

# Computation of Explicit Preimages in One-Dimensional Cellular Automata Applying the De Bruijn Diagram

José Manuel Gomez Soto

SFI WORKING PAPER: 2006-02-005

SFI Working Papers contain accounts of scientific work of the author(s) and do not necessarily represent the views of the Santa Fe Institute. We accept papers intended for publication in peer-reviewed journals or proceedings volumes, but not papers that have already appeared in print. Except for papers by our external faculty, papers must be based on work done at SFI, inspired by an invited visit to or collaboration at SFI, or funded by an SFI grant.

©NOTICE: This working paper is included by permission of the contributing author(s) as a means to ensure timely distribution of the scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the author(s). It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may be reposted only with the explicit permission of the copyright holder.

[www.santafe.edu](http://www.santafe.edu)



SANTA FE INSTITUTE

# Computation of explicit preimages in one-dimensional cellular automata applying the *De Bruijn diagram*.

José Manuel Gómez Soto

*Dirección de Posgrado e Investigación  
Universidad La Salle*

*Benjamín Franklin No. 47. Col. Hipódromo Condesa, 06140, México, D.F. México,  
and*

*Centro de Investigación y de Estudios Avanzados del IPN  
Departamento de Ingeniería Eléctrica, Sección de Computación,  
Av. Politécnico Nacional No. 2508, Col. Pedro Zacatenco, CP 07300, México,  
D.F. México*

*and*

*Santa Fe Institute, Hyde Park Road 1399, Santa Fe, NM 87501, USA.*

---

## Abstract

This paper shows how to simplify the calculation of preimages in cellular automata. The method is based on the so called De Bruijn diagrams and work for any  $k$ -states and  $r$ -radius in one dimensional space. In order to calculate preimages, we construct preimage matrices from the De Bruijn diagram and an operator defined on these matrices. In this way the problem of calculating preimages is reduced to solving the classic *Path-finding Problem* in graph theory, where all possible paths are the preimages of the cellular automaton.

*Key words:* preimages, complex systems, cellular sutomata, De Bruijn diagrams.

---

## 1 Introduction

Cellular automata are discrete mathematical models that have been shown to be useful in studying diverse types of phenomena ranging from physics to biology and from individual to collective behaviour. They originate in von Neumann's work on machine self-reproduction (12) and the studies of shift spaces by Hedlund (3). One important issue in studying a given cellular automata model is how to calculate the preimages of its evolution; since these

preimages can be used to study the dynamics of the global behaviour of cellular automata by the construction of basin of attraction, study surjective and injective properties<sup>1</sup>, or building cyphering schemes, among others.

The *De Bruijn diagram* (2) has been a very useful tool in the study of cellular automata. Nasu (7) used it to study surjective and injective properties. Stephen Wolfram (13) described it as a non-deterministic finite machine. Erica Jen (4) worked with it to consider the concept of recursive relations and the enumeration of preimages. McIntosh (5)(6) used it to enumerate still life and periodic preimages and to study how to proliferate preimages in relation to configuration growth. Sutner(10), who used it to determine if a given one-dimensional cellular automaton is reversible, also used it (11) to characterize surjective mapping property; and recently Yan Deqin (15) applied it to enumerate preimages themselves. However, although in published works the *De Bruijn diagram* is used for computing many properties about preimages, there is no report on the calculation of preimages themselves. The current methods to calculate preimages themselves was proposed in (14) using an algorithmic approach and in (9) by making use of subset diagrams. This paper shows how to calculate preimages in a more simplified scheme, by systematically operating over matrices obtained from the *De Bruijn diagram*.

The importance of the *De Bruijn diagram* is that its edges can be interpreted twofold: either as the mapping of a local rule, which describes the evolution of the cellular automaton or as a description of the neighborhood of this mapping. In this context the edges represent overlaps with their associate vertices, which constitute part of the neighborhood. The De Bruijn diagram can be analyzed using graph theory concepts to obtain the properties of the underlying one-dimensional cellular automata. Specifically, matrix theory can be a powerful tool to perform this analysis. In this context Erica Jen (4), McIntosh (5) and Yan Deqin (15) used De Bruijn matrices to enumerate preimages since these matrices tell us when such preimages actually exist. But if we want to know both the quantity of preimages and their nature, we need to define preimage matrices and a new operator to calculate them. The work presented here provides a systematic and straightforward method to obtain both the quantity and the structure of preimages.

