift	{0, 1, 2, 4}
(0,1,1,0) or it's cyclic shift	{1, 3, 4, 6}
(0,1,0,1) or it's cyclic shift	{2, 5}
(0,1,1,1) or it's cyclic shift	{3, 5, 6, 7}
(1,1,1,1) or it's cyclic shift	{7}

The state transition behaviors of different CA states having same RMT sequence set are identical.

III. Existing works

Wolfram studied the reversible rules for uniform synchronous CA. Then researchers characterized the reversibility (and irreversibility) of nonlinear CA, ACA, and fully asynchronous CA.

A. Reversibility of Synchronous CA

Wolfram identified total 16 reversible rules for n-CA [7], shown in the Table IV.

TABLE IV
REVERSIBLE CA RULES

15, 45, 51, 75, 85, 89, 101, 105,
150, 154, 166 , 170, 180, 204, 210, 240.

B. Reversibility of non-uniform CA

Das and Sikdar have noted the reversibility of non-uniform CA with null boundary [12]. They reported total 62 rules as reversible, as given in the Table V.

TABLE V
REVERSIBLE RULES FOR NON-UNIFORM NULL BOUNDARY CA

15, 23, 27, 30, 39, 43, 45, 51, 53, 54, 57, 58, 60, 75, 77, 78, 83, 85, 86, 89, 90, 92, 99, 101, 102, 105, 106, 108, 113, 114, 120, 135, 141, 142, 147, 149, 150, 153, 154, 156, 163, 165, 166, 169, 170, 172, 177, 178, 180, 195, 197, 198, 201, 202, 204, 210, 212, 216, 225, 228, 232, 240

C. Irreversibility of Asynchronous CA

Sarkar, Mukherjee, and Das made the first attempt to study reversibility of ACA [9]. They reported the irreversible rules for periodic ACA and the irreversible rules for null-boundary ACA separately. These irreversible rules for periodic ACA and null-boundary ACA are listed in Table VI and Table VII respectively.

TABLE VI
IRREVERSIBLE RULES FOR PERIODIC BOUNDARY ACA

0, 2, 4, 6, 8, 10, 12, 14, 16, 20, 24, 28, 64, 66, 68, 70, 72, 74, 76, 78, 80, 84, 88, 92, 141, 143, 157, 159, 173, 175, 189, 191, 197, 199, 205, 207, 213, 215, 221, 223, 229, 231, 237, 239, 245, 247, 253, 255

TABLE VII
IRREVERSIBLE RULES FOR NULL BOUNDARY ACA

0, 4, 13, 15, 29, 31, 45, 47, 61, 63, 69, 71, 77, 79, 85, 87, 93, 95, 101, 103, 109, 111, 117, 119, 125, 127, 141, 143, 157, 159, 160, 168, 170, 173, 175, 189, 191, 197, 199, 205, 207, 213, 215, 221, 223, 224, 229, 231, 232, 234, 237, 239, 240, 245, 247, 248, 250, 253, 255

They concluded that, reversible ACA rules will not belong to this rule set. But they could not deterministically report the reversible rules for ACA.

D. Reversibility of Fully Asynchronous CA

Reversibility of fully asynchronous CA has been studied in [11] and [10]. These works pointed that, some rules exhibit reversibility sometimes but not always, and some rules are reversible always. These class of rules which are always reversible are classified as recurrent rules. Total 46 recurrent rules have been identified for fully asynchronous CA, and are listed in the Table VIII.

TABLE VIII
RECURRENT RULES FOR FULLY ASYNCHRONOUS CA

33, 35, 38, 41, 43, 46, 49, 51, 52, 54, 57, 59, 60, 62, 97, 99, 102, 105, 107, 108, 113, 115, 116, 118, 121, 123, 131, 134, 139, 142, 145, 147, 148, 150, 153, 155, 156, 158, 195, 198, 201, 204, 209, 211, 212, 214

E. Reversibility of $\alpha-$Asynchronous CA

Pattanayak and Dhal have reported simulation based results on reversibility of $\alpha-$Asynchronous CA in [15]. They have noted the reversible rules of $\alpha-$Asynchronous CA for different CA sizes and different update probabilities.

