

On the representation of gliders in Rule 54 by de Bruijn and cycle diagrams

Genaro J. Martínez¹, Andrew Adamatzky¹, and Harold V. McIntosh²

¹ Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England, Bristol, United Kingdom

`{genaro.martinez, andrew.adamatzky}@uwe.ac.uk`

<http://uncomp.uwe.ac.uk/genaro/>

<http://uncomp.uwe.ac.uk/adamatzky/>

² Departamento de Aplicación de Microcomputadoras, Instituto de Ciencias, Universidad Autónoma de Puebla, Puebla, México.

`mcintosh@servidor.unam.mx`

<http://delta.cs.cinvestav.mx/~mcintosh/>

Abstract. Rule 54, in Wolfram’s notation, is one of elementary yet complexly behaving one-dimensional cellular automata. The automaton supports gliders, glider guns and other non-trivial long transients. We show how to characterize gliders in Rule 54 by diagram representations as de Bruijn and cycle diagrams; offering a way to present each glider in Rule 54 with particular characteristics. This allows a compact encoding of initial conditions which can be used in implementing non-trivial collision-based computing in one-dimensional cellular automata.

1 Preliminaries

Amongst one-dimensional cellular automata (CA) studied by Wolfram in [15, 16], one can find a few evolution rules supporting gliders (particles or mobile self-localizations); rules 110 and 54 exhibit particularly rich and somewhat complex behaviour.³ Interaction between gliders can be employed to execute logical operations, and thus ultimately to perform universal computation [1]. Collision-based computing schemes are very sensitive to initial configurations of gliders, even a shift in a glider phase can completely destroy a cascade of logical gates. There is a need for a compact and uniform description of glider types and glider interactions. In the present paper, taking Rule 54 as example, we discuss how such a description can be reached by de Bruijn and cycle diagrams.

Following, Wolfram’s notation — a one-dimensional elementary CA has two parameters (k, r) , number of states k and cell neighborhood radius r — Rule 54 is a CA with parameters $(2, 1)$, i.e. two cell-states and three cell neighborhood (a central cell, its left and right neighbors). The local transition function f is determined as follows: $111 \rightarrow 0$, $110 \rightarrow 0$, $101 \rightarrow 1$, $100 \rightarrow 1$, $011 \rightarrow 0$, $010 \rightarrow 1$, $001 \rightarrow 1$ and $000 \rightarrow 0$. The binary sequence 00110110 in decimal notation represents the evolution rule 54.

³ <http://uncomp.uwe.ac.uk/genaro/Rule54.html>

A detailed study of Rule 54 was initiated in [2], and some basic characteristics of gliders, or particles, were initially derived by Hanson and Crutchfield in [4]. They constructed a subset diagram capable of identifying sequences of gliders in Rule 54. No complete characterization of gliders was provided however. This is a reason why we became interested in representing each glider in Rule 54 by a well-defined way as de Bruijn and cycle diagrams [8, 12, 14, 17].

In a previous paper [5] we demonstrated self-organization in Rule 54 developing a ‘genealogical tree’ of gliders, where every glider can be derived from collisions between other ones, and also classifying all possible scenarios of binary and ternary collisions between gliders. In the present paper we advance our understanding of glider dynamics by de Bruijn and cycle diagram representations.

Results of the present paper are based on de Bruijn diagrams [8, 11–13], cycle diagrams [14, 9] and computing based on interaction between gliders [1].

2 Gliders in Rule 54

A glider is a compact group of non-quiescent states traveling along cellular automata lattice. Rule 54 automaton exhibits relatively a small number of glider types, which makes it particularly attractive for discretization and formal representation. Thus we look tools to characterize these gliders and control collisions in Rule 54 from its initial condition.⁴

