The Game of Life and Other Cellular Automata

Mi Yu/Adam Reevesman

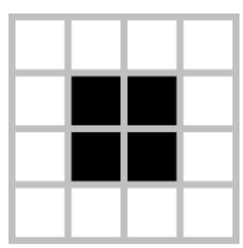
January 28, 2015

The Game Of Life

Last class we talked about Cellular Automata, Elementary Cellular Automata, Rule 110, and the classifications of Cellular Automata. This week, we will look at a special form of Cellular Automata— The Game of Life. This game was devised by the British mathematician, John Horton Conway in 1970. This game is a zero-player game, which means that we only need one initial input, and then we can observe how the configuration evolves. There are many interest patterns about The Game of Life. This week, we will take a glance of the beauty of The Game of Life.
First of all, as a quick recap of the first week's notes, let me define the rules of The Game of Life. People play Life[1] on a two-dimensional grid of square cells, each of which is in one of two possible states, alive or dead. The square will interact with its eight neighbors[2]. The rules for each update are the followings:

1. Any live cell with fewer than two live neighbors dies
2. Any live cell with two or three live neighbors lives
3. Any live cell with more than three live neighbors dies
4. Any dead cell with exactly three live neighbors becomes alive

Now, since we have the rules, let us play the game. If we have the following initial state:



Trying to update this state, we find out that we will always get the initial state as the output. We call this kind of shape still lives. The example we just did is called a block. There are other beautiful shapes of still lives. The common ones are the followings:

[1]It is another name for The Game of Life
[2]It is called Moore Neighborhood from the last class

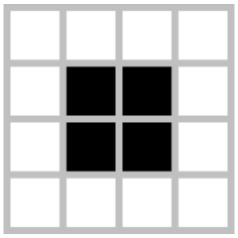


Figure 1: Block

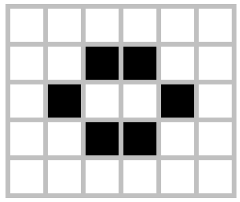


Figure 2: Beehive

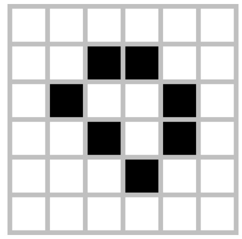


Figure 3: Loaf

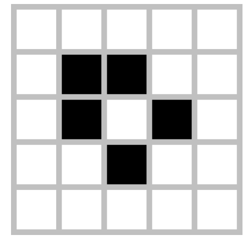
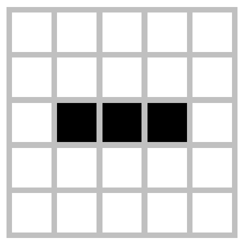
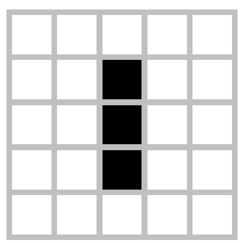


Figure 4: Boat

Now let us look at other different kinds of shapes. If we have the following initial state:



If we update this graph once, we will have the following graph:



However, if we update this graph again, we will get the original graph. This kind of graphs, which have repeating patterns, are oscillators. Notice that still lives are oscillators because they also have repeating patterns. We define an oscillator's period as the number of updates it needs to change back to the initial state. We know that the period for all still lives is one. The example we just did is period 2. Most naturally occurring oscillators are period 2, such as blinker, toad, and beacon shown below. Pulsar is period 3 oscillator. There also are more interesting oscillators with period 4, 8, 14, 15, and 30. Some shapes even take thousands of steps to have repeating patterns. For example, "Acorn" takes 5206 generations to stabilize. Here are some common oscillators in Life:

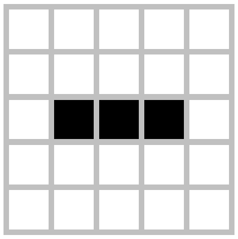


Figure 5: Blinker at the initial state

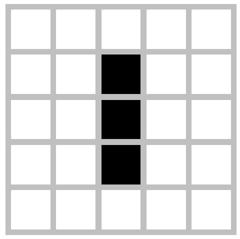


Figure 6: Blinker after one update

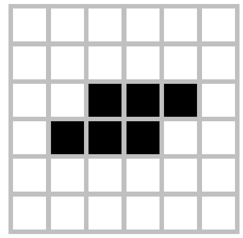


Figure 7: Toad at the initial state

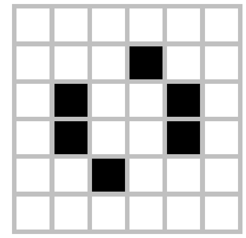
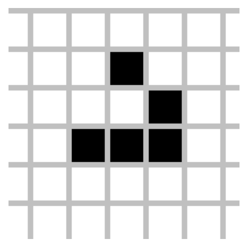


