WOLFRAM'S CELLULAR AUTOMATA

Introduction

Cellular automata (termed CA from now on) are an idealization of a physical system composed of a discrete n-dimensional lattice of homogenous, simple units, the *atoms* or *cells* (different CA might have different cell shapes: triangles, squares, hexagons…). Each cell can be in a finite set of states (physical quantities take only a finite set of values). In this system, both space and time are discrete:

- Each cell's behavior is only affected by cells close to it (its "neighbors"). There are, therefore, no actions at a distance.
- Each cell only updates its current state according to a deterministic transition function at each time step.

The transition rule can be represented as in *Figure 1*. The top row of each square represents a possible neighbor configuration, and the lower row (the cell alone) represents the state of the middle cell in the following time-step.

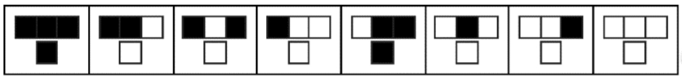

Figure 1

As we can see, there are $2^3 = 8$ possible configurations for a cell and its two immediate neighbors (the number of states of each cell raised to $2 * n + 1$, where n is the radius of neighbors, 1 in this case).

The rule defining the cellular automaton must specify the resulting state for each of these possibilities; the rule shown on *Figure* 1 can be represented as follows, assuming that white cells are represented as 0 and black cells are represented as 1:

111	110	101	100	011	010	001	000
1	0	0	0	1	0	0	0

Since there are 8 possible neighborhood configurations, and two possible states for each cell, there are $2^8 = 256$ possible rules. Stephen Wolfram proposed a scheme, known as the Wolfram code, to assign each rule a number from 0 to 255 which has become standard. To know the number of each rule, we just have to take the binary number formed by the resulting state of the cells in each possible neighborhood configuration (in the same order shown in the table above) and convert it to decimal. The rule represented on *Figure 4* forms the binary number 10001001, which is the number 90. Therefore, this is rule 137.

Starting with random initial conditions, Wolfram went on to observe the behavior of each rule in many simulations. As a result, he was able to classify the qualitative behavior of each rule in one of four distinct classes:

1. Class 1 rules *(Figure 6*) quickly produce uniform configurations, with all cells stably ending up with the same value. Any randomness in the initial pattern disappears.
2. Class 2 rules *(Figure 7*) produce a uniform final pattern, or a cycling between final patterns, depending on the initial configurations. Local changes to the initial pattern tend to remain local.
3. Class 3 *(Figure 8*) rules produce random-looking configurations, although some regular patterns and structures may be present. Any stable structures that appear are quickly destroyed by the surrounding noise. Local changes to the initial pattern tend to spread indefinitely.
4. Class 4 *(Figure 9*) rules lead to complex patterns and structures propagating locally in the lattice. Class 2 type stable or oscillating structures may be the eventual outcome, but the number of steps required to reach this state may be very large, even when the initial pattern is relatively simple. Local changes to the initial pattern may spread indefinitely.

On this exercise, I will study the dynamics of different rules. To do this, I have created a script in R that, giving it the initial conditions and the decimal number of the rule, plots the automata (the cells on the first and last column are considered neighbors).

Mi script generates a matrix composed of TRUE (1) and FALSE (0); to represent this matrix in a way that TRUEs are painted as black and FALSEs as white, I adapted the code from https://stackoverflow.com/questions/28035831/how-to-build-a-crossword-like-plot-for-a-boolean-matrix.

Rule 73

On *Figures 1* to *4* are shown the automaton with 256 (left) and 257 (right) cells with the following initial conditions:

- *Figure 1*: a single black cell in the middle of the array.
- *Figure 2*: Half black cells and half white cells at randomly chosen positions.
- *Figure 3*: 25% of black cells and 75% of white cells (approximately).
- *Figure 4*: 90% of black cells and 10% of white cells (approximately).

