Search of Complex Binary Cellular Automata Using Behavioral Metrics

Juan C. López-González
Wolfram Research, Inc.
100 Trade Center Drive
Champaign, IL 61820
and
Physics and Mathematics in Biomedicine Consortium
Universidad Central de Venezuela
jlopez@cellular-automata.com

Antonio Rueda-Toicen
Instituto Nacional de Bioingeniería
Universidad Central de Venezuela
Av. Miguel Otero Silva, Urb. Sebucán
Caracas, Venezuela 1071
and
Physics and Mathematics in Biomedicine Consortium
Universidad Central de Venezuela
antonio.rueda@ciens.ucv.ve

We propose the characterization of binary cellular automata using a set of behavioral metrics that are applied to the minimal Boolean form of a cellular automaton's transition function. These behavioral metrics are formulated to satisfy heuristic criteria derived from elementary cellular automata. Behaviors characterized through these metrics are growth, decrease, chaoticity, and stability. From these metrics, two measures of global behavior are calculated: (1) a static measure that considers all possible input patterns and counts the occurrence of the proposed metrics in the truth table of the minimal Boolean form of the automaton; and (2) a dynamic measure, corresponding to the mean of the behavioral metrics in n executions of the automaton, starting from n random initial states. We use these measures to characterize a cellular automaton and guide a genetic search algorithm, which selects cellular automata similar to the Game of Life. Using this method, we found an extensive set of complex binary cellular automata with interesting properties, including self-replication.

1. Introduction

Cellular automata with complex behavior exhibit dynamical patterns that can be interpreted as the movement of particles through a physical medium. These particles are interpretable as loci for information

https://doi.org/10.25088/ComplexSystems.24.1.1

storage, and their movement through space is interpretable as information transfer. The collisions of these particles in the cellular automaton's lattice are sites of information processing [1–4]. Cellular automata with complex behavior have immense potential to describe physical systems, and their study has had impact in the design of self-assembling structures [5–8] and the modeling of biological processes like signaling, division, apoptosis, necrosis, and differentiation [9–13]. John Conway's Game of Life [14] is the most renowned complex binary cellular automaton and the archetype used to guide the search methodology for other complex binary cellular automata that we describe in this work. Previously, complex behavior in binary cellular automata has been characterized through measures such as entropy [3], Lyapunov exponents [15, 16], and Kolmogorov–Chaitin complexity [17]. We propose the characterization of the behavior of n-dimensional cellular automata through heuristic measures derived from the evaluation of their minimal Boolean forms. This proposed characterization is derived from heuristic criteria validated in elementary cellular automata with simple Boolean forms. Table 1 illustrates the rationale for this characterization, showing elementary cellular automata whose Boolean forms are minimally simple, and whose behavior can be unequivocally identified. Cellular behaviors of growth, decrease,

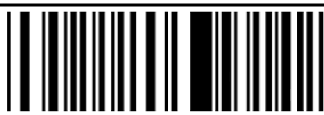
Rule	Sample Evolution	Boolean Form	Behavior
R_{204}		q	Stable
R_{160}		p AND r	Decreasing
R_{252}		p OR q	Growing
R_{90}		p XOR q	Chaotic

Table 1. Elementary cellular automata with simple Boolean forms, which are unequivocally associated to a particular behavior. The Boolean values of the cells in the neighborhood are p for the left neighbor, q for the central cell, and r for the right neighbor. Black cells are in 1 state, white cells are in 0 state.

https://doi.org/10.25088/ComplexSystems.24.1.1

and chaoticity are characterized by the Boolean operations *OR*, *AND*, and *XOR*, respectively. The cellular behavior of stability can be characterized by the absence of a Boolean operator or the use of the *NOT* operator.

We define an evaluation criterion to produce metrics that characterize the behavior of cellular automata whose minimal Boolean expressions are more complex (i.e., have more terms and the combination of various operators) than those appearing in Table 1. The produced metrics are used to create static and dynamic measures of behavior. The static measure of behavior is calculated from the truth table of the minimal Boolean expression of the cellular automaton, and the dynamic measure of behavior is derived from the averaged appearance of the metrics in n executions of the cellular automaton from n random initial conditions. We use the Euclidean distance of these measures in a given cellular automaton to the measures of the Game of Life to assess its capacity for complex behavior and also use this distance as a cost function to guide the genetic search of n-dimensional cellular automata with complex behavior.

2. Definition of Binary Cellular Automaton

A cellular automaton is formally represented by a quadruple $\{Z, S, N, f\}$, where:

- Z is the finite or infinite cell lattice
- S is a finite set of states or values for the cells
- N is the finite cell neighborhood
- f is the local transition function, defined by the state transition rule

Each cell in the lattice Z is defined by its discrete position (an integer number for each dimension) and by its discrete state value S. In a binary cellular automaton, $S = \{0, 1\}$. Time is also discrete. The state of the cell is determined by the evaluation of the local transition function on the cell's neighborhood at time t; $t+1$ is the next time step after time t. The neighborhood is defined as a finite group of cells surrounding and/or including the observed cell.

2.1 Lattice, Cell, and Configuration

The global state is the configuration of all the cells that comprise the automaton, $C \in S^Z$. The lattice Z is the infinite cyclic group of integers $\{\ldots, -1, 0, 1, 2, \ldots\}$. The position of each cell in the lattice is described by the index position $x \in Z$. Configurations are commonly

https://doi.org/10.25088/ComplexSystems.24.1.1

written as sequences of characters, such as

$$C = \dots c_{-1}\, c_0\, c_1\, c_2 \dots. \tag{1}$$

The finite global state is a finite configuration $C \in S^Z$, where Z is a finite lattice, indexed with 0, 1, 2, 3 … $n-1$ integers,

$$C = c_1\, c_2 \dots c_x\, c_{x+1} \cdots c_{n-2}\, c_{n-1}. \tag{2}$$

2.2 Neighborhood and Local Transition Function

The set of neighborhood indices A of size $m = |A|$ is defined by the set of relative positions within the configuration, such that

$$A = a_0, a_1, \dots, a_{m-2}, a_{m-1}. \tag{3}$$

N_x is the neighborhood of the observed cell c_x that includes the set A of indices, and is defined as

$$N_x = c_{x+a_0}\, c_{x+a_1} \cdots c_{x+a_{m-2}}\, c_x + a_{m-1}. \tag{4}$$

This describes the neighborhood as a character string that includes the cells that are considered neighbors of the observed cell x. A compact representation of the neighborhood value N_x is a unique integer, defined as an m-digits, k-based number [2]

$$N_x = \sum_{i=0}^{m-1} k^{m-1-i}\, c_{x+a_i} = c_{x+a_0}\, k^{m-1} + \cdots + c_{x+a_{m-1}}\, k\, k^0. \tag{5}$$