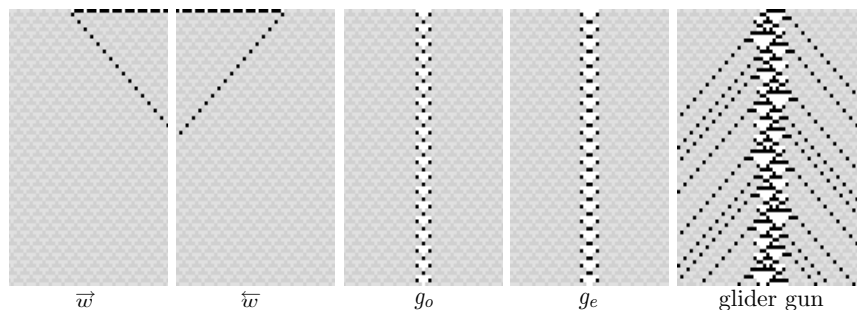


Fig. 1. Gliders in Rule 54 (time goes down). Cell-state 1 is shown by black pixels, also a filter is represented for clarity.

To represent gliders in Rule 54 we follow Boccaro’s *et al.* notation [2]. Thus we can display every glider together with a list of its properties (dynamic, names, speed and periodic width), see examples in Fig. 1 and Tab. 1.

We study automata with periodic boundary conditions, let e_1 and e_2 represent glider phases in the periodic background. Thus we have four primitive

⁴ Single gliders in Rule 54 or packages and extensions of them also can be consulted in <http://uncomp.uwe.ac.uk/genaro/rule54/glidersRule54.html>

Table 1. Properties of gliders in Rule 54.

structure (glider)	v_g	periodic width
e_1	$2/2 = 1$	4
e_2	$2/2 = 1$	4
w^{\rightarrow}	$2/2 = 1$	2
w^{\leftarrow}	$-2/2 = -1$	0-4
g_o	$0/4 = 0$	6-2
g_e	$0/4 = 0$	7-3
glider gun	$0/32 = 0$	14-4

gliders — $w^{\rightarrow}, w^{\leftarrow}, g_o, g_e$ — and a compound glider — the glider gun. Speed v_g of a glider is evaluated using the period between displacements. Therefore w^{\rightarrow} glider moves with positive slope, w^{\leftarrow} glider moves with negative slope and g_o, g_e and the glider gun move with velocity zero (as still live configurations in one dimension).

2.1 De Bruijn diagrams calculating gliders in Rule 54

For a one-dimensional cellular automaton of order (k, r) and a given finite alphabet K , its de Bruijn diagram is defined as a directed graph with k^{2r} vertexes and k^{2r+1} edges. The vertexes are labeled with the elements of the alphabet of length $2r$. 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 $f(i \diamond j)$ (the value of the neighborhood defined by the local function f) [12, 13].

Thus de Bruijn diagram of any one-dimensional CA can be constructed as follow:

$$M_{i,j} = \begin{cases} 1 & \text{if } j = ki, ki + 1, \dots, ki + k - 1 \pmod{k^{2r}} \\ 0 & \text{in other case} \end{cases} \quad (1)$$

Modulo $k^{2r} = 2^2 = 4$ represents the number of vertexes in the de Bruijn diagram and j takes values from $k * i = 2i$ to $(k * i) + k - 1 = (2 * i) + 2 - 1 = 2i + 1$. The vertexes (indexes of M) are labeled by fractions of neighborhoods originated by 00, 01, 10 and 11, the overlap determines each connection. This way Fig. 2 displays Rule 54's matrix evolution and de Bruijn diagram of Rule 54 respectively.

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 [11, 13].

After the de Bruijn diagram is completed, we can calculate an extended de Bruijn diagram [11, 7]. An extended de Bruijn diagram takes into account more significant overlapping of neighborhoods. Thus, we represent $M_{R54}^{(2)}$ by indexes

$i = j = 2r * n$, where $n \in \mathbb{Z}^+$. Moreover the de Bruijn diagram grows exponentially, order k^{2r^n} , for each $M_{R54}^{(n)}$; consequently basic de Bruijn diagram is obtained for $n = 1$.

