Due: Friday, November 13, 2009 in class

Reading assignment: Read Section 2.1 of Sipser’s book and the handout on Chomsky Normal Form.

Problems:

1. Apply the state minimization algorithm to the DFA below. Show each of your steps as in the example on the minimization handout.

![DFA Diagram]

2. Design context-free grammars that generate each of the following languages. Justify your grammar designs.
   (a) The set \( \{ w \in \{0, 1\}^* \mid w = w^R \} \).
   (b) The complement of the language \( \{ a^n b^n \mid n \geq 0 \} \) in \( \{a, b\}^* \).
   (c) The set \( \{ w \in \{0, 1\}^* \mid w \text{ contains twice as many 1’s as 0’s} \} \).


4. In class we gave the following grammar
   \[ S \rightarrow (S) \mid SS \mid \varepsilon \]
   for the set of strings of balanced parentheses.
   (a) Show that this grammar is ambiguous.
   (b) Give a new unambiguous grammar for the same language.
5. Let $G = (V, \Sigma, R, \langle STMT \rangle)$ be the following grammar:

$$\langle STMT \rangle \rightarrow \langle ASSIGN \rangle \mid \langle IF-THEN \rangle \mid \langle IF-THEN-ELSE \rangle$$

$$\langle IF-THEN \rangle \rightarrow \text{if condition then } \langle STMT \rangle$$

$$\langle IF-THEN-ELSE \rangle \rightarrow \text{if condition then } \langle STMT \rangle \text{ else } \langle STMT \rangle$$

$$\langle ASSIGN \rangle \rightarrow a := 1$$

$\Sigma = \{i, f, c, o, n, d, t, h, e, l, s, a, :, =, 1\}$

$V = \{\langle ASSIGN \rangle, \langle STMT \rangle, \langle IF-THEN \rangle, \langle IF-THEN-ELSE \rangle, \langle ASSIGN \rangle\}$

$G$ is a natural-looking grammar for a fragment of a programming language, but $G$ is ambiguous.

(a) Show that $G$ is ambiguous.

(b) Give a new unambiguous grammar for $L(G)$.

6. (Extra credit) A CFG $G = (V, \Sigma, R, S)$ is regular (also known as right-linear) iff every rule of $G$ is of the form $A \rightarrow wB$ or $A \rightarrow w$ for $w \in \Sigma^*$ and $A, B \in V$. In class we showed that every regular language has a regular grammar. Show the converse, namely that for every regular grammar $G$, $L(G)$ is regular, which together with what we showed in class shows that regular grammars generate precisely the regular languages.