TJHSST Computer Systems Lab Senior Research Project
Excursions into Logic Based Computation using Conway's Game of Life
2007-2008

Liban Mohamed

January 23, 2008

Abstract

Conway's Game of Life is a set of rules in a two dimensional cellular automata grid. Specifically, the Game of Life allows cells to have two states and provides them with eight neighbors. The rules state that between one generation and the next, live cells with two or three live neighbors live on, and dead cells with exactly three live neighbors come to life. This ruleset was specifically chosen by John Conway for the ability to create stable patterns as well as the difficulty of creating patterns which grow infinitely. This difficulty was rather quickly overcome by Bill Gosper's glider gun, which opened up the ability to create binary computational devices such as logic gates. This project endeavors to first create a Game of Life interface that is suitable for the creation of large, complex patterns in the Game of Life, then use this functionality to create such a computational device in the Game of Life.

1 Introduction

1.1 Scope of Study

The first portion of this project involves the design and coding of a high functioning Game of Life interface. This interface then consists of two parts: a set of programs intended to allow users to create and run patterns in the Game of Life, and a set of programs which to allow users to search for patterns which exhibit certain end behaviors. The second part of this project involves using the previously created interface to create a computation device in the Game of Life, and explore the time and space limitations of such a computational device. This second part of the project is what necessiates the first part; it would be virtually impossible to create a computational device in the Game of Life without first having created an extremely high functioning interface and then creating the search programs. Specifically, the second objective of this project is to develop a calculator implemented with the Game of Life. This will require the development of algorithms useable and relatively efficient in the Game of Life. After a four function calculator is created, functions such as exponentiation will be added, though one major failing of any Game of Life implementation is the difficulty of representing numbers not in the whole number set. This is possible, however, and will be another extension of the computing device.

1.2 Expected results

The prospective result of this project is the development of two flexible programs, one for constructing and running designs in the Game of Life the other for searching for reactions and oscillators in the Game of Life world, and a design within the Game of Life which will take specific inputs and return the desired output. In doing this project, I will develop a flexible Game of Life interface, which will be the major new contribution to the field. The underlying ideas are not new - one man spent a number of years developing a Turing Machine extensible to universality in the Game of Life - but the completeness of the interface is unparalleled. It is not feasible to design something as large as a Turing Machine in the course of this project, though it will be necessary to create designs for specific purposes which may be new.

1.3 Type of research

This project involves pure applied research. None of the problems that this project attemts to address have not been tackled before and the trick will to find, using computer searches, patterns which have the functionalities that will be required. This will all be new to me, but it is the nature of cellular automata that fundamental understanding is impossible to achieve: cellular automata are noteable precisely for the ability to defy prediction.

2 Background

As previously mentioned, a man by the name of Paul Rendell spent a number of years developing a finite Turing machine extensible to universality in Conway's Game of Life. His project is the extreme of complexity, but others have created logic gates, most notably Andrew Adamatzky's LogiCell, presented in his *Collision Based Computing.* With respect to Rendell, two rather large extensions are possible on his work. Firstly, he created his Turing machine with extensiblity to universality specifically in mind, meainging one could use his Turing machine as a template to create a universal Turing machine. In addition to that, Rendell is currently working on create a stack cell generator, which would generate tape for his Turing machine, effectively giving it infinite tape. This project differs from the other projects in that it endeavors to create a product, a calculator, which must have a useable user interface. My approach differs from that of Adamatzky's because I will endeavor to create one multipurpose design which will perform all of the functions and give understandeable outputs, as opposed to creating a number of different circuits, the outputs of which would hinge on the state of a specific cell at some arbitrary time.

3 Development

The current edition of my program functions to run any 2 dimensional cellular automata setup using von Neumann neighborhoods. The inputs may be either clicked into the grid or loaded from a text file of on and off states. In the interface, a number of useful functions have been implemented, including the selection and running of sub-grids, copying and pasting rectangular areas of the grid, clearing selected areas, and the aforementioned saving and loading of patterns. The program is highly efficient, running at up to 200Hz in the Systems Lab computers. In order tocomplete my task I will continue by writing a program to search for useful patterns and reactions, and put everything together. During the third quarter, I will focus on the development of the search programs. Analysis for the functioning of my interface and search programs will be done by evaluating the speed of the programs. For the computational device, analysis will be performed by taking into consideration the space and time requirements for performing any function. Testing my program will be done mainly with the search function, which will provide me with patterns for use in my project. The purpose for using a search program is to come up with patterns that will do exactly what I want, and theoretically render program testing unnecessary. However, when I have created the calculator, testing will be rather simple; every inputs will by quickly verifiable with an electronic calculator. The design will necessarily be compartmentalized, so I can only test my program by testing specific parts of it. The major requriement that I will impose upon my program is that it work. I have no idea how much space it will take to make the patterns necessary for computation, and no idea how much time it will take to perform computation.

4 Results

At the end of this project, I can expect a functioning computational device which computes solely in the Game of Life. My analysis of this calculator will have two dimensions - time and space. I can optimize the number of generations it takes to perform any computation and reduce the amount of space it takes to do so.

4.1 Discussion

The first purpose of this project was to create a Game of Life interface that exhibits efficiency and flexibility. Although the program was written in Java, which to some extent limits the efficiency of the interface, the program is able to run at up to 200Hz while displaying every change in state. For the flexibility aspect, the interface excels; the following functions have been implemented: the selection and running of sub-grids, copying and pasting rectangular areas of the grid, clearing selected areas, and the saving and loading of patterns. In addition to loading patterns from a text file to paste onto a grid, it is possible to load a full size grid.

4.2 Conclusion

coming when I have concluded the project

4.3 Recommendations

coming when there's stuff to recommend

5 Appendices

none yet

References

[1] www.rennard.org/alife/english/logicellgb.html

[2] Collision Based Computing, by Andrew Adamatzky