Computer Science 81, Spring 2013
Assignment 9
Due Monday April 22

1. (Sipser 2.11) Convert the CFG of Exercise 2.1 to an equivalent PDA, using the procedure given in Theorem 2.20 (which is essentially the Produce-Match method described in class). The CFG is given below and E is the start symbol.

$$E \to E + T \mid T$$
$$T \to T * F \mid F$$
$$F \to (E) \mid a$$

2. Encode your solution to the proceeding into JFLAP (or some equivalent framework) and use JFLAP to determine whether your PDA accepts the following strings. Include a screen shot of the result.

a
(a)
a*a
a+a
a*a+a
a+a*a
a+a+a
a*(a+a)
+a
+a*a
a*+a
()

3. Repeat problem 1, except use the Shift-Reduce method described in class.

4. Repeat problem 2, except use the results of problem 3.

5. (Abbreviated version of Sipser 2.27) Consider the following grammar, with start symbol S and terminal alphabet {if, c, then, else, a} (each element is considered to be a single symbol) and non-terminal alphabet {S, A, I, E}

S → A | I | E	(S is <STMT>)
I → if c then S	(I is <IF-THEN>)
E → if c then S else S	(E is <IF-THEN-ELSE>)
A → a	(a is <ASSIGN>)

   a. Show that G is ambiguous.
   b. Give an unambiguous grammar that generates the same language.

6. For the grammar of problem 1, show how the CYK algorithm would parse the string **a*a*a+a+a*a**.

7. For the grammar of problem 1, give an equivalent grammar having no left recursion.

8. For the grammar you devise in problem 7, compute which symbols are Nullable, and the sets First and Follow for each non-terminal.

9. Using the grammar you devise in problem 7, and the results of problem 8, construct the LL(1) parse table.

10. Trace the parse of the string **a*a*a+a+a*a** using the parse table from problem 9.