Spring25

Monday, July 28, 2025 8:17 AM

Set A

Problem 1 (CO1): DFA and Regular Languages (15 points)

Let $\Sigma = \{a, b\}$. Consider the following languages over Σ.

$$L_1 = \{w : \text{length of } w \text{ is three more than multiple of four}\}$$
$$L_2 = \{w : \text{every even position letter in } w \text{ is the same as the first letter of } w\}$$
$$L_3 = \{w : \text{every } 2k+1 \text{ position in } w \text{ is a, where } k \geq 0\}$$
$$L_4 = \{w : \text{every } 2k+1 \text{ position in } w \text{ is b, where } k \geq 0\}$$

Now solve the following problems.

(a) **Give** the state diagram for a DFA that recognizes L_1. (3 points)

(b) **Give** the state diagram for a DFA that recognizes L_2. (3 points)

(c) **Give** the state diagram for a DFA that recognizes L_3. (3 points)

(d) If you were to use the "cross product" construction to obtain a DFA for the language $L_2 \cap (L_3 \cup L_4)$, how many states would it have? (1 point)

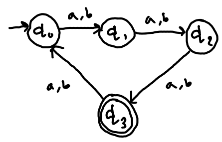
(e) **Find** all four-letter strings in $L_2 \cap (L_3 \cup L_4)$. (1 point)

(f) **Give** the state diagram for a DFA that recognizes $L_2 \cap (L_3 \cup L_4)$ using only four states. (2 points)

(g) **Find** a four-letter string in $\overline{L_3} \circ L_4$. [Recall: $\overline{L}$ denotes the complement of the language L i.e., $\overline{L} = \Sigma^* - L$] (1 point)

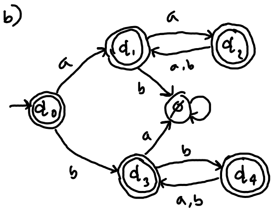
(h) Is $\overline{L_3} \circ L_4 = \overline{L_3}$? **Give** justification for your answer. (1 point)

1\. a) $4n+3$

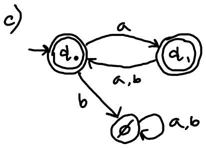


n	len
0	3
1	7
2	11

b)



c)



k	pos
0	1
1	3
2	5
3	7

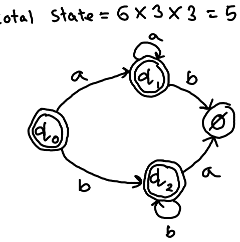
$2k+1$

d) L_4 states $\to 3$

total State $= 6 \times 3 \times 3 = 54$

e) aaaa, bbbb,

f)



g) babb

h) No. Suppose,

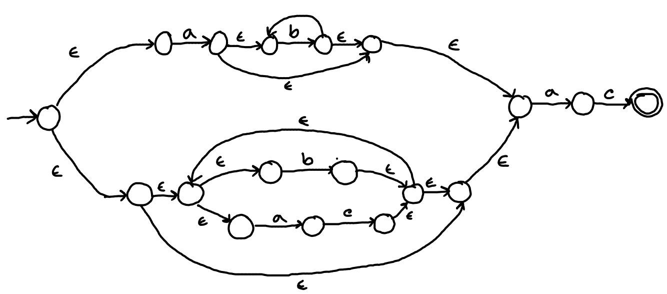
$\overline{L_3} = b$, $L_4 = ba$

$\therefore \overline{L_3} \cdot L_4 = bba$, this is not an accepted str in $\overline{L_3}$

Problem 2 (CO2): Converting Regular Expressions to NFA (10 points)

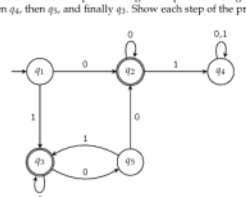
Convert the following regular expression over $\Sigma = \{a, b, c\}$ into an equivalent NFA. Note that $R_1 + R_2$ is the same as $R_1 \cup R_2$.

$$(ab^* + (b + ac)^*)bc$$

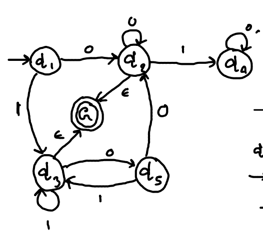


Problem 3 (CO2): Converting Finite Automata to Regular Expressions (10 points)

Convert the following finite automata into an equivalent regular expression using the state elimination method. You must eliminate q_2 first, then q_4, then q_5, and finally q_3. Show each step of the process.



