Problem 1: [Sipser] 1.21

part (a): Multiple solutions are possible, depending on the order in which states are eliminated. Eliminating states 1 and 2 in order yields the following regular expression:

$$\mathtt{a}^*\mathtt{b}(\mathtt{a} \cup \mathtt{b}\mathtt{a}^*\mathtt{b})^*.$$

part (b): Multiple solutions are possible, depending on the order in which states are eliminated. Eliminating states 1, 2, and 3 in order yields the following regular expression:

$$\epsilon \cup (\mathtt{a} \cup \mathtt{b})\mathtt{a}^*\mathtt{b}\Big((\mathtt{b} \cup \mathtt{a}(\mathtt{a} \cup \mathtt{b}))\mathtt{a}^*\mathtt{b}\Big)^*(\epsilon \cup \mathtt{a})$$

Problem 2: [Sipser] 1.29(b)

For the sake of contradiction, assume that the language is regular. The Pumping Lemma must then apply; let k be the pumping length. Consider the string $w = \mathtt{a}^k\mathtt{b}\mathtt{a}^k\mathtt{b}\mathtt{a}^k\mathtt{b} \in A_2$. Since $|w| \geq k$, it must be possible to split w into three pieces xyz satisfying the conditions of the Pumping Lemma. In any legal split $xyz = w$, it must be the case that $|xy| \leq k$. Therefore, y must contain only $\mathtt{a}$ symbols. Now consider the string $xyyz$. The string $xyyz$ is of the form $\mathtt{a}^{k+|y|}\mathtt{b}\mathtt{a}^k\mathtt{b}\mathtt{a}^k\mathtt{b}$, and since $|y| > 0$, this string is no longer an element of A_2. Thus, the Pumping Lemma is violated, and the language in question cannot be regular.

Problem 3: Use the pumping lemma for regular languages to show that the following language is not regular: $L = \{\mathtt{a}^i\,\mathtt{b}^j \mid i > 3j\}$.

For the sake of contradiction, assume that the language is regular. The Pumping Lemma must then apply; let k be the pumping length. Consider the string $w = \mathtt{a}^{3k+1}\mathtt{b}^k \in L$. Since $|w| \geq k$, it must be possible to split w into three pieces xyz satisfying the conditions of the Pumping Lemma. In any legal split $xyz = w$, it must be the case that $|xy| \leq k$. Therefore, y must contain only $\mathtt{a}$ symbols. Now consider the string xz, which is obtained by pumping *down*, i.e. setting $i = 0$ in the statement of the Pumping Lemma. The string xz is of the form $\mathtt{a}^{3k+1-|y|}\mathtt{b}^k$, and since $|y| > 0$, this string is no longer an element of L. Thus, the Pumping Lemma is violated, and the language in question cannot be regular.

Problem 4: [Sipser] 2.4 (b), (c), (e)

part (b): A valid *regular expression* for this language is $(0\Sigma^*0) \cup (1\Sigma^*1) \cup \varepsilon$. We can construct a context-free grammar for this language as follows.

$$\begin{array}{lcll} S & \to & 0A0 \mid 1A1 \mid \varepsilon & \triangleright \text{ Generates } (0\Sigma^*0) \cup (1\Sigma^*1) \cup \varepsilon. \\ A & \to & 0A \mid 1A \mid \varepsilon & \triangleright \text{ Generates } \Sigma^*. \end{array}$$

part (c): A valid *regular expression* for this language is $(\Sigma\Sigma)^*\Sigma$. We can construct a context-free grammar for this language as follows.

$$\begin{array}{lcll} S & \to & AAS \mid A & \triangleright \text{ Generates } (\Sigma\Sigma)^*\Sigma. \\ A & \to & 0 \mid 1 & \triangleright \text{ Generates } \Sigma. \end{array}$$

part (e): All strings in this language are of the form xx^R, $x0x^R$, or $x1x^R$ for some string $x \in \Sigma^*$. We can construct a context-free grammar for this language as follows.

$$S \to 0S0 \mid 1S1 \mid 0 \mid 1 \mid \epsilon$$

Problem 5: [Sipser] 2.9

All strings in this language are of the form $\mathtt{a}^n\mathtt{b}^n\mathtt{c}^*$ or $\mathtt{a}^*\mathtt{b}^n\mathtt{c}^n$ for some $n \geq 0$. We can construct a context-free grammar for this language as follows.

$$\begin{array}{lcll} S & \to & EC \mid AD & \triangleright \text{ Generates } \mathtt{a}^n\mathtt{b}^n\mathtt{c}^* \cup \mathtt{a}^*\mathtt{b}^n\mathtt{c}^n. \\ E & \to & \mathtt{a}E\mathtt{b} \mid \varepsilon & \triangleright \text{ Generates } \mathtt{a}^n\mathtt{b}^n. \\ D & \to & \mathtt{b}D\mathtt{c} \mid \varepsilon & \triangleright \text{ Generates } \mathtt{b}^n\mathtt{c}^n. \\ C & \to & \mathtt{c}C \mid \varepsilon & \triangleright \text{ Generates } \mathtt{c}^*. \\ A & \to & \mathtt{a}A \mid \varepsilon & \triangleright \text{ Generates } \mathtt{a}^*. \end{array}$$

This grammar is ambiguous because the string abc has two distinct leftmost derivations.

$$S \to EC \to \mathtt{a}E\mathtt{b}C \to \mathtt{ab}C \to \mathtt{abc}C \to \mathtt{abc}$$

$$S \to AD \to \mathtt{a}AD \to \mathtt{a}D \to \mathtt{ab}D\mathtt{c} \to \mathtt{abc}$$

In fact, one can show that the context-free language in question is *inherently ambiguous*, i.e., that there cannot exist an unambiguous grammar for this language.

Problem 6: Consider the following CFG grammar: $S \to \mathtt{a}S\mathtt{a}S \mid \mathtt{a}S\mathtt{a} \mid \varepsilon$. Show that the grammar is ambiguous.

This grammar is ambiguous because the string aa has two distinct leftmost derivations.

$$S \to \mathtt{a}S\mathtt{a}S \to \mathtt{aa}S \to \mathtt{aa}$$

$$S \to \mathtt{a}S\mathtt{a} \to \mathtt{aa}$$