CPSC 421: Introduction to Theory of Computing
Practice Problem Set #2, Not to be handed in

1. (Easy) Find the language described by the following context-free grammars:

   (a) $S \to abS \mid a$

   (b) $S \to bSb \mid A$
   $A \to aA \mid \epsilon$

   (c) $S \to aSa \mid bSb \mid L$
   $L \to a \mid b \mid \epsilon$

   (d)
   $$S \to AS \mid B$$
   $$A \to aAc \mid Aa \mid \epsilon$$
   $$B \to bBb \mid \epsilon$$

2. Find a context-free grammar (i.e., give the productions) to describe the following languages.

   (a) (Easy) $L = \{0^n 1^{2n} \mid n \geq 0\}$

   (b) (Easy) $L = \{w \mid \text{the length of } w \text{ is odd}\}$, $L \subseteq \{0,1\}^*$

   (c) (Easy) The set of all strings of a and b that include the substring baa.

   (d) (Medium) The set of strings over the alphabet $\{a, b\}$ with an equal number of a's and b's.

   (e) (Medium) The set of strings over the alphabet $\{a, b\}$ with at least as many a's as b's.

   (f) (Getting tricky) The set of strings over the alphabet $\{a, b\}$ with twice as many a's as b's.

3. Show that the class of context-free languages is closed under the regular operations: union, concatenation, and star.

4. Consider the CFG G defined by productions:

   $S \to aS \mid Sb \mid a \mid b$

   (a) Prove by induction on the string length that no string in $L(G)$ has ba as a substring.

   (b) Describe $L(G)$ informally. Justify your answer using part (a).

5. The following grammar generates the language of regular expression $0^*1(0 \cup 1)^*$:

   $S \to A1B$

   $A \to 0A \mid \epsilon$

   $B \to 0B \mid 1B \mid \epsilon$

   Give (leftmost) derivations of the following strings:

(a) 00101.
(b) 1001.
(c) 00011.

6. Give parse trees for the grammar and each of the strings in 5.

7. Consider the grammar:

   $S \to aS \mid aSbS \mid \epsilon$

   This grammar is ambiguous. Show in particular that the string aab has two

   (a) Parse trees.
   (b) (Leftmost) derivations.

8. Find an unambiguous grammar for the language in question 7.

9. Design a PDA to accept each of the following languages

   (a) The set of all strings of 0's and 1's with an equal number of 0's and 1's.
   (b) The set of all strings with three times as many 0's as 1's.
   (c) Strings of the form $1^*0^n1^n$, $n \geq 0$.

10. Convert each of the grammars in 1. to PDAs.

11. Can pushdown automata accept sets of strings of the form? Either give a PDA to accept the language, or a high-level explanation for why that seems impossible.

    (a) ww where w is a string of zeros and ones.
    (b) ww where w is a string of zeros.

12. When a pushdown machine executes an instruction and does not move its reading head, we say that it has made an *epsilon move.* Does this new capability add power to these automata? Why? That is, do *epsilon moves* make PDAs strictly more powerful?

13. The *reverse* of a string w, written w^R is the string obtained by writing w in the opposite order (i.e., $w^R = w_n w_{n-1} ... w_1$). The reverse of a language L is the set of strings that are the reverse of some string in L (i.e., $L^R = \{s \mid s^R \in L\}$). Prove that if L is a CFL, then so is L^R.

14. (*) Consider the following two languages:

    $L_1 = \{a^n b^{2n} c^m \mid n, m \geq 0\}$
    $L_2 = \{a^n b^m c^{2m} \mid n, m \geq 0\}$

    (a) Show that each of these languages is context-free by giving PDAs that recognize them.
    (b) Is $L_1 \cap L_2$ a CFL? Give a high-level explanation.

15. (*) Convert your PDAs from 14.a into CFGs.

16. (*) The *shuffle* of two strings w and x is the set of all strings that one can get by interleaving the positions of w and x in any way. More precisely, *shuffle*(w, x) is the set of strings z such that

    1. Each position of z can be assigned to w or x, but not both.
    2. The positions of z assigned to w form w when read from left to right.
    3. The positions of z assigned to x form x when read from left to right.

    For example, if $w = 01$ and $x = 110$, then *shuffle*$(01, 110)$ is the set of strings:

    {01110, 01101, 10110, 10101, 11010, 11001}

    To illustrate the necessary reasoning, the third string, 10110, is justified by assigning the second and fifth positions to 01 and positions one, three, and four to 110. The first string, 01110, has three justifications. Assign the first position and either the second, third or fourth to 01, and the other three to 110. We can also define the shuffle of two languages, *shuffle*(L_1, L_2) to be the union over all pairs of strings, w from L_1 and x from L_2, of *shuffle*(w, x).

    (a) What is *shuffle*$(00, 111)$?

    (b) What is *shuffle*(L_1, L_2) if $L_1 = \{0\}^*$ and $L_2 = \{0^n 1^n \mid n \geq 0\}$?

    (c) Show that if L_1 and L_2 are regular languages then so is *shuffle*(L_1, L_2). *Hint*: start with DFA's for L_1 and L_2.

    (d) Show that if L is a CFL and R is a regular language, then *shuffle*(L, R) is a CFL. *Hint*: start with a PDA for L and a DFA for R.

    (e) Give a counterexample, along with a high-level explanation, to show that if L_1 and L_2 are both CFL's, then *shuffle*(L_1, L_2) need not be a CFL.

17. (*) Show that CFL's are closed under the following operation:

    $$cycle(L) = \{w \mid \text{we can write } w \text{ as } w = xy, \text{ such that } yx \text{ is in } L\}$$

    For example, if $L = \{01, 011\}$ then $cycle(L) = \{01, 10, 011, 110, 101\}$. Note that x and y need not be in L. *Hint*: Try a PDA-based construction.