COS 487: Theory of Computation **Fall 1999**

Assignment #2

Due: Monday, October 18

Suggested reading (for lectures 4,5,6): Sipser Chapter 2.

Problems (from lectures 4, 5, 6):

1. Show that the language $\{0^m 1^n : m \leq 2n \leq 4m\}$ is context-free.

2. Let G be the following CFG

$$S \rightarrow \neg S \mid (S \Rightarrow S) \mid p \mid q.$$

   The terminals are $\neg, \Rightarrow, (,), p$, and q. This is a CFG that generates well-formed formulae of propositional calculus with predicates p and q. Find a Chomsky normal-form grammar that generates the same language.

3. Let $G = (V, \Sigma, P, <\text{stmt}>)$ be the following grammar:

$$\begin{array}{rcl}
<\text{stmt}> & \rightarrow & <\text{assignment}> \mid <\text{if-then}> \mid <\text{if-then-else}> \mid \\
 & & <\text{begin-end}> \\
<\text{if-then}> & \rightarrow & \textbf{if condition then } <\text{stmt}> \\
<\text{if-then-else}> & \rightarrow & \textbf{if condition then } <\text{stmt}> \textbf{ else } <\text{stmt}> \\
<\text{begin-end}> & \rightarrow & \textbf{begin } <\text{stmt-list}> \textbf{ end} \\
<\text{stmt-list}> & \rightarrow & <\text{stmt-list}> \; <\text{stmt}> \mid <\text{stmt}> \\
<\text{assignment}> & \rightarrow & \mathbf{a = 1}
\end{array}$$

   $\Sigma = \{\mathbf{if}, \mathbf{condition}, \mathbf{then}, \mathbf{else}, \mathbf{begin}, \mathbf{end}, \mathbf{a = 1}\}$.
   $V = \{<\text{stmt}>, <\text{assignment}>, <\text{if-then}>, <\text{if-then-else}>, <\text{begin-end}>, <\text{stmt-list}>\}$.

   G is a natural-looking grammar for a fragment of a programming language.

   a Show that G is ambiguous.

   b Give a new unambiguous grammar for the same language.

   c Let symbols like **if**, **then** etc. be assigned their usual meanings. Re-state (in words) your modification in part (b) with respect to this meaning. (*Note: Compiler designers have to worry about such semantic effects —sometimes unintended— of the way they define a CFG.*)

4. Let $\Sigma = \{0, 1, \sharp\}$.

a Show that $\{x\sharp y : x, y \in \{0,1\}^*, x \neq y\}$ is context free.

b Show that $\{x\sharp y : x, y \in \{0,1\}^*, x = y\}$ is not context-free.

c Using parts (a) and (b), argue (carefully!) that the set of context-free languages is not closed under complementation.

5. Let $\Sigma = \{0, 1, \sharp\}$.

   a Show that $\left\{w\sharp w^R\sharp w : w \in \{0,1\}^*\right\}$ is not context-free.

   b Express the language in part (a) as $A \cap B$, where A and B are context-free. Note that you have shown that the set of CFL's is not closed under intersection.

6. Briefly desribe an algorithm that performs the following task. Given a CFG G, and a string w of length n, the algorithm decides whether or not G generates w. The algorithm must run in time $O(n^3)$. A Hint: Convert G to Chomsky normal form and use dynamic programming[1]

7. Let $G = (V, \Sigma, S, R)$ be a context-free grammar in Chomsky Normal Form. We define a directed graph whose set of vertices is V (the set of variables) and whose set of edges is

   $$\{(A, B) : A, B \in V \text{ and for some } C \in V, \text{ either } A \to BC \text{ or } A \to CB \text{ is in } P\}.$$

   We say that this graph is *acyclic* if it has no (directed) cycle (and no self-loops either).

   Show that if this graph is acyclic then grammar G generates a regular language.

8. (*Optional*) Show that the context-free language $\{0^m 1^n 2^s : \text{either } m = n \text{ or } n = s\}$ is inherently ambiguous (i.e., every grammar for it is ambiguous).

9. (*Optional*) A *deterministic* pushdown automaton is one in which the transition function is deterministic. Show that the language $\left\{ww^R : w \in \{0,1\}^*\right\}$ is not accepted by any *deterministic* pushdown automaton.

[1] Time capsule: In the mid 1960s when this algorithm was discovered, it was considered a breakthrough in compiler design.