IV. MODELING PROBABILISTIC CA STATE TRANSITIONS

Any CA state can be expressed as a set of RMT sequences. Now, we study the effect of choosing maximum two cells randomly for transitions. Our goal is to identify RMT sequence set of the resultant CA. For this, we first determine the current state's RMT sequence set. Then depending on whether a RMT is passive or active for a rule, the transition to another state will be decided. For example, we take a PCA of length four, and select any two cells randomly for possible update. The initial PCA state is any of the six possible RMT sequence set, as mentioned in Table III. For each of these six cases, we note the resultant PCA state depending on the passive or active RMTs of each member in RMT sequence set in the next.

A. RMT sequence $\{0\}$

First, take the CA state with RMT sequence $\{0\}$. For any CA rule, RMT 0 will be either passive or active. If RMT 0 is passive (P), then the RMT sequence of resultant CA will be $\{0\}$ only. If RMT 0 is active (A), then RMT sequences of the resultant CA will be either RMTs $\{1,3,4,6\}$ or RMTs $\{2,5\}$ only. This is shown in the Table IX.

TABLE IX
POSSIBLE RMT SEQUENCES OF RESULTANT CA STATE WHEN INPUT RMT SEQUENCE IS $\{0\}$

RMT 0	RMT sequence of Resultant CA state
P	$\{0\}$
A	$\{2, 5\}$, $\{1, 3, 4, 6\}$

B. RMT $\{7\}$

Now, consider the PCA state with RMT sequence $\{7\}$. For any CA rule, RMT 7 will be either passive or active. If RMT 7 is passive, then the RMT sequence of resultant CA will be $\{7\}$ only. If RMT 7 is active, then RMT sequences of the resultant CA will be either RMTs $\{1,3,4,6\}$ or RMTs $\{2,5\}$, as shown in the Table X.

TABLE X
RESULTANT CA STATE FOR INPUT RMT SEQUENCE $\{7\}$

RMT 7	RMT sequence of resultant CA state
P	$\{7\}$
A	$\{2, 5\}$, $\{1, 3, 4, 6\}$

C. RMT sequence $\{2, 5\}$

Next, take the PCA state with RMT sequence $\{2, 5\}$. For any CA rule, RMT 2 and RMT 5 will be either P P, or P A, or A P, or A A. RMT sequences of the resultant CA after update of any two cells is shown in the Table XI.

D. RMT sequence $\{0, 1, 2, 4\}$

If we take the RMT sequence $\{0,1,2,4\}$, then for any CA rule, RMT 0, RMT 1, RMT 2, and RMT 4 will be either passive or active. So there will be total sixteen possibilities. Careful inspections of all possible resultant CA state's RMT

TABLE XI
RESULTANT CA STATE FOR INPUT RMT SEQUENCE $\{2, 5\}$

RMT 2	RMT 5	RMT sequence of Resultant CA state
P	P	{2, 5}
P	A	{2, 5}, {3, 5, 6, 7}, {7}
A	P	{0}, {2, 5}, {0, 1, 2, 4}
A	A	{0}, {1, 3, 4, 6}, {7}

sequences after state transition corresponding to these sixteen possibilities will be as per the Table XII.

TABLE XII
CA STATE FOR INPUT RMT SEQUENCE $\{0, 1, 2, 4\}$

RMT 0	RMT 1	RMT 2	RMT 4	RMT sequence of Resultant CA state
P	P	P	P	{0, 1, 2, 4}
P	P	P	A	{0, 1, 2, 4}, {1, 3, 4, 6}
P	P	A	P	{0}, {0, 1, 2, 4}
P	P	A	A	{0}, {0, 1, 2, 4}, {1, 3, 4, 6}
P	A	P	P	{0, 1, 2, 4}, {1, 3, 4, 6}
P	A	P	A	{0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
P	A	A	P	{0}, {0, 1, 2, 4}, {1, 3, 4, 6}
P	A	A	A	{0}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
A	P	P	P	{2, 5}, {0, 1, 2, 4}
A	P	P	A	{2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6},{3, 5, 6, 7}
A	P	A	P	{0}, {2, 5}, {0, 1, 2, 4}
A	P	A	A	{0}, {2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
A	A	P	P	{2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6},{3, 5, 6, 7}
A	A	P	A	{2, 5}, {1, 3, 4, 6}, {3, 5, 6, 7}
A	A	A	P	{0}, {2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
A	A	A	A	{0, 1, 2, 4}, {3, 5, 6, 7}

E. RMT sequence $\{1, 3, 4, 6\}$

Similarly, when PCA state is of RMT sequence $\{1, 3, 4, 6\}$, then the resultant CA state's RMT sequences after update of any two cells will be as per the Table XIII.

