Complex Dynamics in Life-like Rules Described with de Bruijn Diagrams: Complex and Chaotic Cellular Automata

Paulina A. León
Centro de Investigación y de Estudios Avanzados
Instituto Politécnico Nacional
México, D.F.
pleon@ieee.org

Genaro J. Martínez
Unconventional Computing Centre
University of the West of England
Bristol, United Kingdom
genaro.martinez@uwe.ac.uk

Sergio V. Chapa-Vergara
Centro de Investigación y de Estudios Avanzados
Instituto Politécnico Nacional
México, D.F.
schapa@cs.cinvestav.mx

ABSTRACT

De Bruijn diagrams have been used as a useful tool for the systematic analysis of one-dimensional cellular automata (CA). They can be used to calculate particular kind of configurations, ancestors, complex patterns, cycles, Garden of Eden configurations and formal languages. However, there is few progress in two dimensions because its complexity increases exponentially. In this paper, we will offer a way to explore systematically such patterns by de Bruijn diagrams from initial configurations. Such analysis is concentrated mainly in two evolution rules: the famous Game of Life (complex CA) and the Diffusion Rule (chaotic CA). We will display some preliminary results and benefits to use de Bruijn diagrams in these CA.

KEYWORDS: de Bruijn diagrams, cellular automata, complexity, chaos, Life rule, Diffusion rule.

I. INTRODUCTION

Complex dynamics in CA are studied from several perspectives and yielding some interesting results for many years, some of them are: basins of attraction, mean field theory, differential equations, genetic algorithms, cellular complex networks, complexity computation, formal languages, graph theory and so on.

In this way, so called *de Bruijn diagrams* are a special kind of directed graphs, known originally as *shift register sequences*. These diagrams arise due to a series of works focused to study shift register sequences of symbols for the encoding of information; where paths represent sequences of states [3].

The de Bruijn diagrams already had been applied in CA (mainly in one dimension case), previously by McIntosh [9], Wolfram [18], Jen [6], Voorhees [15], Sutner [14], among other researchers.

In this paper, we will give an introduction selecting de Bruijn diagrams in two-dimensional CA. We consider two special cases of study: the famous *Game of Life* (complex behaviour) [2] and the *Diffusion Rule* (chaotic behaviour) [8]. We think that is possible to obtain a practical representation of complex patterns by de Bruijn diagram sequences, mainly for the smallest and compact mobile self localizations in both CA.

II. ANTECEDENTS

For several years global and local behaviour in CA have been studied from several point of views. In this work we

978-1-4673-2362-8/12/$31.00 ©2012 IEEE

will concentrate our attention on the de Bruijn diagrams.

De Bruijn diagrams are a powerful tool to calculate ancestors, Garden of Eden configurations, regular language, set properties, topological properties, and some fragments of complexity in CA. In this way, mainly McIntosh [10] and Voorhees [15] have developed a wide of research reported for several years.

In this direction, we will mention two previous relevant applications of de Bruijn diagrams in two-dimensional CA. First, McIntosh in 1988 wrote the first approximation to construct the de Bruijn diagrams in two-dimensional CA, in two papers: *Life's Still Lifes* [11] and *A Zoo of Life Forms* [12]. By the way, Eppstein has had selected between others, the de Bruijn diagrams algorithms to find new complex patterns in Life [1].

Therefore, following McIntosh' s representation, in this paper we will concentrate our attention in small and compact complex patterns in Life and Diffusion rule from its respective de Bruijn diagrams.

III. DE BRUIJN DIAGRAMS

De Bruijn graph is a directed graph with s^n nodes, which represent length sequences n of s symbols, where at least one overlapping symbol exists.

Thus, applied to cellular automata theory:

- The symbols represent the states, each state value will be represented by v.
- The nodes of such graphs are partial neighbourhoods formed by the half of a neighbourhood.
- The evaluated cell in $t+1$ will be called *center cell*.

