1. [20pts] True (mark 0 ) or false (mark x) questions:
   (a) If A and B are regular, then so is A – B.
   (b) A(B∩C) = AB ∩ AC
   (c) A⊆ B implies A* ⊆ B*
   (d) If A is regular, then { xx | x ∈ A} is regular.
   (e) If A is regular, then { x | xx ∈ A} is regular.
   (f) Every finite language is regular.
   (g) ((A*)*) * = A*
   (h) { ε } ∪ A^2 ⊆ A implies A = A*.
   (i) a(ba)* = (ab)*a
   (j) There is a set that is accepted by a NFA but could not be accepted by any DFA.
   Ans: 0 x 0 x 00000 x

2. (6 pts) Let Σ = {a, b}, A = {ab, bc, ca}, B={ac, cb }.
   (a) AB = _{abac, abcb, bcac, bccb, caac, c acab}_.
   (b) Σ^3 - A = _{aa, bb, ba } Σ = {aaa, bba, baa, aab, bbb, bab}_.

3. (35 pts) Given the following NFA machine M :

   (a) Give a string of length 5 beginning with a and ending with b that is not accepted by
      M. [ 5pts]
      Ans: abbbb

   (b) Fill the following state transition table for the NFA M. Assume the states from left to
      right shown in the diagram are p, q, r and s, respectively. [5 pts]

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>pQRS</td>
<td>{}</td>
</tr>
<tr>
<td>q</td>
<td>{}</td>
<td>p</td>
</tr>
<tr>
<td>r</td>
<td>{}</td>
<td>q</td>
</tr>
<tr>
<td>s</td>
<td>{}</td>
<td>r</td>
</tr>
</tbody>
</table>
(c) Let $M = (Q, \Sigma, \delta, S, F)$. Then what are the contents of $Q$, $\Sigma$, $\delta$, $S$ and $F$, respectively.

\[5\text{ pts}\]

$Q = \{p, q, r, s\}$ $\Sigma = \{a, b\}$

$S = \{p\}$ $F = \{p\}$

$\delta = \delta(p,a) = \{p, q, r, s\}; \quad \delta(p,b) = \{}$

$\delta(q,a) = \delta(r,a) = \delta(s,a) = \{\}$

$\delta(q,b) = \{p\}, \quad \delta(r,b) = \{q\}, \quad \delta(s,b) = \{r\}$

(d) List all the strings of length 3 accepted by $M$. [5 pts]

ANS: \{aaa, aab, cba, abb\}

(e) Find a regular expression equivalent to $M$. [5 pts]

Ans: $(a+ab+abb+abbb)^*$

(f) Convert the automata $M$ to a DFA $M'$ by filling in the following table: [10 pts]

<table>
<thead>
<tr>
<th>$M'$</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Rightarrow {p}$</td>
<td>pqrs</td>
<td>{}</td>
</tr>
<tr>
<td>q</td>
<td>{}</td>
<td>p</td>
</tr>
<tr>
<td>r</td>
<td>{}</td>
<td>q</td>
</tr>
<tr>
<td>s</td>
<td>{}</td>
<td>r</td>
</tr>
<tr>
<td>pqrs F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pq F</td>
<td>pqrs</td>
<td>pq</td>
</tr>
<tr>
<td>p</td>
<td>pqrs</td>
<td>p</td>
</tr>
<tr>
<td>pqr F</td>
<td>pqrs</td>
<td>pq</td>
</tr>
<tr>
<td>pq F</td>
<td>pqrs</td>
<td>p</td>
</tr>
<tr>
<td>p</td>
<td>pqrs</td>
<td>p</td>
</tr>
</tbody>
</table>
4. [15 pts] Consider the following regular set R represented by the regular expression:
$a^*b^* + b*a^*$.

