Investigations of Cellular Automata Dynamics

Timothy Hunter-Kilmer
TJHSST Computer Systems Lab 2006-2007

June 7, 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.

Based on this secondary data, I arranged it in as many meaningful ways as I could think of and looked for anything intersting.

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.

3.2 Data Analysis

Using a fairly simple Python program, I tallied the percentages in each trial in several different ways:

1. The margin of victory - how much more or less of the board the ruleset had
2. The margin of victory if positive
3. For each living and born rule, the margin of victory for all rulesets with that rule
4. For each living and born rule, the margin of victory if positive

5. For each living and born rule, the margin of victory against each other rule

6. For each living and born rule, that margin of victory if positive

4 Results, Assessment

The purpose of this project was to determine what the strongest ruleset with a maximum of three possibilities for being born or surviving was, based on data collected from running trials between various different rulesets. On that, there was indeed success, but there are far more substantial findings here, about the nature of the interactions among different areas of the simulation.

In analyzing my data, I discovered many interesting things. One of the most unfortunate parts is that a tool I was planning to use to gather conclusions would not in fact work; I wanted to examine the born and living rules in isolation from each other, but each trial is affected by the interaction between both sets, and thus any examination of a subset of the possible combinations of the different living and born rules would be incomplete. I did not realize that before beginning the trials, but even if I had, there is no way I could have tested every pairing of 7 living rules and 4 born rules against every other such pairing in less than a few years.

I also found that those different sets can function differently together; the most successful ruleset was 345/234, but the other two rulesets with 234 as the born rule were in 11th and 9th place. In turn, the worst ruleset was 123/234, but two of the other rulesets with 123 as the living rule were in the middle, and the last was 2nd place. I attribute this to three main causes: there were limited trials, since I only had enough time to run each combination twice, so the results I got might be different with more testing; I did not test every combination of the different rules, as I said before, so it's an incomplete view; perhaps most importantly, these rules work differently together than they do separately. In other words, if abc/def beats ghi/jkl by a significant margin, and ghi/jde beats ahi/bdj by a small percentage, those two results will not predict the outcome of testing ace/bik against bcg/efl.

Finally, I saw that small changes can produce radically different results: abc/def may do much better than abc/deg. For example, 345/234 was the most successful ruleset overall, but 345/134 was on the low end of the middle. However, the change does not necessarily have to affect the viability greatly:

while 234/123 is in 3rd place, 234/134 is in 8th, and 234/124 is 4th. It seems to be an essentially unpredictable system, in that similarity of ruleset does not guarantee similarity of performance, and lack of the former will not necessitate lack of the latter.

These results confirm the unstable nature of this design of cellular automata, in that different changes will produce unpredictably different behavior. For example, the combination of two successful rules may or may not fare well against other rulesets, though it should if those characterstics hold true.

4.1 Success of Project

The desired data was obtained, but far more important than that were the more general conclusions about the nature of cellular automata interaction. In this field, any change is going to have effects throughout the simulation; the especially difficult part is that these effects are largely unpredictable, and may be tiny or major. Furthermore, I do not know how the composition of my data affects my results: is it fair to judge a living rule's success against another rule when they are included in a different number of rulesets? Is it any more accurate to take that into account? Dividing the total margin of victory by the number of rulesets each individually has gives the average margin per trial, but does not, and cannot, consider the influence of the success of the born rules in each case. If, for example, abc is far more effective than def, but ghi is roughly the same as jkl, abc/ghi's victory will be recorded against def and jkl equally even though the two parts had an unequal role in the result.

4.1.1 General Advice

As I said before, the most important conclusions I drew did not have much to do with the stated goal of my project: that cellular automata, though fairly simple, still have intimately interconnected parts, and changing one is likely to have unforeseen consequences in another. Thus, researchers need to keep in mind that what they find out along the way to the desired data may well be much more significant than the final data itself.

4.2 Testing

The testing I did was really essential to the development of the project itself; as I corrected bugs in the program and expanded it, my focus changed. I picked different rulesets that I thought would do okay, and ran them against each other to make sure that the field was updating properly; that gave me the data that formed the second stage of analysis. During that second stage, I ran every ruleset against every other ruleset twice, and recorded what percentage of the board was occupied by cells with the respective rulesets, which was the basis for my final conclusions.

