Complete Characterization of Structure of Rule 54

Genaro J. Martínez
Escuela Superior de Cómputo, Instituto Politécnico Nacional, México
and
Unconventional Computing Center, Computer Science Department
University of the West of England, Bristol BS16 1QY, United Kingdom
genaro.martinez@uwe.ac.uk

Andrew Adamatzky
Unconventional Computing Center, Computer Science Department
University of the West of England, Bristol BS16 1QY, United Kingdom
andrew.adamatzky@uwe.ac.uk

Harold V. McIntosh
Departamento de Aplicación de Microcomputadoras
Instituto de Ciencias, Universidad Autónoma de Puebla, Puebla, México
mcintosh@unam.mx

The dynamics of the rule 54 one-dimensional, two-state cellular automaton (CA) are a discrete analog of a space-time dynamics of excitations in a nonlinear active medium with mutual inhibition. A cell switches its state 0 to state 1 if one of its two neighbors is in state 1 (propagation of a perturbation), and a cell remains in state 1 only if its two neighbors are in state 0. A lateral inhibition is because a 1-state neighbor causes a 1-state cell to switch to state 0. The rule produces a rich spectrum of space-time dynamics, including gliders and glider guns just from four primitive gliders. We construct a catalog of gliders and describe them by tiles. We calculate a subset of regular expressions Ψ_{R54} to encode gliders. The regular expressions are derived from de Bruijn diagrams, tile-based representation of gliders, and cycle diagrams sometimes. We construct an abstract machine that recognizes regular expressions of gliders in rule 54 and validate Ψ_{R54}. We also propose a way to code initial configurations of gliders to depict any type of collision between the gliders and explore self-organization of gliders, formation of larger tiles, and soliton-like interactions of gliders and computable devices.

1. Preliminaries

Cellular automata (CAs) are renowned for the simplicity of their rules and the complexity of their space-time evolution. Rule 54 is among the most famous rules that exhibit nontrivial space-time dynamics.

Complex Systems, **23** © 2014 Complex Systems Publications, Inc.
https://doi.org/10.25088/ComplexSystems.23.3.259

The rule belongs to complexity class IV in Wolfram's classification [1, 2].

Rule 54 has always attracted considerable interest from computer scientists, mathematicians, and physicists, and thus, compared to other elementary cellular automaton (ECA) rules, is well investigated. Boccara et al. [3] enumerated a number of gliders in rule 54 and characterized a glider gun. They applied statistical analysis to study the stability of gliders. Hanson and Crutchfield [4] introduced a concept of "computational mechanics," or designing of finite-state machines derived from language representations and motion equations of filtered gliders. Another exploration of rule 54 with automatic filters was presented by Wuensche in [5]. Wolfram [6] exhibited glider collisions with long periods of after-development and several filters for detecting gliders and defects, and Martin [7] designed an algebraic group of order four to represent collisions between basic gliders. A number of new glider guns, self-organization by structures, collisions, and glider-based logic gates were reported in [8]. Guan [9] develops a description of rule 54 dynamics with Bernoulli shift and symbolic sequences. Redeker [10] discusses how a flexible time can be represented in evolutions of rule 54. An exhaustive analysis about solitons in rule 54 was presented in [11], and a projection of rule 54 affected with memory was studied in [12]. Initial analysis of glider representation with rule 54 by de Bruijn and cycle diagrams was given in [13] (see also [14]).

2. Rule 54

A one-dimensional cellular automaton (CA) is represented by an infinite array of cells x_i where $i \in \mathbb{Z}$ and each x takes a value from a finite alphabet Σ. Thus, a sequence of cells $\{x_i\}$ of finite length n represents a string or *global configuration* c on Σ. The set of finite configurations, represented as Σ^n, is denoted by Φ. The CA evolution is given by a sequence of configurations $\{c_i\}$ on Φ:

$$\Phi\left(c^t\right) \to c^{t+1}, \tag{1}$$

where t is time and every global state of c is defined by a sequence of cells. Also the cells of each configuration c^t are updated at the next configuration c^{t+1} simultaneously by a local function φ as follows:

$$\varphi\left(\ldots, x_{i-1}^t, x_i^t, x_{i+1}^t, \ldots\right) \to x_i^{t+1}. \tag{2}$$

A one-dimensional CA can be described by two parameters (k, r) [1], where $k = |\Sigma|$ is a number of states and r is a neighborhood radius. ECAs are defined by parameters $(k = 2, r = 1)$.

https://doi.org/10.25088/ComplexSystems.23.3.259

In all constructs described in the paper, we apply *periodic boundary conditions* to obtain finite configurations of Φ by concatenating the first cell with the last one to form a ring.

The local transition function φ of ECA rule 54 follows:

$$\varphi_{R54} = \begin{cases} 1 & \text{if} \quad 101, 100, 010, 001 \\ 0 & \text{if} \quad 111, 110, 011, 000. \end{cases}$$

The binary sequence 00110110 represents rule number 54 in decimal notation. Initially φ_{R54} presents an initial probability of 50% to each state, and thus the frequency to appear is the same.

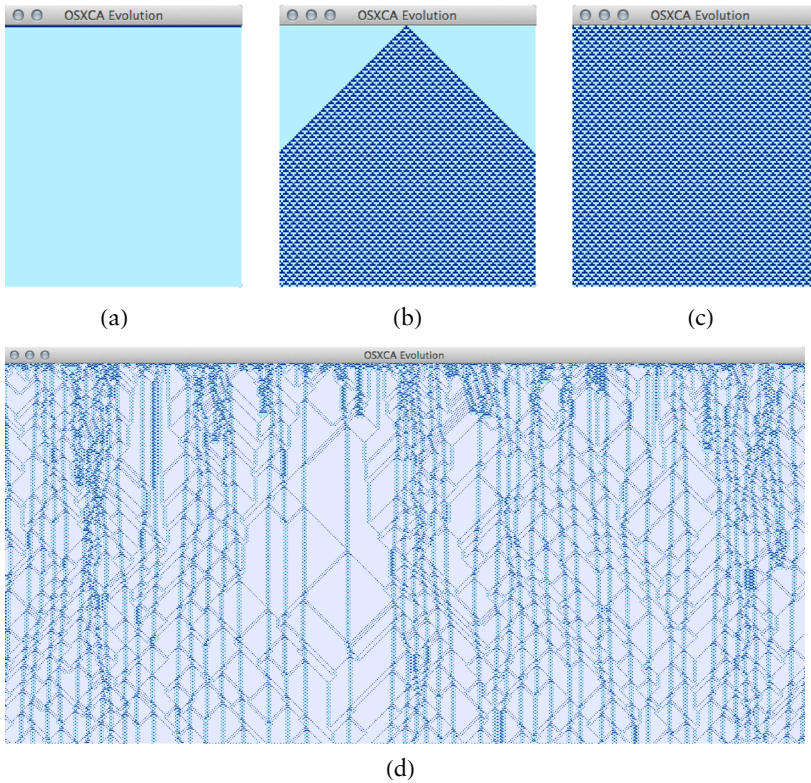


Figure 1. Exemplar scenarios of space-time evolution in rule 54 with 100 cells for 100 generations. (a) A single state of Σ dominates the initial condition. (b) A single cell is in state 1; all other cells are in state 0. (c) Periodic background. (d) Random initial condition with an initial density of 50% on 1000 cells for 500 generations (a filter is applied).

Figure 1 displays some typical snapshots with rule 54. We have chosen classic or specific initial conditions in order to capture different be-

https://doi.org/10.25088/ComplexSystems.23.3.259

haviors. Indeed, this set of figures can represent several CA classes: (a) could represent class I with a uniform evolution, (b) and (c) class II with periodic evolutions, and (d) class IV with complex dynamics.

Rule 54 is a discrete analog of an active, nonlinear, one-dimensional medium. Assume each cell is a micro-volume, which takes two states: resting (0) and excited (1). When a single micro-volume is perturbed, its corresponding cell takes state 1. The perturbation/activation spreads to neighbors of the initially excited micro-volume: $\{100, 001, 101\} \to 1$. For example, the transition $100 \to 1$ encodes the following activation mechanism: if the left neighbor of a resting cell is excited, the resting cell excites. Transition $000 \to 0$ indicates the simple fact that the medium could not activate itself; that is, excitation cannot develop from a totally resting medium.

The three most interesting transitions are $\{111, 110, 011\} \to 1$. They encode the following fact: if an excited micro-volume has at least one excited neighboring micro-volume, then this micro-volume returns to a resting state. This can be interpreted as a mutual inhibition. Each excited micro-volume inhibits excitation of its excited neighboring micro-volume. These features of rule 54 make it also interesting from neurophysiology and machine vision (one-dimensional artificial retina) points of view.

3. Representation of Gliders in Rule 54

This section discusses approaches toward a description of gliders in rule 54. These approaches use tiling theory [15], de Bruijn diagrams [16, 17], and cycle diagrams [18].

3.1 Tiles in Rule 54

Gliders in rule 54 can be represented by polygons as rhomboids. The periodic background, called ether, is represented in rule 110 for a family of triangles [19, 20], although with some differences from rule 54.

A *plane of tiles* $\mathcal{T}$ is a countable family of closed sets $\mathcal{T} = \{T_0, T_1, \ldots\}$ covering the plane without intervals or intersections [15] (the "plane" is the Euclidian plane $\mathbb{Z}\times\mathbb{Z}$ in elementary geometry). Therefore, this can be defined as a join of sets (called a mosaic $\mathcal{T}$):

$$\mathcal{T} = \bigcup_{i=0}^{n} T_i \,\forall\, n \in \mathbb{Z}_0^+; \tag{3}$$

consequently, every set is disjoint $T_i \cap T_j$. Thus, the set of tiles for rule 54 is represented as $\mathcal{T}_{R54}$. Figure 2 displays a number of mosaics of $\mathcal{T}_{R54}$.

https://doi.org/10.25088/ComplexSystems.23.3.259

Table 1 shows relations between tiles in rule 54. A row represents the tile type and a column represents the size of a tile. There are a limited number of kinds but an infinite number of sizes.

Rule 54 can be studied as a tiling problem (as was proposed in rule 110 by McIntosh [19, 21]). Figure 2 shows the relation of $\mathcal{T}_{R54}$ for the first nine polygons, which is summarized in Table 1. If one relation is missing, this means that such a polygon cannot be constructed for rule 54.

Figure 3 displays the composition for the periodic background in rule 54. It is composed of two tiles: T_1 and T_3^{α}. Arrows indicate possible directions in which a glider could emerge. Solid arrows display known gliders and dotted arrows display possible unknown gliders.

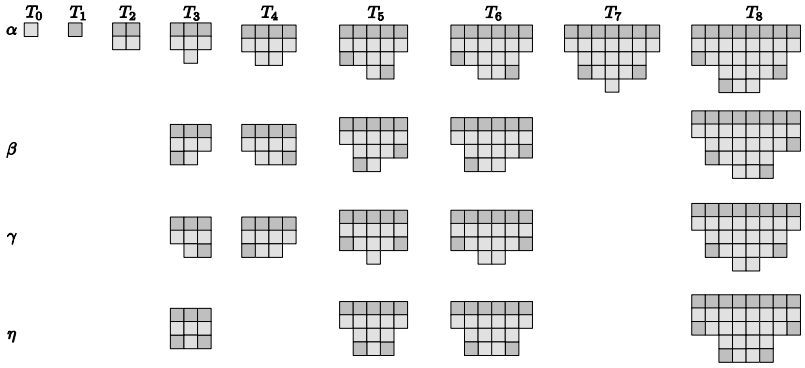


Figure 2. Examples of tiles $\mathcal{T}_{R54}$ derived from space-time configurations in rule 54.

