Search of Complex Binary Cellular Automata Using Behavioral Metrics

Juan C. López-González
Wolfram Research
and
Physics and Mathematics in Biomedicine Consortium, UCV
jlopez@cellular-automata.com

Antonio Rueda-Toicen
Instituto Nacional de Bioingeniería
Universidad Central de Venezuela
and
Physics and Mathematics in Biomedicine Consortium, UCV
antonio.rueda@ciens.ucv.ve

We propose the characterization of binary cellular automata using a set of behavioral metrics based on an evaluation heuristic 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 expression of the automaton; 2) a dynamic measure, corresponding to the mean of the behavioral metrics in n executions of the automaton, starting from n random initial states. The correlation between these measures is used to 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 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, 2, 3, 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, 6, 7, 8] and the modelling of biological processes like signaling, division, apoptosis, necrosis and differentia-

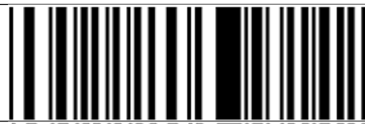
Rule	Sample Evolution	Boolean Form	Behavior
R_{204}		q	Stable
R_{160}		p *AND* r	Decreasing
R_{252}		p *OR* q	Growing
R_{90}		p *XOR* q	Chaotic

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.

tion [9, 10, 11, 12, 18]. John Conway's Game of Life [13] 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 [14, 15], and Kolmogorov-Chaitin complexity [16]. 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, 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. The correlation between these static and dynamic measures can be used as another characterization criterion, and its closeness to the measures of the Game of Life is used 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 written as sequences of characters, such as

$$C = \ldots c_{-1} c_0 c_1 c_2 \ldots \tag{1}$$

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

$$C = c_1 c_2 \ldots c_x c_{x+1} \ldots c_{n-2} c_{n-1} \tag{2}$$

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.

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

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

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

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

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 yields the value of c_x at $t+1$ from the neighborhood of the cell observed at present time t is expressed by

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

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

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

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

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

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

$$F(C_x) = \dots f(N_{x-1}) f(N_x) f(N_{x+1}) \dots \tag{9}$$

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

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

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

$$\mathit{Input_1} < \mathit{operator} > \mathit{Input_2} = \mathit{Output} \tag{11}$$

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

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

$$< \mathit{operator} > \mathit{Input} = \mathit{Output} \tag{12}$$

where $< \mathit{operator} > \in \{NOT, NOP\}$

where NOP stands for "no operator". To characterize the automaton's behavior, we expand the state space

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

Input₁	*Input₂*	*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

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

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.

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.

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.

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,

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

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

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

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 correspond 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

$$term < OPERATOR > term$$

and

$$< OPERATOR > term$$

where $< OPERATOR > \in \{AND, OR, XOR, NOT\}$ 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.

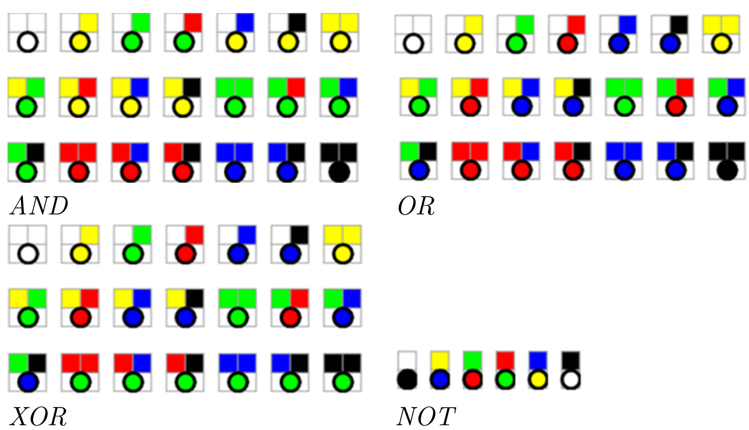


Figure 1. Tree Evaluation rules H, squares correspond to inputs and circles to outputs. White corresponds to $M = 0 = \{0, stable\}$; yellow corresponds to $M = 1 = \{0, decrease\}$; green corresponds to $M = 2 = \{0, chaotic\}$; red corresponds to $M = 3 = \{1, chaotic\}$; blue corresponds to $M = 4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

