Search of Complex Binary Cellular Automata Using Behavioral Metrics

Juan C. López-González, Antonio Rueda-Toicen

We propose the characterization of binary cellular automata using a set of behavioral metrics based on an evaluation heuristic. The characterized behaviors are growth, decrease, chaoticity and stability. From these proposed metrics, we calculate two measures of overall behavior: 1) a static measure that considers all possible input patterns and counts the occurrence of each behavioral metric in the output obtained; 2) a dynamic measure, corresponding to the mean occurrence of 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

Previously, complex behavior in binary cellular automata has been characterized through measures such as Lyapunov exponents, entropy, and Kolmogorov complexity [1, 2, 3]. The observation that has motivated this research comes from the study of elementary cellular automaton R_{94} [3], shown in Figure 1. We have observed the aggregated influences of R_{90} and R_{12} in the evolution of R_{94}.

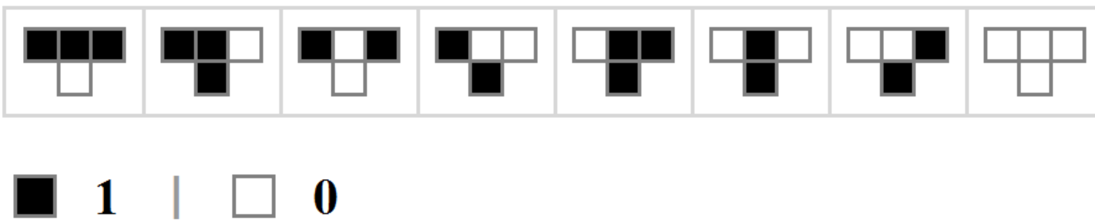


Figure 1a - Elementary cellular automaton R_{94}. For each tuple of states, the bottom cell s shows the state S^{t+1} obtained from the evaluation in t of the neighborhood comprising the cells p (left cell), q (central cell) and r (right cell).

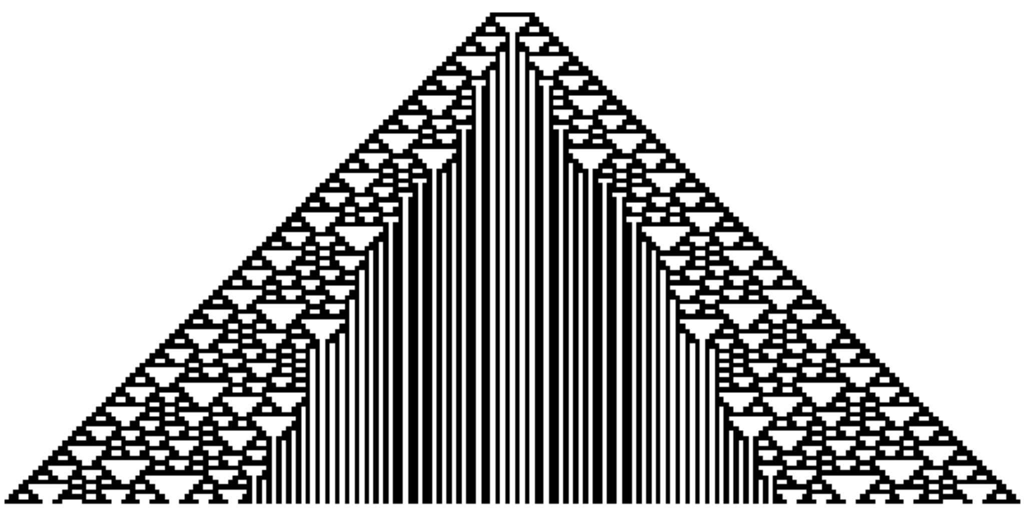


Figure 1b - Evolution for $t = 100$ of R_{94} with initial configuration of 9 contiguous cells at $S = 1$.

In R_{94} we observe a symbiotic relationship between patterns of decreasing behavior, patterns of chaotic behavior, and patterns of stable behavior. In R_{94}, we can observe patterns of chaoticity and growth, which are characteristic of R_{90}, peripheral to a pattern of stable behavior, which is characteristic of R_{12}. Areas of stability and chaoticity grow continuously when the automaton is initialized with an initial configuration of $2n + 1 \geq 5$ adjacent cells in $S = 1$. R_{94} is a complex rule, which can be expressed in Boolean form as

$$R_{94} = R_{90}\ OR\ R_{12} \qquad (1)$$

In R_{94} we observe symbiosis between the characteristic chaotic behavior of R_{90} (Class III [4]) and the characteristic stable behavior of R_{12} (Class II). Based on this observation, we defined a heuristic that characterizes the effect of the aggregated interactions of the operators AND, OR and XOR on cellular automata rules expressed on minimal Boolean form. This heuristic allows us to define behavioral metrics, which we use to create static and dynamic measures of behavior. The correlation between these measures is used to find cellular automata with complex behavior in n dimensions.