T	0	1	2	3	4	5	6	7	8	⋯
α	✓	✓	✓	✓	✓	✓	✓	✓	✓	⋯
β				✓	✓	✓	✓		✓	⋯
γ				✓	✓	✓	✓		✓	⋯
η					✓	✓	✓		✓	⋯

Table 1. Relation of tiles $\mathcal{T}_{R54}$ in rule 54.

https://doi.org/10.25088/ComplexSystems.23.3.259

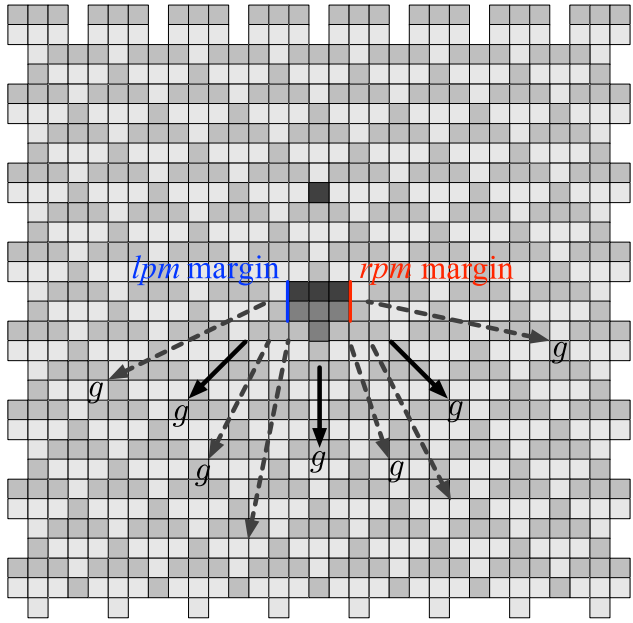


Figure 3. Periodic background in rule 54 is composed with two tiles of the set $\mathcal{T}_{R54}$: T_1 and T_3^{α}.

3.2 Gliders in Rule 54

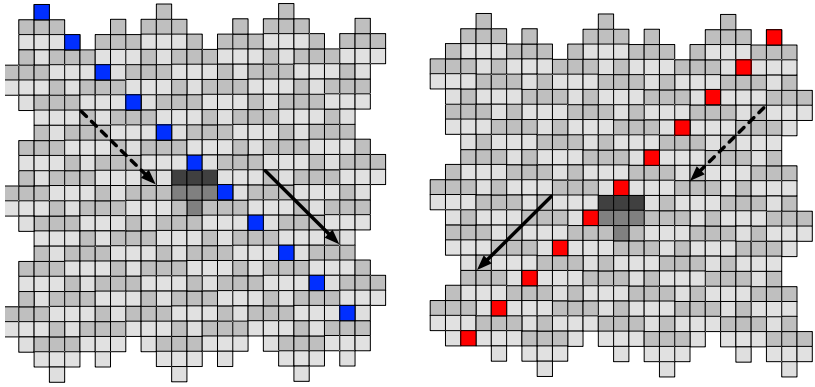
A *glider* is a compact group of nonquiescent states traveling along the CA lattice. To represent gliders in rule 54, we follow the notation of Boccara et al. [3].

Rule 54 has two identical primitive gliders traveling in opposite directions: $\overrightarrow{w}$ glider (Figure 4(a)) and $\overleftarrow{w}$ glider (Figure 4(b)), traveling with the speed of light, that is, translating one cell per iteration. Two stationary gliders can be interpreted as still life configurations in one dimension. They are gliders g_o (Figure 5(a)) and g_e (Figure 5(b)).

This way, we can display each glider, enumerating their properties. Figure 6 gives a systematic representation of gliders in rule 54, and Table 2 summarizes most basic properties.

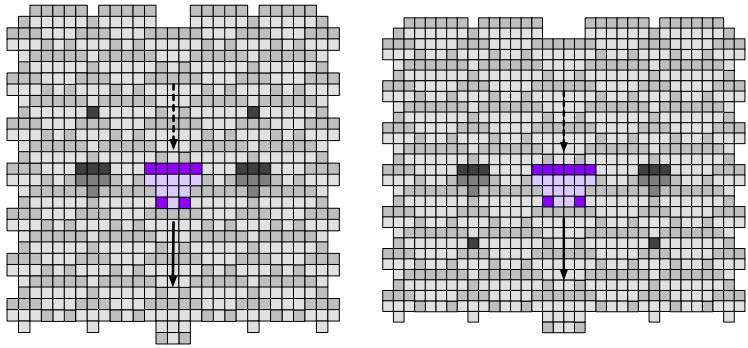
Let e_1 and e_2 represent glider phases in the periodic background. Thus we have four gliders: $\overrightarrow{w}$, $\overleftarrow{w}$, g_o, g_e; and a compound glider: the glider gun. The speed v_g of a glider is evaluated using the period between displacements. Column Cap in Table 2 shows if a glider is able to cover the full space without gaps.

https://doi.org/10.25088/ComplexSystems.23.3.259



(a) (b)

Figure 4. Tiles description for primitive gliders in rule 54.



(a) (b)

Figure 5. Tiles description of composed gliders in rule 54. Both gliders are still life configurations because they are stationary.

Structure	v_g	Lineal Volume	Cap
e_1	2 / 2 = 1	4	T
e_2	2 / 2 = 1	4	T
$\overrightarrow{w}$	2 / 2 = 1	2	P
$\overleftarrow{w}$	−2 / 2 = −1	0–4	P
g_o	0 / 4 = 0	6–2	T
g_e	0 / 4 = 0	7–3	T
glider gun	0 / 32 = 0	14–4	P

Table 2. Properties of gliders in rule 54.

https://doi.org/10.25088/ComplexSystems.23.3.259

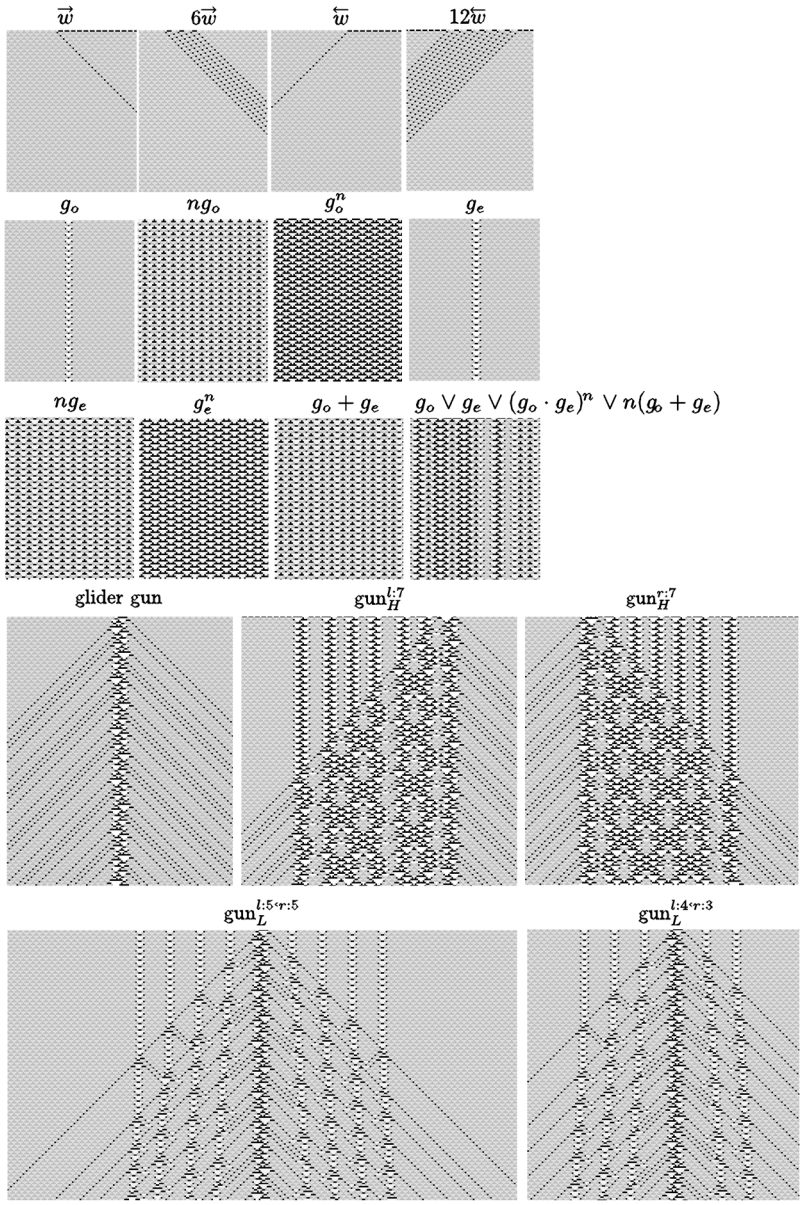


Figure 6. Classification of gliders in rule 54. We illustrate every glider and packages, extensions, and compositions of them.

https://doi.org/10.25088/ComplexSystems.23.3.259

Rule 54 exhibits a relatively small number of gliders, which makes it particularly attractive for discretization and formal representation. We can obtain an exact representation of gliders in rule 54 and show how to construct specific initial conditions based on glider phases. A *phase* means a unique string that represents the glider in the initial condition. Therefore, a finite number of different strings represent the set of valid strings where a glider can be initialized [22].

3.3 De Bruijn Diagram

For a one-dimensional CA of order (k, r) and a finite alphabet given Σ, its de Bruijn diagram is defined as a directed graph with k^{2r} vertices and k^{2r+1} edges. Vertices are labeled with elements of the alphabet of length $2r$, that is, neighborhood states. An edge is directed from vertex i to vertex j if and only if the $2r-1$ final symbols of i are the same as $2r-1$ initial symbols in j, forming a neighborhood of $2r+1$ states represented by $i \diamond j$. In this case, the edge connecting i to j is labeled by $\varphi(i \diamond j)$ (the value of the neighborhood defined by the local function) [23, 26].

Thus, the de Bruijn diagram can be constructed as follows:

$$M_{i,j} = \begin{cases} 1 & \text{if } j = k\,i,\, k\,i+1,\, \ldots,\, \mathrm{ki}+k-1 \left(\mathrm{mod}\, k^{2r}\right) \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Module $k^{2r} = 2^2 = 4$ represents the number of vertices in the de Bruijn diagram, and j takes values from $k * i = 2\,i$ to $(k * i) + k - 1 = (2 * i) + 2 - 1 = 2\,i + 1$. The vertices (indexes of M) are labeled by fractions of neighborhoods beginning with 00, 01, 10, and 11; the overlap determines each connection. Figure 7 displays rule 54's matrix evolution and its de Bruijn diagram.

Paths in the de Bruijn diagram may represent chains, configurations, or classes of configurations in the evolution space. Also, fragments of the diagram itself are useful in discovering periodic blocks of strings, pre-images, codes, and cycles [17, 23].