For one dimensional neighbourhood the nodes are strings of symbols but for two dimensions it is not possible, therefor the partial neighbourhoods are defined by their characteristics:

- *Form:* This depends of the original neighbourhood, where one half of the neighbourhood must overlap at least on cell with another half.
- *Cell number*: It's the n number of cells that the partial neighbourhood contains. Each of this cells are labeled, they have one position in the partial neighbourhood and they have a state value.
- *Overlap condition*: The relationship between nodes is given through overlapping cells to form a full neighbourhood. This cells must have the same state value and the center cell must be inside this group of cells.

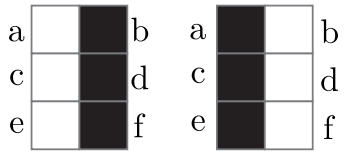

Figure 1. Moore partial neighbourhood

- *State value in cell center at* $t+1$: The state value in cell center v_{ce} is the result of the evolution rule of the CA in study, and labels the edge.

For example, Fig. 1 shows how the Moore neighbourhood is divided in two parts as a rectangle with six cells; the overlapping cells are b, d, f in the fist node, and a, c, e in the second node, and the cell center is the cell d in the first node; where $s = 2$ and $n = 6$ and consequently there are 2^6 nodes.

However, this is the basic case, to know the behaviour of a cell set, there are some considerations such as ensuring that all the necessary cells to evaluate the set are in the neighbourhood, formed by the relation between nodes, and this relation grows exponentially.

Figure 2(a) displays a partial neighbourhood evaluating nine cells, these are the center cell and the eight neighbours. Thus, the partial neighbourhood has 15 cells, and overlaps with 10 cells, the nodes number are 2^{15}. Figure 2(b) shows the partial neighbourhoods evaluating 25 cells, overlapping cells in this case are 21 and nodes are 2^{25}. However, overlapping cells affect directly the number of final relations between nodes, that way the search relation possibilities get smaller.

IV. DE BRUIJN DIAGRAMS IN 2D CA

There are a variety of methods to build de Bruijn diagrams, with advantages and disadvantages in computational complexity. In this section, we will describe a method based on paths of length l, in addition to show the possible results obtained.

Typically, we have two basic kinds of neighbourhoods in 2D CA literature: *von Neumann* and *Moore*. Since each one has specific features and properties its treatment will be different and reflected in all nodes. Algorithm 1 shows the number of steps to build de Bruijn diagrams in 2D CA. However, this section shall describe an example with *Moore* neighbourhood, only.

All the features that must be considered to build de Bruijn diagrams will be described step by step.

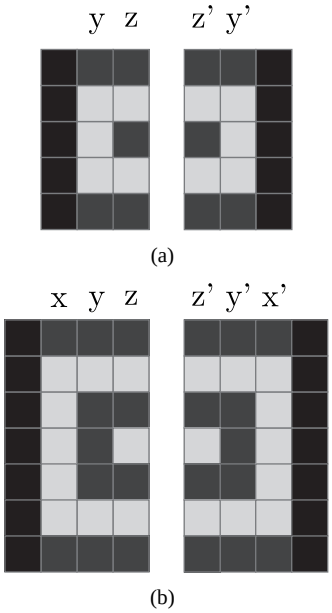

Figure 2. Examples of partial neighbourhoods for a cell set to evaluate.

Algorithm 1 Building de Bruijn Diagrams for 2D C.A.

Compute all the elements or nodes of diagram.
Look for all the relations between all the elements according the *overlapping condition*
Evaluate all the resulting configuration with the cellular automata rule that will be analyzed.
Filter the results in accordance the cellular automata behaviour.
Look for paths of length l into the de Bruijn diagram.
In order to build the second dimension, the paths of the last step will now be the nodes and the steps 2, 3, 4 and 5 will be repeated.

A. Computing nodes

As discussed in section 3, the number of elements of the de Bruijn diagrams depend of the state number of cellular automata and the cell number into the sub neighbourhood.

