COMS W1004              MW 4:10-5:25 PM

Columbia University                          Spring 2014

**Homework 6: Problem Set**

**April 28, 2014**

Your CUNIX ID: ami2119

Your Last Name: Iwamizu

Your First Name: Akiko

**Circle the range that includes your UNI:**

Group 1 (aa3473-am4051)  Group 8 (kea2134-lvt2107)

**Group 2 (ama2231-av2425)** Group 9 (lz2371-mry2109)

Group 3 (ay2289-cme2133) Group 10 (msv2121-pc2627)

Group 4 (cmh2194-emh2213) Group 11 (pfa2103-sc3719)

Group 5 (emm2224-hv2169) Group 12 (sch2148-tat2133)

Group 6 (hwk2106-jk3667) Group 13 (tb2498-zn2116)

Group 7 (jl4161-kdj2109)

**Chapter 12:  7, 9, 11, 22, 25, 27, 28, 32, 37, and 39**

7. Is the following a legitimate Turing machine? Why or why not?

(1,1,0,2,R)

(1,0,0,3,R)

(2,1,1,2,R)

**(3,0,0,3,R)**

(2,0,0,4,L)

**(3,0,1,4,L)**

(4,1,1,5,R)

(4,0,0,5,R)

This is not a legitimate Turing machine because it does not follow the requirement that a Turing machine can never contain two different instructions of the form

(i, j, -, -, -)

(i, j, -, -, -).

The instructions

**(3,0,0,3,R)**

**(3,0,1,4,L)**

do not follow this requirement, thus this Turing machine is **not legitimate** because there is ambiguity in the instructions.

9. Find the output for the Turing machine

(1,1,1,2,L)

(2,b,0,3,L)

(3,b,1,4,R)

(4,0,1,4,R)

When run on the tape: …b 1 b...

(Note: I wrote more blanks than the given tape because when instructions are in the form …b x b…, we can assume that the ellipses are an infinite amount of blanks on either side.)

…b b 1 b b…

…b b 1 b b…

…b 0 1 b b…

…1 0 1 b b…

**…1 1 1 b b…**

Halt.

11. Describe the behavior of the Turing machine

(1,1,1,1,R)

(1,0,0,2,L)

(2,1,0,2,L)

(2,b,1,3,L)

(3,b,b,1,R)

When run on the tape

…b 1 0 1 b…

the instructions result in an infinite loop(no halt) where a 0 is added to the middle of the tape sequence.

…b 1 0 1 b… …b 1 0 0 1 b…

…b 1 0 1 b… …b 1 0 0 1 b…

…b 0 0 1 b… …b 0 0 0 1 b…

…b 1 0 0 1 b… **…b 1 0 0 0 1 b…**

**…b 1 0 0 1 b…** (loops)

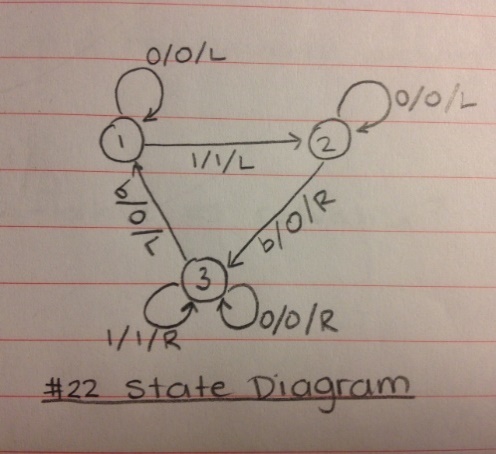
22. Write a Turing machine that begins on a tape containing a single 1 and never halts but successively displays the strings

…b 1 b…

…b 0 1 0 b…

…b 0 0 1 0 0 b…

Instruction set: State diagram:



(1, 0, 0, 1, L)

(1, 1, 1, 2, L)

(2, b, 0, 3, R)

(2, 0, 0, 2, L)

(3, b, 0, 1, L)

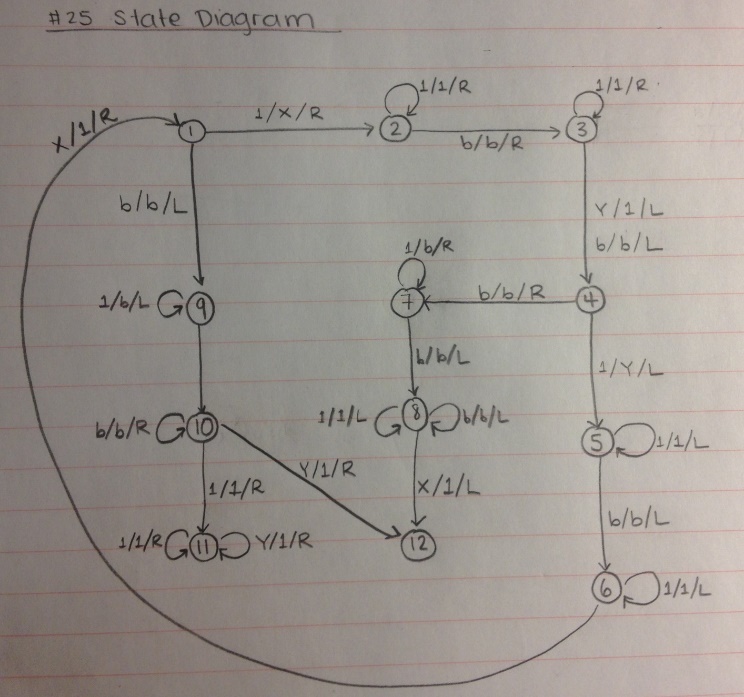
(3, 0, 0, 3, R)

(3, 1, 1, 3, R)

This Turing machine never halts, thus there is no “final output,” so I will explain how the output changes and any patterns in the change of the string on the tape. In this problem, the TM continually adds two 0’s to the tape each time the instruction set repeats. These 0’s replace the first blanks found on the left and right side of the tape.

