Exploring Periodic Boundary Cellular Automaton Rules for One and Two Fixed Points

Baisakhi Das
Institute of Engineering and Management
University of Engineering and Management
Kolkata, West Bengal, India

Mamata Dalui
Mousumi Saha
National Institute of Technology
Durgapur, West Bengal, India

Kasturi Ghosh
University Institute of Technology
Burdwan, West Bengal, India

Nilanjana Das
Biplab K. Sikdar
Indian Institute of Engineering Science and Technology
Shibpur, West Bengal, India

This paper characterizes cellular automaton (CA) rules with a periodic boundary. We find the rules that have the potential to form single length cycle attractors (fixed points) in the CA state space. The characterization enables synthesizing periodic boundary cellular automata (PBCAs), for arbitrary length, that only have fixed points. A tool referred to as the NSRT diagram (NSRTD) has been developed for PBCAs to provide the basis for such characterization and synthesis. The NSRTD identifies rules R_S / R_T that form a uniform single length cycle single-attractor CA (SACA) (with one fixed point) or a single length cycle two-attractor CA (TACA) (with two fixed points) for any length. The NSRTD is also capable of determining the presence of multi-length attractor basins in a CA state space. The framework is further employed to develop the schemes for exploring rules R_b that enable splitting and merging fixed-point attractor basins (required for devising different CA-based solutions) in a uniform SACA or TACA when hybridized by R_b.

Keywords: periodic boundary cellular automata; attractor; fixed point; NSRT diagram; SACA; TACA

1. Introduction

The invention of the cellular automaton (CA) added a new era to the history of computing [1, 2] and aided in modeling parallel processing

https://doi.org/10.25088/ComplexSystems.33.1.31

[2]. The 2-state 3-neighborhood CA [2] proved to be a powerful modeling tool [3–5]. While characterizing the 3-neighborhood CA state space, a set of CA states, referred to as attractors, is identified that act as the sink. The neighboring states of an attractor approach asymptotically during evolution [6]. But an attractor with only one state, called a single length cycle attractor (fixed point) [3], is of special interest to researchers. The most important application domains of such a CA with fixed-point attractors is in authentication and cryptography [3, 7–9] and computer architecture [10]. Characterizing the 3-neighborhood CA state space will enable the synthesis of this special class of cellular automata (CAs). The characterization relies on exploring the properties of individual CA rules [2, 11] and their potential to form the fixed points.

The synthesis of fixed-point attractor CAs in linear or additive domains has been proposed in the literature [7, 9, 12–15]. Identifying attractors in the null boundary, especially for linear or additive CAs, is attempted in [3, 16, 17]. Graph-based solutions are also proposed [18, 19]. The solution reported in [20] targets the synthesis of invertible periodic boundary CAs (PBCAs). Further, the reachability tree (RT) tool is defined in [21] to explore null boundary CAs with fixed-point attractors. The worst-case time complexity, to decide on and synthesize such a CA, is found to be exponential, as the number of edges in the RT grows exponentially. However, characterizing fixed-point attractor PBCAs and their synthesis with the specified fixed points is yet to be formalized in the nonlinear domain [22].

The present scenario motivates us to develop a common theoretical framework for characterizing CA rules that have the potential to form fixed points in a CA state space. In [16] we defined the NSRT diagram (NSRTD), a tool for identifying rules that form fixed points for null boundary CAs. In this paper, the NSRTD is modeled to explore the multi-length cycle attractors as well as the fixed points in PBCAs. The NSRTD identifies the CAs that have a fixed point, that is, a single length cycle attractor CA (SLCA). The single length cycle single-attractor CA (SACA) is a CA with one and only one fixed point in its state space. The single length cycle two-attractor CA (TACA) is a CA with only two fixed points in its state space. This, in effect, identifies the rules that form uniform single length cycle single-attractor CAs (SACAs) or single length cycle two-attractor CAs (TACAs) of any length. The NSRTD is further employed to identify the rules R_h that enable splitting or merging attractor basins of the uniform SACAs or TACAs when hybridized by R_h.

The following section notes the basics of 2-state 3-neighborhood CAs. The NSRTD is introduced in Section 3. The fundamental results of SACA and TACA characterization are reported in Section 4. In

Section 5, we report on the hybridization to split or merge the attractor basins. The paper concludes in Section 6.

2. Cellular Automata

In this paper, the CA is thought of as an autonomous finite-state machine (FSM) (Figure 1). Each cell of such a CA can store either 0 or 1. The value stored at time t is its present state (PS) (configuration at t). Its next state (NS) at time $t+1$ (configuration at $t+1$) is defined by the PS of its immediate two neighbors and itself (3-neighborhood). That is, the NS S_i^{t+1} of cell i is

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

where S_{i-1}^t, S_i^t and S_{i+1}^t are the PS of cells $i-1$, i and $i+1$ at time t, respectively. The f_i defines the NS function. The collection of states $\mathcal{S}^t = (S_1^t, S_2^t, \ldots, S_n^t)$ of the n cells at time t denotes the CA state at t. As a cell can store either 0 or 1 only, it is a 2-state CA. When the truth table corresponding to f_i is constructed in such a 3-neighborhood CA, the eight outputs corresponding to the eight combinations of the PS of cells $i-1$, i and $i+1$ denote the CA rule R_i [11] of cell i. It is the decimal equivalent of the eight outputs.

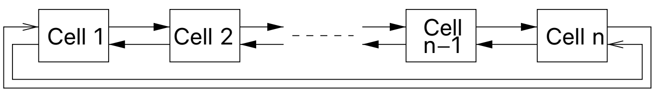


Figure 1. The n-cell periodic boundary CA.

There can be a total of 256 rules in 2-state 3-neighborhood CAs, called elementary rules. Three rules, 184, 226 and 232, are defined in Table 1. The first row of Table 1 denotes the eight combinations of the PS of the three neighbors (left, itself and right) of cell i at time t [10]. The remaining three rows define the NS of cell i at time $t+1$ for the rules shown in the last column.

PS	111	110	101	100	011	010	001	000	Rule
RMT	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
(i) NS	1	0	1	1	1	0	0	0	184
(ii) NS	1	1	1	0	0	0	1	0	226
(iii) NS	1	1	1	0	1	0	0	0	232

Table 1. The CA rules.

This paper deals with the periodic boundary CA (PBCA), where the left (resp. right) neighbor of the leftmost (resp. rightmost) CA cell is the rightmost (resp. leftmost) cell (Figure 1). For the rest of the paper, the PBCA is referred to as the CA.

The set of rules $\langle \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_i, \ldots, \mathcal{R}_n \rangle$ configures the cells of an n-cell CA. For a uniform CA, $\mathcal{R}_1 = \mathcal{R}_2 = \cdots = \mathcal{R}_n$. Otherwise, the CA is nonuniform or hybrid. The CA ⟨184, 184, 184⟩ defines a three-cell uniform CA. On the other hand, CA ⟨232, 184, 184⟩ is nonuniform.

In the CA state space, some states may lie in a cycle (the $0 \to 0$, $10 \to 5 \to 10$ and $15 \to 15$ of Figure 2(a), the four-cell/bit CA states are converted into decimal). Such a cycle is referred to as an *attractor* of the CA. A self-loop (single length cycle) attractor ($0 \to 0$ and $15 \to 15$) is referred to as the fixed point. In rest of the paper, this is referred to as the fixed-point attractor. The attractor $10 \to 5 \to 10$ of Figure 2(a), involving multiple states—the 10 and 5, is called a multi-length cycle attractor. An attractor of a CA forms a basin. The members of a basin are all the states that either form the attractor or asymptotically approach the attractor. In Figure 2(a), the attractor $10 \to 5 \to 10$-basin has four states that include the attractor states 5 and 10.

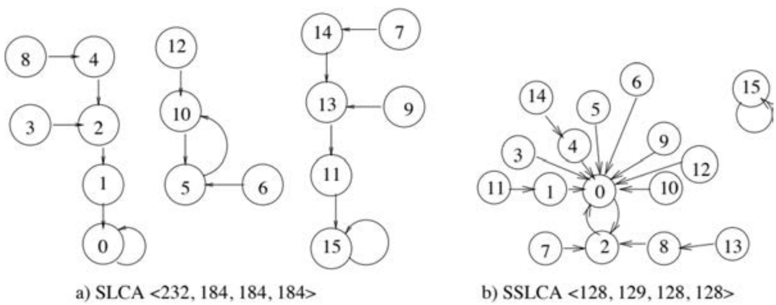


Figure 2. The state space of single length cycle attractor CA.

A combination of S_{i-1}^t, S_i^t, S_{i+1}^t (the PS of cells $i-1$, i and $i+1$), as shown in the first row of Table 1 can be referred to as the rule minterm (RMT) [23]. For example, the column 011 is the RMT 3. The NS for this RMT is 0 in rule 226 and it is 1 in rules 184 or 232. In general, an RMT is denoted as T(m), where $m = 0, 1, 2, 3, 4, 5, 6, 7$, and $\mathrm{T} = \{\mathrm{T}(0), \mathrm{T}(1), \ldots, \mathrm{T}(7)\}$. We represent an RMT T($m$) only by its decimal number m in the diagrams of this paper; that is, T(5) is denoted as 5.

The rule minterms (RMTs) of a rule can be divided into two subsets: $\mathrm{T}/0$ and $\mathrm{T}/1$. The RMT with NS 0 belongs to $\mathrm{T}/0$, whereas an RMT with NS 1 is in $\mathrm{T}/1$. Hence, $\mathrm{T}/0 \bigcap \mathrm{T}/1 = \emptyset$ and $\mathrm{T}/0 \bigcup \mathrm{T}/1 = \mathrm{T}$ for a rule.

Some RMTs in a CA rule can be *passive*. These cannot change the PS of a cell in the next time step. That is, such an RMT $x0y$ (resp. $x1y$) belongs to T/0 (resp. T/1). Whereas, if an RMT $x0y$ (resp. $x1y$) belongs to T/1 (resp. T/0), then it is *active*. An active RMT changes the PS of a cell in the next time step. For example, the RMT 2 (010) belongs to T/0 and RMT 5 (101) belongs to T/1 in the rule 184 (Table 1); that is, these two RMTs in rule 184 are active. On the other hand, RMT 0 (000) belongs to T/0 and RMT 3 (011) belongs to T/1 in rule 184, so these two RMTs are passive.