This paper will show the analysis with cellular automata rules of two states and Moore neighbourhood. Giving values to the variables $s = 2$, $n = 6$, nodes $2^6 = 64$, the overlap condition is showed by figure 1, where black cells are overlapped.

Each node represents a possible combination of state values that the sub neighbourhood can have. Here, in order to label each sub neighbourhood, its binary representation will be used and the sequences will be built from right to left and from bottom to top, so that:

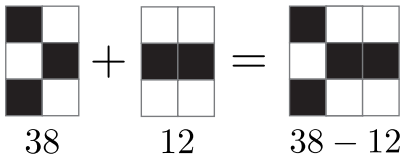

Figure 3. Example of overlap

- 1 will be when cell f is black.
- 32 when cell a is black
- zero if all the cells are white.
- 63 when all cells are black.

B. Working relations

A complete neighbourhood is created through overlapping two sub neighbourhoods using this condition: $b = a$, $d = c$, $f = e$, where b, d, f are cells in node 1, and a, c, e are cells in node 2. The central cell must be within the set of overlapped cells.

Figure 3 shows one example about the overlapping between two sub neighbourhood types figure 1.

C. Evaluating configurations

Once we have all the relationships between the sub neighbourhoods, it is necessary to know the state value of each evaluated cell at $t+1$; this depends on the cellular automata rule. The resulting string will be the size and shape of the region of the original evaluated cell set. That is, if it is only one evaluated cell, the resulting string will have only one state value, if it is a region of $n \times m$ evaluated cells, the resulting string will be also $n \times m$, not modifying the position in space that each cell has as result of the evaluation, where each cell is the centerpiece of a neighbourhood.

D. Filtering results

A way to perform the analysis of cellular automata is through their behaviour. In the literature of the cellular automata, there are two behaviours that are analyzed: the permanent and shifting: due to these features, it is possible to find patterns such as, *still lifes*, and *gliders*.

Although this step is not necessary, in order to find some patterns into the rule (one of de Bruijn diagram goal) it is recommended to perform the filters. There are two kinds of filters: *the permanence* and *shifting*.

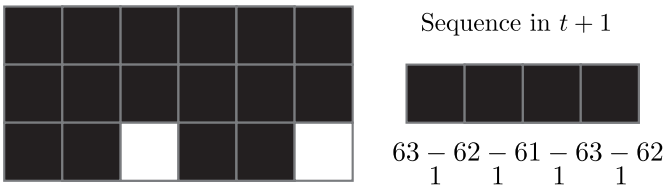

Figure 4. Example

The *permanence filter* is when the evaluating cells have the same state value that the resulting evaluated cells.

The *shifting filter* is when one neighbour has the same state of the evaluating cell. So, there are three kinds of filters in the Moore neighbourhood:

- Bottom to top: taking the first node, is when the evaluated cell has de same value that the cell f.
- Left to right: taking the first node, is when the evaluated cell has de same value that the cell c.
- Diagonal: taking the first node, is when the evaluated cell has de same value that the cell e.

E. Paths in de Bruijn diagrams

In order to build an evolution space whose state at time $t+1$ is known, we have to look for all the possible relationships between nodes.

This query can be carried out by methods of searching paths graphs, such as Euler and Hamiltonian paths, where cycles have size e, with the purpose of having an order among all possible configurations.

The previous query, when done exhaustingly in combination with the filters of the previous step, can lead us to know all the patterns in the rule with a size e.

Figure 4 shows the result of performing the steps described above, where a permanence filter with Diffusion Rule was done.

V. Life-Like patterns via de Bruijn diagrams

Life like rules are a good example of the application of de Bruijn diagrams in two dimensional cellular automata, where the Moore neighbourhood is used.

In order to show the results given by de Bruijn diagrams, this section shows some specific examples with the Game of Life and Diffusion rules.

