NKS2004
wolframscience.com

Departamento de Aplicación de Microcomputadoras
Instituto de Ciencias
Universidad Autónoma de Puebla

A way to construct complex configurations in Rule 110

Genaro Juárez Martínez
genarojm@correo.unam.mx

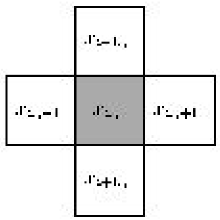
BOSTON, MA
APRIL 2004

History in cellular automata theory

John von Neumann (precursor)

$$\Sigma = \{0, \ldots, 28\}$$

neighborhood von Neumann

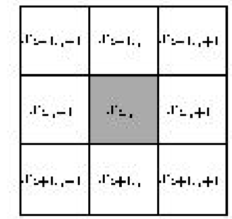


$$c_i, \; x_{i,j} \in Z \otimes Z$$

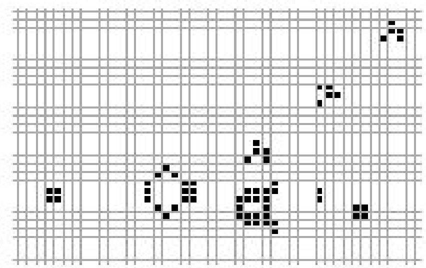
John Horton Conway (gliders system, collisions)

$$\Sigma = \{0, 1\}$$

neighborhood Moore



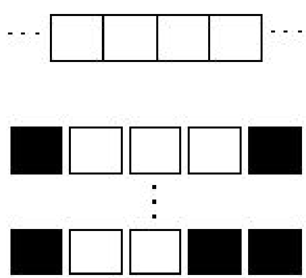
The Game of Life



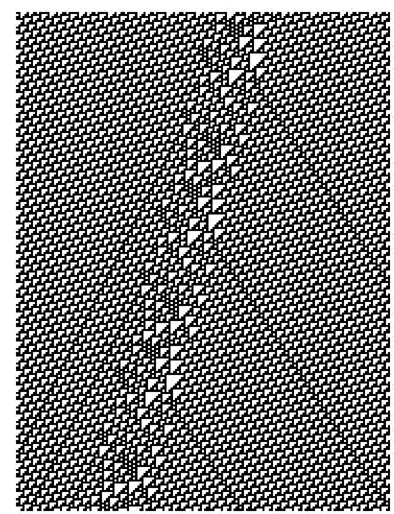
$$c_i, \; x_{i,j} \in Z \otimes Z$$

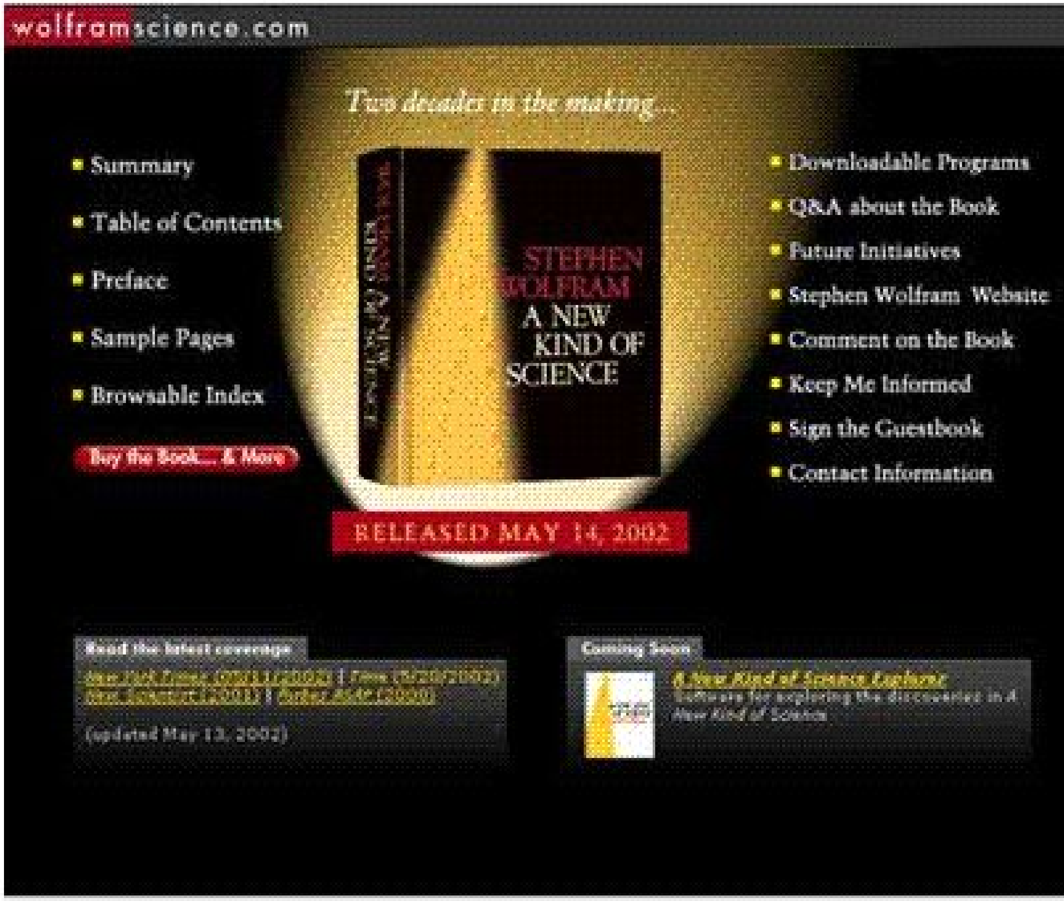
Stephen Wolfram (one dimension)

