ПРОГРЕСИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, ВИСОКОПРОДУКТИВНІ КОМП'ЮТЕРНІ СИСТЕМИ

UDC 518.9 : 004.45
DOI: 10.20535/SRIT.2308-8893.2019.1.03

GAME OF LIFE WITH NON-REGULAR SPACE WITH BOUNDARIES: GLIDER CASE

JORDAN BRAJON, ALEXANDER MAKARENKO

Abstract. The purpose of this article is to present the work done on the implementation of rules for gliders in a game of life with a non-regular network with boundaries. First of all, we will recall the basic principle of the game of life by mentioning some structures that appear regularly and are very important as gliders. We will improve the accuracy of the collision rules between gliders. Then, we will introduce non-regular space by adding a new state for cells in boundaries. Thus it will be necessary to give the rules relating to this new cellular automaton. We will finally deal with logic gates by giving which we obtained this modified game of life.

Keywords: cellular automata, gliders, internal boundaries, logical operations.

INTRODUCTION

The purpose of this article is to present the work done on the implementation of rules for gliders in a game of life with non-regular network with boundaries. First of all we will recall the basic principle of the game of life by mentioning some structures that appear regularly and are very important as gliders. We will precise the collision between gliders. Then we will introduce non-regular space by adding a new state for cells in boundaries. Thus it will be necessary to give the rules relating to this new cellular automaton. We finally will deal with logic gates by giving which we obtained with this game of life modified.

BASICS IN THE GAME OF LIFE

The game of life is a cellular automaton discovered by John Conway in 1970. It is undoubtedly the best known cellular automata and it has been fascinating researchers for almost 50 years. John Conway manages to find a system with simple rules and a complex behavior: it is called emergence. Unpredictable complex phenomena emerge from simple rules. This idea of emergence is at the heart of many fields such as mathematics, physics, artificial intelligence or economics but also the social sciences, philosophy or the media [1]. Thus the game of life is an object of study for mathematicians but not only. The philosopher Daniel Dennett declares that "every philosopher should study the Game of Life carefully and it is only by succeeding in thinking about the ideas of conscience and free will in such a world that we will understand its true nature" [2]. The game of life can be likened to a plan and infinite network of cells. These cells can be in two states: dead or alive. Generally dead cells are represented by white boxes and living cells by black boxes. The game of life is a discrete dynamic system which means that a given configuration will evolve over time, evolution is not continuous but discrete. The evolution rule is applied synchronously to the entire network. This rule is very simple and it can summarized as follows:

- a living cell stays alive if it has two or three living neighbors otherwise it dies;
- a dead cell becomes alive if it has exactly three live neighbors otherwise it remains dead. The neighbors of a cell are the cells in Moore's neighborhood of order 1 [3]. In other words, the eight cells whose distance associated with the infinite norm [4] is 1 (see fig. 1).

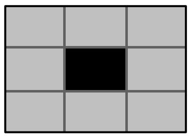
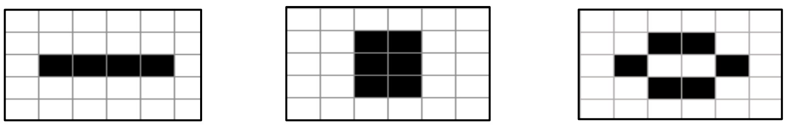


Fig. 1. Living cell (in black) and its eight neighbours (in grey). An example of evolution is given fig. 2.



$t=1$ $t=2$ $t=3$

Fig. 2. Evolution of a simple structure

To deepen the brief notions that we have just seen, the following videos are very complete and very accessible [5] et [6]. Many are working on the game of life. And some of them are studying variants among which we can mention: the addition of a probability in counting the number of neighbors [7] and [8], the modification of the rule of local evolution [9], applying the local transition rule asynchronously [10]. We will also be interested in a variant of the game of life, as we will see later.

SPECIAL PATTERN: GLIDERS

When we consider a random initial configuration with many cells and we study its evolution over time we often observe the same phenomenon. A transitory regime that seems chaotic where the different living cells interact with each other, then an established regime where appear different characteristic patterns of the game of life. Among these patterns there are: still life (see fig. 3), oscillators (see fig. 4 and fig. 5) and the spaceships (see fig. 6). Still life is a pattern that does not change from one generation to the other, oscillators returns to their initial state after a finite number of generations and spaceships translate themselves across the space after a finite number of generations. The vessels are therefore characterized by three numbers (a, b, c) where a denotes the horizontal shift, b the vertical shift and c the number of steps necessary to recover the initial configuration shifted by a cells horizontally and b cells vertically.