2 Definition of Binary Cellular Automaton

A cellular automaton is formally represented by a tuple $\{Z, S, N, f\}$ 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 neighborhood of the cells

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, $t + 1$ the future state of a cell is determined by the evaluation of the local transition function on the cell's neighborhood at 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

$$C = \ldots c_{-1} c_0 c_1\, c_2 \ldots \qquad (2)$$

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,

$$C = c_1\, c_2 \ldots\, c_{x-1}\, c_x\, c_{x+1} \ldots c_{n-2}\, c_{n-1} \qquad (3)$$

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.

$$A = \{a_0, a_1, \ldots, a_{m-2}, a_{m-1}\} \qquad (4)$$

Applying the set A in an observed cell c_x, the neighborhood N_x, which includes the state of each neighboring cell is obtained

$$N_x = c_{x+a_0} c_{x+a_1} \ldots c_{x+a_{m-2}} c_{x+a_{m-1}} \qquad (5)$$

Equation 5 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 [3].

$$N_x = \sum_{i=0}^{m-1} k^{m-1-i}\, c_{x+a_i} = c_{x+a_0} k^{m-1} + c_{x+a_1} k^{m-2} + \cdots + c_{x+a_{m-1}} k^0 \quad (6)$$

$$f: S^N \to S \qquad (7)$$

The local transition function f calculates the value of c_x at $t+1$ from the neighborhood of the cell observed at present time t.

$$f(N_x^t) = c_x^{t+1} \qquad (8)$$

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 1 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 $\{c_{x-1}, c_x, c_{x+1}\}$ $S \in \{0,1\}$.

N_x^t	$f(N_x^t)$
000	0
001	1
…	…
111	1

Table 1 - Local transition function as truth table

2.3 Global Transition Function

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

$$F: S^N \rightarrow S^N \qquad (9)$$

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

$$C^{t+1} = F(C^t) \qquad (10)$$

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

$$F(C_x) = \cdots f(N_{x-1}) f(N_x) f(N_{x+1}) \ldots. \qquad (11)$$

3 Transformation of the Cellular Space

Cellular behaviors of growth, decrease, and chaoticity are characterized by the Boolean operations OR, AND and XOR, respectively. This is illustrated in Figure 2. The cellular behavior of stability is characterized by the absence of a Boolean operator, or by the use of the NOT[1] operator. We propose the inclusion of information on the cellular automaton's behavior to characterize its evolution.

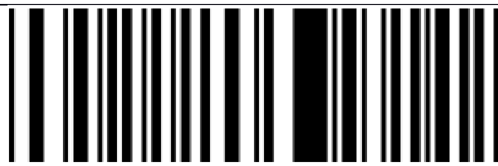
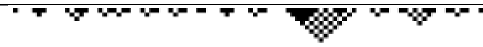
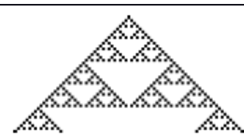
Rule	**Sample Evolution**	**Boolean Form**	**Behavior**
R_{204}		q	Stable
R_{160}		$p\ AND\ r$	Decreasing
R_{252}		$p\ OR\ q$	Growing
R_{90}	 	$p\ XOR\ q$	Chaotic

Figure 2 - Behaviors associated with various elementary cellular automata and their characteristic Boolean operators

3.1 Redefined Local Transition Function *g*

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 defines the next step of evolution of cell c_x as

$$d_x^{t+1} = g(f, N_x^t) \qquad (12)$$

This transformation is necessary to calculate the measure of dynamic behavior during the automaton's evolution.

---

[1] It is our hypothesis that the Boolean operator NOT rather characterizes cyclical behavior. In this work, we group together cyclic with stable behavior. Characterization of cyclical behavior separately from stable behavior is a topic worth further exploration.

We propose applying a transformation that includes the metrics characterizing cell behavior and its associated Boolean patterns to the binary operators of the form

$$Input_1 \ < operator > Input_2 = Output \qquad (13)$$

where

$$< operator > = \{OR, AND, XOR\}$$

The cellular behaviors associated with each binary Boolean operator are shown in Table 1.