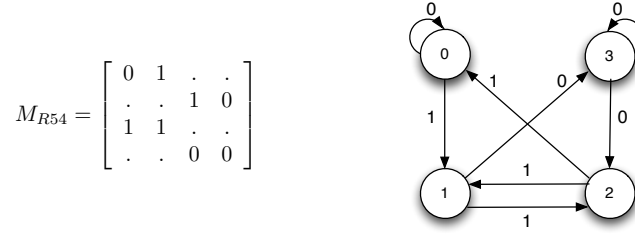


Fig. 2. de Bruijn diagram for Rule 54.

Let us calculate de Bruijn diagrams for gliders w^{\rightarrow} and w^{\leftarrow} with periodic background. Tab. 1 shows that the gliders translate two cells in a time. Hence the extended de Bruijn diagram of order $M_{R54}^{(2)}$ can be used to extract a cyclic structure of gliders.⁵ The constructed diagrams show all possible relations but, we will focus on cycles, or periodic strings generated by local-transition functions.

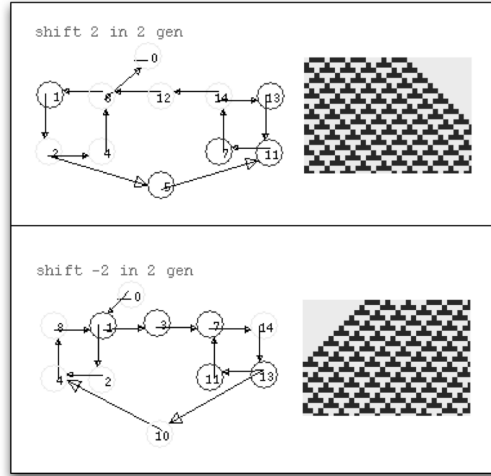


Fig. 3. de Bruijn diagrams corresponding to gliders w^{\rightarrow} (up) and w^{\leftarrow} (down).

⁵ Extended de Bruijn diagrams are calculated with NXLCAU21 software developed by Harold McIntosh. Application and source code are available at <http://delta.cs.cinvestav.mx/~mcintosh/oldweb/software.html>

Fig. 3 demonstrates the de Bruijn diagram when strings are translated two sites to the right (+) or to the left (-). In this case, Rule 54 offers an easy case to extract gliders from their cyclic representation in de Bruijn diagrams. A glider can be identified as a cycle and the glider interactions with regard to their phase changes are reflected in bigger cycles containing two or more cycles. The first diagram displays periodic strings moving two cells to the right in two times, i.e., period and displacement in the periodic background defining the glider w^{\rightarrow} . This diagram has a positive orientation of cycles and also shows that relations of vertexes (1, 2, 4, 6) and (13, 11, 7, 14) representing all possible phases where glider w^{\rightarrow} can be placed. However, the existence of this glider is related to both cycles of the diagram. Thus the periodic background in phase one represents the string 1000 and in phase two the string 1110. So, to represent glider w^{\rightarrow} in its two different phases we must make a transition from one cycle to another, or to concatenate the strings (1000)*-00-(1110)*.

Rule 54 has a particular characteristic where gliders can travel between two or three different backgrounds. Fig. 3 show four cycles, three of them self-contained and one cycle starts with the nil state. Fragments of evolutions in the same picture show what types of gliders are more likely to be defined by these cycles. We can see a large cycle represented by the vertexes (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 w^{\rightarrow} gliders. Finally, a fourth cycle, represented by the cycle 0, determines a transition between two different patterns, “fuse configurations.” The periodic background is formed by a cycle of length four and the existence of gliders is determined by other cycles. Therefore, we see that the problem of representing gliders by de Bruijn diagrams is reduced to the classification of cycles in the diagrams.

The advantage of using a de Bruijn diagram is that many problems concerning automata are thereby transformed into known problems regarding of the tracing of paths through a graph. For instance, no loop can be longer than the total number of nodes in the graph without repeating some segment; but then there must exist still other loops in which the repeated segment is traversed an arbitrary number of times. For example, a binary automaton depending upon nearest neighbors has eight distinct neighborhoods, representable as eight links connecting four nodes, it follows that no static configuration can be more than four cells long without repeating some two-cell partial neighborhood. Thus the static configurations are rather severely constrained.

