Collision-based Computing with Cellular Automata

Genaro Juárez Martínez
genaro.martinez@uwe.ac.uk, gjuarezm@ipn.mx
https://www.comunidad.escom.ipn.mx/genaro/

- Computer Science Laboratory (LCCOMP), IPN Mexico / UWE United Kingdom
- Unconventional Computing Laboratory (UCL), University of the West of England, United Kingdom
- Artificial Life Robotics Lab (ALIROB), National Polytechnic Institute, Mexico
- Centre for Chaos and Complex Networks (CCCN), City University of Hong Kong, Hong Kong
- Foundation of Computer Science Laboratory (FCSL), Hiroshima University, Japan
- Unconventional Algorithms and Computing (NAVY), Czech Republic
- Laboratoire de Recherche Scientifique (LABORES), Paris, France

AUTOMATA 2020
Stockholm, Sweden
August 10, 2020

It is a project worked and shared by several years

International Computer Science Laboratory, IPN, Mexico
https://www.comunidad.escom.ipn.mx/LCCOMP/

Unconventional Computing Laboratory, UWE, UK
https://uncomp.uwe.ac.uk/

Artificial Life Robotics Lab, IPN, Mexico
https://www.comunidad.escom.ipn.mx/ALIROB/

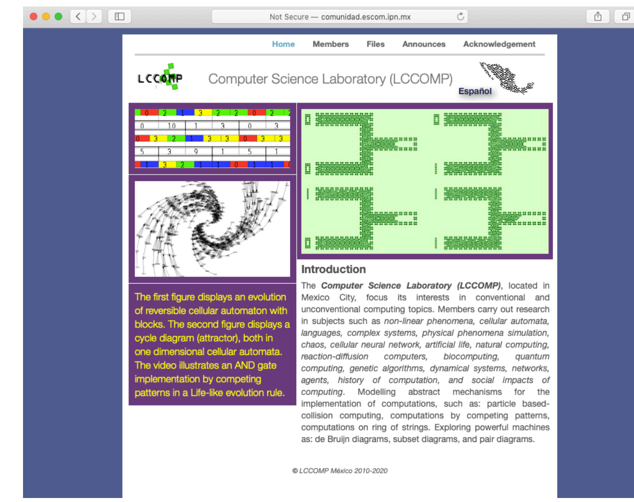
Foundation of Computer Science Laboratory, HU, Japan
http://www.iec.hiroshima-u.ac.jp/

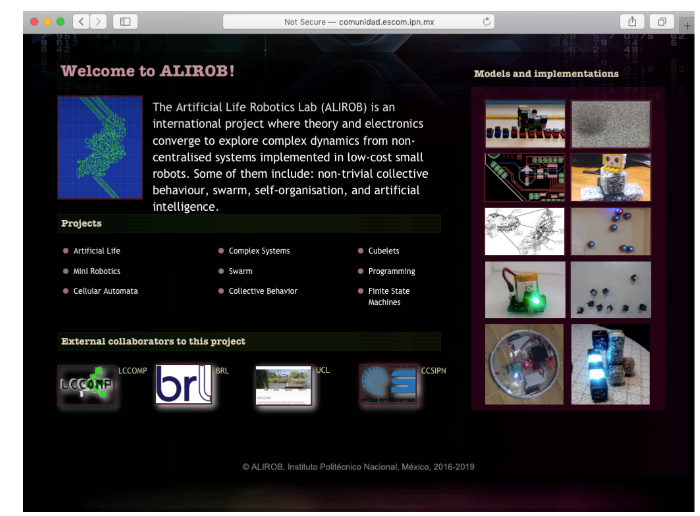
Centre for Chaos and Complex Networks, CUHK, China
http://www.ee.cityu.edu.hk/~cccs/

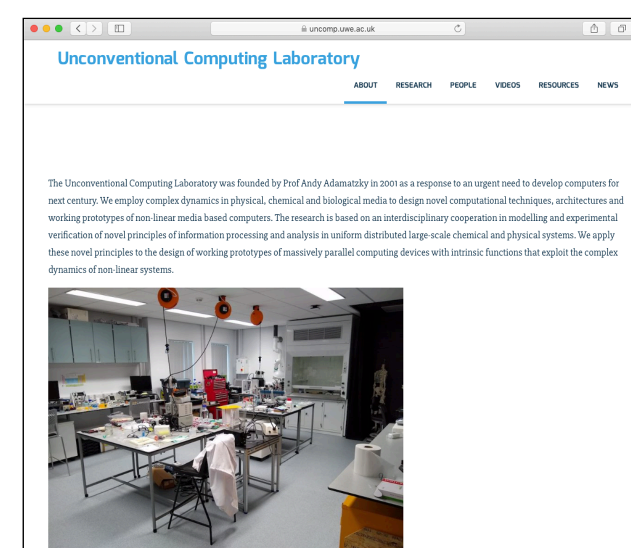
Unconventional Algorithms and Computing, Czech Republic
https://zelinkaivan65.wixsite.com/navy/

Laboratoire de Recherche Scientifique, France
http://labores.eu/

Institut des Systèmes Complexes en Normandie, France
http://iscn.univ-lehavre.fr/







Collaborations (researchers and students) and sponsors


Harold McIntosh


Andrew Adamatzky


Juan Seck


Shigeru Ninagawa

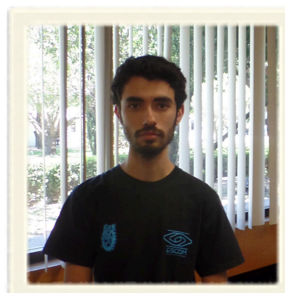

Matthew Cook


Andrew Wuensche

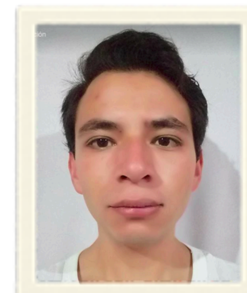

Stephen Wolfram

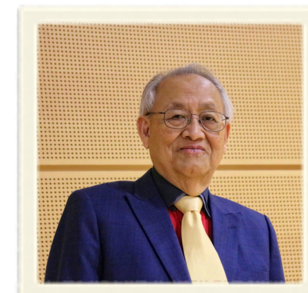

Kenichi Morita

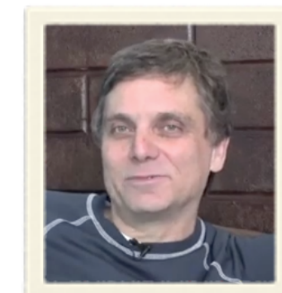

Kenneth Steiglitz

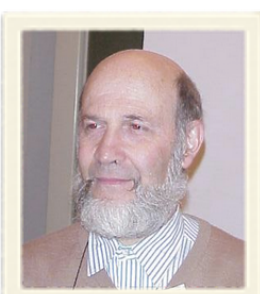

Ricardo Figueroa


Daniel Ayala


José Amador


Leon Chua


Todd Rowland


Maurice Margenstern


Rolf Hoffmann


Dominique Désérable


Ivan Zelinka

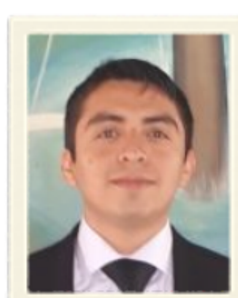

Dmitry Zaitsev

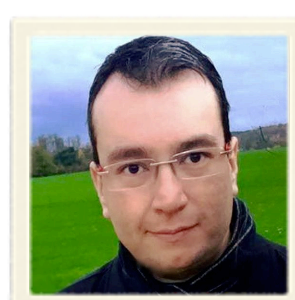

Guanrong Chen


Richard Gordon

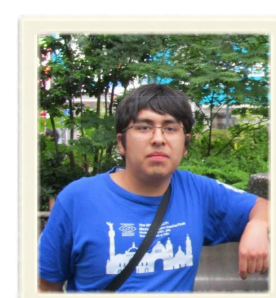

Fangyue Chen


Gabriela Moreno


Sergio Juárez


Héctor Zenil


Sergio Chapa


Iván Manzano


Christopher Stephens

Complex cellular automata a way to unconventional computing

Today, a "computer", without further qualifications, denotes a rather well-specified kind of object; we'll consider a computer "non-conventional" if its physical substrate or its organization significantly depart from this de facto norm. [Toffoli 1998]

Unconventional computers shall exploit molecular computing level, to increase the power of computation, velocity, and storage. Actually, the term about of *unconventional computing* or *natural computing* have a number of directions:

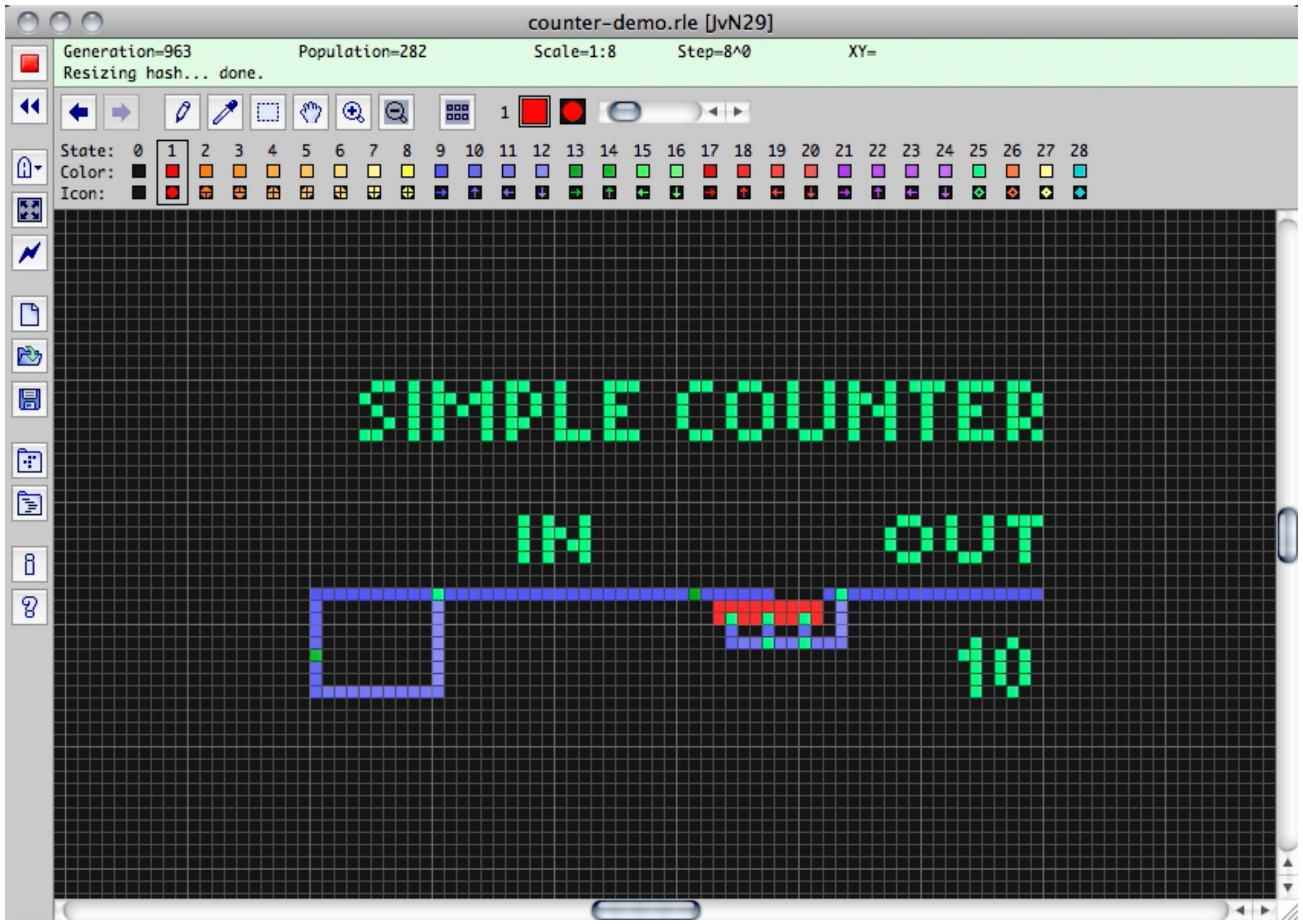
- quantum computing
- DNA computing
- reaction-diffusion computing
- reversible computing
- tiling (pattern) computing
- origami computing
- Pysarum computing
- swarm computing

- Toffoli, T. (1998) **Non-Conventional Computers**, *Encyclopedia of Electrical and Electronics Engineering* (John Webster Ed.), 14:455-471, Wiley & Sons.
- Mills, J.W. (2008) **The Nature of the Extended Analog Computer**, *Physica D*, 237(9):1235-1256.
- Fredkin, E. & Toffoli, T. (2002) **Design Principles for Achieving High-Performance Submicron Digital Technologies**, In: *Collision-Based Computing*, A. Adamatzky (Ed.), Springer, chapter 2, 27-46.

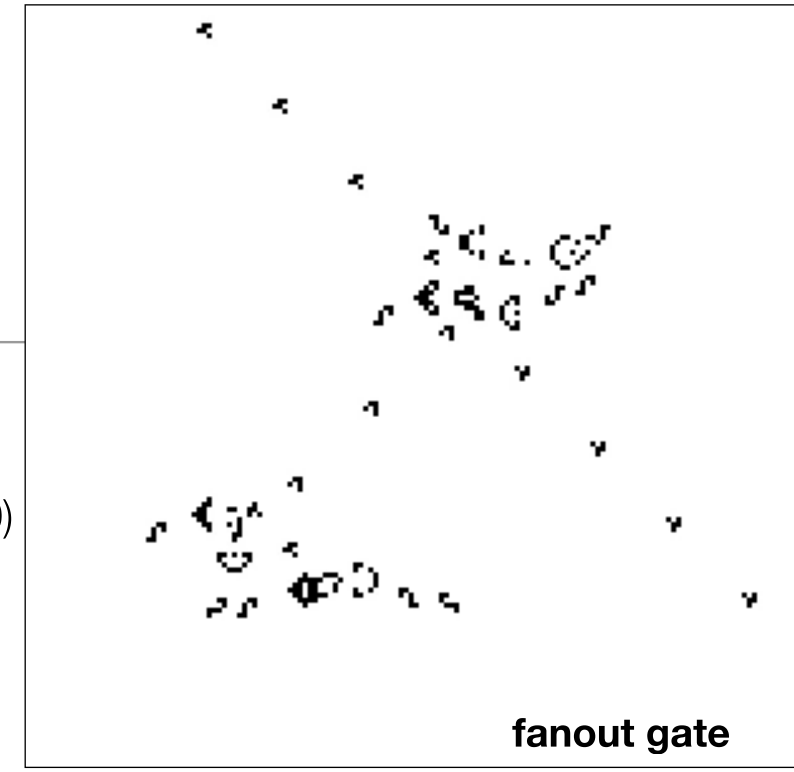
Antecedents and objectives

- Historically complex systems are related to computations.
- We explore a framework to design cellular automata computers on *colliders* and *cyclotrons, a circular way.*
- Basically, we work with complex elementary cellular automata.

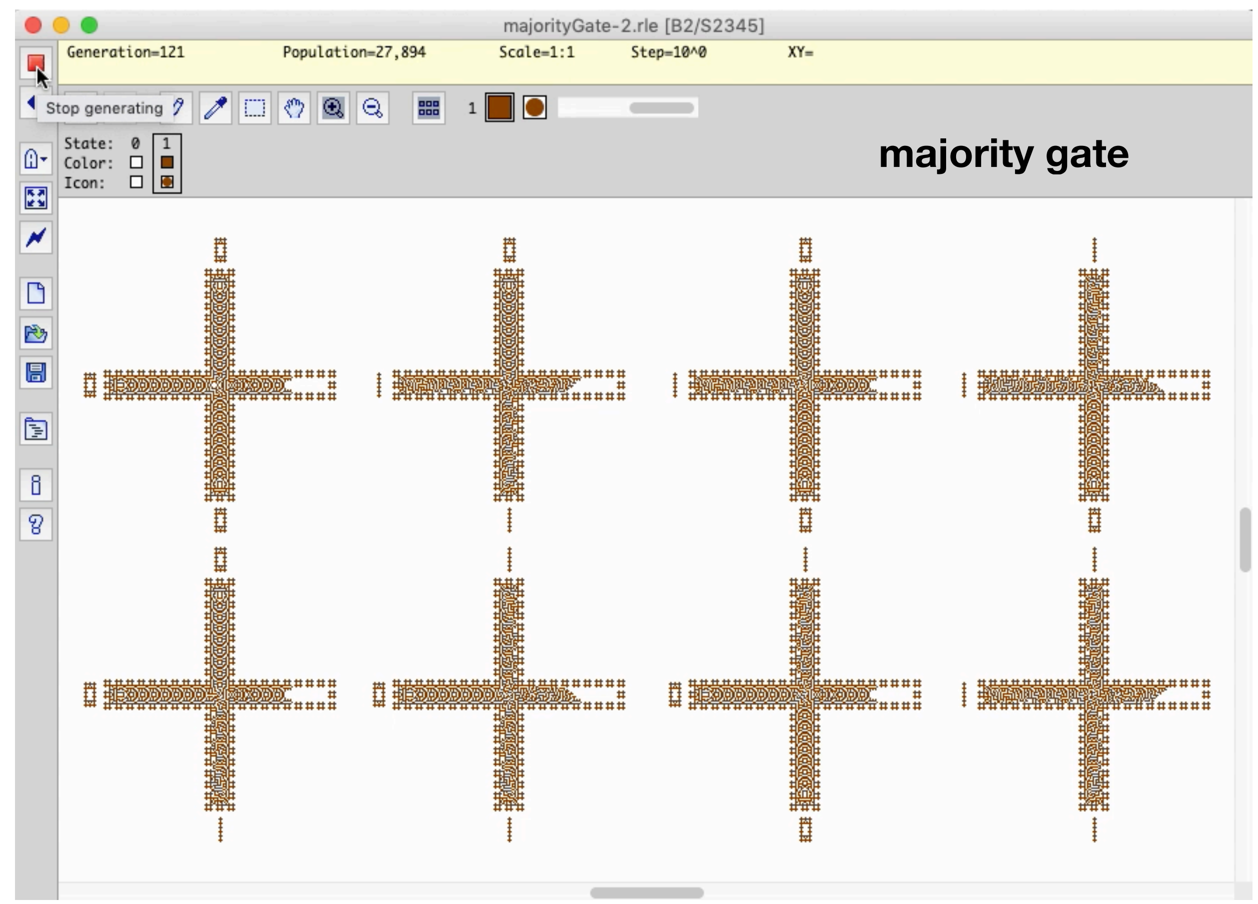
Some samples of computations in cellular automata



von Neumann 29-sates CA (1966)



Game of Life
2-states CA (1970)