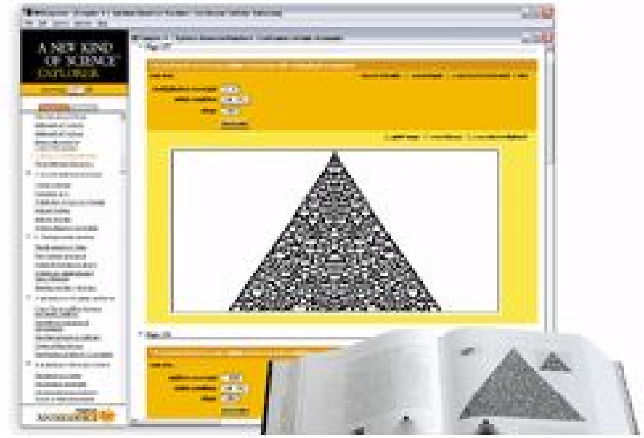
notation (*k*,*r*) and classes

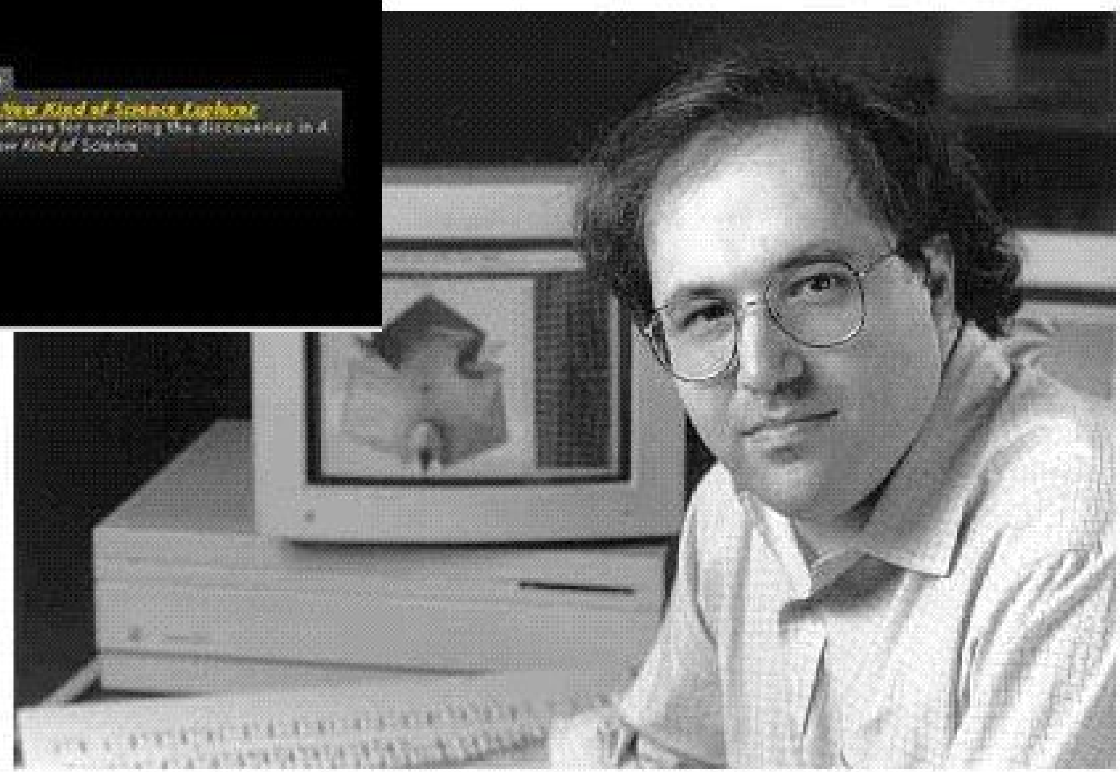


Matthew Cook (universality in Rule 110)