F. RMT sequence $\{3, 5, 6, 7\}$

In the same way, when the PCA state has the RMT sequence $\{3, 5, 6, 7\}$, then the resultant CA state's RMT sequences after update of any two cells will be as per the following Table XIV.

TABLE XIII
CA STATE FOR INPUT RMT SEQUENCE $\{1, 3, 4, 6\}$

RMT 1	RMT 3	RMT 4	RMT 6	RMT sequence of Resultant CA state
P	P	P	P	{1, 3, 4, 6}
P	P	P	A	{0, 1, 2, 4}, {1, 3, 4, 6}
P	P	A	P	{1, 3, 4, 6}, {3, 5, 6, 7}
P	P	A	A	{2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6},{3, 5, 6, 7}
P	A	P	P	{0, 1, 2, 4}, {1, 3, 4, 6}
P	A	P	A	{0}, {0, 1, 2, 4}, {1, 3, 4, 6}
P	A	A	P	{0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
P	A	A	A	{0}, {2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
A	P	P	P	{1, 3, 4, 6}, {3, 5, 6, 7}
A	P	P	A	{0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
A	P	A	P	{1, 3, 4, 6}, {3, 5, 6, 7}, {7}
A	P	A	A	{2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6},{3, 5, 6, 7}, {7}
A	A	P	P	{2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6},{3, 5, 6, 7}
A	A	P	A	{0}, {2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
A	A	A	P	{2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}, {7}
A	A	A	A	{0}, {2, 5}, {1, 3, 4, 6}, {7}

TABLE XIV
CA STATE FOR INPUT RMT SEQUENCE $\{3, 5, 6, 7\}$

RMT 3	RMT 5	RMT 6	RMT 7	RMT sequence of Resultant CA state
P	P	P	P	{3, 5, 6, 7}
P	P	P	A	{2, 5}, {3, 5, 6, 7}
P	P	A	P	{1, 3, 4, 6}, {3, 5, 6, 7}
P	P	A	A	{2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
P	A	P	P	{3, 5, 6, 7}, {7}
P	A	P	A	{2, 5}, {3, 5, 6, 7}, {7}
P	A	A	P	{0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
P	A	A	A	{0}, {2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
A	P	P	P	{1, 3, 4, 6}, {3, 5, 6, 7}
A	P	P	A	{0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
A	P	A	P	{1, 3, 4, 6}, {3, 5, 6, 7}, {7}
A	P	A	A	{2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}, {7}
A	A	P	P	{2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
A	A	P	A	{0}, {2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}
A	A	A	P	{2, 5}, {0, 1, 2, 4}, {1, 3, 4, 6}, {3, 5, 6, 7}, {7}
A	A	A	A	{0}, {2, 5}, {1, 3, 4, 6}, {7}

V. IDENTIFYING REVERSIBLE PCA RULES

Once we obtain the possible RMT sequences of the resultant PCA state for all the input RMT sequences, we are ready to study the PCA state transitions graphically.

A. Reachable State Graph

Now, in attempt to understand the dynamics of transitions of PCA states under different rules, we propose a new tool, *reachable state graph*. We build reachable state graph for every rule for a particular CA length. In this graph, the vertex set is the set of possible RMT sequences, and edge set is the directed edges from one vertex to another. The directed edges from a vertex will go to different vertices or to itself depending on the random selection of two cells for local transitions. Number of out-bound edges from a vertex is the number of active RMTs in the RMT sequence set vertex. For example, if we consider a vertex corresponding to RMT sequence set $\{1, 3, 4, 6\}$ where RMT 1, RMT 4 are passive, and RMT 3, RMT 6 are active then degree of out-bound edges for this vertex is two. If all the RMTs of a vertex are passive, then degree of out-bound edges for this vertex is zero. Now, if the degree of in-bound edges of this vertex is more than zero, and degree of out-bound edges is zero, then this vertex is a *sink vertex*. That is, if the PCA reaches to this state, it will never go out of this state. In other words, presence of sink vertex in a reachable state graph makes sure that the rule is not reversible. Similarly if in-degree of a vertex is zero and out-degree of it is more than zero, then we refer it as a *transient vertex*. That is if PCA makes a transition from a transient state then it will never be able to reach this state again. So presence of transient vertex in the graph also cancels out a rule to be reversible. Also we may get oscillating vertices of two or more vertices, where if PCA reaches in any constituent vertex, the PCA will oscillate forever among these *oscillating vertices*. If the reachable state graph has neither sink vertex, nor transient vertex, nor oscillating vertices, then the CA is bound to come back to the initial state. However, the time steps required can be sufficiently large. That is, *absence of sink vertices, transient vertices, and oscillating vertices in a reachable state graph confirms a rule to be reversible*.