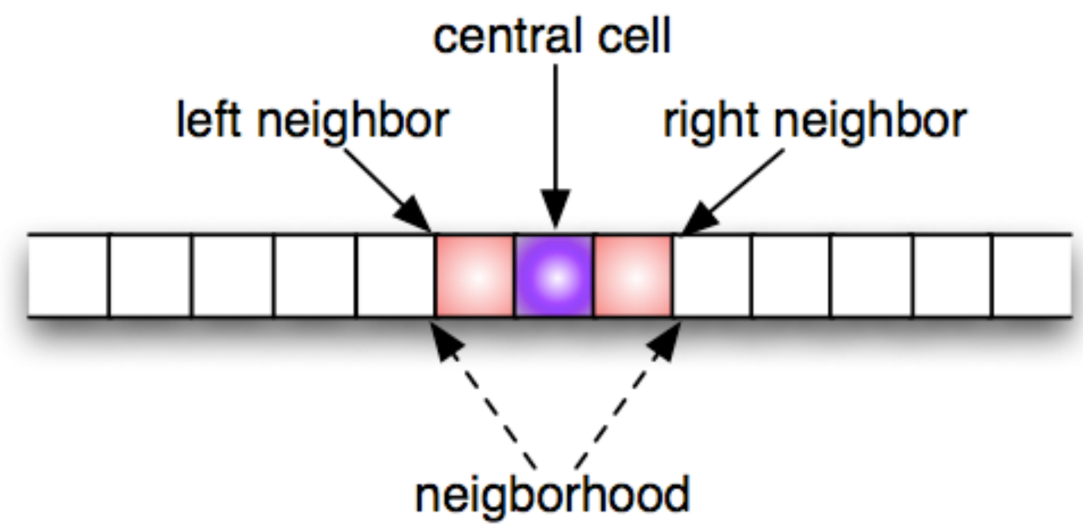


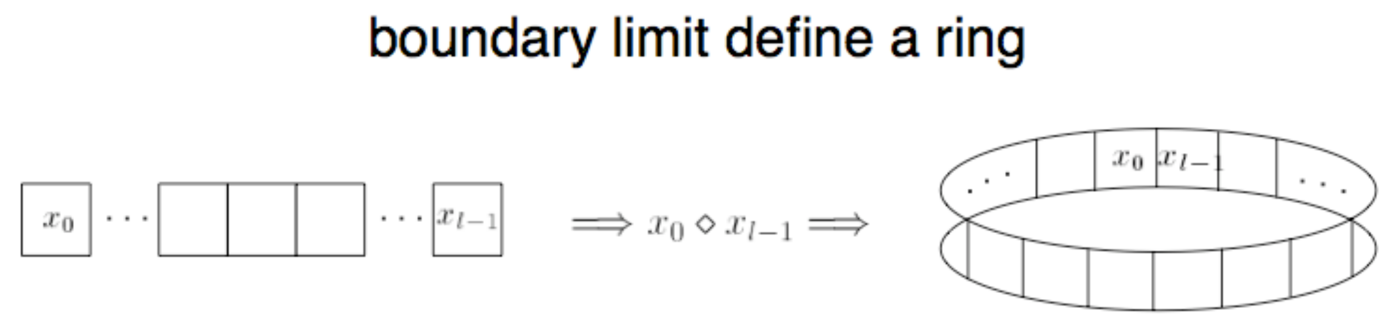
B2/S2345 29-sates CA (2008)

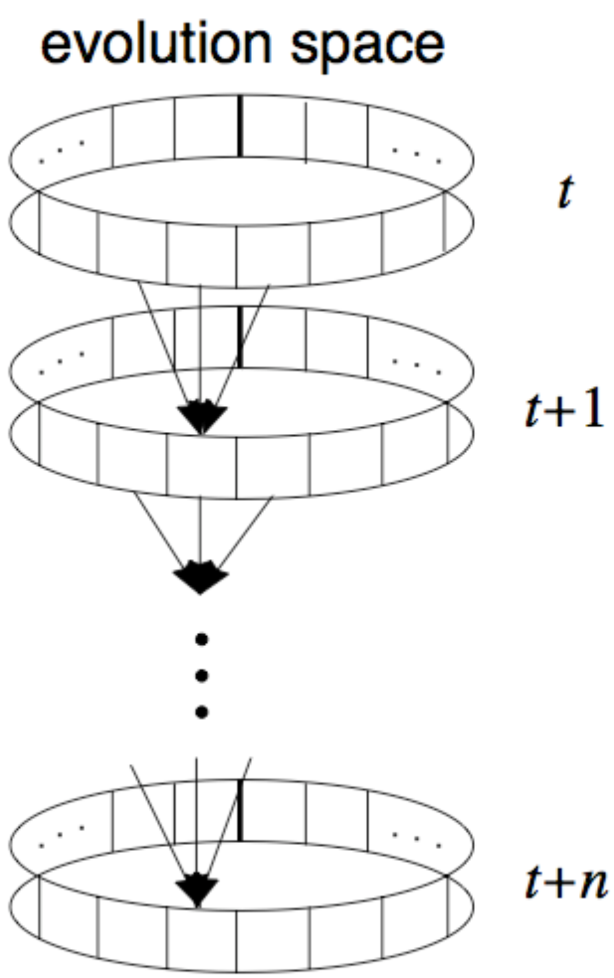
- By signals
- By gliders collision
- Propagation patterns

- Neumann, J. (1966) ***Theory of Self-reproducing Automata*** (edited and completed by A. W. Burks), University of Illinois Press, Urbana and London.
- Rendell, P. (2016) ***Turing machine universality of the game of life***. Springer International Publishing.
- Martínez, G.J., Adamatzky, A., Morita, K. & Margenstern, M (2010) "**Computation with competing patterns in Life-like automaton**." In: Game of Life Cellular Automata, pp. 547-572. Springer, London.
- **Golly** http://golly.sourceforge.net/

Dynamics in one dimension





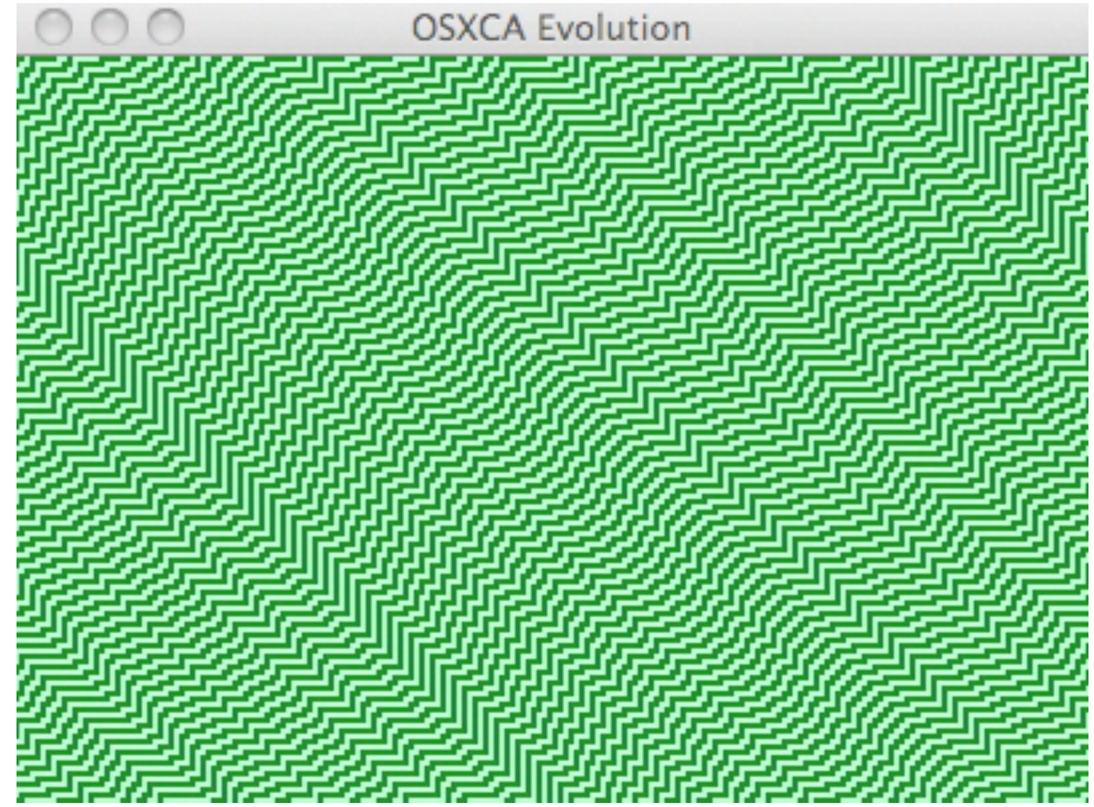


Elemental CA (ECA) is defined as follows:

- $\Sigma = \{0,1\}$
- $\mu = (x_{+1}, x_0, x_{-1})$ such that $x \in \Sigma$
- $\phi : \Sigma^3 \rightarrow \Sigma$
- $\mu = \{c_0 \mid x \in \Sigma\}$ the initial condition is the first ring with $t = 0$

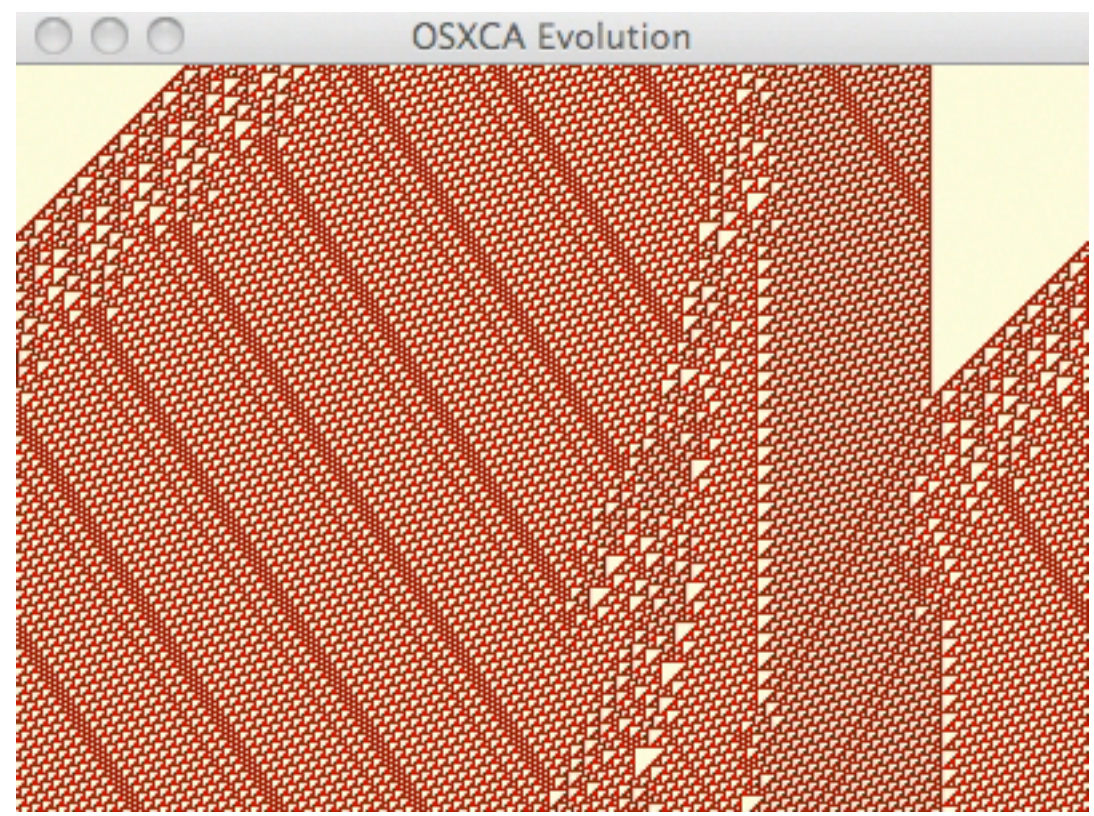
Cellular automata classes

class I: uniform (rule 168)



class II: periodic (rule 15)

class III: chaotic (rule 30)



class IV: complex (rule 110)

Class I: evolve to *uniform* behaviour;

Class II: evolve to *periodic* behaviour;

Class III: evolve to *chaotic* behaviour;

Class IV: evolve to *complex* behaviour.

classification		
type	num.	rules
class I	8	0, 8, 32, 40, 128, 136, 160, 168.
class II	65	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 19, 23, 24, 25, 26, 27, 28, 29, 33, 34, 35, 36, 37, 38, 42, 43, 44, 46, 50, 51, 56, 57, 58, 62, 72, 73, 74, 76, 77, 78, 94, 104, 108, 130, 132, 134, 138, 140, 142, 152, 154, 156, 162, 164, 170, 172, 178, 184, 200, 204, 232.
class III	11	18, 22, 30, 45, 60, 90, 105, 122, 126, 146, 150.
class IV	4	41, 54, 106, 110.

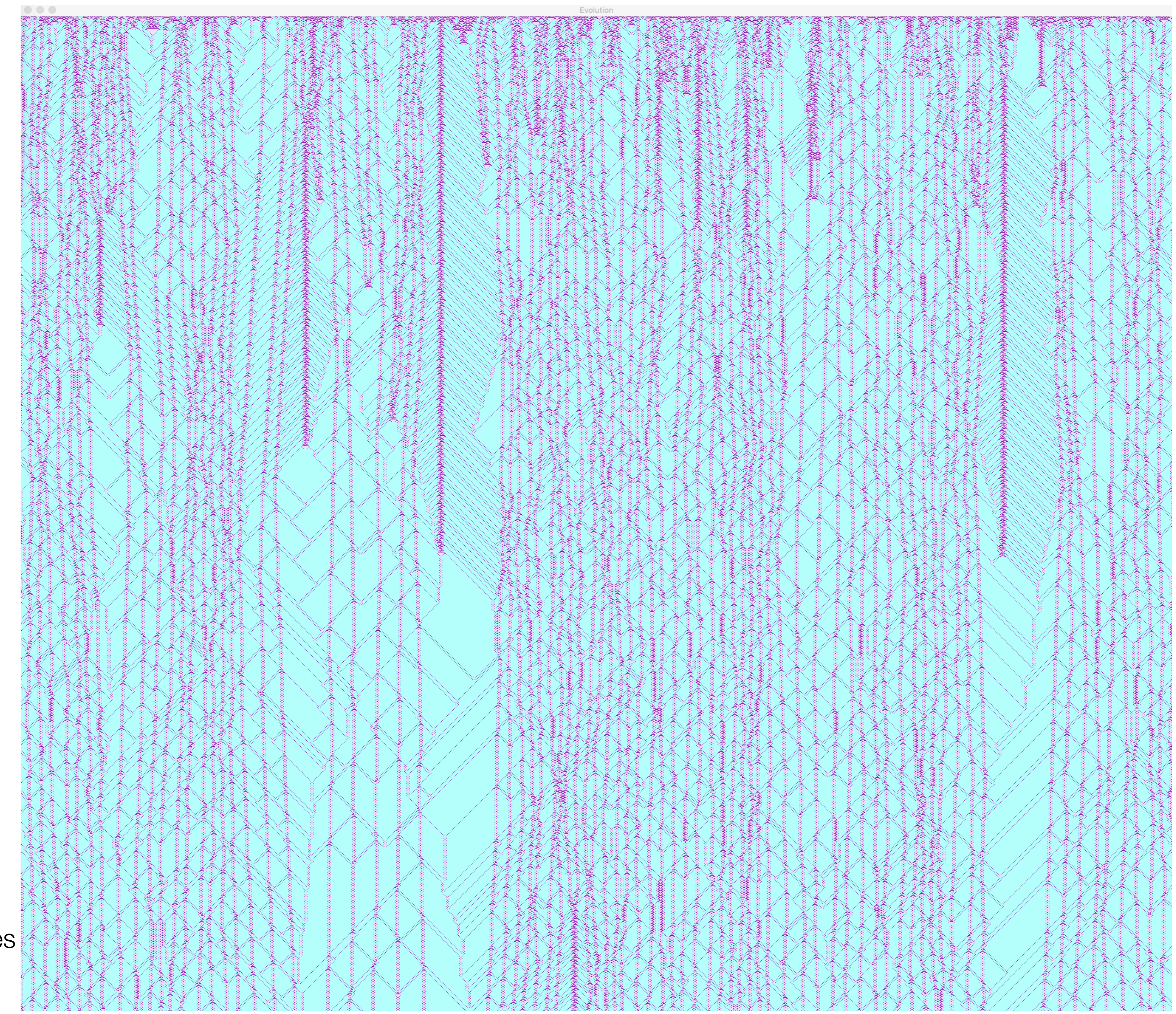
- Wolfram, S. (1994) ***Cellular Automata and Complexity***, Addison-Wesley Publishing Company,.
- Culik II, K. & Yu, S. (1988) **Undecidability of CA Classification Schemes**, *Complex Systems* 2, 177-190.
- McIntosh, H.V. (2009) ***One Dimensional Cellular Automata***, Luniver Press, United Kingdom.
- Martínez, G.J. (2013) **A Note on Elementary Cellular Automata Classification**, *Journal of Cellular Automata* 8(3-4) 233-259.

Elementary cellular automaton rule 54

Some interesting points in rule 54:

- Artificial life
- Complex systems
- Logical computation
- Garden of Eden configurations
- Symmetric evolutions
- Guns emerge from random conditions

1900 cells x 1640 times

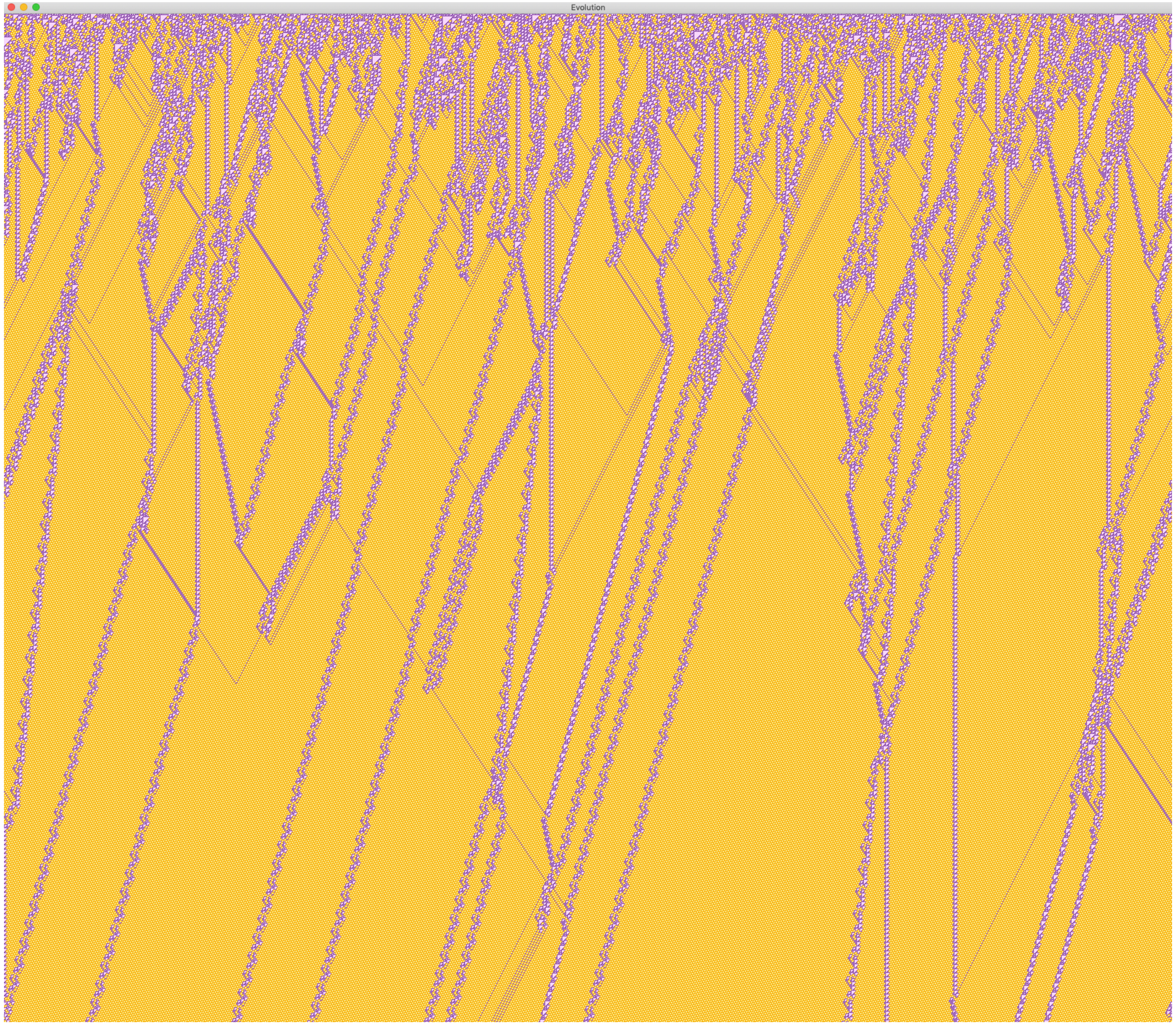


- Boccara, N., Nasser, J. & Roger, M. (1991) **Particle like structures and their interactions in spatio-temporal patterns generated by one-dimensional deterministic cellular automaton rules,** Physical Review A 44(2), 866-875.
- Hanson, J.E. & Crutchfield, J.P. (1997) **Computacional Mechanics of Cellular Automata: An Example**, Physics D 103(1-4), 169-189.
- Martin, B. (2000) **A Group Interpretation of Particles Generated by One-Dimensional Cellular Automaton, Wolfram's Rule 54**, Int. J. of Modern Physics C 11(1), 101-123.
- Martínez, G.J., Adamatzky, A. & McIntosh, H.V. (2006) **Phenomenology of glider collisions in cellular automaton Rule 54 and associated logical gates**, Chaos, Fractals and Solitons 28, 100-111.
- Guan, J. (2012) **Complex Dynamics of the Elementary Cellular Automaton Rule 54**, International Journal of Modern Physics C 23(7), 1250052.