## 2 One-dimensional Cellular Automata

**Definition 1** *A one-dimensional cellular automaton is a quintuple,  $\{\Sigma, \Phi, \varphi, \eta_r(x_i), c_0\}$ , wherein:*

---

<sup>1</sup> Meaning mapping between configurations

- $\Sigma$  is a finite set of states, from which the configurations of  $c$  cells take their values,  $c : \mathbb{Z} \rightarrow \Sigma$ .
- $\eta_r(x_i) = x_{i-r}, \dots, x_i, \dots, x_{i+r}$  is the neighborhood of  $x_i$  of radius  $r$ , whose size is  $\tau = |\eta_r(x_i)|$ .
- $\varphi : \Sigma^\tau \rightarrow \Sigma$ , a local function which maps neighborhoods with size  $\tau$  to a set of states  $\Sigma$ .
- $C_0$ , an initial configuration which is the starting point of the evolution.
- $\Phi$  is a global function that computes transformations between sets of configurations.

The cellular automaton dynamics consists of passing from one configuration to another  $\Phi : \Sigma^\mathbb{Z} \rightarrow \Sigma^\mathbb{Z}$  in discrete steps  $t \in \mathbb{N}$ . To perform the transformation between configurations, the function  $\Phi$  needs the local rule  $\varphi : \Sigma^\tau \rightarrow \Sigma$  that computes the next state of a cell from the current states of all the cells in its neighborhood:

$$x_i^{t+1} = \varphi(\eta(x_i^t)) \quad (1)$$

The space of cellular automaton can be viewed as a one-dimensional bi-infinite array of cells, but for practical studies it is taken as finite and the ends are closed to form a ring, i.e. with periodic boundary conditions.

### 3 The De Bruijn diagram

The De Bruijn diagram is a directed graph  $G = (V, E)$  with a set of vertices  $V(G) = \Sigma^{\tau-1}$ . In terms of one-dimensional cellular automata, each element of  $V(G)$  is formed by all different sequences of  $\tau - 1$  cells. Here  $E(G)$  is a set of edges where each element is associated with a pair of elements of  $V(G)$ ; two vertices  $v_1 = x_1x_2x_{\tau-1}$  and  $v_2 = y_1y_2y_{\tau-1} \in V(G)$  overlap if  $x_2 = x_1$ ,  $x_3 = y_2, \dots, x_{\tau-1} = y_{\tau-2}$ . In this way we associate the edge  $e_{1,2} = (v_1, v_2)$  which is represented graphically by an arrow pointing from  $v_1$  to  $v_2$ . Consequently  $E(G) = \Sigma^\tau$  represents the complete set of neighborhoods of the cellular automaton. We can take each labeled edge as the state in which the neighborhood evolves according to the local rule  $\phi$  or we can take each labeled edge with the neighborhood.

### 4 Preimages

To enumerate preimages Jen(4), McIntosh (5) and Yan Deqin (15) used connectivity matrices of *De Bruijn diagrams*. This matrix tells us when it is

possible to have preimages for each state  $s \in \Sigma$ ; these authors show that by multiplying a sequence of those matrices we can obtain the enumerations of the preimages of the sequence formed for the mapping associated to these sequences of matrices.

However, if we want to determine the preimages, we need other characteristics of the *De Bruijn diagram*. The main focus here is to travel through the graph taking into account the preimages, or the nodes' overlaps. Thus we should label each edge as  $e_{i,j} = N(v_i, v_j)$ , meaning the neighborhood representing the overlaps between node  $v_i$  and  $v_j$  or  $N(v_i, v_j)$ . A sequence of  $h$  edges can be written as:  $u_{i,k_1\dots j} = (v_i, v_{k_1})(v_{k_1}, v_{k_2}), \dots (v_{k_{h-1}}, v_j)$ .