The reader will get more information on these patterns and on the game of life in general in the article written by Jean-Paul Delahaye [2]. In this part we will focus more particularly on the glider spaceship.

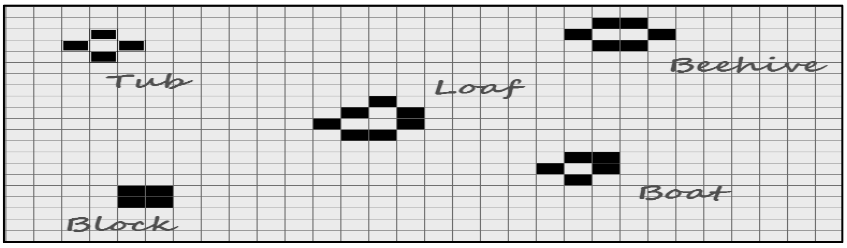


Fig. 3. Still life patterns

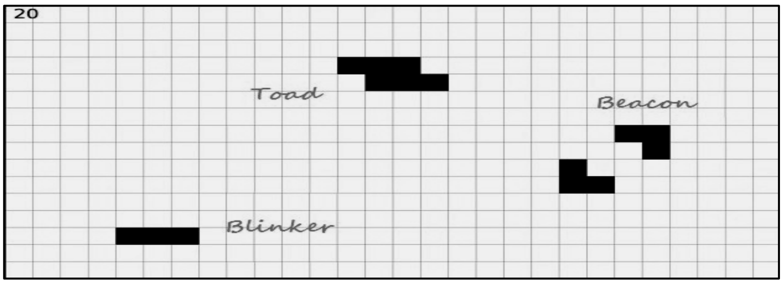


Fig. 4. Oscillators 1/2

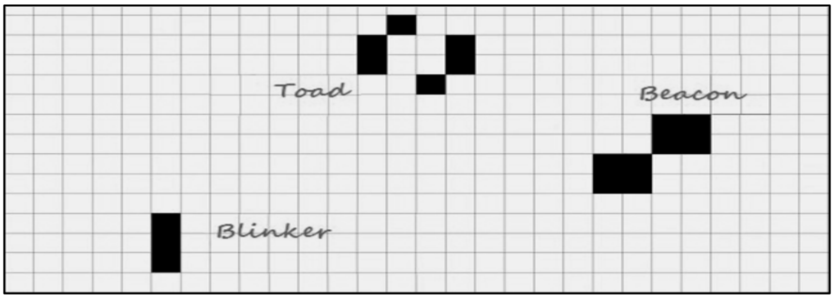


Fig. 5. Oscillators 2/2

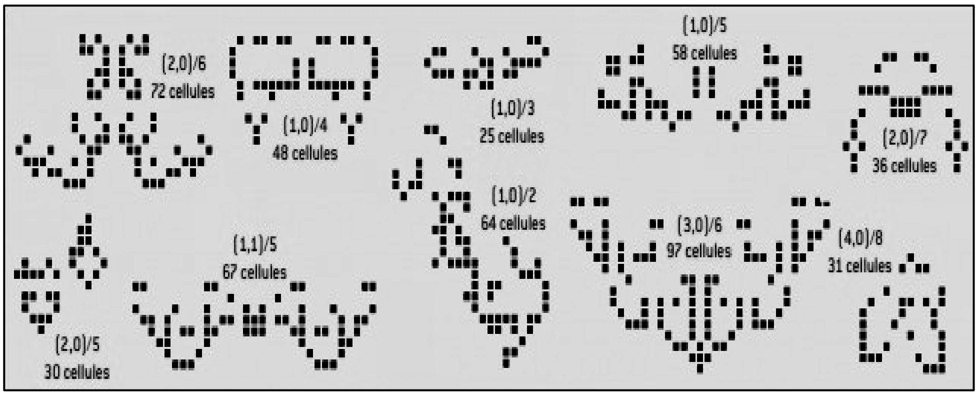


Fig. 6. Some spaceships characterized by (a, b)/c

Let's begin by explaining how ships are particularly interesting objects of study that arouse the interest of different researchers working on the game of life. First of all they allowed to show that there are some configurations whose the growth is infinite in space. Then, and this is with no doubt the most important point, they allow interaction between different regions of space. The spaceships are the vectors of the information and for this reason they will be useful for the implementation of logical gates.

The glider mentioned above is particularly popular because of its simplicity and rapid discovery. It moves from one horizontal cell and from one vertical cell every four generations. Each of them are represented fig. 7.