Elementary cellular automaton rule 110

Some interesting points in rule 110:

- Artificial life
- Complex systems
- Universal computation
- Garden of Eden configurations
- Asymmetric evolutions
- Extendible gliders



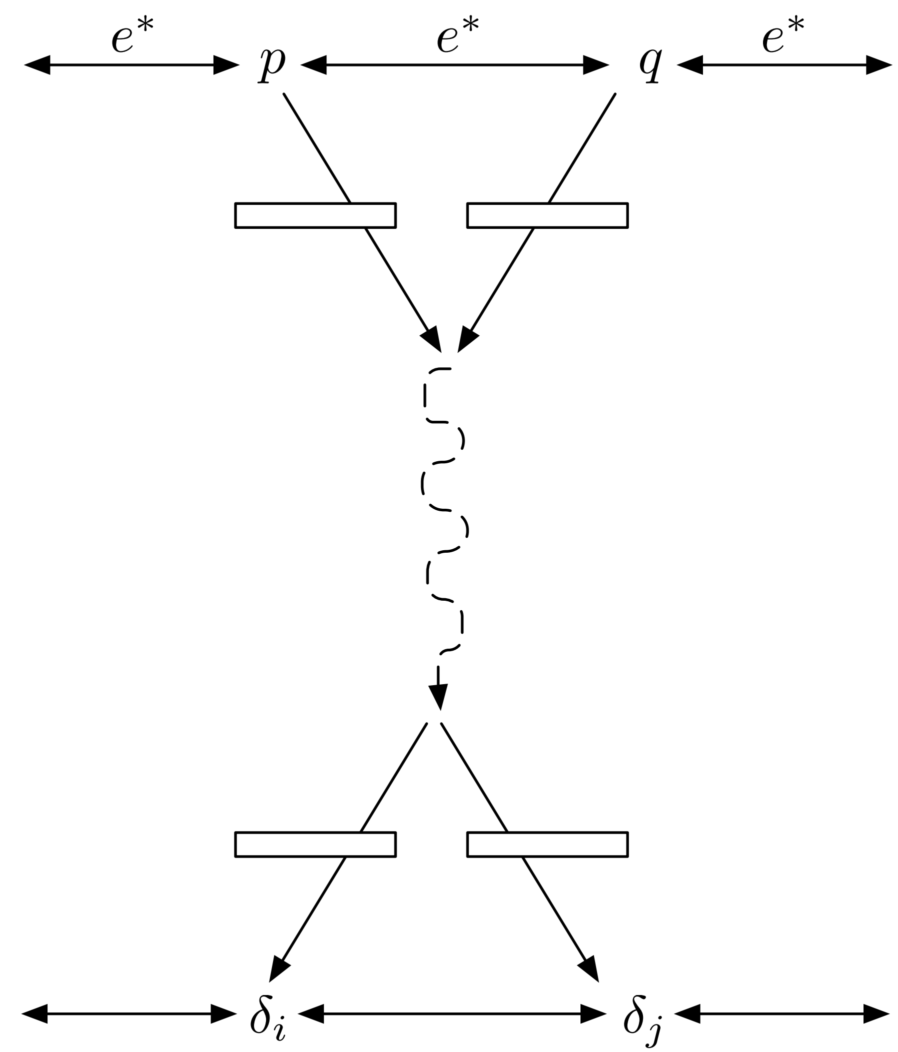
1900 cells x 1640 times

- Wolfram, S. (1994) **Cellular Automata and Complexity: collected papers**, Addison-Wesley Publishing Company.
- Cook, M. (1999) **Introduction to the activity of rule 110** (copyright 1994-1998 Matthew Cook), http://w3.datanet.hu/~cook/Workshop/CellAut/Elementary/Rule110/110pics.html, January.
- McIntosh, H.V. (1999) **Rule 110 as it relates to the presence of gliders**, http://delta.cs.cinvestav.mx/~mcintosh/oldweb/pautomata.html
- Martínez, G.J., McIntosh, H.V. & Mora, J.C.S.T. (2006) **Gliders in Rule 110**, International Journal of Unconventional Computing 2(1), 1-49.
- Mora, J.C.S.T., Martínez, G.J., Romero, N.H. & Marín, J.M. (2010) **Elementary cellular automaton Rule 110 explained as a block substitution**, Computing 88, 193-205.

Particles are strings

The formal languages theory provides a way to study sets of chains from a finite alphabet. The languages can be seen as inputs for some classes of machines or as the final result from a typesetter substitution system i.e., a generative grammar into the Chomsky's classification. This way, following a variation of a Feynman diagram hence we can represent collisions between particles in one-dimensional cellular automata as follows.

- p, q, δ – particles
- e – periodic background
- $\square$ – phase
- $f(p,q) \rightarrow \delta$



- Hurd, L.P. (1987) **Formal Language Characterizations of Cellular Automaton Limit Sets**, *Complex Systems 1*, 69-80.
- Wolfram, S. (1984) "**Computation Theory on Cellular Automata**," *Communication in Mathematical Physics 96*, 15-57.
- Hopcroft, J.E. & Ullman, J.D. (1987) ***Introduction to Automata Theory Languajes, and Computation***, Addison-Wesley Publishing Company.
- Collaborative Research Center SFB 676, **Particles, Strings, and the Early Universe**, Universität Hamburg , http://wwwiexp.desy.de/sfb676/

Particles are strings: Doug Lind starts this representation in ECA from 1986

Tables of Cellular Automaton Properties (1986)

Structures in rule 110.

The previous two pages show patterns produced by evolution according to rule 110, starting from a disordered initial configuration. The first picture shows all sites on a size 400 lattice. The second picture shows every other site in space and time on a size 800 lattice.

The configurations produced after many steps can be represented in terms of particle-like structures superimposed on a periodic background. The background is found to have spatial period 14 and temporal period 7, and corresponds to repetitions of the block **B** = 10011011111000. The configurations are then of the form ··· **BBBBPBBB** ··· , where the particles **P** that have been found so far are:

velocity	**P**
−6/12	10001100111011111111000
−2/4	11111000
−14/42	111000001110111111111111000
−8/30	1001100110001111111000
−4/15	00000
−4/36	111011111111000
−8/20	1111000011000
0/7	11111111000
0/7	100011000
0/7	10011011111111000
2/10	11101011000
2/10	1110100011011111000
2/3	111000

The "velocity" is written as (spatial period)/(temporal period).

One may speculate that the behaviour of rule 110 is sophisticated enough to support universal computation.

Table of particles by Doug Lind (*Mathematics Department, University of Washington, Seattle*).

Lind, D. (1994) **Structures in rule 110**, In: *Cellular Automata and Complexity*, Stephen Wolfram, Table of Properties, page 577, http://www.stephenwolfram.com/publications/articles/ca/86-caappendix/16/text.html

Particles are strings and filters: James Hanson and James Crutchfield describing finite state machines in ECA from 1997

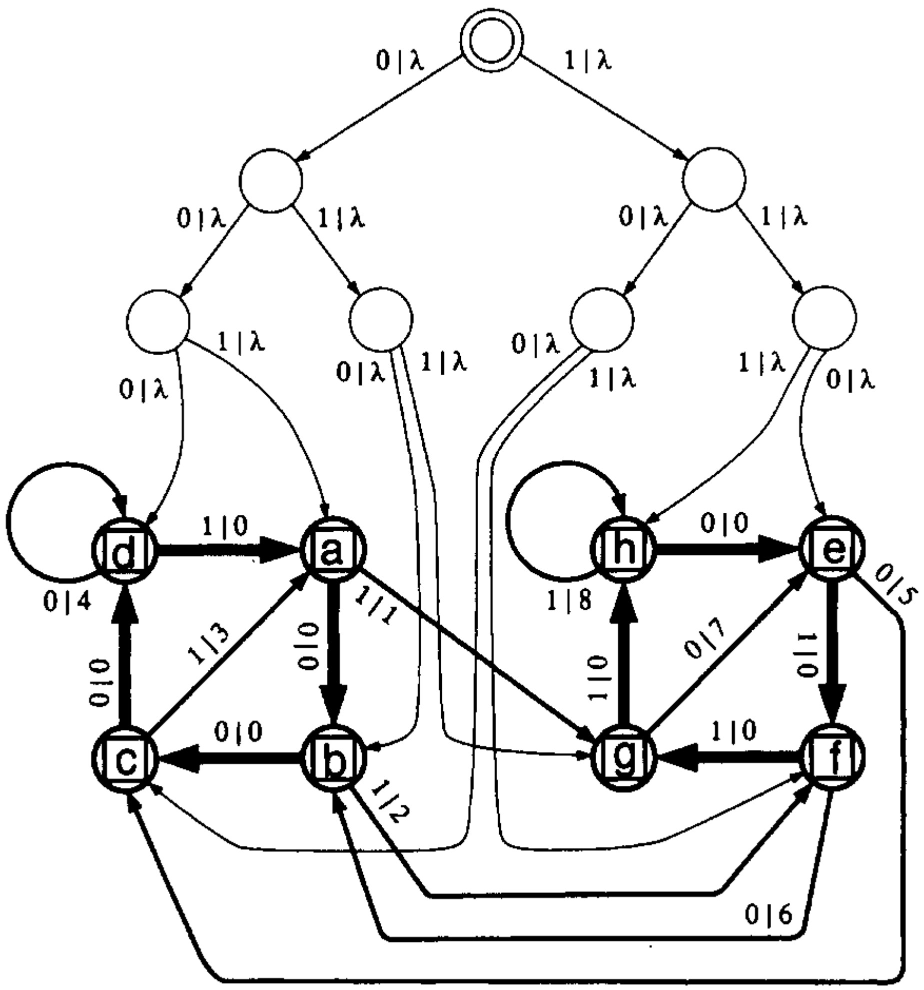


Fig. 5. ECA 54's domain filter T^0_{54}, which maps sites in the domain to 0 and each defect to a unique output in $\{1, \ldots, 8\}$. Labeled machine states correspond to the domain states of Fig. 3.

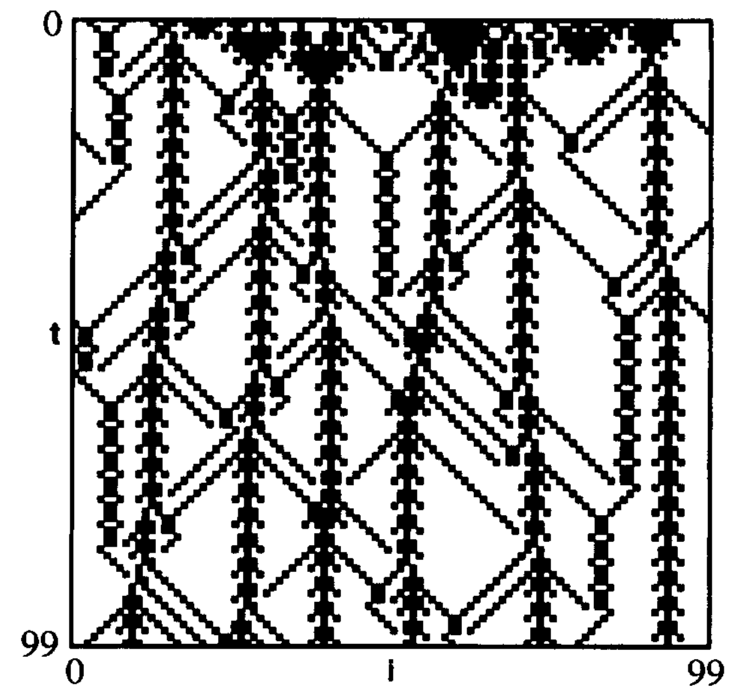


Fig. 6. Space–time data of Fig. 1, filtered with the domain transducer T^0_{54} of Fig. 5. White cells correspond to sites participating in Λ_{54}; black cells, to sites with values $s^i_t \in \{1, \ldots, 8\}$.

Table 1
Fundamental interactions among ECA 54's particles

(a)	$\alpha + \gamma^- \to \gamma^- + \alpha + 2\gamma^+$
(b)	$\gamma^+ + \alpha \to 2\gamma^- + \alpha + \gamma^+$
(c)	$\beta + \gamma^- \to \gamma^+$
(d)	$\gamma^+ + \beta \to \gamma^-$
(e)	$\gamma^+ + \gamma^- \to \beta$
(f)	$\gamma^+ + \alpha + \gamma^- \to \gamma^- + \alpha + \gamma^+$
(g)	$\gamma^+ + \beta + \gamma^- \to \emptyset$

Hanson, J.E. & Crutchfield, J.P. (1997) **Computational mechanics of cellular automata: An example**, *Physica D*, 103(1-4), 169-189.

Particles are strings: Harold McIntosh stablished that the problem of rule 110 is a problem of tiles in 1998

24 *CHAPTER 1. OVERVIEW*

1.4.4 ether crystallography

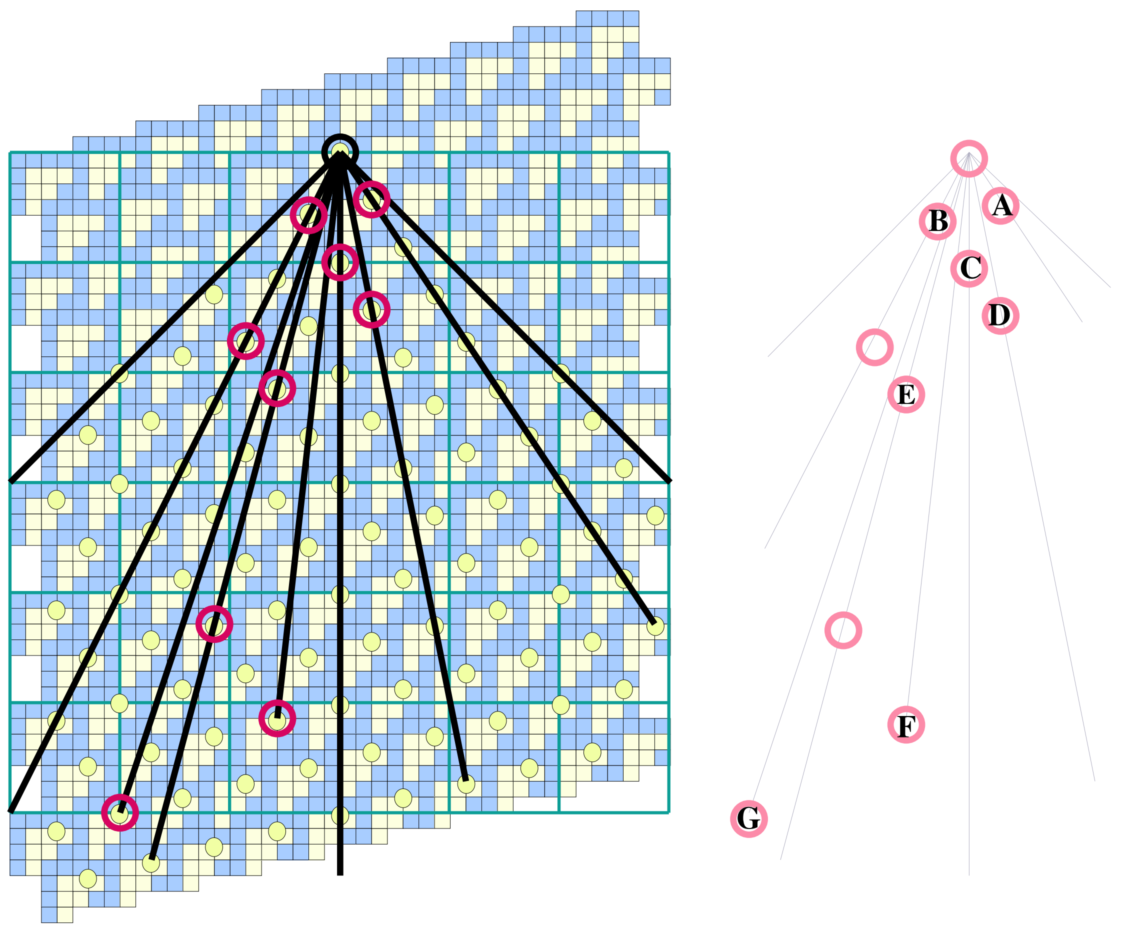


Figure 1.13: The locations of Cook's gliders relative to the ether lattice. The two barred gliders sit lower on the same velocity lines as the unbarred gliders. Small circles on the T3 mosaic show possible positions of compatible gliders, but they could be impossible, duplicates, or so far undiscovered.