We can consider the states of the cellular automata as the set of states  $\Sigma$  or alphabet. A string or word  $w$  will be a finite sequence of symbols. A special string is the *empty string*, which we shall denote by  $\lambda$ . If  $w_1$  and  $w_2$  are strings, then the extended concatenation of  $w_1$  and  $w_2$ , written  $w_1 \bullet w_2$  or just  $w_1 w_2$ , is the string formed by the symbols of  $w_1$  followed by the last element of the string  $w_2$ . For example, "321"  $\bullet$  "321" = "1321". The extended concatenation of the empty string is not valid<sup>2</sup>. To calculate a extended concatenation neither  $w_1$  nor  $w_2$  can be equal to  $\lambda$ . In this way we can approach the problem to calculate preimages as a *Path-finding problem* (1) but with a refined difference in the concatenation operator.

The sequences of preimages are formed by taking the edge  $(v_i, v_{k_1})$  and concatenating it with the last element of the edge  $(v_{k_1}, v_{k_2})$  and the result will be concatenated with the last element of  $(v_{k_{h-1}}, v_j)$ . Then the word formed by the sequence  $h$  is  $u_{i,k_1\dots j} = (((v_i, v_{k_1}) \bullet (v_{k_1}, v_{k_2})) \bullet \dots \bullet (v_{k_{h-1}}, v_j))$ .

Each word in the resulting language consists, in short, of applying the operation  $\bullet$  in the sequences of edges as mentioned before. This word will be the preimage sequence and the set of these words is a language formed by all preimage sequences. If we want to consider all the preimages for a particular sequence, we need a matrix that tells us not only if it is possible to have a preimage but also what the preimage is for each state of the cellular automaton. The matrices that give this information are called preimage matrices and they are defined as:

**Definition 2** Let  $v_i$  and  $v_j \in V(G)$  of the De Bruijn diagram for  $i, j = 1, 2, 3, \dots, |V(G)|$ . The preimages matrix  $M(s)_{i,j}$  of state  $s \in \Sigma$  is defined as:

$$M(s)_{i,j} = \begin{cases} \{N(v_i v_j)\} & \text{If } \phi(N(v_i, v_j)) = s \text{ where } s \in \Sigma \\ \emptyset & \text{elsewhere} \end{cases} \quad (2)$$

---

<sup>2</sup> This is because when the operand is empty the preimages cannot exist

where its element sets are neighborhood that represent  $v_i$  and  $v_j$  for  $i, j = 1 \dots |V(G)|$  it means  $N(v_i, v_j)$ , where the mapping corresponds to state  $s \in \Sigma$ . To simplify the notation we will denote  $M(s)_{i,j}$  as  $M_s$ .

Next we need an operator  $\odot$  that calculates the preimages from  $M_s$  of the *De Bruijn diagram*. In order to calculate preimages from preimage matrices, we put forward the following:

**Definition 3** Let  $A = [\mathbf{a}_{ik}]$  be a preimage matrix  $m \times n$  and  $B = [\mathbf{b}_{kj}]$  a preimage matrix  $n \times p$ . The operator  $\odot$  of  $A$  and  $B$  denoted by  $A \odot B$  is a preimage sequencer matrix  $C$ , whose elements  $\mathbf{c}_{i,j}$  are defined by:

$$\mathbf{c}_{i,j} = \bigcup_{k=1}^n \mathbf{a}_{ik} \otimes \mathbf{b}_{kj} \quad (3)$$

where

$$\mathbf{a}_{ik} \otimes \mathbf{b}_{kj} = \begin{cases} \bullet_u^n a_{ij} v(b_{ji1}) & \text{If } \mathbf{a}_{ik} \neq \emptyset \text{ and } \mathbf{b}_{kj} \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

$v(b_{jk1})$  takes the last element of  $(b_{jk1})$  and  $\bullet$  is the operation concatenation.

Note that this operation is only defined when  $\mathbf{a}_{ik}$  has one or more configurations and when  $\mathbf{b}_{kj}$  contains just one; in this way we concatenate each configuration of  $\mathbf{a}_{ik}$  to the last digit or cell of  $\mathbf{b}_{kj}$ .

