COGS 222 **Take-Home Lab #2** **Kirby & Danyluk**
Spring 2007

Due: beginning of class, Tuesday, February 20

In this exercise, you will program several Turing Machines to perform simple tasks. You will likely find it useful to test your program with the Turing Machine simulator introduced in class. You can find the simulator at

http://ironphoenix.org/tril/tm/

Please note that the simulator should run on any platform (Mac, PC, etc), but it does require that your browser be able to run Java applets.

For each task, please turn in the following:

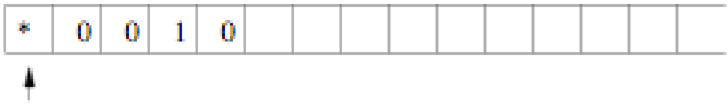
- A complete description of a Turing Machine, including the set of states, the set of tape symbols, and the transition rules. In all cases, the start state should be called "1" and the halt state should be "H" for consistency with the simulator.

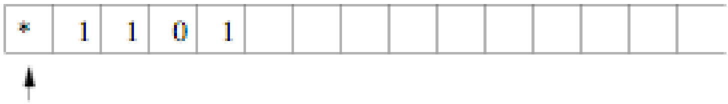
- An electronic copy of the transition rules for each Turing Machine. Instructions for this can be found below.

Please remember to put your name on the lab (both paper and electronic copies).

PROGRAMMING PROBLEMS

1. Design a Turing Machine that takes as input a string represented in binary and flips each of the bits. That is, it changes 0's to 1's and 1's to 0's as in the following diagram:





The Turing Machine should only flip bits up to the first blank encountered on the tape. If there is more input beyond a blank space, it should be ignored. You should assume that a special symbol, "*", marks the left end of the tape. Please be sure that your Turing Machine behaves reasonably even if no input is provided.

2. Design a Turning Machine that takes as input a string of 1's and 0's that represent "True" and "False", respectively, and that returns the logical AND of these values. For example, if all inputs are True, then the Turing Machine should halt with "1" on the tape:

*	1	1	1	1	1									
↑														

*	1													
↑														

If not all inputs are True, then the Turing Machine should halt with "0" on the tape:

*	1	0	1	1	1									
↑														

*	0													
↑														

You may assume that the input will always be non-empty, i.e., that there will always be at least a 0 or 1 input.

3. Design a Turing Machine that reverses a string of 1's and 0's, erasing the original string and writing the duplicate to the right of the position of the original. For example:

*	0	0	0	0	1	1								
↑														

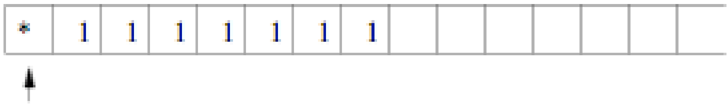
*							1	1	0	0	0	0		
↑														

Your Turing Machine should behave appropriately even if there is no input to reverse.

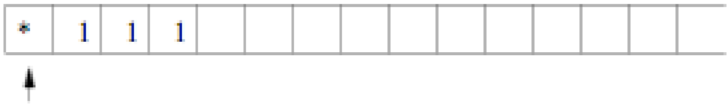
4. Design a Turing Machine that will take as input a natural number represented in unary, divide it by 2, and leave the quotient on the tape. In unary representation,

there is just a single digit, "1". Therefore, 1 in decimal is represented as 1, 2 as 11, 3 as 111, 4 as 1111, etc. Note that the representation described here differs slightly from that described in the reading. We represent 0 as the absence of input.

In your Turing Machine, if the number is odd, then the result should be truncated (i.e., the fractional component of the quotient should be ignored). Thus an input of 1111111 should result in 111 (i.e., 7 divided by 2 is 3). For example:

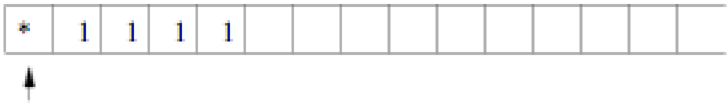


*	1	1	1	1	1	1	1							

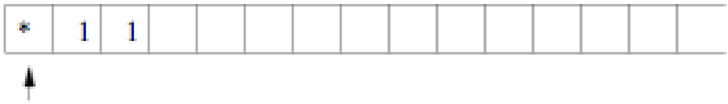


*	1	1	1											

If the input is even, the expected behavior would be:



*	1	1	1	1										



*	1	1												

INSTRUCTIONS FOR ELECTRONIC SUBMISSIONOF TRANSITION RULES

In addition to turning in paper copy of your Turing Machines, you will submit a file with the Turing Machines (or at least the transition rules) electronically. The following instructions indicate how you would do so from one of the Macs in TCL 216 or 217.

First, make sure the name of your file includes your name.

If there is a **Courses** icon on your desktop, skip the rest of this paragraph and simply continue with the rest. From the **Go** menu at the top of the screen, select **Connect to Server**. For the server address, type in **cortland** (or cortland.cs.williams.edu) and then connect. Connect as **Guest**. Select the **Courses** volume on cortland. The Courses icon should now appear on your Desktop.

Double click on the Courses icon and find the **cogs222** folder. Within that folder, you will see a **DropOff** folder. Copy your file into the DropOff folder by dragging it over to the DropOff icon. You will get a message saying that you do not have permission to see the results of the operation. Click ok! (The message is simply telling you that you can't look inside of the folder to see anyone else's work.)

And that's all there is to it!

Before leaving the lab, drag the Courses icon to the trash to disconnect from the server. And, of course, don't forget to log out.