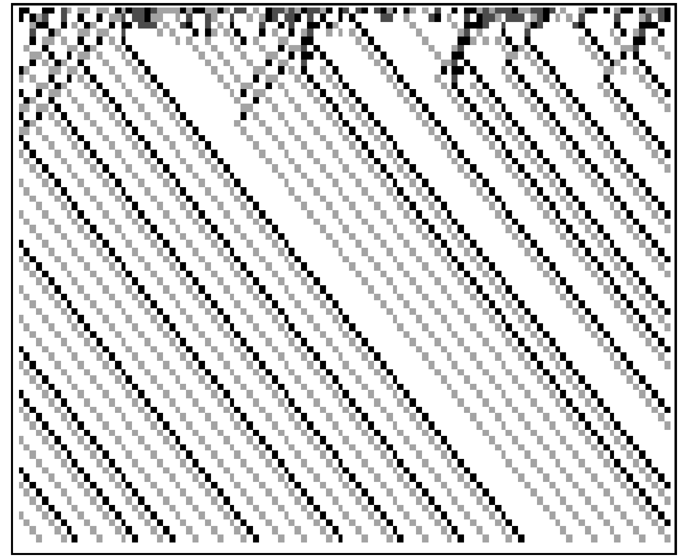


Stephen Wolfram

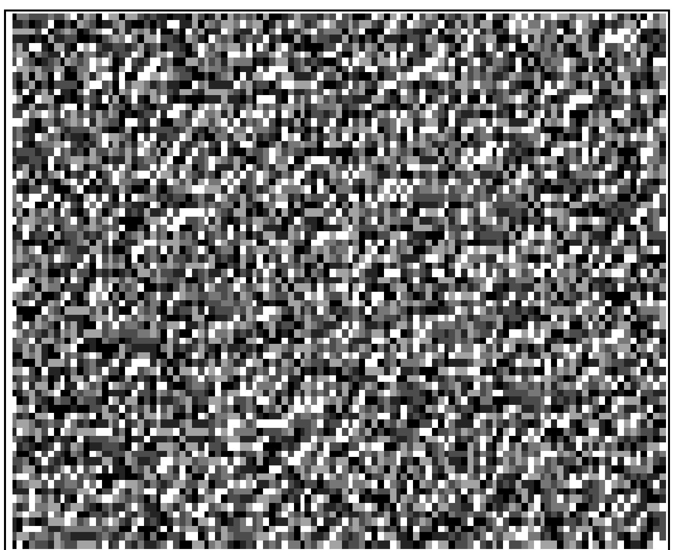
Wolfram's Classes

Class I - Uniform

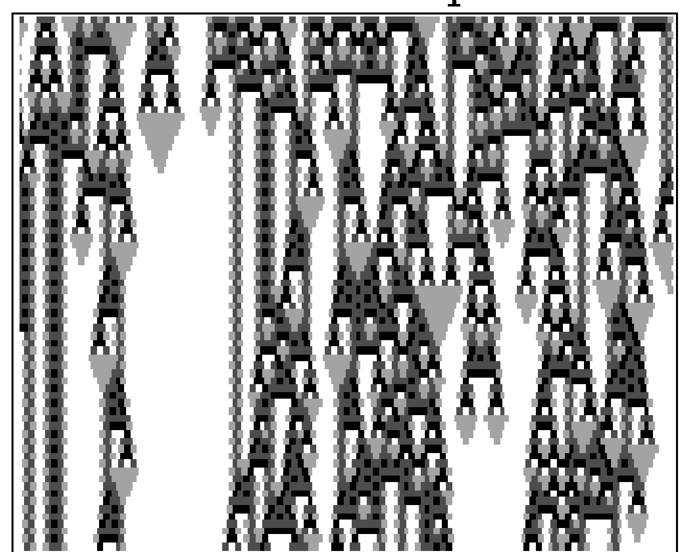
Class II - Periodic



Class III - Chaotic



Class IV - Complex

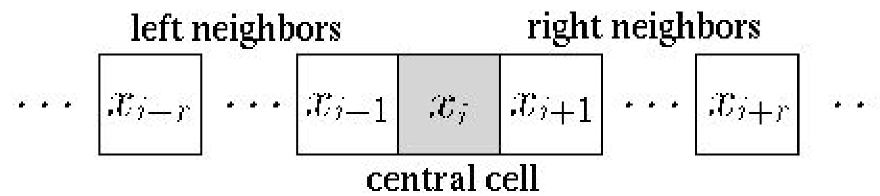


One-dimension cellular automata

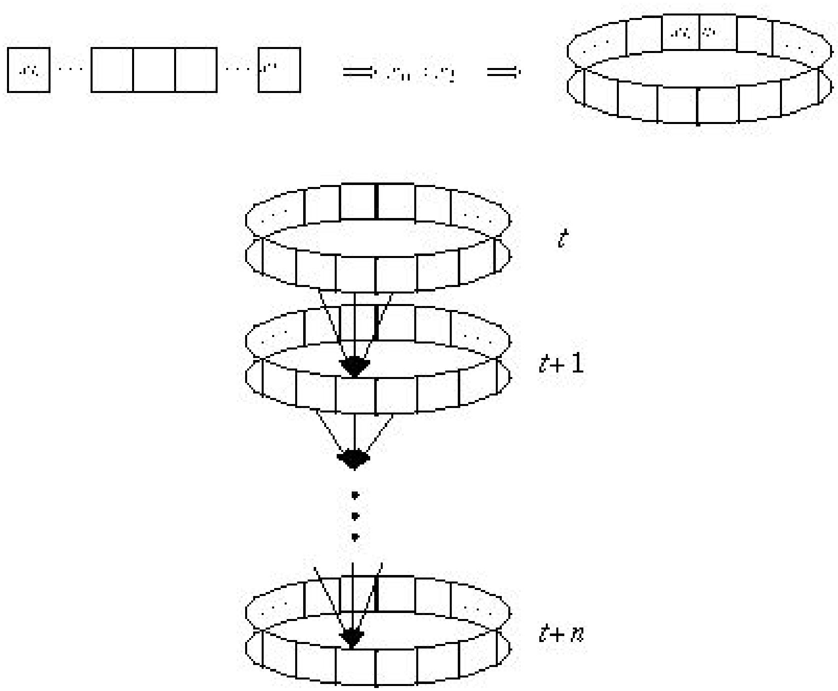
CA of order *(k,r)*

$$\{\Sigma,\ r,\ \varphi,\ c_i\}$$

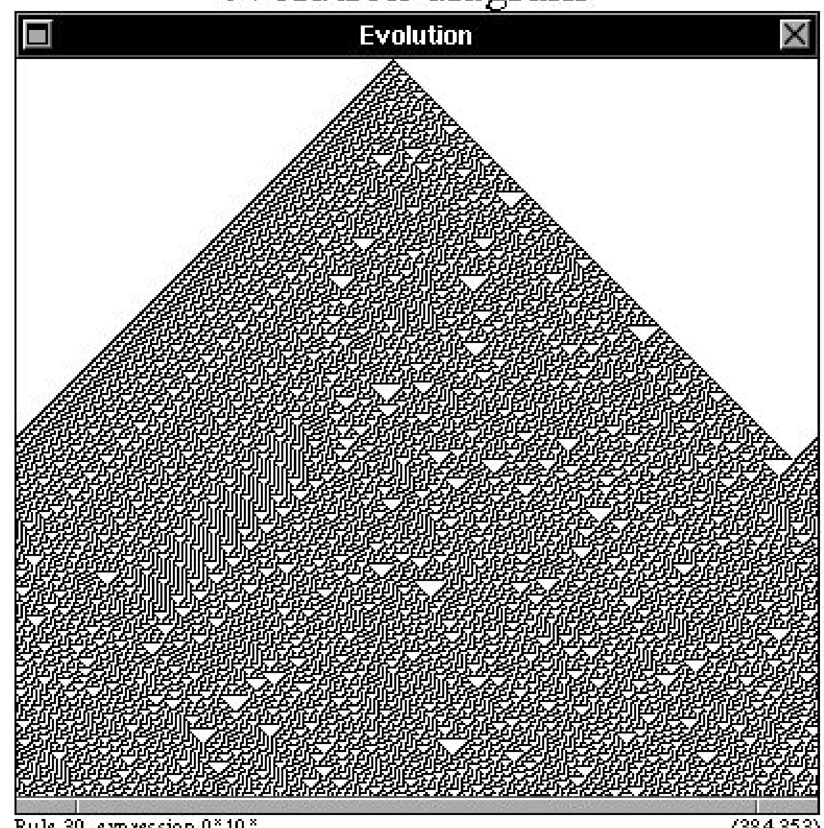
neighborhood



configurations



evolution diagram



Matthew Cook

http://vigeland.paradise.caltech.edu/~cook/

In November of 1998 at the Santa Fe Institute
Matthew Cook demostrates that Rule 110 is Universal!
Simulating a cyclic tag system

[†] Matthew Cook, "Universality in Elementary Cellular Automata," *Complex Systems*, Volume 15, Number 1, pp. 1-40, 2004.

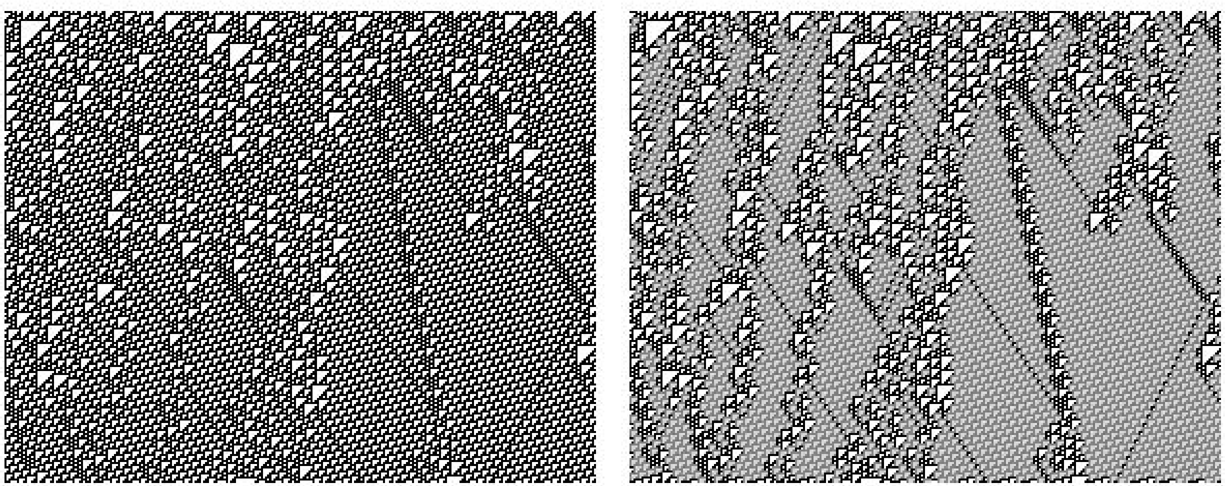
[‡] Stephen Wolfram, *A New Kind of Science*, Wolfram Media, Inc., Champaign, Illinois, 2002. (ISBN 1-57955-008-8)