In order to calculate the preimages of a cellular automaton configuration  $a_1 a_2 a_3 \dots a_w$  where  $a_i \in \Sigma$  for  $i = 1 \dots w$ , we need to calculate the composition of the operator  $\odot$  over the matrices  $M_{a_1} M_{a_2} M_{a_3} \dots M_{a_k}$  which represent the sequence  $a_1 a_2 a_3 \dots a_k$ ,

$$M_{a_1 a_2 a_3 \dots a_k} = (((M_{a_1} \odot M_{a_2}) \odot M_{a_3}) \odot \dots) \odot M_{a_k}.$$

The result of this operation  $M_{a_1 a_2 a_3 \dots a_k}$  will contain all the sequences that are preimages of  $a_1 a_2 a_3 \dots a_k$ . If the elements of  $M_{a_1 a_2 a_3 \dots a_k}$  are all  $\emptyset$  then  $a_1 a_2 a_3 \dots a_k$  is a *Garden of Eden*.

## 5 Study cases

Consider a one-dimensional cellular automaton with  $\Sigma = \{0, 1, 2\}$ ,  $\tau=3$ , and the following evolution rule:

$$\begin{aligned}
&\{0,0,0\} \rightarrow 1, \{0,0,1\} \rightarrow 1, \{0,0,2\} \rightarrow 0 \quad \{0,1,0\} \rightarrow 1, \{0,1,1\} \rightarrow 2, \{0,1,2\} \rightarrow 2 \\
&\{0,2,0\} \rightarrow 0, \{0,2,1\} \rightarrow 1, \{0,2,2\} \rightarrow 0 \quad \{1,0,0\} \rightarrow 1, \{1,0,1\} \rightarrow 1, \{1,0,2\} \rightarrow 0 \\
&\{1,1,0\} \rightarrow 0, \{1,1,1\} \rightarrow 2, \{1,1,2\} \rightarrow 1 \quad \{1,2,0\} \rightarrow 2, \{1,2,1\} \rightarrow 2, \{1,2,2\} \rightarrow 1 \\
&\{2,0,0\} \rightarrow 1, \{2,0,1\} \rightarrow 0, \{2,0,2\} \rightarrow 2 \quad \{2,1,0\} \rightarrow 0, \{2,1,1\} \rightarrow 1, \{2,1,2\} \rightarrow 0 \\
&\{2,2,0\} \rightarrow 0, \{2,2,1\} \rightarrow 2, \{2,2,2\} \rightarrow 2
\end{aligned}$$

This rule can be represented by a graph. We can build a De Bruijn diagram (see Figure 1) in which vertices are a portion of the neighborhood. Under this example the vertices are “00”, “01”, “02”, “10”, “11”, “12”, “20”, “21”, “22”. The edges associate two nodes to create a neighborhood by overlapping. The neighborhood or edges that can be formed are “000”, “001”, “002”, “010”, “011”, “012”, “020”, “021”, “022”, “100”, “101”, “102”, “110”, “111”, “112”, “120”, “121”, “122”, “200”, “201”, “202”, “210”, “211”, “212”, “220”, “221”, “222”. Finally, the direction of the edges indicates the way the overlap is possible, and the type of edge denotes the mapping of the neighborhood.

From this graph we can obtain the preimage matrices. Below we have 3 preimage matrices, one for each state. The entries of these matrices give us the neighborhood which maps the state representing each matrix. The preimage matrices are as follows:

$$M_0 = \begin{pmatrix}
0 & 0 & \{“002”\} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \{“020”\} & 0 & \{“022”\} \\
0 & 0 & \{“102”\} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \{“110”\} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \{“201”\} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \{“210”\} & 0 & \{“212”\} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \{“220”\} & 0 & 0
\end{pmatrix}$$

$$M_1 = \begin{pmatrix}
\{“000”\} & \{“001”\} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \{“010”\} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \{“021”\} & 0 \\
\{“100”\} & \{“101”\} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \{“112”\} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \{“122”\} \\
\{“200”\} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \{“211”\} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

