CS 2200 Spring 2019 HW 03

Dr. George Markowsky
Computer Science Department
Missouri Science & Tech

Due Tuesday, February 26, 2019, 11:59PM

Submit your HW as a SINGLE PDF file that includes your answers, your programs, and their output. Also include any data files if you used them. Please make sure that your PDF is well-organized so we can quickly find all the pieces of each problem that belong together.

PROBLEMS

1. (20 points) (Merge Sort) This problem will take you through the steps of writing a recursive Python Merge Sort. This function will not use any variables other than the parameters and completely avoids all iterative constructions.

   (a) (5 points) Write a purely recursive Merge(L1,L2) function that takes two lists of numbers sorted in monotone-increasing order, L1 and L2, and merges them correctly into a single list of numbers sorted in monotone-increasing order. You may not use any local or global variables or loops of any sort.

   (b) (5 points) Write a purely recursive MergeSort function that takes a single list of numbers and produces a list of numbers sorted in monotone-increasing order. You may not use any local or global variables or loops of any sort.

   (c) (10 points) Prove rigorously that the MergeSort function that you wrote is correct.

2. (15 points) (Peano's Postulates) Define an operation $\dot{-}$ called *bounded subtraction* as follows:

   (a) $0 \dot{-} a = 0$

   (b) $a \dot{-} 0 = a$

   (c) $a' \dot{-} b' = a \dot{-} b$

   Prove that for all a, b in N, $(a + b) \dot{-} b = a$.

3. (35 points) Write a Python simulator for finite-state automata that meets the following specifications. The machine definition and tape will be stored in a text file that has the extension ".fsa".

   Each line of the text file shall begin with one of the following capital letters: A, B, D, O, S, T. We will explain the meaning of these letters below. The letters appear at the very beginning of the line and there is a single blank space after the letter for readability. Each line must begin with a letter. If the file does not meet the specifications, it is rejected. If a line does not have a letter on some line, the file is rejected.

   The first line must be an A line which contains just a single string consisting of all the characters in the alphabet. There is only one A line per file.

   The A line is followed by a series of S lines. S stands for state. An S line states that the line gives information about states such as their names and whether they are final or not. An S line has the following structure S S1,Bool1,S2,Bool2,...,Sk,Boolk where Si is the name of a state and Booli is 1 if the state is final and 0 is the state is not final.

   Next comes a B line which contains the beginning (start) state There is only one such line and it looks like B State.

   Next come the D lines where D stands for delta. The structure of a D line is as follows: D S_1,C_1,N_1,S_2,C_2,N_2...,S_k,C_k,N_k where S_i is the current state, C_i is the character being read, and N_i is the new state that the machine enters. Each line contains a whole number of triples although the number of complete triples per line is not specified. The lines could lack any entries, but they still should have the blank following the letter.

   After the last D line, come a series of alternating T and O lines. T stands for tape and contains a character string representing a initial tape. Every T line is followed by an O line. The O line is blank if the preceding T line has not been processed by the machine. The line contains the word "accepted" if the machine would accept the string on the T line and contains the word "rejected" if the machine would reject the string on the T line.

   After the program processes the file it should process any T lines for which there are no answers on the corresponding O lines. Once the simulator finishes, it should reconstruct the file that it read, but add any new computations to the updated file.

   Include your program with comments in the PDF file that you will submit. This should be easy since the program is a text file. Be sure that the comments include the information that I gave you about the format of the file. You will use this simulator to test all of the solutions to the remaining problems. For each problem you will include the .fsa file that contains the solution.

4. (15 points) Find a deterministic finite-state automaton that recognizes the language, L, consisting of all strings in {a, b}* that contain an odd number of b's

such that there is at least one "a" between every two b's in the string. Submit the .fsa file for this machine and make sure that there are at least 10 computations included in the file. Make at least 3 of the computations non-trivial.

5. (15 points) Let L be a language over the alphabet {d, e} be the language of all strings having length ≤ 6. Construct a deterministic finite-state machine that recognizes L. Submit the .fsa file for this machine and make sure that there are at least 10 computations included in the file. Make at least 3 of the computations non-trivial.