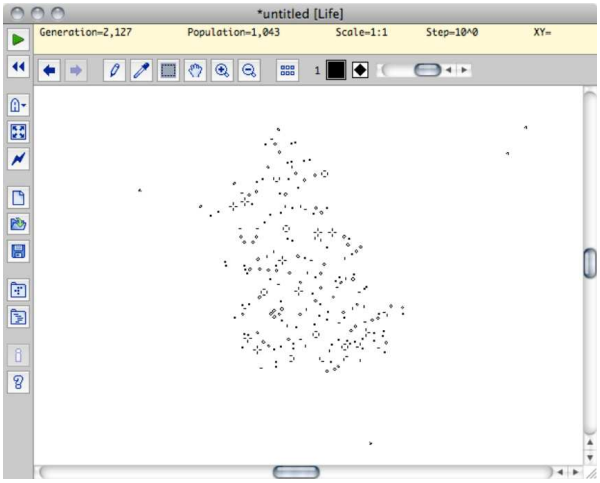

Figure 5. Typical evolution in Life from a random initial condition, some complex patterns emerge, such as: gliders, oscillators, and still life configurations.

A. The Game of Life

The complex behaviour in the *Game of Life* ($B3/S23$ or Life) has been studied since the 70's. In order to make universal computation with Life, interesting patters have been used. These patters emerge as resulting of evaluating cells.

To play the Life's game, we will consider the next rules:

- One dead cell will live if it has 3 cells alive in its neighbourhood.
- If one cell is alive it will remain alive if it has 2 or 3 living cells in its neighbourhood.
- If one cell is alive it will die if has less than 2 or more than 3 living cells in its neighbourhood.

A way to find patterns in the cellular automata behaviour is using de Bruijn diagrams. Figure 6 and 7 show a part of the complete diagram; it shows the relation between nodes and the state value after one evaluation.

Figure 6 uses *permanence filter* so that the center cells will remain with the original state value. In this case the string 7-26-33-7 (top of figure 6) shows that the resulting string will be 100 after evaluating all the neigbourhoods.

Figure 7 uses *shifting filter* in diagonal, so that string 18-33-7-10-1-18 will give as a result the string 10110. In this way, the de Bruijn diagrams could be used to build strings with length l.