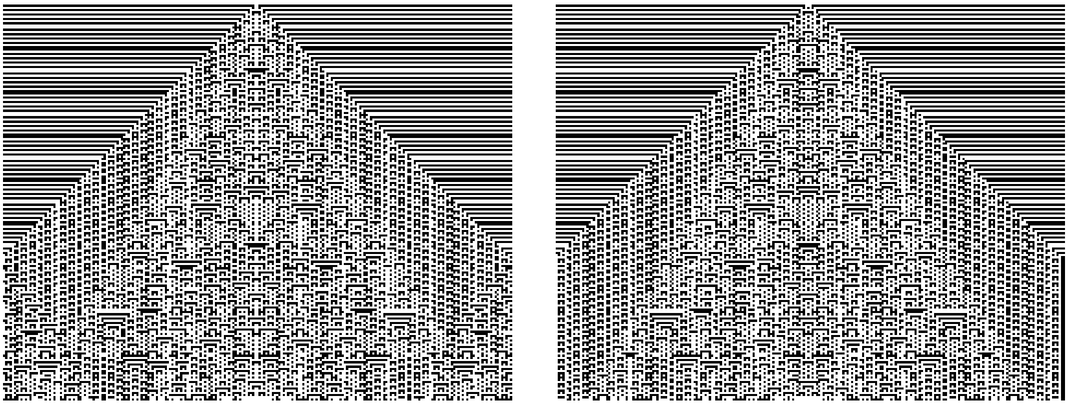

Figure 1

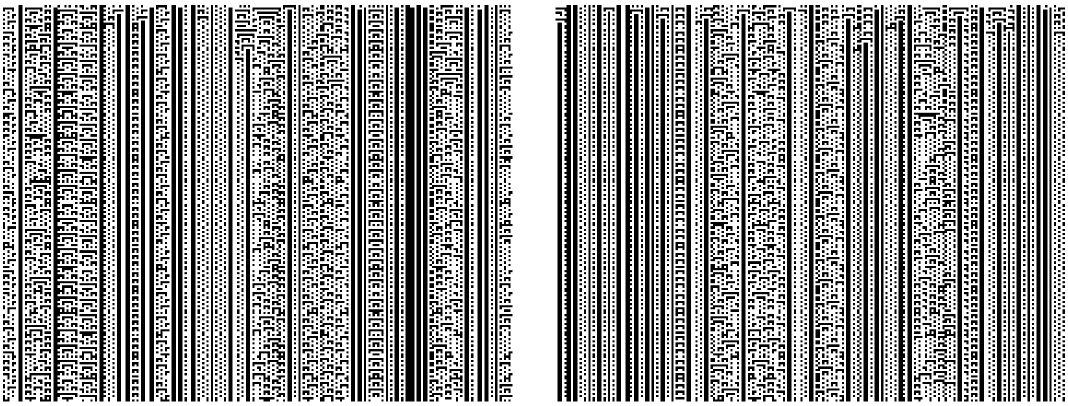

Figure 2

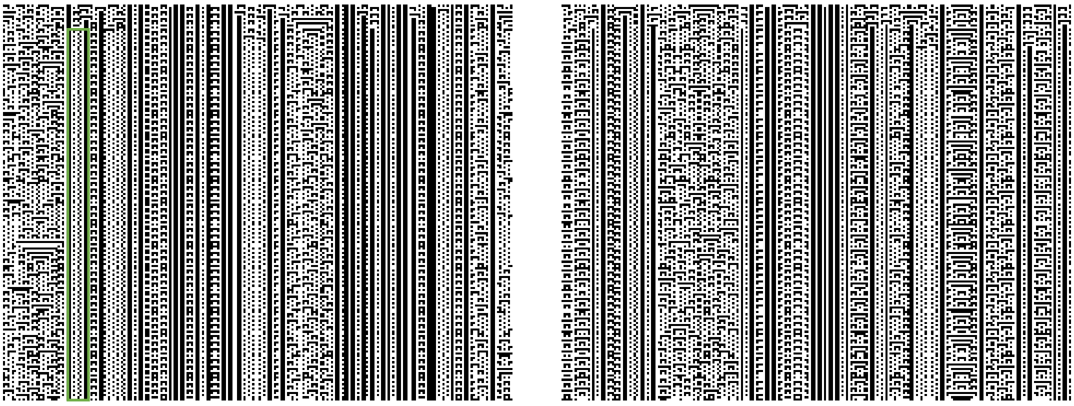

Figure 3

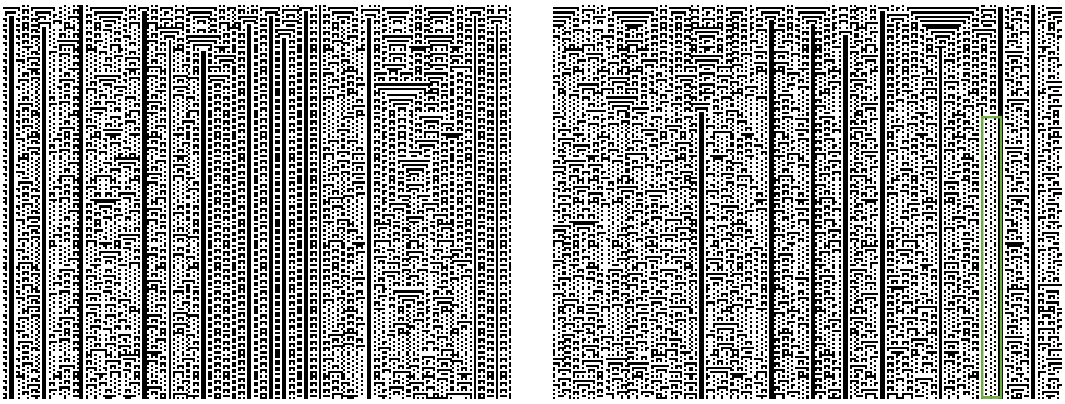

Figure 4

This rule forms complex patterns that do not disappear, such as the ones highlighted in *Figure 3* and *Figure 4*. From random initial conditions, rule 73 is a class 2 rule. Other complex patterns can be seen, as the one shown in *Figure 5* (taken from https://www.wolframalpha.com/input/?i=rule+73). On Wikipedia, we see that any time there are two consecutive 1s surrounded by 0s, this feature is preserved in succeeding generations. As we can see in our *Figures*, this is the cause for the black “walls” that we can repeatedly see. This is what makes this rule class 2.

Figure 5

On the other side, when we look at *Figure 1*, this seems to be a class 4 rule, with local propagation of complex patterns and structures. It also seems to be a symmetric rule.

Rule 136

On *Figures 6* to *8* are shown the automaton with 256 (left) and 257 (right) cells with the following initial conditions:

- *Figure 6*: Half black cells and half white cells at randomly chosen positions.
- *Figure 7*: 25% of black cells and 75% of white cells (approximately).
- *Figure 8*: 90% of black cells and 10% of white cells (approximately).

The figures showing the initial condition of just one cell black are not shown, because all the other cells of the plot are white.

Figure 6

Figure 7

Figure 8

This is clearly a class 1 rule, in which any set of initial conditions (even the one with 90% black cells) quickly leads to a uniform configuration in which all cells are white.

Rule 184

On *Figures 9* to *12* are shown the automaton with 256 (left) and 257 (right) cells with the following initial conditions:

- *Figure 9*: a single black cell in the middle of the array.
- *Figure 10*: Half black cells and half white cells at randomly chosen positions.
- *Figure 11*: 25% of black cells and 75% of white cells (approximately).
- *Figure 12*: 90% of black cells and 10% of white cells (approximately).