$\cancel{q_2} \to \cancel{q_4} \to \cancel{q_5} \to q_3$



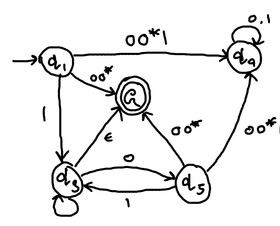
q_2

q_1 to q_f
$\to q_1 q_2 q_f$
$- 00^*$

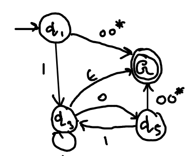
q_5 to q_f
$\to q_5 q_2 q_f$
$- 00^*$

q_1 to q_4
$\to q_1 q_2 q_4$
$- 00^*1$

q_5 to q_4
$\to q_5 q_2 q_4$
$- 00^*1$



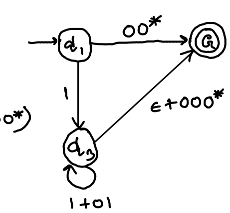
q_4 trap state



q_5

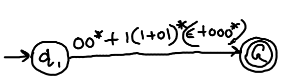
q_3 to q_f
$\to q_3 q_5 q_f - 000^*$

q_3 to q_3
$\to q_3 q_5 q_3 - 01$



q_3

q_1 to q_f
$\to q_1 q_3 q_f$
$- 1(1+01)^*(\epsilon + 000^*)$



ANS: $00^* + 1(1+01)^*(\epsilon + 000^*)$

Problem 4 (CO1): Regular Expressions (10 points)

Let $\Sigma = \{0, 1\}$. Consider the following languages over Σ.

$$L_1 = \{w : \text{length of } w \text{ is exactly } 4\}$$
$$L_2 = \{w : \text{the third last digit of } w \text{ is } 0\}$$
$$L_3 = \{w : w \text{ contains at most two } 11\}$$
$$L_4 = \overline{L_1^* \cap L_2}$$

Now solve the following problems.

(a) **Give** a regular expression for the language L_1. (1 point)

(b) **Give** a regular expression for the language L_1^*. (1 point)

(c) **Give** a regular expression for the language $\overline{L_1^*}$. [Recall: $\overline{L}$ denotes the complement of the language L i.e., $\overline{L} = \Sigma^* - L$] (2 points)

(d) **Give** a regular expression for the language L_2. (2 points)

(e) **Give** a regular expression for the language $\overline{L_3}$. (2 points)

(f) **Give** a regular expression for the language L_4. (2 points)

a) $\Sigma\Sigma\Sigma\Sigma$

b) $(\Sigma\Sigma\Sigma\Sigma)^*$

c) $(\Sigma\Sigma\Sigma\Sigma)^* \Sigma\Sigma?\Sigma?$

d) $\Sigma^* 0 \Sigma\Sigma$

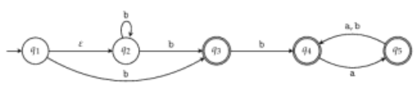
e) $\Sigma^* 11 \Sigma^* 11 \Sigma^* 11 \Sigma^*$

f) $(\Sigma\Sigma\Sigma\Sigma)^* 1 \Sigma?\Sigma?$

3nd last $\rightarrow$ 1
not a mul. of 4

Problem 5 (CO2): Subset Construction Method (5 points)

Consider the following NFA:



Now answer the following questions. [Note: You do not need to convert the given NFA into its equivalent DFA to answer the questions.]