The local transition function f that yields the value of c_x at $t+1$ from the neighborhood of the cell observed at present time t is expressed by

$$f\left(N_x^t\right) = c_x^{t+1}, \tag{6}$$

where N_x^t specifies the states of the neighboring cells to the cell x at time t. The transition table defines the local transition function, listing an output value for each input configuration. Table 2 is a sample transition table for an elementary cellular automaton with a neighborhood of radius 1, wherein adjacent neighboring cells of c_x are c_{x-1} and c_{x+1}, forming a tuple $\{c_{x-1}, c_x, c_{x+1}\}$, $S \in \{0, 1\}$.

2.3 Global Transition Function

The global dynamics of the cellular automaton are described by the global transition function F:

$$F : S^N \to S^N. \tag{7}$$

https://doi.org/10.25088/ComplexSystems.24.1.1

N_x^t	$f(N_x^t)$
000	0
001	1
010	1
011	1
100	1
101	0
110	1
111	0

Table 2. Local transition function of R_{94} as a truth table.

F is the transition between the current global configuration C^t and the next global configuration C^{t+1}:

$$C^{t+1} = F\left(C^t\right). \tag{8}$$

The global transition function F is defined by the local transition function f as

$$F\left(C_x\right) = \ldots f(N_{x-1})\, f(N_x)\, f(N_{x+1}) \ldots. \tag{9}$$

3. Transformation of the Cellular Space

We redefine the local transition function to incorporate behavioral knowledge of the automaton's evolution, given an input/output pair. This redefined function is applied to all cells of the automaton at a given evolution step t to quantify its overall behavior.

3.1 Redefined Local Transition Function

The redefined local transition function g calculates the behavioral metric of a single cell c_x, evaluating the local transition function f on its neighborhood N_x^t. Through the local transition function g, we define the transformation d_x^{t+1} that yields the next step of the evolution of cell c_x as

$$d_x^{t+1} = g\left(f, N_x^t\right). \tag{10}$$

This transformation is necessary to calculate the measure of dynamic behavior during the automaton's evolution, and we propose the inclusion of a metric characterizing the cell behavior obtained af-

https://doi.org/10.25088/ComplexSystems.24.1.1

ter evaluating a particular input. Input for the Boolean operators considered may be of the form

$$Input_1 \langle \, operator \, \rangle \, Input_2 = Output, \tag{11}$$

where $\langle \, operator \, \rangle \in \{OR, AND, XOR\}$. The behaviors associated with each binary Boolean operator and its possible inputs and outputs are shown in Table 3.

$Input_1$	$Input_2$	$Output$	Behavior	OR	AND	XOR
0	0	0	Stability	X	X	
1	0	0	Decrease		X	
0	1	0	Decrease		X	
1	1	0	Chaoticity			X
0	0	1	Chaoticity			X
1	0	1	Growth	X		
0	1	1	Growth			X
1	1	1	Stability	X	X	

Table 3. Behaviors associated to binary Boolean patterns.

The behaviors associated with unary patterns are shown in Table 4:

$$\langle \, operator \, \rangle \, Input \; = \; Output, \tag{12}$$

where $\langle \, operator \, \rangle \in \{NOT, \; NOP\}$ and NOP stands for "no operator." To characterize the automaton's behavior, we expand the state space

$$g : \left\{S^N, f\right\} \to M, \tag{13}$$

where

$$M = \{0, 1, 2, 3, 4, 5\}. \tag{14}$$

The different values of M abbreviate the duples of state and behavior shown in Table 5. Each tuple is obtained from the result of the local transition function g applied to a particular configuration of the cell x and its neighborhood N.

The M code eases the implementation of an algorithmic search for cellular automata with interesting behavior, using the proposed metrics. According to the M code, chaotic and stable behaviors may generate 1 or 0 as output from 1 or 0 as input, growing behavior may only generate 1 as output from 0 as input, and decreasing behavior may only generate 0 as output from 1 as input.

https://doi.org/10.25088/ComplexSystems.24.1.1

Input	*Output*	Behavior	*NOT*	*NOP*
1	1	Stability		X
0	0	Stability		X
1	0	Stability	X	
0	1	Stability	X	

Table 4. Behavior associated to unary Boolean patterns.

M	$\{S_x^{t+1}, behavior\}$
0	{0, *stable*}
1	{0, *decreasing*}
2	{0, *chaotic*}
3	{1, *chaotic*}
4	{1, *growing*}
5	{1, *stable*}

Table 5. M code, abbreviation of duples of cell state and behavior obtained when applying the local transition function g.

3.2 Global Transition Function

The global behavioral metric of the cellular automaton is characterized as

$$G : \{S^N, f\} \to M^N. \tag{15}$$

G represents the transition between the current global configuration C^t and the next global configuration C^{t+1}. We set $D^0 = (0, f)$ and express the automaton's global behavioral metric as

$$D^{t+1} = G(C^t, f), \tag{16}$$

for example, from the initial state,

$$\begin{gathered} C^0 \ (\textit{initial state}) \\ C^1 = F(C^0) \to D^1 = G(C^0, f) \\ C^2 = F(C^1) \to D^2 = G(C^1, f) \\ C^3 = F(C^2) \to D^3 = G(C^2, f) \\ C^4 = F(C^3) \to D^4 = G(C^3, f) \\ \vdots \end{gathered}$$

https://doi.org/10.25088/ComplexSystems.24.1.1

The redefined global transition function G is expressed as the concatenated string obtained when the redefined local transition function g is applied to all of the automaton's cells c_i:

$$G(\dots c_{x-1}\, c_x\, c_{x+1}, f) = \\ \dots g(n_{x-1}, f)\, g(n_x, f)\, g(n_{x+1}, f) \dots. \tag{17}$$

3.3 Implementation of $g(f, N_x^t)$

The g function incorporates heuristic information that enables the measurement of behaviors in the automaton's lattice. The g function performs the following steps, given a pattern N_x^t and the transition function f:

1. The local transition function f is simplified to its minimal Boolean expression.
2. f is expressed as a binary execution tree.
3. N_x^t is evaluated on the binary execution tree obtained in 2.