Rule 110

Rule 110 is a binary cellular automaton that evolves in one dimension. The transition function is defined by the evolution rule of the following way:

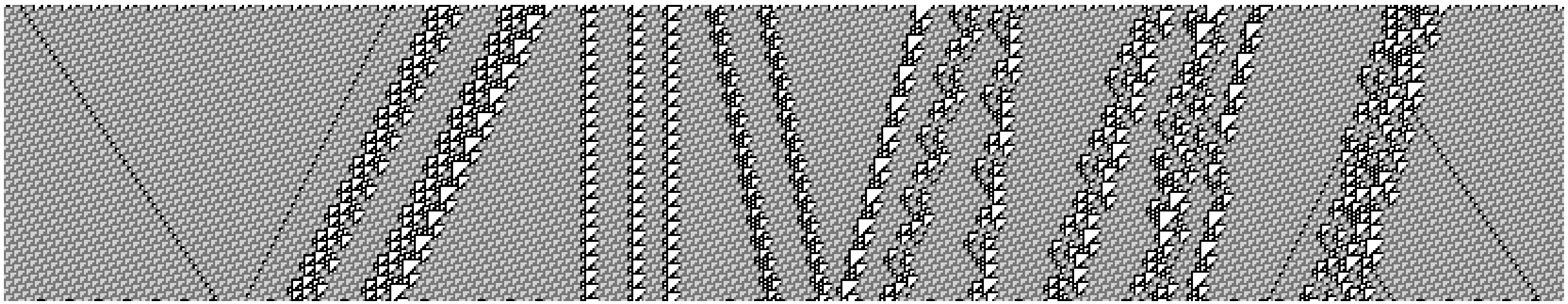
Rule 110	
$\varphi(0,0,0) \rightarrow 0$	$\varphi(1,0,0) \rightarrow 0$
$\varphi(0,0,1) \rightarrow 1$	$\varphi(1,0,1) \rightarrow 1$
$\varphi(0,1,0) \rightarrow 1$	$\varphi(1,1,0) \rightarrow 1$
$\varphi(0,1,1) \rightarrow 1$	$\varphi(1,1,1) \rightarrow 0$

Table 1: Evolution rule 110 - 01101110_2



Gliders in Rule 110

Clasification of gliders[n]



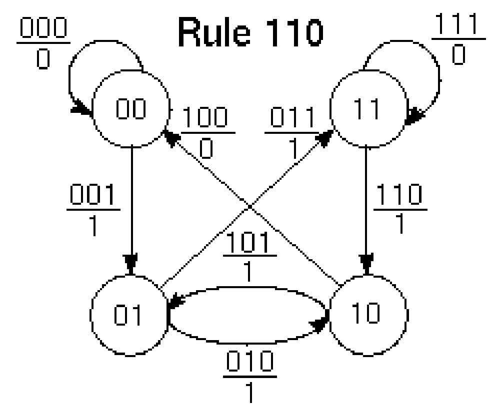
A B $B\text{-}^{n}$ $B\hat{}^{n}$ C_1 C_2 C_3 D_1 D_2 E^n $E\text{-}$ F G^n H gun

[n] Matthew Cook, "Introduction to the activity of rule 110" (copyright 1994-1998 Matthew Cook), http://w3.datanet.hu/~cook/Workshop/CellAut/ Elementary/Rule110/110pics.html, January 1999.

[†] Genaro Juárez Martínez, Harold V. McIntosh and Juan Carlos Seck Tuoh Mora, "Gliders in Rule 110," submitted to *International Journal of Unconventional Computing*, July 2004.

de Bruijn diagram

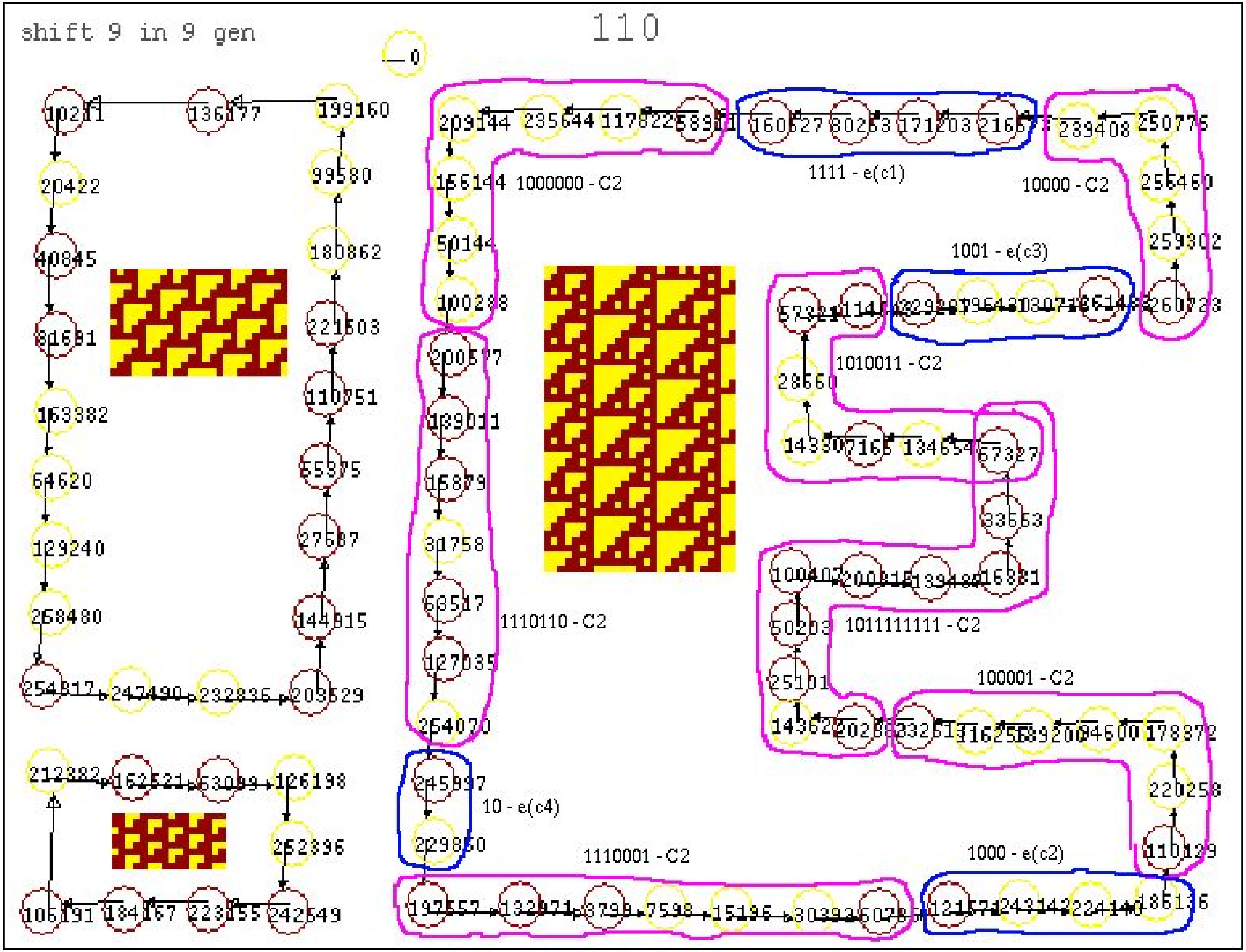
De Bruijn diagrams are relevant because they provide a graph presentation for the evolution rule of a one-dimensional cellular automaton, in a de Bruijn diagram a given path represents a sequence of states produced by the evolution of the nodes in the path describe the ancestor of the sequence. Thus the analysis of ancestors may be realized studying the paths of the de Bruijn diagram.[9]