A CA having at least one fixed-point attractor and with or without the multi-length cycle attractor in its state space is referred to as an SLCA. The four-cell CA ⟨232, 184, 184, 184⟩ of Figure 2(a) has two fixed-point attractors (state 0 and 15) and one multi-length cycle attractor $10 \to 5 \to 10$ and it is an SLCA. On the other hand, the SLCA of Figure 2(b) has only one fixed-point attractor (state 15) and one multi-length cycle attractor $0 \to 2 \to 0$.

Definition 1. (SACA and SACA rule) An SLCA that produces a connected graph with only one of its fixed-point attractors is an SACA. A rule that can configure the n-cell uniform SACA, for all n, is an SACA rule.

The four-cell uniform CA configured with rule 64 of Figure 3 has one fixed-point attractor. It does not have any multi-length cycles in its state space. The CA is an SACA. Rule 64 can be found as an SACA rule.

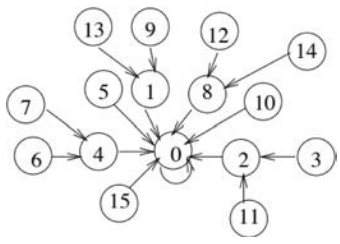


Figure 3. State space of SACA ⟨64, 64, 64, 64⟩.

Definition 2. (TACA and TACA rule) If an SLCA forms exactly two connected components, each with one fixed-point attractor, then it is a TACA. A rule that can configure the n-cell uniform TACA, for all n, is a TACA rule.

The uniform CA, configured with rule 128 of Figure 4, forms exactly two connected components in its state space, each having exactly one fixed point. Rule 128 can be found as a TACA rule.

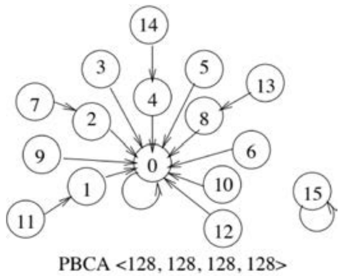


Figure 4. State space of TACA ⟨128, 128, 128, 128⟩.

3. NSRT Diagram

This section introduces the NSRTD for periodic boundary CAs. It is in line with the tool defined in [16] for null boundary CAs. It is formulated [24] for characterizing CA rules to determine fixed-point and multi-length cycle attractors in a CA state space and also to determine the cycle length of multi-length cycle attractors. The four components NCR, NSR, NSRS, NSRS-G are important for an NSRTD and are required for exploring SACAs and TACAs with a periodic boundary. The terminology is similar to that used in [16].

- *NCR*. This defines the RMTs on which a CA cell can change its state at time t if its left neighbor changes state on RMT r. If cell i in a CA changes its state at time t based on the RMT T_i^t of rule R_i, then it is definite that the cell $i+1$ can change its state (at t) based either on the RMT (T_{i+1}^t) $(2 \times T_i^t) \bmod 8$ or the RMT $(2 \times T_i^t + 1) \bmod 8$ of R_{i+1}. These two RMTs are referred to as the next cell RMTs (NCRs) of T_i^t.

Example 1. If cell i changes its state based on the RMT $T_i^t = T(1)$ of R_i, then for the change of state, cell $i+1$ can follow $T_{i+1}^t = T(2)$ or T(3) of R_{i+1}. Here, T(2) and T(3) are the NCRs of T(1). The NCR of an RMT is independent of rule R_i. Table 2 contains the NCRs of all eight RMTs.

RMT	NCR
T(0)/T(4)	T(0), T(1)
T(1)/T(5)	T(2), T(3)
T(2)/T(6)	T(4), T(5)
T(3)/T(7)	T(6), T(7)

Table 2. The NCRs of eight RMTs.

- *NSR and NSRS*. These two define how a cell changes its state during evolution of the CA. Let T_i^t (resp. T_i^{t+1}) be the RMT of R_i on which cell

i changes its state at time t (resp. at $t+1$), then T_i^{t+1} is referred to as the next state RMT (NSR) of T_i^t. During evolution, cell i changes its state following the sequence of RMTs T_i^0, T_i^1, …, T_i^t, T_i^{t+1}, …. This is referred to as the next state RMT sequence (NSRS) and for cell i it is denoted as NSRS_i.

Example 2. The computation of the NSRs of a uniform CA cell i is shown in Figure 5. The CA is configured with rule 116. Let cell i change its state at time t on RMT xyz; that is, the PSs of cell $i-1$, cell i and cell $i+1$ are x, y and z, respectively. Further, assume that the l (r) is the PS of cell($i-2$) (cell($i+2$)). That is, cell $i-1$ (resp. cell $i+1$) changes its state on RMT lxy (resp. RMT yzr). Now, if the NS of cells $i-1$, i and $i+1$, respectively, are d_l, d and d_r, then the cell i can change its state at time $t+1$ on the RMT d_l, d, d_r; that is, the NSR of cell i is d_l, d, d_r.

When $xyz = 000$, the $d_l = T(l00)$, where l is either 0 or 1. That is, $d_l = T(000)$ or $T(100) = 0$ or 1, as cell $i-1$ is configured with rule 116; and the T(000) is 0 and T(100) is 1 in rule 116 (Figure 5(a)). Similarly, $d_r = T(00r)$, for $xyz = 000$, and as r can be either 0 or 1, $d_r = T(000)$ or $T(001)$. That is, $d_r = 0$, as cell $i+1$ is configured with rule 116; and the T(000) is 0 and T(001) is also 0 in rule 116 (Figure 5(a)). That is, the NSR T_i^{t+1} for RMT 000 (T_i^t) is 0 or 4 (Figure 5(b)).

- *NSRS-G*. The NSRS graph (NSRS-G) explores all possible sequences of states (NSRSs) that can be generated by a CA cell during its evolution. The i^{th}-cell rule R_i of a CA is portrayed by a directed graph G(V, E). The V consists of the states' T_i^t ($t = 0, 1, 2, \ldots$) of cell i and $e_i(T_i^t, T_i^{t+1}) \in E$, where T_i^{t+1} is the NSR of T_i^t. Figure 6(a) is the NSRS-G of rule 116.

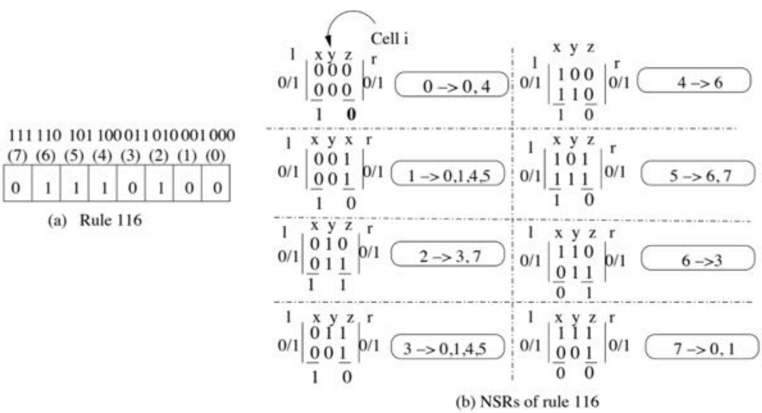


Figure 5. NSRs of a uniform CA cell configured with rule 116.

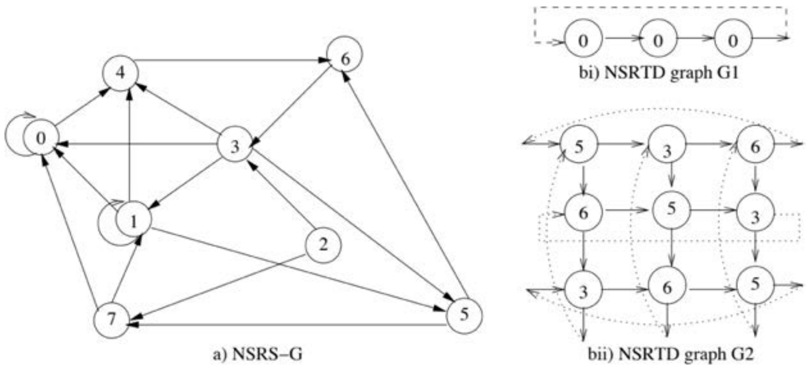


Figure 6. NSRTD of the CA with rule 116.

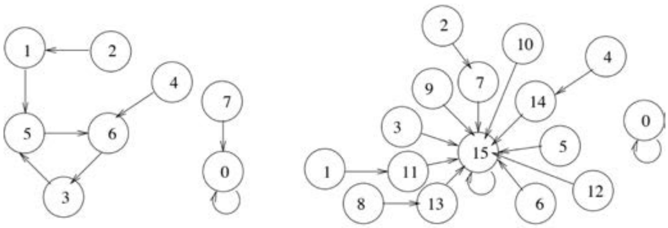
- *NSRS-G*. The NSRS graph (NSRS-G) explores all possible sequences of states (NSRSs) that can be generated by a CA cell during its evolution. The i^{th}-cell rule R_i of a CA is portrayed by a directed graph G(V, E). The V consists of the states' T_i^t ($t = 0, 1, 2, \ldots$) of cell i and $e_i(T_i^t, T_i^{t+1}) \in E$, where T_i^{t+1} is the NSR of T_i^t. Figure 6(a) is the NSRS-G of rule 116.
- *NSRTD*. An NSRTD explores all possible attractors of a CA. It is a set of directed graphs (each corresponds to an attractor) constructed from NCR, NSR, NSRS and NSRS-G of a CA and represented as the two-dimensional arrangement of nodes. Each node in a graph is an RMT. There is an edge (T_i^t, T_i^{t+1}) in a graph of the NSRTD if and only if T_i^{t+1} is the NSR of T_i^t and also T_{i+1}^t is the NCR of T_i^t. Further, T_i^t ($\forall\, t$) form an NSRS for cell i. Figure 6(b) shows a part of the NSRTD.

It is reported in [16] that a CA with rule R can have only fixed-point attractors if NSRS-G or the NSRTD does not have a multi-length cycle.