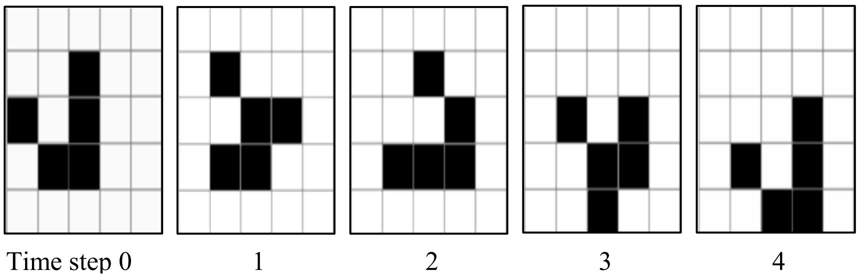


Fig. 7. Configurations of a glider which moves down and to the right

It is important to note that the symmetry of the network on which we study the game of life (in an infinite plane) assures us that from a ship moving in a given direction, we can obtain by symmetry three other ships moving in three other directions by successive rotation of $\pi/2$ angle. For this reason it is enough to specify the horizontal and vertical displacement of a ship without specifying the direction of movement. Then we can get four gliders moving each along the four diagonals of space. There are represented on the fig. 8.

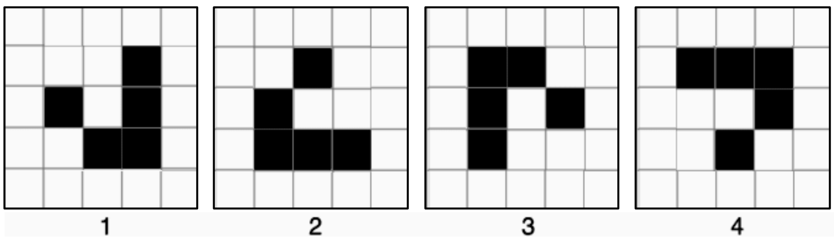


Fig. 8. Four gliders which moves down right (1), down left (2), up left (3) and up right (4)

As mentioned above these gliders will allow interactions between different space areas. More specifically what will be interesting and will be at the end of this part is the interaction of two gliders. When two gliders meet, these will interact to give a few generations later a new configuration. We intuitively call it a collision. Between two gliders there are 73 different collisions. After a collision, two gliders can disappear entirely or reveal certain configurations such as still life or oscillators or even give birth to a new glider. In his article [11], Jean Philippe Renard show a few configurations where two gliders can collide. We will just deal with the collisions useful for the implementation of logical gates. We need two kind of collisions: those that annihilate the two gliders (see fig. 9) and those giving birth to a new glider (see fig. 10 and 11). As for annihilation, the fig. 9 gives the position of the two gliders just before the collision (the one on the left moving down right and the one on the right moving down left). After 4 iterations there are no living cells left, the two gliders have completely disappeared.

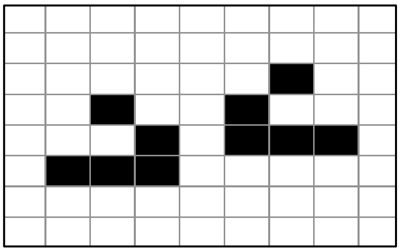


Fig. 9. Configuration of two gliders before collision which would annihilate them

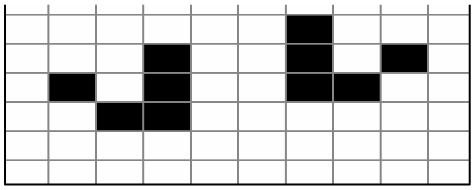


Fig. 10. Configuration of two gliders before collision which would give another glider

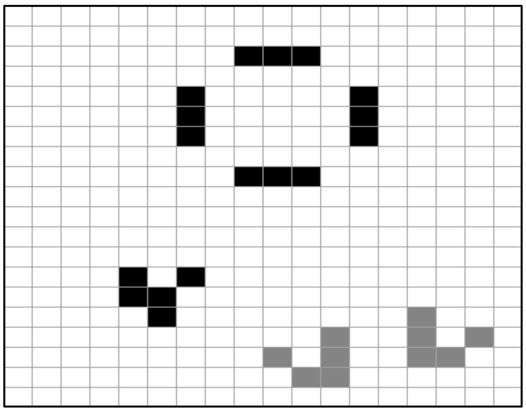


Fig. 11. Configuration of two gliders before (grey) and after (black) collision which would give another glider. Non-regular space

Regarding the creation of a new glider, the fig. 10 gives the position of the gliders just before collision. After 62 calculation steps, we get 4 blinkers and a new glider moving down left. The fig. 11 superimposes the relative position of the two gliders before collision and the result of collision obtained 62 generations later. There are other faster collisions (62 steps being relatively long on the time scale that interest us in this study) that give rise to a new glider. In addition they do not let appear unwanted blocks (the blinkers in this case). Unfortunately the gliders then created do not move in the desired direction.