After the de Bruijn diagram is completed, we can calculate an extended de Bruijn diagram [17]. An extended de Bruijn diagram takes into account more significant overlapping of neighborhoods. Thus, we represent $M^{(2)}_{R54}$ by indexes $i = j = 2\,r * n$, where $n \in \mathbb{Z}^+$. The de Bruijn diagram grows exponentially, order k^{2r^n}, for each $M^{(n)}_{R54}$; the basic de Bruijn diagram is obtained for $n = 1$ (Figure 7).

https://doi.org/10.25088/ComplexSystems.23.3.259

$$M_{R54} = \begin{bmatrix} 0 & 1 & . & . \\ . & . & 1 & 0 \\ . & . & 0 & 0 \end{bmatrix}$$

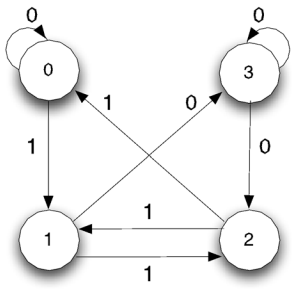


Figure 7. De Bruijn diagram of rule 54.

An important indication derived from de Bruijn diagrams is that the set of regular expressions Ψ_{R54} describes all possible strings to initialize gliders in rule 54. Of course, this representation does not include codes to initialize packages or groups of gliders. Therefore, the number of sequences w in the set Ψ_{R54} is the union of the periods for every glider, as follows:

$$\Psi_{R54} = \bigcup_{i=1}^{p} w_{i,g} \; \forall \; (w_i \in \Sigma^* \wedge g \in G), \tag{5}$$

where $\mathcal{G}$ is the whole set of gliders in rule 54 and $p > 0$ its period. This way, we can speak of a regular language L_{R54} that is constructed from the expressions of Ψ_{R54}. We notice that this language is a subset of the whole language in rule 54, because it is defined by regular expressions derived from gliders. Therefore, the regular language L_{R54} is defined as follows:

$$\begin{aligned} L_{R54} = \{w \mid w \text{ operating under the basic rules}: \; \cdot, \\ +, \; * \text{ from } \Psi_{R54}\}. \end{aligned} \tag{6}$$

Let us calculate de Bruijn diagrams for gliders $\overrightarrow{w}$ and $\overleftarrow{w}$ with periodic background. Table 2 shows that these gliders move two cells at each time step. Then the extended de Bruijn diagram of order $M_{R54}^{(2)}$ would be necessary to extract a cyclic structure of gliders (all extended de Bruijn diagrams are calculated with NXLCAU21, free software developed by H. V. McIntosh [24]). These diagrams can show all possible relations, but cycles are important for us to detect gliders or other periodic patterns.

Figure 8 displays de Bruijn diagrams with shift registers to the right (+) or to the left (−). A glider can be identified as a cycle, and the glider's interaction will be a connection with other cycles. Diagram (2, 2) (x-displacements, y-generations), displays periodic strings moving two cells to the right in two time steps, that is, the period between dis-

https://doi.org/10.25088/ComplexSystems.23.3.259

placement in the periodic background of a $\vec{w}$ glider. This way, we can enumerate each string for every structure in this domain.

- Periodic background e is fixed as:
  - vertices (1, 2, 4, 6) $\equiv e_1 = 1000$
  - vertices (13, 11, 7, 14) $\equiv e_2 = 1110$
- $\vec{w}$ glider is placed as:
  - vertices (1, 2) $\equiv \vec{w}_1 = 10$
  - vertices (12, 6) $\equiv \vec{w}_2 = 00$
- $\overleftarrow{w}$ glider is placed as:
  - vertices (1, 3, 7, 14) $\equiv \overleftarrow{w}_1 = 1110$
  - vertices (13, 10, 4, 8) $\equiv \overleftarrow{w}_2 = 1000$

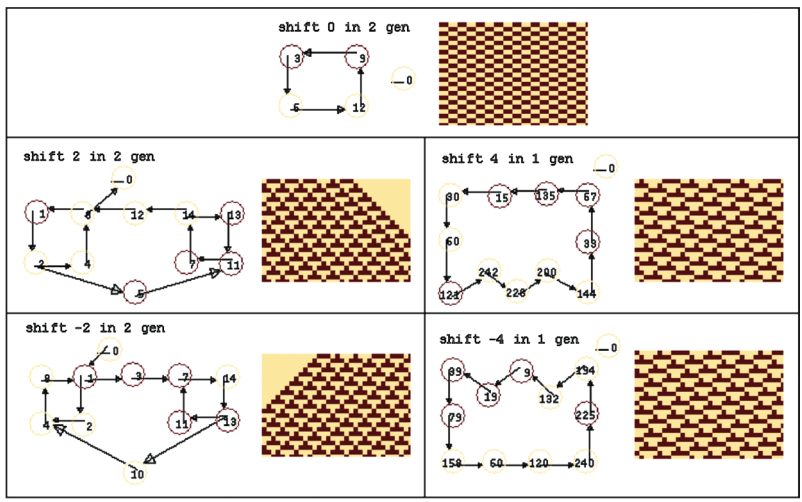


Figure 8. De Bruijn diagrams determining primitive gliders, periodic background, and other meshes (0, 2), (4, 1), (−4, 1) in rule 54.

The periodic background in phase one represents the string 1000 and in phase two the string 1110. Also, this diagram has a positive orientation of cycles and shows the relations of vertices (1, 2, 4, 6) and (13, 11, 7, 14) that represent all possible phases where a $\vec{w}$ glider may be initialized. However, the existence of this glider is related to both cycles.

Diagrams (0, 2), (4, 1), and (−4, 1) display three different periodic backgrounds that cannot coexist with gliders but can cover the whole evolution space.

https://doi.org/10.25088/ComplexSystems.23.3.259

Rule 54 has a particular characteristic because the periodic background needs a displacement to preserve the existence of gliders. Figure 8 shows four cycles, three of them self-contained and one that starts with a stable state. Evolution fragments in the same picture show what kinds of gliders are defined by these cycles. For example, we can see a large cycle following the vertices (1, 2, 5, 11, 13, 14, 12, 6). This cycle is equivalent to the periodic string 10111000, which produces an evolution space covered with just a pair of $\overrightarrow{w}$ gliders. Finally, a fourth cycle, represented by the cycle 0, determines a transition between two different patterns, known as "fuse configurations" [17]. The periodic background is formed by a cycle of length four, and the existence of gliders is determined by other cycles. Therefore, we can see that the problem of representing gliders by de Bruijn diagrams is reduced to the classification of cycles.

To represent gliders g_o and g_e, we should construct de Bruijn diagrams of order $M_{R54}^{(4)}$, because the gliders have period 4 (see Table 2). These gliders can be considered as still life configurations because they are stationary structures.

Figure 9 shows the full de Bruijn diagram (0, 4) used to calculate g_o and g_e gliders. There are four main cycles: two largest cycles represent phases of g_o and g_e plus its periodic background, and two smaller cycles characterize two different periodic patterns in rule 54, including the stable state.

Again, to extract phases we shall follow routes in the diagram and enumerate all the routes, that is, their regular expressions. The larger cycles contain internal cycles that represent each glider phase. So the periodic background is represented by two cycles and they relate all possible phases for g_o and g_e gliders.

The left cycle in the diagram of Figure 9 represents the whole phases of gliders g_o and g_e with the periodic background e_1 and vertices (17, 34, 68, 136), and the right cycle represents phases with the periodic background e_2 and vertices (221, 187, 119, 238). Therefore, we can extract periodic sequences to encode gliders traveling alone or in trains of gliders. Encoding samples are provided with some strings:

- String 1010001001000 encodes g_o-g_e, where both gliders are in phase three (f_3) and have periodic background in phase one (e_1) (Figure 10(a)).
- String 111000 encodes g_o glider in phase four (f_4) with a periodic background in phase two (e_2) (Figure 10(b)).
- String 10000010 encodes trains of two g_o gliders covering the whole evolution space. To reach this configuration it is necessary to use two different phases, f_1 and f_3 of g_o glider (Figure 10(c)).

https://doi.org/10.25088/ComplexSystems.23.3.259

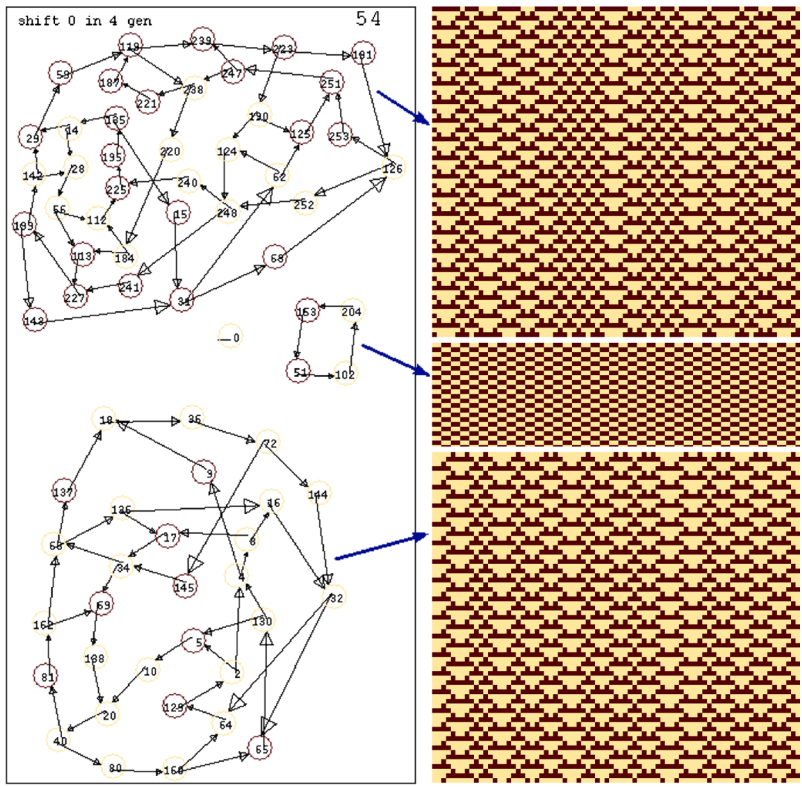


Figure 9. De Bruijn diagram representing stationary gliders.

- String 111111011111000011100 0 produces a sophisticated pattern with singular and compound gliders and periodic background g_e-g_o-g_e-e_2-g_o (Figure 10(d)).

Thus, we can calculate systematically all periodic patterns for each (x, y) position in the de Bruijn diagrams. Figure 11 shows the full evolutions to 10 generations. Indeed, symmetries are preserved during its evolutions with displacements, and some positions are dominated by the stable state. Of course, we can find the periodic background and basic gliders in several positions where they match with this period and other interesting periodic patterns that emerge in rule 54.