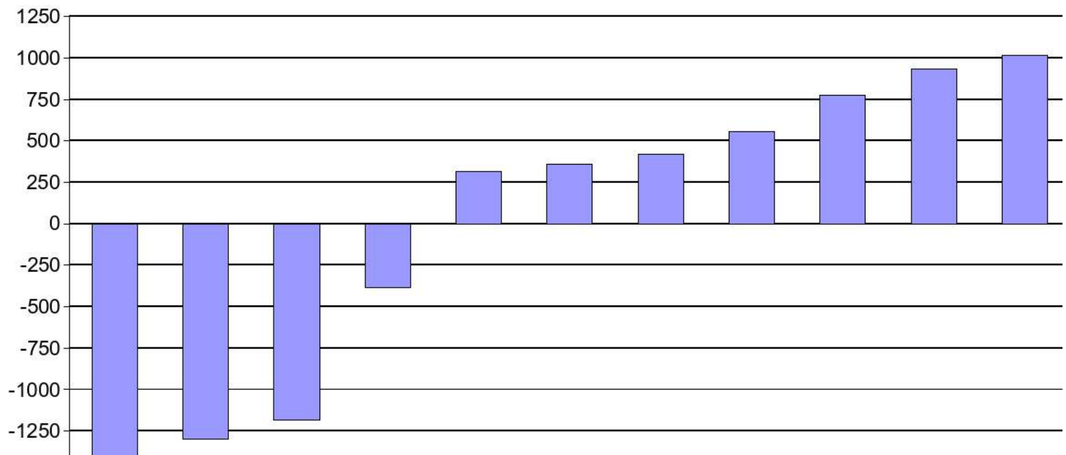
4.3 Recommendations

Any further exploration would most likely need to have a smaller but more thorough set of rulesets; for example, testing every combination of 3 born rules and 4 living rules would involve about the same number of rulesets, but allow for analysis of individual rules in isolation, as I was unable to do effectively in this project. Also, other projects might wish to explore a few things that I did not do to try to reduce the time necessary; I did not run the trials automatically, instead using my own judgement to assess whether the trial was over. Having the program analyze that could reduce the time it takes to run the trials. Also, writing a method to fill the field with the two rulesets may be more reliable and faster than doing it manually.

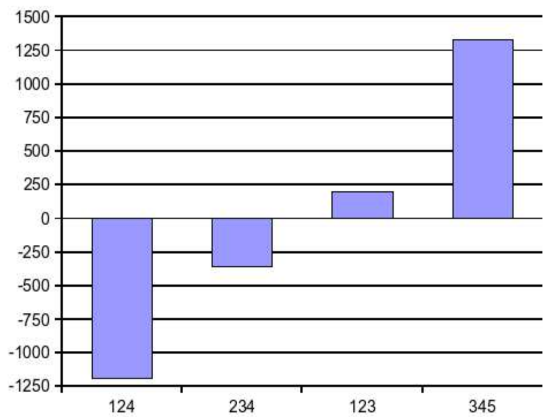
4.4 Connections

Conway's Game of Life involves two different areas of interaction, in each part of the ruleset. Given those relationships, any one change is likely to have various effects in other aspects, and these effects seem not to follow any sort of pattern. Fields such as game theory and quantum mechanics share this quality, and researchers should keep in mind that one cannot really separate one part of a system and examine it alone.