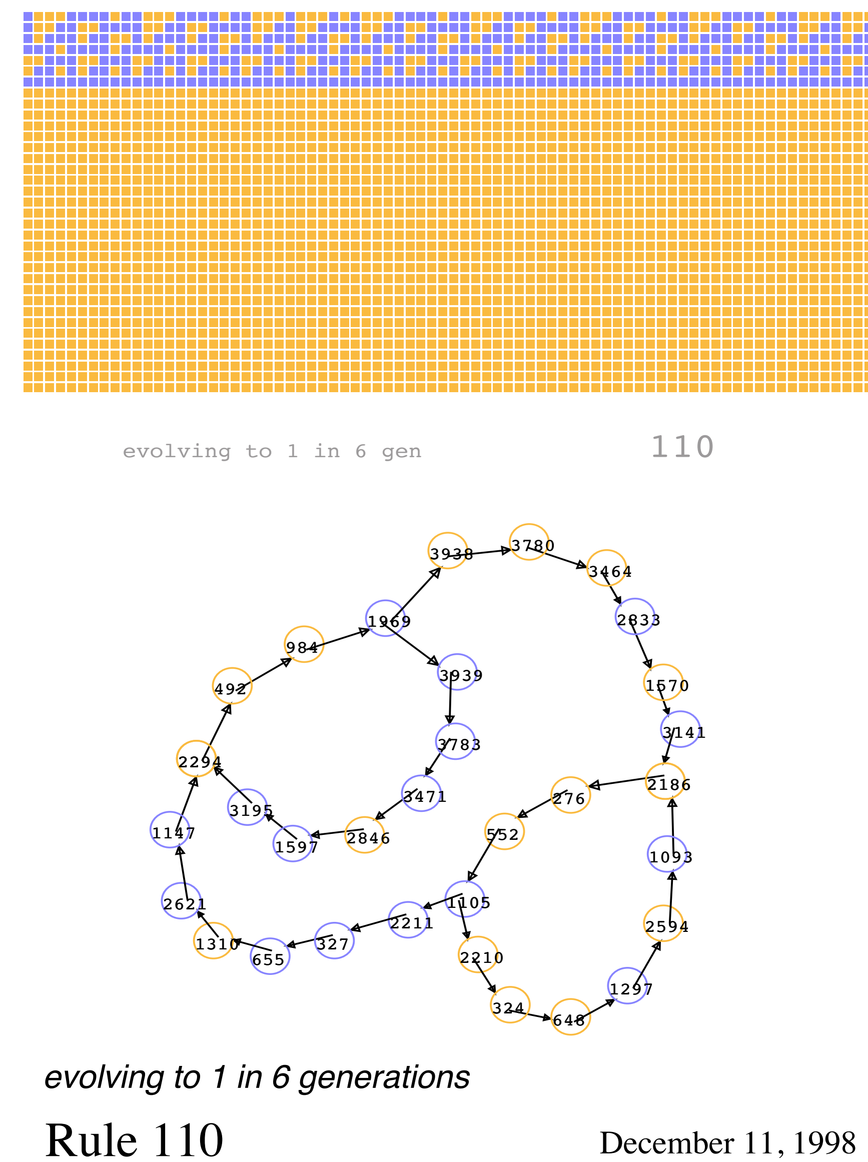
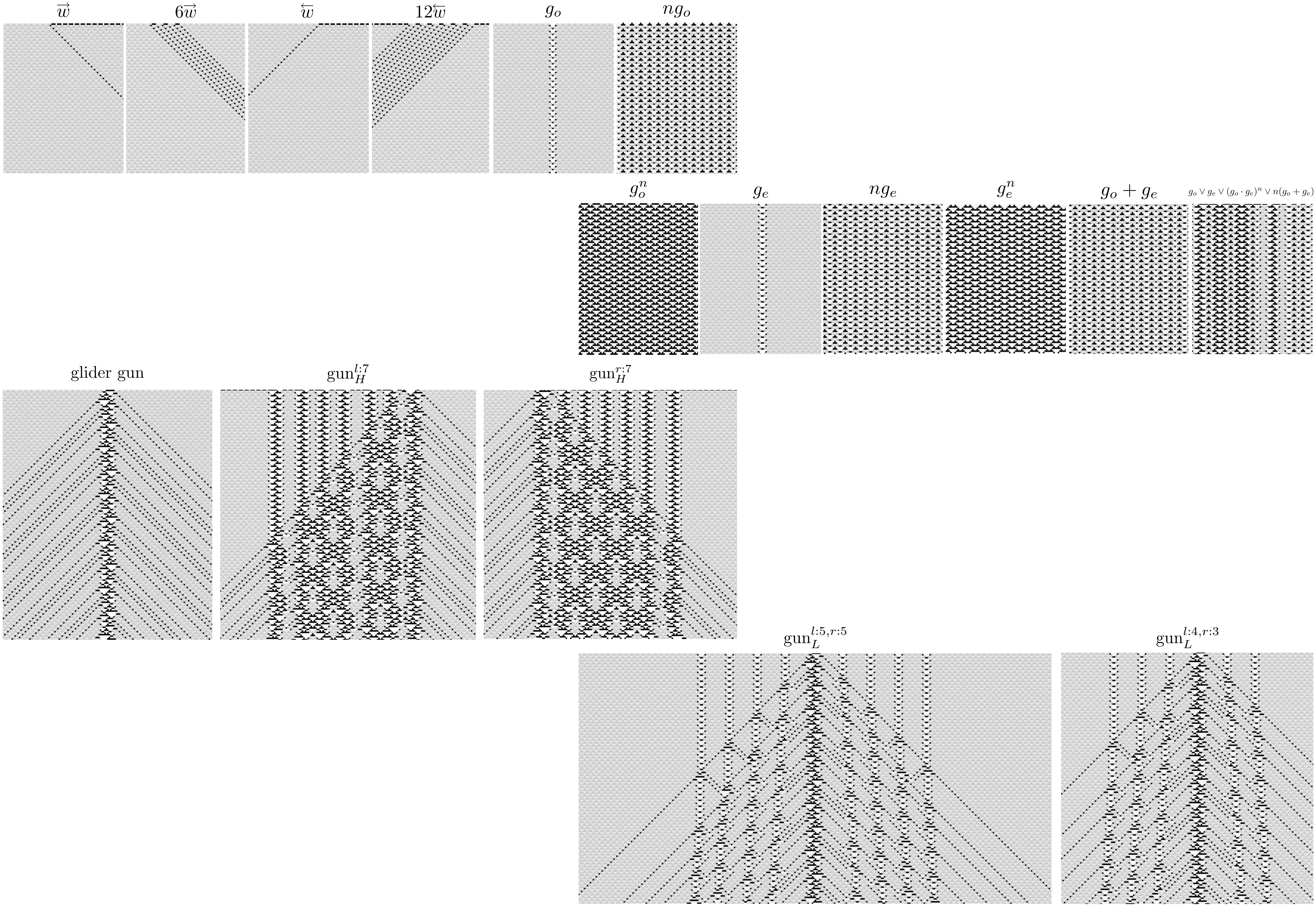


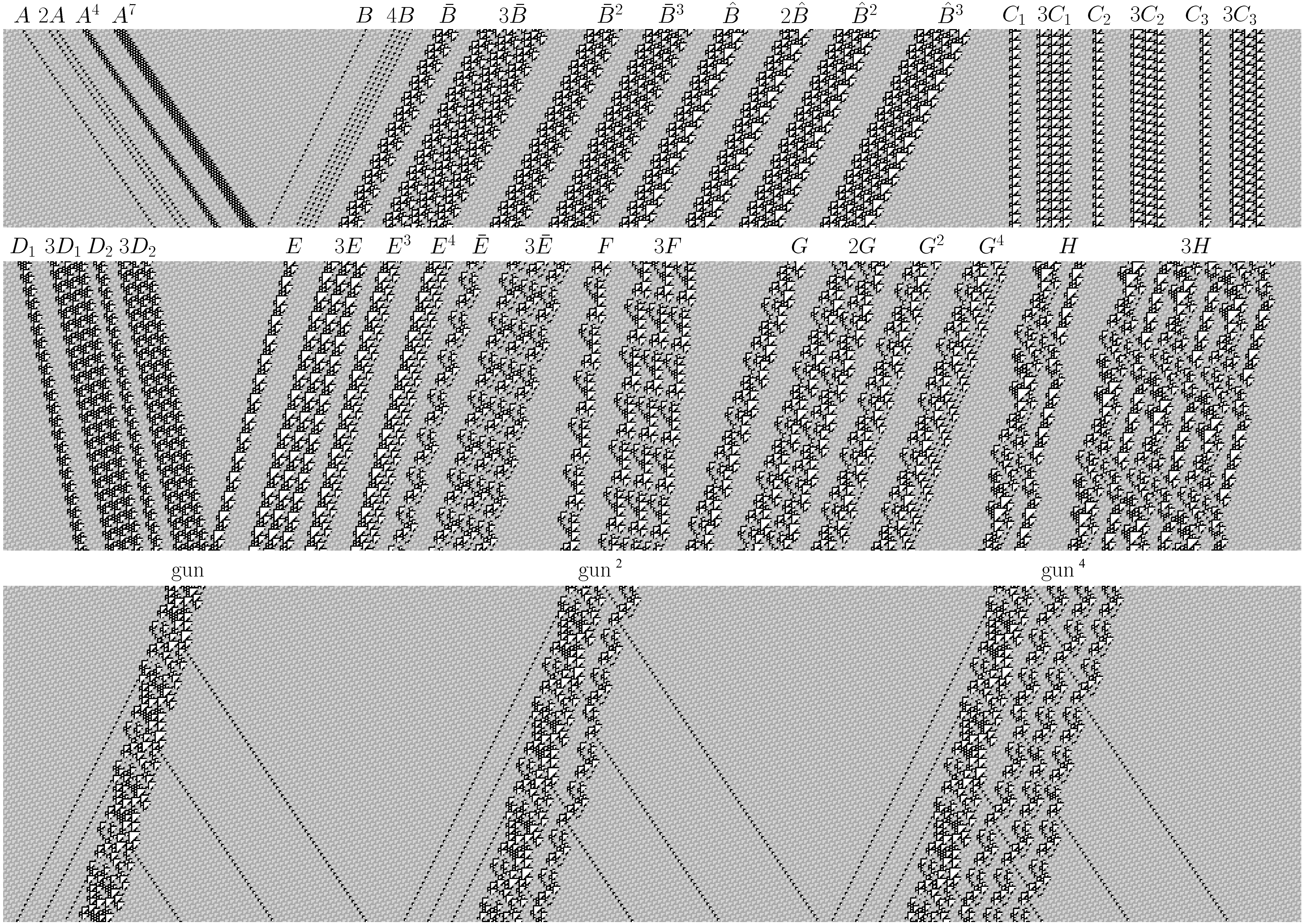
Figure 1.22: The de Bruijn diagram for evolution to the constant 1 after six generations.

McIntosh, H.V. (1999) **Rule 110 as it Relates to the Presence of Gliders.** Technical report UAP. http://delta.cs.cinvestav.mx/~mcintosh/cellularautomata/Papers.html

Set of particles in rule 54 — https://www.comunidad.escom.ipn.mx/genaro/Rule54.html



Set of particles in rule 110 — https://www.comunidad.escom.ipn.mx/genaro/Rule110.html



Particles as strings from finite states machines

For an one-dimensional cellular automaton of order (k,r), the *de Bruijn diagram* is defined as a directed graph with k^{2r} vertices and k^{2r+1} edges. The vertices 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 that the 2r-1 initial ones in j forming a neighbourhood of 2r+1 states represented by $i \lozenge j$. In this case, the edge connecting i to j is labeled with $\phi(i \lozenge j)$.

The connection matrix M corresponding with the de Bruijn diagram is as follows:

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

Basins of attraction or cycle diagrams calculate attractors in a dynamical system, as was extensively studied by Andrew Wuensche in CA and random Boolean networks. Given a sequence of cells x_i we define a configuration c of the system. An evolution is represented by a sequence of configurations $c_0, c_1, c_2, \ldots, c_{m-1}$ given by the global mapping,

$$\Phi : \Sigma^n \to \Sigma^n$$

and the global relation is given for the next function between configurations,

- McIntosh, H.V. (1991) **Linear cellular automata via de Bruijn diagrams**, http://delta.cs.cinvestav.mx/~mcintosh/oldweb/pautomata.html
- Sutner, K. (1991) **De Bruijn Graphs and Linear Cellular Automata**, Complex Systems 5(1) 19-30.
- Wuensche, A. & Lesser, M. (1992) **Global Dynamics of Cellular Automata,** Addison-Wesley Publishing Company.
- Voorhees, B.V. (1996) ***Computational analysis of one-dimensional cellular automata***, World Scientific Series on Nonlinear Science, Series A, Vol. 15.
- McIntosh, H.V. (2009) ***One Dimensional Cellular Automata***, Luniver Press.

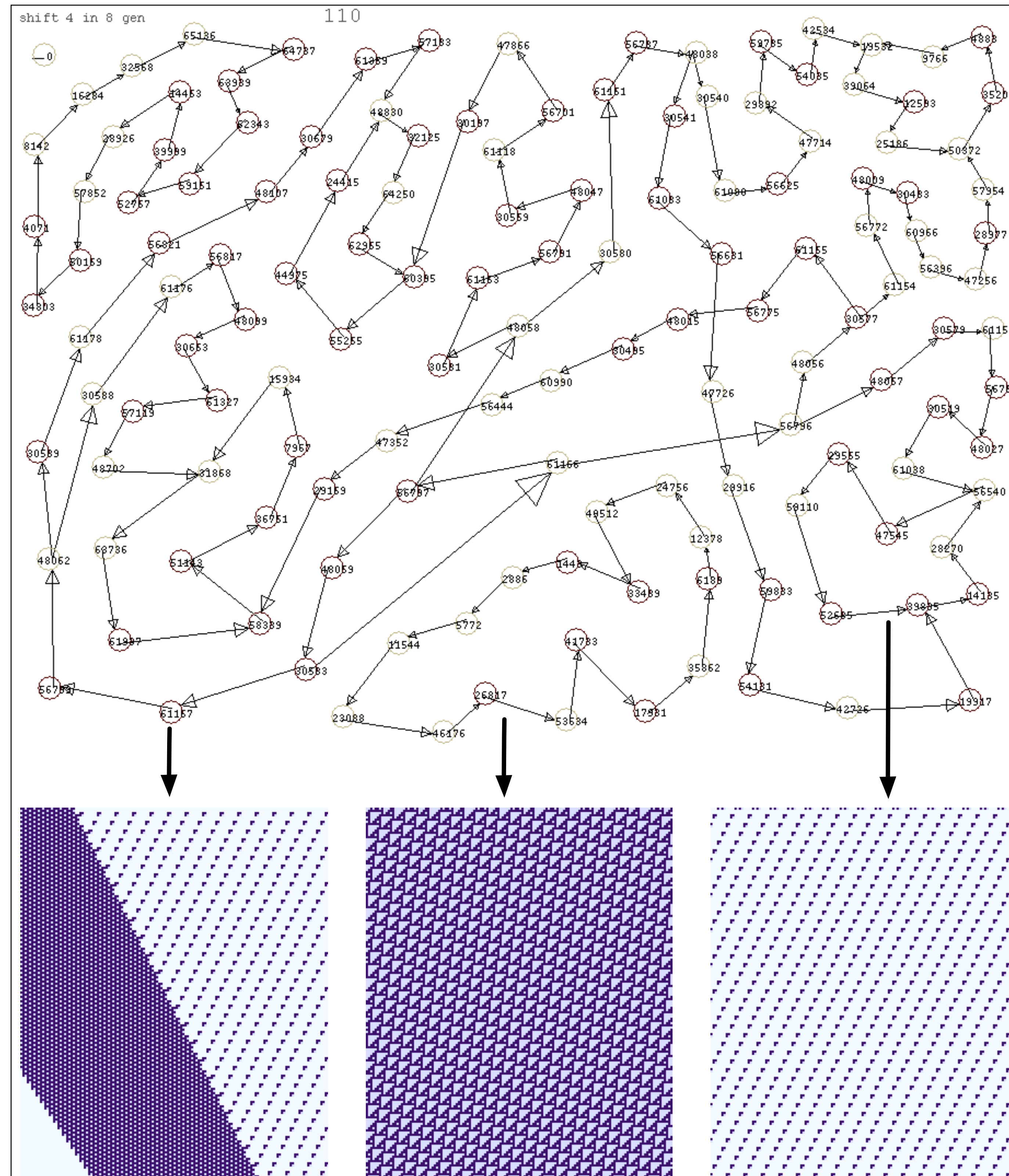
Particles as strings from finite states machines

De Bruijn diagram (displacement 4, period 8) calculating non-stationary particles in rule 110. The left evolution displays a *fuse pattern* produced by two particles colliding and both annihilated. The center evolution displays a periodic pattern and the right evolution displays particles with displacement to the left.

$$N = \{61166, 56799, 48059, 30583, 61167, 56703, 48062, 30589, 61178, 56821, 48107, 30679, 61369, 57183, 48830, 32125, 64250, 62965, 60395, 55255, 44975, 24415\}$$

$$((0111)^* + 11(01011111)^*)^*$$

It is an expression to codify *A* and *B* particles.



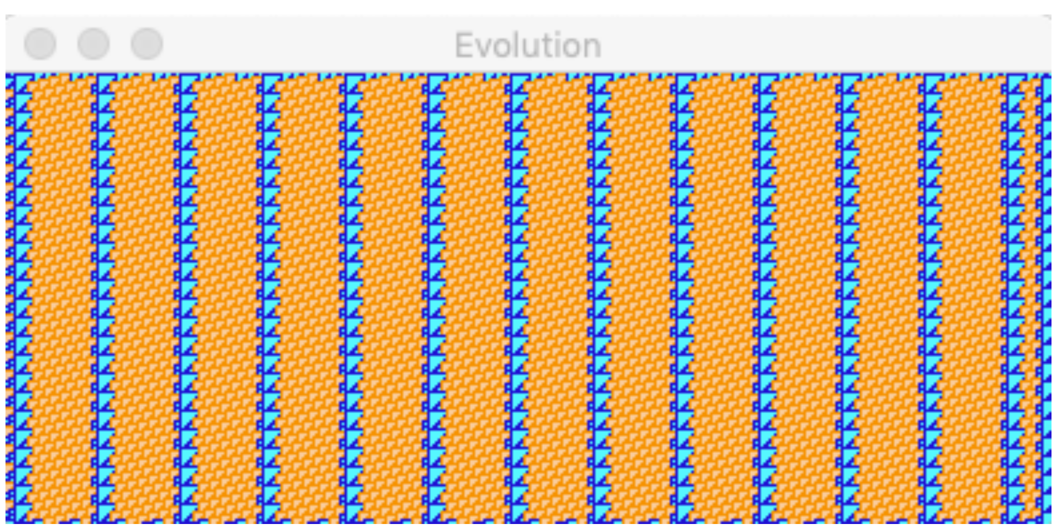
Martínez, G.J., Adamatzky, A., Chen, B., Chen, F. & Mora, J.C.S.T. (2018). **Simple networks on complex cellular automata: from de Bruijn diagrams to jump-graphs**. In: Evolutionary Algorithms, Swarm Dynamics and Complex Networks (pp. 241-264). Springer, Berlin, Heidelberg.

