QUESTION 1. [10 pts] Give a regular expression for the following language:

\[ L = \{ x \mid x \in \{0, 1\}^*, x \text{ does not begin with } 010 \} \]

Answer:

\[ 1(0 + 1)^* + 00(0 + 1)^* + 011(0 + 1)^* + \epsilon + 0 + 01 \]

The regex is not unique. Give partial credits for regex’s containing some correct components.
QUESTION 2. [10 pts] Following the procedure given in class, convert the following regular expression to an $\epsilon$-NFA:

$$R = (10)^* + 101$$

Answer:

<table>
<thead>
<tr>
<th></th>
<th>$\epsilon$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rightarrow q_0$</td>
<td>${q_1, q_7}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>${q_2, q_5}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>${q_3}$</td>
</tr>
<tr>
<td>$q_3$</td>
<td>$\emptyset$</td>
<td>${q_4}$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$q_4$</td>
<td>${q_2, q_5}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$q_5$</td>
<td>${q_6}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$q_6$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$q_7$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>${q_8}$</td>
</tr>
<tr>
<td>$q_8$</td>
<td>$\emptyset$</td>
<td>${q_3}$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$q_9$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>${q_{10}}$</td>
</tr>
<tr>
<td>$q_{10}$</td>
<td>${q_6}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

Note that the answer may not be unique. For example, some of the states can be merged to simplify the $\epsilon$-NFA without losing the correctness. Give most partial credits as long as the final $\epsilon$-NFA works.
QUESTION 3.

1. [10 pts] Convert the following DFA to a regular expression by using the state elimination algorithm:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>→ *q₀</td>
<td>q₀</td>
<td>q₁</td>
</tr>
<tr>
<td>q₁</td>
<td>q₂</td>
<td>q₀</td>
</tr>
<tr>
<td>q₂</td>
<td>q₁</td>
<td>q₂</td>
</tr>
</tbody>
</table>

2. [5 pts] What is the language accepted by the above DFA? You may describe the language by giving the (mathematical) property of its strings.

Answer:

1. After eliminating state q₂, the arc label from q₁ to q₁ becomes 01*0. (5 pts)

   After eliminating state q₁, the arc label from q₀ to q₀ becomes 0 + 1(01*0)*1, which leaves the final answer as (0 + 1(01*0)*1)*. (5 pts)

2. Binary numbers divisible by 3.

   Give partial credits for correct steps. If state q₁ is eliminated first, the final answer will be (0 + 11 + 10(00 + 1)*01)*.
QUESTION 4. [10 pts] Prove that the following language is not regular using the Pumping Lemma:

\[ L = \{0^i1^j2^j \mid i, j \geq 0\} \]

Answer:

Let \( n \) be the constant in the Pumping Lemma. Consider \( w = 0^{2n}1^n2^n \in L \). Let \( w = xyz \) be any partition satisfying (i) \(|y| > 0 \) and \(|xy| \leq n\). Clearly, \( x = 0^i \) and \( y = 0^j \) for some \( i \geq 0 \) and \( j > 0 \). Then \( xyyz = 0^{2n+j}1^n2^n \notin L \), and thus a contradiction.

Give partial credits for correct/reasonable steps.
QUESTION 5. [10 pts] Convert the following DFA to the minimum-state equivalent DFA step-by-step using the TF algorithm.

\[
\begin{array}{c|cc}
&a & b & c \\
\hline
q_0 & q_1 & q_2 \\
q_1 & q_0 & q_2 \\
q_2 & q_3 & q_0 \\
q_3 & q_2 & q_4 \\
q_4 & q_2 & q_5 \\
q_5 & q_2 & q_3 \\
\end{array}
\]

Answer:

The DFA in Q3. Must show the state equivalence table first.

Give partial credits for correct steps.
QUESTION 6. [10 pts] Give a context-free grammar for the following language:

\[ L = \{ 0^i1^j2^k3^n \mid i, j, k, n \geq 0; i = j + 2k + 3n \} \]

Answer:

\[ G = (V, \{0, 1, 2, 3\}, P, S) \], where \( V = \{S, A, B\} \) and \( P \) contains the following rules: (4 pts)

\[
\begin{align*}
S & \rightarrow 000S3A \\
A & \rightarrow 00A2B \\
B & \rightarrow 0B1\epsilon \\
\end{align*}
\]

(6 pts)
QUESTION 7. [10 pts] Let $\#_0(x)$ and $\#_1(x)$ denote the numbers of 0’s and 1’s in a binary string $x$, respectively. Design a PDA to accept the following language by empty stack.

$$L = \{x \mid x \in \{0,1\}^*, \#_0(x) = \#_1(x)\}$$

Answer:

The following is a PDA that accepts $L$ by empty stack. Note that, it has no $\epsilon$-moves when symbols $O$ or $I$ are on the top of the stack.

<table>
<thead>
<tr>
<th>$\rightarrow$</th>
<th>$0, Z_0$</th>
<th>$0, O$</th>
<th>$0, I$</th>
<th>$1, Z_0$</th>
<th>$1, O$</th>
<th>$1, I$</th>
<th>$\epsilon, Z_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rightarrow q$</td>
<td>$q, OZ_0$</td>
<td>$q, OO$</td>
<td>$q, \epsilon$</td>
<td>$q, 1Z_0$</td>
<td>$q, \epsilon$</td>
<td>$q, II$</td>
<td>$q, \epsilon$</td>
</tr>
</tbody>
</table>