				$< operator >$		
$Input_1$	$Input_2$	$Output$	$Behavior$	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 2 - Patterns of binary Boolean operators and associated behavior

And the following unary patterns shown in Table 3

$$< operator > = \{NOT, NOP\}, \quad (14)$$

$$< operator > \ Input \ = \ Output,$$

			$< operator >$	
$Input$	$Output$	$Behavior$	NOT	NOP
1	1	Stability		X
0	0	Stability		X
1	0	Stability	X	
0	1	Stability	X	

Table 3 – Behavioral Patterns in Unary Logical Operators

where NOP stands for “no operator”.

To characterize the automaton’s behavior, we expand the state space

$$g: \{S^N, f\} \rightarrow M \tag{15}$$

where

$$M = \{0,1,2,3,4,5\} \tag{16}$$

The different values of M abbreviate the tuples of states and behaviors, shown in Table 4. 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 4 – M code, abbreviation of tuples of cell state and behavior obtained when applying local transition function g.

The M code facilitates the implementation of algorithmic search for cellular automata with interesting behavior. 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 are characterized as

$$G: \{S^N, f\} \to M^N \qquad (17)$$

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

$$D^{t+1} = G(C^T, f) \qquad (18)$$

$$C^0 \ (initial\ state)$$

$$C^1 = F(C^0) \rightarrow D^1 = G(C^0, f)$$

$$C^2 = F(C^1) \rightarrow D^2 = G(C^1, f)$$

$$C^3 = F(C^2) \rightarrow D^3 = G(C^2, f)$$

$$C^4 = F(C^3) \rightarrow D^4 = G(C^3, f)$$

$$\vdots$$

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

$$G(\ldots c_{x-1} c_x c_{x+1}, f) = \cdots g(n_{x-1}, f) g(n_x, f) g(n_{x+1}, f) \ldots. \qquad (19)$$

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 defined transformation rules $\{S, S\} \rightarrow M$ for binary logical operators and $\{S\} \rightarrow M$ for unary logical operators. These transformations are insufficient to characterize the behavior of cellular automata whose minimal expressions have several Boolean operators. To tackle this problem, we express the automaton's transition function in a binary evaluation tree and propose a set of evaluation rules of 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 being considered at its leaves, according to heuristic rules of precedence. The result of the evaluation of the tree corresponds to the evaluation obtained at the root node. The heuristics considered are based on criteria 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

$$term\ OPERATOR\ term, \qquad \text{and}$$

$$OPERATOR\ term$$

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

Figure 3 shows the heuristic precedence rules defined for each logical operator.

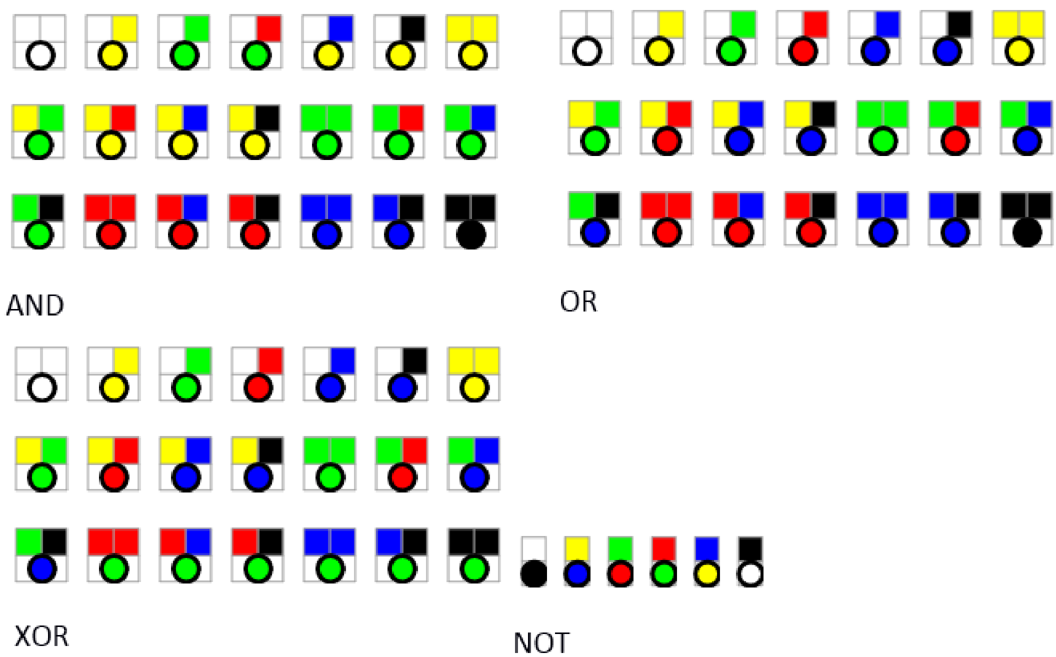


Figure 3 –Tree Evaluation Rules *H* – Squares correspond to inputs and circles to outputs. White corresponds to $M = 0 = \{0, Stable\}$, yellow corresponds to $M = 1 = \{0, Decrease\}$, green corresponds to $M = 2 = \{0, Chaotic\}$, red corresponds to $M = 3 = \{1, Chaotic\}$, blue corresponds to $M = 4 = \{1, Growth\}$, black corresponds to $M = 5 = \{1, Stable\}$

