Problem 3.1 (10 points)

[Cassandras and LaFortune] Answer true or false.

(a) $baa \in a^*b^*a^*b^*$
(b) $b^*a^* \cap a^*b^* = a^* \cup b^*$
(c) $a^*b^* \cap c^*d^* = \emptyset$
(d) $abcd \in (a(cd)^*b)^*$

Problem 3.2 (10 points)

For each of the following languages, give the two least strings that are members and the two least strings that are not members—a total of four strings for each part, where the term least string implies ordering “by least length, then by alphabetical order”. Thus,

$$\epsilon < a < b < aa < ab < ba < bb < aaa < aab < \ldots$$

Assume the alphabet $\{a, b\}$ in all parts.

(a) $ab + aab$
(b) $b^*(ab)^*a^*$
(c) $(a^* + b^*)(a^* + b^*)(a^* + b^*)$
(d) $a^*(baa^*)^*b^*$
(e) $b^*(a + ba)^*b^*$

Problem 3.3 (10 points)

[Cassandras and LaFortune] Consider the event set $E = \{a, b, g\}$. Find:

(a) A regex for the language with each string containing at least one $b$.
(b) A regex for the language with each string containing exactly two $b$’s.
(c) A regex for the language with each string containing at least two $b$’s.

Problem 3.4 (10 points)

[Sipser] Give regexes for the following languages. In all cases, the the alphabet is $B = \{0, 1\}$.

(a) $\{w \mid w \text{ begins with a } 1 \text{ and ends with a } 0\}$.
(b) $\{w \mid w \text{ contains at least three } 1\text{s}\}$.
(c) $\{w \mid w \text{ contains the substring } 0101\}$.
(d) $\{w \mid w \text{ has length at least three and its third symbol is a } 0\}$.
(e) $\{w \mid w \text{ starts with } 0 \text{ and has odd length, or starts with } 1 \text{ and has even length}\}$. 
(f) \( \{w \mid \text{the length of } w \text{ is at most 5} \} \).

(g) \( \{w \mid \text{every odd position of } w \text{ is a 1} \} \).

(h) \( \{w \mid \text{w contains an even number of zeros, or exactly two 1s} \} \).

(i) All strings except the empty string.

**Problem 3.5 (20 points)**

Do Problem 2.26 of *Cassandras and LaFortune*.

**Problem 3.6 (10 points)**

[Hopcroft, Motwani, and Ullman] Consider the following \( \epsilon \)-NFA:

\[
\begin{array}{c|c|c|c|c}
   & \epsilon & a & b & c \\
\hline
\rightarrow p & \{q, r\} & \emptyset & \{q\} & \{r\} \\
q & \emptyset & \{p\} & \{r\} & \{p, q\} \\
Fr & \emptyset & \emptyset & \emptyset & \emptyset \\
\end{array}
\]

(a) Compute the \( \epsilon \)-closure of each state.

(b) Give all strings of length three or less accepted by the automaton.

(c) Convert the automaton to a DFA.

**Problem 3.7 (10 points)**

Do Problem 2.30 of *Cassandras and LaFortune*.

**Problem 3.8 (20 points)**

Do Problem 2.18 of *Cassandras and LaFortune*.

**Problem 3.9 (20 points)**

Consider the mutual exclusion problem, shown in Figure 1. Assume there are two trains, \( T_1 \) and \( T_2 \), traveling on the left and right tracks, in clockwise and counterclockwise directions, respectively. The requirement is that the trains are to have maximal freedom of movement subject to the constraints that (i) each train must always be able to eventually return to its source (and initial state), \( S_i \), and (ii) the shared section of track, \( R \), can only be used by one train at a time.

You may assume that, within the direction-of-travel constraint, each train’s action is completely controllable. Also, you may assign events which serve to mark the passage of a train over certain points—also assignable—on its track.

![Figure 1: The Two-Trains Mutual Exclusion Problem](image)

Given the above problem statement:
• Model each train/track by a DES. Hint: Add events $\delta_i$, $\sigma_i$, and $\rho_i$, that occur when train $i$ passes points on the track near (i.e., enters the track segments of) $D_i$, $S_i$, and $R$, respectively.

• Find the language of $T_i$, $i = 1, 2$.

• Find the marked language of $T_i$, $i = 1, 2$.

• Find the shuffle product DES representing the coupled system.

• Find a language $K$ that captures the given constraints.

• Design a supervisor satisfying the requirement.

---

**Problems 3.G1 (10 points)**

Do Problem 2.23, parts (a), (c), and (d) only, of *Cassandras and LaFortune*.

**Problem 3.G2 (10 points)**

Do Problem 2.31 of *Cassandras and LaFortune*.

**Problem 3.G3 (10 points)**

Do Problem 2.36 of *Cassandras and LaFortune*.

**Problem 3.G4 (20 points)**

Re-solve Problem 3.9(d)–(f) using the library of routines UMDES-LIB.

Note: the parallel composition ($\text{par\_comp}$) is the same as the shuffle product when the two machines have no events in common.

Executables of the UMDES-LIB (for Solaris, Linux, Mac, and Windows), as well as an Overview with examples is at

http://www.eecs.umich.edu/umdes/toolboxes.html

**3.G5 (10 points)**

Use the UMDES-LIB to re-solve 3.G2 above.