A given sequence of states may have several, one or no ancestor, hence this sequence has analogous several, one or no path representing it in the de Bruijn diagram. Therefore if we want to know if a given sequence is a Garden of Eden one, we have to review if there is no path labeled with this sequence in the corresponding de Bruijn diagram.

[9] Harold V. McIntosh, "Linear cellular automata via de Bruijn diagrams," http://delta.cs.cinvestav.mx/~mcintosh/oldweb/pautomata.html, 1991.

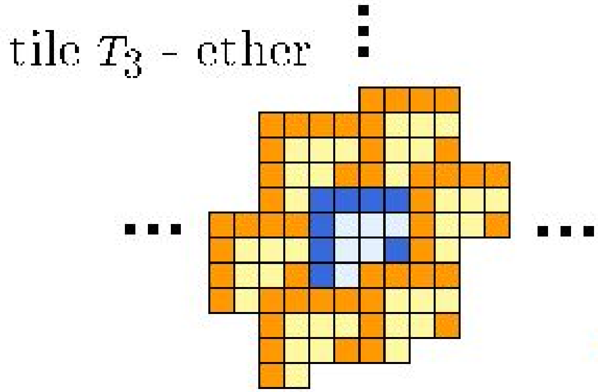
extended de Bruijn diagram



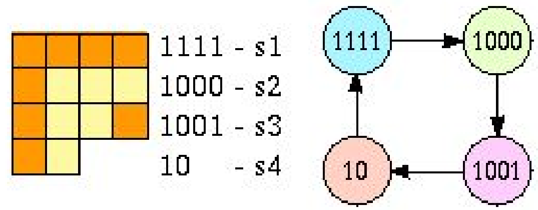
262,144 vertex

NXLCAU21

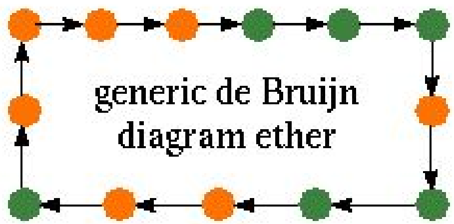
Phases in Rule 110



determining phases



1111-1000-1001-10 (f 1)
1000-1001-10-1111 (f 2)
1001-10-1111-1000 (f 3)
10-1111-1000-1001 (f 4)

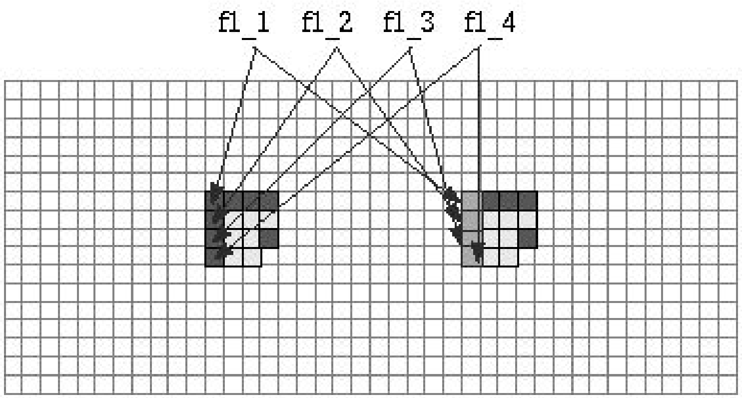


11111000100110 - e(f1_1)

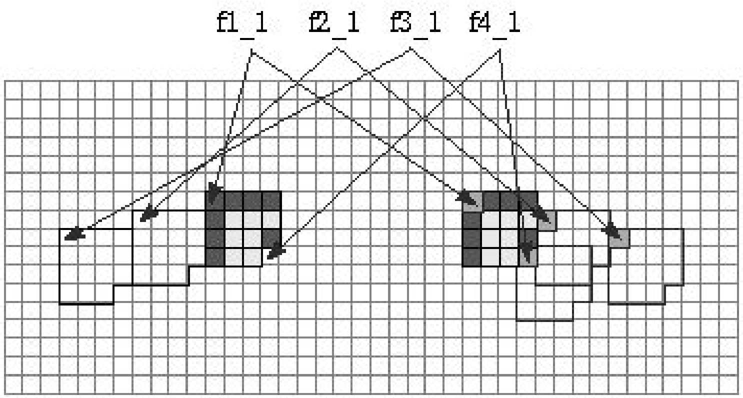
(11111000100110)*

e(f1_1)*

phases f_1_i (firts level)

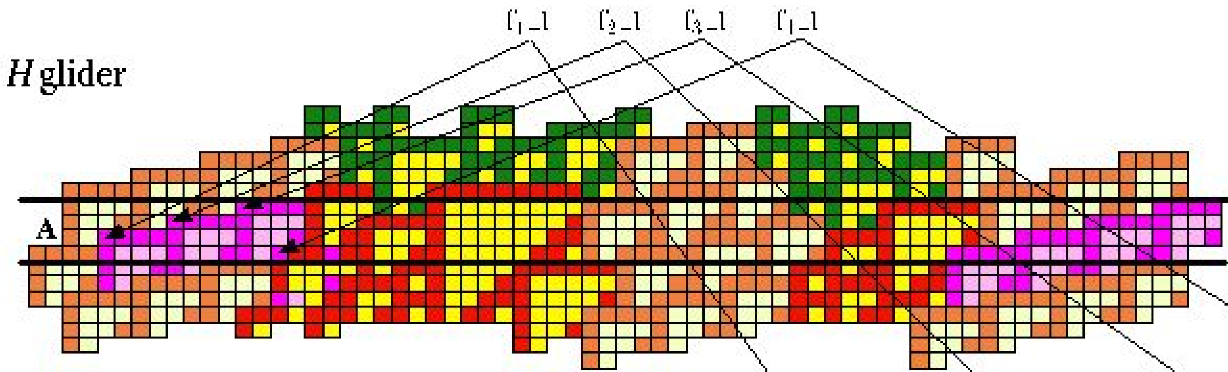


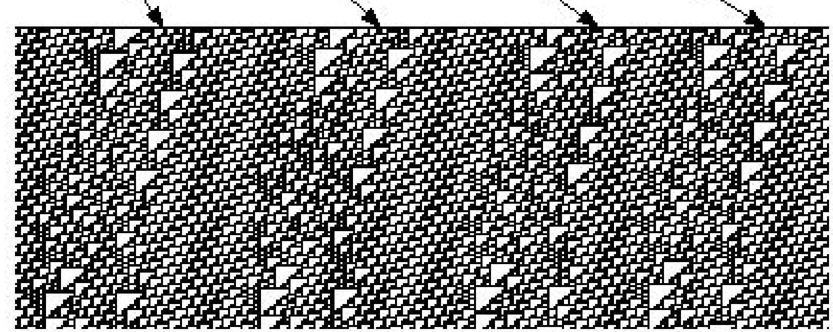
phases f_i_1 (second level)