The tree evaluation rules were crafted to produce results consonant with the following heuristic criteria, based on the on the ratios of chaoticity, stability, decrease, and growth identified as the dynamic behavioral measure of various elementary cellular automata.

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

Criterion 2 - Chaoticity measured in $R_{150} = p\ XOR\ q\ XOR\ r$ must be greater than chaoticity measured in $R_{90} = p\ XOR$.

The proposed heuristic H produces the following behavioral metrics in these automata:

$$R_{150}\ chaoticity\ =\ 0.375$$

R_{90} $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} $chaoticity$ = 0.25

R_{204} $chaoticity$ = 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} $decrease = $ 0.75

R_{160} $decrease = $ 0.5

Criterion 5 - Decrease measured in $R_{160} = p\ AND\ r$ must be greater than decrease measured in $R_{204} = q$

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

$R_{250}\ growth = 0.5$

Criterion 7 - Growth measured in $R_{250} = p\ OR\ r$ must be greater than growth measured in rule $R_{204} = q$

The proposed heuristic H produces the following behavioral metrics in these automata:

$R_{250}\ growth = 0.5$

$R_{204}\ growth = 0$

Figure 4 shows percentages of measured behaviors, using the proposed set of tree evaluation rules H, in the elementary cellular automata considered in the criteria.

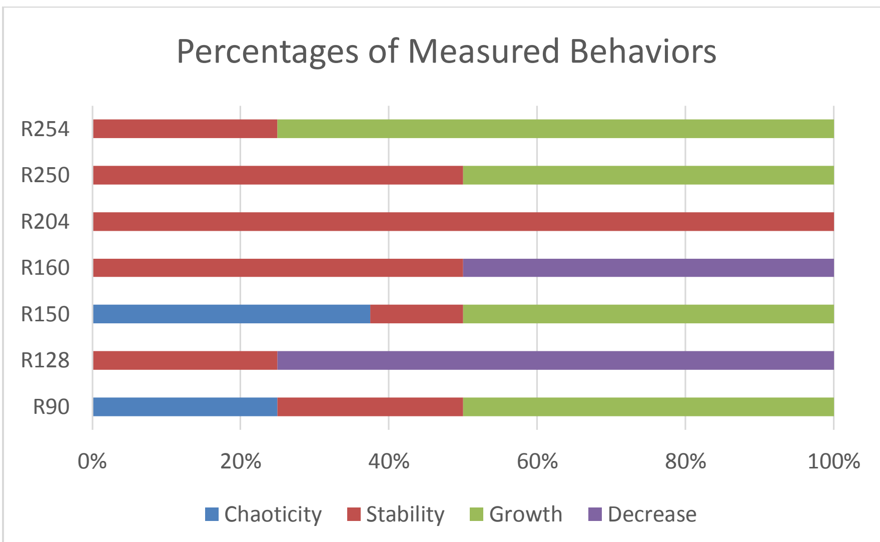


Figure 4 - Proportions of behavior in elementary cellular automata rules considered as criteria for evaluating the proposed heuristic

3.3.3.1 Evaluation Example with R_{94}

The minimal Boolean expression of R_{94}, $f = \big(q\ AND\ (NOT\ p)\big)\ OR\ (p\ XOR\ r)$, is placed in a binary evaluation tree, as shown in Figure 5. Each node in the tree is evaluated using the rules in Figure 2. The considered input is $N_q^{t=0} = \{p = 1,\ q = 0,\ r = 1\}$.

