Instructions : **Answer All Questions (5)**

1. Answer with short justifications (example or counter example, as the case may be), whether the following **claims** are **true** or **false**. No credit will be given for writing only *true* or *false*. Assume that the alphabet is $\Sigma = \{0, 1\}$ unless specified otherwise.

   (a) $A \subseteq \Sigma^*$ is said to be *co-finite* if $\Sigma^* - A$ is *finite*.
   Claim: The size of $2^{\Sigma^*}_{\textit{co-finite}} = \{A \subseteq \Sigma^* : A \text{ is } \textit{co-finite}\}$ is *uncountably infinite*.
   Ans. (false) Let there be k elements in Σ. Any string over Σ may be viewed as a numeral over *radix*-$(k+1)$ number system e.g. $\Sigma = \{a, b, c\} \equiv \{1, 2, 3\}$. So '*aacb*' may be viewed as 1132, a radix-4 numeral that is equal to 94 in decimal (ε may be treated as 0).
   Any finite subset of Σ^* may be viewed as a finite subset of $\mathbb{N}$ and can be encoded as an element of $\mathbb{N}$ e.g. $\{3, 11, 5, 0\}$ is encoded as $2^{11} + 2^5 + 2^3 + 2^0 = 2089$. So there is an *one-to-one* map from $2^{\Sigma^*}_{\textit{finite}}$ to $\mathbb{N}$ and $2^{\Sigma^*}_{\textit{finite}}$ is countably infinite. But then there is a natural bijection (by definition) from $2^{\Sigma^*}_{\textit{co-finite}}$ to $2^{\Sigma^*}_{\textit{finite}}$. So $2^{\Sigma^*}_{\textit{co-finite}}$ is also countably infinite.

   (b) **Claim:** Every finite subset of Σ^* is a *regular set*.
   Ans. (true) Let the finite language $L = \{x_1, x_2, \cdots, x_k\}$. This can be generated by the *right-linear* grammar $S \rightarrow x_1 \mid x_2 \mid \cdots \mid x_k$.

   (c) Let $L_{c1} \subseteq L_{c2} \subseteq \Sigma^*$.
   Claim: If L_{c1} is not a regular set, then L_{c2} can never be a regular set.
   Ans. (false) L_{c2} may be equal to Σ^* which is a regular set. Another example may be - $\{0^n1^n : n > 0\} \subset \{0^m1^n : m, n > 0\} \subset (0 + 1)^*$.

   (d) Let $L_d = \{1^n : n \leq 1000 \text{ and } n \text{ is a prime}\}$.
   Claim: A DFA accepting L_d may have less than 900 states.
   Ans. (false) The last prime less than 1000 is 997 and to detect that the machine should have at least $998 + 1$ states. No loop is possible other than the *sink* state.

   (e) **Claim:** Every regular set can be accepted by some NFA without ε-transition with no more than two final states.
   Ans. (true) We know that an NFA with ε-move can be constructed from a regular expression. Such an NFA may have only one final state. This NFA can be converted to an NFA without ε-move. If ε is not in the language, the new NFA has exactly one final state. But if ε is in the language, the initial state of the NFA is also a final state.

   (f) **Claim:** If we restrict the production rules of a CFG to the form $A \rightarrow \sigma B$ or $A \rightarrow B\sigma$ or $A \rightarrow \sigma$, where $\sigma \in \Sigma$ and A, B are non-terminals, then the generated language of the grammar is regular.
   Ans. (false) Consider the grammar $G = (\{S, A, B\}, \{0, 1\}, P, S)$, where the production rules are:

$$
\begin{aligned}
S &\rightarrow A0 \mid 1B \mid 0 \mid 1 \mid \varepsilon \\
A &\rightarrow 0S \\
B &\rightarrow S1
\end{aligned}
$$

   The language is $\{x \in \Sigma^* : x = x^R\}$.

free (non-right-linear) grammar and a context-sensitive (non-context-free) grammar.

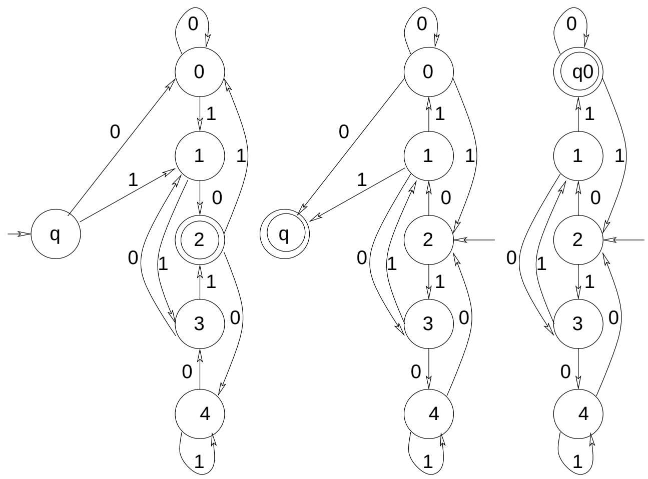
Ans. (true) *Right-linear grammar*: $S \rightarrow 0S \mid 01$,
Context-free grammar: $S \rightarrow 0A1$ and $A \rightarrow 0A \mid \varepsilon$, where S is the start symbol,
Context-sensitive grammar: $S \rightarrow 0A$ and $0A \rightarrow 00A \mid 01$.

(h) Let $Inf_\Sigma = \{L \subseteq \Sigma^* :$ number of strings in L is infinite$\}$.
Claim: Inf_Σ is closed under union, intersection, and complementation.