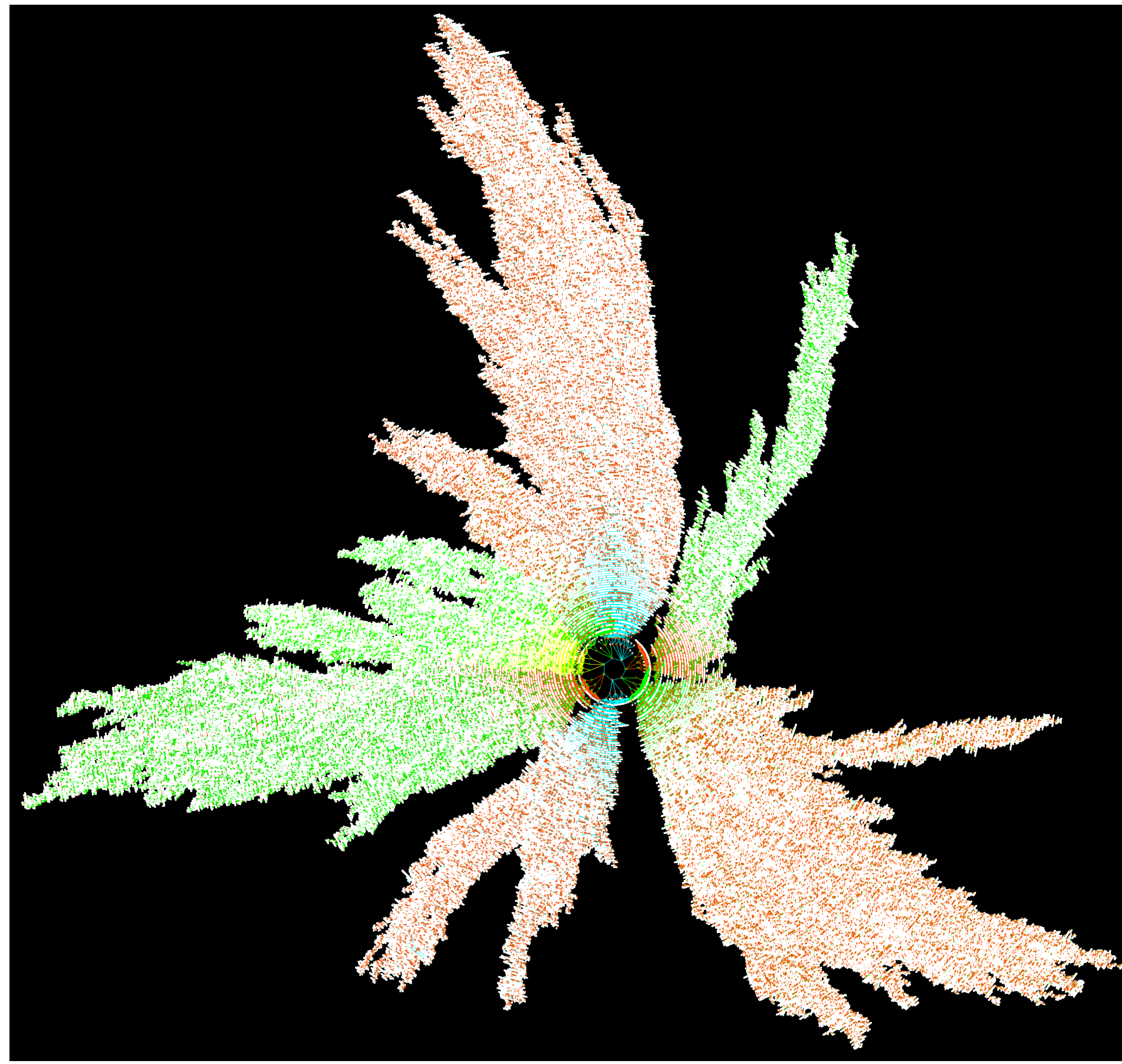
Particles as strings from finite states machines

Attractor length 31, period 7, with a mass of $5.487\text{x}10^7$ configurations for rule 110. This attractor is in a field of 6,326 basins.

$$
\begin{aligned}
w_0 &= 0110111111110001001101111100010\\
w_1 &= 1111100000010011011111000100110\\
w_2 &= 1000100000110111110001001101111\\
w_3 &= 1001100001111100010011011111000\\
w_4 &= 1011100011000100110111110001001\\
w_5 &= 1110100111001101111100010011011\\
w_6 &= 0011101101011111000100110111110
\end{aligned}
$$

They are expressions to codify stationary (*C*s) particles.





Martínez, G.J., Adamatzky, A., Chen, B., Chen, F. & Mora, J.C.S.T. (2018). **Simple networks on complex cellular automata: from de Bruijn diagrams to jump-graphs**. In: Evolutionary Algorithms, Swarm Dynamics and Complex Networks (pp. 241-264). Springer, Berlin, Heidelberg.

Particles as strings in rule 54

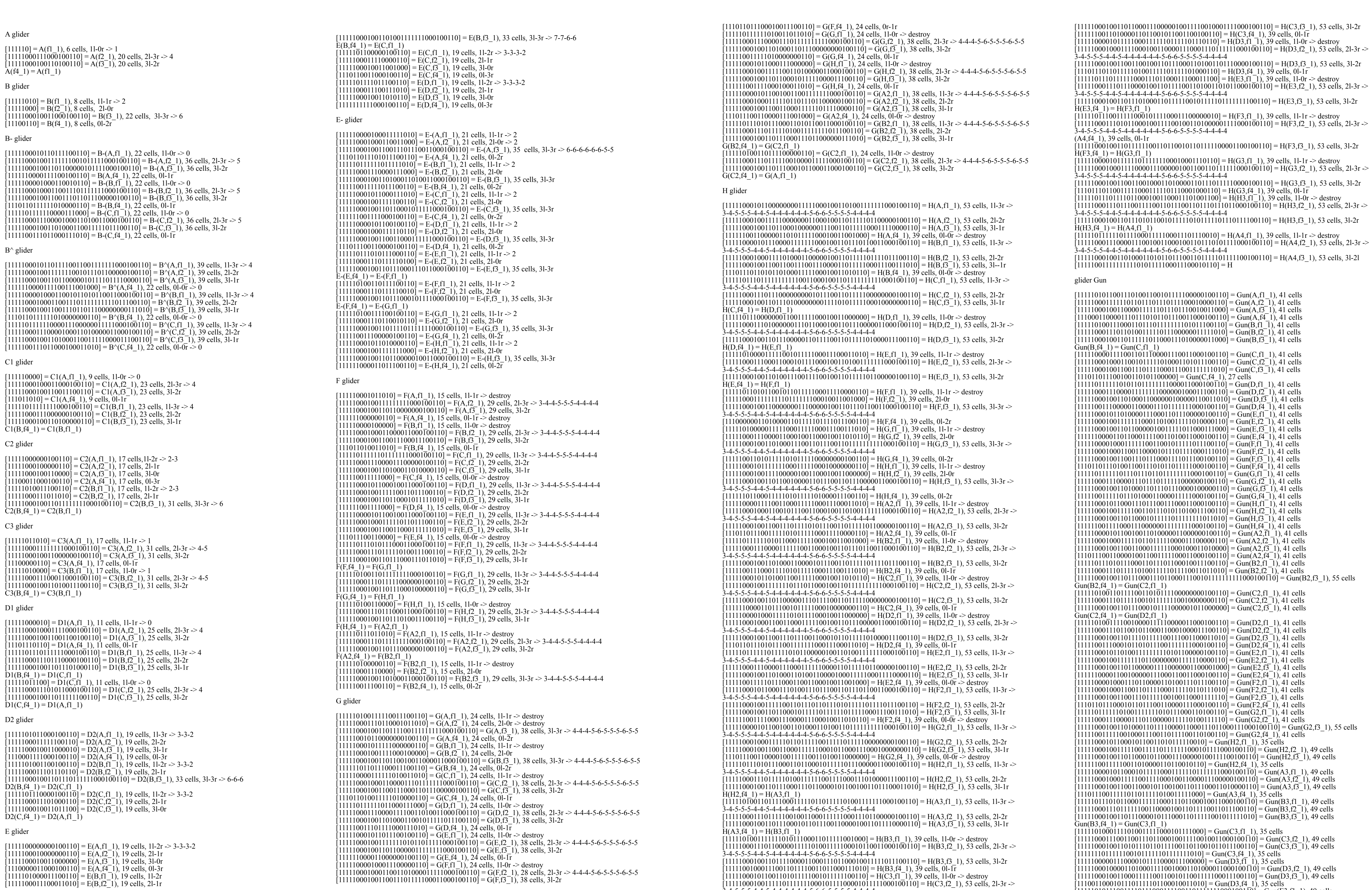
e	$\overrightarrow{w}$	$\overleftarrow{w}$
$e_1 = 1000$	$\overrightarrow{w}(\mathrm{f}_1) = e_1$-10-$e_2$	$\overleftarrow{w}(\mathrm{f}_1) = e_1$-$e_2$
$e_2 = 1110$	$\overrightarrow{w}(\mathrm{f}_2) = e_2$-00-$e_1$	$\overleftarrow{w}(\mathrm{f}_2) = e_2$-$e_1$
	$2\overrightarrow{w}(\mathrm{f}_1) = e_1$-10111000-$e_1$	$2\overleftarrow{w}(\mathrm{f}_1) = e_1$-11101000-$e_1$
		$2\overleftarrow{w}(\mathrm{f}_2) = e_2$-11101000-$e_2$

	g_o	g_e
	$g_o(\mathrm{A,f}_1) = e_1$-100000-$e_1$	$g_e(\mathrm{A,f}_1) = e_1$-1000000-$e_1$
	$g_o(\mathrm{A,f}_2) = e_2$-111110-$e_2$	$g_e(\mathrm{A,f}_2) = e_2$-000-$e_2$
	$g_o(\mathrm{B,f}_1) = e_1$-10-$e_1$	$g_e(\mathrm{B,f}_1) = e_1$-100-$e_1$
	$g_o(\mathrm{B,f}_2) = e_2$-00-$e_2$	$g_e(\mathrm{B,f}_2) = e_2$-1111110-$e_2$

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

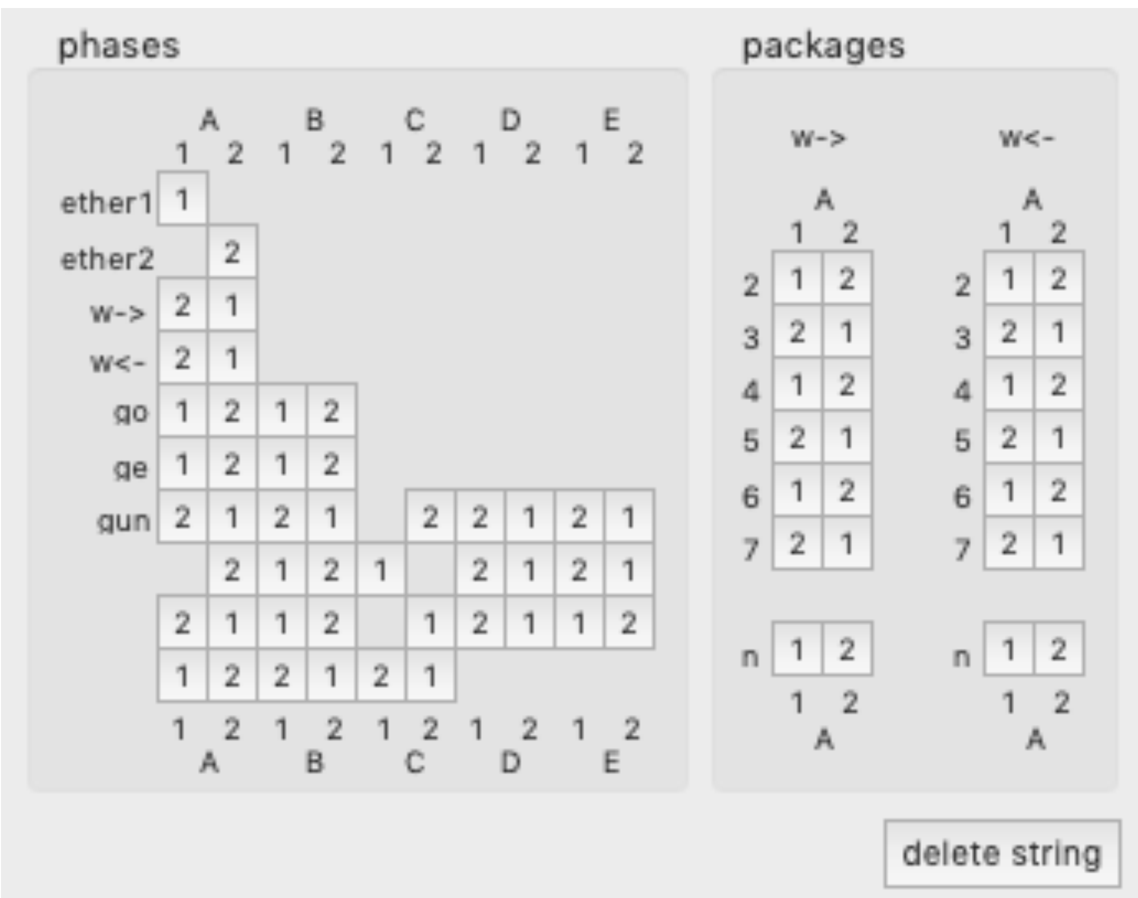
- Martínez, G.J., Adamatzky, A. & McIntosh, H.V. (2014) **Complete Characterization of Structure of Rule 54**, Complex Systems 23(3), 259-293.
- Martínez, G.J., Adamatzky, A. & McIntosh, H.V. (2008) **On the representation of gliders in Rule 54 by de Bruijn and cycle diagrams**, Lecture Notes in Computer Science 5191, 83-91.

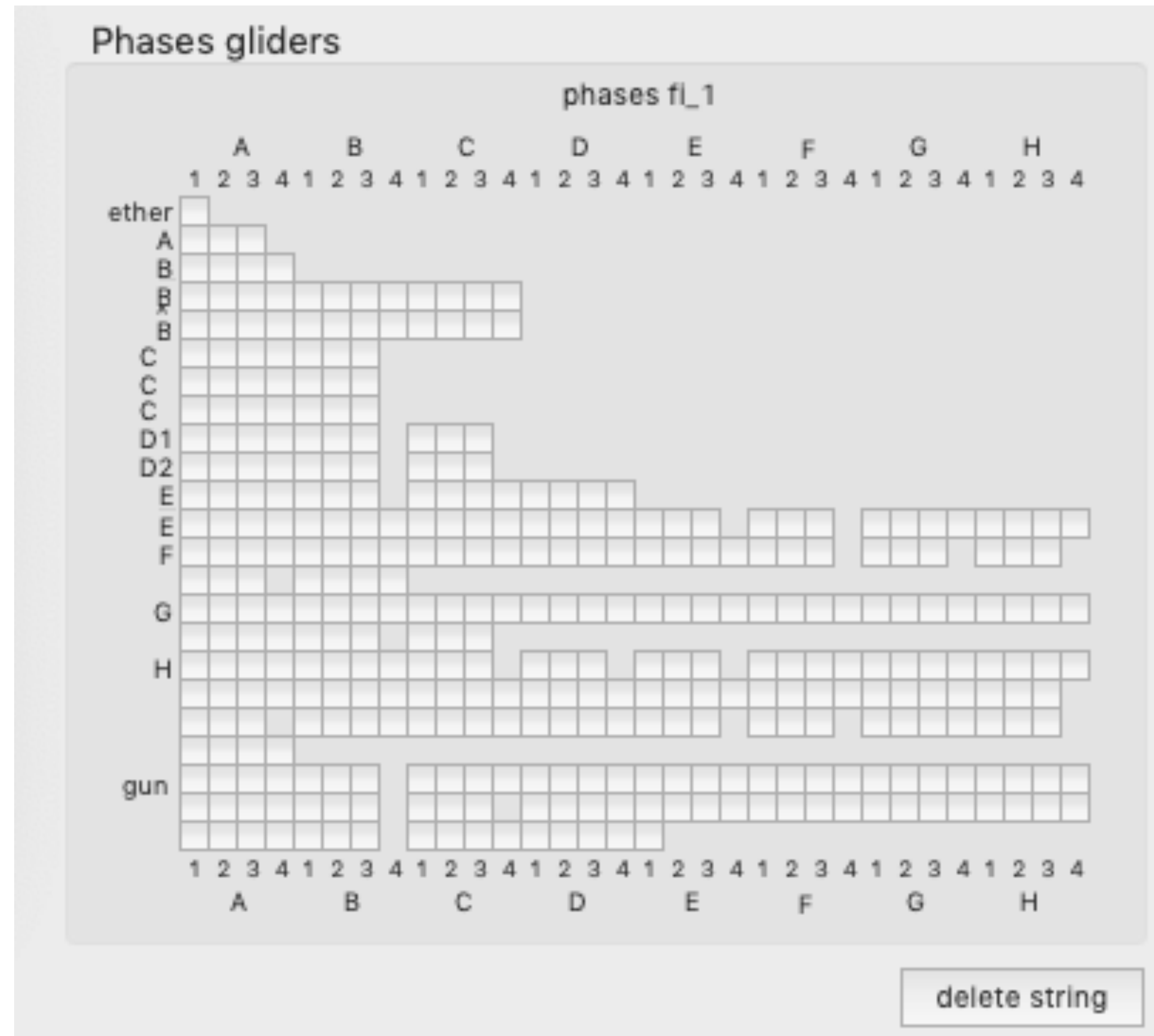
Particles as strings in rule 110



- Martínez, G.J., McIntosh, H.V., Mora, J.C.S.T. & Vergara, S.V.C. (2008) **Determining a regular language by glider-based structures called phases fi_1 in Rule 110**, Journal of Cellular Automata 3(3), 231-270.
- Martínez, G.J., McIntosh, H.V., Mora, J.C.S.T. & Vergara, S.V.C. (2008) **A note about the regular language of Rule 110 and its general machine: the scalar subset diagram**, Japan Society for Artificial Intelligence C3004, 39-49, Yokohama, Japan.
- Martínez, G.J. (2001) **Particles as strings in rule 110**. https://www.comunidad.escom.ipn.mx/genaro/rule110/listPhasesR110.txt

Particles as strings are implemented in OSXLCAU21 and CAViewer software



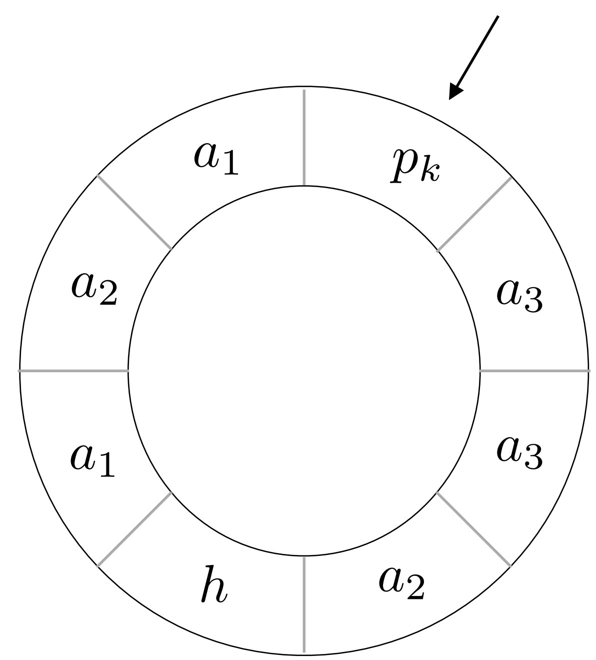


Cellular automata software. https://www.comunidad.escom.ipn.mx/genaro/Cellular_Automata_Repository/Software.html

Circular computation

We have two important previous results in computer science theory to think about of circular computation. Arbib presents a circular Turing machine in 1962 and Kudlek and Rogozhin presents circular Post machines in 2001.

- a_i, h : symbols
- h : the limit of the type
- p_k : state
- $\rightarrow$: head



Important features: (1) label and limiting the end of the type, (2) the movement is turning the type, (3) the type can introduce new squares.

- Arbib, M.A. (1962) **Monogenic Normal Systems are Universal**, Monogenic normal systems are universal. Journal of the Australian Mathematical Society, 3(3) 301-306.
- Kudlek, M. & Rogozhin, Y. (2001) **Small Universal Circular Post Machines,** Computer Science Journal of Moldova 9(1) 34-52.