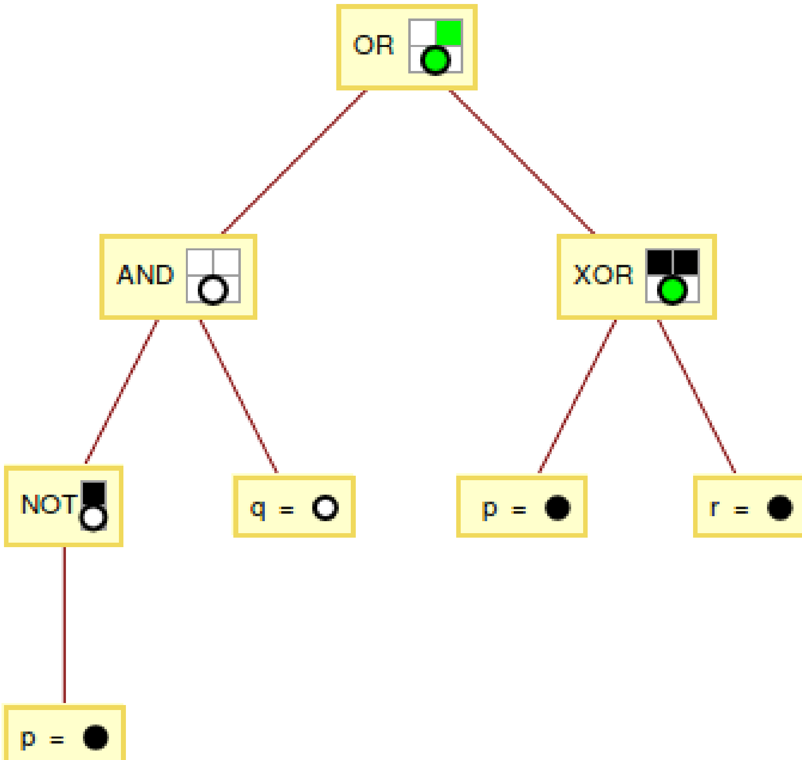


Figure 5 - Evaluation of R_{94} with the input pattern

$$N_q^{t=0} = \{p = 1,\ q = 0,\ r = 1\}$$

1. In the leaf nodes, the values $N_q^{t=0} = \{p = 1,\ q = 0,\ r = 1\}$**,** are transformed as follows:

$$S_M(p) = \ 5 \ = \{1, stable\}$$

$$S_M(q) = \ 0 \ = \{0, stable\}$$

$$S_M(r) = \ 5 \ = \{1, stable\}$$

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

2. Leaf $p = 5$ is evaluated at the NOT node producing output $0 = \{0, Stable\}$

3. Leaf $q = 0$ and the result of step 3 are evaluated at the AND node, producing output $0 = \{0, Stable\}$.

4. Leaves $p = 5$ and $r = 5$ are evaluated at the XOR node producing output $2 = \{0, Chaotic\}$.

5. The output of step 4 and the output of step 5 are evaluated the OR node, producing as final output $2 = \{0, Chaotic\}$.

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, that corresponds to the occurrence of behaviors associated to each output M, accounting all possible input patterns in the definition of f.

2) A dynamic measure, that corresponds to the median occurrence of behaviors associated to each output 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 each value of M in the output obtained and totalize behaviors. 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 5.

N_x	$f(N_x)$	$g(N_x, f)$
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 5 –Truth table for R_{94}, with associated M code

To obtain the static measure of R_{94}, the we count M occurrences, shown in Table 6.

The static measure of the rule is the percentage of behavioral occurrence, as shown in Table 7.

Stability	*Decrease*	*Growth*	*Chaoticity*
M = {0, 5}	M = 1	M= 4	M = {2, 3}
0 occurrences, 0%	1 occurrence, 12.5%	5 occurrences, 62.5%	2 occurrences, 25%

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

We express this as a vector of percentages

$$M_E\,(g) = \{\%\,of\ stability,\ \%\,of\ decrease, \%\,of\ growth, \%\,of\ chaoticity\} \quad (26)$$

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

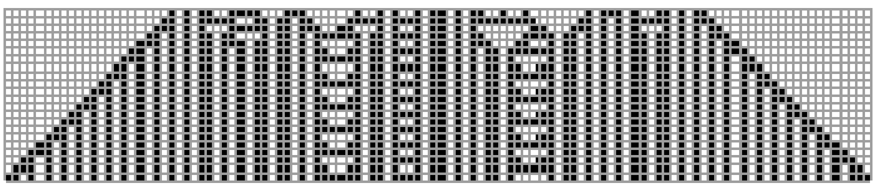
$$M_E(g) = \{0, 12.5, 62.5, 25\} \quad (27)$$

4.2 Dynamic Measure of Behavior

To calculate the dynamic measure of behavior M_D, we execute n times the cellular automaton from n random initial configuration $C_i^{t=0} | \; i \in n$. 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.

$$M_D(g) = \lim_{n \to \infty} \frac{\left(M_D^{t=k}(g, C_i^{t=0})\right)}{n} \tag{28}$$

We exclude cells at $t = 0$ from the sampling. The percentages of behavioral occurrences are calculated from the mean of samples. Figure 6 shows the sampling of R_{94} in $k = t = 20$.