In Table 1 we mentioned the behavioral characterization corresponding to cellular automata whose minimal expression corresponds to a single Boolean operator. This characterization needs to be extended to describe cellular automata whose minimal forms have several distinct Boolean operators. To tackle this problem, we express a cellular automaton's transition function in a binary evaluation tree and propose a set of evaluation rules for its nodes, based on heuristic criteria.

We write the transition function of the minimal expression of the automaton's rule in a tree graph. We assign to each node of the tree a Boolean operation. The transition function is evaluated, with input placed at the tree's leaves, according to heuristic rules. The result of the evaluation is obtained at the root node. The heuristics considered are crafted to fit criteria derived from the characteristic behaviors of several elementary cellular automata.

The proposed heuristic H consists of rules for evaluation of the nodes in the binary tree. These tree evaluation rules are defined for

$$term\, \langle\, OPERATOR \,\rangle\, term$$

and

$$\langle\, OPERATOR \,\rangle\, term,$$

where $\langle\, OPERATOR \,\rangle \in \{AND,\ OR,\ XOR,\ NOT\}$ and *term* corresponds to the set $M = \{0, 1, 2, 3, 4, 5\}$.

Figure 1 shows the heuristic precedence rules defined for each logical operator. Figure 2 shows the elementary cellular automata used to

https://doi.org/10.25088/ComplexSystems.24.1.1

define the heuristic characterization criteria, alongside their minimal Boolean forms.

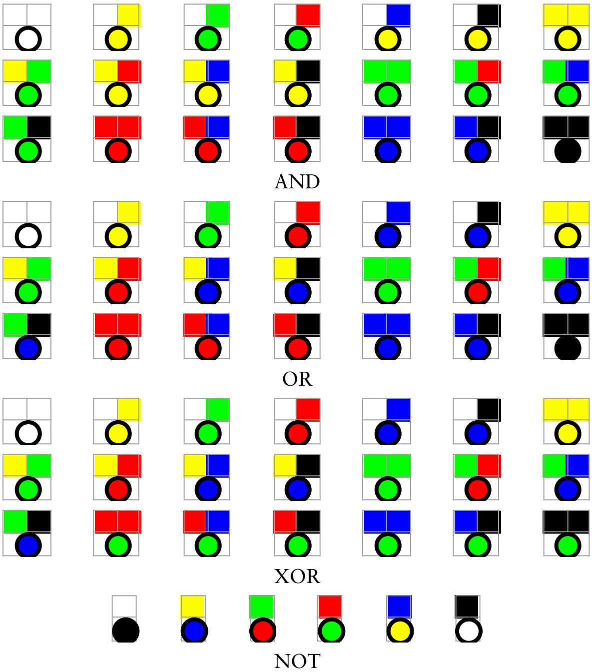


Figure 1. Tree evaluation rules H: squares correspond to inputs and circles to outputs. White corresponds to $M = 0 = \{0, \textit{stable}\}$; yellow corresponds to $M = 1 = \{0, \textit{decrease}\}$; green corresponds to $M = 2 = \{0, \textit{chaotic}\}$; red corresponds to $M = 3 = \{1, \textit{chaotic}\}$; blue corresponds to $M = 4 = \{1, \textit{growth}\}$; black corresponds to $M = 5 = \{1, \textit{stable}\}$.

- *Criterion 1.* In the leaf nodes, $S = 0$ must be equivalent to $M = 0 = \{0, \textit{stable}\}$ and $S = 1$ must be equivalent to $M = 5 = \{1, \textit{stable}\}$.
- *Criterion 2.* Chaoticity measured in $R_{150} = p\ XOR\ q\ XOR\ r$ must be greater than chaoticity measured in $R_{90} = p\ XOR\ r$. The proposed heuristic H produces the following behavioral metrics in these automata:

  $$R_{150}\ \textit{chaoticity} = 0.375$$
  $$R_{90}\ \textit{chaoticity} = 0.25.$$

- *Criterion 3.* Chaoticity measured in $R_{90} = p\ XOR\ r$ must be greater than chaoticity measured in $R_{204} = q$. The proposed heuristic H produces the following behavioral metrics in these automata:

  $$R_{90}\ \textit{chaoticity} = 0.25$$
  $$R_{204}\ \textit{chaoticity} = 0.$$

https://doi.org/10.25088/ComplexSystems.24.1.1

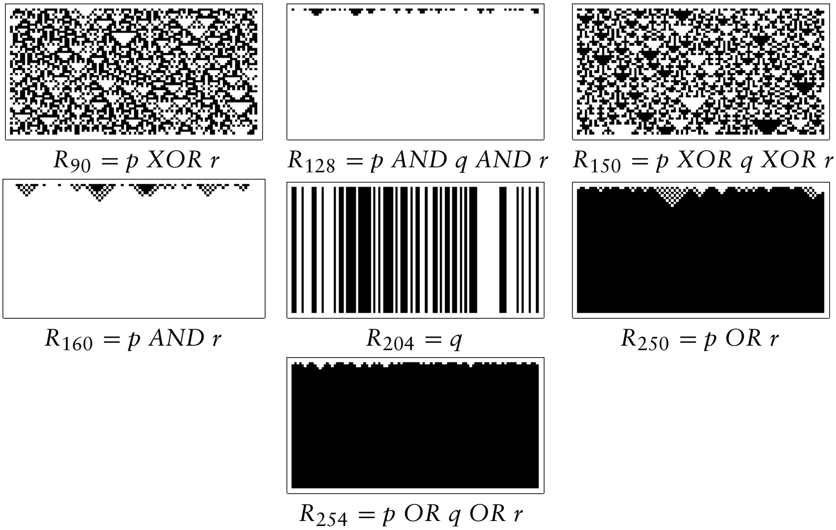


Figure 2. Elementary cellular automata used to define the criteria in H.

