Homework Assignment #2

Due: Wednesday, April 19, 2006

1. Write an unambiguous grammar for each of the following languages. **Hint**: One way of verifying that a grammar is unambiguous is to run it through Yacc, the UNIX parsing tool, and get no conflicts.
   a. Balanced parentheses and square brackets. Example:
      `([[](()[()][])])`
   b. Balanced parentheses and brackets, where a closing bracket also closes any outstanding open parentheses (up to the previous open bracket). Example: `[([](()[(][]))]` **Hint**: First make a grammar for 1a, where extra open parentheses are allowed; then make sure this nonterminal must appear within brackets.
   c. Statement blocks in Pascal or ML where the semicolons **separate** the statements:
      `{ statement ; {statement ; statement} ; statement }`
   d. Statement blocks in C where the semicolons terminate the statements:
      `{ expression ; {expression ; expression; } expression; }`

2. Write a grammar that accepts the straight line code language given below, but that is suitable for LL(1) parsing. That is, eliminate the ambiguity, eliminate the left recursion, and (if necessary) left-factor.

```
S ::= S ; S     S ::= id := E   S ::= print( L )
E ::= id        E ::= num       E ::= E + E
E ::= ( S, E)   L ::= E         L ::= L , E
```

3a. Build the LR(0) DFA for this grammar:

```
S := E $
E := id
E ::= id ( E )
E ::= E + id
```

b. Is this an LR(0) grammar? Give evidence.

4. Consider the following MiniJava statement:

```
while (x < 3+4*5) {
   System.out.println(x);
   y = new C().test(a,b,c);
}
```

a. Draw the concrete syntax tree for this statement. Use the MiniJava grammar embedded in the `Parser/minijava.cup` file. (You may abbreviate non-terminal names if unambiguous, e.g. `Id` for Identifier.)
b. Draw the abstract syntax tree for this statement, labeling each AST node with the name of an AST node class from the MiniJava compiler and labeling each child edge with the name of an instance variable of the parent node's class. (Child edges that lead to non-AST nodes, such as integers or strings, can be drawn to lead to a particular integer or string value.)

Produce a hard-copy of your answers and turn them in by the start of class on the due date.

Do these exercises individually, not with your project partner.