**The Complexity**

The way that Conway’s Game of Life can be represented by a Turing Machine is by having the grid be represented in a 2-D tape. The start of the game is generated randomly, and each subsequent screen is known as a “generation.” There are a few simple rules that determine what the next generation will look like that the Turing Machine must follow in order to proceed.

To start off, it is necessary for the TM to have two different 2-D tapes, one that holds the current state of the cells, and another one that is procedurally generating the next generation. It is also necessary for the TM machine to have a supplementary tape that keeps track of how many live neighbors the current cell has, as well as an instruction tape that tells it what to do per cell and how to handle alive or dead cells.

Because this all happens in one single cycle, the complexity of this algorithm is determined by the complexity it takes to generate a single generation. This solely depends on the size of the grid, which in this case we will refer to the variable N as the number of cells in the grid. If the grid is 10x10, then N is 100. To compute the total amount of actions, it is only necessary to see how many actions computing a single cell takes, then multiplying by N. For each cell, the read head must move Up, Right, Down, Down, Left, Left, Up, and Up to read each neighbor. On each move, the head must read the state of the cell and determine whether it shall increment the count of live neighbors or not. Finally, it must look read the state of the current cell and determine if it stays alive, dies, or revives. This comes up to about 3 actions per neighbor, and since each cell has 8 neighbors (excluding the ones around the edges), we can say that each cell of the new generation takes somewhere between 10-24 actions to compute. The majority of the cells will have the full 8 neighbors, so let’s round it to 20 actions per cell. This means that the total complexity of the computation of a single generation is about 20N. But since N can grow arbitrarily large, when determining asymptotic complexities, constants can be dropped from the analysis, which means that the complexity of the program is O(N), or linear. Even though there are other things that are being done, such as the drawing of the cells, they all depend on a constant amount of factors, which keeps the complexity linear.