- *Criterion 4*. Decrease measured in $R_{128} = p\ AND\ q\ AND\ r$ must be greater than decrease measured in $R_{160} = p\ AND\ r$. The proposed heuristic H produces the following behavioral metrics in these automata:

  $R_{128}\ decrease = 0.75$

  $R_{160}\ decrease = 0.5.$

- *Criterion 5*. Decrease measured in $R_{128} = p\ AND\ q\ AND\ r$ must be greater than decrease measured in $R_{160} = p\ AND\ r$. The proposed heuristic H produces the following behavioral metrics in these automata:

  $R_{160}\ decrease = 0.5$

  $R_{204}\ decrease = 0.$

- *Criterion 6*. Growth measured in $R_{254} = p\ OR\ q\ OR\ r$ must be greater than growth measured in $R_{250} = p\ OR\ r$. The proposed heuristic H produces the following behavioral metrics in these automata:

  $R_{254}\ growth = 0.75$

  $R_{250}\ growth = 0.5.$

Figure 3 shows the percentage of measured behaviors, using the proposed set of evaluation rules H, in the elementary cellular automata considered in the criteria.

https://doi.org/10.25088/ComplexSystems.24.1.1

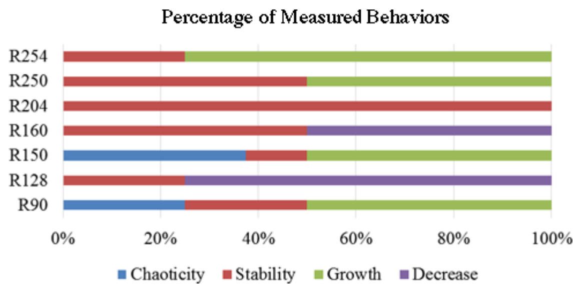


Figure 3. Behavioral percentages in elementary cellular automata considered as criteria for evaluating the proposed heuristic.

3.4 Evaluation Example with R_{94}

The minimal Boolean expression of R_{94}, $f = (q\ AND\ (NOT\ p))$ OR $(p$ XOR $r)$ is placed in a binary evaluation tree, as shown in Figure 4. Each node in the tree is evaluated using the rules shown in Figure 1. This process is demonstrated in Steps 1–5.

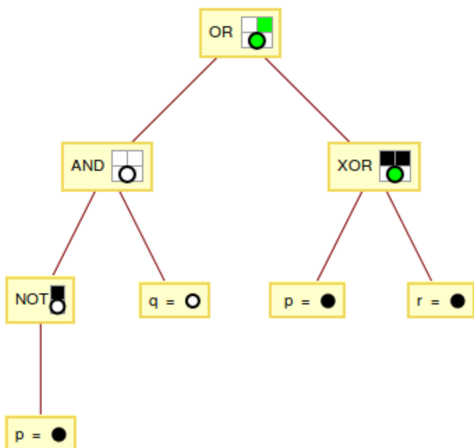


Figure 4. Evaluation of the input pattern 101 in R_{94} with the proposed rules.

- *Step 1*. In the leaf nodes, the values in the neighborhood $N_q^{t=0}$ are $\{p = 1, q = 0, r = 1\}$; these are transformed using the M code mentioned in Section 3.1, as follows:

  $S_M(p) = \{1, stable\} = 5$
  $S_M(q) = \{0, stable\} = 0$
  $S_M(r) = \{1, stable\} = 5.$

Thus, the input tuple $\{p = 1, q = 0, r = 1\}$ is converted into $\{p = 5,\ q = 0,\ r = 5\}$.

- *Step 2*. Leaf $p = 5$ is evaluated at the *NOT* node, producing output $0 = \{0,$ *stable*$\}$.
- *Step 3*. Leaf $q = 0$ and the result of step 2 are evaluated at the *AND* node, producing output $0 = \{0,$ *stable*$\}$.
- *Step 4*. Leaves $p = 5$ and $r = 5$ are evaluated at the *XOR* node, producing output $2 = \{0,$ *chaotic*$\}$.
- *Step 5*. The output of step 4 and the output of step 5 are evaluated at the *OR* node, producing as final output $2 = \{0,$ *chaotic*$\}$. The cell q gets assigned to state 0 in $t + 1$, and a counter for the occurrence of chaotic behavior in the states of R_{94} would get incremented by one.

4. Behavioral Characterization

To characterize the overall behavior of a cellular automaton with the proposed metrics, we consider the correlation between two measures:

1. A static measure, which is the counted occurrence of behaviors associated to the code M, in the output of the truth table of the minimal Boolean expression of the cellular automaton.
2. A dynamic measure, which is the median occurrence of behaviors associated to the code M in n executions of the cellular automaton, starting from n random initial states.

4.1 Static Measure of Behavior

The local transition function transition f is expressed as a truth table, which is converted to g when we include behavioral information. To calculate the static measure of behavior, we count the occurrence of behaviors associated with the values of M in the output of the truth table. This static measure is a vector, with the percentages of chaoticity, stability, growth, and decrease measured in the cellular automaton.

For example, in R_{94} the rule is characterized using the M code as shown in Table 6.

To obtain the static measure of R_{94}, we count the occurrences of M. The static measure of the rule is the percentage of behavioral occurrence in the automaton, as shown in Table 7.

N_x^t	$f(N_x^t)$	$g(N_x, f)$
000	0	$M = 1$
001	1	$M = 4$
010	1	$M = 4$
011	1	$M = 4$
100	1	$M = 4$
101	0	$M = 2$
110	1	$M = 4$
111	0	$M = 2$

Table 6. Truth table of R_{94}, with associated M code.

Stability	Decrease	Growth	Chaoticity
$M = \{0, 5\}$	$M = 1$	$M = 4$	$M = \{2, 3\}$
0%	12.5%	62.5%	25%

Table 7. Behavioral percentages in R_{94}, static measure.