- **Criterion 1 -** In the leaf nodes, $S = 0$ must be equivalent to $M = 0 = \{0, \mathit{stable}\}$ and $S = 1$ must be equivalent to $M = 5 = \{1, \mathit{stable}\}$.
- **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:

$$R_{150}\ \mathit{chaoticity} = 0.375$$
$$R_{90}\ \mathit{chaoticity} = 0.25$$

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

$$R_{90}\ \mathit{chaoticity} = 0.25$$
$$R_{204}\ \mathit{chaoticity} = 0$$

- **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:

$$R_{128}\ \mathit{decrease} = 0.75$$
$$R_{160}\ \mathit{decrease} = 0.5$$

- **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:

$$R_{160}\ \mathit{decrease} = 0.5$$
$$R_{204}\ \mathit{decrease} = 0$$

- **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:

$$R_{254}\ \mathit{growth} = 0.75$$
$$R_{250}\ \mathit{growth} = 0.5$$

Figure 2 shows percentage of measured behaviors, using the proposed set of evaluation rules H, in the elementary cellular automata considered in the criteria.

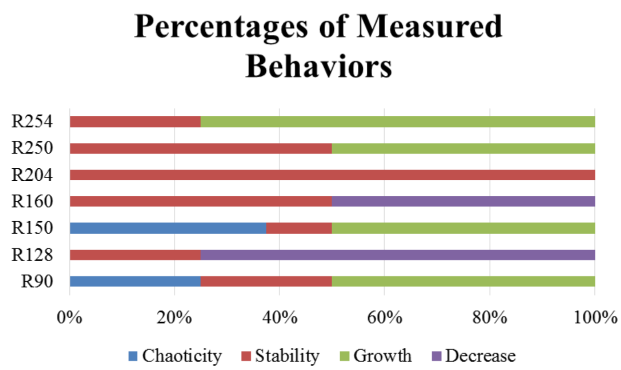


Figure 2. Behavioral percentages in elementary cellular automata considered as criteria for evaluating the proposed heuristic

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 3. Each node in the tree is evaluated using the rules shown in Figure 1. This process is demonstrated in the Steps 1-5 listed below.

- **Step 1.** In the leaf nodes, the values $N_q^{t=0}$ are $\{p=1, q=0, r=1\}$, 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.

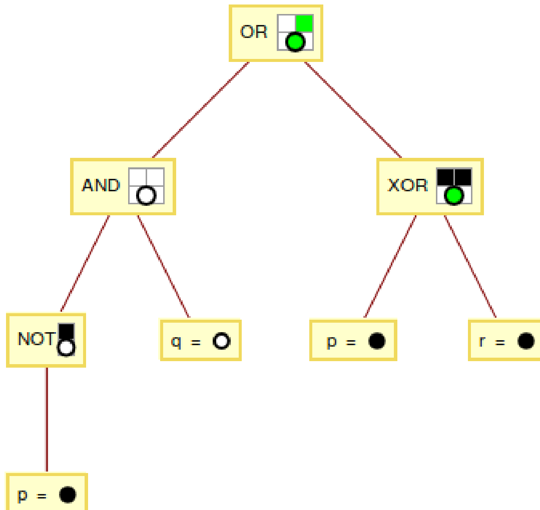


Figure 3. Evaluation of the input pattern 101 in R_{94} with the proposed rules.

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. This static measure is represented as 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.

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

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.

We express this measure as a vector of percentages.

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

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 $C_i^{t=0} | i \in n$. 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.

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

We exclude cells at $t = 0$ from the sampling. The percentages of behavioral occurrences are calculated from the mean of samples. Figure 4 shows the sampling of R_{94} in $k = t = 20$.

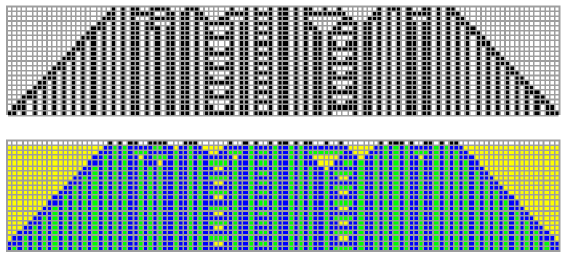


Figure 4. Evolution of $R_9 4$ from a random initial configuration, yellow coloring for $M = 1$, green coloring for $M = 2$ and blue coloring for $M = 4$. The code M was applied to cells in $t \geq 1$. The percentage of cells with $M = 1$ (decreasing behavior) is 18.658%, cells with $M = 2$ (stable behavior) are 32.467% and cells with $M = 4$ (chaotic behavior) occupy 48.874% of the lattice.

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 and the absolute difference between the static measure and the dynamic measure in the Game of Life.