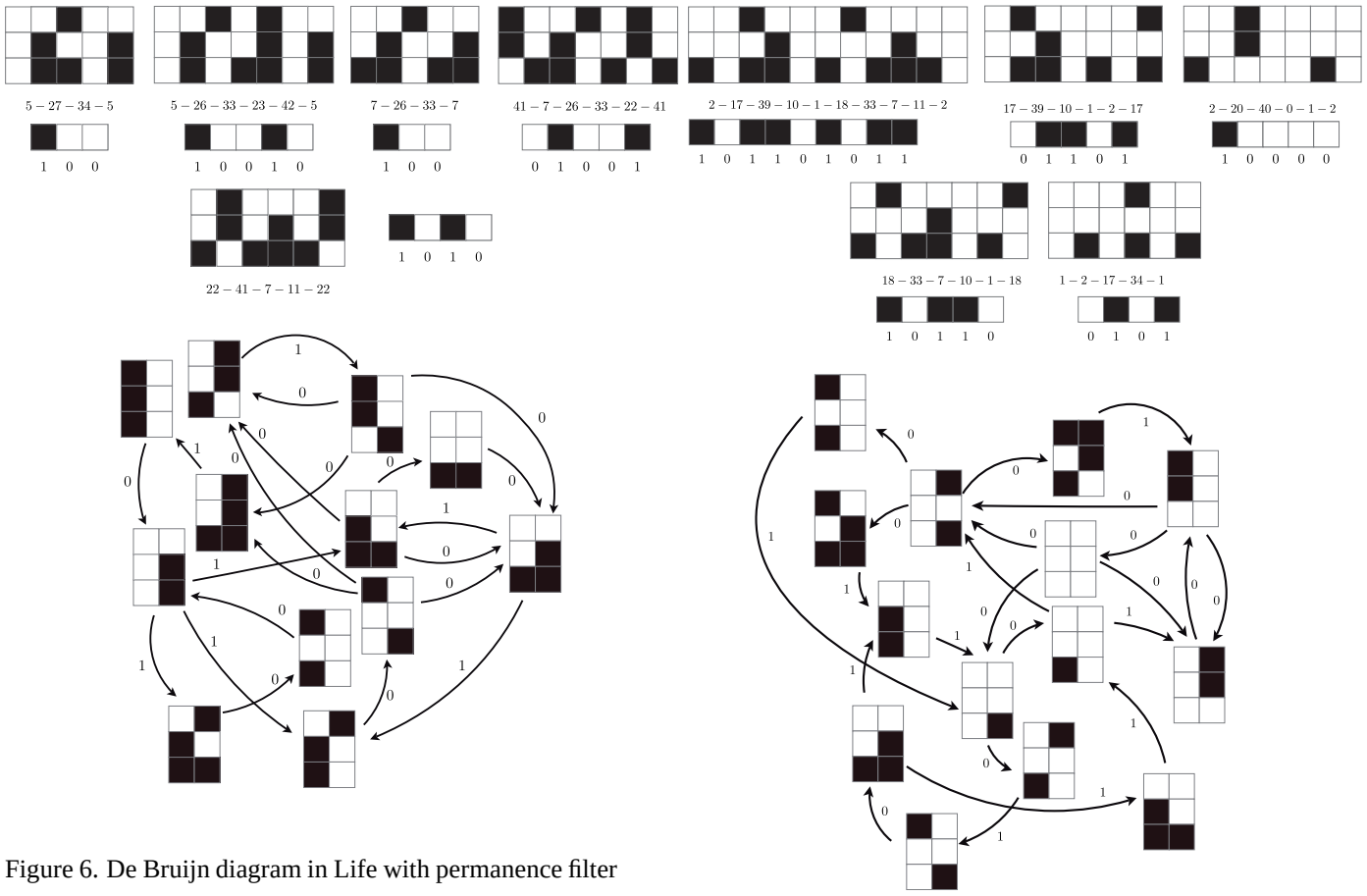

Figure 6. De Bruijn diagram in Life with permanence filter

Figure 7. De Bruijn diagram in Life with shifting filter in diagonal.

B. Diffusion Rule

Another interesting rule is known as *Diffusion Rule* ($B2/S7$). Such CA presents typically a chaotic evolution (see Fig. 8), but also a complex behaviour from very low initial densities. Starting with very low densities in state 1. Typically any random initial condition shall yield different complex patterns as gliders or oscillators [8].

The *Diffusion rule* is defined as follows:

- One cell in state 0 will be in state 1, if it has exactly two neighbours in state 1. If not the state remains 0.
- One cell in state 1 will remain in state 1 if it has exactly seven neighbours in state 1. If not it will change to state 0.

In Fig. 9 Diffusion was made using permanence filter showing strings with length 4 and 5; button of each configuration is shown the result of the evaluation, they have length 3 and 4. Also, Fig. 10 uses *shifting filter* in diagonal and shows the strings that are built with length 4, 5, 6 and below of each configurations is showed the evaluated cells with length 3, 4 and 5. Consequently, concatenations of these patterns are useful to inspect the behaviour of periodic structures. Indeed, sometimes they form new complex patterns.

VI. FINAL NOTES

This paper gives a brief introduction about the de Bruijn diagrams in 2D CA Life-like rules. This way, Moore neighbourhood which display a symmetry, such that is convenient to create spaces for evaluation of cells. Diagrams grown quickly but following some simple filters hence we can avoid some complicated calculus and concentrate our attention on fragments of these diagrams that are useful, the cycles.

Cycles in the de Bruijn diagrams represent precisely a kind of formal language, derived from the evolution rule.