We express this measure as a vector of percentages:

$$M_E = \{\textit{stability}\ \%, \textit{decrease}\ \%, \textit{growth}\ \%, \textit{chaoticity}\ \%\}. \tag{18}$$

For R_{94}, the static measure of behavior is

$$M_E = \{0, 12.5, 62.5, 25\}.$$

4.2 Dynamic Measure of Behavior

To estimate the dynamic measure of behavior M_D, we execute the cellular automaton n times, from n random initial configurations $C_i^{t=0} \,\big|\, i \in n$. We sample occurrences of M in the cell space up to the k^{th} evolution step, where k is an integer > 0, obtained from a uniform distribution:

$$M_D(g) = \lim_{x \to \infty} \frac{\left(M_D^{t=k}\left(g, c_i^{t=0}\right)\right)}{n}. \tag{19}$$

We exclude cells at $t = 0$ from the sampling. The percentages of behavioral occurrences are calculated from the mean of samples. Figure 5 shows the sampling of R_{94} in $k = t = 20$.

https://doi.org/10.25088/ComplexSystems.24.1.1

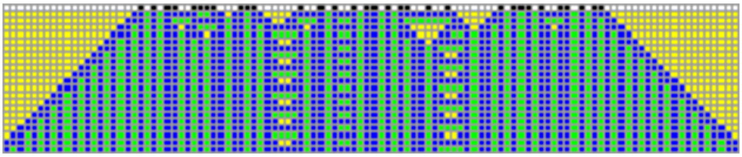


Figure 5. Evolution of R_{94} from a random initial configuration: yellow coloring for $M = 1$, green coloring for $M = 2$, and blue coloring for $M = 4$. The code M was applied to cells in $t \geq 1$. Cells with $M = 1$ (decreasing behavior) occupy 18.658% of the lattice, cells with $M = 2$ (stable behavior) occupy 32.467%, and cells with $M = 4$ (chaotic behavior) occupy 48.874%.

5. Analysis of the Game of Life

The Game of Life is a complex cellular automaton, class IV according to the classification proposed by Wolfram [1, 3]. In this cellular automaton, there is a negative correlation between the static measure of behavior and the dynamic measure of behavior. Table 8 shows this negative correlation.

Some observations pertinent to the measured behavior in the Game of Life:

- *Static measure*. Chaotic behavior predominates, an important characteristic of class III automata.
- *Dynamic measure*. Decreasing behavior predominates, an important characteristic of class I automata.

Looking at the transition function f of the Game of Life, we can find patterns such as

$$\begin{gathered}(\ldots)\\ NOT\ x0\ AND\ NOT\ x1\ AND\ NOT\ x2\ AND\ x8\\ AND\ (x3\ XOR\ x4)\ AND\ (x5\ XOR\ x6)\ OR\\ NOT\ x0\ AND\ NOT\ x1\ AND\ NOT\ x3\ AND\ x8\\ AND\ (x2\ XOR\ x4)\ AND\ (x5\ XOR\ x6)\ OR\\ (\ldots)\end{gathered}$$

It is our hypothesis that the emergence of complex behavior in the Game of Life is determined by the appearance of islands of chaotic behavior surrounded by decreasing patterns. Taking a close look at the Boolean expression of f in the Game of Life, chaotic subexpressions can be observed like (*x3 XOR x4*) being "restricted" with *AND* by decreasing subexpressions such as (*AND NOT x2 AND x8*).

https://doi.org/10.25088/ComplexSystems.24.1.1

	M_E	M_D
Chaoticity	67.96	13.38
Decrease	4.68	75.23
Growth	27.34	11.37
Stability	0	0

Table 8. Static and dynamic measures in the Game of Life; their correlation is -0.29.

In Figure 6, yellow cells have value $M = 1$ (decreasing behavior), and blue cells have value $M = 4$ (growth behavior). Green cells have $M = 2$, exhibiting chaotic behavior. Note that in Figure 6, decreasing cells ($M = 1$) cover the largest proportion of the lattice, which corresponds with the dynamic measure of decrease shown in Table 8. It can also be seen that the isolated patterns exhibit a combination of growth ($M = 4$) and chaoticity ($M = 2$).

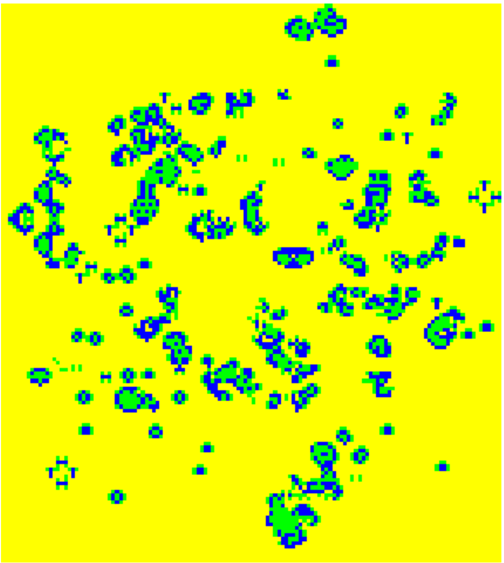


Figure 6. The Game of Life, colored according to behavior.

6. Search of Complex Binary Cellular Automata in Two Dimensions

The proposed behavioral metrics were crafted using heuristic criteria from one-dimensional binary cellular automata, yet are applicable to characterize binary cellular automata with different neighborhoods in

https://doi.org/10.25088/ComplexSystems.24.1.1

lattices of higher dimensions. To demonstrate this, we developed a genetic search algorithm [18] of nontotalistic two-dimensional cellular automata in the Moore neighborhood with radius equal to one. This algorithm searches for automata with behavioral measures similar to those in the Game of Life in a space of size 2^{512}. The genetic algorithm uses a cost function to evaluate each cellular automaton in the population, with this cost being the Euclidean distance between the behavioral measures of each cellular automaton and the behavioral measures of the Game of Life. Another selection condition was added: like the Game of Life, the selected cellular automaton must have $stability = 0$ in both its static and dynamic measures. We found a large number of cellular automata with interesting complex behaviors, like gliders, blinkers, and self-replicating patterns.