For example, the reachable state graph of 4-PCA for rule 19 is shown in the Fig. 1.

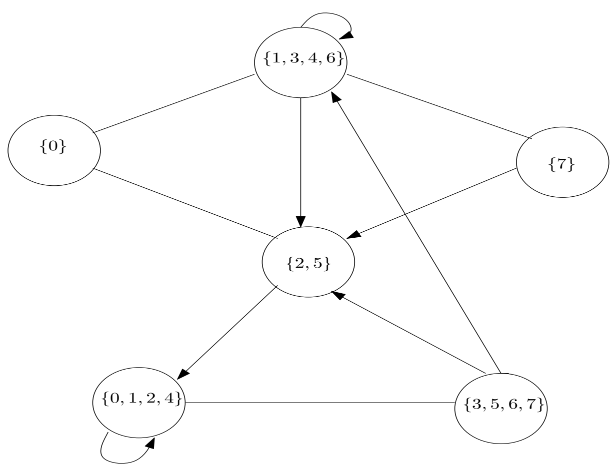

Fig. 1. Reachable state graph of 4-PCA for rule 19

Here, six vertices represent the RMT sequence sets $\{1, 3, 4, 6\}$, $\{0\}$, $\{7\}$, $\{2, 5\}$, $\{0, 1, 2, 4\}$, and $\{3, 5, 6, 7\}$. Here, both unidirectional edges and bidirectional edges are present. The edges without arrow represent bidirectional edges. This reachable state graph is connected. We see that, there is neither a sink vertex nor a transient vertex. That is, if we start from any vertex, then we are bound to comeback to that vertex. However, the number of time steps required to comeback to the initial state can be sufficiently large and may not be predetermined due to the randomness involved in the transitions. Therefore, rule 19 is reversible for 4-PCA.

As another example, the reachable state graph of 4-PCA for rule 47 is shown in the Fig. 2.

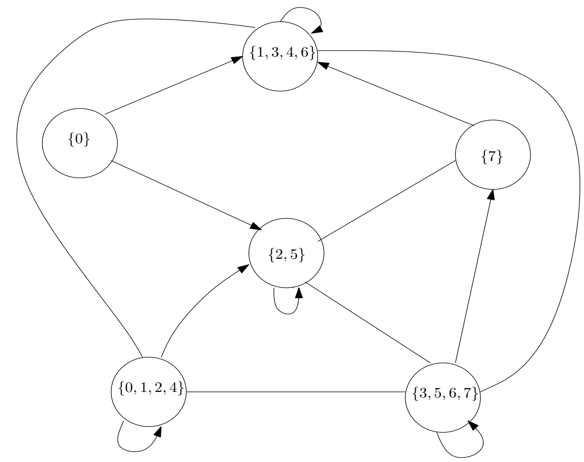

Fig. 2. Reachable state graph of 4-PCA for rule 47

Here, it can be seen that, there is no sink vertex, but the vertex $\{0\}$ is a transient vertex. That is, if the initial state is $\{0\}$, then the PCA will never reach back to this state, as there is no incoming edge to this vertex. So, rule 47 is not reversible for 4-PCA.

B. Algorithm to Find Reversible Rule

Having constructed the reachable state graph for a rule, we are in a position to deterministically find out if a given PCA is reversible or not. Now, we propose the deterministic algorithm to check if a given rule, r is reversible for n-PCA of an arbitrary size, and given below in Algorithm 1.

Algorithm 1 Algorithm FindReversiblePCA

Input: PCA size n, Rule number r.
Output: True, if r is reversible. False, otherwise.