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, one can find patterns such as

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

It is our hypothesis that the emergence of complex behavior in the Game of Life is determined by the appearance of islands of chaotic

	M_E	M_D	$\lvert M_E - M_D \rvert$
Decrease	4.68	75.23	70.55
Chaoticity	67.96	13.38	-54.58
Growth	27.34	11.37	-15.97
Stability	0	0	0

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

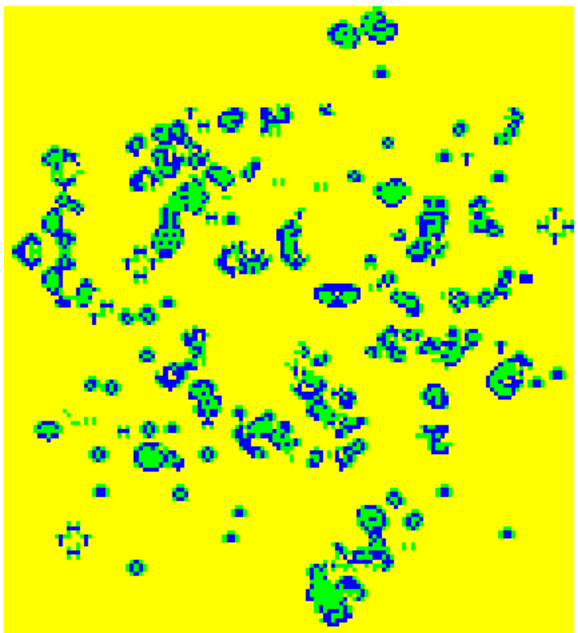


Figure 5. The Game of Life, colored according to behavior

behavior, surrounded by decreasing patterns. Taking a close look at the boolean expression of f in the Game of Life, one can observe chaotic sub-expressions like $\boldsymbol{(x3\; XOR\; x4)}$ being "restricted" with *AND-ing* by decreasing sub-expressions such as $(AND\; NOT\; x2\; AND\; x8)$.

In Figure 5, yellow cells have value $M = 1$ (decreasing behavior), and blue cells have value $M = 4$ (growth behavior). Green cells cells have $M = 2$, exhibiting chaotic behavior. Note that in Figure 5, decreasing cells ($M = 1$) cover the largest proportion of the lattice, which corresponds with the dynamic measure of measure of decrease shown in Table 8. One can also appreciate how the isolated patterns exhibit a combination of growth ($M = 4$) and chaoticity ($M = 2$).

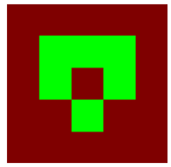
6. Search of complex binary cellular automata in two dimensions

The proposed behavioral metrics were crafted using heuristic criteria from one-dimensional binary cellular automata, yet are applicable to binary cellular automata in lattices of higher dimensions. We developed a genetic search algorithm [17] of non-totalistic 2D 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} . We found a large number of cellular automata with interesting complex behaviors, like gliders, blinkers and self-replicating patterns. The genetic algorithm uses a cost function that evaluates each randomly generated transition rule, with cost being the distance between the behavioral measures of each generated cellular automaton with the behavioral measures of the Game of Life. Another selection condition was added: the selected cellular automaton must have *stability* $= 0$ in both its static and dynamic measures.

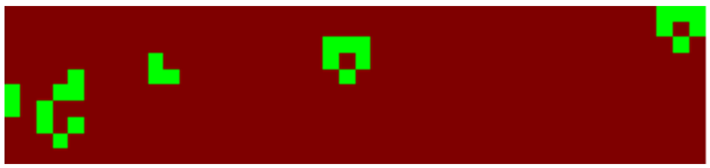
6.1 Tests and results

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. In a space of 2^{512} possible cellular automata, we generated about 10000 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 one can appreciate gliders, blinkers, and other interesting complex behaviors. Among the cellular automata with complex behavior found, we identified a self-replicating cellular automata, corresponding to code [1, 3] 168956220003150428540506549680417619769424 995409487733442556339612333081717128579374366701058219674682 1 66161189003344417085092864463435208181849268244448.
In this automaton, we can appreciate a pattern that is replicated twice after 91 steps, as shown in Figure 6.

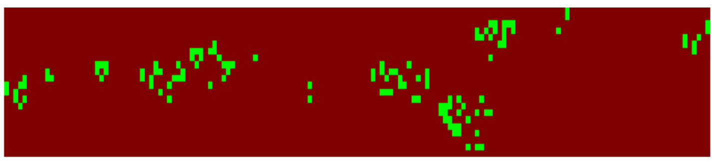
We present examples of complex cellular automata found with the proposed search method[1]. Mean spacetime visualizations of the evolving state of the automaton are provided for each; the lower rows of the lattice being the latter time steps. A list of 277 selected complex binary cellular automata can found at the Bitbucket repository at A list of 277 selected complex cellular automata can be found in the Bitbucket repository at `http://bit.ly/complexbinaryca`. A Java implementation of the search algorithm is also available in this repo.