6.1 Tests and Results

The proposed genetic search algorithm evolved an initial population of 20 individuals through 5000 generations, each individual being a cellular automaton with a randomly generated transition function f. Each cellular automaton's transition function is represented in the population as a chromosome of 512 Boolean values. One-point crossover and random mutation (with probability 0.01) were applied at each evolution step [19]. The sampling used to measure dynamic behavior was taken at random intervals at least 10 times for each cellular automaton in the population. In a space of 2^{512} possible cellular automata, we generated about 10 000 different cellular automata through crossover and mutation and selected the 1000 closest to the behavioral measures of the Game of Life. These automata were qualitatively evaluated. We found 300 cellular automata in which gliders, blinkers, and other interesting complex behaviors can be observed. Among the cellular automata with complex behavior found, we identified a self-replicating cellular automaton, corresponding to Wolfram rule number

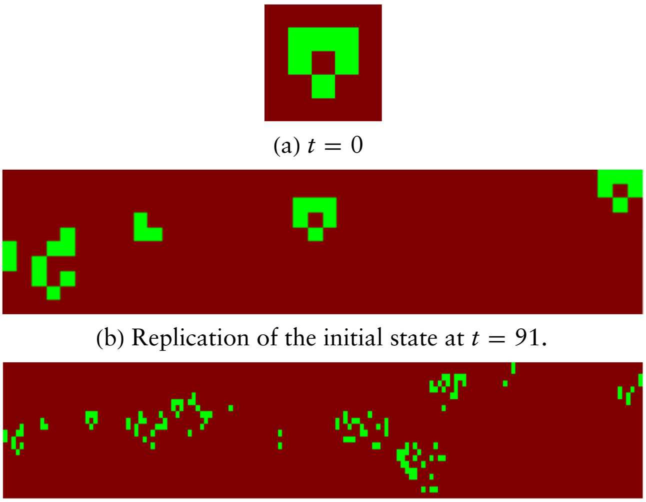
$$
\begin{aligned}
&16\,895\,622\,000\,315\,042\,854\,050\,654\,968\,041\,761\,976\,942\,499 \ddots \\
&\quad 540\,948\,773\,344\,255\,633\,961\,233\,308\,171\,712\,857\,937\,436 \ddots \\
&\quad 670\,105\,821\,967\,468\,216\,616\,118\,900\,334\,441\,708\,509\,286 \ddots \\
&\quad 446\,343\,520\,818\,184\,926\,824\,448.
\end{aligned}
$$

In this automaton, we can see a pattern that is replicated twice after 91 steps, as shown in Figure 7. Curiously, this complex cellular automaton is more distant from the behavioral measures of the Game of Life than other cellular automata found using the proposed methodology. However, one characteristic is prevalent in this and the other cel-

https://doi.org/10.25088/ComplexSystems.24.1.1

lular automata found: a negative correlation between their static and dynamic behavioral measures.

This self-replicative pattern is a particular kind of localized structure that moves across the cellular automaton's lattice. These localized structures, or "gliders" (which are not always self-replicative) can be seen as streaks in averaged spacetime slices that depict the evolution of the cellular automaton from random initial conditions. Figure 8 shows the spacetime slices depicting gliders on the self-replicative cellular automaton found and on the Game of Life, as comparison.



(a) $t = 0$

(b) Replication of the initial state at $t = 91$.

(c) Persistence of the pattern and its copy at $t = 307$.

Figure 7. Self replication in cellular automaton with behavioral metrics similar to the Game of Life; the feature vector of behavioral metrics used to find it is shown in Table 9.

	M_E	M_D
Chaoticity	61.72	5.61
Decrease	3.32	90.63
Growth	34.96	3.77
Stability	0	0

Table 9. Euclidean distance to the feature vector of behavioral metrics of the Game of Life is 21.31. The correlation between measures of static and dynamic behavior in this cellular automaton is -0.45.

https://doi.org/10.25088/ComplexSystems.24.1.1

Self–Replicative Cellular Automaton Game of Life

Figure 8. Averaged spacetime evolutions, showing gliders as streaks.

We present examples of complex cellular automata found with the proposed search method. Mean spacetime visualizations of the evolving state of the automaton are provided for each, the lower rows of the lattice being the later time steps. A list of 277 selected complex binary cellular automata can be found in the Bitbucket repository at https://bitbucket.org/antonio_rt/search-of-complex-binary-ca/src. A Java implementation of the genetic search algorithm based on behavioral metrics is available at http://discoverer.cellular-automata.com.

These cellular automata can be executed in *Mathematica*, replacing **⟨Rule⟩** with the corresponding rule number (see Appendix A).

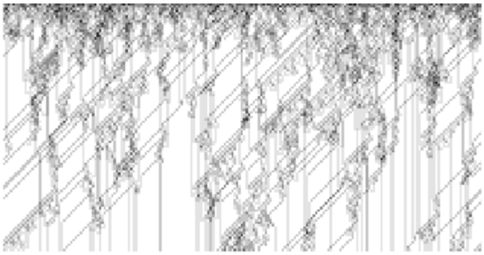
```
ListAnimate[ArrayPlot /@
  CellularAutomaton[{⟨Rule⟩, 2, {1, 1}},
   {RandomInteger[1, {100, 100}], 0}, 150]]
```

Rule:
354 830 437 430 697 307 314 658 045 280 649 922 899 653 607 237 ⋱
152 783 088 733 395 073 850 801 752 918 249 535 088 820 853 ⋱
655 864 680 729 189 540 963 997 737 594 766 246 170 112 169 ⋱
867 440 686 203 456

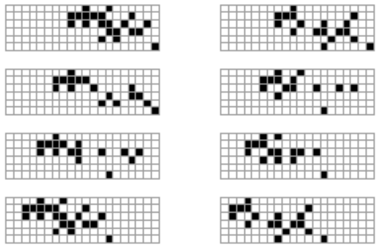
	M_E	M_D
Chaoticity	62.11	12.06
Decrease	4.88	78.88
Growth	33.01	9.06
Stability	0	0

Table 10. Euclidean distance to the feature vector of behavioral metrics of the Game of Life is 9.32. The correlation between measures of static and dynamic behavior in this cellular automaton is -0.34.

https://doi.org/10.25088/ComplexSystems.24.1.1



Averaged Spacetime Evolution



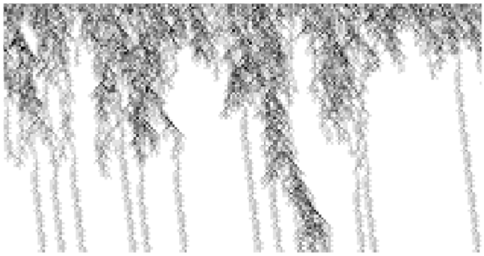
Identified Gliders

Rule:
196 928 112 803 567 351 078 509 513 317 947 776 313 717 639 009 ⋱
629 192 334 193 923 037 233 645 856 780 601 181 782 252 315 ⋱
349 164 603 950 024 916 004 629 851 769 274 774 088 586 292 ⋱
232 688 540 354 568