where $1 \leq i \leq 4$

Phases in Rule 110





e*-H(A,f$_1$_1)-e-H(A,f$_2$_1)-e-H(A,f$_3$_1)-e-H(A,f$_4$_1)-e*

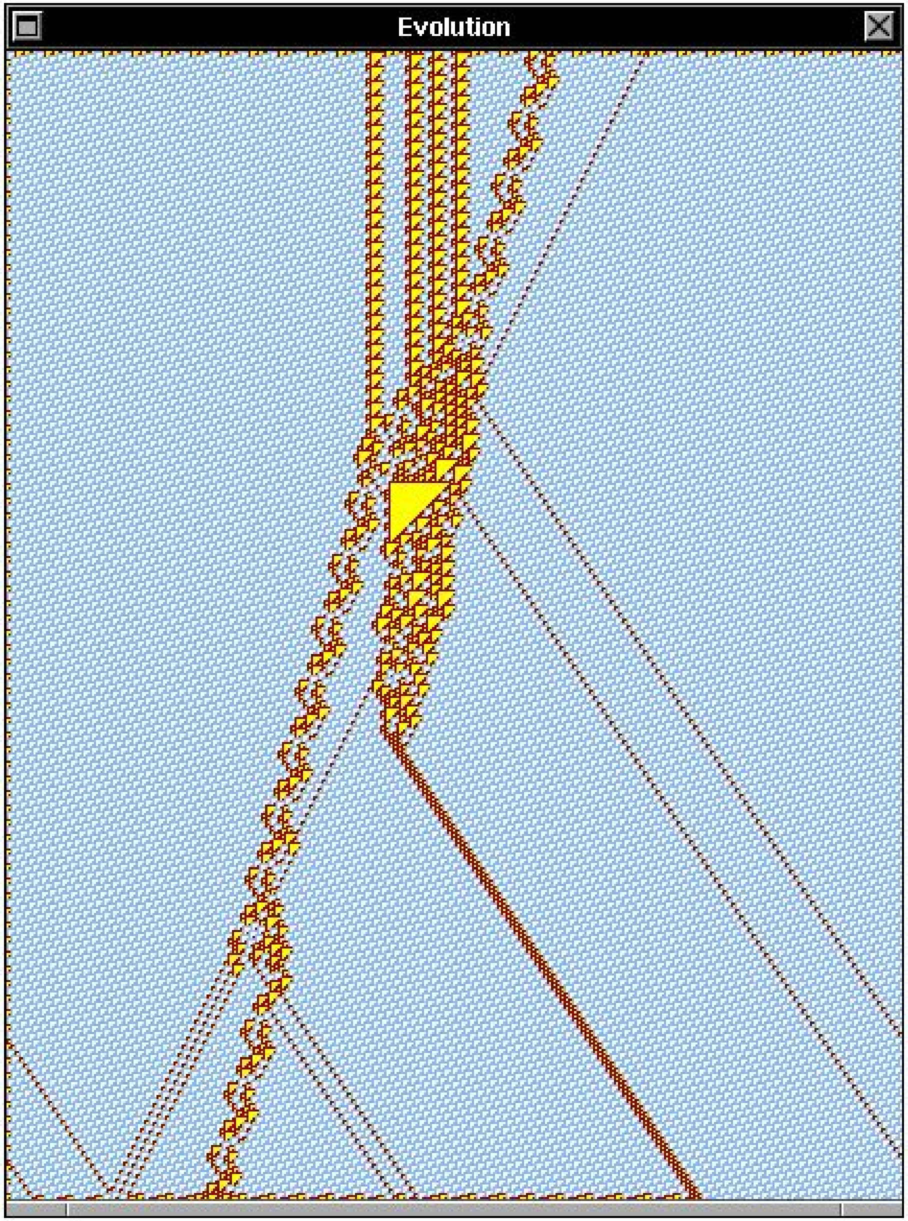
The phases f$_i$_1 for the case H(A,f$_1$_1) are:

sequence	cells	phase
11111000010110000000001111100010001101001111111000100110	53	f$_1$_1
11111000010011111100000000110001000110111110110000000100110	53	f$_2$_1
1111100001001101100010000000111001101111110001111100000110	53	f$_3$_1
111111001110000011010111110001001110010000	39	f$_4$_1

