A note about the regular language of Rule 110 and its general machine: the scalar subset diagram

Genaro J. Martínez[1], Harold V. McIntosh[2],
Juan C. Seck Tuoh Mora[3], and Sergio V. Chapa Vergara[4]

[1] Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England, Bristol, United Kingdom.
http://uncomp.uwe.ac.uk/genaro/
Email: genaro.martinez@uwe.ac.uk

[2] Departamento de Aplicación de Microcomputadoras, Instituto de Ciencias, Universidad Autónoma de Puebla, Puebla, México.
http://delta.cs.cinvestav.mx/~mcintosh/
Email: mcintosh@servidor.unam.mx

[3] Centro de Investigación Avanzada en Ingeniería Industrial, Universidad Autónoma del Estado de Hidalgo Pachuca, Hidalgo, México.
Email: jseck@uaeh.edu.mx

[4] Departamento de Computación, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México.
Email: schapa@cs.cinvestav.mx

Abstract. As it was published in other papers, a regular language can be derived in the elemental cellular automaton (ECA) Rule 110 from a subset of regular expressions produced from its set of gliders. This way, a full description of this subset too is known and reported. This paper will discuss in detail a *general machine* able to validate completely the subset of regular expressions in Rule 110 and other characteristics, such as, the calculus of Garden of Eden configurations in Rule 110. Such machine is the subset diagram.

Keywords: Rule 110, regular expressions, gliders, Garden of Eden, de Bruijn diagram, subset diagram

1 Introduction

ECA Rule 110 has received special attention by the results exposed by Cook [1] and Wolfram [18] mainly. Nevertheless several details were reached for numerous researchers related on this ECA.

In [12] it was reported a subset of regular expressions derived from its family of gliders to code initial conditions in Rule 110. This subset was obtained applying tools as de Bruijn diagrams [4], tiling theory [5, 8], and cycle diagrams [6].

The present paper must discuss how this regular language should be proved from a more general machine. This machine is called *the subset diagram*. The subset diagram was idealized and proposed by McIntosh originally in [4] to study one dimensional cellular automata with other very useful tools, such as, the de Bruijn diagrams, pair diagrams, cycle diagrams, and more.

In this paper, we consider that the reader has basic notions on ECA Rule 110. If you are a novel reader on this subject then we should suggest the next references [1, 18, 10, 5].[5] To the subjects related to the de Bruijn and subset diagrams; please read [4, 6, 16].

2 Regular language in Rule 110

This section is devoted to give a brief introduction in phases and de Bruijn diagrams in Rule 110; a more extended explanation can be found in [12].

Let us remember that a finite subset of regular expressions represented as Ψ_{R110} determines a regular language L_{R110}. The number of sequences w from the binary alphabet Σ given in the set, is the union of the periods for every glider:

$$\Psi_{R110} = \bigcup_{i=1}^{p} w_{i,g} \ \forall \ (w_i \in \Sigma^* \wedge g \in \mathcal{G}) \tag{1}$$

where $\mathcal{G}$ is the whole set of gliders in Rule 110 and $p \geq 3$ is the corresponding period. This way, we can speak of a regular language L_{R110} that is constructed from the expressions of Ψ_{R110}. We most notice that this language is a subset of the whole language in Rule 110, that is, it is only the one defined by the expressions representing gliders, therefore we have:

$$L_{R110} = \{w | w \in \Psi_{R110} \text{ operating under the basic rules: } \cdot, +, *\}. \tag{2}$$

This way, L_{R110} is established by the de Bruijn diagrams and the characterization of tiles, where both have been analyzed for defining

[5] Also you can see a fast introduction, subjects and references related on Rule 110 from `http://uncomp.uwe.ac.uk/genaro/Rule110.html`

useful features called "phases f_i_1." The phases indicate with precision both the position and the exact moment where each glider must be positioned into a given initial condition [12].

Applying the set of regular expressions and their basic operations, we are able to construct desired initial conditions which yield evolutions with important characteristics; the main interest is to control and produce collisions among gliders. In this way L_{R110} is a powerful tool to codify initial conditions in Rule 110. Immediate applications with relevant results in the study of Rule 110 has been performed over hundreds, thousands, millions and thousands of million of cells [9, 8, 11].

