CS21204 : Formal Language and Automata Theory (Spring 2026)

Class Test 1

29-Jan-2026 (Thursday) Maximum Marks: 30 06:30pm – 07:45pm

Roll: ____________ **Name:** ______________________________

[*Write your answers in question paper. Answer all questions. Be brief and precise.*]

Q1. [Languages – Strings and Operations]

Let us define languages L_1, L_2, L_3 as sets of strings over the singleton alphabet $\Sigma = \{c\}$. Present examples (with proper justifications) satisfying the conditions as specified below.

(a) Give an example of a language $L_1 \subseteq \Sigma^*$ such that $L_1^* = L_1^+$. **(2)**

Solution:

Let $L_1 = \{\varepsilon, c\}$.

Then, $L_1^* = \{\varepsilon, c, cc, ccc, \ldots\} = \{c^n \mid n \geq 0\}$ and $L_1^+ = \{\varepsilon, c, cc, ccc, \ldots\} = \{c^n \mid n \geq 0\}$.

Hence, $L_1^* = L_1^+$.

(b) Give an example of a language $L_2 \subseteq \Sigma^*$ such that $L_2 = L_2^*$. **(2)**

Solution:

Let $L_2 = \{\varepsilon, c, cc, ccc, \ldots\} = \{c^n \mid n \geq 0\}$.

Then, $L_2^* = \{\varepsilon, c, cc, ccc, \ldots\} = \{c^n \mid n \geq 0\}$.

Hence, $L_2 = L_2^*$.

(c) Give an example of a language $L_3 \subseteq \Sigma^*$ such that L_3^* is finite. **(2)**

Solution:

Let $L_3 = \{\varepsilon\}$.

Then, $L_3^* = \{\varepsilon\}$.

Hence, L_3^* is finite.

Q2. [NFA and DFA – Construction and Properties]

Consider the following language over $\{a,b\}$ and answer the questions asked below.

$$L_y = \{\ y \mid y \in \{a,b\}^* \text{ and } y \text{ does not start and end with the same symbol } \}$$

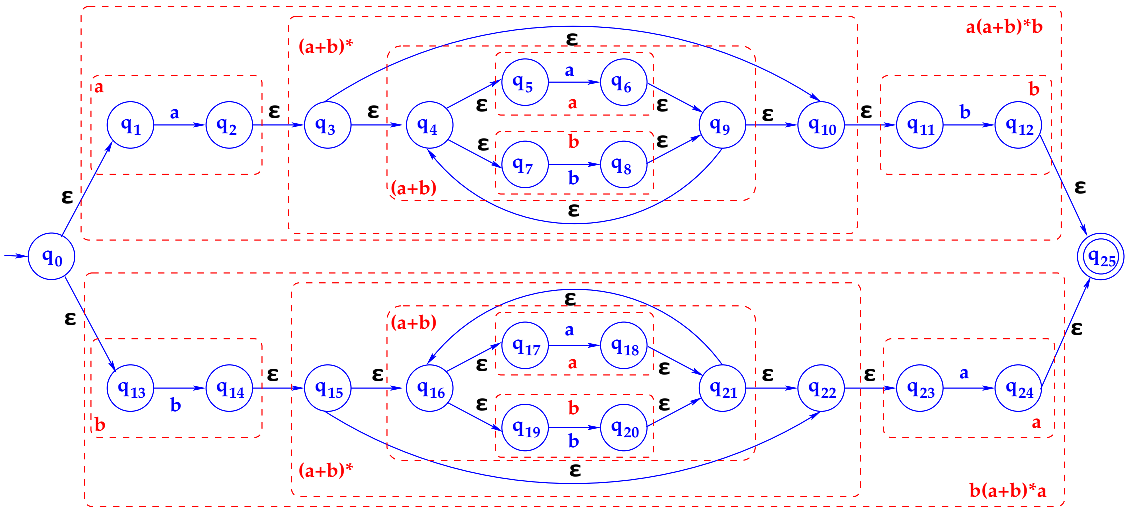
(a) Give a regular expression for the language L_y with a brief justification. **(2)**