Symbol super colliders (Tommaso Toffoli, 2002)

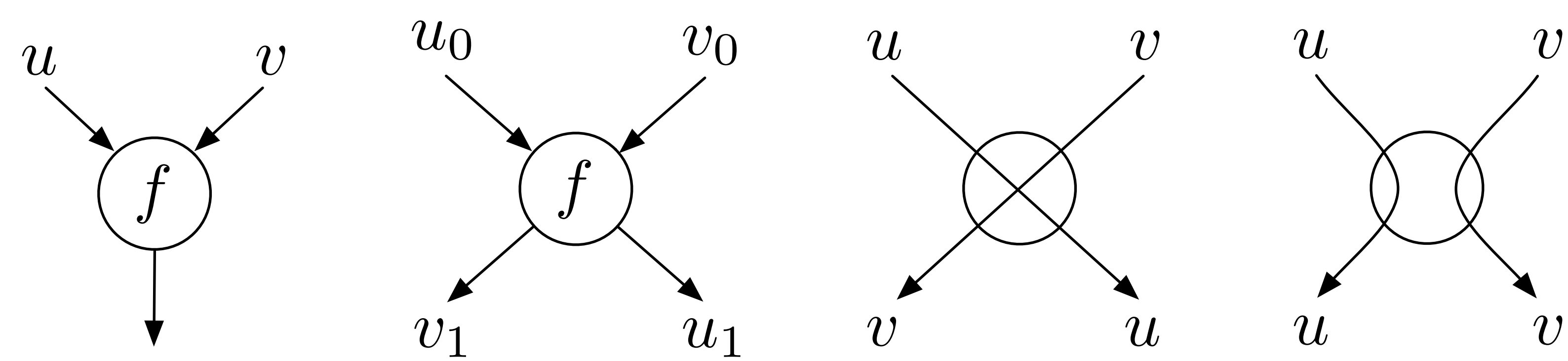
In 2002, Toffoli exposed the concept of “symbol super collider”. To map Toffoli's supercollider onto a one-dimensional cellular automata we use the notion of an idealized particle $p \in Z^+$ (without energy and potential energy). The particle p is represented by a binary string of cell states. Typically, we can find all types of particles manifest in cellular automata particles, including positive p^+, negative p^-, and neutral p^0 displacements, and also composite particles assembled from elementary particles.

(a) $f(u, v)$ is a product of one collision

(b) $f(u, v) = u + v$ union

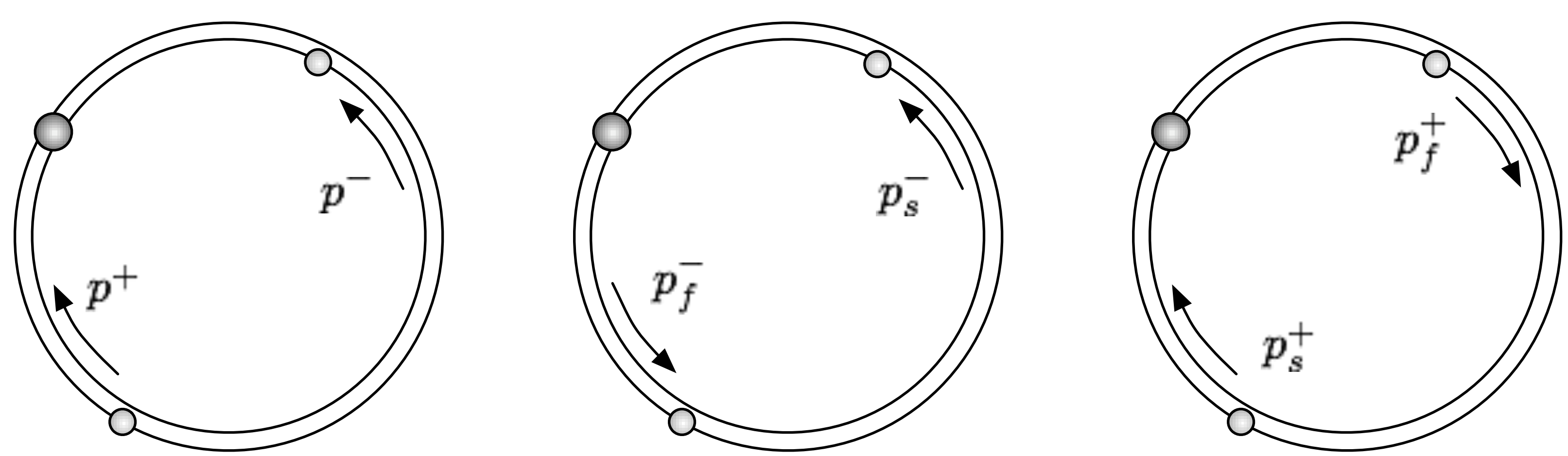
(c) $f_i(u, v) \mapsto (u, v)$ identity

(d) $f_r(u, v) \mapsto (v, u)$ reflection



(a) (b) (c) (d)

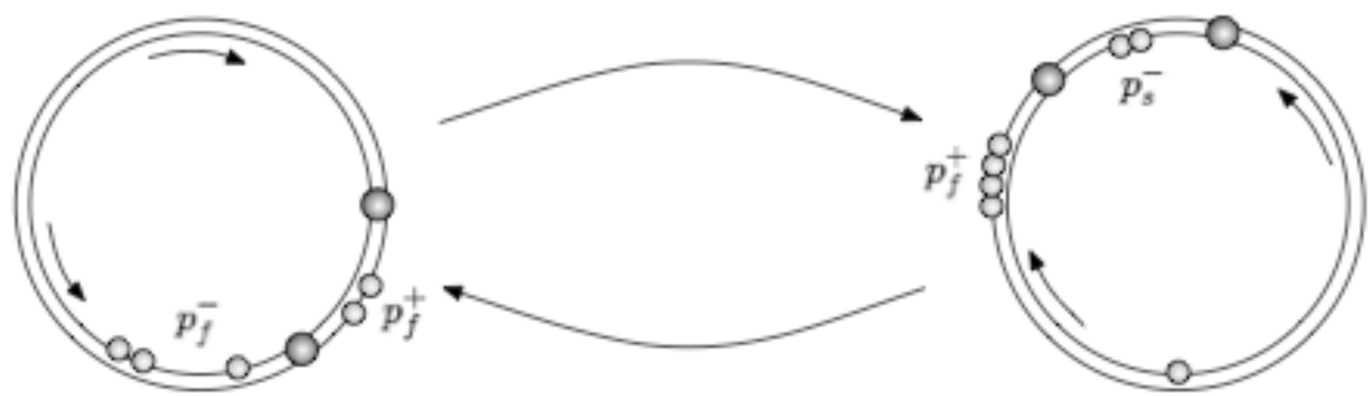
Schemes of ballistic collisions between particles evolving in *cyclotrons*. Gray circles represent the contact point of collision.



Toffoli, T. (2002) **Symbol Super Colliders**, In: Adamatzky, A. (Ed.) *Collision-Based Computing*, Springer, chapter 1, pages 1-22.

Symbol super colliders

Transition between two beam routing synchronizing multiple reactions. When the first set of collisions are done a new beam routing is defined with other particles, so that when the second set of collisions is done then one returns to the initial condition of the first beam, constructing a meta-glider or mesh in Rule 110.



In this way, we can design more complex constructions synchronizing multiple collisions with a diversity of speeds and phases on different particles. Figure displays a more sophisticated beam routing design, connecting two of beams and then creating a new beam routing diagram where edges represent a change of particles and collisions contact point on ECA Rule 110. In such a transition, a number of new particles emerge and collide to return to the first beam, thus oscillating between two beam routing forever.

$$p_A^+, p_A^+ \leftrightarrow p_{\bar{B}}^-, p_B^-, p_B^-$$

changing to the set of particles (second beam routing):

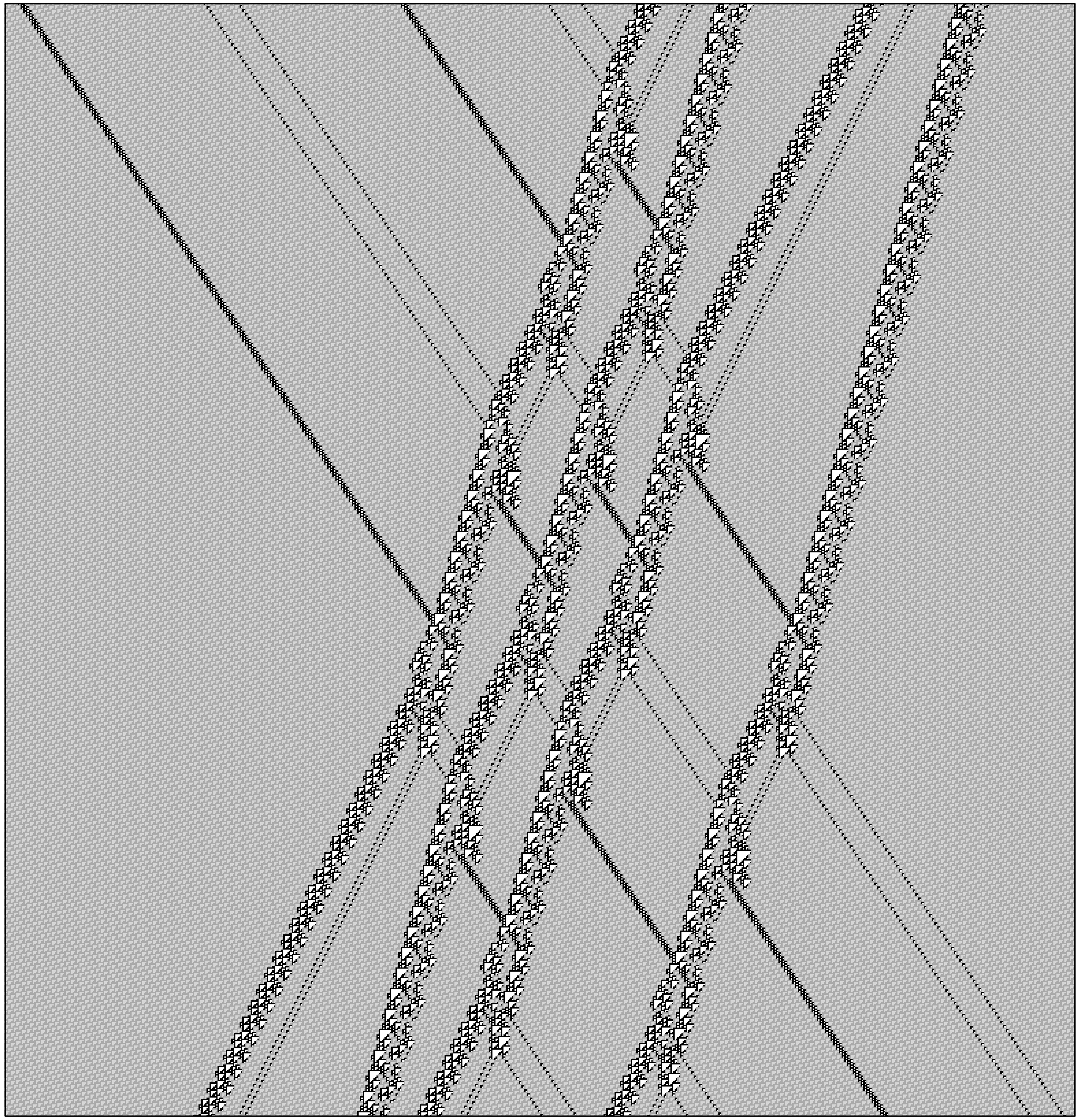
$$p_{A^4}^+ \leftrightarrow p_E^+, p_{\bar{E}}^+$$

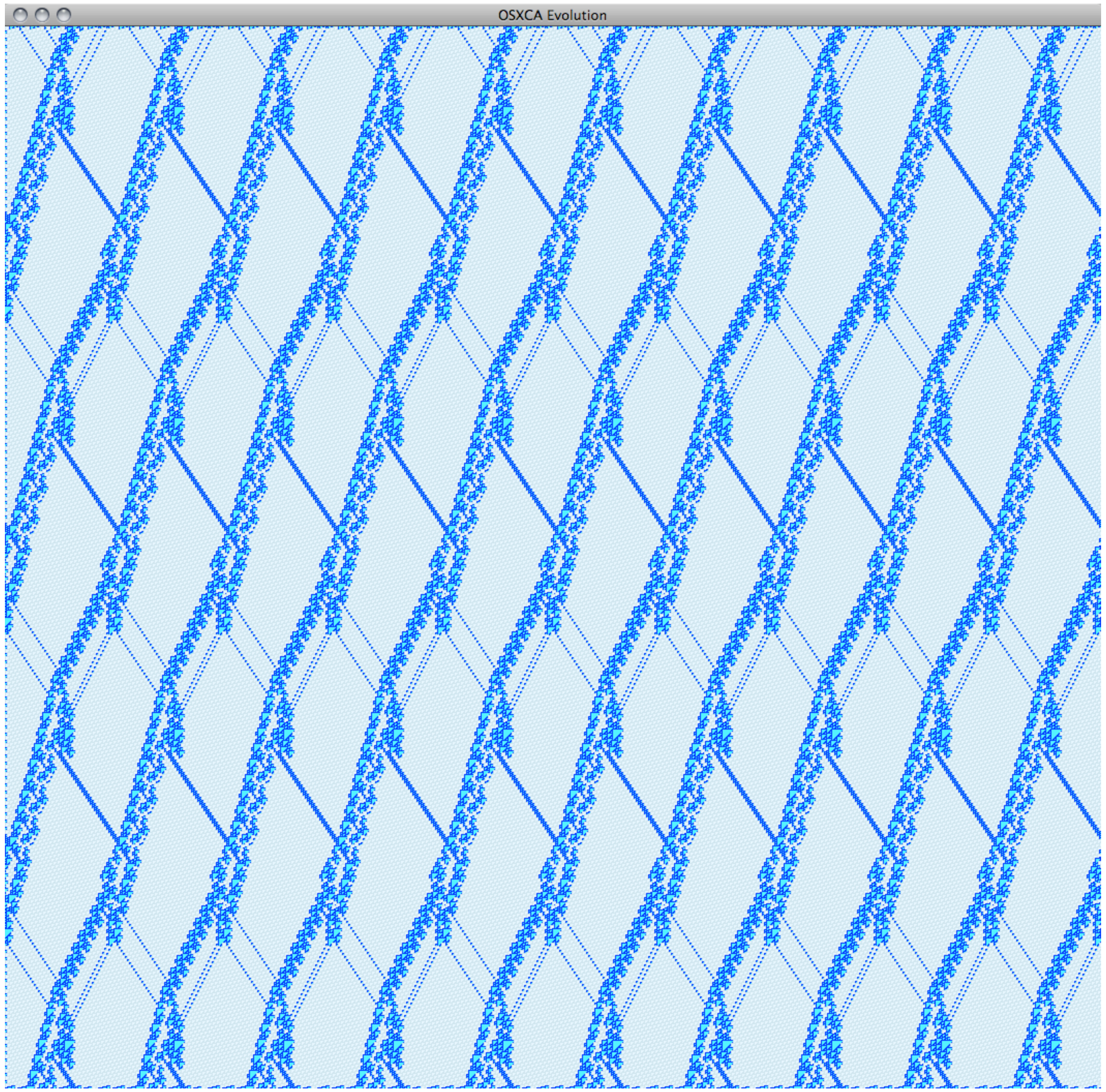
defining two beam routing connected by a transition of collisions as:

$$(p_A^+, p_A^+ \leftrightarrow p_{\bar{B}}^-, p_B^-, p_B^-) \rightarrow (p_{A^4}^+ \leftrightarrow p_E^+, p_{\bar{E}}^+), \text{ and}$$

$$(p_{A^4}^+ \leftrightarrow p_E^+, p_{\bar{E}}^+) \rightarrow (p_A^+, p_A^+ \leftrightarrow p_{\bar{B}}^-, p_B^-, p_B^-).$$

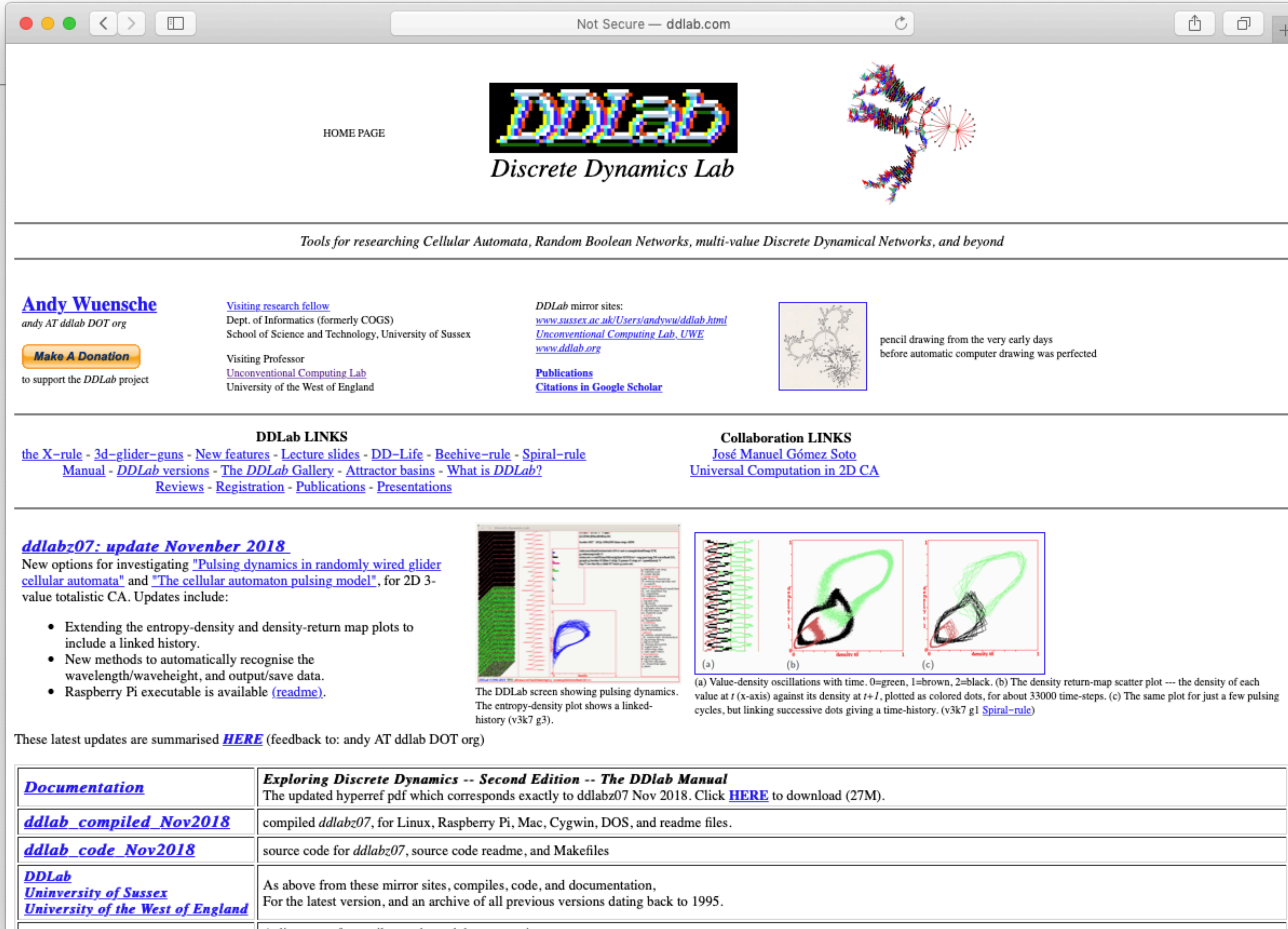
Symbol super colliders





$$(A(f_3\_1)\text{-}e\text{-}A(f_1\_1)\text{-}e\text{-}\overline{B}(C, f_1\_1)\text{-}e\text{-}2B(f_4\_1))^*$$

We use “Discrete Dynamics Lab” (DDLab). A free software created by Andrew Wuensche from 1992 to work with circular simulations. http://www.ddlab.com/



Not Secure — ddlab.com

HOME PAGE

DDLab

Discrete Dynamics Lab

Tools for researching Cellular Automata, Random Boolean Networks, multi-value Discrete Dynamical Networks, and beyond

Andy Wuensche
andy AT ddlab DOT org

Make A Donation
to support the *DDLab* project

Visiting research fellow
Dept. of Informatics (formerly COGS)
School of Science and Technology, University of Sussex

Visiting Professor
Unconventional Computing Lab
University of the West of England

DDLab mirror sites:
www.sussex.ac.uk/Users/andywu/ddlab.html
Unconventional Computing Lab, UWE
www.ddlab.org

Publications
Citations in Google Scholar

pencil drawing from the very early days
before automatic computer drawing was perfected

DDLab LINKS
the X-rule - 3d-glider-guns - New features - Lecture slides - DD-Life - Beehive-rule - Spiral-rule
Manual - *DDLab* versions - The *DDLab* Gallery - Attractor basins - What is *DDLab*?
Reviews - Registration - Publications - Presentations

Collaboration LINKS
José Manuel Gómez Soto
Universal Computation in 2D CA

ddlabz07: update Novenber 2018

New options for investigating "Pulsing dynamics in randomly wired glider cellular automata" and "The cellular automaton pulsing model", for 2D 3-value totalistic CA. Updates include:

- Extending the entropy-density and density-return map plots to include a linked history.
- New methods to automatically recognise the wavelength/waveheight, and output/save data.
- Raspberry Pi executable is available (readme).