Thus de Bruijn diagrams [4, 16] are very adequate for describing evolution rules in one-dimensional cellular automata, although originally they were used in shift-register theory (the treatment of sequences where their elements overlap each other). These diagrams can extract any periodic string of Rule 110 or another CA; particularity we employ the connected cycles from extended de Bruijn diagrams to calculate any string and its shifts over a number of generations.

A glider in Rule 110 can be seen as a periodic construction preserving a defined cyclic border with ether in the evolution space. Essentially a glider has the following characteristics: *volume* (number of cells representing its form), *period* (number of evolutions to recover the original sequence), *displacement* (change of horizontal position measured in cells on finishing its period) and *speed* (velocity produced by the period between the displacement). Thus a set of gliders with different volume and speed can be represented.

In order to explain how the sequences of each glider are determined, a de Bruijn diagram for an A glider is firstly calculated and it is decribed how the periodic sequences are extracted from it or representing this glider and specifying as well its set of regular expressions.

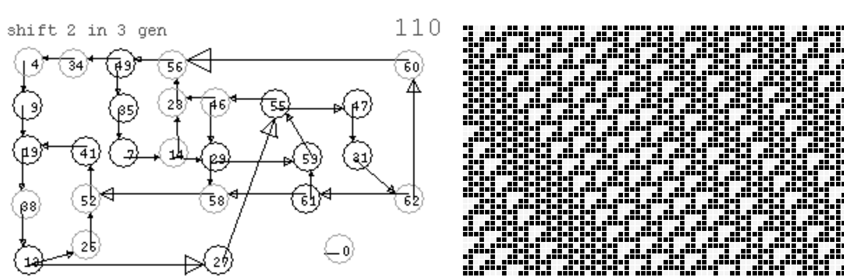

Fig. 1. De Bruijn diagram calculating A gliders within ether configurations.

The A glider moves two cells to the right in three times, the corresponding extended de Bruijn diagram (2-shift, 3-gen) is depicted in Figure 1. Cycles in the diagram are periodic sequences describing each phase in the glider; however these sequences are not ordered yet, hence they must be classified.

Diagram in Figure 1 still has two cycles: a cycle formed by just a vertex 0 and another large one of 26 vertices composed by other nine internal cycles. The evolution of the right illustrates the location of the different periodic sequences producing the A glider in distinct numbers.

The sequences or regular expressions determining the phases of the A glider are obtained following paths through the edges in the diagram. For instance, the following cycles specify different formations:

I. The expression (1110)*, vertices 29, 59, 55, 46 determine A^n gliders.
II. The expression (111110)*, vertices 61, 59, 55, 47, 31, 62 define nA gliders with a T_3 tile among each glider.
III. The expression (11111000100110)*, vertices 13, 27, 55, 47, 31, 62, 60, 56, 49, 34, 4, 9, 19, 38 describe ether configurations in a specific phase, i.e., the expression $e(\mathrm{f}_1\_1)$ [12].

As it was reported as well in [12] the full subset of regular expressions to represent A gliders is formed by the next number of sequences:

$A(\mathrm{f}_1\_1) = 111110$
$A(\mathrm{f}_2\_1) = 11111000111000100110$
$A(\mathrm{f}_3\_1) = 11111000100110100110$
$A(\mathrm{f}_4\_1) = A(\mathrm{f}_1\_1)$

where these sequences are represented in the de Bruijn diagram.

Finally a cycle with period 1 represented by vertex 0 produces an homogeneous evolution in state 0. The evolution space in Figure 1 shows different packages of A gliders. Also this regular language L_{R110} is restricted to gliders in Rule 110. The application of this regular subset allows to solve some important problems, on defining initial conditions codified by phases; offering as well a powerful tool to codify the evolution space of Rule 110.[6]

3 Scalar subset diagram in Rule 110

The scalar subset diagram [4] is derived from the de Bruijn diagram, representing a general machine to verify what sequences belong to the

[6] The regular language L_{R110} does not imply that the evolution of Rule 110 is regular in the sense of limit sets [17, 2, 14], because L_{R110} is only conserved in the composition of the initial conditions.

language produced by Rule 110, and besides this diagram can calculate Garden of Eden configurations [13] and surjective cellular automata [15].

