CSE 322: Introduction to Formal Models in Computer Science
Assignment #6
November 15, 2006
due: Wednesday, November 22

1. Let $\Sigma = \{a, b, \#\}$. Give a pushdown automaton for the language
$$\{u\#v \mid u, v \in \{a, b\}^* \text{ and } v^R \text{ is a substring of } u\}.$$
v^R denotes the reversal of the string v. You should specify the transition function by giving the state diagram. You need not turn in a proof of correctness, though it would be good reassurance for yourself to do such a proof.

2. Give a pushdown automaton for the language
$$\{a^m b^n \mid n \leq m \leq 2n\}.$$
You should specify the transition function by giving the state diagram. You need not turn in a proof of correctness, though it would be good reassurance for yourself to do such a proof.

3. Use the procedure of Lemma 2.21 [1st Ed: Lemma 2.13] to convert the grammar G_3 of Example 2.3 [1st Ed: Example 2.2] into an equivalent pushdown automaton M. You may use the shorthand allowing the automaton to push more than one symbol in a single step in your state diagram. Show an accepting computation of M on the input *aababb*, together with the corresponding derivation of this string in G_3.

4. Use the procedure of Lemma 2.27 [1st Ed: Lemma 2.15] to convert the pushdown automaton of Figure 2.19 [1st Ed: Figure 2.8] into an equivalent grammar. Here are the steps you should follow:

   (a) q_1 needn't be in F, since this doesn't change the language accepted. Explain why not. This change leaves a unique accepting state, which we need for the conversion procedure.

   (b) You will have to add a state p between states q_2 and q_3. Explain why. What are the transitions from q_2 to p and from p to q_3? (You will save yourself a little effort later if you use a brand new stack symbol here.)

   (c) Write down all the rules that the conversion procedure requires of the types $A_{pq} \rightarrow aA_{rs}b$ and $A_{pp} \rightarrow \varepsilon$.

   (d) The conversion procedure requires 125 rules of the third type (why?), and I won't make you write them all down. Instead, explain why you never need any rules of this third type if the pushdown automaton does all of its pushes before any of its pops. (Such a pushdown automaton is called a "one-turn pushdown automaton".)

   (e) What is the set V of variables of your resulting grammar, and which is the start variable S?