Introduction to Complexity

Unit 6 Homework

(optional)

You can choose whether you want to do the Beginner, Intermediate, or Advanced level. You are free to discuss this part of the homework with anyone, or to ask questions about it on the course forum. You can also use any resources you like, including our course NetLogo models. Note that homework in this course is not turned in or graded.

Beginner Level:

1. Run Mini-Life.nlogo several times with random initial patterns (use "randomize"). Let Life run until it settles down on a set of fixed or oscillating small patterns. Make a record of the different patterns that you see. Compare your list with the list of "non-trivial objects" at the beginning of http://pentadecathlon.com/lifenews/2006/06/single_glider_collision_survey.html

2. In the Code tab of Mini-Life, change the rules of Life to be:

- If a cell has less than 4 black neighbors, then turn black, otherwise turn white.

Try this new rule out on some random initial lattices (vary "percent black" to see if you can get interesting behavior). What is the effect of this rule change on the behavior of the cellular automaton after several time steps? Does this variant of Life still produce interesting behavior?

3. Make another change of your own choosing to the Game of Life rule and experiment with the behavior of your rule on several random initial lattices.

4. In the Code tab of Mini-Life.nlogo, change these two lines in the `go` procedure:

```
[ask cells [count-black-nbrs]]
[ask cells [change-color]]
```

to the following single line:

```
ask cells [count-black-nbrs change-color]
```

This is now an *asynchronous* system, where cells (in random order, once per generation) change their colors immediately upon counting their neighbors, without waiting for them to finish their own counts. Is this new system *deterministic*: does it always calculate the same resulting patterns from the same initial conditions? How does this new system's behavior compare with that of the original Game of Life cellular automaton?

5. What is the Wolfram code number in base 10 of the elementary CA rule **1 0 1 0 0 1 0 1** ? What "Wolfram class" is it in?

6. What is the Wolfram code number in base 10 of the elementary CA rule **0 0 1 0 1 1 0 0** ? What "Wolfram class" is it in?

7. What is the Lambda value of ECA rule 90?

8. Langton's hypothesis was that an ECA with Lambda close to 1/2 should typically have more complex behavior than an ECA with a much lower Lambda value. Is this true when you compare the typical behavior of rule 32 and rule 90? How about when you compare rule 24 and rule 30?

9. Choose five different Elementary CAs that have Lambda = 1/2. Test each of these on several initial configurations (using ElementaryCAs.nlogo). Which ones have "chaotic" looking behavior (Wolfram class 3)?

Intermediate Level:

Implement your own version of ElementaryCAs.nlogo — that is, an Elementary CA simulator—and give the user the option of running it in either "synchronous" or "asynchronous" mode.

In *synchronous* mode (used by current version of ElementaryCAs.nlogo), all the cells update simultaneously. That is, each cell has a current state, and each cell updates its state based on its current state and the current state of its neighbors. A cell's updated state isn't seen by its neighbors until the next time step.

In *asynchronous* mode, instead of all cells being updated at the same time at each time step, the cells are updated one at a time, in a random order. Once a cell is updated, its neighbors use its updated state to calculate their own update.

Investigate how the asynchronous mode changes the behavior of various elementary CAs.

Hint: You may want to start from scratch on the code instead of using the ElementaryCAs.nlogo code, since this model's code is somewhat complicated. You can leave out options such as "edit rule".

Advanced Level:

Implement a version of the Evolving Cellular Automata with Genetic Algorithms project described in Unit 6.6 and in the paper "Evolving Cellular Automata to Perform Computations: A Review of Recent Results" (linked from the Course Materials page). See what results you obtain from evolving with small one-dimensional lattices (otherwise, the computation time can get very large!). Feel free to share your results on the class forum!