In this way, the subset diagram has $2^{k^{2r}}$ vertices with k states and r neighbors. If all the configurations of certain length have ancestors then all the configurations with extensions both to the left and the right with the same equivalence must have ancestors. But if this is not the case, then they describe configurations in the Garden of Eden and represent paths going from the maximum set to the minimum one in the subset diagram.

The nodes are grouped into subsets, note being taken of the subsets to which one can arrive 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 one can get to from all the different combinations of starting points. Sometimes, but far from always, the possible destinations narrow down as one goes along; in any event one has all the possibilities cataloged.

Thus we can define the subset diagram as follow [4, 6]. Let a and b be vertices, S a subset and $|S|$ the cardinality of S; then the subset diagram is defined by the following equation:

$$\sum_i(S) = \begin{cases} \phi & S = \phi \\ \{b \mid \text{edge}_i\ (a,b)\} & S = \{a\}. \\ \bigcup_{a \in S} \Sigma_i(a) & |S| > 1 \end{cases} \tag{3}$$

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. There is a certain image of the de Bruijn diagram, in the sense that given an origin and a destiny, there is always a subset containing the accessible destiny and another subset containing the origin, besides the destiny 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.

One point to be observed is that if one thinks that there should be a link at a certain node and there is not, the link should be drawn to the empty set instead; a convention which 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). Would be useful first write its de Bruijn diagram – expressing its local function φ – which still has a symbolic variant in two matrices [6].

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 to Rule 110 we have the next 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 CA order $(2,1)$ we have four sequences of states in the Bruijn diagram enumerated as $\{0\}$, $\{1\}$, $\{2\}$ and $\{3\}$. Therefore 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 distinguish; 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.

Table 1. Relation between states of the subset diagram in Rule 110.

subset	decimal	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

In order to determine the type of union between the subsets, the state in which each sequence evolves must be reviewed to know towards which states (subset that form it) may be connected; this way the relation for Rule 110 is constructed in Table 1. So its respective scalar subset diagram for Rule 110 is showed in the Figure 2.

There is another important reason for working with subsets. Labelled links resemble functions, by associating things with one another. But if two links with the same label emerge from a single vertex, they can hardly represent a function. Forging the subset of all destinations, leaves one single link between subsets, bringing functionality to the subset diagram even though it did not exist originally. Including the null set ensures that every point has an image, avoiding partially defined functions.

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 [13].

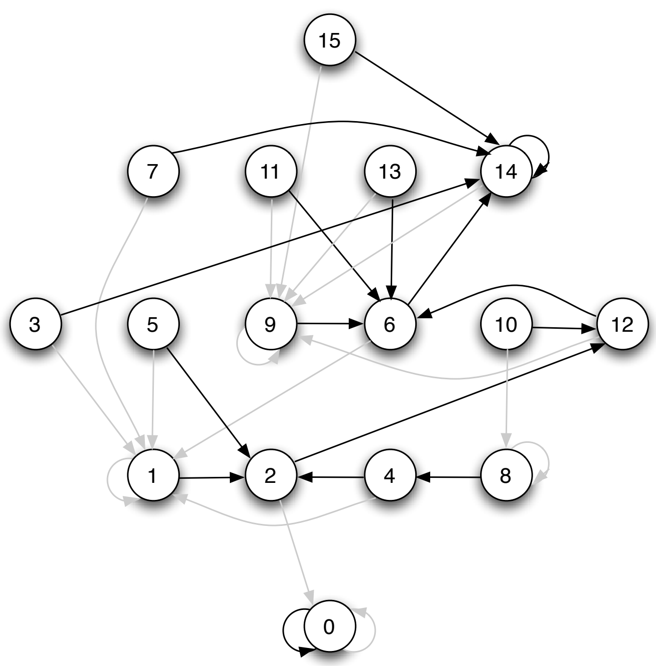

Fig. 2. Scalar subset diagram of Rule 110.

Although the edges between subsets do not define a function, it is well defined for the whole graph by the inclusion of the empty set. Each class of edges defines a function: Σ_0 or Σ_1. The subset diagram describes the join of $\Sigma_0 \cup \Sigma_1$, that by itself is not functional [4, 6].