Example 3. The NSRS-G shown in Figure 6 contains the multi-length cycles $5 \to 6 \to 3 \to 5$, $6 \to 3 \to 1 \to 4 \to 6$, $4 \to 6 \to 3 \to 4$, $0 \to 4 \to 6 \to 3 \to 1 \to 0$, and so on. It has the self-loops $0 \to 0 \to$ and $1 \to 1 \to$. It can be observed in the NSRTD of Figure 6(bii) that when a cell i changes its state on T(5), the next cell $i+1$ changes its state on T(3) (T(3) is the NCR of cell i (Table 2)). Now, as per NSRS $5 \to 6 \to 3 \to 5$, the NSR of cell i is T(6). Simultaneously, T(5) is the NSR of cell $i+1$ (also as per the NSRS $5 \to 6 \to 3 \to 5$), which is the NCR of T(6) (at cell i). In this case, it can be observed that if cell i evolves following the $\text{NSRS}_i = 5 \to 6 \to 3 \to 5$, then cell $i+1$ can evolve as per the NSRS_{i+1} $3 \to 5 \to 6 \to 3$. These two NSRSs are referred to as compatible.

Definition 3. (Compatibility of NSRSs) The NSRS_i and NSRS_{i+1} are compatible if the j^{th} NSR of NSRS_{i+1} is the NCR of the j^{th} NSR of NSRS_i for $\forall\, j$.

Figure 6(bii) denotes that the CA with rule 116 of length $3n$ ($n = 1, 2, 3, \ldots$) has the multi-length cycle attractor. On the other hand, Figure 6(bi) denotes the existence of a fixed-point attractor (Figure 7(a)).

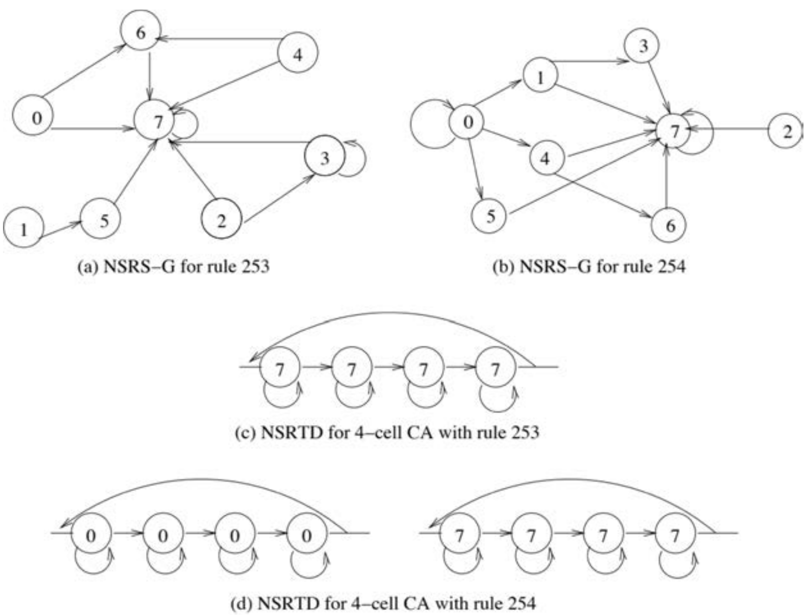


a) The PBCA <116, 116, 116>

b) The PBCA <254, 254, 254, 254>

Figure 7. State spaces of two uniform CAs with rules 116 and 254.

Figures 8(a) and 8(b) show the NSRS-Gs for the uniform CAs configured with rules 253 and 254, respectively. The self-loops in Figure 8(a) are $3 \to 3$ and $7 \to 7$, and in Figure 8(b) these are $0 \to 0$ and $7 \to 7$. The NSRTDs of the two CAs are Figures 8(c) and 8(d). It can be observed that rule 253 configures an SACA and rule 254 configures a TACA.



(a) NSRS–G for rule 253

(b) NSRS–G for rule 254

(c) NSRTD for 4–cell CA with rule 253

(d) NSRTD for 4–cell CA with rule 254

Figure 8. NSRS-Gs and NSRTDs for rules 253 and 254.

https://doi.org/10.25088/ComplexSystems.33.1.31

The algorithmic steps to decide on whether the state space of an n-cell CA with rule R contains a multi-length cycle and/or fixed points, for all n, are summarized in Algorithm 1. It accepts R, n and NCRs. It then constructs NSRS-G for cell i to decide on the fixed points.

The algorithmic steps to decide on whether the state space of an n-cell CA with rule R contains a multi-length cycle and/or fixed points, for all n, are summarized in Algorithm 1. It accepts R, n and NCRs. It then constructs NSRS-G for cell i to decide on the fixed points.

Algorithm 1. Cycle_Count (R, n, NCRs).

Input: R, n, NCRs

Output: MLC: "true" if multi-length cycle exists, SLA: number of fixed-point attractors.

1. Find NSRSs
2. Construct NSRS-G
3. (a) Find self-loops SL and multi-length cycles ML in NSRS-G

   (b) Combine LP = SL, ML
4. If ML = $\emptyset$ (/* that is, no multi-length cycle in the CA with rule R */),

   then (to find number of fixed-point attractors)

   {

   (i) Construct all unique sequences of RMTs (nodes) P_u of length n, where
   $P_u = P_{1u} \to P_{2u} \to P_{3u}$ .... $P_{(n-2)u} \to P_{(n-1)u} \to P_{nu}$, such that
   $P_{iu} \in$ SL $\forall_i$ and P_{iu} is NCR of $P_{(i-1)u}$ $\forall_{i\neq 1}$; P_{1u} is NCR of P_{nu};

   (ii) SLA = number of P_u (each of n-length);

   (iii) return SLA

   }

   else

   {

   (I) Find compatibility class C among elements LP_1, LP_2, ... of LP, where
   C = {C_j | C_j is compatible pair (LP_k, LP_l); LP_k & LP_l belong to L

   };

   /*(LP_k, LP_l) is a compatible pair if and only if

   (i) $LP_{lq} \in LP_l$ is the NCR of $LP_{kr} \in LP_k$, and

   (ii) NSR(LP_{lq}) is NCR(NSR(LP_{kr})), $\forall_q$*/

(II) Construct all sequences of RMTs (nodes) P_u of length n, where
$P_u = P_{1u} \to P_{2u} \to P_{3u} \to P_{4u} \ldots P_{(n-2)u} \to P_{(n-1)u} \to P_{nu}$
such that

(a) for $m = 1$ to n-1 {
$P_{(m+1)u}$ is an NCR of P_{mu}, where $P_{mu} \in LP_o$, $P_{(m+1)u} \in LP_p$ for some o and p, and $(LP_o, LP_p) \in C$;
$PLP_m = LP_0$;
$LP_0 = LP_p$; }

(b) P_{1u} is an NCR of P_{nu} and $(PLP_n, PLP_1) \in C$;
/*PLP_1 and PLP_n are the result of (a)*/

(III) $LPA_u = PLP_1, PLP_2, PLP_3, \ldots PLP_n$ selected in step 4(II)

(IV) Number of attractors = number of P_u (n-length)
/*Attractor's cycle length is the maximum of cycle lengths of $PLP_h \in$L-PA$_u$ for a P_u*/

(V) if there is a $PLP_h \in LPA_u$ without self-loop for a P_u, then
/*It denotes existence of multi-length cycle*/
MLC = true; and Exit (return 0)

else

{
SLA = number of P_u
return SLA
}

end

Example 4. Consider the CA with rule 254. The NSRSs for the CA cells are $0 \to 0$; $7 \to 7$; $0 \to 1 \to 7 \to 7$; $0 \to 1 \to 3 \to 7 \to 7$; $2 \to 7 \to 7$; $3 \to 7 \to 7$; $4 \to 7 \to 7$; $4 \to 6 \to 7$; … (computation not shown). Its NSRS-G is shown in Figure 8(b). The self-loops are $SL_1 = T(0) \to T(0)$ and $SL_2 = T(7) \to T(7)$. That is, $SL = T(0) \to T(0), T(7) \to T(7)$. Here, $ML = \emptyset$.

As per step 4 of Algorithm 1, as $ML = \emptyset$ in NSRS-G, 4(i), (ii) and (iii) are computed.

As per 4(i), two possible unique sequences of RMTs $0 \to 0 \to 0 \to 0$ and $7 \to 7 \to 7 \to 7$, for a four-cell CA, are shown in Figure 8(d). That is, the number of fixed-point attractors in the four-cell CA is two (step 4(ii)). The attractors are all-0s and all-1s.

Example 5. This example considers the CA with rule 116. Its NSRSs are $0 \to 0$; $1 \to 1$; $1 \to 1 \to 0 \to$; $2 \to 7 \to 0 \to 0$; $2 \to 7 \to 1 \to 1$; $4 \to 6 \to 3 \to 4$; ….

The NSRS-G is shown in Figure 6(a). Self-loops and multi-length cycles in the NSRS-G (step 3(a) of Algorithm 1) $SL = 0 \to 0, 1 \to 1$; and $ML = 5 \to 6 \to 3 \to 5, 0 \to 4 \to 3 \to 0, \ldots$.

Therefore, the combined loops (step 3(b)) $LP = 0 \to 0, 1 \to 1, 5 \to 6 \to 3 \to 5, 0 \to 4 \to 3 \to 0, \ldots$. Here, $ML \neq \emptyset$.

Therefore, steps 4(I) to 4(II) of Algorithm 1 are to be computed. As per step 4(I), the compatible pairs are ((0), (0)); ((0), (1)); ((5, 6, 3, 5), (3, 5, 6, 3)) and ((3, 5, 6, 3), (5, 6, 3, 5)).

As per step 4(II), the sequences of RMTs are $P_1 = 0 \to 0 \to 0 \to 0\ldots$ (corresponds to G1 of Figure 6(a)), and

$P_2 = 5 \to 3 \to 6 \to 5\ldots$ (corresponds to G2 of Figure 6(b)).

Here, for P_1, the $PLP_1 = PLP_2 = \ldots PLP_n = 0 \to 0$. For P_2, the $PLP_1 = 5 \to 6 \to 3 \to 5$, $PLP_2 = 3 \to 5 \to 6 \to 3$, $PLP_3 = 6 \to 3 \to 5 \to 6$, $PLP_4 = 5 \to 6 \to 3 \to 5, \ldots.$

Now, the $LPA_1 = (0 \to 0), (0 \to 0), (0 \to 0)\ldots$; and $LPA_2 = (5 \to 6 \to 3 \to 5), (3 \to 5 \to 6 \to 3), (6 \to 3 \to 5 \to 6), \ldots$ (step 4(III)).