The DDLab screen showing pulsing dynamics. The entropy-density plot shows a linked-history (v3k7 g3).

(a) Value-density oscillations with time. 0=green, 1=brown, 2=black. (b) The density return-map scatter plot --- the density of each value at *t* (x-axis) against its density at *t+1*, plotted as colored dots, for about 33000 time-steps. (c) The same plot for just a few pulsing cycles, but linking successive dots giving a time-history. (v3k7 g1 Spiral-rule)

These latest updates are summarised ***HERE*** (feedback to: andy AT ddlab DOT org)

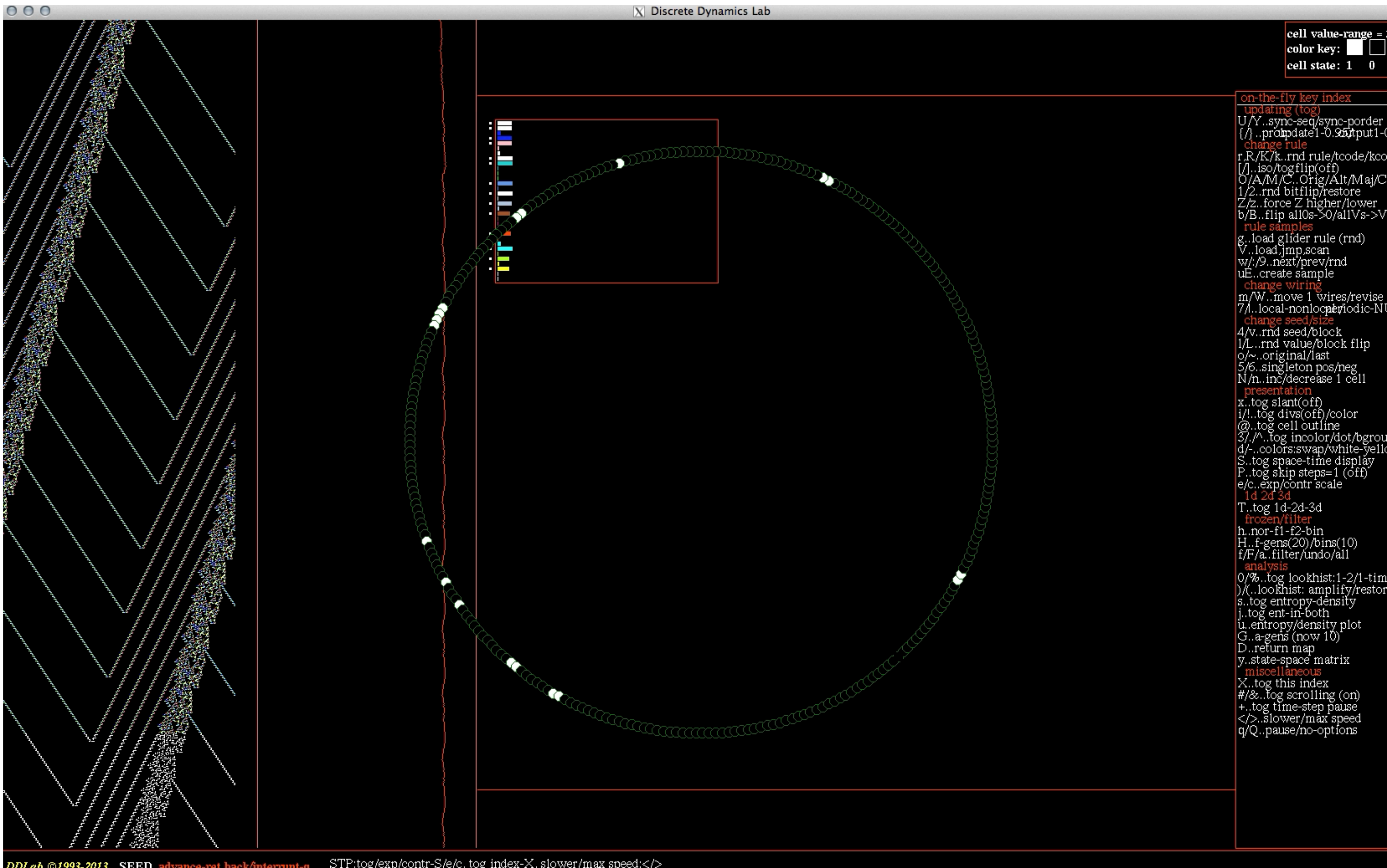
Documentation	***Exploring Discrete Dynamics -- Second Edition -- The DDlab Manual*** The updated hyperref pdf which corresponds exactly to ddlabz07 Nov 2018. Click HERE to download (27M).
ddlab_compiled_Nov2018	compiled *ddlabz07*, for Linux, Raspberry Pi, Mac, Cygwin, DOS, and readme files.
ddlab_code_Nov2018	source code for *ddlabz07*, source code readme, and Makefiles
DDLab ***Uninversity of Sussex*** ***University of the West of England***	As above from these mirror sites, compiles, code, and documentation, For the latest version, and an archive of all previous versions dating back to 1995.
	A directory of compiles, code, and documentation.

Wuensche, A. (2011) **Exploring Discrete Dynamics; The DDLab Manual**, Luniver Press, UK.

DDLab evolving cellular automata as rings (cyclotrons)

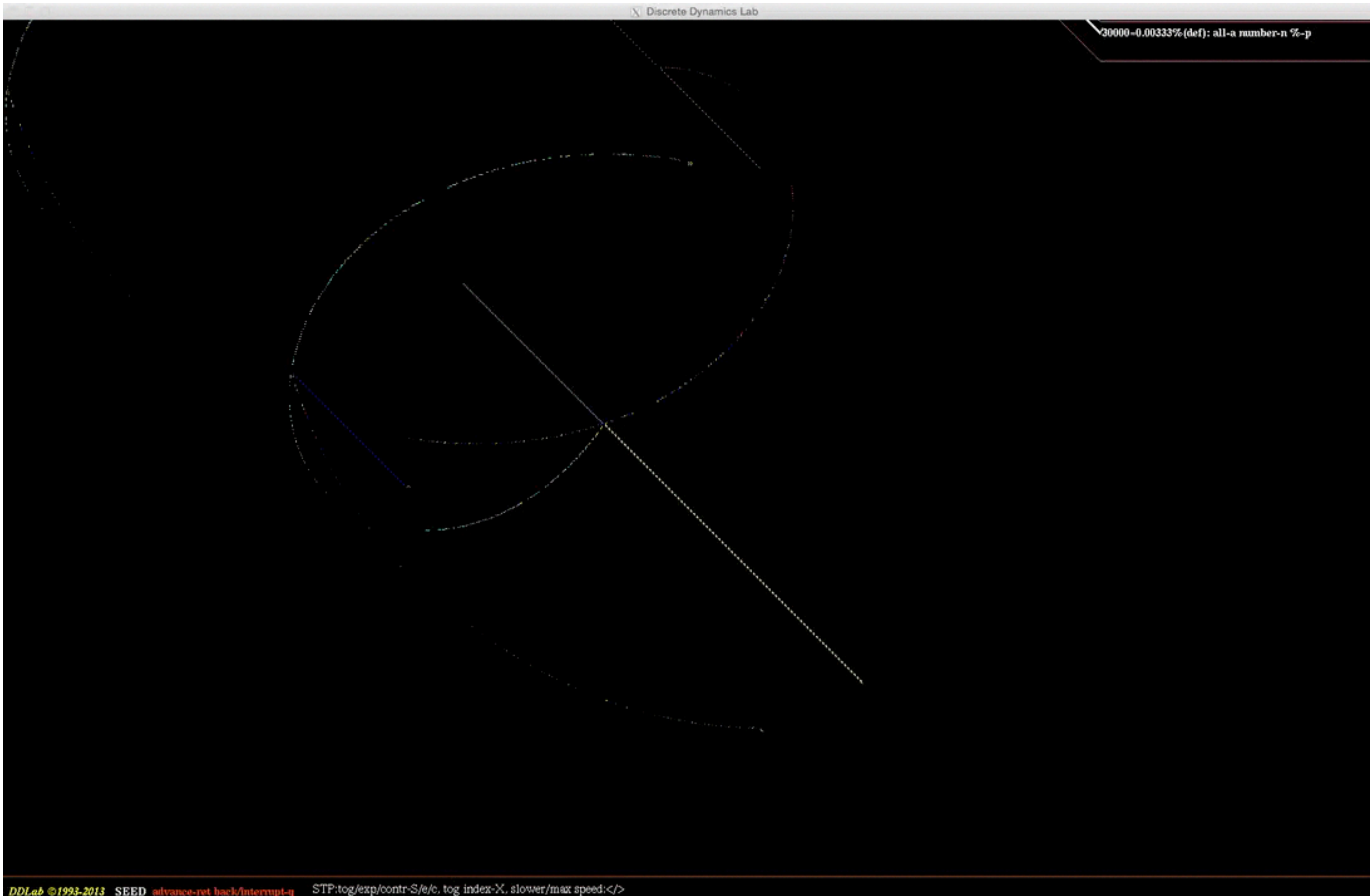
Cyclotrons are the first stage where we can see periodic collisions or simple dynamics of particles traveling around the ring.

A multiple collision between eight particles we can produce a glider gun in rule 110. In this simulation using DDLab, we can see the typical two dimensional representation a nd at the center the cyclotronic view, with the periodic background filtered.



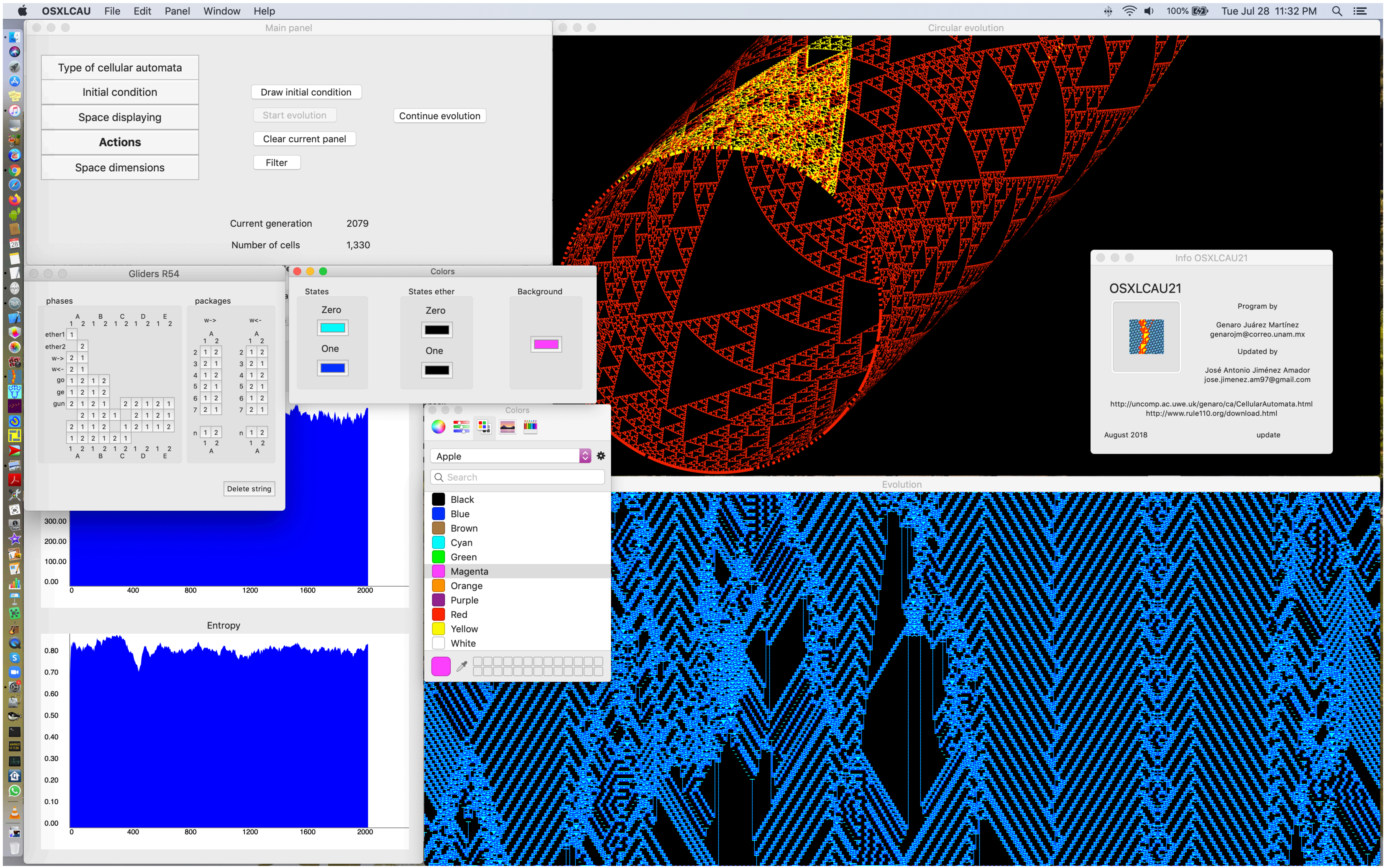
Implementing a counter of positive integers by particle collisions in rule 54

This collider calculate the sequence of positive integers counting the number of *T*s in a stationary particle at the center by the number of collisions rule 54



* Martinez, G.J., Adamatzky, A. & Stephens, C.R. (2011) **Cellular automaton supercolliders**, *International Journal of Modern Physics C* 22(4):419-439.
* Martinez, G.J., Adamatzky, A. & McIntosh, H. (2012) **Computing on rings.** In: *A computable Universe: Understanding and Exploring Nature as Computation*, H. Zenil (ed.), World Scientific Press, chapter 14, pages 283--302.

Cellular Automata Viewer (CAViewer) is a free software for mac where you can simulate circular evolutions (developed in LCCOMP, ALIROB, UCL; Labs)
https://www.comunidad.escom.ipn.mx/genaro/Cellular_Automata_Repository/Software.html

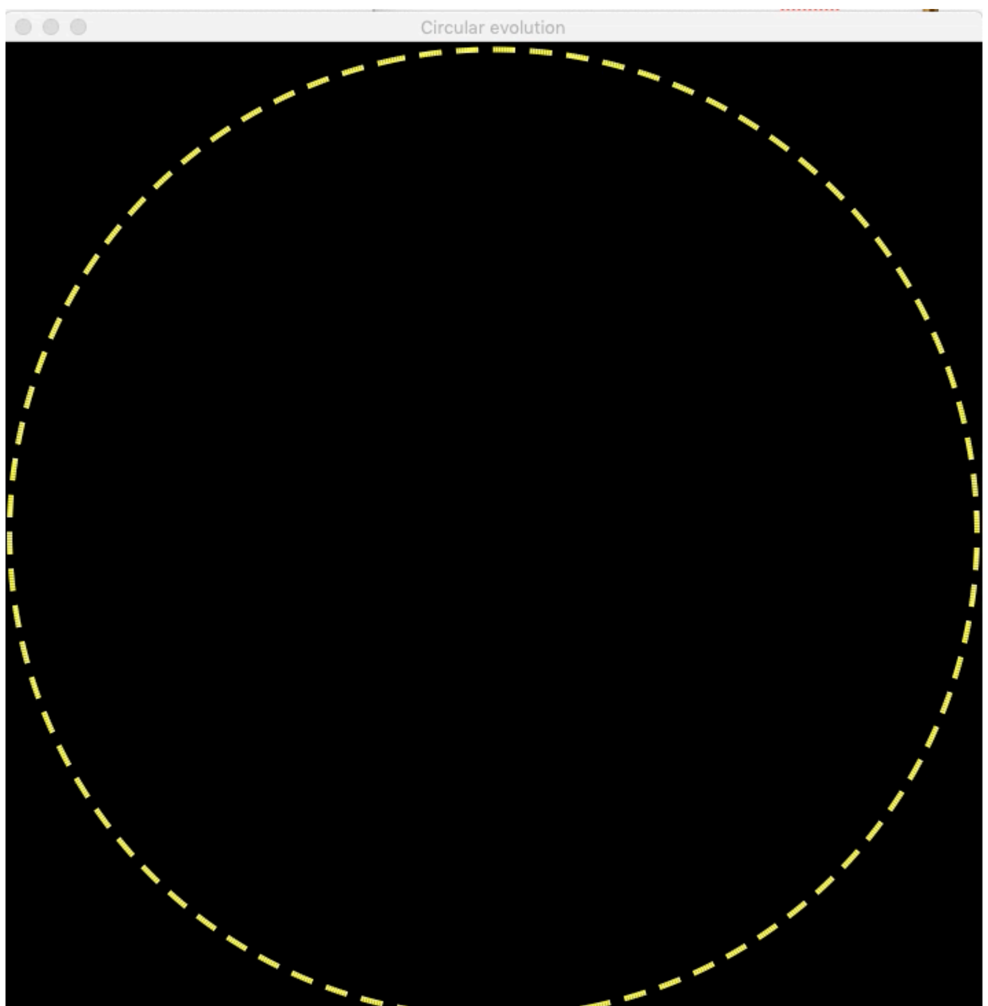


Construction of patterns by synchronization of multiple collisions in rule 54

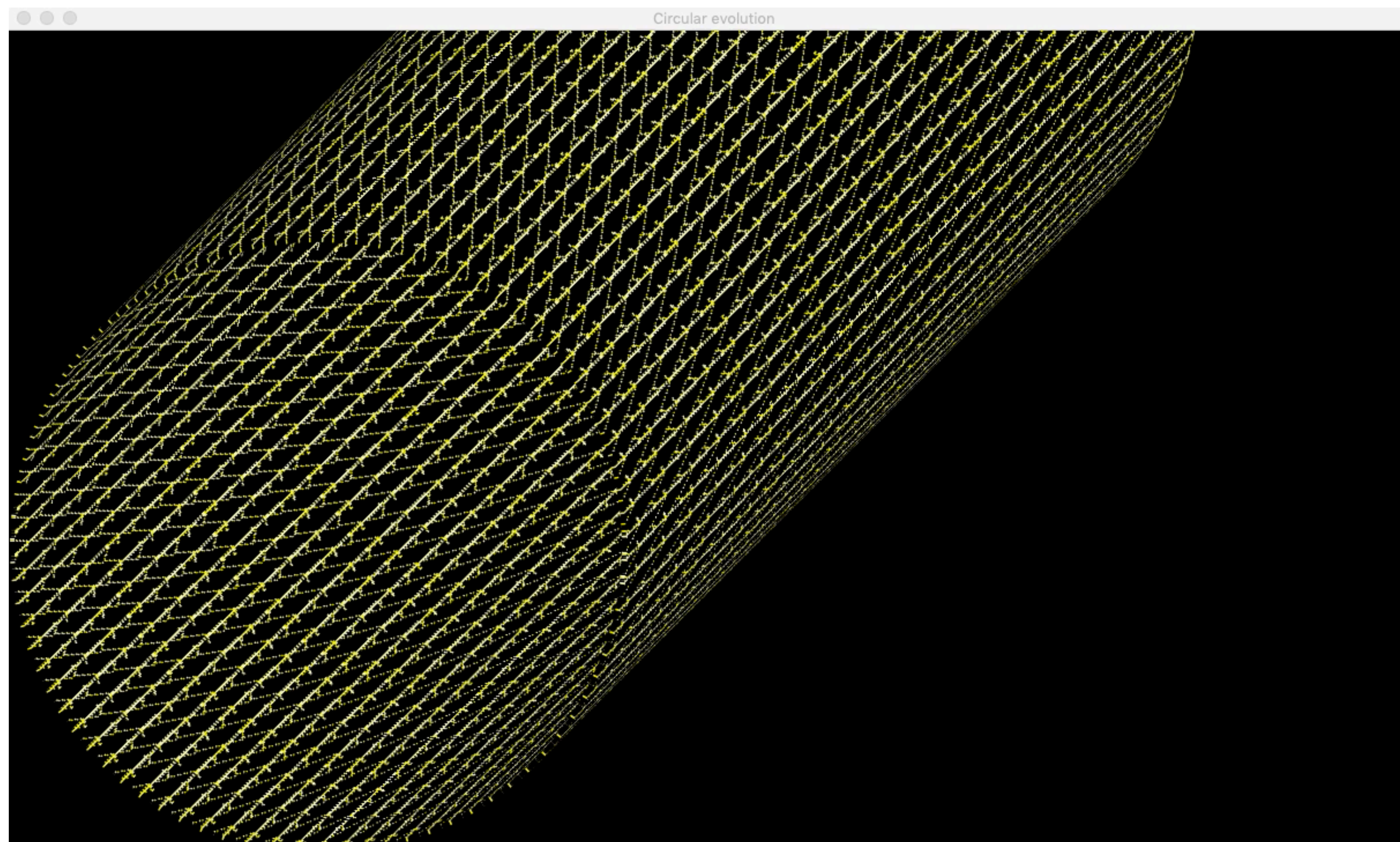
Using a cyclotron we can design patterns with other views. This simulation starts with an initial condition of 3,214 cells codifying 216 particles in rule 54. The reaction is a triple collision controller by two basic interactions:

$$f(g_e, \overleftarrow{w}) \rightarrow \overleftarrow{w}, g_e, 2\overrightarrow{w} \;\; ; \;\; f(2\overrightarrow{w}, \overleftarrow{w}) \rightarrow \overleftarrow{w}$$