As we saw above (cf part 1), the game of life is defined on a two-dimensional network. Many are those who have studied the game of life and some of them have worked on modified versions. On the other hand, few have proposed a study on a different network than the plane space usually used. However we can quote the work of Alexander Makarenko [12]. The implementation of a game of life defined on an irregular network will be the subject of this part. To do this we propose, like Alexander Makarenko [12], to add a third frozen state that will represent the irregularities of our initially two-dimensional network. Thus, in addition to the two current states: living cells (black boxes) and dead cells (white boxes), a third state which will be called "walls" (represented by green boxes) will be taken into account. This will allow us to modify the networks as we want (an example is given fig. 12).

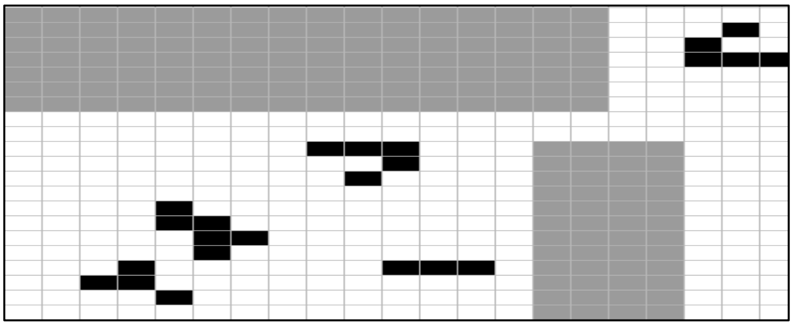


Fig. 12. Non-regular network with walls (in grey)

This new three-state cellular automaton is not entirely defined since it remains to give the local evolution rules. The walls being in a permanent state and the living or dead cells behaving like in the traditional game of life as long as they do not touch the walls it remains only to define the behavior of the living and dead cells when they are in contact with a wall. We will get as many different results as it is possible to choose different rules. This leaves us with an important choice (cf property 1) and gives us hope that the study of such games of life with non-regular networks is a vast subject of research that could be exploited in the future.

Property 1 (number of rules in a network with walls) There are 3^{12610} over cellular automata of game life having three states with one of them is permanent and having a order one neighborhood of Moore as the game of life.

Proof 1 (proposition 1) Let be an over cellular automata of the game life with three states: state 0, state 1 et state 2. Suppose that state 2 is a permanent state so the restriction of A to states 0 and 1 is isomorphic to the game of life. Counting the number of such cellular automata is equivalent to counting the number of local rules that can be chosen under such conditions. First, if a box is in state 2, it remains in this state. There is therefore no choice. So let's take the example of a box in state 0 or 1 (two possible choices). If all it's neighborhood consists of boxes in state 0 or 1, another time we have no choice because the evolution will be governed by the rule of the game of life. Only neighborhoods with at least one cell in state 2 are interesting. It is therefore necessary to choose k cells out of 8 that will be in state 2 with $k \in [1, 8]$ which give $\binom{8}{k}$ possible choices.

With the remaining 8-*k* cells there is a choice between cell in state 0 or cell in state 1 that is 2^{8-k} choices. Finally, there is:

$$2 \cdot \sum_{k=1}^{8} 2^{8-k} = 2 \cdot (3^8 - 2^8) = 12610$$

patterns for which the next state of the cell is not yet defined.

For each of these patterns we have the choice between state 0, state 1 or state 2 which give 3^{12610} possible rules.

Remark 1 (*scientific notation*) $3^{12610} = 3{,}16 \times 10^{6016}$.

New local transition rules. In this part we will give the rules we have chosen but especially how we got them and in what interest.

Motivation and approach

The first idea was to modify the network by adding frozen cells called "walls" in order to find some basic optical results. Among them are the laws of reflection of Snell Descartes. The light rays represent the information (modeled by gliders), in contact with a diopter (the walls) they are reflected and refracted. Only reflection has been retained since the first idea of obtaining an analogy with optics has been replaced by the desire to implement logic gates. The goal is to obtain, compared to what has already been achieved, different results: simpler and more practical to use (see part 5).

From this objective we have therefore looked for rules that allow the gliders to bounce on the walls. At first, we focused on the study of the bounce on a horizontal wall of a glider moving down and to the right (fig. 13).