As per step 4(IV), the number of attractors is 2. The cycles (PLP_h) in LPA_2 are multi-length, therefore, MLC = true and exit (step 4(V)).

Finding NSRs (step 1 of Algorithm 1) requires constant time.

The maximum number of nodes in an NSRS is eight. The most the out-degree of a node can be is four. Therefore, step 2 of Algorithm 1 also requires constant time.

Step 3 explores all possible cycles from a graph of eight nodes (with out-degree 4). However, the number of such cycles is finite and also independent of n. That is, step 3 is also independent of n.

The major computation done in steps 4(i) to (iii) is the exploration of all possible n-length sequences of RMTs (step 4(i)). Such an exploration is of $O(n)$ complexity. The sequence count (number of graphs in the NSRTD) depends on the CA rule. In the worst case, the maximum number of such sequences can be 2^n (corresponding to the 2^n fixed-point attractors). Hence, time complexity is exponential in the worst case. But in reality, the sequence count is much less than 2^n for most of the rules. Step 4(ii), however, counts the sequences in $O(p)$ time, while p is the number of sequences identified in step 4(i).

Similarly, step 4(I) of Algorithm 1 computes the compatible classes. It does not depend on n but does depend on the cycle count in LP. Extracting compatible classes for an R is a one-time cost. It can be stored in a dictionary and input to Algorithm 1. Exploring all possible sequences of RMTs of length n in step 4(II) has the same complexity as that of step 4(i). Further, the complexity of steps 4(III), 4(IV) and 4(V) is similar to 4(II).

The primary target of this research is to explore the SACA and TACA rules from the 256 ECA rules. The next section finds such rules.

4. SACA and TACA Rules

This section explores the 256 rules in 2-state 3-neighborhood CAs that can form uniform SACA and TACA rules of arbitrary length n.

The scheme is based on the NSRTD. The following properties reduce the search space (out of 256 rules) for SACA and TACA rules.

Property 1. [23] A rule R can contribute to the formation of fixed-point attractor(s) if at least one of the RMTs T(0), T(1), T(4) or T(5) belongs to T / 0, or the T(2), T(3), T(6) or T(7) belongs to T / 1, that is, a passive RMT.

Following Property 1, the 256 ECA rules are classified into nine classes (class 0 to 8) in [23]. For example, rule 232 (11101000) belongs to class 6. The six RMTs of rule 232 (T(7), T(6), T(4), T(3), T(1) and T(0)) are passive. The following observations were made after extensive experimentation.

Observation 1. A TACA rule must be unbalanced; that is, the number of RMTs in T / 0 and T / 1 is unequal.

Observation 2. An SACA rule belongs to either class 4 or class 5 of [23].

Observation 3. A TACA rule belongs to either class 5 or class 6 of [23].

Property 1 and subsequent observations effectively reduce the candidate rules from 256 to 89 (resp. 65) for SACAs (resp. TACAs).

Theorem 1. In uniform SACAs, there can be fixed-point attractors that are either all-0s or all-1s.

Proof. Consider the fixed-point attractor of a CA $b_1 b_2 \ldots b_{i-1} b_i b_{i+1} \ldots b_n$ and A1(……xyz……) is a nonzero fixed-point attractor, where x represents b_{i-1}, y represents b_i and z is for b_{i+1}. This implies A2(……$b_{i-1}xy$……), where $b_i = x$ and $b_{i+1} = y$; and A3(……yzb_{i+1}……), where $b_{i-1} = y$ and $b_i = z$, are also fixed-point attractors. Now, if these three denote the same state, $b_i = x = y = z$. That is, xyz is either 000 or 111. Hence, a uniform SACA can have fixed-point attractors that are either all-0s or all-1s. ■

Theorem 1 suggests that an even rule between 128 and 254 cannot be an SACA rule, as both the conditions: T(7) is in T / 1 (realizes the all-1s fixed-point attractor) and T(0) is in T / 0 (realizes the all-0s fixed-point attractor) cannot hold simultaneously in an SACA rule. Similarly, an odd rule between 1 and 127 cannot be an SACA rule. Since T(7) is in T / 0 and T(0) is in T / 1, neither can form the all-0s or all-1s fixed-point attractor. This effectively reduces the number of candidate SACA rules from 89 to 52. The NSRTD for each of these rules is constructed following Algorithm 2. This exercise results in the six SACA rules (Table 3) that exist for the periodic boundary CA.

Theorem 2. For uniform TACAs, the fixed-point attractors are only all-0s or all-1s.

Rule	Class	Fixed Point	Property 0	Remarks
		Attractor	Denial RMT	
0	4	000000	T(7),T(6),T(3),T(2)	Unbalanced
8	5	000000	T(7),T(6),T(2)	Unbalanced
64	5	000000	T(7),T(3),T(2)	Unbalanced
239	5	111111	T(5),T(1),T(0)	Unbalanced
253	5	111111	T(5),T(4),T(0)	Unbalanced
255	4	111111	T(5),T(4),T(1),T(0)	Unbalanced

Table 3. SACA rules in periodic boundary CAs.

Proof. Consider the fixed-point attractor of a CA is $b_1 b_2 \ldots b_{i-1} b_i b_{i+1} \ldots b_n$ and A1(……xyz……) is a nonzero fixed point, where x represents b_{i-1}, y represents b_i and z is for b_{i+1}. This implies A2(……$b_{i-1}xy$……), where $b_i = x$ and $b_{i+1} = y$; and A3(……yzb_{i+1}……), where $b_{i-1} = y$ and $b_i = z$, are also fixed-point attractors. Now, if these three states are to be the same, then $b_i = x = y = z$. That is, xyz can either be 000 or 111. Hence, a uniform TACA cannot have a fixed point other than all-0s or all-1s. ■

Theorem 2 points to the fact that a TACA rule must be even (as T(0) is in T / 0, for a TACA rule, to ensure that the all-0s state is a fixed-point attractor). Further, such rules must be between 128 and 254 (as T(7) is in T / 1 for the TACA rule). This further reduces the number of candidates for the TACA rules from 65 to 23. Constructing the NSRTD for each of the 23 rules enables identifying all six TACA rules (Table 4) that exist for the periodic boundary CAs. Here, we define R as an SACA (resp. TACA) rule if for all $n \geq 3$, the n-cell uniform CA configured with R is the SACA (resp. TACA).

Rule	Class	Fixed Points	Property 0	Remarks
		Attractors	Denial RMT	
128	5	000000, 111111	T(6),T(3),T(2)	Unbalanced
136	6	000000, 111111	T(6),T(2)	Unbalanced
192	6	000000, 111111	T(3),T(2)	Unbalanced
238	6	000000, 111111	T(5),T(1)	Unbalanced
252	6	000000, 111111	T(5),T(4)	Unbalanced
254	5	000000, 111111	T(5),T(4),T(1)	Unbalanced

Table 4. TACA rules in periodic boundary CAs.

Algorithm 2 is devised to find the SACA (resp. TACA) rules.

Algorithm 2. Find_SACA_TACA_rules (R, NCRs).

Input: R, NCRs

Output: DECISION: decision on SACA/TACA rule

1. Find NSRSs (step 1 of Algorithm 1)
2. Construct NSRS-G (step 2 of Algorithm 1)
3. Find self-loops SL and multi-length cycles ML in NSRS-G and combine LP = SL, ML (step 3 of Algorithm 1)
4. If ML = $\emptyset$, **then**

{

5. Construct all unique sequences of RMTs

$P_u = P_{1u} \to P_{2u} \to P_{3u} \ldots P_{(n-2)u} \to P_{(n-1)u} \to P_{nu}$,

of length n (step 4(i) of Algorithm 1)

6. SLA = number of P_u (step 4(ii) of Algorithm 1)
7. if SLA = 1, then return (DECISION = R is a SACA rule)
   elseif SLA = 2, then return (DECISION = R is a TACA rule)
   else Exit (return 0)

}

else

{

8. Find compatibility class C (step 4(I) of Algorithm 1)
9. Construct all sequences of RMTs

$P_u = P_{1u} \to P_{2u} \to P_{3u} \to P_{4u} \ldots P_{(n-2)u} \to P_{(n-1)u} \to P_{nu}$

of length n (step 4(II) of Algorithm 1)

10. LPA_u = PLP_1, PLP_2, PLP_3, … PLP_n as selected in step 4(II) of Algorithm 1
11. if there is a $PLP_h \in LPA_u$ without self-loop for a P_u, then
    Exit (return 0);
    else

{

SLA = number of P_u;

12. if SLA = 1, then (DECISION = R is a SACA rule)
    elseif SLA = 2, then (DECISION = R is a TACA rule)
    else Exit (return 0)

}

}

end

5. Nonuniform SACAs or TACAs

This section reports on the synthesis of a set of nonuniform SACAs (TACAs). The synthesis is realized through hybridizing uniform

SACAs (TACAs) configured with rule R_S (R_T) by the rule R_h. When a uniform SACA (TACA) with rule R_S (R_T) is hybridized by rule R_h, then we need to construct the NSRS-G for $R_S R_h R_S$ ($R_T R_h R_T$), $R_S R_S R_h$ ($R_T R_T R_h$), and $R_h R_S R_S$ ($R_h R_T R_T$) in addition to the NSRS-G for $R_S R_S R_S$ ($R_T R_T R_T$). The NSRTD of the hybridized CA $\langle R_S, \ldots, R_h, \ldots, R_S \rangle$ ($\langle R_T, \ldots, R_h, \ldots, R_T \rangle$) is then constructed following these four NSRS-Gs. However, our objective in this section is to synthesize such nonuniform SACAs (TACAs) through hybridizing a single point or cell. The following properties reduce the search space while identifying the R_h.

Theorem 3. If a uniform SACA or TACA is hybridized by a class 0 rule, then the hybridized nonuniform CA cannot form a fixed-point attractor.