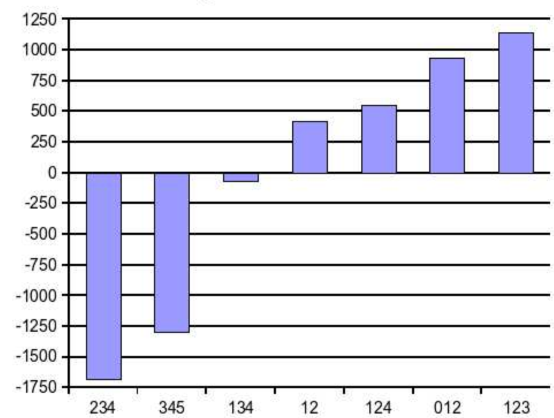
Overall Percentage in Testing



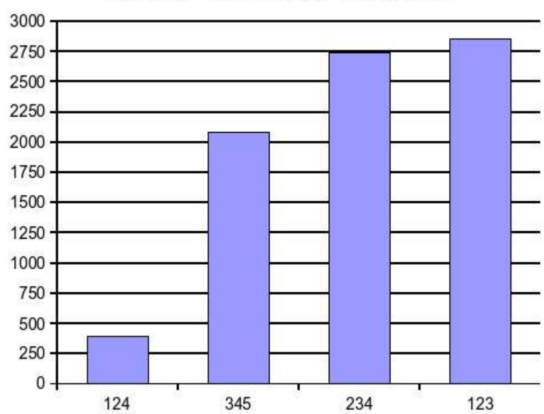
Born Rules Success



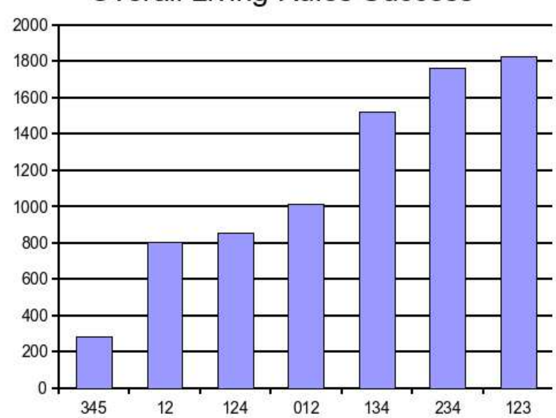
Living Rules Success

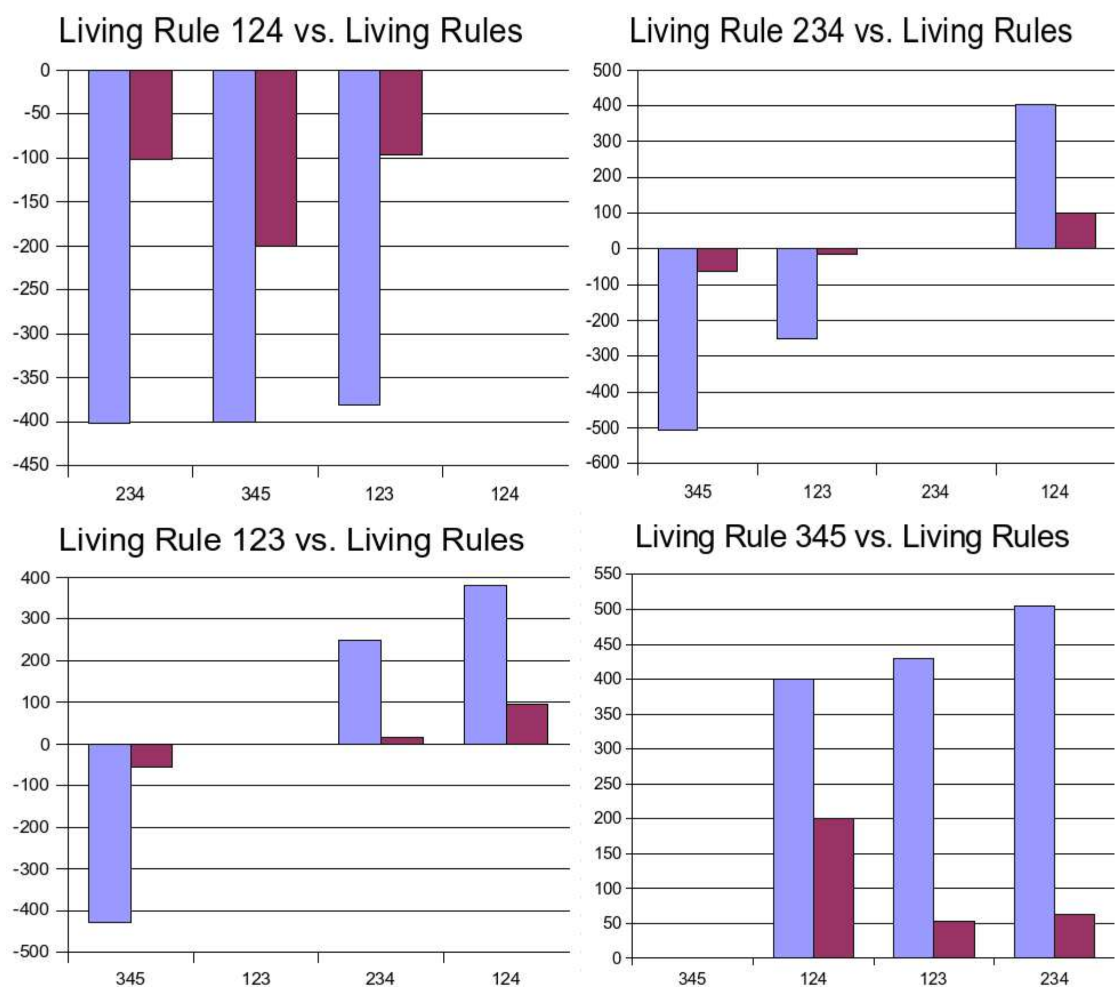


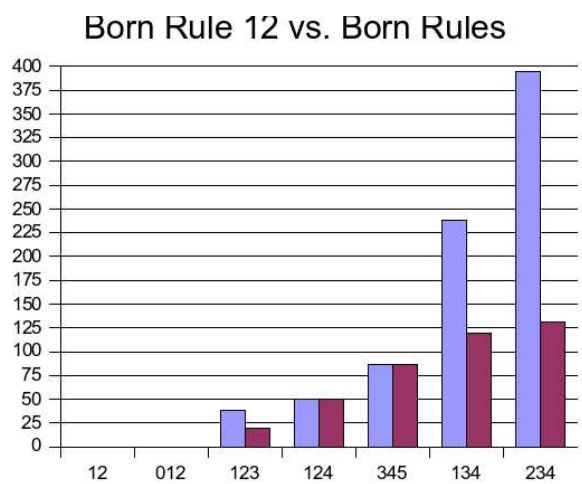
Overall Born Rule Success

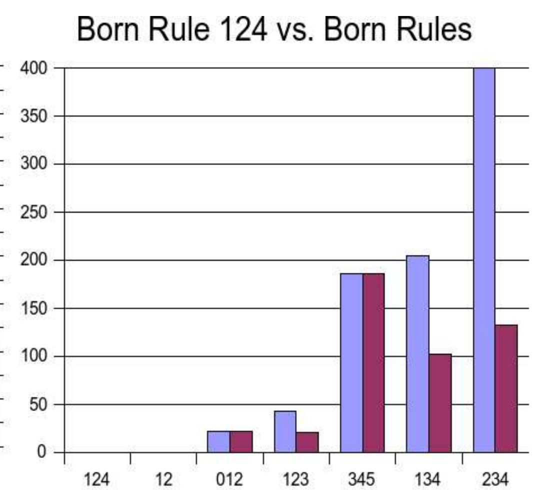


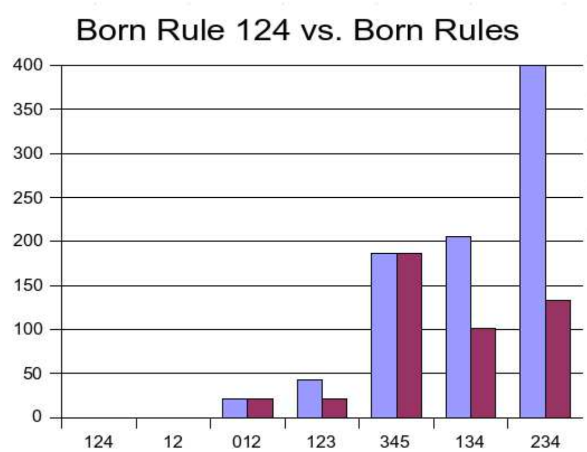
Overall Living Rules Success

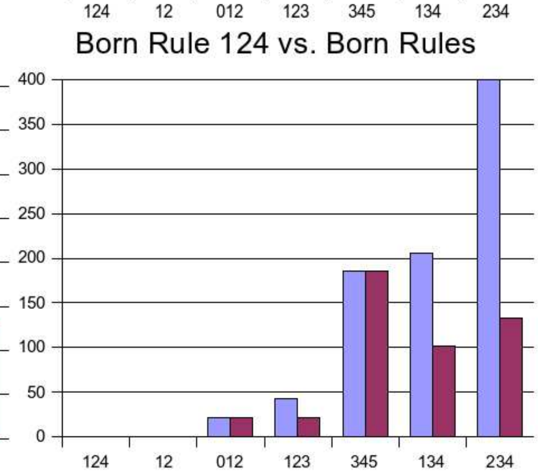



Living Rule 124 vs. Living Rules
0
-50
-100
-150
-200
-250
-300
-350
-400
-450
234
345
123
124
Living Rule 234 vs. Living Rules
500
400
300
200
100
0
-100
-200
-300
-400
-500
-600
345
123
234
124
Living Rule 123 vs. Living Rules
400
300
200
100
0
-100
-200
-300
-400
-500
345
123
234
124
Living Rule 345 vs. Living Rules
550
500
450
400
350
300
250
200
150
100
50
0
345
124
123
234


Born Rule 12 vs. Born Rules
400
375
350
325
300
275
250
225
200
175
150
125
100
75
50
25
0
12
012
123
124
345
134
234


Born Rule 124 vs. Born Rules
400
350
300
250
200
150
100
50
0
124
12
012
123
345
134
234


Born Rule 124 vs. Born Rules
400
350
300
250
200
150
100
50
0
124
12
012
123
345
134
234


Born Rule 124 vs. Born Rules
400
350
300
250
200
150
100
50
0
124
12
012
123
345
134
234

5 Appendix: Graphs

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.