Problem 1 [15 points]

Prove that, for any deterministic FSA $A = (Q, \Sigma, \delta, q_0, F)$,

$$\hat{\delta}(q, wy) = \hat{\delta} \left( \hat{\delta}(q, w), y \right)$$

for $w, y \in \Sigma^*$. Use the definition of $\hat{\delta}$ provided in lecture:

1. $\hat{\delta}(q, \epsilon) = q$
2. $\hat{\delta}(q, x\sigma) = \delta \left( \hat{\delta}(q, x), \sigma \right)$

where $\epsilon$ is the empty string, $x \in \Sigma^*$, and $\sigma \in \Sigma$.

Problem 2 [15 points]

Note: The following problems require you to design finite state machines. We ask that you do not use any automated tools to solve these problems as you will not have access to such tools for similar problems on the exams.

Give deterministic FSAs accepting the following languages. Please make sure your DFSAs are fully defined: clearly mark start and final states and do not use shorthand notation for arcs.

1. [5 points] The set of strings over $\{a, b, c\}$ in which the substring $bc$ never occurs.
2. [5 points] The set of strings over $\{1, 7\}$ ending in 11711.
3. [5 points] The set of strings over $\{0, 1\}$ which are divisible by three when interpreted as a binary number (ignoring leading zeroes).

For example $00000_b = 0$, which is divisible by 3, so 00000 should be accepted. $001001_b = 9$, and thus should be accepted also. $101_b = 5$, is not divisible by 3, and thus should be rejected.

Note: $\epsilon$ should be interpreted as 0, and thus should be accepted.
Problem 3 [10 points]

Give a non-deterministic FSA (possibly with \( \epsilon \) moves) accepting the following language. Please clearly mark start and final states.

1. The set of strings over \( \{1, 7\} \) ending in 11711. How does your NDFSA differ from the DFSA you constructed in Problem 2.2?

Problem 4 [15 points]

Give regular expressions for each of the following languages:

1. [7 points] The set of strings over \( \{a, b, c\} \) in which the substring \( bc \) never occurs.

2. [8 points] The language described in Problem 2.3: The set of strings over \( \{0, 1\} \) which are divisible by three when interpreted as a binary number (ignoring leading zeroes).
   
   For example, \( 00000_2 = 0 \), which is divisible by 3, so \( 00000 \) should be accepted. \( 001001_2 = 9 \), and thus should be accepted also. \( 101_2 = 5 \), is not divisible by 3, and thus should be rejected.
   
   Note: \( \epsilon \) should be interpreted as 0, and thus should be accepted.

Problem 5 [20 points]

Use the Pumping Lemma for regular languages to show that the following language over \( \Sigma = \{a, b\} \) is not regular: strings consisting of an equal number of \( a \)'s and \( b \)'s such that no prefix of the string contains more \( b \)'s than \( a \)'s. Intuitively, this is the language of balanced parentheses (imagine \( a \) is open parenthesis and \( b \) is close parenthesis).

Problem 6 [25 points]

For each of the following statements, answer whether the claim is true or false, and give a short (two- to three-sentence) explanation if true or counterexample if false.

1. [5 points] Let \( L_1 \subset L_2 \). If \( L_1 \) is not regular, then \( L_2 \) must also be not regular.

2. [5 points] \( L = L_1 \cup L_2 \). If \( L_1 \) and \( L \) are regular languages, then \( L_2 \) must also be a regular language.

3. [5 points] \( L = \cap_{i=1}^{\infty} L_i \). If all of the \( L_i \) are regular languages, then \( L \) is also a regular language.

4. [5 points] Let \( \alpha \) and \( \beta \) be regular expressions. \( L((\alpha^* \beta)^* \alpha^*) = L((\alpha + \beta)^*) \)

5. [5 points] Let \( \alpha, \beta \) and \( \gamma \) be regular expressions.
   
   If \( L(\beta + \alpha \gamma) \subseteq L(\gamma) \), then \( L(\alpha^* \beta) \subseteq L(\gamma) \).