Investigations of Cellular Automata Dynamics

Timothy Hunter-Kilmer
TJHSST Computer Systems Lab 2006-2007

March 27, 2007

Abstract

John Conway's Game of Life was the first cellular automaton, showing how simple rules can generate amazingly complex patterns. He designed a field filled with cells, each of which could be dead or alive, and devised three rules to govern how these cells changed from one step to the next. Any living cell that had two or three living neighbor cells, out of a total of eight, survived into the next generation; any dead cell that had three living neighbors was born. Conway chose these rules to provide some stability but also allow a great variety of changes; I will look at many different sets of rules, both in their own stability and in their interactions with other sets.

Keywords: cellular automaton, simulated evolution

1 Introduction

Conway's Game of Life provides a foundation for further examinations of cellular automata. In his simulation, one set of rules governed the whole field, and every cell in it; these rules were designed to limit stagnation and allow substantial growth. The trial field contains 14,400 cells, each of which stores its own private set of rules under which it operates; in the same spirit as Conway, I limited the individual cell to having three ways to survive, and three ways to be born. When a cell is dead, it inherits rules from its neighbors based on the numerical preponderance of each rule. It is then judged to live or die based on those new rules; when any cell is dead, its rules are cleared, and it begins the next generation fresh.

The main method of gathering information was at first simply testing the program; as I fiddled with the GUI, making sure everything worked properly, I noticed which rules came up most often. With this data, I ran systematic trials, testing two rulesets that I judged to be moderately successful, and began to assemble a list of what rules were the most prolific, survived the most, and had the most variations.

Each trial consisted of a simple, though lengthy, process; beginning with two previously selected sets of rules, approximately half the board was covered manually with each, resulting in equally-sized semi-occupied sections of one ruleset each. Every few hundred generations, I judged based on the colors on the field and the statistics in a separate panel whether one set had dominated the board, or if the trial was inconclusive, or if it needed more time. Whatever results I got, I recorded under the beginning rulesets.

With data on which sets work the best, I will be able to run those rules against each other and form a rough hierarchy based on characteristics such as variability, individual progress, and overall success.

2 Background

Cellular automata have largely been overlooked, many researchers preferring to focus on more complex multi-agent based models; the most notable exceptions are John Conway [1], who to a large extent created the field, and Stephen Wolfram [2], who examined Conway's ideas in fewer dimensions. Wolfram did much the same thing that I am, conceptually; taking the Game of Life, making large changes, and seeing what comes out.

There are two essential algorithms for this project: processing the whole field to go through a generation, and handling dead cells. The former is simply an exercise in looping over rows and columns, performing appropriate actions on each cell; the second can be implemented in several different ways. I chose to have each dead cell regenerate its rules at the beginning of each generation; if most of its neighbors survive with two living cells around them, the new cell will probably do the same thing, unless it mutates. Mutation is expressed as a probability with each cell that it will go against the majority position in retaining or discarding a rule when it is born; like the rulesets, mutation probabilities are passed down to neighboring cells.

3 Structure of the Program

The GUI allows me two main options for creating a cell: either click the button, and alter the rules manually, or use the checkbox at the bottom to set the ruleset that all new cells will have until that set is changed again. There are menus at the top, both to condense the different functions of the GUI into a much smaller space and to allow keyboard shortcuts. On the right side, there is a table showing how much of the board possesses each rule, expressed as percentages, and judged both from all the cells on the field, and all the living cells. Along with the graphic representation of each cell, using different colors for different sets of rules, this is the method for determining what data I can gather from each trial.

3.1 Writing the GUI

I made extensive use of the Java API in creating this GUI, as little of what I created is taught in this school. Menus and Menuitems provided a way to organize my different functions concisely and compactly, occupying little space on the panel but easily accessible. The various checkboxes form the heart of the rules-changing mechanism, without which running trials would be a great deal more difficult. The statistics table, whose data is gathered during the main generation process and put into a formal Table, is half of my basis for assessing each trial, and judging whether anything is going to change.

4 Results, Assessment

I don't yet have enough data to make any conclusions on the main goal of this project. However, there are a few things I have observed:

4.1 Observations

1. If one ruleset is a simple pair of triples (e.g. 123/234 or 234/123), and the other is a slight variation on it (e.g. 123/134 or 234/124), the original set will almost certainly dominate the entire field.

2. If one ruleset is a variation on a strong pair of triples, and the other is an original weaker pair of triples, there's no way to predict which will

cover the board.

3. The preliminary data, coming from trials with weak paired rulesets, is not a good indicator of the strength of the resulting rulesets. That is, if a certain set was the result of only a couple of preliminary trials out of a couple dozen, it still may well beat many of its opponents in subsequent trials.

References

[1] Martin Gardner, “MATHEMATICAL GAMES”,*Scientific American 223*, pp. 120-123, 1970.

[2] Stephen Wolfram, *Cellular Automata and Complexity*, 1994.

[3] ibid., *Theory and Applications of Cellular Automata*, World Scientific Publishing Co. Ltd., pp. 485-557, 1986.

[4] ibid., “Cellular Automata”,*Los Alamos Science 9* pp.2-21, 1983.

[5] N.H. Packard and Stephen Wolfram, “Two-Dimensional Cellular Automata”, *Journal of Statistical Physics 38*, pp.901-946 1985.