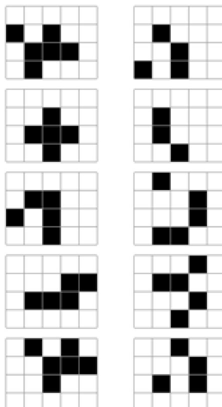
	M_E	M_D
Chaoticity	64.45	9.28
Decrease	2.54	84.80
Growth	33.01	5.92
Stability	0	0

Table 11. Euclidean distance to the feature vector of behavioral metrics of the Game of Life is 13.68. The correlation between measures of static and dynamic behavior in this cellular automaton is -0.40.

https://doi.org/10.25088/ComplexSystems.24.1.1



Averaged Spacetime Evolution



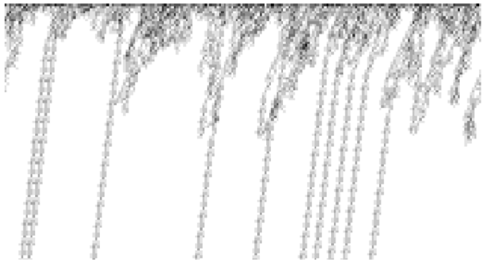
Identified Gliders

Rule:
2 536 962 858 330 445 998 944 606 509 915 061 353 621 502 280 ⋱
763 765 013 218 910 019 118 617 632 623 181 726 351 808 015 ⋱
804 669 971 129 335 990 123 389 394 577 484 439 270 322 287 ⋱
946 219 773 078 676 008

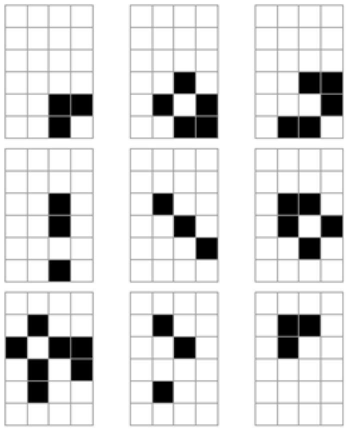
	M_E	M_D
Chaoticity	65.63	5.53
Decrease	3.91	90.54
Growth	30.47	4.00
Stability	0	0

Table 12. Euclidean distance to the feature vector of behavioral metrics of the Game of Life is 19.13. The correlation between measures of static and dynamic behavior in this cellular automaton is -0.42.

https://doi.org/10.25088/ComplexSystems.24.1.1



Averaged Spacetime Evolution



Identified Glider

Acknowledgments

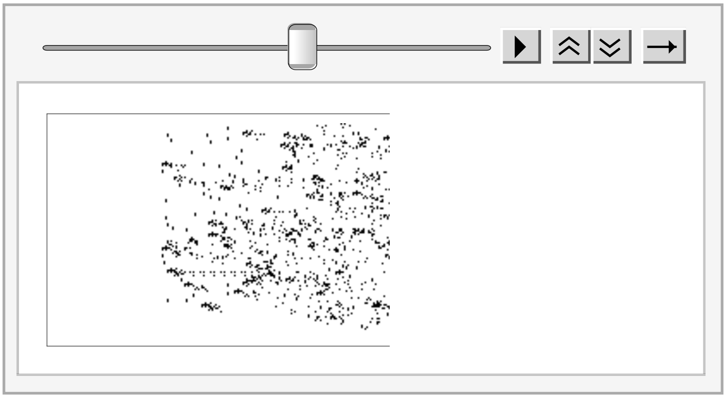
We wish to thank Jan Baetens, Hector Zenil, Alyssa Adams, and Nima Dehghani for their helpful comments. We appreciate the support of the Physics and Mathematics in Biomedicine Consortium. We also wish to thank Todd Rowland for his encouragement and continued interest in the project.

Appendix

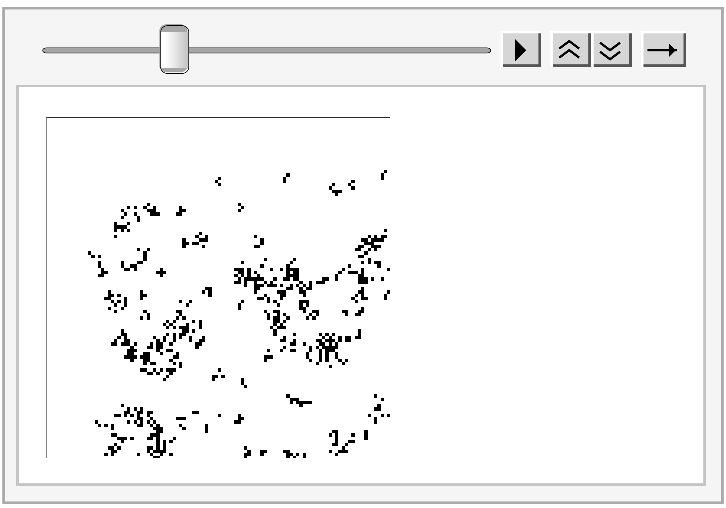
A. *Mathematica* Examples

```
animateCA[rule_] := ListAnimate[
  ArrayPlot /@ CellularAutomaton[{rule, 2, {1, 1}},
    {RandomInteger[1, {100, 100}], 0},
    150], ImageSize -> Small]
```

