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 \( \{ w \in \{0, 1\}^* \mid w \text{ contains more 1's than 0's} \} \).

4. Let $G = (V, \Sigma, R, \langle \text{STMT} \rangle)$ be the following grammar:

$$
\langle \text{ASSIGN} \rangle \rightarrow \langle \text{ASSIGN} \rangle \mid \langle \text{IF-THEN} \rangle \mid \langle \text{IF-THEN-ELSE} \rangle \\
\langle \text{IF-THEN} \rangle \rightarrow \text{if condition then } \langle \text{STMT} \rangle \\
\langle \text{IF-THEN-ELSE} \rangle \rightarrow \text{if condition then } \langle \text{STMT} \rangle \text{ else } \langle \text{STMT} \rangle \\
\langle \text{ASSIGN} \rangle \rightarrow a := 1
$$

$$
\Sigma = \{i, f, c, o, n, d, t, h, e, l, s, a, :, =, 1\} \\
V = \{\langle \text{ASSIGN} \rangle, \langle \text{STMT} \rangle, \langle \text{IF-THEN} \rangle, \langle \text{IF-THEN-ELSE} \rangle, \langle \text{ASSIGN} \rangle\}
$$

$G$ 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 $L(G)$.

5. Convert the following grammar to Chomsky normal form using the procedure on the handout.

$$
S \rightarrow A \mid ABa \mid AbA \\
A \rightarrow Aa \mid \epsilon \\
B \rightarrow Bb \mid BC \\
C \rightarrow CB \mid CA \mid bB
$$


7. (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.