Solution:

$$\underbrace{a\,(a+b)^*\,b}_{\substack{\text{strings begin with } a\\ \text{and end with } b}} \quad + \quad \underbrace{b\,(a+b)^*\,a}_{\substack{\text{strings begin with } b\\ \text{and end with } a}}$$

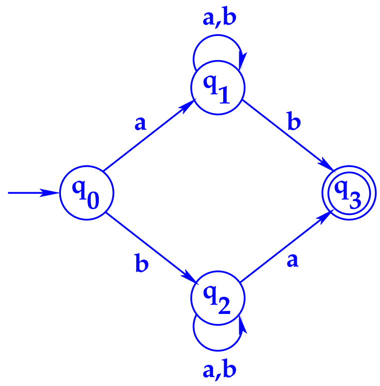
(b) From the regular expression formed in part (a), algorithmically construct a non-deterministic finite automaton (NFA) with ε-transitions accepting the language L_y and show stepwise how you proceed to do the same. Show state-transition diagram(s), not its mathematical definition(s). **(4)**

Solution:



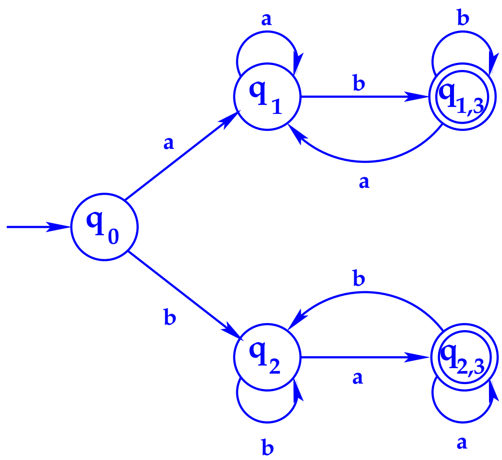
(c) Present a 4-state NFA N (without any ε transition) accepting L_y. Show the state-transition diagram for N, not its mathematical definition. **(2)**

Solution:



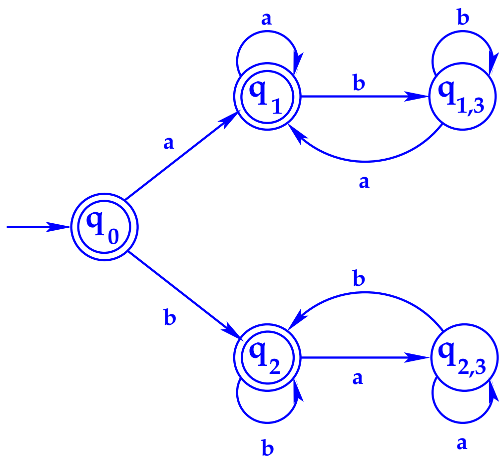
(d) Apply subset construction procedure to convert the above NFA N, that you produced in part (c), to an equivalent DFA D. Show state-transition diagram for D, not its mathematical definition. **(3)**

Solution:



(e) Modify your DFA D, that you constructed in part (d), to create $\overline{D}$ so that it accepts the complement of L_y, i.e. $\overline{L_y} = \{a,b\}^* - L_y$. Show the state-transition diagram for $\overline{D}$, not its mathematical definition. **(2)**

Solution:



Q3. [Regular Expression and DFA]

Consider the following language over $\{0,1,2\}$ and answer the questions asked below.

$$L_z = \left\{ z \;\middle|\; z \in \{0,1,2\}^* \text{ and } z \text{ contains an even number of 0s, } \textbf{or } z \text{ contains exactly two 1s} \right\}$$

(a) Give a regular expression for the language L_z with a brief justification. **(2)**

Solution:

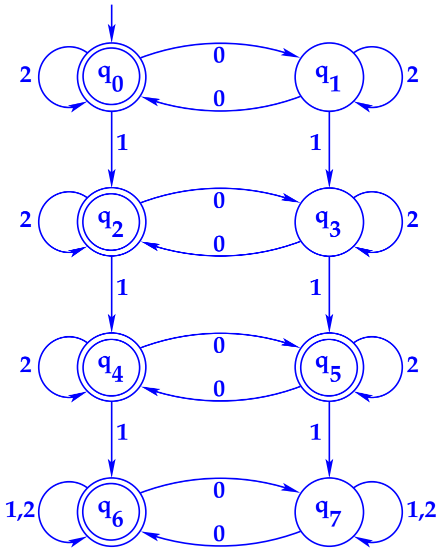
$$\overbrace{\left((1+2)^*0(1+2)^*0\right)^*(1+2)^*}^{\substack{\text{strings repeating even number of 0's} \\ \text{separated (optionally) with 1's or 2's}}} \quad + \quad \overbrace{(0+2)^*1(0+2)^*1(0+2)^*}^{\substack{\text{strings with exactly two 1's} \\ \text{separated (optionally) with 0's or 2's}}}$$

OR

$$\underbrace{(1+2)^*\left(0(1+2)^*0(1+2)^*\right)^*}_{\substack{\text{strings repeating even number of 0's} \\ \text{separated (optionally) with 1's or 2's}}} \quad + \quad \underbrace{(0+2)^*1(0+2)^*1(0+2)^*}_{\substack{\text{strings with exactly two 1's} \\ \text{separated (optionally) with 0's or 2's}}}$$

(b) Present a (DFA) accepting the language L_z. Show the state-transition diagram for the DFA, not its mathematical definition. **(4)**

Solution:



Q4. [Regular Languages]

Suppose that $L \subseteq \{0,1\}^*$ is a regular language. Let us define the following language (set),

$$\mathtt{DEL}_1(L) = \left\{uv \mid u1v \in L \text{ and } u,v \in \{0,1\}^*\right\}$$

The set $\mathtt{DEL}_1(L)$ essentially consists of all strings that can be obtained from strings in L by deleting *exactly* one 1. Prove that $\mathtt{DEL}_1(L)$ is also regular. (Hint: Use non-determinism) **(5)**

Solution:

Let $M = (Q, \Sigma, \delta, s, F)$ be a DFA accepting L where $\Sigma = \{0,1\}$. We construct a NFA $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$ for $\mathtt{DEL}_1(L)$. Define $Q_N = Q \times \{X, Y\}$, $s_N = (s, X)$, $F_N = \{(p, Y) \mid p \in F\}$ and the transition function δ_N as follows: for every $p \in Q$, $b \in \{0,1\}$ let

$$\begin{aligned} \delta_N\big((p,X),\varepsilon\big) &= \Big\{\big(\delta(p,1),Y\big)\Big\} & \qquad \delta_N\big((p,Y),\varepsilon\big) &= \phi \\ \delta_N\big((p,X),b\big) &= \Big\{\big(\delta(p,b),X\big)\Big\} & \qquad \delta_N\big((p,Y),b\big) &= \Big\{\big(\delta(p,b),Y\big)\Big\} \end{aligned}$$

The machine N basically chooses the 1 to delete non-deterministically. State of the form (p, X) indicates that the 1 has not yet been chosen. There is an ε-transition from every such state to state (q, Y) so that M makes a move from state p to state q upon reading 1. Here, the second component in (q, Y) indicates 1 has been seen and skipped/deleted. This ensures that the computation of M processing that particular occurrence of 1 is ignored. All other transitions remain activated similarly as they were in M. Also once a 1 has been deleted, the machine N behaves exactly as M, that is, there are no ε-transitions from states of the form $(p, 1)$. A string is accepted if after processing it, N reaches an accepting state after deleting a 1 i.e., it reaches a state of the form (p, Y) where $p \in F$. This proves that $\mathtt{DEL}_1(L)$ is also regular as it is accepted by N.

— End of Question Paper —