1) Construct the valid RMT sequence sets S for n-PCA.
2) For each set $s_i \in S$, construct destination RMT sequences set table.
3) Construct directed reachable state graph G for r.
4) **If** a sink vertex is present in G **Then**
   Return False
   Else If a transient vertex is present in G **Then**
   Return False
   Else If oscillating vertices are found in G **Then**
   Return False
   Else
   Return True
   End If

VI. Simulation-based Results

Here, we report our observations based on our simulations. We have simulated the PCA using C programming language. The simulation is done using gcc compiler in Ubuntu platform. We have done simulations for PCA of sizes 4, 5, 6, 7, 8, 9, and 10. We list out all the reversible rules for PCA of lengths ranging from 4 to 10 that we have obtained through simulation in Table XV.

ISBN: 978-1-7281-4283-8 PART NO.: CFP20O01-ART

TABLE XV
REVERSIBLE RULES FOR PCAS OF DIFFERENT SIZES

PCA size	Reversible Rules	Remarks
4	19, 27, 35, 37, 39, 43, 46, 49, 51, 53, 55, 57, 59, 60, 62, 83, 91, 99, 102, 113, 115, 118, 119, 131, 139, 142, 145, 153, 195, 204, 209, 212	Total 32 rules
5	19, 23, 27, 30, 35, 37, 39, 41, 43, 46, 49, 51, 53, 55, 57, 59, 60, 62, 83, 86, 91, 97, 99, 102, 105, 107, 113, 115, 116, 118, 121, 131, 135, 139, 142, 145, 149, 153, 195, 204, 209, 212.	Total 42 rules
6	19, 27, 33, 35, 39, 41, 43, 46, 49, 51, 53, 55, 57, 59, 60, 62, 83, 97, 99, 102, 105, 107, 113, 115, 116, 118, 121, 123, 131, 139, 145, 153, 195, 204, 209.	Total 35 rules
7	19, 23, 27, 30, 33, 35, 37, 39, 41, 43, 46, 49, 51, 53, 55, 57, 59, 60, 62, 83, 91, 97, 99, 102, 105, 107, 113, 115, 116, 118, 121, 123, 131, 139, 145, 149, 153, 195, 204, 209.	Total 40 rules
8	19, 27, 33, 35, 37, 39, 43, 46, 49, 51, 53, 55, 57, 59, 60, 62, 83, 91, 97, 99, 102, 107, 113, 115, 116, 118, 123, 131, 139, 145, 153, 195, 204, 209.	Total 34 rules
9	19, 23, 33, 35, 39, 41, 43, 46, 49, 51, 55, 57, 59, 62, 83, 97, 99, 102, 107, 113, 115, 118, 121, 123, 131, 145, 195, 204, 209	Total 29 rules
10	19, 27, 33, 35, 37, 39, 41, 43, 46, 49, 51, 53, 55, 57, 59, 60, 62, 83, 91, 97, 99, 102, 105, 107, 113, 115, 116, 118, 121, 123, 131, 139, 145, 153, 195, 204, 209.	Total 37 rules

This result confirms with our findings through structural analysis of PCA by reachable state graph. We see here that, the number of rules that are reversible varies with PCA size. For example, for 4-PCA, total thirty two rules are reversible, whereas it is forty two for 5-PCA. We see that, following twenty two rules are always reversible for all the reported PCA sizes: rule 19, 35, 39, 43, 46, 49, 51, 55, 57, 59, 62, 83, 99, 102, 113, 115, 118, 131, 145, 195, 204, and 209.

VII. Conclusion

In this work, we study the reversibility of PCA when maximum two cells are chosen randomly for update. We demonstrate how the reversible rules can be identified deterministically with the help of our proposed reachable state graph. The algorithm for finding whether a rule is reversible for a given PCA has been proposed. We have noted down the reversible rules as per our simulation result. This result matches with the PCA structural analysis through reachable state graph.

Acknowledgement

The authors thank Cellular Automata Research Group of IT Department, IIESTS for introducing this research topic.

References

[1] P. P. Chaudhuri, D. R. Chowdhury, S. Nandi, and S. Chattopadhyay, "CA and Its Applications: A Brief Survey," in *Additive cellular automata: theory and applications*, vol. 1, John Wiley & Sons, NJ, 1997, pp. 9-16.

[2] R. Fernández, P. Louis, and F. Nardi, "Overview: PCA Models and Issues," in *Probabilistic Cellular Automata, Emergence, Complexity and Computation*, Springer, Cham, Switzerland, 2018, pp. 1-30.

[3] J. L. Schiff, "Reversibility," in *Cellular Automata: a Discrete View of the World*, vol. 45, John Wiley & Sons, NJ, 2011, pp. 60-69.