25. Write a Turing machine that takes as input the unary representation of any two different numbers, separated by a blank, and halts with the representation of the larger of the two numbers on the tape. (Hint: you may need to use a “marker” symbol such as X or Y to replace temporarily any input symbols you have already processed and do not want to process again; at the end, your program must “clean up” any marker symbols.)

Instruction set: State diagram:

(1, b, b, 9, L)

(1, 1, X, 2, R)

(2, b, b, 3, R)

(2, 1, 1, 2, R)

(3, b, b, 4, L)

(3, 1, 1, 3, R)

(3, Y, 1, 4, L)

(4, b, b, 7, R)

(4, 1, Y, 5, L)

(5, b, b, 6, L)

(5, 1, 1, 5, L)

(6, 1, 1, 6, L)

(6, X, 1, 1, R)

(7, b, b, 8, L)

(7, 1, b, 7, R)

(8, b, b, 8, L)

(8, 1, 1, 8, L)

(8, X, 1, 12, L)

(9, b, b, 10, R)

(9, 1, b, 9, L)

(10, b, b, 10, R)

(10, 1, 1, 11, R)

(10, Y, 1, 12, R)

(11, 1, 1, 11, R)

(11, Y, 1, 11, R)

27. Draw a state diagram for a Turing machine that takes any string of 1s and changes every third 1 to a 0.

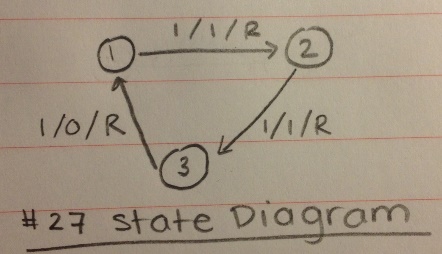
Thus, for example,

…b 1 1 1 1 1 1 b…

Becomes

…b 1 1 0 1 1 0 b…

Instruction set: State diagram:



(1, 1, 1, 2, R)

(2, 1, 1, 3, R)

(3, 1, 0, 1, R)

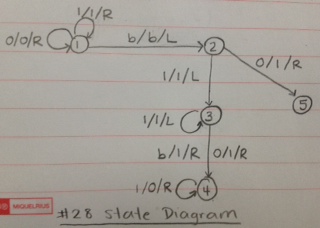
28. Draw a state diagram for a Turing machine that increments a binary number. Thus, if the binary representation of 4 is initially on the tape,

…b 1 0 0…

Then the output is the binary representation of 5,

…b 1 0 1…

Instruction set: State diagram:

(1, 1, 1, 1, R)

(1, 0, 0, 1, R)

(1, b, b, 2, L)

(2, 0, 1, 5, R)

(2, 1, 1, 3, L)

(3, 1, 1, 3, L)

(3, 0, 1, 4, R)

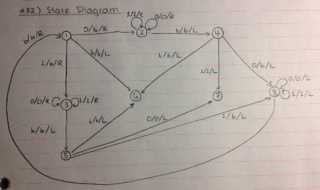
(3, b, 1, 4, R)

(4, 1, 0, 4, R)

32. A palindrome is a string of characters that reads the same forward and backward, such as radar or IUPUI.

Write a Turing machine to decide whether any binary string is a palindrome by halting with a blank tape if the string is a palindrome and halting with a NONBLANK tape (does not have to be the same for every input) if the string is NOT a palindrome.

Instruction set: State diagram:

(1, 0, b, 2, R)

(1, 1, b, 3, R)

(1, b, b, 6, L)

(2, 0, 0, 2, R)

(2, 1, 1, 2, R)

(2, b, b, 4, L)

(3, 0, 0, 3, R)

(3, 1, 1, 3, R)

(3, b, b, 5, L)

(4, 0, b, 8, L)

(4, 1, 1, 7, L)

(4, b, b, 6, L)

(5, 0, 0, 7, L)

(5, 1, b, 8, L)

(5, b, b, 6, L)

(8, 0, 0, 8, L)

(8, 1, 1, 8, L)

(8, b, b, 1, R)

37. Your boss gives you a computer program and a set of input data and asks you to determine whether the program will get into an inﬁnite loop running on these data. You report that you cannot do this job, citing the Church–Turing thesis. Should your boss ﬁre you? Explain.

The reason that the job cannot be done is that the computer program contradicts the definition of algorithm and thus the Church-Turing Thesis. An algorithm must be a well-ordered collection of unambiguous and effectively computable operations that when executed produces a result that halts in a finite amount of time. Hence, because we do not know whether the program will get into an infinite loop running on the data, the computer program must cannot be a valid algorithm. Since the given computer program is not a valid algorithm for solving a symbolic manipulation problem, there does not exist a Turing Machine for that problem. In addition, according to the practical consequence of the uniform halting program, no program can be written to decide whether any given program always stops eventually, no matter what the input. Thus, your boss should not fire you.

39. The uniform halting problem is to decide, given any collection of Turing machine instructions, whether that Turing machine will halt for every input tape. This is an unsolvable problem. Which of the three practical consequences of insolvability problems described in Section 12.7 follows from the uniform halting problem?

The practical consequence of insolvability problems that describes the uniform halting problem is that no problem can be written to decide whether any given program always stops eventually, no matter what the input. This describes the uniform halting program because it explains the consequence that it is impossible to determine if a TM will halt for every input given any collection of TM instructions.