

CS 410/510
Nonstandard Computation
Fall, 2005

Homework 2: Cellular automata

Due Wednesday, October 19.

1. (a) Use the format of Figure 7 (b) and (c) in the assigned paper to show how to build a NAND gate in the game of Life, using gliders, glider guns, and / or eaters. **(b)** Use the same format to show how to build a circuit that computes the XOR function.

2. (This problem is adapted from Charles Ofria's course on Digital Evolution and Biocomplexity, MSU.) In the game of Life, what is the smallest possible initial configuration (i.e., using the fewest number of living cells) that will completely die out (i.e., leave all the cells white) in a single update? To make this question non-trivial, at least one cell must die from overcrowding (from having 4 or more neighbors).

Hint: The solution has more than 5 and fewer than 10 living cells in it.

Here is a good Java applet to use in testing your solution: <http://www.bitstorm.org/gameofflife/>

3. Go to <http://www.bitstorm.org/gameofflife/> and load the "Gosper glider gun" as the initial configuration. Watch it run for a little while. Then clear the CA, load the glider gun again, and test its stability by changing the value of one cell state (chosen randomly by you) in the initial configuration and seeing if it still is able to generate a stream of gliders. Repeat this five times, each time changing the state of a different cell in the initial configuration (you can choose both black and white cells to change). What is your conclusion about the stability of the Game of Life to small random errors in state updates?

4. The firing squad problem and its solution were described in the assigned paper. Explain why the solution illustrated in Figure 9 takes $3N - 4$ time steps from the original general's command to the time when all soldiers are firing.