Proof. All the RMTs of class 0 rules are active. The next state of a fixed-point attractor state A1 is the A1. A uniform SACA (resp. TACA) has one (resp. two) and only one (resp. two) fixed-point attractor(s). Consider that S_a ($b_0 b_1 \ldots b_{i-1} d b_{i+1} \ldots b_{n-1}$) is a fixed-point attractor of the nonuniform CA after hybridizing an SACA by a class 0 rule at cell i. The NS S_a^1 of S_a then differs at least at the position of cell i; that is, it is d' because the RMT at which cell i changes its state is active. That is, S_a^1 cannot be the S_a. Therefore, an SACA hybridized by a class 0 rule cannot form a fixed-point attractor. Similarly, this can be proved for uniform TACA hybridization. ■

Example 6. Rule 64 is an SACA rule (Figure 3). Rule 51 is in class 0. The NSRTD of the uniform CA with rule 51 in Figure 9(a) shows that all the attractors have multi-length cycles. The relevant four sets of NSRS-Gs, that is, for ⟨64, 64, 64⟩ (⟨$R_S R_S R_S$⟩), ⟨64, 64, 51⟩ (⟨$R_S R_S R_h$⟩), ⟨64, 51, 64⟩ (⟨$R_S R_h R_S$⟩) and ⟨51, 64, 64⟩ (⟨$R_h R_S R_S$⟩), for the nonuniform CA ⟨64, 64, …51…64⟩ hybridized by the rule 51 are

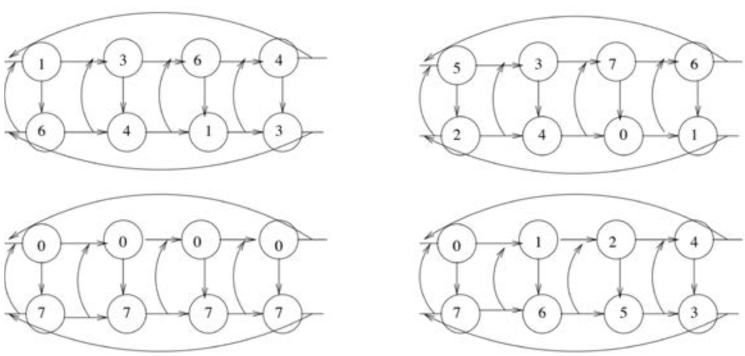


Figure 9(a). (*continues*)

shown in Figure 10. Figure 9(b) shows the NSRTD for the CA ⟨64, 64, 51, 64⟩. It can be observed from the figure that the nonuniform CA has a multi-length cycle attractor. That is, if a class 0 rule

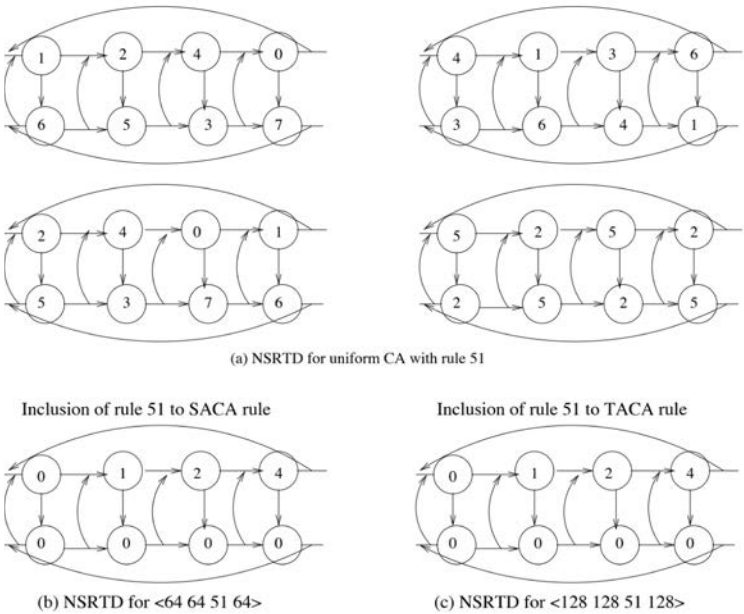


Figure 9. Characteristics of attractors of a CA hybridized with rule 51.

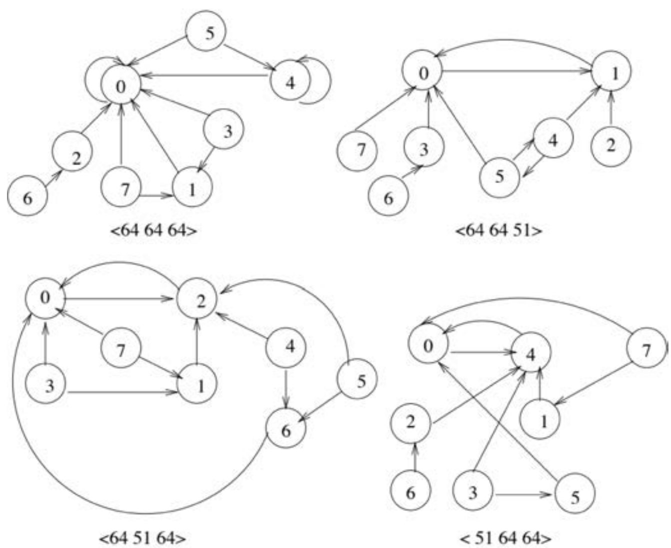


Figure 10. NSRS-Gs for the hybridization of rule 51 in an SACA with rule 64.

(rule 51) is considered for hybridizing uniform SACAs, then it turns into a multi-length cycle attractor CA. A similar result can be found when hybridizing a TACA as observed in Figure 9(c) for ⟨128, 128, 51, 128⟩. The relevant NSRS-Gs are shown in Figure 11.

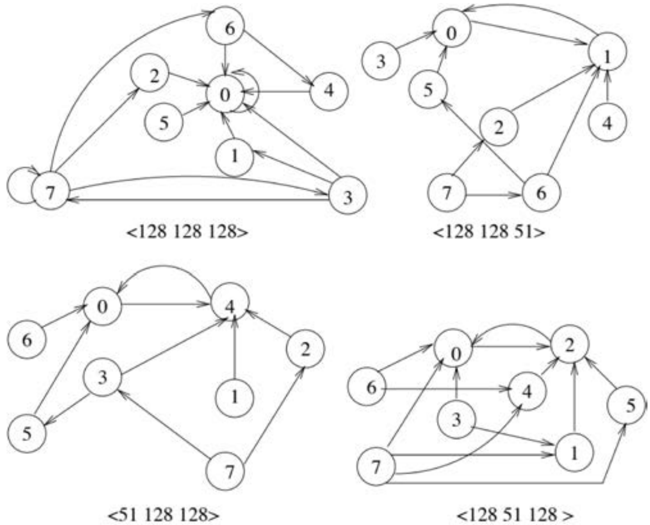


Figure 11. NSRS-Gs for the hybridization of rule 51 in a TACA with rule 128.

The following subsections explore the set of rules R_h that can form nonuniform SACAs (TACAs) through hybridizing uniform SACAs (TACAs).

5.1 Uniform SACA to Nonuniform SACA

There are hybridization rules that get absorbed in uniform SACAs; that is, the attractor of the nonuniform SACA thus formed and the attractor of the uniform SACA are the same. The hybridization rules that form nonuniform SACAs with a different attractor than that of the uniform SACAs are identified first in this subsection.

Conjecture 1. The hybridization rule that converts a uniform SACA to a nonuniform SACA with an attractor that is different from that of the uniform SACA should have passive T(2) (resp. T(5)), while the uniform SACA has all-0 (resp. all-1) attractors.

Conjecture 2. The attractor of a nonuniform SACA resulting from hybridizing a uniform SACA with all-0 (resp. all-1) attractors is either an all-0 (resp. all-1) attractor or A (resp. B).

For example, the attractor of a four-cell uniform SACA with rule 64 is 0000 (Figure 3). The nonuniform SACA hybridized with rule 23 has the attractor 0010 (Figure 12), which is different from that of the uniform SACA.

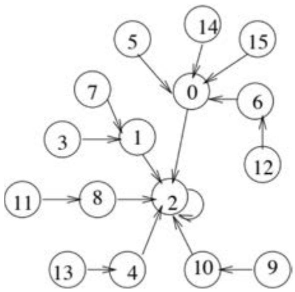


Figure 12. State transitions of CA ⟨64, 64, 23, 64⟩.

Theorem 4. The hybridization rule R_h that results in nonuniform SACAs from the uniform SACAs with an attractor that is different from the attractor of the uniform SACA is an odd rule (resp. less than 128) when the uniform SACA has all-0 (resp. all-1) attractors.

Proof. For the all-0 attractor SACA rules R_S (rules 0, 8 and 64 (Table 3)), the T(0) is always 0, that is, passive. Now, if hybridization rule R_h is even, that is, T(0) of R_h is 0, then the nonuniform CA has all-0 attractors. Therefore, to get an attractor other than all-0s, the R_h must have the T(0) as 1, that is, R_h is to be odd. Similarly, for the all-1 attractor SACA rules (rules 239, 253 and 255), the T(7) is passive. Therefore, the hybridization rule that results in the attractor, other than all-1s, must have the T(7) as active (0), that is, the R_h must be less than 128. ■

Theorem 4 indicates that the number of probable hybridization rules that transform a uniform SACA to a nonuniform SACA having other than all-0 attractors (resp. all-1s) is 128 (resp. 128). For example, consider the uniform SACA with rule 64. It has all-0 attractors. The hybridization rules that can transform the uniform SACA to an SACA with nonzero attractors are odd. Further, out of 128 odd rules in 3-neighborhood, 27 transform the uniform SACA with rule 64 to a TACA (Section 5). The remaining 101 odd rules are verified with NSRS-Gs and NRSTD. It is observed that 37 rules are the desired hybridization rules (third row of Table 5). The list of hybridization rules that form nonuniform SACAs with a different attractor than that of the corresponding uniform SACA is given in Table 5.

Example 7. Rule 253 is an SACA rule. Figure 13(b) displays the NSRSG and NSRTD when rule 252 is considered for the hybridization of the uniform SACA with rule 253. It can be observed that the hybridization rule 252 gets absorbed (Figure 8(c)).

