Written Assignment 2
Due October 21 at 5:00 PM

This assignment asks you to prepare written answers to questions on context-free grammars, parse trees, and parsing. Each of the questions has a short answer. You may discuss this assignment with other students and work on the problems together, but your write-up must be your own individual work. Remember that written assignments are to be turned in either at the start of lecture, electronically, or in Prof. Aiken's Office (Gates 411) by 5:00 PM on the due date.

1. Give a context-free grammar (CFG) for each of the following languages over the alphabet $\Sigma = \{0, 1\}$:
   (a) All nonempty strings that start and end with the same symbol.
   (b) All strings that contain more 1s than 0s.
   (c) All palindromes (a palindrome is a string that reads the same forwards and backwards).

2. Consider the following grammar:

$$\begin{aligned} S &\rightarrow aSb \\ S &\rightarrow aS \\ S &\rightarrow \epsilon \end{aligned}$$

   (a) Give a one-sentence description of the language generated by this grammar.
   (b) Show that this grammar is ambiguous by giving a string that can be parsed in two different ways. Draw both parse trees.
   (c) Give an unambiguous grammar that accepts the same language as the grammar above.

3. The Cool Reference Manual, in Chapter 11, has the context-free grammar defining the syntax for Cool. Create the parse tree from the grammar definition for the following class definition.

```
class FOO inherits BAR {
  i : Int <- 42;
  baz (x:Int) : Foo { i <- x + i };
};
```

You do not need to construct nonterminals that immediately produce terminals, or include terminals which add no meaning because they are captured by the nature of the production. For example, for the expression "1 + 2", the following parse tree would be sufficient:

```
         ADD
        /   \
       /     \
     INT     INT
     (1)     (2)
```

4. Give a one-sentence description of the language generated by each of the following CFGs.

   (a)

$$\begin{aligned} S &\rightarrow Z1Z1Z1A \\ Z &\rightarrow Z0 \mid \varepsilon \\ A &\rightarrow A0 \mid A1 \mid \varepsilon \end{aligned}$$

   (b)

$$S \rightarrow 0 \mid 1 \mid 0S0 \mid 0S1 \mid 1S0 \mid 1S1$$

5. Give an example of a simple grammar that is $LL(2)$ but not $LL(1)$.

6. Consider the following CFG, which has the set of terminals $T = \{\mathbf{id}, \mathbf{(}, \mathbf{)}, [,], \mathbf{;}\}$.

$$\begin{aligned} E &\rightarrow \mathbf{id} \mid \mathbf{id(}A\mathbf{)} \mid \mathbf{id}[E] \\ A &\rightarrow E \mid E\,\mathbf{;}\,A \end{aligned}$$

   (a) Left-factor this grammar so that no two productions with the same left-hand side have right-hand sides with a common prefix.
   (b) Construct an LL(1) parsing table for the left-factored grammar.
   (c) Show the operation of an LL(1) parser on the input string **id(id[id]; id)**.

7. Consider the following CFG, which has the set of terminals $T = \{\mathbf{a}, \mathbf{b}\}$.

$$\begin{aligned} S &\rightarrow X\mathbf{a} \\ X &\rightarrow \mathbf{a} \mid \mathbf{a}X\mathbf{b} \end{aligned}$$

   (a) Construct a DFA for viable prefixes of this grammar using LR(0) items.
   (b) Identify a shift-reduce conflict in this grammar under the SLR(1) rules.
   (c) Assuming that an SLR(1) parser resolves shift-reduce conflicts by choosing to shift, show the operation of such a parser on the input string **aaba**.
   (d) Suppose that the production $X \rightarrow \varepsilon$ is added to this grammar. Identify a reduce-reduce conflict in the resulting grammar under the SLR(1) rules.