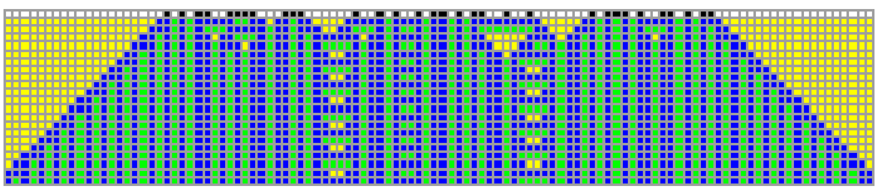


Figure 6 – Evolution of R_{94} from a random initial configuration, yellow coloring for $M = 1$, green coloring for $M = 2$ and blue coloring for $M = 4$. Code M applied to cells in $t \geq 1$. The percentage of cells with $M = 1$ (decreasing behavior) is 18.658%, cells with $M = 2$ (stable behavior) are 32.4675% and cells with $M = 4$ (chaotic behavior) are 48.8745%.

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 [4, 5]. In this cellular automaton, there's a significant difference between the static measurement of behavior and the dynamic measure of behavior, there is a negative correlation between them. Table 8 shows the dynamic measure, the static measure and their absolute difference in the Game of Life.

	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
Correlation	-0.29		

Table 8 – Correlation between static and dynamic measure in the Game of Life

Some observations pertinent to the measures in the Game of Life:

º Static measure: chaotic behavior predominates, an important characteristic of class III automata.

º 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

(...)

$NOT\ x0\ AND\ NOT\ x1\ AND\ NOT\ x2\ AND\ x8\ AND\ (\ x3\ XOR\ x4\ )\ AND\ (\ x5\ XOR\ x6\ )\ OR$

$NOT\ x0\ AND\ NOT\ x1\ AND\ NOT\ x3\ AND\ x8\ AND\ (\ x2\ XOR\ x4\ )\ AND\ (\ x5\ XOR\ x6\ )\ OR$

$NOT\ x0\ AND\ NOT\ x2\ AND\ NOT\ x3\ AND\ x8\ AND\ (\ x1\ XOR\ x4\ )\ AND\ (\ x5\ XOR\ x6\ )\ OR$

$NOT\ x1\ AND\ NOT\ x2\ AND\ NOT\ x3\ AND\ x8\ AND\ (\ x0\ XOR\ x4\ )\ AND\ (\ x5\ XOR\ x6\ )\ OR$

$NOT\ x0\ AND\ NOT\ x1\ AND\ NOT\ x2\ AND\ x8\ AND\ (\ x3\ XOR\ x5\ )\ AND\ (\ x4\ XOR\ x6\ )\ OR$

$NOT\ x0\ AND\ NOT\ x1\ AND\ NOT\ x3\ AND\ x8\ AND\ (\ x2\ XOR\ x5\ )\ AND\ (\ x4\ XOR\ x6\ )\ OR$

$NOT\ x0\ AND\ NOT\ x2\ AND\ NOT\ x3\ AND\ x8\ AND\ (\ x1\ XOR\ x5\ )\ AND\ (\ x4\ XOR\ x6\ )\ OR$

(….)

It is our hypothesis that the emergence of complex behavior in the Game of Life is determined by the appearance of “islands” of chaotic patterns, 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 as $(x1\ XOR\ x2)$ being "restricted" by ANDing with decreasing sub-expressions such as $(x1\ AND\ x2\ AND\ x3)$.

