Homework 3

Due midnight, Thursday, 2/9/2012

Please submit your homework solutions online at `http://www.dci.pomona.edu/tools-bin/cs081upload.php`. If you have more than one file to be turned in, please put it in a folder and zip it up before turning it in.

Problems from the texts are given in the form c.n where c is the chapter and n is the problem number. Thus problem 2.7 is problem 7 from Chapter 2.

1. (5 points) **Regular Languages**

   Let $G = (V, \Sigma, R, S)$ where $V = \{a, b, S\}$, $\Sigma = \{a, b\}$, and R contains:

   - $S \rightarrow aSb$
   - $S \rightarrow aSa$
   - $S \rightarrow bSa$
   - $S \rightarrow bSb$
   - $S \rightarrow \epsilon$.

   Show $L(G)$ is regular.

2. (15 points) **Context-free Language**

   Let L $= \{w \in \{a, b\}^* | w = w^{rev}\}$.

   (a) Design a cfg generating L.

   (b) Design a pda accepting L.

   (c) Design a cfg generating the complement of L

3. (10 points) **Chomsky Normal Form**

   We consider the strings of balanced parentheses of two types over the alphabet is $\Sigma = \{(,), [,]\}$. The language PAREN_2 is the smallest set of strings satisfying the following three properties.

   (a) $\epsilon \in \text{PAREN}_2$;

   (b) if x is in PAREN_2, then so are (x) and $[x]$; and

   (c) if x and y are in PAREN_2, then so is xy.

   Give a context-free grammar in Chomsky normal form for PAREN_2.

4. (10 points) **Ambiguity and Parse Trees**

   Rich 11.10

5. (10 points) Use the construction in the text or lecture to create a pda that accepts the language generated by the grammar for arithmetic expressions with V = {Exp, Addop, Term, Factor, Mulop, num, *, +, -, /, (,) }, Σ = {num, *, +, -, /, (,) }, start symbol Exp, and the following rules:

   - Exp $\rightarrow$ Exp Addop Term | Term

- Term → Term Mulop Factor | Factor
- Factor → (Exp) | num
- Addop → + | -
- Mulop → * | /

6. (20 points) **PDA Variants**

   Let $M = (K, \Sigma, \Gamma, \Delta, s, A)$ be a pda. The language accepted by M by final state is defined as follows:

   $$L_f(M) = \{w \in \Sigma^* | (s, w, e) \vdash_M^* (f, \epsilon, \alpha) \text{ for some } f \in A, \alpha \in \Gamma^*\}$$

   This differs from the definition in Rich by allowing acceptance even when the stack is non-empty.

   Note that to show two languages L_1 and L_2 are the same (e.g., as in the two parts below), it is generally simplest to show that $L_1 \subseteq L_2$ *and* the reverse.

   (a) For every pda M, show there is a pda M' s.t. $L(M') = L_f(M)$. Notice that one uses the subscript f and the other does not.

   Hint: Let M' be a variant of M that allows the possibility of emptying its stack whenever M enters a final state.

   (b) For every pda M, show there is a pda M' s.t. $L_f(M') = L(M)$.

   Hint: Let M' begin by putting a special marker on top of the stack that will allow a check to see if the stack is otherwise empty.