This way, Figure 2 displays the full scalar subset diagram for Rule 110. Each connection was defined from their relations between subsets (see Table 1). We must distinguish four levels of subsets, where it is possible to transit on its four unit classes. 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 original diagram.

At first instance, we can see that a number of relations are more frequent than others. Also there are nodes without receive any connection as input, or nodes with most connections including loops itself. However more interesting are cycles forming cycles of different lengths. They are important to recognize words or sequences that a cellular automaton could recognize.

Therefore, it is evident that a small subset diagram may be deduced from its original diagram. This diagram shall include only vertexes with cycles, the universal and empty set and the subset of cardinality one, yielding a new diagram that will be more practical to our proposes. The reduction give yet a more small diagram showed in the Figure 3.

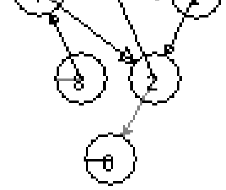

Fig. 3. Reduced scalar subset diagram of Rule 110.

We shall discuss how to exploit this last diagram to describe a general machine recognizing regular expressions in Rule 110. However we can still mention in this moment, that consequently into the subset diagram, we can quickly see paths from the universal set to the empty one; guaranteeing strings without ancestors.

3.1 Garden of Eden in Rule 110

Thus the local function φ of Rule 110 has an injective correspondence, with this correspondence we can find paths in the subset diagram representing Garden of Eden configurations. In this way, we have that two minimal configurations in the Garden of Eden for Rule 110 they are: $(101010)^*$ and $(01010)^*$.

3.2 A general machine recognizing and accepting any regular expression of Rule 110

The subset diagram represents too a powerful *general machine* recognizing and accepting any string of $w_{i,g}$, remembering how a regular language is recognized in classic automata theory [3, 7]. In order to verify this, it is just necessary to take a sequence from the subset of regular expressions and, reviewing if there exists the corresponding path in the subset diagram starting from the maximum set and ending into a nonempty subset.

For example, we can analyze the subset of regular expressions representing a C_2 glider [12].

C_2(A,f$_1$_1) = 11111000000100110
C_2(A,f$_2$_1) = 11111000100000110
C_2(A,f$_3$_1) = 11111000100110000
C_2(A,f$_4$_1) = 11100011000100110
C_2(B,f$_1$_1) = 11111010011100110
C_2(B,f$_2$_1) = 11111000111011010
C_2(B,f$_3$_1) = 1111100010011011111111000100110
C_2(B,f$_4$_1) = C_2(B,f$_1$_1)

In order to verify that any $w_{i,C_2} \in \Psi_{R110}$, we must follow the path of each word in the subset diagram (Figure 3) starting from universal set. This way, the sequences of vertices are given as follows:

C_2(A,f$_1$_1) $\equiv (15, 14, 14, 14, 14, 9, 9, 9, 9, 9, 9, 6, 1, 1, 2, 12, 9)$
C_2(A,f$_2$_1) $\equiv (15, 14, 14, 14, 14, 9, 9, 9, 6, 1, 1, 1, 1, 1, 2, 12, 9)$
C_2(A,f$_3$_1) $\equiv (15, 14, 14, 14, 14, 9, 9, 9, 6, 1, 1, 2, 12, 9, 9, 9, 9)$
C_2(A,f$_4$_1) $\equiv (15, 14, 14, 9, 9, 9, 6, 14, 9, 9, 9, 6, 1, 1, 2, 12, 9)$
C_2(B,f$_1$_1) $\equiv (15, 14, 14, 14, 14, 9, 6, 1, 1, 2, 12, 6, 1, 1, 2, 12, 9)$
C_2(B,f$_2$_1) $\equiv (15, 14, 14, 14, 14, 9, 9, 9, 6, 14, 14, 9, 6, 14, 9, 6, 1)$
C_2(A,f$_3$_1) $\equiv (15, 14, 14, 14, 14, 9, 9, 9, 6, 1, 1, 2, 12, 9, 6, 14, 14, 14, 14, 14, 14, 14, 9, 9, 9, 6, 1, 1, 2, 12, 9)$
C_2(B,f$_4$_1) $\equiv C_2$(B,f$_1$_1)

therefore they are accepted by its scalar subset diagram.