SACA Rule	Hybridization Rules
0	5, 7, 13, 15, 21, 23, 29, 31, 37, 39, 45, 47, 53, 55, 61, 63, 69, 71, 77, 79, 85, 87, 93, 95, 101, 103, 104, 109, 111, 117, 119, 125, 127, 133, 135, 141, 143, 149, 151, 157, 159, 165, 167, 173, 175, 181, 183, 189, 191, 197, 199, 205, 207, 213, 215, 221, 223, 229, 231, 237, 239, 245, 247, 253, 255
8	5, 7, 13, 15, 21, 23, 29, 31, 37, 39, 45, 47, 53, 55, 61, 63, 133, 135, 141, 143, 149, 151, 157, 159, 165, 167, 173, 175, 181, 183, 189, 191
64	5, 7, 13, 21, 23, 31, 37, 39, 53, 55, 63, 69, 71, 77, 85, 87, 101, 103, 117, 119, 125, 133, 135, 149, 151, 165, 167, 181, 183, 197, 199, 213, 215, 229, 231, 245, 247
239	2, 3, 6, 7, 10, 11, 14, 15, 18, 19, 22, 23, 26, 27, 30, 31, 66, 67, 70, 71, 74, 75, 78, 79, 82, 83, 86, 87, 90, 91, 94, 95
253	16-31, 80-95
255	0-31, 64-95

Table 5. Hybridization rules for the synthesis of SACAs with different attractors.

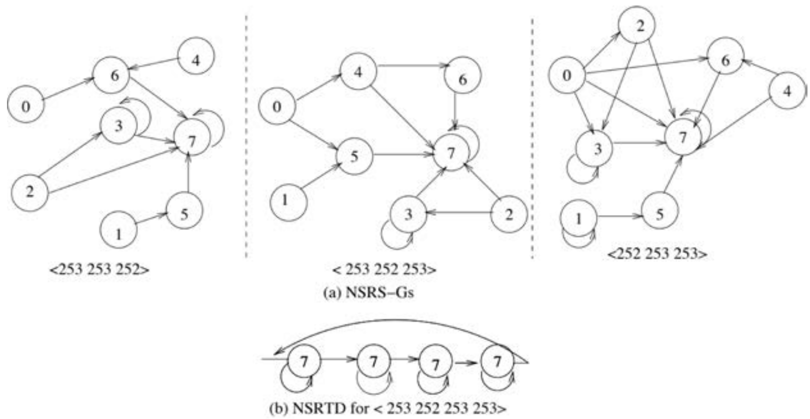


Figure 13. Hybridization rule 252 in an SACA with rule 253.

Theorem 5. The hybridization rule that gets absorbed in a uniform SACA cannot be from class 8.

Proof. The proof is similar to that of Theorems 3 and 7(c). ■

Example 8. Figure 14 displays the NSRS-Gs for the hybridization of an SACA with rule 64 by the class 8 rule 204. The NSRTD of a nonuniform CA ⟨64, 204, 64, 64⟩ is shown in Figure 15. It can be observed that the class 8 rule transforms the uniform SACA to a multi-attractor CA; that is, such a rule cannot be absorbed in the uniform SACA.

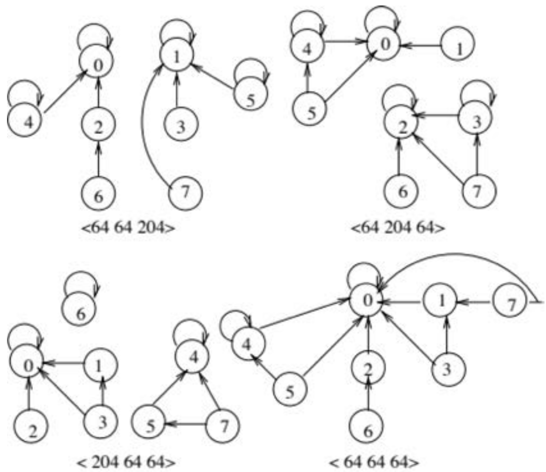


Figure 14. NSRS-Gs for the hybridization of rule 204 in an SACA with rule 64.

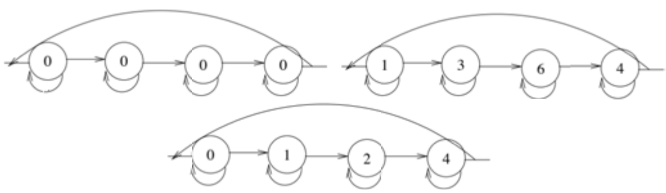


Figure 15. NSRTD for CA ⟨64, 204, 64, 64⟩.

Table 6 contains the rules that can form nonuniform SACAs having the same attractor as that of the corresponding uniform SACA.

5.2 TACA to SACA

An SACA can also be synthesized through the hybridization of a uniform TACA. An example hybridization is shown in Figure 16. The NSRS-G of the TACA rule 254 is shown in Figure 8(b). There are two loops, $0 \to 0$ and $7 \to 7$ in the NSRS-G. The loops form two sequences of RMTs in the NSRTD (Figure 8(d)) that correspond to the two fixed-point attractors, all-0s and all-1s (0000 and 1111 for a four-cell CA). To consider rule 253 as the hybridization rule, we need to analyze the 4-set of the NSRS-Gs, that is, for ⟨254 254 254⟩, ⟨254 254 253⟩, ⟨254 253 254⟩ and ⟨253 254 254⟩. Figures 8(b) and 16(a) depict the NSRS-Gs. The NSRTD of CA ⟨254 253 254 254⟩ is shown in Figure 16(b). It points to only one fixed-point attractor (all-1s state). That is, the hybridization rule 253 transforms the uniform TACA with rule 254 to a nonuniform SACA.

SACA Rule	Hybridization Rules
0	2, 8, 10, 16, 18, 24, 26, 32, 34, 40, 42, 48, 50, 56, 58, 64, 66, 72, 74, 80, 82, 88, 90, 96, 98, 106, 112, 114, 120, 122, 128, 130, 136, 138, 144, 146, 152, 154, 160, 162, 168, 170, 176, 178, 184, 186, 192, 194, 200, 202, 208, 210, 216, 218, 224, 226, 232, 234, 240, 242, 248, 250
8	0, 2, 10, 16, 18, 24, 26, 32, 34, 40, 42, 48, 50, 56, 58, 98, 106, 128, 130, 136, 138, 144, 146, 152, 154, 160, 162, 168, 170, 176, 178, 184, 186
64	0, 2, 16, 18, 32, 34, 48, 50, 66, 80, 82, 88, 96, 98, 100, 112, 114, 128, 130, 144, 146, 160, 162, 176, 178, 192, 194, 208, 210, 224, 226, 240, 242
239	162, 163, 166, 167, 170, 171, 174, 175, 178, 179, 182, 183, 186, 187, 190, 191, 226, 227, 230, 231, 234, 235, 238, 242, 243, 246, 247, 250, 251, 254, 255
253	176-191, 240-249, 250-252, 254, 255
255	160-191, 224-254

Table 6. Hybridization rules that get absorbed into an SACA rule.

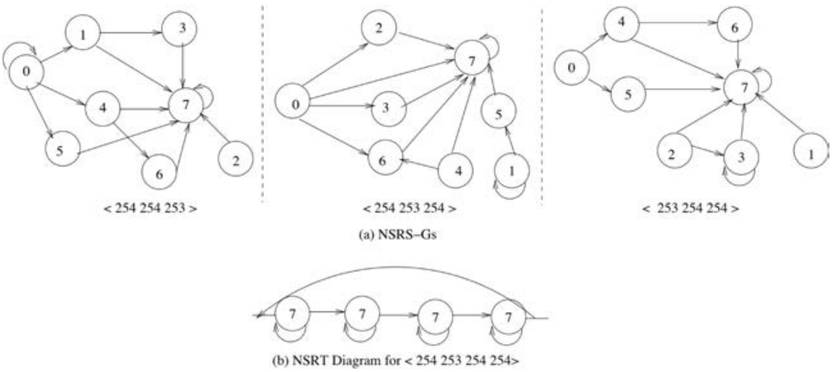


Figure 16. Hybridization of rule 253 in a TACA with rule 254.

Conjecture 3. A uniform TACA can be transformed to a nonuniform SACA through hybridization. The hybridization rule for the TACA rules 128, 136 and 192 should have T(7) as 0, and for TACA rules 238, 252 and 254, T(0) should be 1.

Theorem 6. The hybridization rule that transforms a uniform TACA to a nonuniform SACA cannot be of class 8.

Proof. The attractors of the uniform TACAs are the all-0s and all-1s. That is, T(0) is 0 and T(7) is 1 of a TACA rule R_T. As T(0) is 0 and T(7) is 1 for the class 8 rule R_h, the CA constructed through hybridization of the uniform TACA by the rule R_h at cell i also forms all-0s and all-1s attractors. Therefore, if hybridized by a class 8 rule, the uniform TACA cannot form an SACA. ■

Table 7 reports on the details of hybridization rules that transform a uniform TACA to an SACA.

TACA Rule	Hybridization Rules
128	0, 2, 5, 7, 8, 10, 13, 15, 16, 18, 21, 23, 24, 26, 29, 31, 32, 34, 37, 39, 40, 42, 45, 47, 48, 50, 53, 55, 56, 58, 61, 63, 64, 66, 69, 71, 72, 74, 77, 79, 80, 82, 85, 87, 88, 90, 93, 95, 96, 98, 101, 103, 104, 106, 109, 111, 112, 114, 117, 119, 120, 122, 125, 127
136	0, 2, 5, 7, 8, 10, 13, 15, 16, 18, 21, 23, 24, 26, 29, 31, 32, 34, 37, 39, 40, 42, 45, 47, 48, 50, 53, 55, 56, 58, 61, 63
192	0, 2, 5, 7, 16, 18, 21, 23, 32, 34, 37, 39, 48, 50, 53, 55, 64, 66, 69, 71, 80, 82, 85, 87, 96, 98, 101, 103, 112, 114, 117, 119
238	3, 7, 11, 15, 19, 23, 27, 31, 67, 71, 75, 79, 83, 87, 91, 95, 163, 167, 171, 175, 179, 183, 187, 191, 227, 231, 235, 239, 243, 247, 251, 255
252	17, 19, 21, 23, 25, 27, 29, 31, 81, 83, 85, 87, 89, 91, 93, 95, 177, 179, 181, 183, 185, 187, 189, 191, 241, 243, 245, 247, 249, 251, 253, 255
254	1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 161, 163, 165, 167, 169, 171, 173, 175, 177, 179, 181, 183, 185, 187, 189, 191, 225, 227, 229, 231, 233, 235, 237, 239, 241, 243, 245, 247, 249, 251, 253, 255

Table 7. Hybridization rules that synthesize nonuniform SACAs from uniform TACAs.