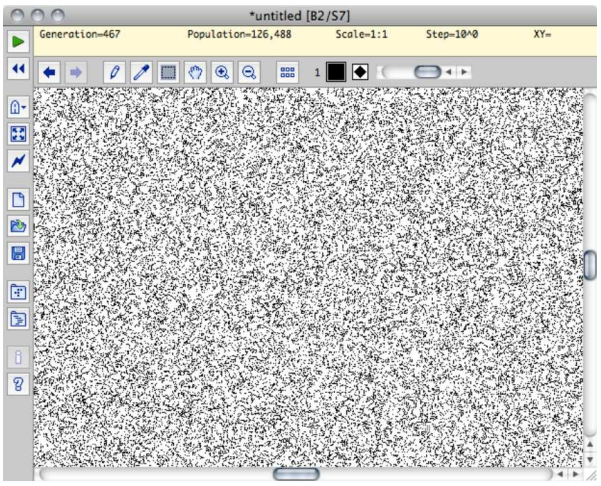

Figure 8. Typical evolution in Diffusion rule from a random initial condition, dominated for chaos.

In this case, also such strings will help us to find small complex patterns emerging in Life and Diffusion Rule. Of course, Life has the most bigger research in this direction with other tools, here only we want to test and compare our results, and yield partial new results for the Diffusion Rule. Our goal is classify systematically these patterns and construct most bigger and complex configurations. Of course, to generate such patterns we need increase the number of shifts and generations that is the future work.

As an initial result we have discovered a new puffer train configuration in Diffusion Rule following the first cycle in figure 9.

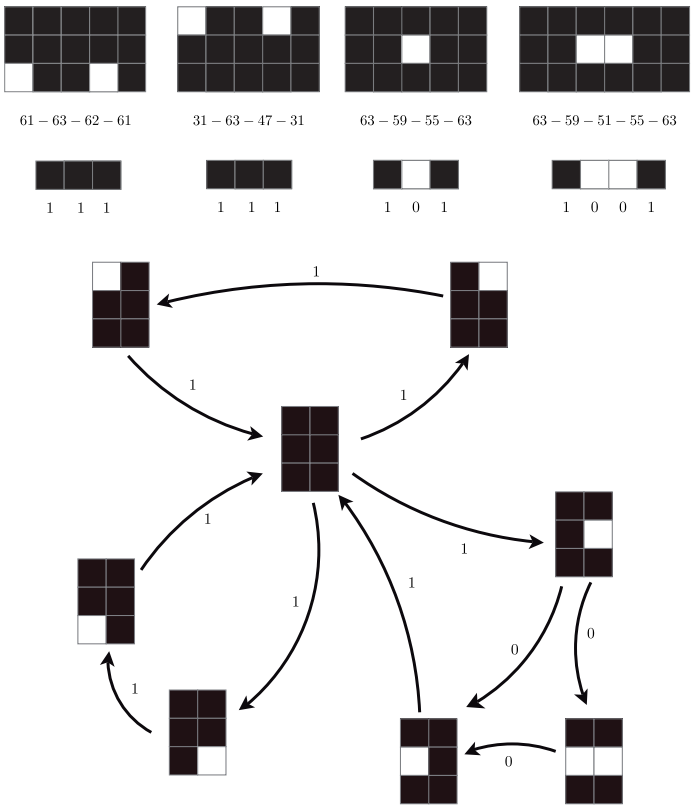

Figure 9. De Bruijn diagram in Diffusion rule with permanence filter.

ACKNOWLEDGMENT

Paulina A. León (scholarship: 50730) would like to thank CINVESTAV and CONACYT for their great support.

REFERENCES

[1] D. Eppstein, Searching for Spaceships, MSRI Publications, vol. 42, 2002, pp. 433–452.

[2] M. Gardner, “Mathematical games: The fantastic combinations of John H. Conway’s new solitaire game Life”, Scientific American, vol. 223, 1970, pp. 120–123.

[3] S. W. Golomb, Shift Register Sequences, Holden-Day, San Francisco, USA, 1967.