[4] P. Rosin, A. Adamatzky, and X. Sun, "Edge Detection Using Cellular Automata," in *Cellular Automata in Image Processing and Geometry*, Springer, Cham, Switzerland, 2014, pp. 85-101.

[5] A. Agapie, A. Andreica, and M. Giuclea, "Probabilistic Cellular Automata," *Journal of Computational Biology*, vol. 21, no. 9, pp. 699-708, 2014.

[6] P. Sarkar, "A Brief History of Cellular Automata," ACM *Computing Surveys*, vol. 32, no. 1, pp. 80-107, 2000.

[7] S. Wolfram, "Cellular Automata: Collected Papers," Addison-Wesley, NY, 1994, pp. 1-30.

[8] J. Kari, "Reversible Cellular Automata: from Fundamental Classical Results to Recent Developments," *New Generation Computing*, vol. 36, no. 3, pp. 145-172, 2018.

[9] A. Sarkar, A. Mukherjee, and S. Das, "Reversibility in Asynchronous Cellular Automata," *Complex Systems*, vol. 21, no. 1, pp. 71-84, 2012.

[10] B. Sethi, N. Fatès, and S. Das, "Reversibility of Elementary Cellular Automata Under Fully Fsynchronous Update," *Proc. of* 11^{th} *Int. Conf. on Theory and Applications of Models of Computation*, pp. 39-49, 11-13 April 2014, Chennai, India.

[11] N. Fatès, B. Sethi, and S. Das, "On the Reversibility of ECAs with Fully Asynchronous Updating: the Recurrence Point of View," in *Reversibility and Universality*, Springer, Cham, Switzerland, 2018, pp. 313-332,

[12] S. Das, and B. K. Sikdar, "Analysis and Synthesis of Nonlinear Reversible Cellular Automata in Linear Time," arXiv preprint arXiv:1311.6879, 2013.

[13] K. Imai, B. Martin, and R. Saito, "On radius 1 Nontrivial Reversible and Number-conserving Cellular Automata," in *Reversibility and Universality*, Springer, Cham, Switzerland, 2018, pp. 269-277.

[14] K. Morita, "One-Dimensional Universal Reversible Cellular Automata," in *Theory of Reversible Computing*, Springer, Cham, Sitzerland, 2017, pp. 299-329.

[15] A. Pattanayak, and S. Dhal, "Reversibility of α-Asynchronous Cellular Automata: A Simulation-Based Study," *Proc. of Conf. on Advancement in Computation, Communication and Electronics Paradigm*, pp. 32-36, 18 January 2019, Belur Math, India, 2019.

[16] D. Nayak, R. Dash, and B. Majhi, "Salt and Pepper Noise Reduction Schemes using Cellular Automata," *Proc. of 3rd Int. Conf. on Advanced Computing, Networking and Informatics*, pp. 427-435, 22 - 24 September 2016, Rourkela, India.

[17] F. Qadir, and I. Q. Shoosha, "Cellular Automata-based Efficient Method for the Removal of High-density Impulsive Noise from Digital Images," *Int. Journal of Information Technology*, vol. 10, no. 4, pp. 529-536, 2018.

[18] B. Sethi, and S. Das, "On the Use of Asynchronous Cellular Automata in Symmetric-key Cryptography," *Proc. of* 4^{th} *Int. Symp. on Security in Computing and Communication*, pp. 30-41, 21-24 September, Jaipur, India, 2016.

[19] S. Maiti, and D. R. Chowdhury, "Achieving Better Security Using Nonlinear Cellular Automata as a Cryptographic Primitive," *Proc. of* 4^{th} *Int. Conf. on Mathematics and Computing*, pp. 3-15, 9-11 January 2018, Varanasi, India.

[20] R. Wille, A. Lye, and P. Niemann, "Checking Reversibility of Boolean Functions," *Proc. of* 8^{th} *Int. Conf. on Reversible Computation*, pp. 322-337, 7-8 July 2016, Bologna, Italy.

[21] A. Ilachinski, "Cellular Automata: a Discrete Universe," pp. 463-505, World Scientific, Singapore, 2001.

[22] A. Adamatzky, and G. J. Martínez, "Asynchronous Cellular Automata Simulating Complex Phenomena," in *Designing Beauty: The Art of Cellular Automata*, vol. 20, Springer, Cham, Switzerland, 2016, pp. 111-119.