(a) If you convert the given NFA into an equivalent DFA using the subset construction method, what is the maximum number of states that the DFA can have? (1 point)

(b) what is the maximum number of accepting states that the equivalent DFA can have? (1 point)

(c) **Write** the ε-closure of state q_1 in the given NFA. (1 point)

(d) **Write** the subset of states of the given NFA that will be the starting state in its equivalent DFA. (1 point)

(e) What is $\delta(\{q_1, q_3\}, b)$ in the given NFA? [Recall: $\delta(\{q\}, a)$ is the set of states in which the NFA transitions when it is in state q and receives input a.] (1 point)

a) $2^5 = 32$

b) reject states $\rightarrow 4$

$\therefore$ accept states $\rightarrow 32 - 4 = 28$

c) $q_1 \xrightarrow{\epsilon} \{q_1, q_2\}$

d) $q_1 q_2$

e) $q_1 q_3 \xrightarrow{b} q_3 q_4$

Set B

Problem 1 (CO1): DFA and Regular Languages (15 points)

Let $\Sigma = \{a, b\}$. Consider the following languages over Σ.

$$L_1 = \{w : \text{length of } w \text{ is two more than multiple of four}\}$$
$$L_2 = \{w : \text{every even position letter in } w \text{ is different from the first letter of } w\}$$
$$L_3 = \{w : \text{every } 2k+1 \text{ position in } w \text{ is b, where } k \geq 0\}$$
$$L_4 = \{w : \text{every } 2k+1 \text{ position in } w \text{ is a, where } k \geq 0\}$$

Now solve the following problems.

(a) **Give** the state diagram for a DFA that recognizes L_1. (3 points)

(b) **Give** the state diagram for a DFA that recognizes L_2. (3 points)

(c) **Give** the state diagram for a DFA that recognizes L_3. (3 points)

(d) If you were to use the "cross product" construction to obtain a DFA for the language $L_2 \cap (L_3 \cup L_4)$, how many states would it have? (1 point)

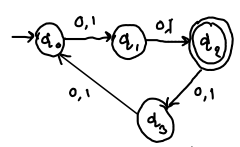
(e) **Find** all four-letter strings in $L_2 \cap (L_3 \cup L_4)$. (1 point)

(f) **Give** the state diagram for a DFA that recognizes $L_2 \cap (L_3 \cup L_4)$ using only four states. (2 points)

(g) **Find** a four-letter string in $\overline{L_3} \circ L_4$. [Recall: $\overline{L}$ denotes the complement of the language L i.e., $\overline{L} = \Sigma^* - L$] (1 point)

(h) Is $\overline{L_3} \circ L_4 = \overline{L_3}$? **Give** justification for your answer. (1 point)

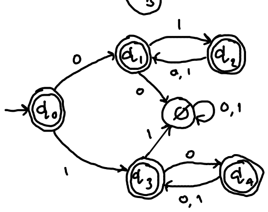
a)



$4n+2$

n	len
0	2
1	6
2	8

b)

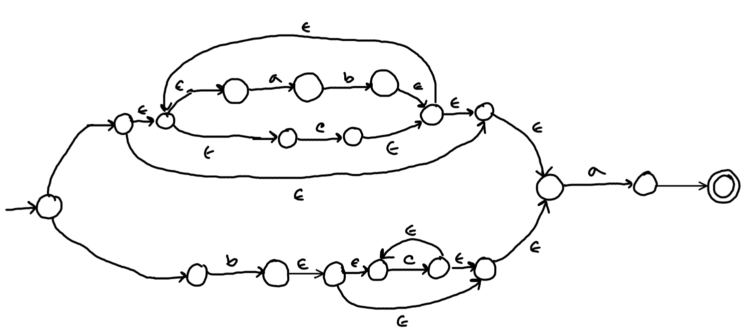


c) ✓ d) ✓ e) ✓ f) ✓ g) ✓ h) ✓

Problem 2 (CO2): Converting Regular Expressions to NFA (10 points)

Convert the following regular expression over $\Sigma = \{a, b, c\}$ into an equivalent NFA. Note that $R_1 + R_2$ is the same as $R_1 \cup R_2$.

$$((ab + c)^* + bc^*)ac$$



3. ✓ 4. ✓ 5. ✓