The complete description of regular expressions representing gliders in rule 54 is provided in Table 3. This set of regular expressions is implemented in the OSXLCAU21 system [25].

https://doi.org/10.25088/ComplexSystems.23.3.259

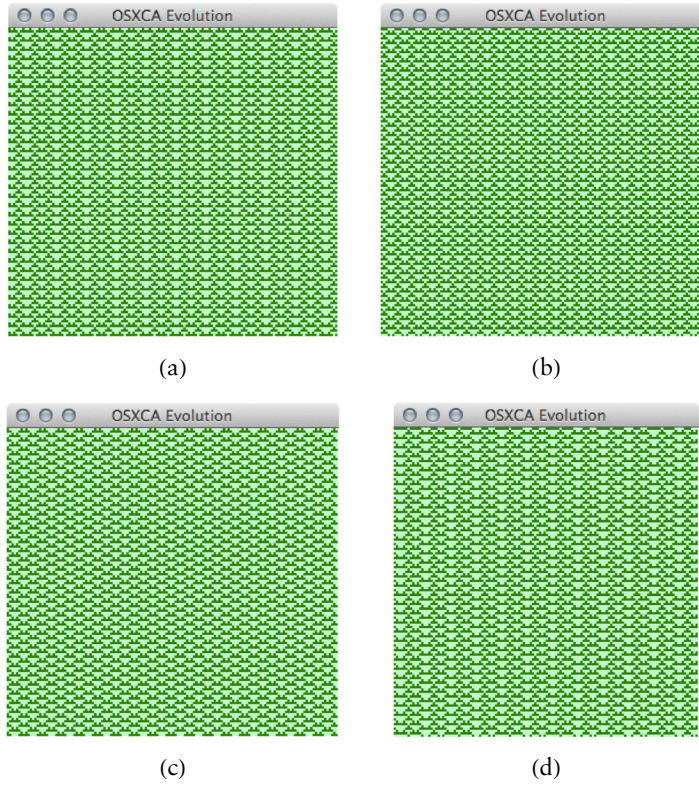


Figure 10. Evolutions encoded from cycles in the de Bruijn diagram shown in Figure 9.

e	$\overrightarrow{w}$	$\overleftarrow{w}$
$e_1 = 1000$ $e_2 = 1110$	$\overrightarrow{w}(\mathrm{f}_1) = e_1\text{-}10\text{-}e_2$ $\overrightarrow{w}(\mathrm{f}_2) = e_2\text{-}00\text{-}e_1$ $2\,\overrightarrow{w}(\mathrm{f}_1) = e_1\text{-}10111000\text{-}e_1$	$\overleftarrow{w}(\mathrm{f}_1) = e_1\text{-}e_2$ $\overleftarrow{w}(\mathrm{f}_2) = e_2\text{-}e_1$ $2\,\overleftarrow{w}(\mathrm{f}_1) = e_1\text{-}11101000\text{-}e_1$ $2\,\overleftarrow{w}(\mathrm{f}_2) = e_2\text{-}11101000\text{-}e_2$
	g_o	g_e
	$g_o(\mathrm{A},\mathrm{f}_1) = e_1\text{-}100000\text{-}e_1$ $g_o(\mathrm{A},\mathrm{f}_2) = e_2\text{-}111110\text{-}e_2$ $g_o(\mathrm{B},\mathrm{f}_1) = e_1\text{-}10\text{-}e_1$ $g_o(\mathrm{B},\mathrm{f}_2) = e_2\text{-}00\text{-}e_2$	$g_e(\mathrm{A},\mathrm{f}_1) = e_1\text{-}1000000\text{-}e_1$ $g_e(\mathrm{A},\mathrm{f}_2) = e_2\text{-}000\text{-}e_2$ $g_e(\mathrm{B},\mathrm{f}_1) = e_1\text{-}100\text{-}e_1$ $g_e(\mathrm{B},\mathrm{f}_2) = e_2\text{-}1111110\text{-}e_2$

Table 3(a). Set of regular expressions for gliders in rule 54.

https://doi.org/10.25088/ComplexSystems.23.3.259

Gun
$\mathrm{gun(A,f_1)} = e_1\text{-}1111111100\text{-}e_1$
$\mathrm{gun(A,f_2)} = e_2\text{-}1000000001\text{-}e_1$
$\mathrm{gun(B,f_1)} = e_1\text{-}11100000010010\text{-}e_2$
$\mathrm{gun(C,f_1)} = e_1\text{-}10001000011100\text{-}e_2$
$\mathrm{gun(C,f_2)} = e_2\text{-}010001\text{-}e_1$
$\mathrm{gun(D,f_1)} = e_1\text{-}1111010010\text{-}e_2$
$\mathrm{gun(D,f_2)} = e_2\text{-}1000011111\text{-}e_1$
$\mathrm{gun(E,f_2)} = e_2\text{-}11100100000010\text{-}e_2$
$\mathrm{gun(A2,f_1)} = e_1\text{-}10001111000011\text{-}e_1$
$\mathrm{gun(A2,f_2)} = e_2\text{-}10000100\text{-}e_1$
$\mathrm{gun(B2,f_1)} = e_1\text{-}111001111110\text{-}e_2$
$\mathrm{gun(C2,f_1)} = e_1\text{-}11000000\text{-}e_1$
$\mathrm{gun(C2,f_2)} = e_2\text{-}10010000\text{-}e_2$
$\mathrm{gun(D2,f_1)} = e_1\text{-}11111100\text{-}e_1$
$\mathrm{gun(D2,f_2)} = e_2\text{-}100000011110\text{-}e_2$
$\mathrm{gun(E2,f_1)} = e_1\text{-}111000010000\text{-}e_1$
$\mathrm{gun(A3,f_1)} = e_1\text{-}10011100\text{-}e_2$
$\mathrm{gun(A3,f_2)} = e_2\text{-}11110001\text{-}e_1$
$\mathrm{gun(B3,f_1)} = e_1\text{-}100001010010\text{-}e_2$
$\mathrm{gun(B3,f_2)} = e_2\text{-}01111111\text{-}e_1$
$\mathrm{gun(C3,f_1)} = e_1\text{-}110000000010\text{-}e_2$
$\mathrm{gun(C3,f_2)} = e_2\text{-}100100000011\text{-}e_1$
$\mathrm{gun(D3,f_1)} = e_1\text{-}1111110000\text{-}e_1$
$\mathrm{gun(D3,f_2)} = e_2\text{-}1000000100\text{-}e_2$
$\mathrm{gun(E3,f_1)} = e_1\text{-}1110000111\text{-}e_1$
$\mathrm{gun(E3,f_2)} = e_2\text{-}10001001000010\text{-}e_2$
$\mathrm{gun(A4,f_2)} = e_2\text{-}1111110011\text{-}e_1$
$\mathrm{gun(B4,f_1)} = e_1\text{-}10000001\text{-}e_1$
$\mathrm{gun(B4,f_2)} = e_2\text{-}00010010\text{-}e_2$
$\mathrm{gun(C4,f_1)} = e_1\text{-}10011111\text{-}e_1$
$\mathrm{gun(C4,f_2)} = e_2\text{-}111100000010\text{-}e_2$
$\mathrm{gun(D4,f_1)} = e_1\text{-}01000011\text{-}e_1$
$\mathrm{gun(D4,f_2)} = e_2\text{-}011100\text{-}e_1$
$\mathrm{gun(E4,f_1)} = e_1\text{-}110001\text{-}e_2$
$\mathrm{gun(E4,f_2)} = e_2\text{-}1001010000\text{-}e_1$

Table 3(b). Set of regular expressions for gliders in rule 54.

https://doi.org/10.25088/ComplexSystems.23.3.259

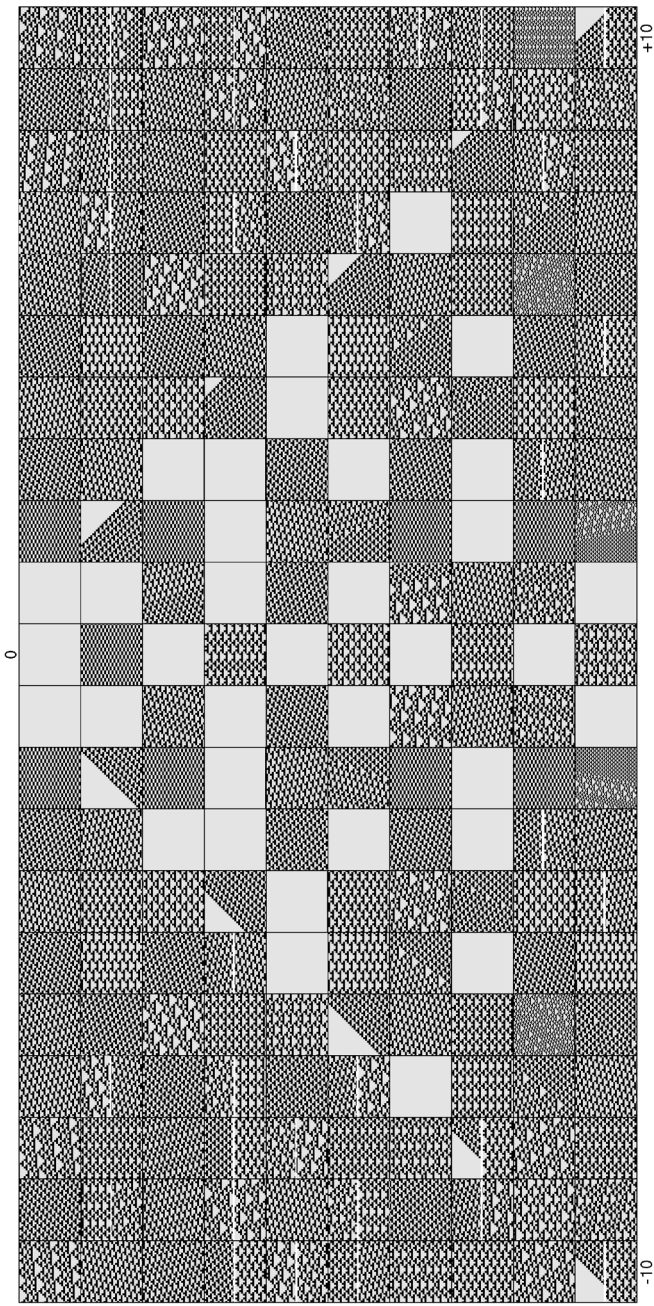


Figure 11. Periodic patterns in rule 54 calculated with extended de Bruijn diagrams for 10 generations. Each square (x, y) (small snapshot evolution) displays its respective pattern.

https://doi.org/10.25088/ComplexSystems.23.3.259

3.4 The Scalar Diagram in Rule 54

The scalar subset diagram is derived from the de Bruijn diagram. The scalar subset diagram represents an abstract machine to verify what sequences belong to the language produced by rule 54. Also the diagram can calculate *Garden of Eden* configurations and other properties, as was demonstrated by McIntosh in [16, 17]. A Garden of Eden configuration is a configuration that cannot be achieved from any other configuration in the evolution of a CA. This is a configuration without ancestors.

The subset diagram has $2^{k^{2r}}$ vertices. If all the configurations of a certain length have ancestors, then all extended (with additional cells added on both ends) configurations must have ancestors. Otherwise, they describe configurations in the Garden of Eden and represent paths going from the maximum set to the minimum one.

Nodes are grouped into subsets. A node should be composed of the subsets that can be arrived at through systematic departures from all the nodes in any given subset. The result is a new graph, with subsets for nodes and links summarizing all the places that can be traveled to from all the different combinations of starting points. Sometimes, but far from always, the possible destinations narrow down as progress is made; in any event, all the possibilities have been cataloged.