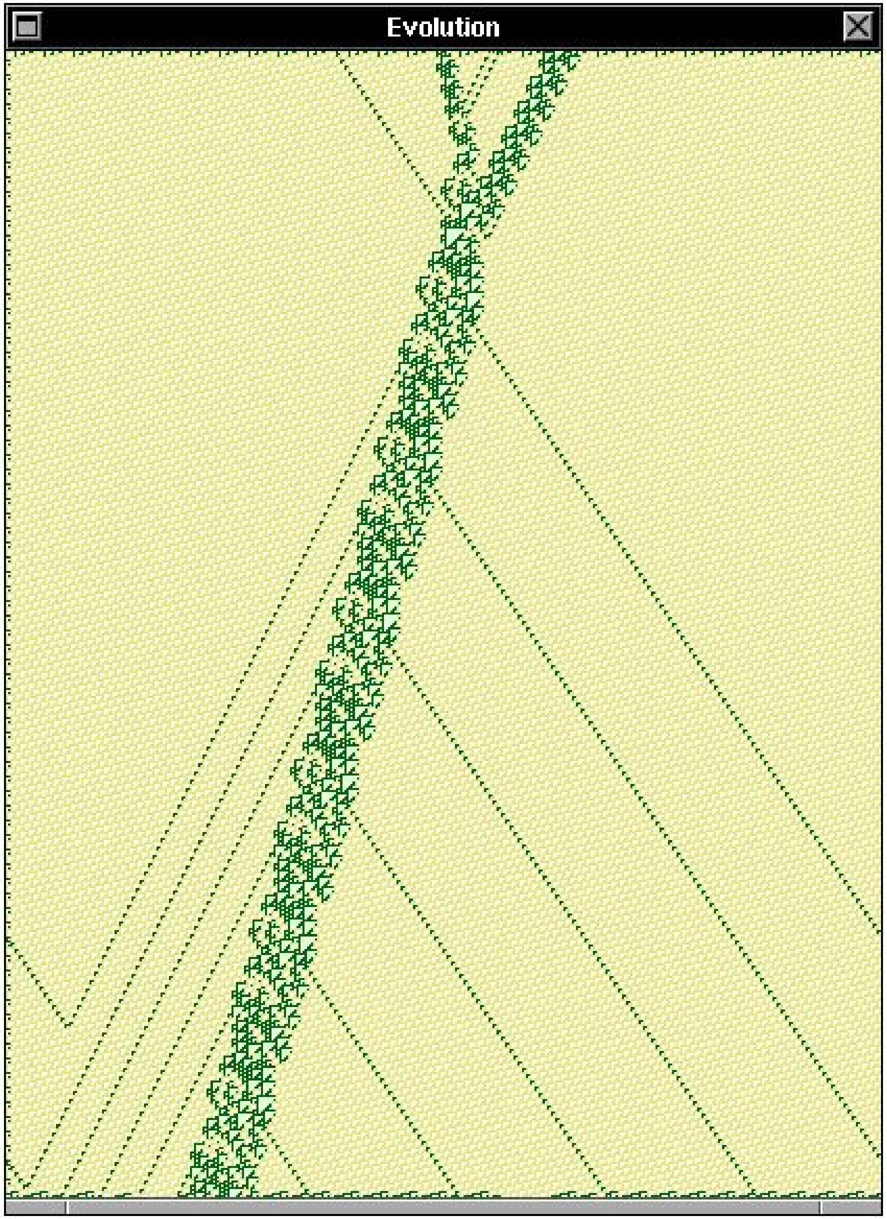
Table 1: Sequences to form glider H in the phases (A,f$_i$_1)

this way the other phases are obtained f$_i$_i for the 24 remaining phases, glider H has 25 phases the have 100 phases f$_i$_1 and 400 phases f$_i$_i.

Producing tile T_{28}



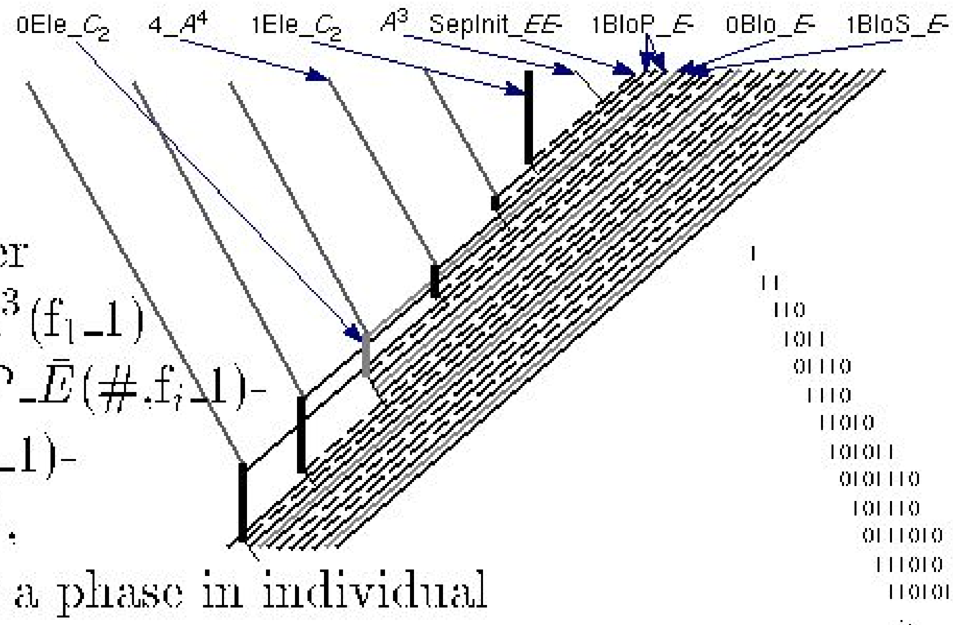
Producing glider gun



Cyclic tag system in phases

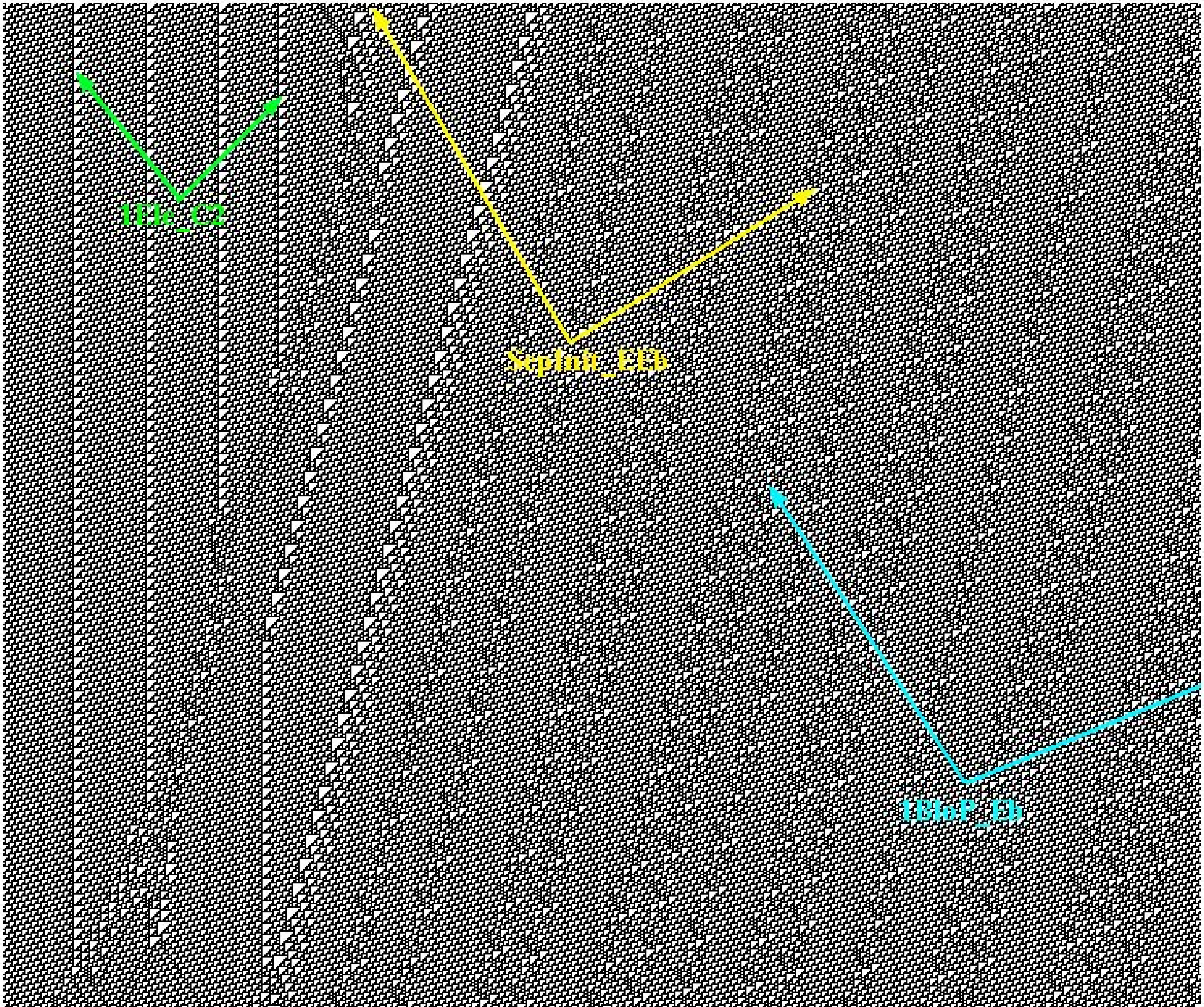
Universality in Rule 110 is demonstrated simulating a cyclic tag system. The last reduction obtained in cellular automata after Life.