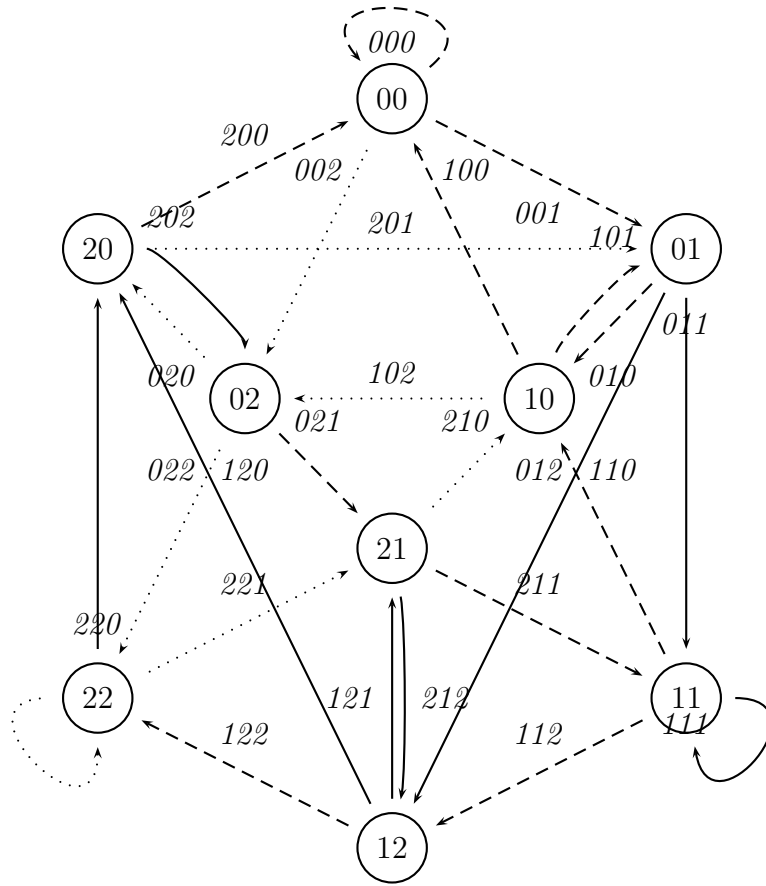


Fig. 1. The De Bruijn diagram representing mapping through its edges. Dotted edges correspond to mapping in 0, dashed edges mapping in 1, and solid edges mapping in 2.

$$M_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \{“011”\} & \{“012”\} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \{“111”\} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \{“120”\} & \{“121”\} & 0 \\ 0 & 0 & \{“202”\} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \{“221”\} & \{“222”\} \end{pmatrix}$$

For instance, if we want to know the preimages of “01221”, we have to calculate:

$$M_{01221} = (((M_0 \odot M_1) \odot M_2) \odot M_2) \odot M_1$$

The result for this case reads:



$$M_{01221} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \{ "1101200" \} & 0 & 0 & 0 & \{ "1101211" \} & & \{ "1101112" \} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & & 0 & 0 & 0 & 0 \\ \{ "2101200" \} & 0 & 0 & 0 & \{ "2101211", "2122211" \} & & \{ "2101112" \} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & & 0 & 0 & 0 & 0 \end{pmatrix}$$

The configurations “1101200”, “1101211”, “1101112”, “2101200”, “2101211”, “2122211” and “2101112” of  $M_{01221}$  are the preimages of “01221”. Some of these preimages can be illustrated in the evolution (see Figure 2) of this cellular automaton.

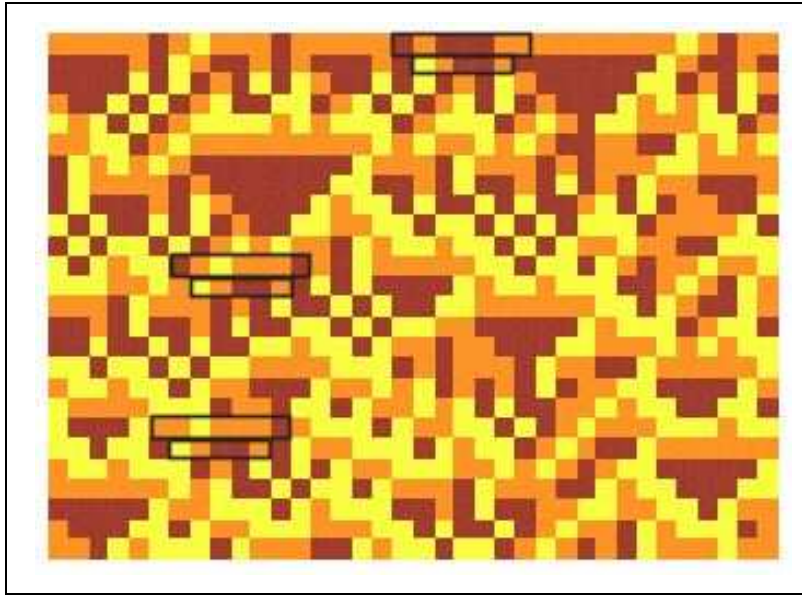


Fig. 2. An example of the spatiotemporal dynamics of the cellular automaton described in this text. Here the preimage configuration and its image are marked by bold lines. The states 0,1 and 2 are represented by different gray levels, from white to black.