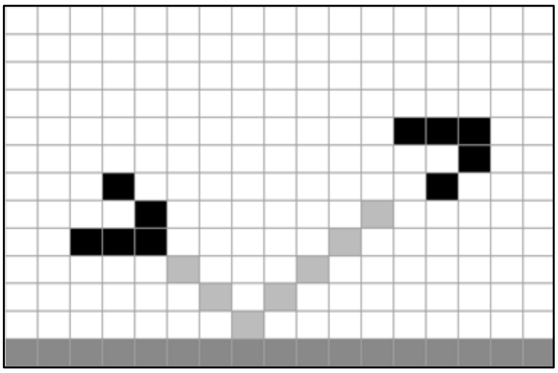


Fig. 13. Wanted trajectory (in weak grey) of a glider (in black) before (on the left) and after (on the right) interaction

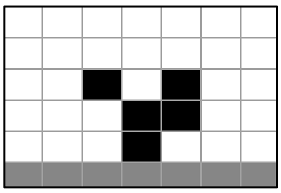


Fig. 14. First contact between a wall and a glider moving down right

There are too many different rules (cf property 1) to look into all of them one by one. By observing all possible configurations of a glider moving down and to the right on the fig. 7, we can notice that it will collide with a wall in the position described fig.14 (cf remark 2).

Therefore we need to know only a tiny part of the rules to calculate the evolution of this pattern. In our example only the four configurations shown in fig. 15 are useful. Indeed, we assume that a dead cell with three walls below and dead cells around (see fig. 16) remains dead.

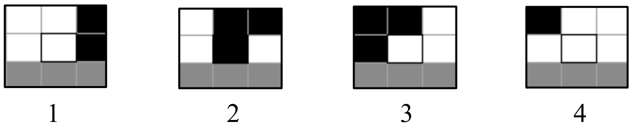


Fig. 15. The four configurations in which we need to give the next state of the red framed cell in order to have the next generation of the pattern given fig. 14

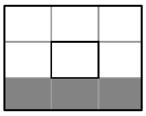


Fig. 16. Dead cell resting on a wall (bottom) surrounded by dead cells

It is therefore necessary to determine the next state of the red framed box for each of these four configurations. In each case two choices are possible: alive state or dead state. There is no creation of walls and the walls are in a permanent state.

Remark 2 (first contact with a wall). We previously stated that the first contact of a glider moving down and right was given by the configuration of the fig. 14 and therefore that only the four patterns shown fig. 15 were interesting. This is true only if the glider is not modified before coming into contact with the wall. For this we considered that a dead box with three walls below, two dead boxes (left and right) remained dead (see fig. 19 and 20 from configuration 10 to 17).

The approach chosen is to focus only on the configurations encountered (fig. 15) then to examine the different possible cases. In the next step, the red framed cell becomes either alive or dead. Thus, noting n the number of configurations ($n = 4$ in the first step), we have 2^n cases to consider. For each of them, we calculate the evolution of the glider in contact with the wall. For example, by choosing the red framed cells of patterns 1, 2, 3 and 4 of fig. 15 respectively become a living, dead, living and dead cell, we obtain the evolution described in the fig. 17.

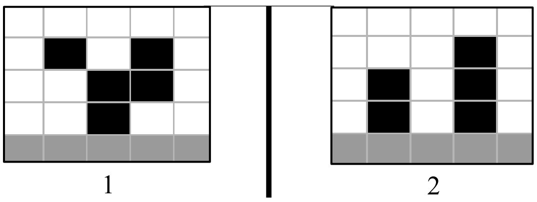


Fig. 17. Evolution from configuration 1 to configuration 2 with the rules described above

At this point we reiterate what we have just realized, which means that we only retain the necessary configurations to predict the evolution of the new pattern (knowing that the evolution of a cell in one of the four configurations of fig. 15 is already given). We then obtain three new configurations (fig. 18 for which it will be necessary, in each of the three cases, to choose if the red framed cell becomes alive or dead.

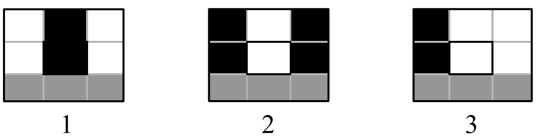


Fig. 18. The three configurations in which we need to give the next state of the framed cell in order to have the next generation of the pattern on the right of fig. 17

We continue until we obtain one of the four patterns of a glider moving up and right (this pattern should not touch the wall).