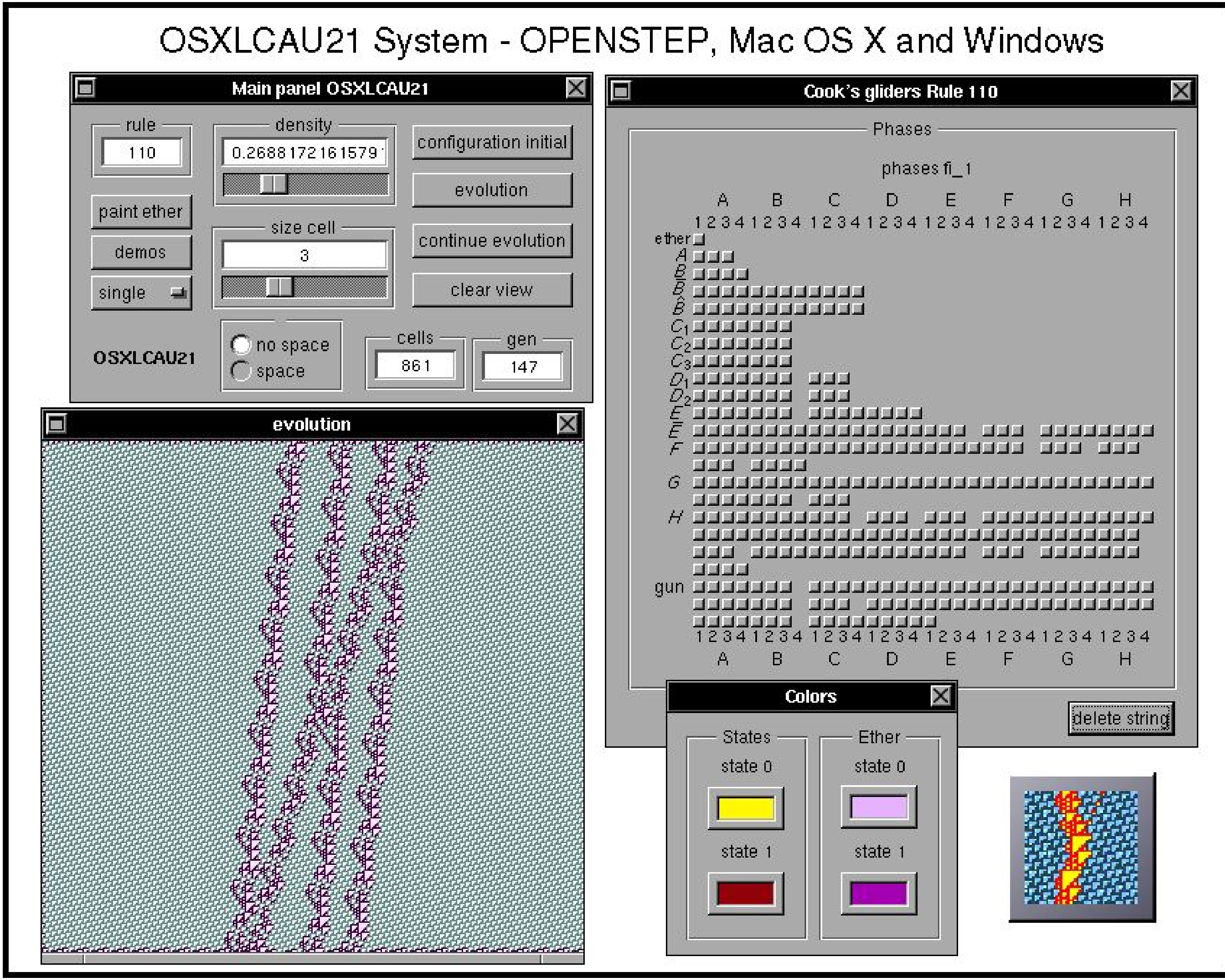
Unlike the tag system the cyclic tag system has two rules of transformation for a same value and these are applied periodically in each step. Cook comments that its system is undecidable, proposing two types of productions (0$→$∅, 1$→$11) y (0$→$∅, 1$→$10), where ∅ means that any element is not added.



Construction in phases f_i_1:
left: $\{649e\text{-}4\_A^4(\text{F}\_i)\}^*$,
where $1 \le i \le 3$ in sequential order
center: $246e\text{-}1\text{Ele}\_C_2(\text{A},\text{f}_1\_1)\text{-}e\text{-}A^3(\text{f}_1\_1)$
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 $1 \le i \le 4$ and # represents a phase in individual (3'228,176,000 cells)

Cyclic Tag System in Rule 110




OSXLCAU21 System - OPENSTEP, Mac OS X and Windows
Main panel OSXLCAU21
rule
110
density
0.2688172161579
configuration initial
evolution
paint ether
size cell
3
continue evolution
demos
single
clear view
OSXLCAU21
no space
space
cells
861
gen
147
evolution
Cook's gliders Rule 110
Phases
phases fi_1
A B C D E F G H
1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
ether
gun
delete string
Colors
States
state 0
state 1
Ether
state 0
state 1

Conclusions

Phases can establish a reproduction/control of collisions in Rule 110. This one series of steps determine a procedure that facilitates the study in this type of automata in one dimension. Once obtained the procedure, it is applied to solve interesting problems in Rule 110.

Procedure is not elegant but if very practical. Then a mechanism is due to develop so that the system classifies the information.

Acknowledgements

In special to Harold V. McIntosh, Juan Carlos Seck Tuoh Mora and Department of Microcomputers of UAP. To Catherine Boucher and Todd Rowland of Wolfram Research Inc.