An important remark is that this general machine (the subset diagram), similar to the de Bruijn diagrams, is not a linear one and therefore any vertex can be a terminal state. However in the case of the subset diagram, we need to start from the maximum subset while at the de Bruijn diagram we could start from any vertex.

Therefore the regular language L_{R110} constructed from a subset of regular expressions Ψ_{R110} is verified with its subset diagram.

4 Final remarks

Altogether, the principal value of the scalar subset diagram is to establish such things as:

1. The shortest excluded words, the occurrence of any one of which creates a Garden of Eden configuration.
2. A maximum length for a minimal excluded word, which is the number of nodes in the portion of the subset diagram connected to the full subset.
3. Whether a exclusion occurs in stages, as key segments are built up.
4. A regular expression describing excluded words.

Acknowledgements

First author thanks to EPSRC grant reference EP/D066174; and the third author thanks the support of CONACyT with reference number CB-2007/83554.

References

1. Cook, M. (2004) Universality in Elementary Cellular Automata, *Complex Systems* **15 (1)** 1–40.
2. Hurd, L.P. (1987) Formal Language Characterizations of Cellular Automaton Limit Sets, *Complex Systems* **1** 69–80.
3. Hopcroft, J.E. & Ullman, J.D. (1987) *Introduction to Automata Theory Languages, and Computation*, Addison-Wesley Publishing Company.
4. McIntosh, H.V. (1991) Linear cellular automata via de Bruijn diagrams, `http://delta.cs.cinvestav.mx/~mcintosh/oldweb/pautomata.html`
5. McIntosh, H.V. (1999) Rule 110 as it relates to the presence of gliders, `http://delta.cs.cinvestav.mx/~mcintosh/oldweb/pautomata.html`.
6. McIntosh, H.V. *One Dimensional Cellular Automata*, by publish.

7. Minsky M. (1967) *Computation: Finite and Infinite Machines*, Prentice Hall.
8. Martínez, G.J. & McIntosh, H.V. (2001) ATLAS: Collisions of gliders like phases of ether in Rule 110, `http://uncomp.uwe.ac.uk/genaro/papers.html`
9. Martínez, G.J., McIntosh, H.V., Seck Tuoh Mora, J.C., & Chapa Vergara, S.V. Reproducing the cyclic tag systems developed by Matthew Cook with Rule 110 using the phases f_{i}_1, `http://uncomp.uwe.ac.uk/genaro/papers.html`.
10. Martínez, G.J., McIntosh, H.V., & Seck Tuoh Mora, J.C. (2006) Gliders in Rule 110, *Int. J. of Unconventional Computing* **2 (1)** 1–49.
11. Martínez, G.J., McIntosh, H.V., Seck Tuoh Mora, J.C. & Chapa Vergara, S.V. (2007) Rule 110 objects and other collision-based constructions, *Journal of Cellular Automata* **2 (3)** 219–242.
12. Martínez, G.J., McIntosh, H.V., Seck Tuoh Mora, J.C., & Chapa Vergara, S.V. (2008) Determining a regular language by glider-based structures called *phases* f_{i}_1 in Rule 110, *Journal of Cellular Automata* **3 (3)** 231–270.
13. Moore, E.F. (1956) Gedanken Experiments on Sequential Machines, *in* C. E. Shannon and John McCarthy (eds), *Automata Studies*, Princeton University Press, Princeton, New Jersey.
14. Nordahl, M. (1989) Formal languages and finite cellular automata, *Complex Systems* **3** 63–78.
15. Seck Tuoh Mora, J.C., Chapa Vergara, S.V., Martínez, G.J., & McIntosh, H.V. (2005) Procedures for calculating reversible one-dimensional cellular automata, *Physica D* **202** 134–141.
16. Voorhees, B.H. (1996) *Computational analysis of one-dimensional cellular automata*, World Scientific Series on Nonlinear Science, Series A, Vol. 15.
17. Wolfram, S. (1984) Computation Theory on Cellular Automata, *Communication in Mathematical Physics* **96** 15–57.
18. Wolfram, S. (2002) *A New Kind of Science*, Wolfram Media, Inc., Champaign, Illinois.