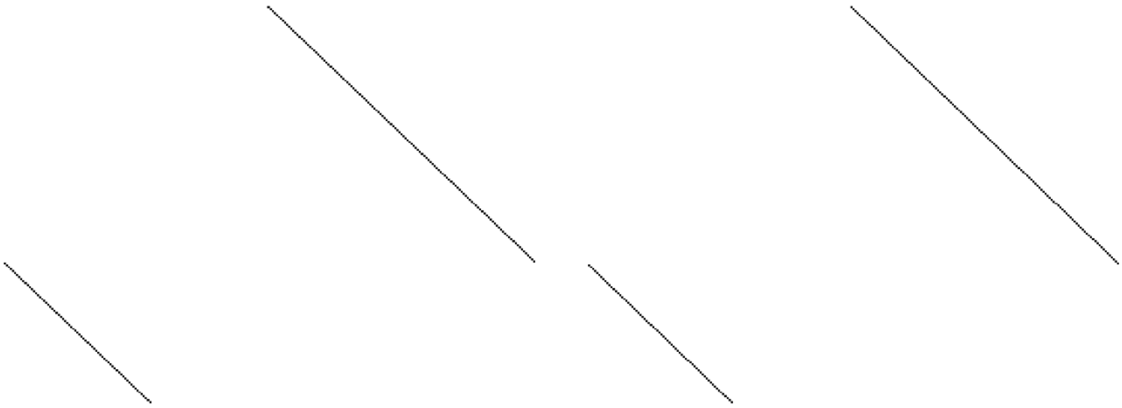

Figure 9

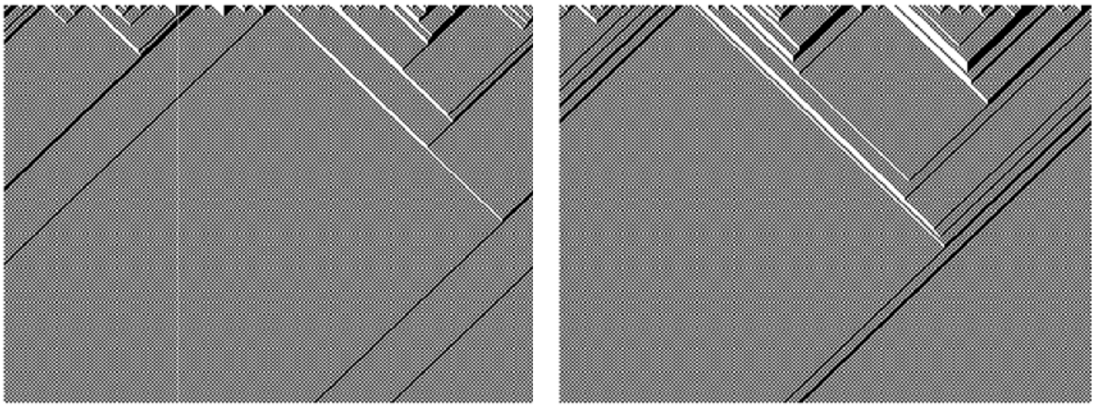

Figure 10

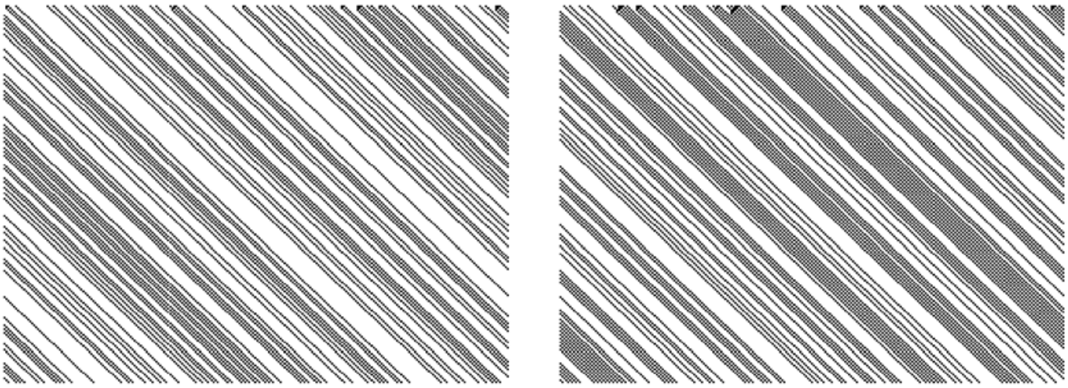

Figure 11

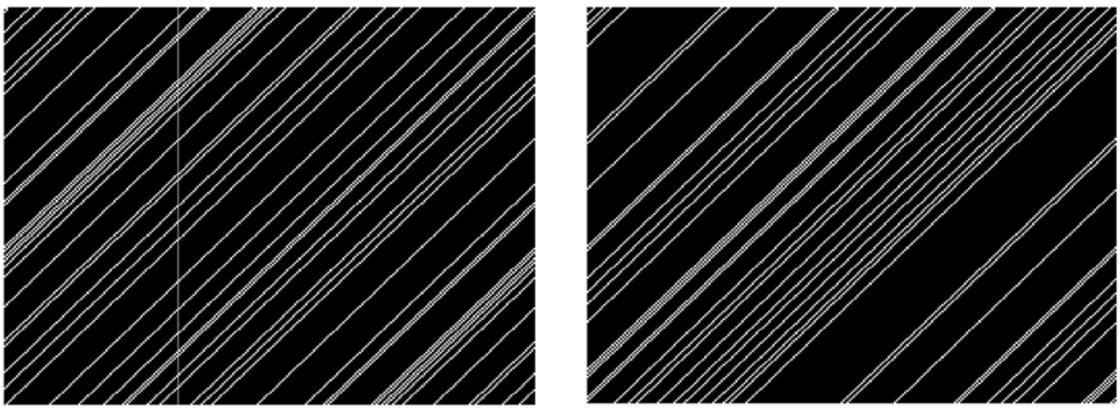

Figure 12

Rule 184 also is a class 2 rule, this time much more clearly than rule 73, since it produces uniform patterns.

This is a rule with many applications, due to two important properties of its dynamics:

- For any finite set of cells with periodic boundary conditions, the number of 1s and the number of 0s in a pattern remains invariant throughout the pattern's evolution.

- This rule is not right-left symmetric, but it has a different symmetry: reversing left and right and at the same time swapping the roles of the 0 and 1 symbols produces a cellular automaton with the same update rule.

This rule can be used, for example, to solve the majority problem, which is the problem of constructing a CA that, when run on any finite set of cells, can compute the value held by a majority of its cells. It can also be used as a simple model for traffic flow in a single lane of a highway, and is used as a base for more complex CA models of traffic flow.

We have seen class 1 and class 2 rules, so I will add a rule of class 3 and a rule of class 4.

Rule 90

On *Figures 13* to *16* are shown the automaton with 256 (left) and 257 (right) cells with the following initial conditions:

- *Figure 13*: a single black cell in the middle of the array.
- *Figure 14*: Half black cells and half white cells at randomly chosen positions.
- *Figure 15*: 25% of black cells and 75% of white cells (approximately).
- *Figure 16*: 90% of black cells and 10% of white cells (approximately).