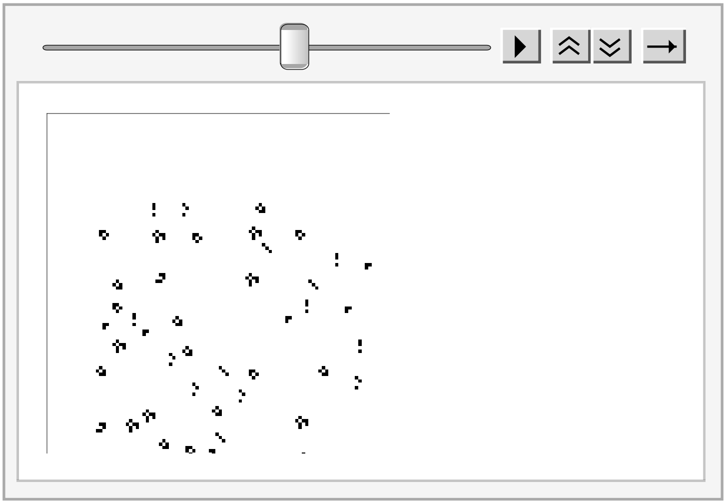
```
animateCA[
 354 830 437 430 697 307 314 658 045 280 649 922 899 653 607 237 \
  152 783 088 733 395 073 850 801 752 918 249 535 088 820 853 655 \
  864 680 729 189 540 963 997 737 594 766 246 170 112 169 867 440 \
  686 203 456]
```



```
animateCA[
 196 928 112 803 567 351 078 509 513 317 947 776 313 717 639 009 \
  629 192 334 193 923 037 233 645 856 780 601 181 782 252 315 349 \
  164 603 950 024 916 004 629 851 769 274 774 088 586 292 232 688 \
  540 354 568]
```



```
animateCA[
 2 536 962 858 330 445 998 944 606 509 915 061 353 621 502 280 763 
  765 013 218 910 019 118 617 632 623 181 726 351 808 015 804 669 
  971 129 335 990 123 389 394 577 484 439 270 322 287 946 219 773 
  078 676 008]
```



References

[1] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

[2] S. Wolfram, "Statistical Mechanics of Cellular Automata," *Reviews of Modern Physics*, **55**(3), 1983 pp. 601–644. doi:10.1103/RevModPhys.55.601.

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

[4] A. Adamatzky and J. Durand-Lose, "Collision-Based Computing," in *Handbook of Natural Computing* (G. Rozenberg, T. Bäck, and J. N. Kok, eds.), Springer, 2012 pp. 1949–1978.

[5] M. Nilsson and S. Rasmussen, "Cellular Automata for Simulating Molecular Self-Assembly," in *Discrete Models for Complex Systems (DMCS'03)*, Lyon, France, 2003 (M. Morvan and É. Rémila, eds.), Dijon, France: Discrete Mathematics and Theoretical Computer Science, 2003 pp. 31–42. www.dmtcs.org/proceedings/html/dmAB0103.abs.html.

https://doi.org/10.25088/ComplexSystems.24.1.1

[6] P. W. K. Rothemund, N. Papadakis, and E. Winfree, "Algorithmic Self-Assembly of DNA Sierpinski Triangles," *PLOS Biology*, **2**(12), 2004 e424. doi:10.1371/journal.pbio.0020424.

[7] H. Abelson, D. Allen, D. Coore, C. Hanson, E. Rauch, G. J. Sussman, G. Homsy, J. Thomas F. Knight, and R. W. Radhika Nagpal, "Amorphous Computing," *Communications of the ACM*, **43**(5), 2000 pp. 74–82.

[8] M. Hirabayashi, S. Kinoshita, S. Tanaka, H. Honda, H. Kojima, and K. Oiwa, "Cellular Automata Analysis on Self-Assembly Properties in DNA Tile Computing," *Lecture Notes in Computer Science*, **7495**, 2012 pp. 544–553. doi:10.1007/978-3-642-33350-7_ 56.

[9] M. Hwang, M. Garbey, S. A. Berceli, and R. Tran-Son-Tay, "Rule-Based Simulation of Multi-Cellular Biological Systems—A Review of Modeling Techniques," *Cellular and Molecular Bioengineering*, **2**(3), 2009 pp. 285–294. doi:10.1007/s12195-009-0078-2.

[10] G. B. Ermentrout and L. Edelstein-Keshet, "Cellular Automata Approaches to Biological Modelling," *Journal of Theoretical Biology*, **160**, 1993 pp. 97–133.

[11] G. Rozenberg, T. Bäck, and J. Kok, (eds.), *Handbook of Natural Computing*, New York: Springer, 2012 pp. 1–287, p. 1168.

[12] L. B. Kier, D. Bonchev, and G. A. Buck, "Modeling Biochemical Networks: A Cellular-Automata Approach," *Chemistry and Biodiversity*, **2**(2), 2005 pp. 233–243. doi:10.1002/cbdv.200590006.

[13] J. Wurthner, A. K. Mukhopadhyay, and C.-J. Peimann, "A Cellular Automaton Model of Cellular Signal Transduction," *Computers in Biology and Medicine*, **30**(1), 2000 pp. 1–21.
doi:10.1016/S0010-4825(99)00020-7.

[14] M. Gardner, "Mathematical Games: The Fantastic Combinations of John H. Conway's New Solitaire Game 'Life'," *Scientific American*, **223**, 1970 pp. 120–123.

[15] J. M. Baetens and B. De Baets, "Towards the Full Lyapunov Spectrum of Elementary Cellular Automata," in *Proceedings of the International Conference on Numerical Analysis and Applied Mathematics (ICNAAM 2011)*, Halkidiki, Greece (T. E. George Ch. Zacharias, S. Psihoyios, and T. Anastassi, eds.), Melville, NY: AIP Publishing, 2011 pp. 981–986. doi:10.1063/1.3637774.

[16] J. M. Baetens and J. Gravner. "Stability of Cellular Automata Trajectories Revisited: Branching Walks and Lyapunov Profiles."
arxiv.org/abs/1406.5553.

[17] H. Zenil and E. Villareal-Zapata, "Asymptotic Behavior and Ratios of Complexity in Cellular Automata," *International Journal of Bifurcation and Chaos*, **23**(9), 2013 p. 1350159. doi:10.1142/S0218127413501599.

[18] M. Mitchell, J. P. Crutchfield, and P. T. Hraber, "Evolving Cellular Automata to Perform Computations: Mechanisms and Impediments," *Physica D: Nonlinear Phenomena*, **75**(1–3), 1994 pp. 361–391. doi:10.1016/0167-2789(94)90293-3.

[19] M. Mitchell, *An Introduction to Genetic Algorithms*, Cambridge, MA: MIT Press, 1996.