CS470 Homework Assignment 6

Due date: __________________

General info: Turn in all code on both paper and by email (to dbahr@regis.edu with “CS470 Homework” in the subject line). Use old-fashion paper for everything else. For calculations, please show all your work.

Problem #1: Create a simple Turing Machine (with an alphabet of 0 and 1) that accepts if there is a 1 on the tape and otherwise rejects. Remember that by convention, the tape begins with a # before the data on the tape, and has an infinite number of #’s after the data.
   (a) Draw a digraph for this machine.
   (b) List all of the transition functions for this machine.

Problem #2: Using no more than 10 short sentences, succinctly summarize why NAND gates and a tape are sufficient to create a UTM. If you really understand, then your answer can be short but complete. Pretend that you have to explain this to another CS faculty member who has not taken this class but will understand what you are saying if you say it well. You might try listing your explanation as a set of bullet points or numbers. E.g.,
   1. NAND gates can be combined to create AND, OR, and NOT gates.
   2. Using disjunctive normal form, AND, OR, and NOT can be combined to…
   3. A TM, blah, blah, blah…
   4. …

Problem #3: Create a working NOT gate using the Game of Life. Test it with a data stream of 10101. Turn in a .jpg picture of your initial state, and email me the initial state as a data file. (Please do not email me a gigantic .jpg. A hardcopy of the picture is fine.)

Note: If you have troubles with the wrap around boundaries, then don’t forget about glider eaters.

Problem #4: Consider the two-state square lattice next-nearest neighbor totalistic rule 16. Play with this rule and get a feel for its behavior. Can you devise any logic gates such as “AND”, “NOT”, “OR”, etc.? Turn
in your solution, or if unable to devise a logic gate, then explain what feature(s) or cell configuration(s) you need but could not find.

**Problem #5:** Explore various totalistic and outer totalistic two-dimensional rules and find another candidate for universal computation. Briefly describe why it might be a candidate for universal computation.

Note: You do not have to prove your conjecture. **Do not choose any of the rules previously used in homework or class.** Do not choose the Game of Life’s trivial counterpart with the 0’s and 1’s switched. Do not choose any well-known rules that have already been proven to be capable of universal computation.

**Extra Credit (+10 on your end-of-semester final grade!):** Prove that the rule in problem #5, is capable of universal computation. Before you begin, show me your choice for the rule. Your proof must be very well-written and complete. *This will bump up your final grade by one full letter!* What a deal!