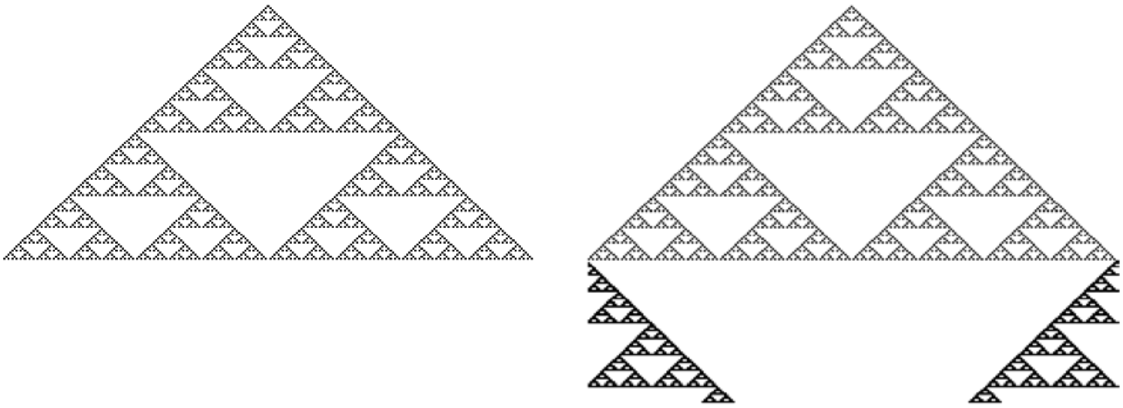

Figure 13

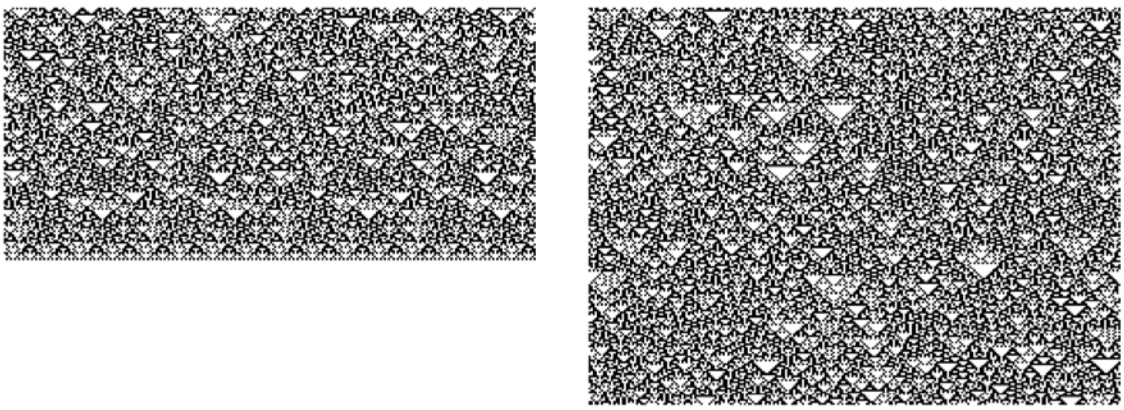

Figure 14

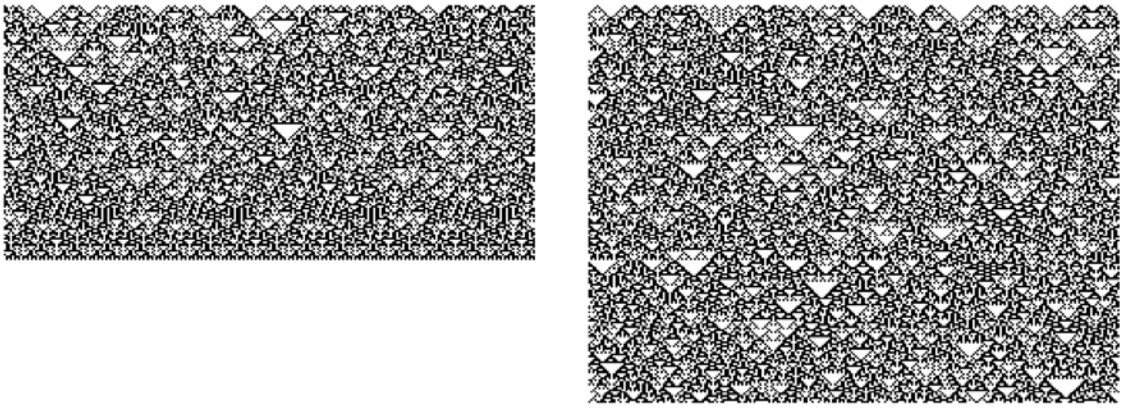

Figure 15

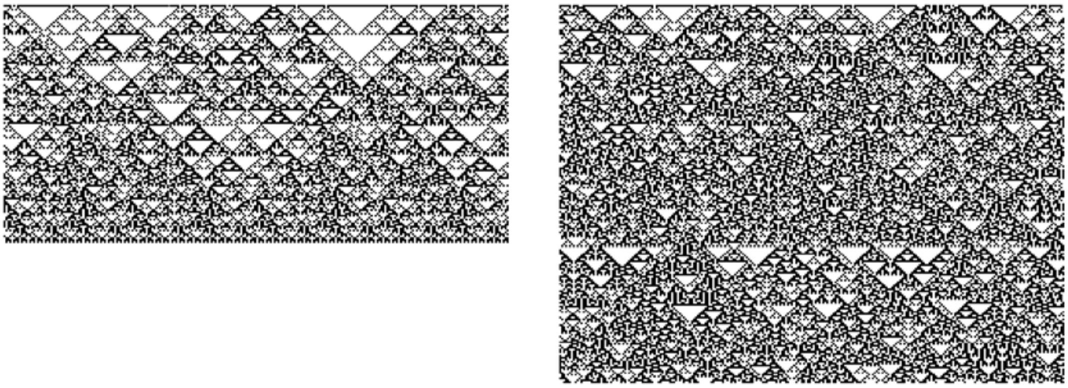

Figure 16

The first thing noteworthy is the fact than, when the number of cells is 256, it always leads to a white pattern. On the other side, when the number of cells is 257, we get a class 3 rule.

Rule 90, when started from a single live cell, forms a Sierpinski triangle, and the behavior of any other configuration can be explained as a superposition of copies of this pattern.

Rule 137

On *Figures 17* to *20* are shown the automaton with 256 (left) and 257 (right) cells with the following initial conditions:

- *Figure 17*: a single black cell in the middle of the array.
- *Figure 18*: Half black cells and half white cells at randomly chosen positions.
- *Figure 19*: 25% of black cells and 75% of white cells (approximately).
- *Figure 20*: 90% of black cells and 10% of white cells (approximately).