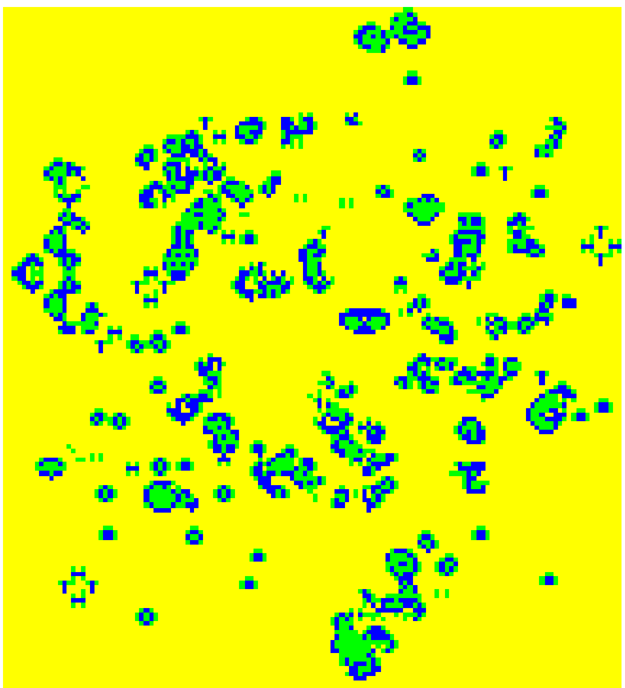


Figure 8- The Game of Life, colored according to behavior

In Figure 8, the yellow cells represent decreasing behavior of $M = 1$, blue cells represent growth behavior of $M = 4$ and the red cells represent chaotic behavior $M = 2$.Note that in Figure 8, the patterns with dense activation $(M = 1)$ correspond to decreasing patterns, covering the largest proportion of the lattice. We can also appreciate that the patterns of complex behavior here are a combination of growth ($M = 4$) and chaoticity ($M = 2$).

6 Search of Complex Cellular Automata in Two Dimensions

The proposed behavioral measures were initially developed 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 [6] of 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 new cellular automaton must have $Stability = 0$ in both its static and dynamic measures.

7.1 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 [3] 16895622000315042854050654968041761976942499540948773344255633961233308171712857937436670105821967468216616118900334441708509286446343520818184926824448. In this automaton we can appreciate a pattern that is replicated twice after 91 iterations, as shown in Figure 9. A list of 277 selected complex cellular automata can be found in the Bitbucket repository at http://bit.ly/1q0TK4i.

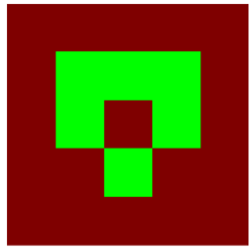


Figure 9a) Cell configuration at $t = 0$

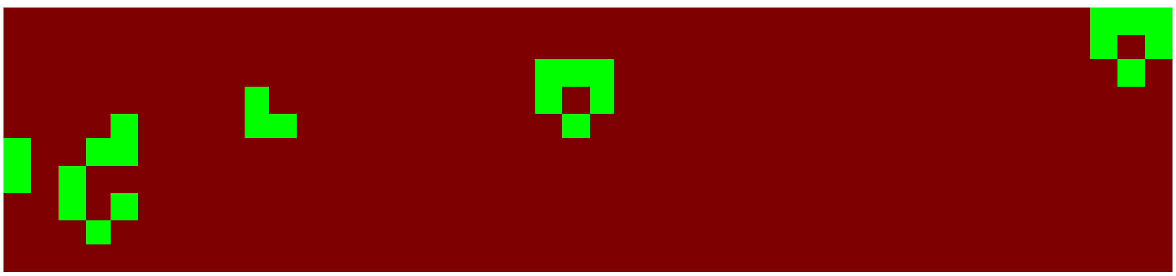


Figure 9b) Replication of the initial state at $t = 91$

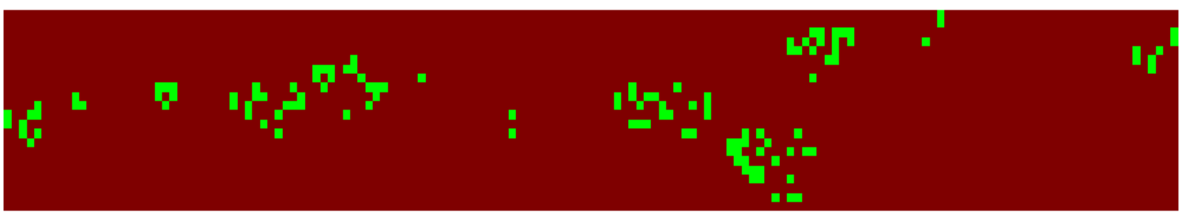
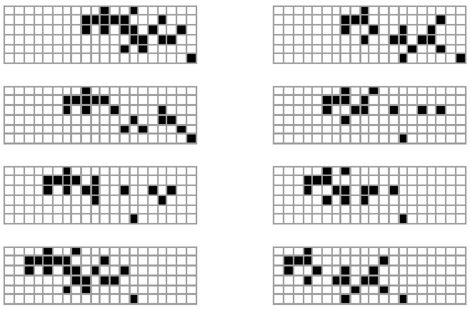
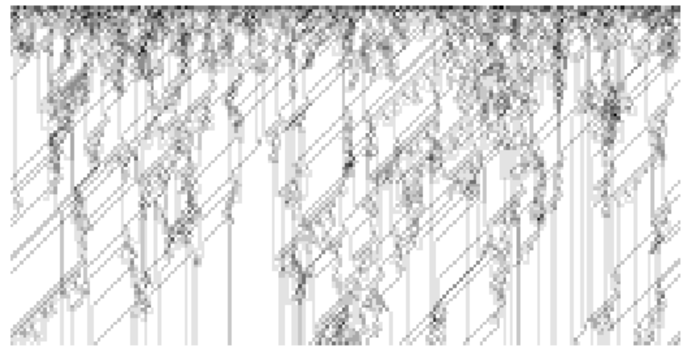


