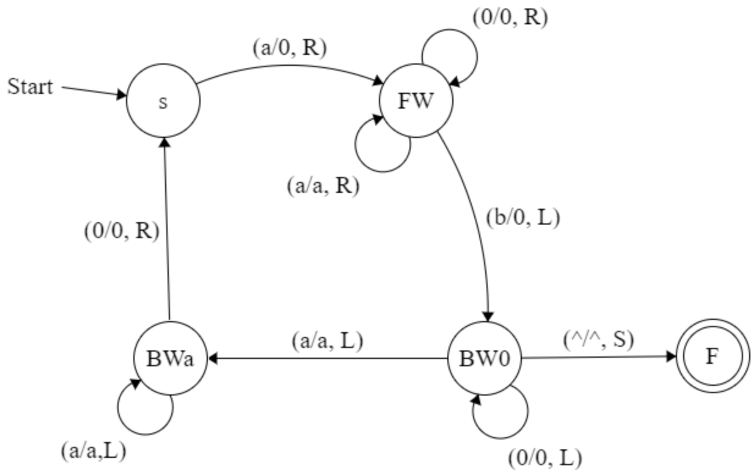
CSE396 Problem Set 8 Due 4/19, 11:59pm

Reading:

The **Second Prelim Exam** is fixed for **Thursday, April 26** in class period. It is technically "cumulative" because stuff from the first month may figure on the exam the way they have on Assignments 5 and 6. But it will focus on the domain of Assignments 5–8. Tuesday's lecture will finish Chapter 3 and begin Chapter 4.

Assignment 8, due Thu. 19 April, 11:59pm under the usual terms.

(1) This problem is "HW8 Online Part" on *TopHat*, worth 20 pts. as before. The questions were written by the TAs Minwei Ye, Jiayi Xian, and Junxuan Huang with additions and edits by me. Here for convenient reference is the diagram of the Turing machine for the first seven questions. It was drawn using the "Finite State Machine Designer" page by Evan Wallace but you can draw Turing machines too because it's doesn't care how you label arcs.



(2) Over $\Sigma = \{a, b, c\}$, Consider the following context-free grammar $G = (V, \{a, b, c\}, R, S)$ with rules:

$$\begin{aligned} S &\to BAc \mid SB \mid bc \\ A &\to cBA \mid Bcb \\ B &\to cS \mid a \end{aligned}$$

Let T be the language of strings x such that between any two b's in x there are at least two c's.

(a) Prove that $L(G) \subseteq T$ using the structural induction technique from lectures and notes. All variables will include the property that all strings they derive by themselves belong to T but you will need to add further properties that at least one of the variables must uphold. Defining effective properties P_S, P_A, P_B that are upheld by all the rules counts for 9 pts. and verifying each non-terminal rules is 3 pts., for 24 total.

(b) Suppose we change the first rule for B from $B \rightarrow cS$ to $B \rightarrow aS$. Give a parse tree of a string in the new grammar that does not belong to T. (6 pts.)

(3) Let L be the language of balanced-parentheses strings that are mirror-image symmetric from problem (2) of assignment 7. Design a two-tape Turing machine M such that $L(M) = L$. For starters, you are welcome to grab the code for the balanced-parentheses part (in the handout `https://www.cse.buffalo.edu/˜regan/cse396/BalParens.pdf`) and build on it. You may draw M by hand or use the "Finite State Machine Designer" page (but then you can't run it) or be adventurous and set up the "Turing Kit" to design it (the directions on the course webpage worked for a student earlier this week; its Postscript print output is wonky but you can always take a screenshot). However you do it, your diagram must be well-commented in the drawing and/or below.

Also answer: what is the maximum number of steps your M can take on an input x of length n? It is not enough to say "$O(n)$"; we care about the constant factor on the "n" though we don't care about any lower-order terms. (21 + 3 = 24 pts., for 74 on the set)