Rules obtained

Finally, we found a local rule involving only the evolution of 24 configurations allowing a glider moving down and right to bounce from above on a horizontal wall. This local rule is represented fig. 19.

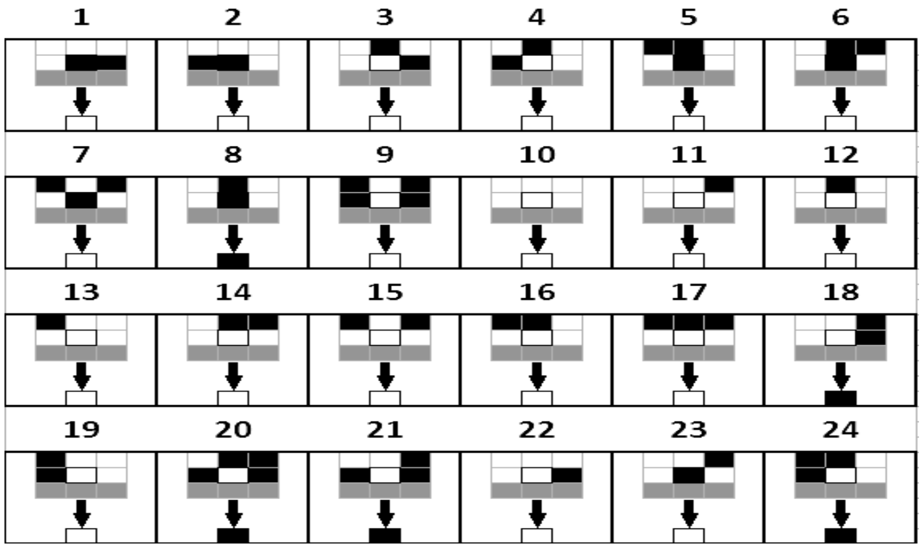


Fig. 19. Local rule allowing a glider moving down and right to bounce from above on a horizontal wall

By symmetry, one can easily find a local rule allowing a glider moving down and left to bounce from above on a horizontal wall. We have shown it fig. 20.

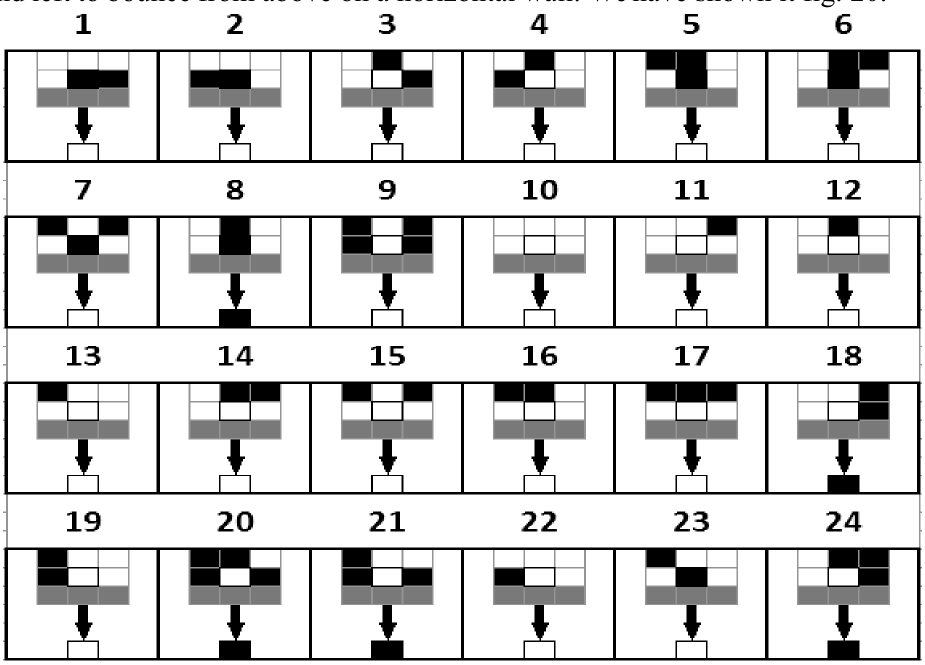


Fig. 20. Local rule allowing a glider moving down and left to bounce from above on a horizontal wall

We can notice that the evolution of the red framed cells having for neighborhood the patterns 18 and 19 of the two rules represented fig. 19 and fig. 20 are incompatible. In other words, it will not be possible from these results to find a local rule to bounce upwards on a horizontal wall at the same time a glider moving down right and a glider moving down left.

This does not mean that there is none, but we did not continue our research to find one since, as we will see (see part 5), we do not need such a rule for the implementation of logic gates.

