

There are a total of five problems. You have to solve the first four. Problem 5 is optional.

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

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

$$\begin{aligned}L_1 &= \{w \text{ starts with } 01\} \\L_2 &= \{w \text{ doesn't contain } 00\} \\L_3 &= \{w \text{ doesn't contain } 11\} \\L_4 &= \{w = 01\}\end{aligned}$$

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) If you were to use the “cross product” construction shown in class to obtain a DFA for the language $L_2 \cap L_3$, how many states would it have? (1 point)
- (d) **Find** all four-letter strings in $L_2 \cap L_3$. (1 point)
- (e) **Give** the state diagram for a DFA that recognizes $L_2 \cap L_3$ using only four states. (2 points)
- (f) **Find** one six-letter string in L_4^* . (1 point)
- (g) **Give** the state diagram for a DFA that recognizes L_4^* . (2 points)
- (h) Is L_4^* and $L_1 \cap L_2 \cap L_3$ same? **Give** justification for your answer. (2 points)

Problem 2 (CO1): Regular Expressions (15 points)

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

$$\begin{aligned}L_1 &= \{w \text{ contains exactly two } 1\} \\L_2 &= \{w \text{ doesn't start with } 0\} \\L_3 &= \{\text{every third position in } w \text{ is } 1\} \\L_4 &= \{\text{every } 1 \text{ in } w \text{ is followed by at least two } 0\} \\L_5 &= L_3 \cap L_4\end{aligned}$$

Now solve the following problems.

- (a) **Give** a regular expression for the language L_1 . (3 points)
- (b) **Give** a regular expression for the language L_2 . (3 points)
- (c) **Give** a regular expression for the language L_3 . (3 points)
- (d) **Write** a five-letter string that belongs to L_5 . (1 point)
- (e) **Give** a regular expression for the language L_5 . (2 points)
- (f) **Give** a regular expression for the language \bar{L}_4 . Here \bar{L} denotes the complement of the language L i.e., $\bar{L} = \Sigma^* - L$. (3 points)

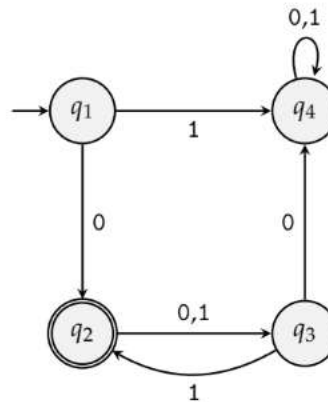
Problem 3 (CO2): Converting Regular Expressions to NFAs (10 points)

Convert the following regular expression over $\Sigma = \{0, 1\}$ into an equivalent NFA. Note that $R_1 + R_2$ is the same as $R_1 \cup R_2$.

$$(0(0 + 01)^*10) + 01^*0$$

Problem 4 (CO2): Converting Finite Automata to Regular Expressions (10 points)

Convert the following DFA into an equivalent regular expression using the state elimination method. First eliminate q_2 , then q_3 and finally q_4 . You must show work.



Problem 5 (Bonus): Ternary Number System (5 points)

Disclaimer: This is a bonus problem. Attempt it only after you are done with everything else. Even if you do not attempt it, you can get a perfect score. So, do not worry if you find it too hard!

Let $\Sigma = \{0, 1, 2\}$.

$$L = \{w, \text{ when interpreted in three base number system, is divisible by nine} \}$$

Give a state diagram for a DFA that recognizes L .