Let us define the subset diagram following [16, 17]. Let a and b be vertices, S a subset, and $|S|$ the cardinality of S. Then the subset diagram is defined as follows:

$$\sum_i(S) = \begin{cases} \phi & S = \phi \\ \{b \mid \mathrm{edge}_i(a, b)\} & S = \{a\} \\ \bigcup_{a \in S} \Sigma_i(a) & |S| > 1. \end{cases} \tag{7}$$

Three important properties are given here:

1. If there is a path from the maximum subset to the minimum one, then there exists a similar path starting from some smaller subset to the empty one. On the other hand, if all the unitary classes do not have edges going to the empty set, then there are no configurations in the Garden of Eden.
2. Given an origin and a destination, there is always a subset containing the accessible destination and another subset containing the origin; also, the destination can have additional vertices.
3. The subset diagram is not connected, and it is interesting to know the accessible greatest subset as well as the smallest one from a given subset.

The important convention in constructing the diagram is that if it seems there should be a link toward a certain node and if there is no such link, the link must be drawn to the empty set instead. This con-

https://doi.org/10.25088/ComplexSystems.23.3.259

vention assures every label of having a representation at every node in the subset diagram.

Vertices of the subset diagram are formed by the combination of each subset formed from the states of the de Bruijn diagram (a power set). Below we discuss the de Bruijn diagram—expressing the local function φ—symbolized in two matrices [17].

Symbolic de Bruijn matrices $D_{k,s}$ or D_s are characterized by k states and s number of states in the partial neighborhood. Thus for rule 54 we have the following symbolic matrices:

$$D_{2,2} = \begin{bmatrix} 0 & 1 & . & . \\ . & . & 1 & 1 \\ 0 & 1 & . & . \\ . & . & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & . & . & . \\ . & . & . & . \\ 0 & . & . & . \\ . & . & . & 0 \end{bmatrix} + \begin{bmatrix} . & 1 & . & . \\ . & . & 1 & 1 \\ . & 1 & . & . \\ . & . & 1 & . \end{bmatrix}.$$

Therefore, for any ECA order (2, 1) we have four sequences of states in the de Bruijn diagram enumerated as {0}, {1}, {2}, and {3}. All the possible subsets are {0, 1, 2, 3}, {0, 1, 2}, {0, 1, 3}, {0, 2, 3}, {1, 3, 2}, {0, 1}, {0, 2}, {0, 3}, {1, 2}, {1, 3}, {3, 2}, {3}, {2}, {1}, {0}, and {}. In these subsets, four unitary classes can be distinguished; the incorporation of the empty set guarantees that all subsets have at least one image, although this one does not exist in the original diagram.

In order to determine the type of union between the subsets, the state in which each sequence evolves must be reviewed to know toward which states (subsets that form it) this subset can be connected; this way the relation for rule 54 is constructed in Table 4. The corresponding scalar subset diagram for rule 54 is shown in Figure 12.

Each connection is defined from its relation between subsets (see Table 4). We must distinguish four levels of subsets. Also, we should observe that a residual of the de Bruijn diagram can be founded in the subset diagram. This is because a unit class is precisely the nodes of the original diagram.

At first glance, we can see that some relations are more frequent than others. There are nodes without any inputs, or nodes with most types of connections including self-loops. However, more interesting are cycles of different lengths. They are important to recognize words or sequences that a CA could recognize, as a general machine for this language.

A small subset diagram may be deduced from its original diagram. This diagram shall include only vertices with cycles, the universal and empty set, and the subset of cardinality one, yielding a new diagram that will be more practical for our proposes. The reduction gives a yet smaller diagram, shown in Figure 12.

https://doi.org/10.25088/ComplexSystems.23.3.259

Subset	Node	Link with 0	Link with 1
0,1,2,3	15	9	14
1,2,3	14	9	14
0,2,3	13	9	6
0,1,3	11	9	6
0,1,2	7	1	14
2,3	12	9	6
1,3	10	8	12
1,2	6	1	14
0,3	9	9	6
0,2	5	1	2
0,1	3	1	14
3	8	8	4
2	4	1	2
1	2	0	12
0	1	1	2
ϕ	0	0	0

Table 4. Relation between states of the subset diagram in rule 54.

Once the subset diagram has been formed, if a path leads from the universal set to the empty set, that is conclusive evidence that such a path exists nowhere in the original diagram.

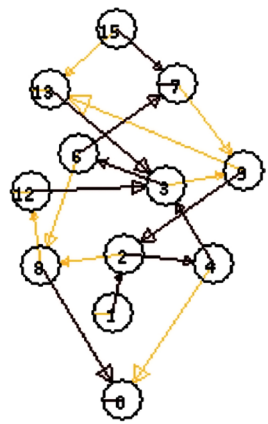


Figure 12. The scalar subset diagram of rule 54.

3.4.1 Garden of Eden Configurations in Rule 54

We know that the local function φ of rule 54 has an injective correspondence exploring its subset diagram. With this correspondence, we can find paths in the subset diagram representing Garden of Eden

https://doi.org/10.25088/ComplexSystems.23.3.259

configurations. In this manner, we can obtain two minimal configurations that calculate Garden of Eden configurations for rule 54, represented by the strings 101010 and 01010. Of course, concatenations and compositions of these strings will produce a more extended Garden of Eden configuration.

3.4.2 An Abstract Machine for Rule 54

A practical application of the subset diagram is that it can recognize any valid string in rule 54. Another way to verify if a string derived from the de Bruijn diagram, cycle diagram, or tiles representation is to evaluate such a string in the subset diagram, in the same way as regular language is recognized in classic automata theory [27, 28].

In order to verify this property, it is necessary to take a sequence from the set of regular expressions Ψ_{R54} and check for a route match into the subset diagram. Otherwise, if such a string does not follow any route, then it does not belong to L_{R54}.

4. Cycle Diagrams

Another way to get periodic structures in rule 54 is to calculate cycle diagrams (or attractors), similar to what Wuensche [18] did by deriving an ECA classification based in basins of attraction properties.

In this section, we explore some cases with particular evolutions or attractors.

Figure 13 (left) determines a cycle diagram for a configuration with 16 cells. This attractor has a root cycle of four states with a total of 6432 vertices. If you choose a leaf (vertex 50795), then it is the periodic configuration that will evolve during 32 generations to reach the attractor, which is precisely the periodic background configurations. Figure 13 (right) determines a cycle diagram for a configuration with 15 cells, it has an attractor with just one state, the stable state, that can be reached after 21 generations starting with the configuration vertex 11491, this attractor has 1583 vertices.

Figure 14 displays a basin of attraction for configurations with 23 cells. We show this attractor to demonstrate the complex behavior of rule 54. It is determined by asymmetric long-transient attractors. They imply the existence of gliders and nontrivial behavior.

https://doi.org/10.25088/ComplexSystems.23.3.259

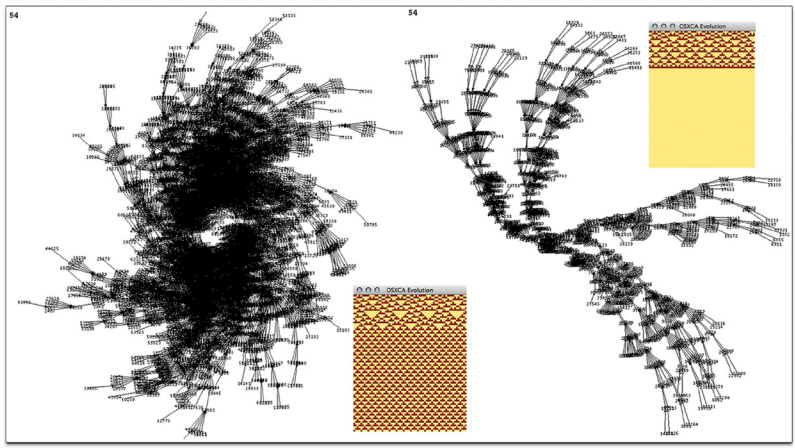


Figure 13. Cycle diagrams calculating periodic background from their attractors with $l = 16$ (left) and $l = 15$ (right).

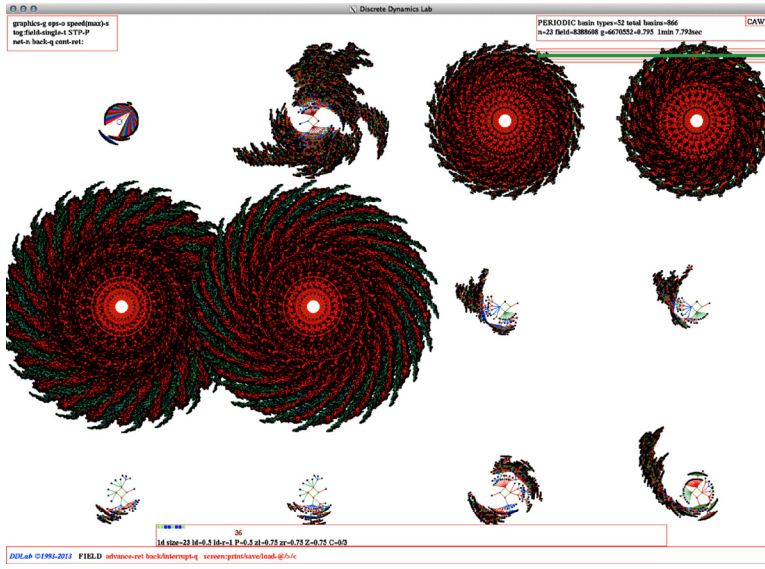


Figure 14. Basin of attractors in rule 54 for rings with 23 cells.

https://doi.org/10.25088/ComplexSystems.23.3.259

Figure 15 shows a "meta-glider," meshes, or agar configuration (an agar configuration comes from Conway's CA *Game of Life* literature, for details see [29]). The meta-glider in this figure is a periodic structure moving to the left (composed of one T_8, two T_4, and one T_2 tiles); it is preserved during a triple permanent collision of three $\overleftarrow{w}$ gliders. We have selected vertex 7577, which needs 20 generations to reach the attractor that represents this meta-glider, which corresponds to 169 vertices. The full attractor is composed of 1274 vertices.

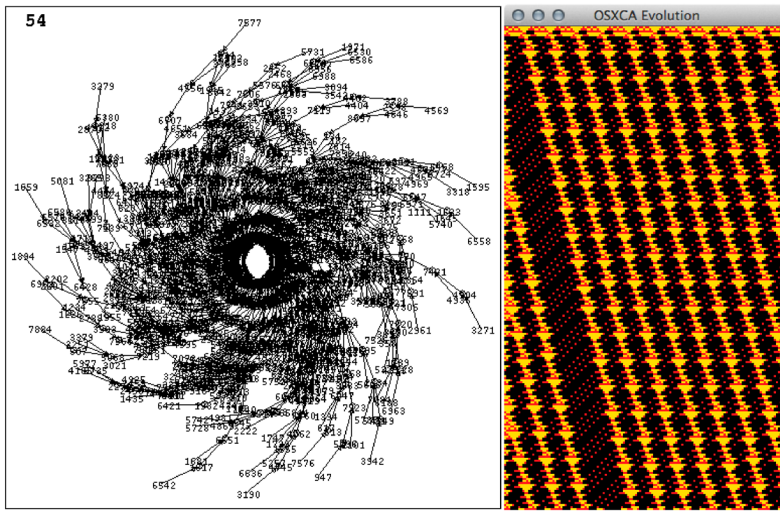


Figure 15. A cycle diagram of 13 cells calculating a meta-glider or agar configuration in rule 54.