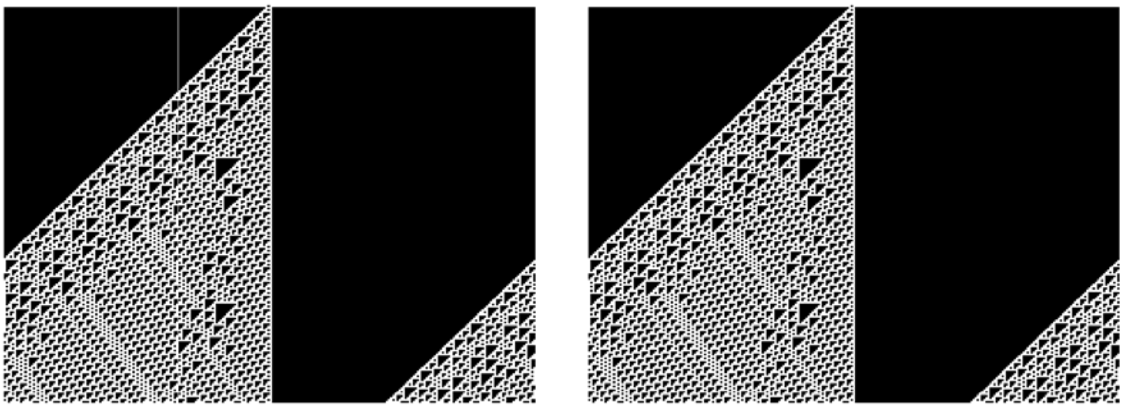

Figure 17

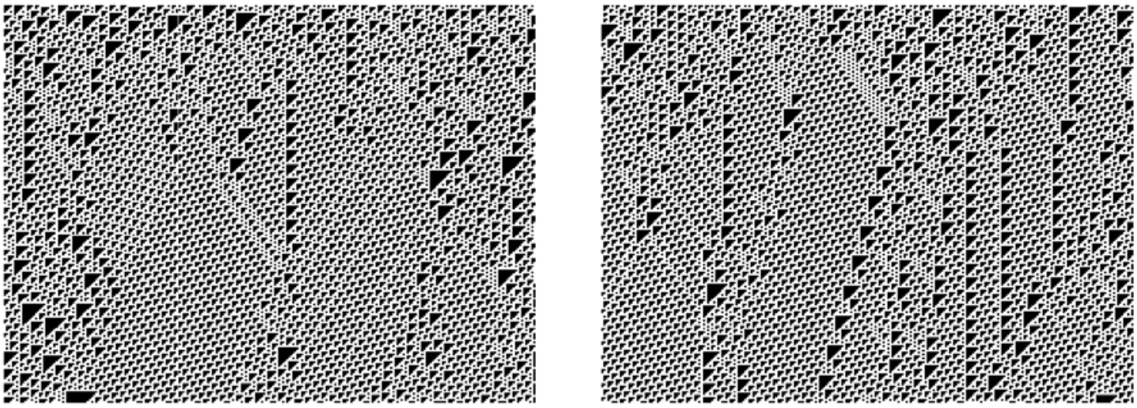

Figure 18

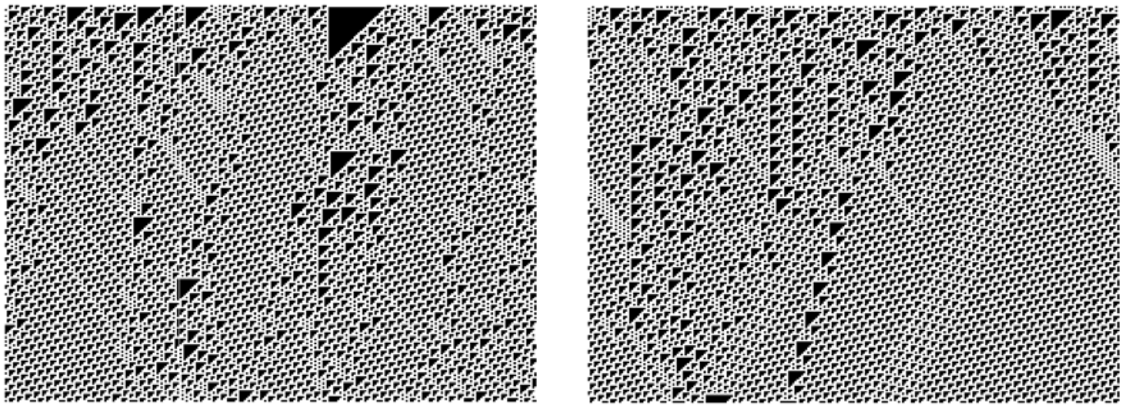

Figure 19

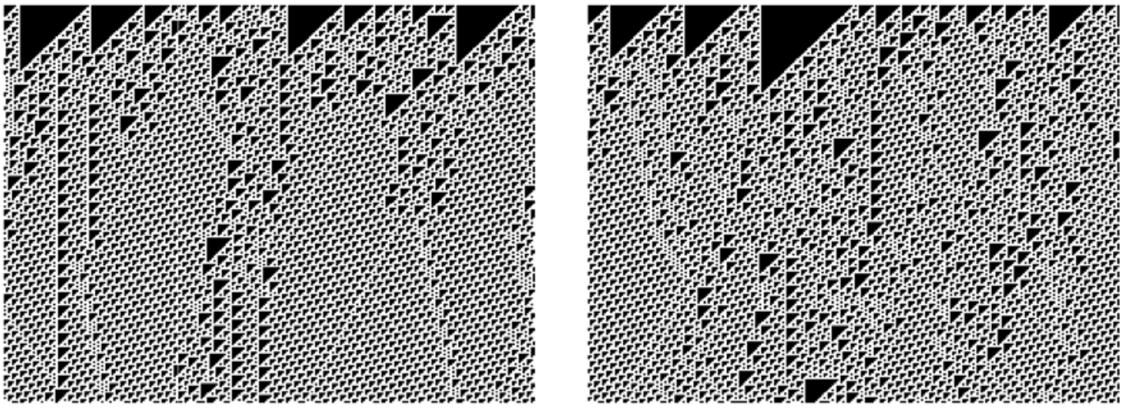

Figure 20

Finally, rule 137 is our class 4 rule. It forms complex patterns and structures that propagate locally. As a rule of the class 4, it presents both regular patterns and chaotic behavior, which allows it to compute universally.