## 6 Conclusion

In order to know both the quantity and the nature of preimages in an arbitrary one-dimensional cellular automata, we redefined the times operation between matrices in terms of unions of concatenations in the same way as in the approach of regular algebras in *Path-finding problems* (1), but with a subtle difference: the definition of the concatenation operator. This difference makes it straightforward to calculate the preimages in cellular automata in terms of the *Path-finding problem*. With this change in the concatenation operator, a new algebraic structure remains to be described. The *Path-finding*

approach in the *De Bruijn diagram* reduces the problem of calculating preimages into operating on matrices. This approach thus provides a powerful and elegant framework to calculate preimages in cellular automata. This tool will be useful to study further properties in cellular automata, such as characterizing time-density behavior in cellular automata, calculating preimage trees, obtaining accurate calculations of topological measures based on the spaces of preimages and obtaining accurate statistical measures that can improve the previous scheme of classification of cellular automata.

## 7 Acknowledgments

This work was finished during a research visit to the Santa Fe Institute. The author thanks the support by CONACYT, La Salle University, CINVESTAV-IPN, and Santa Fe Institute. Also I want to thank to Sergio V. Chapa Vergara, Ricard Solé and Andrew Adamatzky for useful comments.

## References

- [1] Backhouse, R. C. and Carre, B. A., “Regular Algebra Applied to Path-finding Problems,” *J. Inst. Maths. Applics*, **15**, (1975), 161-189.
- [2] De Bruijn N.G., “A combinatorial problem,” *Nederl. Akad. Wetensch. Proc.*, **49**, (1946), 758-764.
- [3] Hedlund, G.A., “Endomorphism and automorphism of the shift dynamical systems,” *Math. Syst. Theory*, **3**, (1969), 320-375.
- [4] Jen, E., “Enumeration of Preimages of Cellular Automata,” *Complex Systems*, **3**, (1989), 421-456.
- [5] McIntosh, Harold V., “Linear Cellular Automata via deBruijn Diagram,” <http://cellular.ci.ulsal.mx/comun/cf/debruijn.pdf>, (1991).
- [6] McIntosh, Harold V., “Ancestors: Commentaries on The Global Dynamics of Cellular Automata,” <http://cellular.ci.ulsal.mx/comun/wandl/global.pdf>, (1992).
- [7] Nasu, M., “Local maps inducing surjective local maps of one dimensional tessellation automata,” *Math. Syst. Theory*, **11**, (1978), 327-351.
- [8] Ralston, A., “De Bruijn Sequences, *A Model example of the Interaction of Discrete Mathematics and Computer Science*,” *Mathematics Magazine*, **55**, No. 3, (1982), 131-143.
- [9] Seck, T. J. C., Juárez G, and McIntosh H. V., “Calculating ancestors in one-dimensional cellular automata,” *International Journal of Modern Physics C*, **15**, No. 8, (2004), 1151-1169.
- [10] Sutner, K., “De Bruijn Graphs and Linear Cellular Automata,” *Complex Systems*, **5**, (1991), 19-30.

- [11] Sutner, K., "Linear cellular automata and finite automata," M. De-  
lorne and J. Mazoyer, eds., Cellular Automata: A Parallel Model, Kluwer,  
<http://www-2.cs.cmu.edu/~sutner/papers/epit96.ps.gz> (1999)
- [12] von Neumann, J., "Theory of Self-reproducing Automata," (edited and  
completed by A. W. Burks), University of Illinois Press, (1966).
- [13] Wolfram, S., "Computation theory of cellular automata," Communication  
in Mathematical Physics, **96**, (1984), 15-57.
- [14] Wuensche, A. and Lesser M., "The Global Dynamics of Cellular Au-  
tomata," Addison-Wesley, (1992).
- [15] Yan, D. Hou, Q., "The Enumeration of Preimages and Garden-of- Eden  
in Sequential Cellular Automata," Complex Systems, **12**, (2000), 83-89.