Figure 8: Toad after one update

Let us look at the third type of shapes. If we have the following initial state:



If we update this initial states, we will have the following graphs:

.png

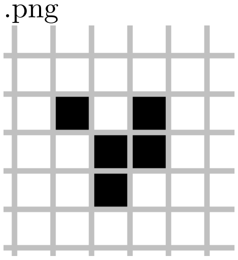


Figure 9: After one update

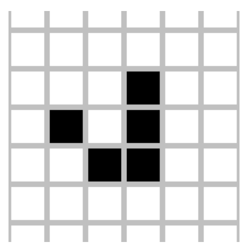


Figure 10: After two updates

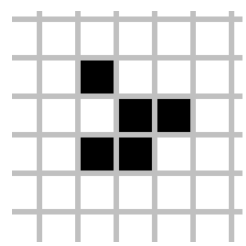


Figure 11: After three updates

After three updates, we will find out that the graph returns to its initial state. However, this graph is different from the oscillators because although it returns to its initial state, the entire graph travels along the grid for some distances. In Life, a finite pattern is called a spaceship if it reappears after a certain number of generations in the same orientations but in a different position. The smallest such number of generations is called the period of the spaceship. The example we just did is called the glider. There are also many other examples of spaceships. Another common spaceship is Lightweight spaceship[3], as the following:

[3] Shortly denoted as LWSS

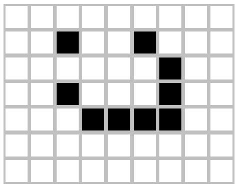


Figure 12: LWSS at the initial state

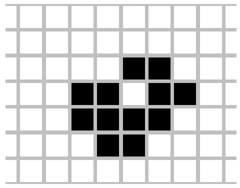


Figure 13: LWSS after one update

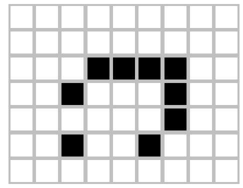


Figure 14: LWSS after two updates

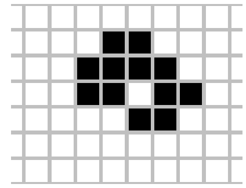


Figure 15: LWSS after three updates

We know that all spaceships are moving through the grids. The natural thing we want to know next is that how fast the spaceships move. In the next section of this write up, we will define the speed for a spaceship and prove the speed of light in Life, just like Physics in real life.

Speed

In Conway's Game of Life, we refer to a pattern that returns to its initial state but ends up in a new position as a **spaceship**. It returns to its initial state after a certain number of generations, otherwise known as its **period**. One question that we may want to ask ourselves is if there a is a method to describe how long it takes a spaceship to return to its initial state, and can we tell where it will end up. What we are asking for here is a notion of **speed**. Speed describes the relationship between the number of generations required to make a spaceship appear in a new position and how far that new position is from the old one. We will denote the speed of a spaceship as S, where

$$\mathrm{S} = \frac{kc}{g}.$$

In this formula, we let k represent the number of cells that the glider moves in g generations, and c is the metaphorical speed of light (1 cell per generation).

Let's look at the Lightweight Spaceship, referenced in the preceding pages. The original pattern moves exactly 2 cells to the right in exactly 4 generations. This means that for the lightweight spaceship,

$$S = \frac{2c}{4}$$
$$\Rightarrow S = \frac{c}{2}.$$

We refer to a spaceship moving up, down, to the right, or to the left as **orthogonal** motion. As a further example, look at the Glider. The Glider moves one 1 cell diagonally every 4 generations. So, its speed is

$$S = \frac{c}{4}.$$

Note that the direction that the spaceship moves does not affect its speed.

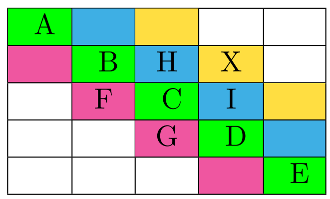
Speed Limits

So, we have shown examples of spaceships that move at a speed of $c/4$ diagonally, and $c/2$ orthogonally, but can we find any that move faster? In other words, we want to find the speed limit. The **speed limit** is the fastest speed that any spaceship can have given a set of rules. In this case, the rules are those of Conway's Game of Life. We can create other cellular automata with other sets of rules, but for now, we will only discuss the Game of Life.

Our first claim is that $c/4$ is the speed limit for any diagonally moving spaceship. The proof of this claim follows.

Proof. By Contradiction.

Look at the following grid.



Take any spaceship who's initial state consists of live cells only on or to the left of the green squares. We call this initial state generation 0.