Sometimes the de Bruijn diagram reveals information about localized aspects of a configuration. For example if an acceptable path terminates at a node in which all the outgoing links are acceptable, it does not need to continue. Likewise, if all the incoming links are acceptable, the path may begin just as it has been part of a loop. Thus semi infinite structures may be located, or even finite ones if both ends have such universal terminations. This leads to the phenomenon of membranes and macrocells which Wolfram noticed during the course of his investigations. That is, an automaton may have patches which are isolated from one another by static regions, whose evolutions proceed quite independently [11].

2.2 Cycle diagrams calculating gliders in Rule 54

De Bruijn diagrams have demonstrated their power in representing periodic strings based in gliders in elemental CA [7]. Some other tools could also be used to derive these sequences, e.g. the cycle diagrams (Fig. 4). Of course, we should remember that de Bruijn diagrams grow exponentially, so our potential for computational classification of the diagrams is fairly limited. Thus cycle diagrams can help at least calculating attractors as was extensively studied by Wuensche in [14, 9] and thus precisely represent some selected periodic patterns.

There are two ways to obtain the cycles for a given automaton. The first is to enumerate all the rings of the desired length, and follow up the evolution of each. In doing so task, various shortcuts can be taken, such as generating the configurations in Gray code order so that only a single cell changes state from one to the next. Still lifes can be detected very quickly in this way. Numerical comparison of successive generations means that whenever the new generation is smaller, it has been already examined and there is no need for further exploration.

The second way is more systematic and is worth the bookkeeping effort involved. A graph whose links are determined by evolution is prepared, following which a path enumerating procedure is followed to locate all the loops, whose lengths will give the periods of all the cycles of that length. Cycles of length up to ten can be obtained easily, twenty with effort, but passing thirty requires dedication; for binary automata it is slightly easier, increasingly more difficult for others [11].

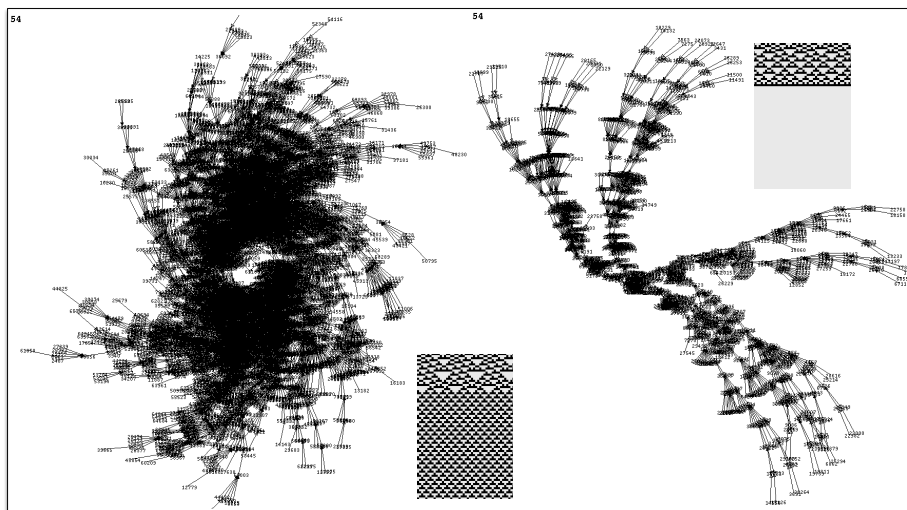


Fig. 4. Cycle diagrams calculating their two possible periodic and stable background in Rule 54 from their attractors.

Also cycle diagrams represent very well and easily the concept of global configurations and Garden of Eden configurations. The first involucre a global state of a CA in each vertex therefore each global state represents an instantaneous description of the evolution space and also is guaranteed that each global state has an image. The second concept is represented with configurations without ancestors, i.e., leaves as vertexes that have not images from another global configurations, these configurations are called Garden of Eden configurations.