The reaction starts with a negative particle *w* colliding versus a stationary particle *g* yielding a negative particle plus two positive particles *w*. But the first negative particle finds these new pairs of positive particles and annihilate them. Therefore, we can codify these particles with the next expression: $(\overleftarrow{w}\text{-}2e\text{-}g_e(\mathrm{A},\mathrm{f}_1)\text{-}2e\text{-}\overleftarrow{w})^*$.

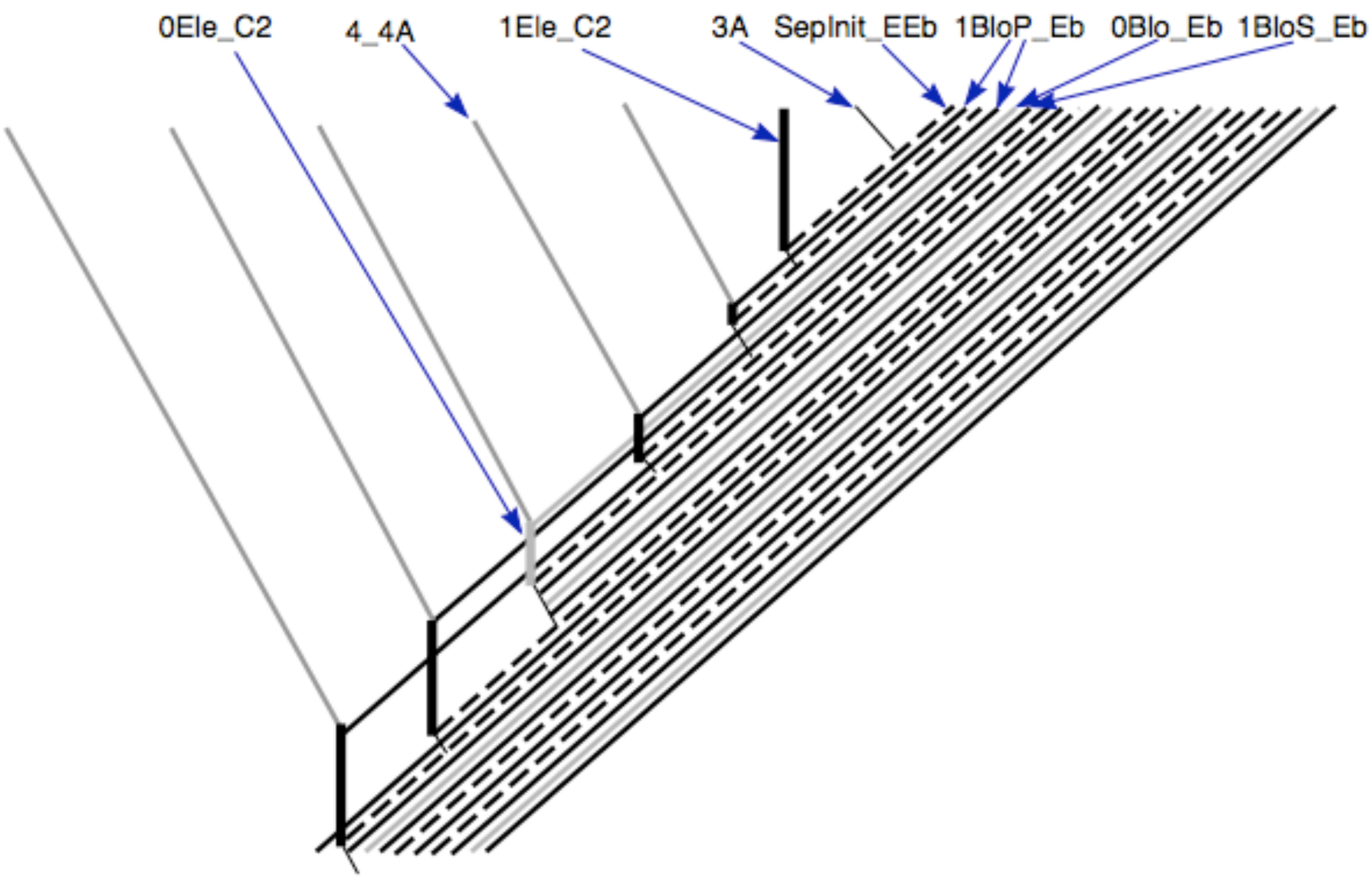


circular

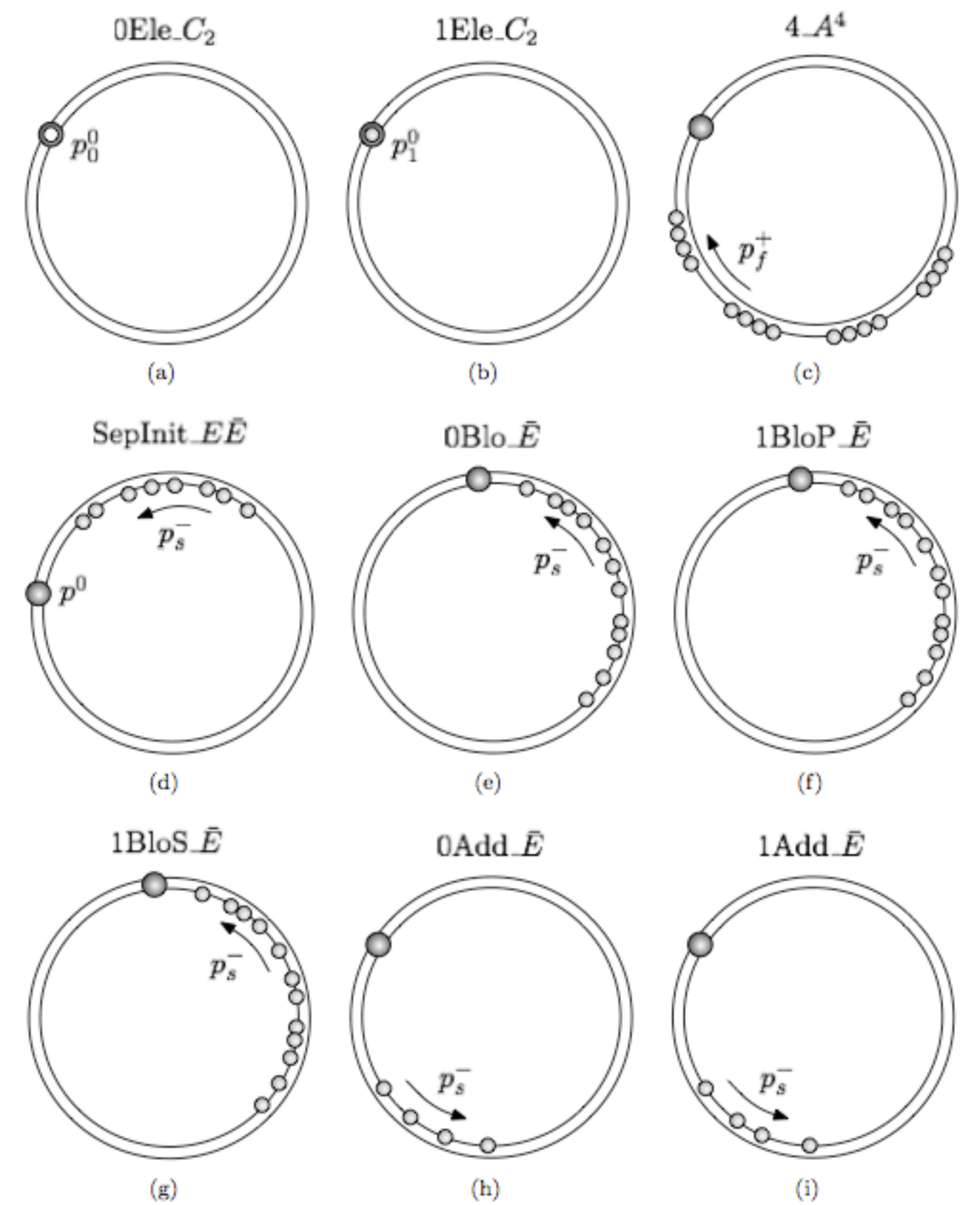


three-dimensional projection

Cyclic tag systems in a finite codification



A diagram of a cyclic tag system (CTS) working in rule 110



Beam routing codification representing package of particles which reproduces a CTS in rule 110

- Cook, M. (2004) **Universality in Elementary Cellular Automata**. Complex Systems 15(1), 1-40.
- Cook, M. (2008) **A Concrete View of Rule 110 Computation**. In: The Complexity of Simple Programs, T. Neary, D.Woods, A.K. Seda and N. Murphy (Eds.), 31-55.
- Wolfram, S. (2002) **A New Kind of Science**, Wolfram Media, Inc., Champaign, Illinois.
- Neary, T. & Woods, D. (2006) **P-completeness of cellular automaton Rule 110**. Lecture Notes in Computer Science 4051, 132-143.
- Martínez, G.J., McIntosh, H.V., Mora, J.C.S.T. & Vergara, S.V.C. (2011) **Reproducing the cyclic tag system developed by Matthew Cook with Rule 110 using the phases f1_1**, *Journal of Cellular Automata 6(2-3)*, 121-161.

Cyclic tag systems in a finite codification as a collider

This way, the cyclic tag system working in rule 110 can be simplified as follows:

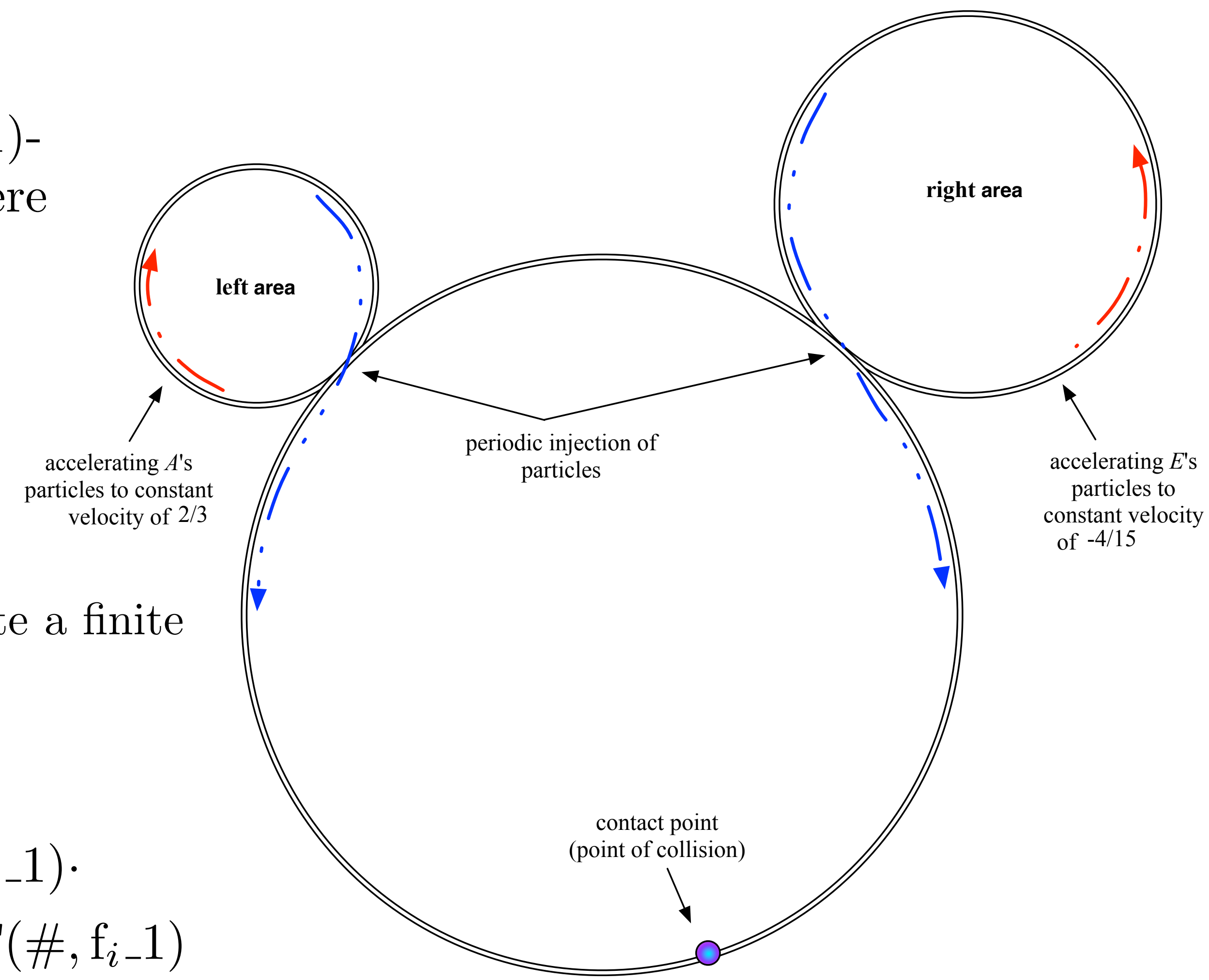
left: $\{649e\text{-}4\_A^4(\text{F}\_i)\}^*$, for $1 \leq i \leq 3$ in sequential order

center: $246e\text{-}1\text{Ele}\_\text{C}2(\text{A},\text{f}_1\_1)\text{-}e\text{-}A^3(\text{f}_1\_1)$

right: $\{\text{SepInit}\_E\bar{E}(\#,\text{f}_i\_1)\text{-}1\text{BloP}\_\bar{E}(\#,\text{f}_i\_1)\text{-}\text{SepInit}\_E\bar{E}(\#,\text{f}_i\_1)\text{-}1\text{BloP}\_\bar{E}(\#,\text{f}_i\_1)\text{-}0\text{Blo}\_\bar{E}(\#,\text{f}_i\_1)\text{-}1\text{BloS}\_\bar{E}(\#,\text{f}_i\_1)\}^*$ (where $1 \leq i \leq 4$ and $\#$ represents a particular phase).

This way, we have that the string $w_{CTSR110}$ is a word able to simulate a finite state machine into a cellular automata collider.

$$w_{CTSR110} = (649e \cdot 4\_A^4(\text{F}\_i))^* \cdot (246e \cdot 1\text{Ele}\_C_2(\text{A},\text{f}_1\_1) \cdot e \cdot A^3(\text{f}_1\_1)) \cdot (\text{SepInit}\_E\bar{E}(\#,\text{f}_i\_1) \cdot 1\text{BloP}\_\bar{E}(\#,\text{f}_i\_1) \cdot \text{SepInit}\_E\bar{E}(\#,\text{f}_i\_1) \cdot 1\text{BloP}\_\bar{E}(\#,\text{f}_i\_1) \cdot 0\text{Blo}\_\bar{E}(\#,\text{f}_i\_1) \cdot 1\text{BloS}\_\bar{E}(\#,\text{f}_i\_1))^*.$$



A diagram of a cyclic tag system (CTS) working in rule 110

Genaro J. Martínez, Andrew Adamatzky, Harold V. McIntosh (2017) **A Computation in a Cellular Automaton Collider Rule 110**, In: *Advances in Unconventional Computing: Volume I Theory*, A. Adamatzky (Ed). chapter 15, pages 391-428.

A Computation in a Cellular Automaton Collider Rule 110

Video available in: https://youtu.be/i5af0tQiVd4

Final remarks

Complex ECA rules with different capacities explored with cyclotrons.

rule	class	particle	particle^n	slopes	gun	gun^n	soliton	complex with memory	fractals
41	4	yes	no	+	no	no	no	yes	no
54	4	yes	no	-,+,s	yes	yes	yes	yes	no
106	4	yes	no	-	no	no	no	yes	no
110	4	yes	yes	-,+,s	yes	yes	yes	yes	no
22	3	yes	no	-,+	no	no	no	yes	yes
126	3	yes	no	-,+,s	no	no	no	yes	yes
26	2	yes	no	-,+	no	no	yes	yes	yes
62	2	yes	no	-,+	no	no	no	yes	no

- Martínez, G.J., Adamatzky, A., Hoffmann, R., Désérable, D. & Zelinka, I. (2019) **On Patterns and Dynamics of Rule 22 Cellular Automaton**. *Complex Systems* 28(2), 125-174.
- Martínez, G.J., Adamatzky, A. & Alonso-Sanz, R. (2013) **Designing Complex Dynamics in Cellular Automata with Memory**. *International Journal of Bifurcation and Chaos* 23(10), 1330035-131.

THE END

THANK YOU FOR YOUR KIND ATTENTION

Cellular Automata Repository
https://www.comunidad.escom.ipn.mx/genaro/CA_repository.html

Complex Cellular Automata Repository
https://www.comunidad.escom.ipn.mx/genaro/Complex_CA_repository.html

Cellular Automata Software
https://www.comunidad.escom.ipn.mx/genaro/Cellular_Automata_Repository/Software.html