For the following cycles or attractors diagrams we can list a number of periodic strings as well. In this case, every primitive glider may be reproduced from different cycles, as Table 5 shows. The Length column indicates the attractor period, the Cycle column indicates the number of components selected that have the same cycle length, the Total Vertices column is the total number of nodes for each attractor (including branches and leaves), and the Structures column describes the number of periodic structures evolving with these strings.

https://doi.org/10.25088/ComplexSystems.23.3.259

Length	Cycle	Total Vertices	Structures
4	4	4	T_3 andT_2 tiles
6	4	5	g_e glider
8	4	14	g_e gliders joined
	6	28	g_e glider with aT_2
9	4	44	g_e-g_o gliders joined
	27	45	T_4 transporting a$\overleftarrow{w}$ (extensible as a T_5 in rule 110)
10	30	90	two T_4 tiles joined
11	4	125	g_o glider with a T_6 tile
	11	55	packages of T_4 tiles
	99	231	meta-glider ($\vec{w}$-T_5-T_6-T_4-T_2 tiles)
12	10	124	periodic background ($2T_6$-$2T_3$-T_2 tiles)
	12	102	$2\vec{w}$ gliders
13	4	406	(g_e-g_o) gliders concatenated
	169	1274	meta-glider (T_8-$2T_4$-T_2 and $\vec{w}$ gliders)
14	112	805	meta-glider (T_8-$3T_4$-T_2 tiles)
15	330	7680	meta-glider (T_5-$2T_6$-T_4-T_2 tiles)
16	6	116	periodic background (T_6-T_2 tiles)
	8	8	$\vec{w}$ gliders
	14	944	meta-glider ($\vec{w}$-g_o-$\overleftarrow{w}$ gliders)
	16	2896	$2\vec{w}$ gliders
	40	1246	meta-glider (T_8-$5T_6$-$2T_2$-$3T_4$-T_5 tiles)

Table 5. Cycle diagrams calculating periodic structures in rule 54.

5. A Way to Encode Gliders in Rule 54

We can encode gliders in regular expressions via the gliders' phase representations:

$$\#_1\,(\#_2,\, p_i), \tag{8}$$

where $\#_1$ represents a glider of rule 54 of the set of gliders $\mathcal{G}_{R54}$, $\#_2$ represents its block of phases, and p_i is a phase determined for each block of phases, where $i = \{1, 2\}$. All sets of phases for gliders in rule 54 are detailed in Table 3.

The displacement for each glider g in $\mathcal{G}_{R54}$ is represented with the following equation:

$$d_g = 2 * l\,p\,m - 2 * r\,p\,m. \tag{9}$$

https://doi.org/10.25088/ComplexSystems.23.3.259

All periodic structures have a period length defined by the amount of margins $l\,p\,m$ and $r\,p\,m$, given the number of tiles and contact points in the structure (see Table 2). Therefore the period of gliders is determined as

$$p_g = 2 * l\,p\,m + 2 * r\,p\,m, \tag{10}$$

and the speed of gliders in rule 54 is determined as

$$v_g = \frac{2 * l\,p\,m - 2 * r\,p\,m}{2 * l\,p\,m + 2 * r\,p\,m}. \tag{11}$$

Collisions between gliders have a maximum level that is determined by the number of margins $l\,p\,m$ and $r\,p\,m$, although they could not all be viable collisions. This way, a glider with $l\,m\,s$ contact points and another glider with $r\,p\,m$ contact points have the next number of possible collisions:

$$c \leq lpm * r\,p\,m, \tag{12}$$

where c represents the maximum number of possible collisions.

Frequently, however, gliders have contact and noncontact points where the maximum level is not fulfilled. Simplifying the equation, we obtain the number of collisions between two gliders g_i and g_j, where $i \neq j$, which is represented by the following equation:

$$c = \left|\left(l\,p\,m_{g_i} * r\,p\,m_{g_j}\right) - \left(r\,p\,m_{g_j} * l\,p\,m_{g_i}\right)\right|. \tag{13}$$

Therefore, following is the set of regular expressions and codification in phases for gliders in rule 54 (see Table 3). We are able to codify easily the initial conditions to control and synchronize collisions between gliders. In the next sections we select some problems, such as construction of gliders by collisions, unlimited growth, holes, solitons, and some simple computable devices.

5.1 Self-Organization by Glider Reaction

In [8] we show how to construct all gliders in rule 54 from collisions between gliders. This problem is referred to as glider self-organization by collisions in complex systems [31]. Figure 16 displays the production of primitive gliders in rule 54, and Table 6 shows encoding of the collisions.

https://doi.org/10.25088/ComplexSystems.23.3.259

	Collisions	
Glider	By Gliders	By Sequences
$\vec{w}$	g_o,$\overleftarrow{w}$	$e_1{}^*0^{4n-2}\,e_2{}^*\ \forall\ n > 0$
$\overleftarrow{w}$	$\vec{w}$,g_o	$e_1{}^*0^{4n}\,e_2{}^*\ \forall\ n > 0$
g_o	$\vec{w}$,$\overleftarrow{w}$	$e_1{}^*10^n\,e_1{}^*\ \forall\ n > 0$ and odd
g_e		$e_1{}^*10^n\,e_1{}^*\ \forall\ n > 0$ and even
glider gun	$\vec{w}$,2 g_e or 2 g_e,$\overleftarrow{w}$	
glider gun^n	$\vec{w}$,g_e,2 g_e or 2 g_e,g_e,$\overleftarrow{w}$	

Table 6. Collision sequence for glider production in rule 54.

We can choose between production by gliders or by sequences. If we want to produce a $\vec{w}$ glider, then we need to collide a g_o glider with a $\overleftarrow{w}$ glider and so on. We enumerate each expression to reproduce every collision presented in Figure 16.

1. $\vec{w} = n\,e_1\text{-}\vec{w}\text{-}0^{10}\text{-}\overleftarrow{w}\text{-}n\,e_2$ (Figure 16(a)),
2. $\vec{w} = n\,e_1\text{-}(g_o(\mathrm{A},\mathrm{f}_1) \Vert\, g_o(\mathrm{B},\mathrm{f}_1))\text{-}10\,e_1\text{-}\overleftarrow{w}\text{-}n\,e_2$ (Figure 16(b)),
3. $\overleftarrow{w} = n\,e_1\text{-}\vec{w}\text{-}0^{12}\text{-}\overleftarrow{w}\text{-}n\,e_2$ (Figure 16(c)),
4. $\overleftarrow{w} = n\,e_1\text{-}\vec{w}\text{-}8\,e_2\text{-}(g_o(\mathrm{A},\mathrm{f}_1) \Vert\, g_o(\mathrm{B},\mathrm{f}_1))\text{-}n\,e_2$ (Figure 16(d)),
5. $g_o = n\,e_1\text{-}\vec{w}\text{-}0^{10}\text{-}\overleftarrow{w}\text{-}n\,e_1$ (Figure 16(e)),
6. $g_o = n\,e_1\text{-}\vec{w}\text{-}10\,e_2\text{-}\overleftarrow{w}\text{-}n\,e_1$ (Figure 16(f)),
7. $g_e = n\,e_1\text{-}\vec{w}\text{-}0^{12}\text{-}\overleftarrow{w}\text{-}n\,e_1$ (Figure 16(g)),
8. $\mathrm{gun} = n\,e_1\text{-}g_e(\mathrm{A},\mathrm{f}_1)\text{-}g_e(\mathrm{B},\mathrm{f}_1)\text{-}4e_1\text{-}\overleftarrow{w}\text{-}n\,e_2$ (Figure 16(h)),

where n is a number of copies of the string.

Of course, different parameters will yield a glider with different intervals or a different number of gliders.

https://doi.org/10.25088/ComplexSystems.23.3.259

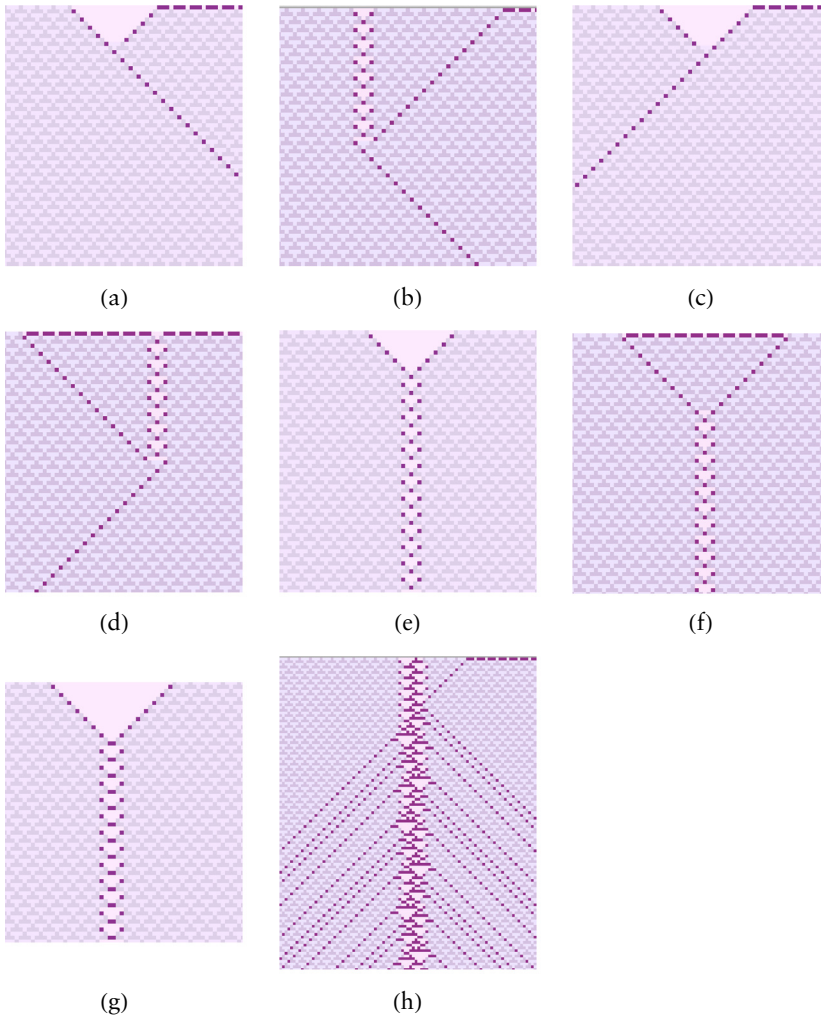


Figure 16. Producing gliders by collisions in rule 54.