[4] H. Gutowitz, Cellular Automata Theory and Experiment, MIT Press, Amsterdam, Netherlands, 1991.

[5] A. Ilachinski, Cellular Automata: A Discrete Universe, World Scientific Publishing, Singapore, 2001.

[6] E. Jen, “Global Properties of Cellular Automata”, Journal of Statistical Physics vol. 43 (1-2), 1986, pp. 219–242.

[7] E. Jen, “Cylindrical Cellular Automata”, Communication in Mathematical Physics, vol. 118 (4), 1988, pp. 569–590.

[8] G. J. Martínez, A. Adamatzky, H. V. McIntosh, “Localization dynamics in a binary two-dimensional cellular automaton: the Diffusion Rule”, Journal of Cellular Automata, vol. 5 (4-5), 2010, pp. 289–313.

[9] H. V. McIntosh, “Linear cellular automata via de Bruijn diagrams”, Unpublished.

[10] H. V. McIntosh, One Dimensional Cellular Automata, Luniver Press, Bristol, UK, 2009.

[11] H. V. McIntosh, “Life’s Still Lifes”, In: Game of Life Cellular Automata (A. Adamatzky, (Ed.)), Springer-Verlag London, 2010, chapter 4, pp. 35–50.

[12] H. V. McIntosh, “A Zoo of Life forms”, In: Game of Life Cellular Automata (A. Adamatzky, (Ed.)), Springer-Verlag London, 2010, chapter 5, pp. 51–68.

[13] J. von Neumann, Theory of Self-Reproducing Automata (Edited and completed by A. W. Burks), University of Illinois Press, IL, USA, 1966.

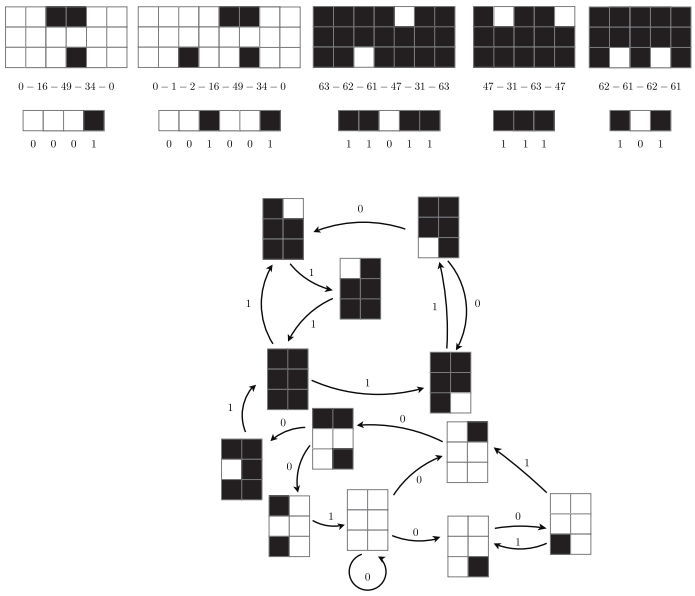

Figure 10. De Bruijn diagram in Diffusion rule with shifting filter in diagonal.

[14] K. Sutner, “De Bruijn graphs and linear cellular automata”, Complex Systems, vol. 5(1), 1991, pp. 19–30.

[15] B. H. Voorhees, Computational Analysis of One-Dimensional Cellular Automata, World Scientific Series on Nonlinear Science, Series A, vol. 15, Singapore, 1996.

[16] B. H. Voorhees, “Remarks on applications of de Bruijn diagrams and their fragments”, Journal of Cellular Automata, vol. 3(3), 2008, pp. 187–204.

[17] S. Wolfram, “Statistical mechanics of cellular automata”, Rev. Modern Physics, vol. 5, 1983, pp. 601–644.

[18] S. Wolfram, “Computation theory of cellular automata”, Communications in Mathematical Physics, vol. 96, 1984, pp. 15–57.