(a) Draw a NFA for R. [5 pts]

(b) Draw a minimized DFA for R. [10 pts]

5. [15 pts] Let $A$ be an infinite subset of the set \{a^{k}b^{k} \mid k > 0 \}. Using (game-theoretic) pumping lemma to show that $A$ is not regular.

pf: 1. D pick a number $k > 0$.
2. Y choose a number $n$ with the property that $n > k$ and the string $\alpha = a^n b^n \in A$. Since $A$ is infinite, there must exist such $n$ for any $k$ picked by D. Let $\alpha = x y z$ where $x = a^n , y = b^n$ and $z = \varepsilon$.
3. Suppose D decompose $y = b^n$ into uvw with $|v| \neq 0$.
4. Y let $i = 0$.
Now the resulting string is $xuv^i z = a^n b^n \cdot 1 \notin A$. Since Y always win the game, we thus conclude that $A$ is not regular.
6. [14pts] Consider the following CFG:
   \[ S \rightarrow AB | A \rightarrow a | B \rightarrow bA \]
   (a) Find a regular expression equivalent to the grammar. [6pts]
   (b) Find a derivation for the input ‘abaaba’ [4 pts]
   (c) Find all strings in \{aabaab, aaaaba, aabbaa, ababab\} that could be generated by G. [4pts]

ANS: (a) \( L(S) = (AB)^* = (a^*ba^*)^* \).
   (b) \[ S \rightarrow AB \rightarrow ABAB \rightarrow aBAB \rightarrow abAAB \rightarrow abaAB \rightarrow abaaB \rightarrow abaabA \rightarrow abaaba \]
   (c) aaaaba

7. [15pts] Let \( \Sigma \) be a finite alphabet. For any set \( A \subseteq \Sigma^* \), \( a \in \Sigma \), define
   \begin{align*}
   \text{permute}(A) &= \{ y \mid \exists x \in A \text{ s.t. } y \text{ is a permutation of } x \}, \\
   \text{project}(A, a) &= \{ a^p \mid \exists x \in A \text{ s.t. } p \text{ is the number of } a \text{'s occurring in } x. \}.
   \end{align*}
   For instance, if \( A = \{aab, ab\} \), then \( \text{permute}(A) = \{ab, ba, aab, aba, baa\} \) and \( \text{project}(A, a) = \{aa, a\} \).
   (a) Show that if \( A \) is regular, then so is \( \text{project}(A, a) \). Hint: How to get \( \text{project}(A, a) \) by applying a homomorphism to \( A \)? [7 pts]
   (b) Show that if \( A \) is regular, then so is \( \text{permute}(A) \). [8 pts]

pf: (a) Define a homomorphism \( h: \Sigma^* \rightarrow \Sigma^* \) as follows:
   \[ h(a) = a \text{ and } h(x) = \varepsilon \text{ if } x \in \Sigma - \{a\}. \]
   For any \( \alpha \in \Sigma^* \), we have \( h(\alpha) = a^p \) where \( p \) is the number of \( a \)'s occurring in \( \alpha \).
   As a result, \( h(A) = \text{project}(A,a) \). Since homomorphism preserves regularity of a language, \( \text{project}(A,a) \) thus is regular.

(b) \( \text{permute}(A) \) can be accepted by the product machine \( \Pi_{a \in \Sigma} M_a \), where the machine \( M_a \) is essentially similar to the one for computing \( \text{project}(A, a) \) and is constructed from \( M \) as follows: on input \( x \), \( M_a \) will
   1. behaves the same as \( M \) if the scanned symbol is \( a \) (i.e., \( \delta_{M_a}(p, a) = \delta_M(p, a) \))
   2. and goes to any state that could be reached via any symbol \( c \neq a \) if the scanned symbol is \( b \neq a \). I.e., \( \delta_{M_a}(p, b) = \bigcup_{c \in \Sigma - \{a\}} \delta_M(p, c) \) for all \( b \neq a \).
(c) [10pts] A context-free grammar is said to be bilinear if every production is in one of the following forms: $A \rightarrow aB$, $A \rightarrow Ba$, $A \rightarrow B$, $A \rightarrow \varepsilon$, where $A$ and $B$ are non-terminals and $a, b$ are any terminals. Bilinear grammar can accept non-regular sets and hence is more expressive than linear grammar. Now you are asked to design a bilinear grammar containing no more than 3 nonterminals that accepts the non-regular set $L_1 = \{a^n b^m \mid 0 \leq n \leq m \}$.

Ans:

$$\begin{align*}
S & \rightarrow A \mid S \ b \\
A & \rightarrow aA \ b \mid \varepsilon \\
\underline{aA} & \rightarrow a \ A
\end{align*}$$