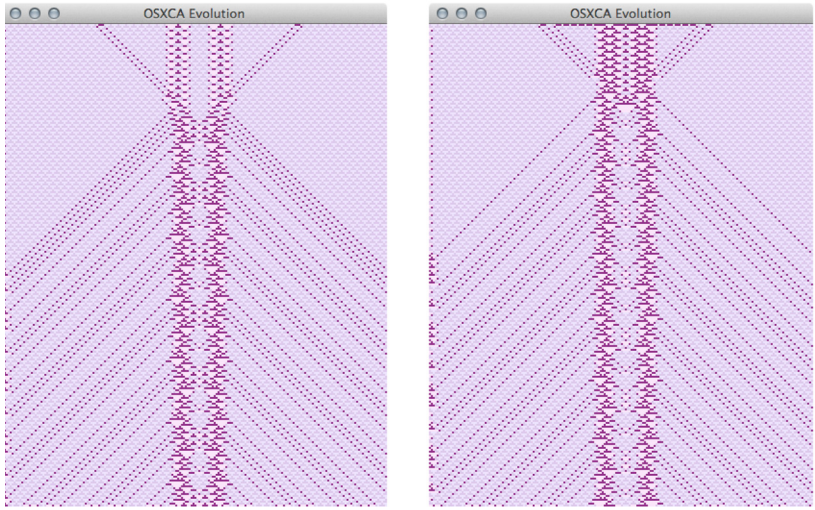
5.2 Unlimited Growth

A famous problem established in Conway's Game of Life was the discovery of a configuration that will grow permanently, into an infinite evolution space. This problem was solved by Gosper and colleagues at MIT Artificial Intelligence Lab [30].

The same problem can be established in rule 54. Of course, the construction of a glider gun or some other extension is sufficient to demonstrate unlimited growth in rule 54 (Figure 16(h)). Here we show the production of double glider guns.

https://doi.org/10.25088/ComplexSystems.23.3.259

1. Double glider gun = $n\,e_1$-2 $\overrightarrow{w}$-8e_1-2g_e(A,f$_1$)-2e_1-2g_e(A,f$_1$)-8e_1-2 $\overleftarrow{w}$-$n\,e_1$ (Figure 17(a)).
2. Double glider gun = $n\,e_1$-3 $\overrightarrow{w}$-5 e_2-g_e(B,f$_2$)-g_e(A,f$_2$)-g_e(B,f$_2$)-g_e(B,f$_2$)-g_e(A,f$_2$)-g_e(B,f$_2$)-5 e_2-3 $\overleftarrow{w}$-$n\,e_1$ (Figure 17(b)).



(a) (b)

Figure 17. Double glider guns in rule 54 produced from multiple collisions of (a) eight gliders, and (b) 12 gliders.

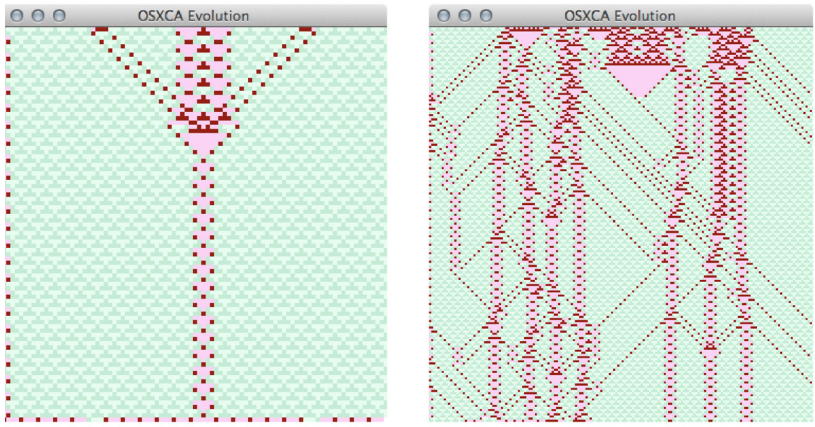
5.3 Holes and Big Tiles

In [21] McIntosh determined that ECA rule 110 can be studied as a tile problem. What is a largest tile produced via collision between gliders in rule 54? Some answers are given in [8] via studying reactions between gliders.

Figure 18(a) shows the construction of a T_{16} tile by synchronizing multiple collisions between $\overrightarrow{w}$, $\overleftarrow{w}$, and g_e gliders. Figure 18(b) shows a T_{33} tile produced by a chaotic decomposition. Codes to reproduce these reactions are as follows:

1. $T_{16} = n\,e_1$-2 $\overrightarrow{w}$-4e_1-2g_e(A,f$_1$)-4e_1-2 $\overleftarrow{w}$(A,f$_1$)-$n\,e_1$ (Figure 18(a)).
2. T_{33} = ne_1-1100101010010101001110000010011101100000101001010011110110011101110011000000101011010001110101010000001010011000101101000-ne_1 (Figure 18(b)).

https://doi.org/10.25088/ComplexSystems.23.3.259



(a) (b)

Figure 18. Big tiles emerging in rule 54. (a) T_{16} tile from six colliding gliders, (b) T_{33} tile as a decomposition from a specific string.

5.4 Memory Functions

Rule 54 has been proved to be a "universal dynamics rule" in the ECA memory (ECAM) classification of [12]. This means that rule 54 operated with some memory functions is able to reach any Wolfram class, including class IV, to which the memoryless rule 54 belongs [2].

Figure 19 presents evolutions of rule 54 with memory. Each snapshot illustrates four different behaviors. Figure 19(a) shows a uniform evolution with rule $\phi_{R54\,\mathrm{maj:6}}$, Figure 19(b) a periodic behavior with rule $\phi_{R54\,\mathrm{maj:10}}$, Figure 19(c) a chaotic evolution with rule $\phi_{R54\,\mathrm{maj:3}}$, and Figure 19(d) a complex behavior with rule $\phi_{R54\,\mathrm{maj:8}}$. Of course, every memory function represents a different evolution rule but with elements of the original rule.

5.5 Computing Potential

In [11] we show how a number of solitonic collisions can be simulated in rule 54. These solitons can be manipulated to develop some basic computable systems, such as simple substitution systems. In [8] basic logic functions were simulated from basic collisions in rule 54. So far no one has ever implemented an equivalent Turing machine in rule 54. However, taking advantage of codification of gliders in rule 54, we have explored some basic computable functions that could help us to emulate the Turing machine with rule 54 in the future.

https://doi.org/10.25088/ComplexSystems.23.3.259

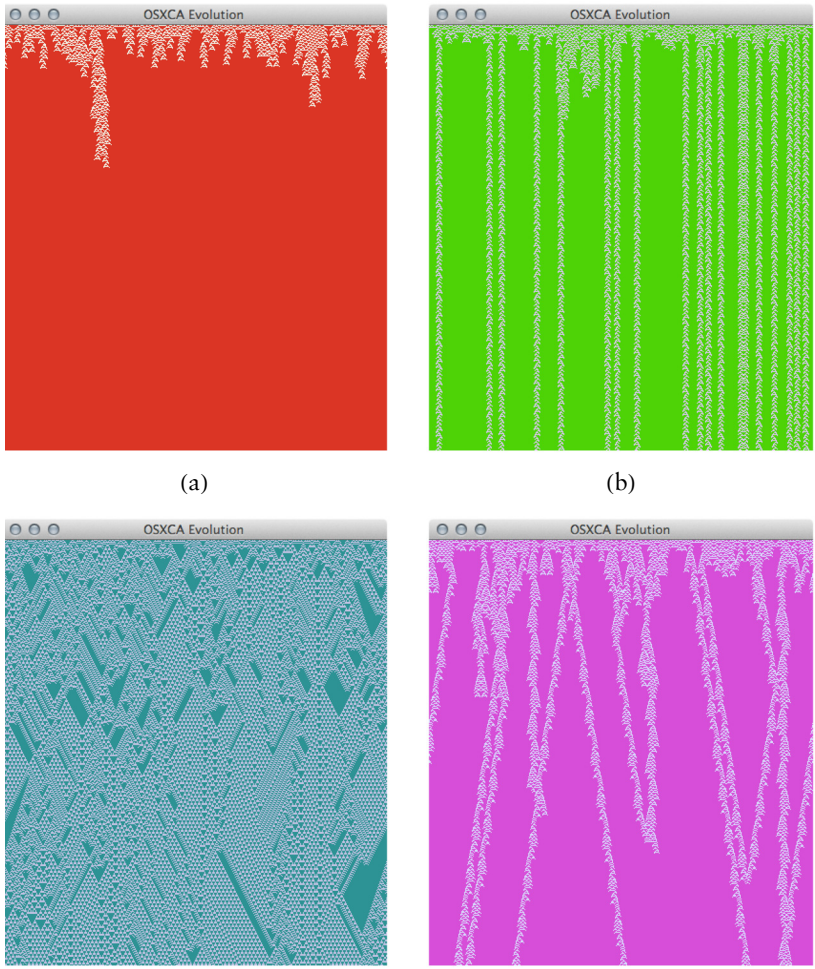


Figure 19. Rule 54 affected with memory functions. (a) ECAM $\phi_{R54\,\mathrm{maj:6}}$, (b) ECAM $\phi_{R54\,\mathrm{maj:10}}$, (c) ECAM $\phi_{R54\,\mathrm{maj:3}}$, (d) ECAM $\phi_{R54\,\mathrm{maj:8}}$.

Some series by reacting gliders are presented in [6]. Here we have three cases.

1. $\mathbb{Z}^n \forall\ n > 3 = n\, e_1\text{-}2\, g_e(\mathrm{A},\mathrm{f}_1)\text{-}6e_1\text{-}\overleftarrow{w}\text{-}n\, e_2$ (Figure 20).
2. Parity $= n\,[g_o(\mathrm{A},\mathrm{f}_1)\text{-}g_o(\mathrm{B},\mathrm{f}_1)]\text{-}2e_1\text{-}\vec{w}\text{-}2\, e_2\text{-}n[g_o(\mathrm{A},\mathrm{f}_2)\text{-}g_o(\mathrm{B},\mathrm{f}_2)]$ (Figure 21).
3. Flip-flop $= n\, e_1\text{-}g_e(\mathrm{A},\mathrm{f}_1)\text{-}2e_1\text{-}4\,\overleftarrow{w}\text{-}n\, e_1$ (Figure 22).

https://doi.org/10.25088/ComplexSystems.23.3.259

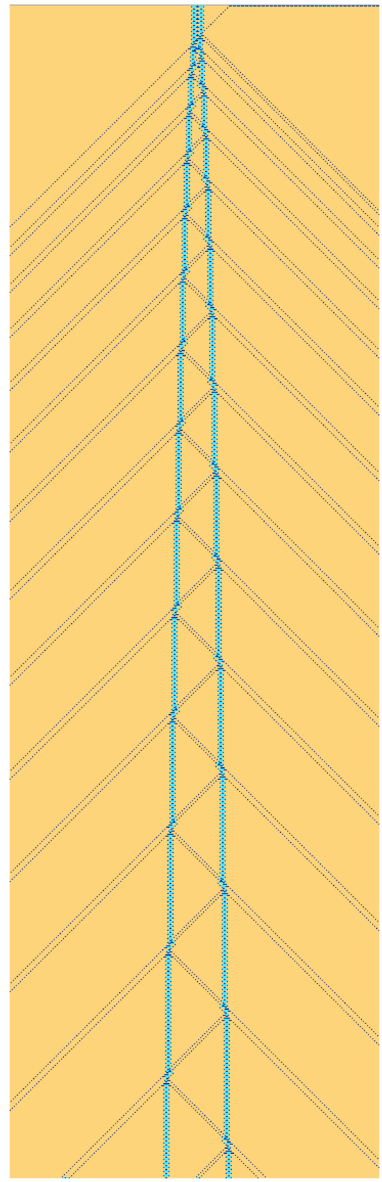


Figure 20. ECA rule 54 evolution deriving a series that yields $\mathbb{Z}^n$ for $n > 2$.

