**Jim Lambers**
**CS143 Compilers**
**Summer Quarter 2000-01**
**Homework Assignment 1**

This assignment must be submitted by Wednesday, July 18. On-campus students must submit the assignment in class, while SITN students must submit it to the courier on the due date.

This assignment is worth 100 points.

1. The ANSI C specification dictates that nested comments are not permitted. In other words, once a `/*` has been seen in the input to begin a comment, the comment extends to the first `*/` that occurs, even if another `/*` has been seen. Some languages, or compilers for languages, allow nested comments, so that each `/*` must be matched by a `*/`.

   (a) Write a C program that will compile regardless of whether the compiler allows nested comments. If the compiler does support nested comments, the program must print "Cheep cheep!" Otherwise, it must print "Moo!"

   (b) Explain why the language of nested C comments cannot be described using a regular expression. Can it be described using a context-free grammar?

2. Consider the following grammar:

   $$R \rightarrow R'|'R \quad | \quad RR \quad | \quad R^* \quad | \quad (R) \quad | \quad a \quad | \quad b \quad | \quad c \quad | \quad '\epsilon' \qquad (1)$$

   Note that the first $|$ is a terminal symbol denoting the character $|$, rather than a separator between productions. Also, the $\epsilon$ denotes the character $\epsilon$, rather than an $\epsilon$-production.

   (a) Show that this grammar generates all regular expressions over the alphabet $\Sigma = \{a, b, c\}$.

   (b) Show that this grammar is ambiguous.

   (c) Construct an equivalent unambiguous grammar that conforms to the following rules for precedence and associativity:

   - The unary operator $*$ has the highest precedence, followed by concatentation, with $|$ having the lowest precedence.
   - All operators are left associative.

(d) Using the new grammar, construct a parse tree for the sentence $(ab|b)^*c$.

3. Consider the following grammar. As usual, upper case letters denote non-terminals while lower-case letters denote terminals.

$$\begin{array}{lcl}
S & \rightarrow & Em \quad | \quad sB \\
B & \rightarrow & lCr \\
C & \rightarrow & AC \quad | \quad \epsilon \\
A & \rightarrow & DF \\
F & \rightarrow & SF \quad | \quad \epsilon \\
D & \rightarrow & cEnD \\
E & \rightarrow & w \quad | \quad aqE
\end{array}$$

(a) Compute the FIRST and FOLLOW sets for each nonterminal.

(b) Prove that this grammar is LL(1).

(c) Create a LL(1) parsing table for this grammar.

(d) (BONUS) Show that no LL(1) grammar is ambiguous.