Finally we must mention that the evolution space in Rule 54 can be characterized as well by means of tiles as it was studied in Rule 110 [10, 7]. Although in this paper it is not extended the analysis, several of them are classified from their de Bruijn and cycle diagrams.

Thus we can enumerate some important characteristics to determine gliders from their cycle diagrams as follow:

Table 2. Cycle diagrams calculating gliders in Rule 54.

cycle length	cycle	total vertexes	structures
4	4	4	T_3 and T_2 tiles
6	4	5	g_e glider
8	4	14	g_e gliders joined
	6	28	g_e glider with a T_2
9	4	44	g_e-g_o gliders joined
	27	45	T_4 transporting a w^{\leftarrow} (extensible as a T_5 in R110)
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 with $w^{\rightarrow}-T_5-T_6-T_4-T_2$ tiles
12	10	124	periodic background composed of $2T_6-2T_3-T_2$ tiles
	12	102	$2w^{\rightarrow}$ gliders
13	4	406	$(g_e-g_o)^*$ gliders concatenated
	169	1274	meta-glider with $T_8-2T_4-T_2$ and w^{\rightarrow} gliders
14	112	805	meta-glider with $T_8-3T_4-T_2$ tiles
15	330	7680	meta-glider with $T_5-2T_6-T_4-T_2$ tiles
16	6	116	periodic background composed of T_6-T_2 tiles
	8	8	w^{\rightarrow} gliders
	14	944	meta-glider with $w^{\rightarrow}-g_o-w^{\leftarrow}$ gliders
	16	2896	$2w^{\rightarrow}$ gliders
	40	1246	meta-glider with $T_8-5T_6-2T_2-3T_4-T_5$ tiles

In this way we can list a number of periodic strings derived from the de Bruijn and cycle diagrams. In this case, every primitive glider (see Fig. 1) can be reproduced from different cycles as shows Tab. 2. The concept of meta-glider refers to a synchronization of several gliders preserving their form, reactions and structure periodically as was also reported in Rule 110 in [6].

3 Coding Rule 54

Quickly we must show how to code Rule 54 to solve some problems reported in [5]. We can select a production by gliders or sequences and code its respective initial condition.⁶ If we select a production by gliders, a number of them are important, e.g. if we want to produce a w^\rightarrow glider, we need to collide a g_o with a w^\leftarrow glider. The encoding can be described as follows:

- (a) $w^\rightarrow = e_1^+ - g_o(A, f_1) - e_1^+ - w^\leftarrow(f_2) - e_2^+$
- (b) $w^\rightarrow = e_1^+ - g_o(B, f_1) - e_1^+ - w^\leftarrow(f_2) - e_2^+$
- (c) glider gun = $e_1^+ - g_e(A, f_1) - g_e(B, f_1) - 4e_1 - w^\leftarrow(f_1) - e_2^+$
- (d) double glider gun = $e_1^+ - 2w^\rightarrow(f_1) - 8e_1 - 2g_e(A, f_1) - 2e_1 - 2g_e(A, f_1) - 8e_1 - 2w^\leftarrow(f_1) - e_1^+$
(see fig. 5).

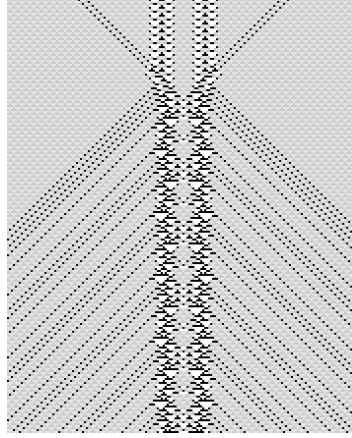


Fig. 5. Double glider gun in Rule 54 produced from multiple collisions between gliders.

4 Conclusions

The rows of the period-cycle table can be found from de Bruijn diagrams in the same way that the cycles can be found from the evolution diagram; since $2r + 1$ cells are needed to deduce a generation of evolution, only about half as many periods as cycles can be worked out for a given amount of effort. This anomaly is really an artifact of the way r parameterizes the neighborhood, and would disappear if half-integral increments were taken for r .