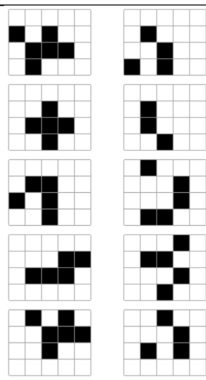
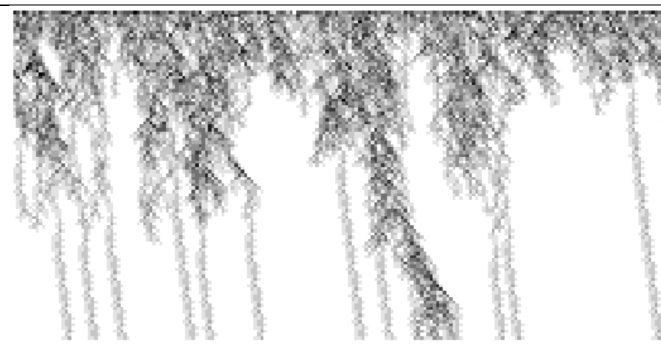
Figure 9c) Persistence of the initial pattern and its copy at $t = 307$

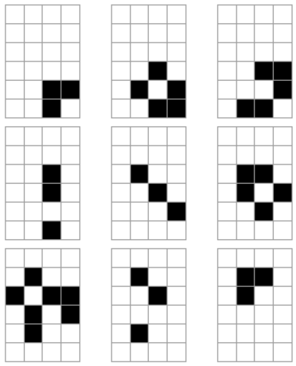
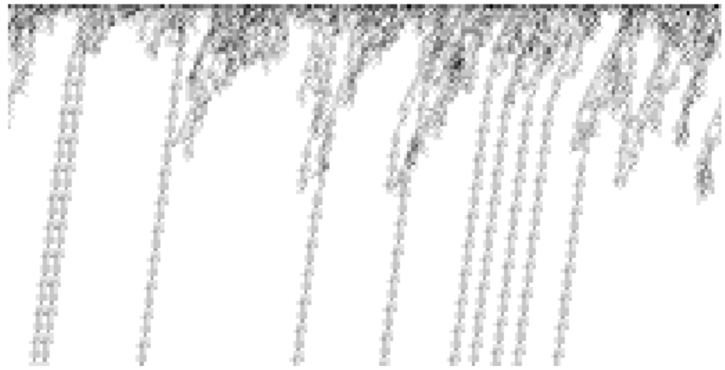
We present some examples of complex cellular automata found with the proposed genetic search algorithm[2].

[2] One can execute this cellular automata in *Mathematica* replacing <Rule> with the corresponding rule numbering

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

Rule 3548304374306973073146580452806499228996536072371527830887333950738508017529182 495350888208536558646807291895409639977375947662461701121698674406862032456	
 **Identified Gliders**	 **Averaged Spacetime Slice**

Rule 1969281128035673510785095133179477763137176390096291923341939230372336458567806 01181782252315349164603950024916004629851769274774088586292232688540354568	
 **Identified Gliders**	 **Averaged Spacetime Slice**

<table>
<tr><td colspan="2">Rule
2536962858330445998944606509915061353621502280763765013218910019118617632623181
726351808015804669971129335990123389394577484439270322287946219773078676008</td></tr>
<tr><td>
Identified Glider</td><td>
Averaged Spacetime Slice</td></tr>
</table>

About the Authors

Acknowledgements

References

[1] S. Wolfram, "Cellular automata as models of complexity," *Nature Vol 311,* pp. 419-424, 1984.

[2] H. Zenil and E. Villareal-Zapata, "Asymptotic Behaviour and Ratios of Complexity in Cellular Automata," *International Journal of Bifurcation and Chaos,* vol. 23, no. 9, 3013.

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

[4] S. Wolfram, "Statistical Mechanics of Cellular Automata," *Review of Modern Physics,* vol. 55, no. 3, pp. 601-644, 1983.

[5] S. Wolfram, A New Kind of Science, Wolfram Media, 2002.

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

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