Ans. (false) It is closed under union but not under intersection or complementation. $\Sigma^* \in Inf_\Sigma$, but its complement set is the *null set* and not in Inf_Σ. **[8 × 3]**

2. Design a DFA for the language $L_2 = \{x \in \{0,1\}^* : x$ is interpreted as a unsigned binary numeral and $x \bmod 5 = 2\}$. From the DFA of L_2, design an NFA (without ε-transition) for L_2^R (reverse of L_2). Convert the NFA to a DFA. **[6]**

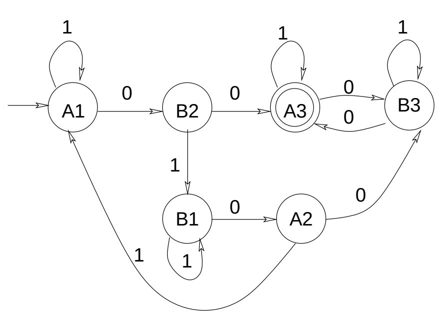
Ans. Following (from left to right) is a DFA for L_2, corresponding NFA for L_2^R and the DFA of L_2^R. It is not necessary to have a separate state q in the original DFA. The state



0 may be the *initial state*.

3. Design a DFA that accepts the language $L_3 = \{x \in \{0,1\}^* : |x|_0$ is even and '00' is a substring of $x\}$. Write down the equations of states and find the regular expression for L_3. **[5]**

Ans. The DFA is



The equations are

$$a_1 = 1a_1 + 0b_2$$

$$
\begin{aligned}
a_3 &= 1a_3 + 0b_3 + \varepsilon \\
b_1 &= 1b_1 + 0a_2 \\
b_2 &= 0a_3 + 1b_1 \\
b_3 &= 1b_3 + 0a_3
\end{aligned}
$$

From $a_3 = 1a_3 + 0b_3 + \varepsilon$ we get $a_3 = 1^*(0b_3 + \varepsilon) = 1^*0b_3 + 1^*$. Again $b_3 = 1b_3 + 0a_3 = 1b_3 + 0(1^*0b_3 + 1^*) = (1 + 01^*0)b_3 + 01^*$. So $b_3 = (1 + 01^*0)^*01^*$. So $a_3 = 1^*0((1 + 01^*0)^*01^*) + 1^*$.
$b_1 = 1b_1 + 0a_2$, so $b_1 = 1^*0a_2$.
$a_2 = 0b_3 + 1a_1 = 0(1 + 01^*0)^*01^* + 1a_1$.
$a_1 = 1a_1 + 0b_2 = 1a_1 + 0(0a_3 + 1b_1) = 1a_1 + 00(1^*0((1 + 01^*0)^*01^*) + 1^*) + 01b_1 = 1a_1 + 00(1^*0((1 + 01^*0)^*01^*) + 1^*) + 011^*0a_2 = 1a_1 + 00(1^*0((1 + 01^*0)^*01^*) + 1^*) + 011^*0(0(1 + 01^*0)^*01^* + 1a_1) = (1 + 011^*01)a_1 + 00(1^*0((1 + 01^*0)^*01^*) + 1^*) + 011^*00(1 + 01^*0)^*01^*$
So the final regular expression is

$$a_1 = (1 + 011^*01)^*00(1^*0((1 + 01^*0)^*01^*) + 1^*) + 011^*00(1 + 01^*0)^*01^*.$$

There may be mistake!

4. Design a CFG for the language $L_4 = \{x_j^R\#\#x_1\#x_2\# \cdots \#x_j\# \cdots \#x_k : k \geq 1, 1 \leq j \leq k, x_i \in \{0,1\}^+ \text{ for } i = 1, \cdots, k\}$, x_j^R is the reverse of x_j. Clearly explain the production rules. **[5]**

   Ans. A CFG is $G = (\{S, A, B, D\}, \{0,1\}, P, S)$, where the production rules are

$$
\begin{aligned}
S &\rightarrow AB \\
A &\rightarrow 0A0 \mid 1A1 \mid 0\#B\#0 \mid 1\#B\#1 \\
B &\rightarrow \#DB \mid \varepsilon \\
D &\rightarrow 0D \mid 1D \mid 0 \mid 1
\end{aligned}
$$

5. If L_1 and L_2 are regular sets, then $L_1 \backslash L_2 = \{x : \exists y \in L_2 \text{ such that } xy \in L_1\}$ is a regular set.
   Given the DFAs of L_1 and L_2, give a construction of a DFA accepting the language $L_1 \backslash L_2$. Clearly explain the construction and justify your claim **[5]**

   Ans. Let $M_1 = (Q_1, \Sigma, \delta_1, q_{10}, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_{20}, F_2)$ be the DFAs corresponding to the languages L_1 and L_2 respectively. Construct the machine $M' = (Q', \Sigma, \delta', q_0', F')$ so that $Q' = Q_1 \times Q_2$, $q_0' = (q_{10}, q_{20})$, $F' = F_1 \times F_2$, and $\delta'((p_1, p_2), a) = (\delta_1(p_1, a), \delta_2(p_2, a))$.
   Consider the set $F = \{p \in Q_1 :$ such that there is a path in M' from $(p, q_{20}) \in Q'$ to $(f_1, f_2) \in F'\}$. This implies that there is a string $y \in \Sigma^*$, so that $\delta_1(p, y) \in F_1$ in M_1, and $\delta_2(q_{20}, y) \in F_2$ i.e. $y \in L_2$. So any string $x \in \Sigma^*$ that takes M_1 from q_{10} to p is in $L_1 \backslash L_2$.
   So the machine for $L_1 \backslash L_2$ is $M = (Q_1, \Sigma, \delta_1, q_{01}, F)$.

Sig.of the Paper-Setter ....