Gliders can bounce from eight different ways:

- on a horizontal wall from the top (a glider moving down right and a glider moving down left);
- on a horizontal wall from the bottom (a glider moving up right and a glider moving up left);
- on a vertical wall from the right (a glider moving down left and a glider moving up left);
- on a vertical wall from the left (a glider moving down right and a glider moving up right). By symmetry and with the rules of figures 19 and 20, it is possible to obtain a single rule allowing four different types of rebounds among the eight described above (a choice to be made on the two possible for each dash because of the incompatibility).

In our case, we chose to keep the following rules:

- bounce from the top of a glider moving down right on a horizontal wall;
- bounce from the bottom of a glider moving up right on a horizontal wall;
- bounce from the right of a glider moving down left on a vertical wall;
- bounce from the left of a glider moving down right on a vertical wall.

Finally we obtain a local rule giving the evolution of 96 of the 12 610 possible configurations. This leaves many opportunities to work and obtain new results by keeping what has already been done. The rule giving the evolution of the 96 configurations is not explicitly given in this report. Indeed it is directly obtained by applying the appropriate symmetries of the rule represented fig. 19 or fig. 20.

LOGICAL GATES

The purpose of this part is to present the logical gates [13, 14] that have been implemented from game of life with non-regular network we have just seen. We will begin by recalling a few generalities about logical functions, then briefly recall what has already been done about the implementation of logic gates with the game of life before presenting our study. Finally we will give a striking comparison showing the difference between the complexity of the current implementation and the simplicity of the implementation carried out during this study.

Logic gates using Gosper glider guns

John Conway proved that the game of life was a universal cellular automaton [15]. This means that the game of life is able to simulate all calculations made by a computer. For more information, consult Nicolas Ollinger's [16] and Guillaume Theyssier's [17] thesis which deal with universality.

The universality of Conway's cellular automaton makes it possible, among other things, to generate prime numbers [18], to create a Turing machine [19] and even more surprisingly to create a game of life from the game of life itself. What will interest us here is the implementation of logic gates.

As we saw in the section on gliders (part 2), these can carry information. It is for this reason that we find them without exception in all the applications that we have just mentioned and the implementation of logic gates does not deviate from the rule. Specifically, the structure that appears in each of these applications is the glider gun (see fig. 21). The latter makes it possible to continuously gene- rate gliders, which makes it an extremely interesting pattern. Bill Gosper is an American computer scientist who, by introducing this glider gun, at the same time proved the conjecture of Conway asserting that there is a pattern whose number of living cells increase all the time.

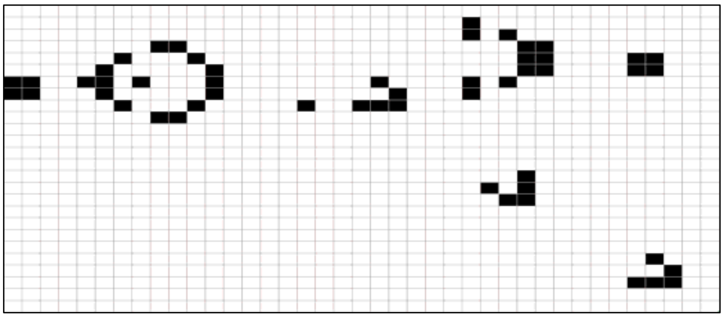


Fig. 21. Glider guns

Currently the implementation of logic gates is based on the combination of several glider guns whose gliders interact with each other to finally let or not pass a glider beam. Thus the value at the entrance or exit is 1 if there is a beam of gliders otherwise it is 0. This implementation is difficult and tedious, we will not detail it here since it is very well explained by Jean-Philippe Renard [11].

Implementation of logic gates in a non-regular network

As far as we are concerned, the implementation of logic gates we have made is based on three points. First of all the information is no longer represented by a glider beam as described above but by a single glider. We no longer need to resort to glider guns, which is a big novelty. Then we set up a particular network with "walls". Each logical gate is a particular configuration of space, a feature that is exploited by bouncing the gliders wisely as described in part 4. Finally, the method relies on collisions between gliders. And especially the two collisions that we analyzed in part 2.

Generally the logic gates set up have two ducts at the top (representing the two inputs). A glider in the conduit means that the entry is at 1 otherwise it is at 0 (see fig. 22). And a conduit down (representing the exit). The particular configuration of the rest of the network will allow or not to obtain a glider in the lower conduit depending on the nature of the logic gate.

Eeach of the four logic gates have been implemented. Namely: the NOT gate, the OR gate, the AND gate and the XOR gate.