(a) $t = 0$



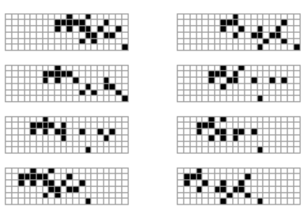
(b) Replication of the initial state at $t = 91$



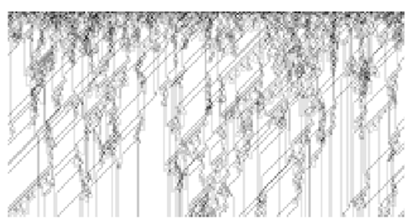
(c) Persistence of the initial state and its copy at $t = 307$

Figure 6. Self replication in cellular automaton with behavioral measures close to those of the Game of Life.

Rule: 3548304374306973073146580452806499228996536072371527830887333950738508017529182495350888208536558646807291895409639977375947662461701121698674406862034 56



Identified gliders



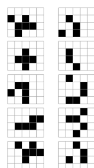
Averaged spacetime evolution

---

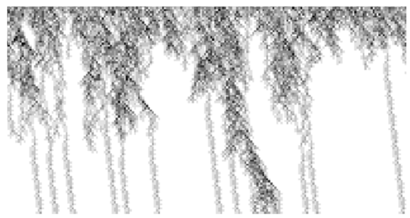
[1]One can execute these cellular automata in *Mathematica* replacing `<Rule>` with the corresponding rule number

```
ListAnimate[ArrayPlot[#]&/@CellularAutomaton[
  <Rule>,2,1,1, RandomInteger[1,100,100],0,100]]
```

Rule: 196928112803567351078509513317947776313717639009629 192334193923037233645856780601181782252315349164603950024916004629851769274774088586292232688540354568

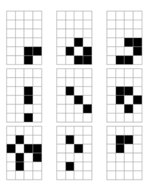


Identified gliders

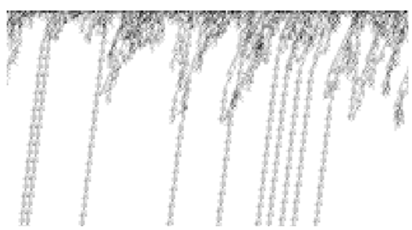


Averaged spacetime evolution

Rule: 196928112803567351078509513317947776313717639009629192334193923037233645856780601181782252315349164603950024916004629851769274774088586292232688540354568



Identified gliders



Averaged spacetime evolution

Acknowledgements

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

References

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

[2] S. Wolfram, "Statistical Mechanics of Cellular Automata," *Reviews of Modern Physics*, **55**(3), 1983, pp. 601-644.

[3] S. Wolfram, "Universality and Complexity in Cellular Automata," *Physica D: Nonlinear Phenomena*, **10**(1), 1984, pp. 1-35.

[4] A. Adamatzky and J. Durand-Lose, "Collision Based Computing," section of the *Handbook of Natural Computing*, Springer, 2012, pp. 1949-1978

[5] M. Nilsson and S. Rasmussen, "Cellular Automata for Simulating Molecular Self-Assembly," *Discrete Mathematics and Theoretical Computer Science*, 2003, pp. 31-42.

[6] P. Rothemund, N. Papadakis and E. Winfree, "Algorithmic Self-Assembly of DNA Sierpinski Triangles," *PLOS Biology*, 2004.

[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.

[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.

[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, (editors), *Handbook of Natural Computing*, Springer, 2012.

[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.

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

[14] J. M. Baetens and B. De Baets, "Towards the Full Lyapunov Spectrum of Cellular Automata," *AIP Conference Proceedings*, **1389**(1), 2011, pp. 981-986.

[15] J.M. Baetens and J. Gravner, "Stability of cellular automata trajectories revisited: branching walks and Lyapunov profiles," 2014. [Online]. Available: `http://arxiv.org/pdf/1406.5553.pdf`.

[16] H. Zenil and E. Villareal-Zapata, "Asymptotic Behaviour and Ratios of Complexity in Cellular Automata," *International Journal of Bifurcation and Chaos*, **23**(9), 2013.

[17] M. Mitchell, J. P. Crutchfield and P. T. Hraber, "Evolving Cellular Automata to Perform Computations: Mechanisms and Impediments," *Physica D: Nonlinear Phenomena*, 75(1), 1994, pp. 361-391.

[18] J. Wurthner, A. Mukhopadhyay, and Claus-Jürgen Peimann. "A cellular automaton model of cellular signal transduction." Computers in biology and medicine **30**(1), 2000, pp. 1-21.