Test 2

CS385-Theory of Computation

November 15, 2007

There are two problems on this test, each with several parts—a finite automaton problem and a context-free language problem. The usual rules apply—you should show all your work carefully, and justify any claim carefully but succinctly. It is permissible to refer to results presented in class, homework problems, and the textbook, but you must do this accurately

1 Finite Automaton Problem

1. Draw the state-transition graph of a 4-state NFA that recognizes the language L_1 generated by the regular expression:

$$(a \cup b)^* aba(a \cup b)^*.$$

2. Use the subset construction to find a DFA equivalent to your NFA of problem 1. When you do this in the most straightforward way possible, you get a DFA with 6 states (if I did it right!) but you should find that the device can be simplified considerably by merging several states, to obtain a four-state device.

3. Use your result in part 2 to find a 3-state automaton for the set of strings over $\{a, b\}$ that contain no occurrence of aba. The device you produce will be deterministic except that there may be some unspecified transitions (*i.e.*, pairs q, σ for which $\delta(q, \sigma)$ is not defined).

2 Context-Free Language Problem

Consider the language L_2 generated by the context-free grammar G given below.

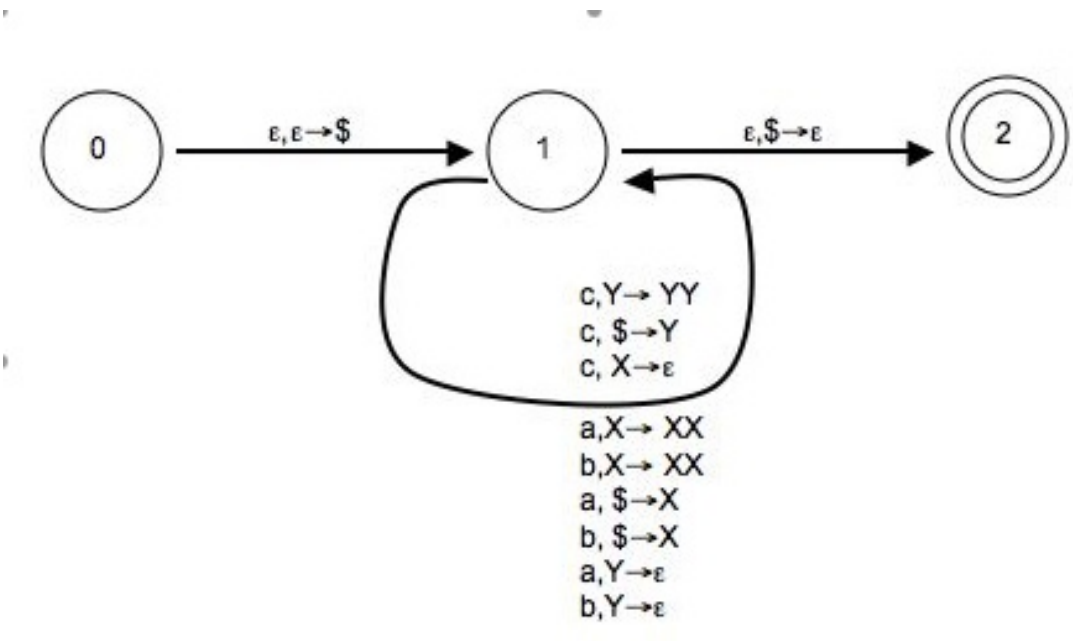
$$S \longrightarrow TScS|cSTS|\epsilon$$

$$T \longrightarrow a|b.$$

1. Show $cacb \in L_2$ by producing a derivation tree for this string.

2. Show G is ambiguous. (HINT: Your answer in 1 should help!)

3. Pictured below is a PDA that recognizes L_2.



Note the large number of transitions labeling the loop at state 1. You should observe that the definition of PDA has been tweaked slightly: We are allowing transitions like

$$a, X \to XX,$$

which means "on input a with X on top of the stack, push X onto the stack" whereas the official definition given in the text requires that this be done in two steps, with an additional state. The device is deterministic as long as you restrict the transition into state 3 to be carried out only when there is no longer any input.

Show the trace of the PDA's behavior on the input string *cacb*. The trace should indicate at each step the state, the remaining input, and the contents of the stack.

4. Give a *precise description* of L_2. (You might be able to determine this from the grammar, but it is probably easier to use the PDA. As a hint, you should observe that the stack can never contain both X's and Y's.What is the significance of the number of X's or the number of Y's in the stack?)

5. Is L_2 a regular language? Explain.