5.3 SACA to TACA

Figures 17 and 18 explain hybridizing a uniform SACA with rule 253 (R_S) by the rule 238 (R_h). The set of NSRGs for ⟨253 253 238⟩ (⟨$R_S R_S R_h$⟩), ⟨253 238 253⟩ (⟨$R_S R_h R_S$⟩), ⟨238 253 253⟩ (⟨$R_h R_S R_S$⟩) (Figure 17(a)) and ⟨253 253 253⟩ (Figure 8(a)) dictates the state space

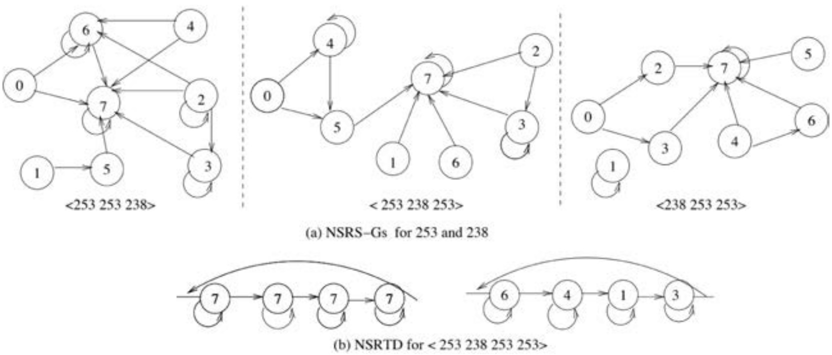


Figure 17. Hybridization of rule 238 in an SACA with rule 253.

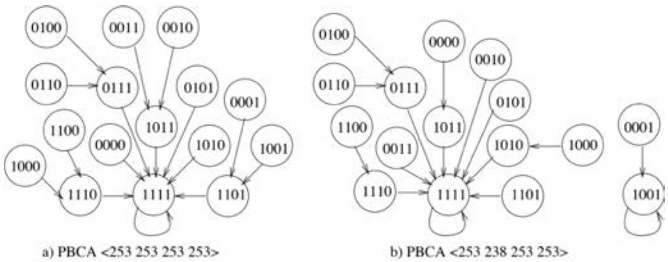


Figure 18. State space of an SACA with rule 253 and a TACA hybridized by rule 238.

of hybridized CAs. The NSRTD of the CA resulting from hybridization, shown in Figure 17(b), points to the existence of two fixed-point attractors (Figure 18(b)); that is, the nonuniform CA is a TACA.

Conjecture 4. A TACA cannot be synthesized through hybridizing a uniform SACA by a rule from class 1.

Example 9. Consider hybridizing an SACA with rule 64 by the rule 55 of class 1. Figure 19 shows that the uniform CA with rule 64 is hybridized by rule 55. This points to the fact that the hybridization does not form a TACA.

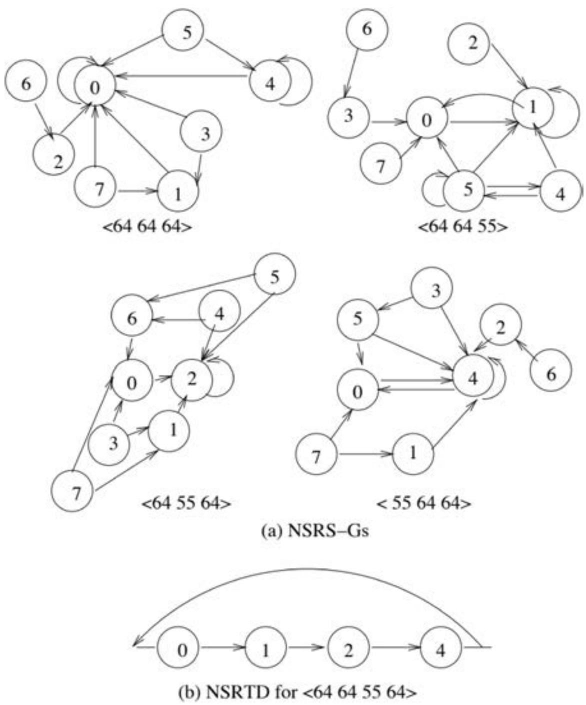


Figure 19. NSRS-Gs and NSRTD of a uniform SACA hybridized by rule 55.

Table 8 reports on the set of hybridization rules, other than from classes 0 (Theorem 3) and 1 (Conjecture conjec-4), that can be considered for the synthesis of TACAs from uniform SACAs. All such rules are verified through the construction of an NSRTD.

SACA Rule	Hybridization Rules
0	4, 6, 12, 14, 20, 22, 28, 30, 36, 38, 44, 46, 52, 54, 60, 62, 68, 70, 76, 78, 84, 86, 92, 94, 100, 102, 108, 110, 116, 118, 124, 126, 132, 134, 140, 142, 148, 150, 156, 158, 164, 166, 172, 174, 180, 182, 188, 190, 196, 198, 204, 206, 212, 214, 220, 222, 228, 230, 236, 238, 244, 246, 252, 254
8	4, 6, 12, 14, 20, 22, 28, 30, 36, 38, 44, 46, 52, 54, 60, 62, 64, 66, 69, 71, 72, 74, 77, 79, 80, 82, 85, 87, 88, 90, 93, 95, 96, 98, 101, 103, 104, 106, 109, 111, 112, 114, 117, 119, 120, 122, 125, 132, 134, 140, 142, 148, 150, 156, 158, 164, 166, 172, 174, 180, 182, 188, 190, 192, 194, 197, 199, 200, 202, 205, 207, 208, 210, 213, 215, 216, 218, 221, 223, 224, 226, 229, 231, 232, 234, 237, 239, 240, 242, 245, 247, 248, 250, 253, 255
64	4, 6, 8, 10, 15, 20, 22, 24, 26, 29, 36, 38-40, 42, 45, 47, 52, 54, 56, 58, 61, 68, 70, 72, 74, 79, 84, 86, 90, 92, 93, 94, 95, 102, 104, 106, 108, 109, 111, 116, 118, 120, 122, 127, 132, 134, 136, 138, 141, 143, 148, 150, 152, 154, 156, 157, 159, 164, 166, 168, 170, 172, 173, 175, 180, 182, 184, 186, 189, 191, 196, 198, 200, 202, 205-207, 212, 214, 216, 218, 220, 221, 223, 228, 230, 232, 234, 237, 239, 244, 246, 248, 250, 253, 255
239	0, 1, 4, 5, 8, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28, 29, 64, 65, 68, 69, 72, 73, 76, 77, 80, 81, 84, 85, 88, 89, 92, 93, 130, 131, 134, 135, 138, 139, 142, 143, 146, 147, 150, 151, 154, 155, 158-161, 164, 165, 168, 169, 172, 173, 176, 177, 180, 181, 184, 185, 188, 189, 194, 195, 198, 199, 202, 203, 206, 207, 210, 211, 214, 215, 218, 219, 222-225, 228, 229, 232, 233, 236, 237, 240, 241, 244, 245, 248, 249, 252, 253
253	0-15, 64-79, 144-160, 162-175, 208-239
255	128-159, 192-223

Table 8. Hybridization rules for synthesizing nonuniform TACAs from uniform SACAs.

5.4 TACA to TACA

The following property of CA rules dictates the synthesis of hybrid TACAs from uniform TACAs.

Theorem 7. The R_h (hybridization rule) that gets absorbed into a uniform TACA with rule R_T is (a): greater than 127; (b): even (T(0) is 0); (c): not from class 0 or class 8.

Proof. Uniform TACAs can only have all-0 and all-1 attractors (Theorem 2). For all-0 (resp. all-1) attractors, T(0) (resp. T(7)) must be passive. As R_h is absorbed, its T(7) is also passive, that is, T(7) = 1. That is, R_h is greater than 127 (condition (a)). Similarly, T(0) of R_h is 0. Therefore, R_h is even (condition (b)). A rule that belongs to class 0 does not have a passive RMT. On the other hand, for a rule that belongs to class 8, all eight RMTs are passive. Such a rule introduces additional sequences of RMTs and with self-loops in the NSRTD, thus it does not restrict the number of fixed points to two. Hence, if the R_h is from class 0 or class 8, then the hybridized CA no longer remains a TACA (condition (c)). ■

Observation 4. Nonuniform TACAs synthesized from uniform TACAs through hybridization can have at most one fixed-point attractor that is different from the attractors of the uniform TACA.

The fixed-point attractors of the four-cell uniform TACA with rule 192 (0000 and 1111) are shown in Figure 20(a). The nonuniform TACA resulting from the hybridization by rule 245 has 0010 and 1111 fixed-point attractors (Figure 20(b)). Table 9 includes the rules that form nonuniform TACAs with a different fixed point set than that of the uniform TACAs. On the other hand, Table 10 reports on the hybridization rules that are absorbed in a uniform TACA rule.

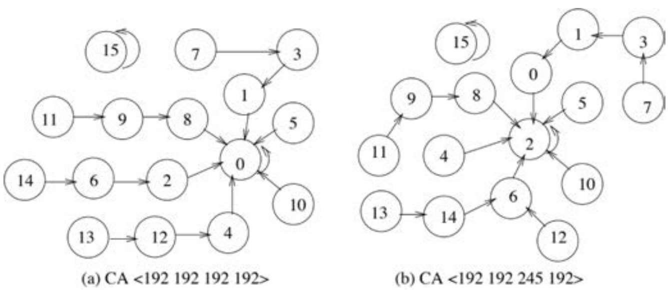


Figure 20. State space of a uniform TACA with rule 192 and the nonuniform CA with rules 192 and 245.