Assume that there is a spaceship that has speed greater than $c/4$ diagonally. Then we must be able to make cell X come alive by generation 2. This is because if we can only get to the blue cells by generation 2, then it will take us two more generations to get to the yellow cells.

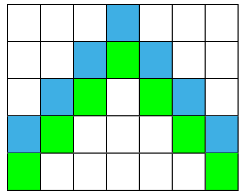
In order to make cell X live in generation 2, cells C, H, and I had to have been alive in generation 1, because X needs 3 live neighbors to come to life.

If C, H, and I were alive in generation 1, then they must have had 3 live neighbors in generation 0. The only cells that were alive in generation 0 were on or to the left of the green line, so the the cells that were alive in generation 0 must have been B, C, D, F, and G.

If this is the case, then in generation 0, cell C would have four live neighbors, so it must be dead in generation 1. This gives us a contradiction, because C had to be alive in generation 1 to make X alive in generation 2. Therefore, we know that we cannot make X come to life with only 2 generations. It follows that the diagonal speed limit is $c/4$. □

A similar argument can be used to show that the orthogonal speed limit is $c/2$.

Proof. Look at the following grid.
In the preceding proof, we showed that if the initial state of a spaceship consists of all of the



live cells on or below the green line, then it cannot be past the blue line after 2 generations. It follows that the orthogonal speed limit is $c/2$, because we can only move up by one cell every 2 generations. $\square$

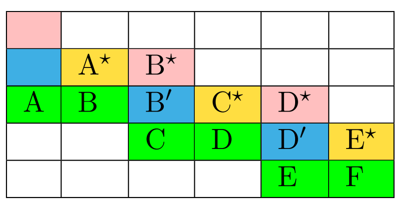
Now we know the speed limits when we play Conway's Game of Life, but what happens when we use other rules? One should not expect the same results. In the next section we will explore speed limits for other sets of rules.

Speed Limits For Other Rules

Imagine that we played our game with a slightly different set of rules. For example, what if we allowed cells to survive even if they had 3 or more neighbors. Our proofs in the last section would not apply, because we would not have the same contradiction of whether or not cell C is alive in generation 1. It is then plausible that we could attain faster speed limits.

We present the following claim: For any cellular automaton where birth in a cell occurs when there are 3 live neighbors, the speed limit is $c/3$ diagonally and $c/2$ orthogonally.

Proof. Consider the following grid.



Call generation 0 of the spaceship live cells that are only on or below the green line. Asumming that we can extend the green line to a larger grid, the blue cells are the only ones that can have 3 neighbors in generation 0 so they are the only ones that can come to life in generation 1.

We have named the cells B' and D' because they are exactly one cell to the right of B and D, respectively. Note that it took 1 generation to move the spaceship 1 cell in the x direction.

The yellow cells are the only new cells that can come to life in generation 2, because they had 3 live neighbors in generation 1. In generation 3, it is possible to bring the pink cells to life because they can have 3 live neighbors in generation 2. Note that $B^\star$ is a pink cell, and it is 1 cell above B' in the y direction, and located one diagonally from B. Also note that this took 2 more generations after bringing B' to life.

In total, moving a cell in our spaceship from B to $B^\star$ took 3 generations (and the same for D to $D^\star$), so the speed limit for the cellular automata that use the set of rules we described above is $c/3$ diagonally.

As for the orthogonal speed limit, we showed that it takes 2 generations to move from the green line to the blue and yellow line, so the speed limit is $c/2$. □

So, we have shown an orthogonal and diagonal speed limit for cellular automata in which birth occurs for exactly 3 neighbors. Though the Game of Life is an example of such cellular automata, its speed diagonal speed limit is $c/4$. This does not contradict our new result because it is not faster than this speed limit that applies to all cellular automata with the birth rule. $c/3$ is the maximum speed that a spaceship in any of the "birth at 3" rules can move.

We can use some observations that we made in the above proof to find general speed limits of certain spaceships; specifically, those that move a certain number of cells in the y direction for another number of cells that it moves in the x direction.

We noticed that it takes 1 generation to move 1 cell in the x direction, and 2 more generations to move 1 cell in the y direction. This means that for a spaceship to move over x and up y, it takes at least $(x + 2y)$ generations. Furthermore, when a spaceship moves in this fashion, we define the number of cells that it moves as the maximum of x and y. So, in general, the speed limit of this type of spaceship is:

$$S = \frac{\max(x, y)c}{x + 2y}.$$

As we have experienced, different cellular automata behave quite differently, and there are many rules that we know nothing about. However, when we look at what happens when we set a rule of birth at 3 neighbors, these are some of the results that we can achieve.

References

This information references Nathaniel Johnston's article: "Speed Limits in "B3" Life-Like Cellular Automata" and Wikipedia's pages on Spaceships and the Game of Life.