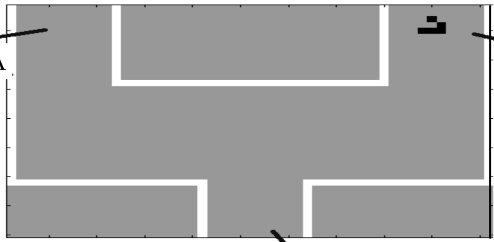


Fig. 22. Logic gate with two inputs $A = 0$ and $B = 1$

CONCLUSION

The previous work provides a significant improvement in what has been done so far. Indeed, the consideration of a variant of the game of life with a non-regular network allowed us to introduce new local rules near irregularities. These rules were chosen in such a way as to be able to obtain a particular property: the rebound of the gliders on a wall. From this specificity, it is then possible to implement logic gates much more intuitive and much easier to use than the logic gates that have been created so far.

Moreover, the large number of rules that can be chosen and the networks that can be considered gives hope that many interesting results can be obtained by deepening the subject. This study therefore provides an innovative result but it also opens up new and interesting perspectives.

REFERENCES

1. *EMERGENCE*. — Available at: https://en.wikipedia.org/wiki/emergence . 2019
2. *Delahaye J.-P.* Le royaume du jeu de la vie / J.-P. Delahaye // Pour la Science. — 2009. — N. 378. — P. 86–91.
3. *Wolfram S.* New kind of science / S. Wolfram // Wolfram Media Inc., USA. — 2002.
4. *Illiachinski A.* Cellular Automata. A Discrete Universe / A. Illiachinski. — Singapore: World Scientific Publishing, 2001.
5. *Delahaye J.P.* L'sutomata des chifferes / J.P. Delahaye // Pour la Science. — 2010. — N. 394. — P. 80–85.
6. *Wuensche A.* Discrete Dynamics Lab. www.ddlab.com/ 2016
7. *Goldengorin B.* Some applications and prospects of cellular automata in tra@c problems Cellular Automata / B. Goldengorin, A. Makarenko, N. Smilianec. — 2006.
8. *Vlassopoulos N.* An FPGA design for the stochastic Greenberg-Hastings cellular automata / N. Vlassopoulos, N. Fates, H. Berry, B. Girau // Proc. Int. Conf. on High Performing Computing &Simulation 2010. — P. 565–574.
9. *Sipper M.* Co-evolving non-uniform cellular automata to perform computation / M.Sipper // PhysicaD. — 1990. — Vol. 92. — P. 193–208.
10. *Fates N.* A Guided Tour of Asynchronous Cellular Automata / N. Fates. — arXiv: 1406.0792v2. 2014. — 33 p.
11. *Rennard J.-P.* Implementation of logical functions in the Game of Life / J.-P. Rennard // Collision-Based Computing, ed. A. Adamatszky, Springer, London, 2002. — P. 491–512.

12. *Faccetti L.* 'Game of Life' with Modifications: Non-regular Space, Different Rules and Many Hyerarchical Levels / L. Faccetti, A. Makarenko // Int. J. Information Content & Processing. — 2017. — Vol. 4, N. 1. — P. 21–50.
13. *Hopcroft J.E.* Introduction to Automata Theory, Languages and Computation / J.E. Hopcroft, R. Mortwani, J.D. Ullman // Pearson, 2006.
14. *Linz P.* An Introductin to Formal Languages and Automat 4^{th} Edition / P. Linz. — Jones&Parlett Publishers, 2006.
15. *Berlekamp E.* Winning Ways for your Mathematical Plays / E. Berlekamp, J. Conway, R. Guy. — 1982. — Vol. 2, chapter 25 What is Life. Academic Press.
16. *Ollinger N.* Automates cellulaires: structures / N. Ollinger // LIP, ENS Lyon PhD Thesis. — 2002. — 90 p.
17. *Theyssier G.* Automates cellulaires: un modèle de complexities / G. Theyssier // PhD Thesis, ENS LYON, Lyon, 2005. — 178 p.
18. *Wolfram S.* Random sequences generation by cellular automata / S. Wolfram // Applied Mathematics. — 1986. — Vol. 7. — P. 123–169.
19. *Rendell P.* A Turing Machine In Conway's Game Life. In: Designing Beauty. The Art of Cellular Automata, Emergence? Complexity and Computation. Eds. Adamatzky A., Martines G. Springer, Cham, 2016. — P. 149–154.

Received 27.01.2019

From the Editorial Board: the article corresponds completely to submitted manuscript.