⁶ See demos of encoding collision between single gliders and their trains in <http://uncomp.uwe.ac.uk/genaro/rule54/collisionsRule54.html>

Similar theoretical conclusions are possible, since the periods are taken from a subset of the de Bruijn diagram. A $2r$ -stage de Bruijn diagram for k symbols has k^{2r} nodes; k times as many links. Once this number of links has been used up in constructing a path through the diagram, one of them would have to be repeated. Thus there is also an exponential upper bound in the rows of the period-cycle table. For example, if an automaton has a cycle of period 2, it must already show up in some short ring; if it has not appeared in rings below a certain limit, it will never appear in longer rings [11].

Finally a way to code Rule 54 is needed to obtain a complete description of gliders. With this system we would be able to construct arbitrary initial conditions in Rule 54 to control simple or complicated reactions with dozen or hundred of gliders. Potentially this feature will provide a powerful tool to develop spatiotemporal solutions in Rule 54 such as: Rule 54 objects, solitons, eaters and so on. Thus a complete regular glider-based language in Rule 54 with de Bruijn diagrams will be presented in our next paper.

Acknowledgement

Our research was partly supported by Engineering and Physical Sciences Research Council (EPSRC), United Kingdom, grant EP/F054343.

References

1. Adamatzky, A. (Ed.) (2003) *Collision-Based Computing*, Springer.
2. Boccara, N., Nasser, J. and Roger, M. (1991) Particle like structures and their interactions in spatio-temporal patterns generated by one-dimensional deterministic cellular automaton rules, *Physical Review A* **44** (2), 866–875.
3. Golomb, S.W. (1967) *Shift Register Sequences*, Holden-Day, San Francisco.
4. Hanson, J.E. and Crutchfield, J.P. (1997) Computational Mechanics of Cellular Automata: An Example, *Physics D* **103** (1-4), 169–189.
5. Martínez, G.J., Adamatzky, A. and McIntosh, H.V. (2006) Phenomenology of glider collisions in cellular automaton Rule 54 and associated logical gates, *Chaos, Solitons and Fractals* **28**, 100–111.
6. Martínez, G.J., McIntosh, H.V., Seck Tuoh Mora, J.C. and Chapa Vergara, S.V. (2007) Rule 110 objects and other collision-based constructions, *J. Cellular Automata* **2** (3), 219–242.
7. Martínez, G.J., McIntosh, H.V., Seck Touh Mora, J.C. and Chapa Vergara, S.V. Determining a regular language by glider-based structures called phases $f_{i,1}$ in Rule 110, *J. Cellular Automata*, in press.
8. McIntosh, H.V. (1991) Linear cellular automata via de Bruijn diagrams, <http://delta.cs.cinvestav.mx/~mcintosh/oldweb/pautomata.html>
9. McIntosh, H.V. (1993) Commentaries on: The Global Dynamics of Cellular Automata, (by Wuensche, A. and Lesser, M.), <http://delta.cs.cinvestav.mx/~mcintosh/oldweb/pautomata.html>
10. McIntosh, H.V. (1999) Rule 110 as it relates to the presence of gliders, <http://delta.cs.cinvestav.mx/~mcintosh/oldweb/pautomata.html>
11. McIntosh, H.V. *One Dimensional Cellular Automata*, by publish.

12. Voorhees, B.H. (1996) *Computational analysis of one-dimensional cellular automata*, World Scientific Series on Nonlinear Science, Series A, Vol. 15.
13. Voorhees, B.H. Remarks on Applications of De Bruijn Diagrams and Their Fragments, *Journal of Cellular Automata*, in press.
14. Wuensche, A. and Lesser, M. (1992) *The Global Dynamics of Cellular Automata*, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley Publishing Company.
15. Wolfram, S. (1986) *Theory and Applications of Cellular Automata*, World Scientific Press, Singapore.
16. Wolfram, S. (2002) *A New Kind of Science*, Wolfram Media, Inc., Champaign, Illinois.
17. Wuensche, A. (1999) Classifying Cellular Automata Automatically, *Complexity* **4** (3), 47–66.