TACA Rule	Hybridization Rules
128	4, 6, 12, 14, 20, 22, 28, 30, 36, 38, 44, 46, 52, 54, 60, 62, 68, 70, 76, 78, 84, 86, 92, 94, 100, 102, 108, 110, 116, 118, 124, 126, 133, 135, 141, 143, 149, 151, 157, 159, 165, 167, 173, 175, 181, 183, 189, 191, 197, 199, 205, 207, 213, 215, 221, 223, 229, 231, 237, 239, 245, 247, 253, 255
136	4, 6, 12, 14, 20, 22, 28, 30, 36, 38, 44, 46, 52, 54, 60, 62, 133, 135, 141, 143, 149, 151, 157, 159, 165, 167, 173, 175, 181, 183, 189, 191
192	4, 6, 20, 22, 36, 38, 52, 54, 68, 70, 84, 86, 100, 102, 116, 118, 133, 135, 149, 151, 165, 167, 181, 183, 197, 199, 213, 215, 229, 231, 245, 247
238	2, 6, 10, 14, 18, 22, 26, 30, 66, 70, 74, 78, 82, 86, 90, 94, 131, 135, 139, 143, 147, 151, 155, 159, 195, 199, 203, 207, 211, 215, 219, 223
252	16, 18, 20, 22, 24, 26, 28, 30, 80, 82, 84, 86, 88, 90, 92, 94, 145, 147, 149, 151, 153, 155, 157, 159, 209, 211, 213, 215, 217, 219, 221, 223
254	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 151, 153, 155, 157, 159, 193, 195, 197, 199, 201, 203, 205, 207, 209, 211, 213, 215, 217, 219, 221, 223

Table 9. Hybridization rules for nonuniform TACAs with different fixed-point sets.

TACA Rule	Hybridization Rules
128	130, 136, 138, 144, 146, 152, 154, 160, 162, 168, 170, 176, 178, 184, 186, 192, 194, 200, 202, 208, 210, 216, 218, 224, 226, 232, 234, 240, 242, 248, 250
136	128, 130, 138, 144, 146, 152, 154, 160, 162, 168, 170, 176, 178, 184, 186
192	128, 130, 144, 146, 160, 162, 176, 178, 194, 208, 210, 224, 226, 240, 242
238	162, 166, 170, 174, 178, 182, 186, 190, 226, 230, 234, 242, 246, 250, 254
252	176, 178, 180, 182, 184, 186, 188, 190, 240, 242, 244, 246, 248, 250, 254
254	160, 162, 164, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 224, 226, 228, 230, 232, 234, 236, 238, 240, 242, 244, 246, 248, 250, 252

Table 10. Hybridization rules that get absorbed in TACA rules.

6. Conclusion

A common theoretical framework for characterizing fixed-point attractor periodic boundary cellular automata (PBCAs) as well as synthesis of PBCAs with specified fixed points is currently under effort and to be formalized. This paper explores the properties of fixed-point attractors in a PBCA. Two classes of irreversible CAs, called the single length cycle single-attractor CA (SACA) and the single length cycle two-attractor CA (TACA), have been introduced. A tool, referred to as the NSRT diagram (NSRTD), is developed to provide the analytical foundation for synthesizing fixed-point CAs having a single attractor (SACA) and two attractors (TACA). Further, the NSRTD tool has been extensively used to identify the hybridization rules that can result in the splitting or merging of attractor basins. As the nonuniform CAs can perform complex tasks and the fixed-point attractor CAs are found to be an effective model for inherent irreversibility present within a physical system, such a characterization is of the utmost necessity. The NSRTD framework can also assist in determining the multi-length cycle attractors in a CA state space.

References

[1] J. von Neumann, *Theory of Self-Reproducing Automata* (A. W. Burks, ed.), Urbana, IL: University of Illinois Press, 1966.

[2] S. Wolfram, "Universality and Complexity in Cellular Automata," *Physica D: Nonlinear Phenomena*, **10**(1–2), January 1984 pp. 1–35. doi:10.1016/0167-2789(84)90245-8.

[3] P. P. Chaudhuri, D. R. Chowdhury, S. Nandi and S. Chatterjee, *Additive Cellular Automata: Theory and Applications*, Vol. 1, Los Alamitos, CA: IEEE Computer Society Press, 1997.

[4] E. F. Moore, *Machine Models of Self-Reproduction*, Urbana, IL: University of Illinois Press, 1962.

[5] J. W. Thatcher, "Universality in von Neumann Cellular Model," *Technical Report 03105-30-T, ORA*, University of Michigan, 1964.

[6] A. Wuensche and M. Lesser, *The Global Dynamics of Cellular Automata: An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata*, Reading, MA: Addison-Wesley, 1992.

[7] P. Dasgupta, S. Chattopadhyay and I. Sengupta, "An ASIC for Cellular Automata Based Message Authentication," in *VLSI Design 2000, Wireless and Digital Imaging in the Millennium, Proceedings of 13th International Conference on VLSI Design*, Calcutta, India, Los Alamitos, CA: IEEE Computer Society, 2000 pp. 538–541.
doi:10.1109/ICVD.2000.812663.

[8] H. Gutowitz, "Cryptography with Dynamical Systems," in *Cellular Automata and Cooperative Phenomena* (N. Boccara, E. Goles, S. Martinez and P. Picco, eds.), Dordrecht: Springer, 1993 pp. 237–274. doi:10.1007/978-94-011-1691-6_21.

[9] M. Mukherjee, N. Ganguly and P. P. Chaudhuri, "Cellular Automata Based Authentication (CAA)," *Cellular Automata (ACRI 2002)*, Geneva, Switzerland (S. Bandini, B. Chopard and M. Tomassini, eds.), Berlin, Heidelberg: Springer, 2002 pp. 259–269. doi:10.1007/3-540-45830-1_25.

[10] B. Das, N. S. Maiti, S. Das and B. K. Sikdar, "An Efficient Scheme for Data Block Migration in Tiled CMPS Cache System," in *2013 Annual IEEE India Conference (INDICON)*, Mumbai, India, Piscataway, NJ: IEEE, 2013. doi:10.1109/INDCON.2013.6725990.

[11] S. Wolfram, *Cellular Automata and Complexity: Collected Papers*, Reading, MA: Addison-Wesley Publishing Company, 1994.

[12] A. Dennunzio, E. Formenti, D. Grinberg and L. Margara, "Decidable Characterizations of Dynamical Properties for Additive Cellular Automata over a Finite Abelian Group with Applications to Data Encryption," *Information Sciences*, **563**, 2021 pp. 183–195. doi:10.1016/j.ins.2021.02.012.

[13] A. Dennunzio, E. Formenti, D. Grinberg and L. Margara, "An Efficiently Computable Characterization of Stability and Instability for Linear Cellular Automata," *Journal of Computer and System Sciences*, **122**, 2021 pp. 63–71. doi:10.1016/j.jcss.2021.06.001.

[14] S. Das, S. Mukherjee, N. Naskar and B. K. Sikdar, "Characterization of Single Cycle CA and Its Application in Pattern Classification," *Electronic Notes in Theoretical Computer Science*, **252**, 2009 pp. 181–203. doi:10.1016/j.entcs.2009.09.021.

[15] N. Ganguly, A. Das, P. Maji, B. K. Sikdar and P. P. Chaudhuri, "Evolving Cellular Automata Based Associative Memory for Pattern Recognition," in *High Performance Computing—HiPC 2001*, Hyderabad, India (B. Monien, V. K. Prasanna and S. Vajapeyam, eds.), Berlin, Heidelberg: Springer, 2001 pp. 115–124. doi:10.1007/3-540-45307-5_11.

[16] M. Dalui, B. Chakraborty, N. Das and B. K. Sikdar, "NSRT Diagram for Identification of SACA and TACA Rules in Null-Boundary," *International Journal of Modern Physics C*, **33**(6), 2022 2250071. doi:10.1142/S0129183122500711.

[17] N. Ganguly, "Cellular Automata Evolution: Theory and Applications in Pattern Recognition and Classification," Ph.D. thesis, Durgapur, West Bengal, India, Bengal College of Engineering and Technology (DU), 2003.
scholar.google.co.in/scholar?q=%E2%80%9Ccellular+automata+evolution:+theory+and+applications+in+pattern+recognition+and+classification,+niloy+ganguly&hl=en&as_sdt=0&as_vis=1&oi=scholart#d=gs_qabs&t=1717120978686&u=%23p%3DvESUf1WBOQYJ.

[18] S. Munshi, N. S. Maiti, D. Ray, D. R. Chowdhury and P. P. Chaudhuri, "An Analytical Framework for Characterizing Restricted Two Dimensional Cellular Automata Evolution," *Journal of Cellular Automata*, **3**(4), 2008 pp. 313–335.
www.oldcitypublishing.com/journals/jca-home/jca-issue-contents/jca-volume-3-number-4-2008/jca-3-4-p-313-335.

[19] K. Sutner, "De Bruijin Graphs and Linear Cellular Automata," *Complex Systems*, **5**(1), 1991 pp. 19–30.
complex-systems.com/pdf/05-1-3.pdf.

[20] S. Munshi, S. Das and B. K. Sikdar, "Characterization of Single Hybridization in 'Non-interesting' Class of Cellular Automata for SMACA Synthesis," in *Proceedings of AUTOMATA 2010: 16th International Workshop on Cellular Automata and Discrete Complex Systems*, Nancy, France (N. A. Fatès, J. Kari and T. Worsch, eds.), HAL Open Science, 2010 pp. 265–276.
inria.hal.science/inria-00549645/document.

[21] S. Kamilya, S. Adak, S. Das and B. K. Sikdar, "SACAS: (Non-uniform) Cellular Automata That Converge to a Single Fixed Point," *Journal of Cellular Automata*, **14**(1–2), April 2019 pp. 27–49.
www.oldcitypublishing.com/journals/jca-home/jca-issue-contents/jca-volume-14-number-1-2-2019/jca-14-1-2-p-27-49.

[22] P. Maji, C. Shaw, N. Ganguly, B. K. Sikdar and P. P. Chaudhuri, "Theory and Application of Cellular Automata for Pattern Classification," *Fundamenta Informaticae*, **58**(3–4), 2003 pp. 321–35.
dl.acm.org/doi/10.5555/2371016.2371023.

[23] S. Das and B. K. Sikdar, "Characterization of 1-D Periodic Boundary Reversible CA," in *Electronic Notes in Theoretical Computer Science*, **252**, 2009 pp. 205–227. doi:10.1016/j.entcs.2009.09.022.

[24] B. Das, M. Saha, N. Das and B. K. Sikdar, "Identification of Periodic Boundary SACA Rules Exploring NSRT Diagram," in *Cellular Automata: 15th International Conference on Cellular Automata for Research and Industry (ACRI 2022)*, Geneva, Switzerland (B. Chopard, S. Bandini, A. Dennunzio and M. A. Haddad, eds.), Cham: Springer, 2022, pp. 29–39. doi:10.1007/978-3-031-14926-9_3.