In Figure 20, starting from a collision among three gliders yields an infinite series $\mathbb{Z}^n$ for $n > 2$ (without limit boundaries). This sequence is defined by a vertical number of T_6 tiles without some perturbation that evolves on each collision. Figure 21 displays an evolution that

https://doi.org/10.25088/ComplexSystems.23.3.259

simulates a parity function $2k \forall k \in \mathbb{Z}$. This parity is preserved by the number of generations or by the number of T_5 tiles (g_o gliders) (without limit boundaries). So, Figure 22 shows a very simple flip-flop configuration that is restricted to limit boundaries. All previous simulations need more than 1000 generations.

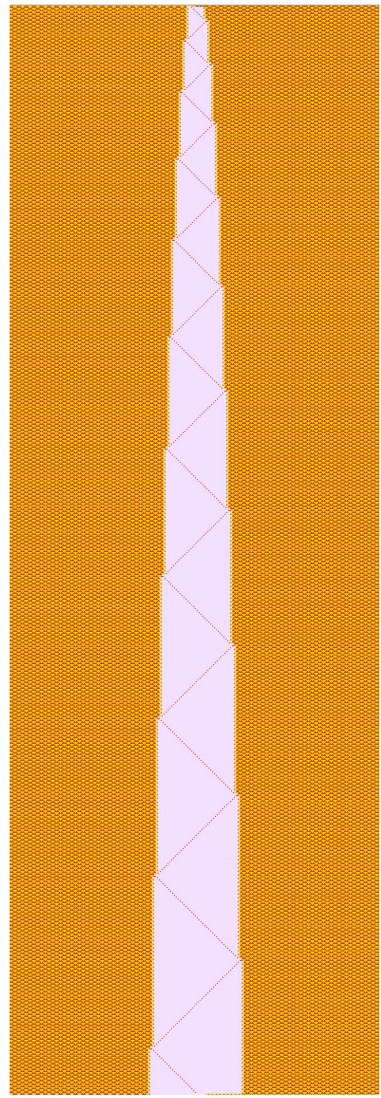


Figure 21. ECA rule 54 evolution deriving a parity function.

https://doi.org/10.25088/ComplexSystems.23.3.259

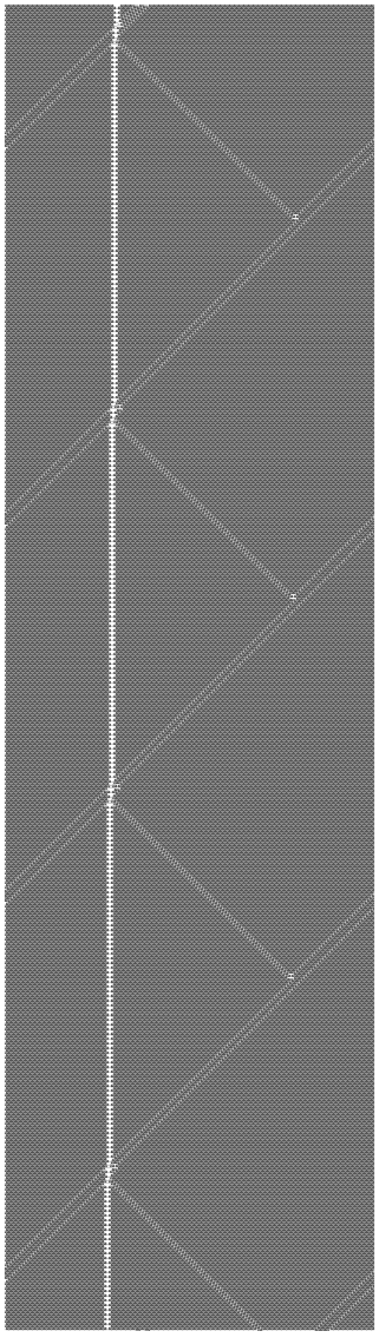


Figure 22. ECA rule 54 evolution implements a simple flip-flop.

https://doi.org/10.25088/ComplexSystems.23.3.259

6. Final Remarks

Cellular automaton gliders are analogs of optical solitons, kinks in polymer chains, excitation in molecular arrays (reaction diffusion computers [32], wave packets using slime mold to communicate information to distant parts of the body [33]), and defects in micro-tubules [34]. Also, rule 54 per se is a discrete analog and active nonlinear medium with lateral inhibition between micro-volumes. The lateral inhibition in the nervous system sharpens and strengthens sensor perception and is widely employed in vision and olfactory systems. Thus we can speculate that rule 54 is a simplest abstract model of the affective nervous system. The gliders then play the role of propagating action potential wave packets, and glider guns symbolize activity in the sources of sensorial stimulation. As we can see, there are many analogies of rule 54 behavior in physical and biological systems. And therefore, the behavior of these systems can be described by unique subsets of regular expressions, where phase, distance, momentum, position, period, and speed are taken into consideration.

References

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

[2] G. J. Martínez, "A Note on Elementary Cellular Automata Classification," *Journal of Cellular Automata*, **8**(3–4), 2013 pp. 233–259.

[3] N. Boccara, J. Nasser, and M. Roger, "Particlelike Structures and Their Interactions in Spatiotemporal Patterns Generated by One-Dimensional Deterministic Cellular-Automaton Rules," *Physical Review A*, **44**(2), 1991 pp. 866–875. doi:10.1103/PhysRevA.44.866.

[4] J. E. Hanson and J. P. Crutchfield, "Computational Mechanics of Cellular Automata: An Example," *Physics D: Nonlinear Phenomena*, **103**(1–4), 1997 pp. 169–189. doi:10.1016/S0167-2789(96)00259-X.

[5] A. Wuensche, *Exploring Discrete Dynamics*, Frome, England: Luniver Press, 2011.

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

[7] B. Martin, "A Group Interpretation of Particles Generated by One-Dimensional Cellular Automaton, Wolfram's Rule 54," *International Journal of Modern Physics C*, **11**(1), 2000 pp. 101–123.

[8] G. J. Martínez, A. Adamatzky, and H. V. McIntosh, "Phenomenology of Glider Collisions in Cellular Automaton Rule 54 and Associated Logical Gates," *Chaos, Solitons & Fractals*, **28**(1), 2006 pp. 100–111. doi:10.1016/j.chaos.2005.05.013.

https://doi.org/10.25088/ComplexSystems.23.3.259

[9] J. Guan, "Complex Dynamics of the Elementary Cellular Automaton Rule 54," *International Journal of Modern Physics C*, **23**(7), 2012 p. 1250052. doi:10.1142/S0129183112500520.

[10] M. Redeker, "Gliders and Ether in Rule 54." arxiv.org/abs/1007.2920v1.

[11] G. J. Martínez, A. Adamatzky, F. Chen, and L. Chua, "On Soliton Collisions between Localizations in Complex Elementary Cellular Automata: Rules 54 and 110 and Beyond," *Complex Systems* **21**(2), 2012 pp. 117–142. http://www.complex-systems.com/pdf/21-2-2.pdf.

[12] G. J. Martínez, A. Adamatzky, and R. Alonso-Sanz, "Designing Complex Dynamics in Cellular Automata with Memory," *International Journal of Bifurcation and Chaos*, **23**(10), 2013 p. 1330035. doi:10.1142/S0218127413300358.

[13] G. J. Martínez, A. Adamatzky, and H. V. McIntosh, "On the Representation of Gliders in Rule 54 by de Bruijn and Cycle Diagrams," *Lecture Notes in Computer Science*, **5191**, 2008 pp. 83–91. doi:10.1007/978-3-540-79992-4_11.

[14] G. J. Martinez. "Elementary Cellular Automaton Rule 54." (Jul 11, 2014) http://uncomp.uwe.ac.uk/genaro/Rule54.html.

[15] B. Grünbaum and G. C. Shephard, *Tilings and Patterns*, New York: W. H. Freeman and Company, 1987.

[16] H. V. McIntosh. "Linear Cellular Automata via de Bruijn Diagrams." (Jul 11, 2014) http://delta.cs.cinvestav.mx/~mcintosh/newweb/marcodebruijn.html.

[17] H. V. McIntosh, *One Dimensional Cellular Automata*, Beckington, UK: Luniver Press, 2009.

[18] A. Wuensche and M. Lesser, *The Global Dynamics of Cellular Automata*, Reading: Addison-Wesley Publishing Company, 1992.

[19] H. V. McIntosh. "Rule 110 as It Relates to the Presence of Gliders." (Jul 23, 2014) http://delta.cs.cinvestav.mx/~mcintosh/comun/RULE110W/RULE110.html.

[20] G. J. Martínez, H. V. McIntosh, and J. C. Seck-Tuoh-Mora, "Gliders in Rule 110," *International Journal of Unconventional Computing*, **2**(1), 2006 pp. 1–49. http://uncomp.uwe.ac.uk/genaro/Papers/Papers_on_CA _files/MARTINEZ.pdf.

[21] H. V. McIntosh. "A Concordance for Rule 110." (Jul 11, 2014) http://delta.cs.cinvestav.mx/~mcintosh/comun/ccord/ccord.html.

[22] G. J. Martínez, H. V. McIntosh, J. C. Seck-Tuoh-Mora, and S. V. Chapa-Vergara, "Determining a Regular Language by Glider-Based Structures Called Phases f_i_1 in Rule 110," *Journal of Cellular Automata*, **3**(3), 2008 pp. 231–270.

[23] B. H. Voorhees, *Computational Analysis of One-Dimensional Cellular Automata*, River Edge, NJ: World Scientific, 1996.

https://doi.org/10.25088/ComplexSystems.23.3.259

[24] H. V. McIntosh. "Cellular Automata Packages." (Jul 11, 2014) http://delta.cs.cinvestav.mx/~mcintosh/oldweb/software.html.

[25] G. J. Martinez. "OSXCA Systems." (Jul 11, 2014) http://uncomp.uwe.ac.uk/genaro/OSXCASystems.html.

[26] B. H. Voorhees, "Remarks on Applications of de Bruijn Diagrams and Their Fragments," *Journal of Cellular Automata*, **3**(3), 2008 pp. 187–204.

[27] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory Languages, and Computation*, Reading: Addison-Wesley Publishing Company, 1979.

[28] M. Minsky, *Computation: Finite and Infinite Machines*, Englewood Cliffs, NJ: Prentice Hall, 1967.

[29] Wikipedia. "Agar."(Jul 11, 2014) http://conwaylife.com/wiki/Agar.

[30] Wikipedia. "Chess Programming: Bill Gosper." (Jul 30, 2014) http://chessprogramming.wikispaces.com/Bill+Gosper.

[31] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, New York: Oxford University Press, 1993.

[32] A. Adamatzky, B. L. Costello, and T. Asai, *Reaction-Diffusion Computers*, Boston: Elsevier, 2005.

[33] A. Adamatzky, *Physarum Machines: Computers from Slime Mould*, Hackensack, NJ: World Scientific Publishing Co. Pte. Ltd. 2010.

[34] A. Adamatzky, ed., *Collision-Based Computing*, London: Springer, 2002.

https://doi.org/10.25088/ComplexSystems.23.3.259