Registration No.: | | | | | | | | | | |

Total Number of Pages: 03 **Course: B.Tech/IDD**
Sub_Code: CSPC3001

5th Semester Regular Examination: 2025-26
SUBJECT: Theory of Computation
BRANCH(S): CE, CSE, CSEAI, CSEAIML, CSIT, CST, IT, CSE
Time: 3 Hours
Max Marks: 100
Q.Code: U277
Answer Q1 (Part-I) which is compulsory, any eight from Part-II and any two from Part-III.
The figures in the right-hand margin indicate marks.

Part-I

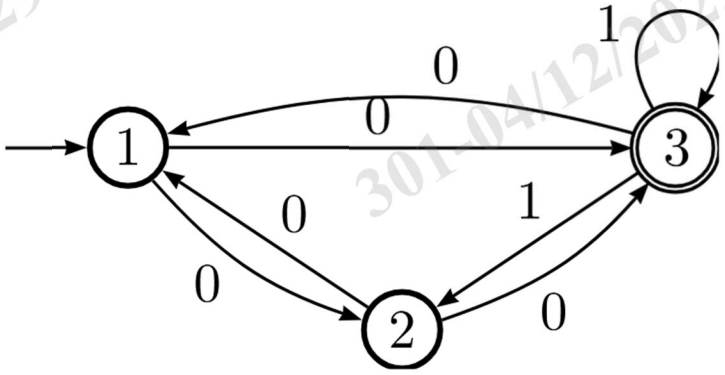
Q1 **Answer the following questions:** **(2 x 10)**

a) Define alphabet and language with examples.
b) Construct a transition diagram for a DFA that accepts all strings ending with 01.
c) Write any two identities of regular expressions.
d) Give the right-linear grammar for the language {strings ending with a}.
e) Define leftmost and rightmost derivations with suitable example.
f) Draw a simple transition diagram of a PDA that accepts $a^n b^n$ by empty stack.
g) Define recursively enumerable language. How it is different from recursive languages.
h) Define Turing machine. Draw a schematic diagram of a Turing Machine tape and head movement.
i) Define NP-complete problem.
j) Write the four levels of the Chomsky hierarchy of Languages.

Part-II

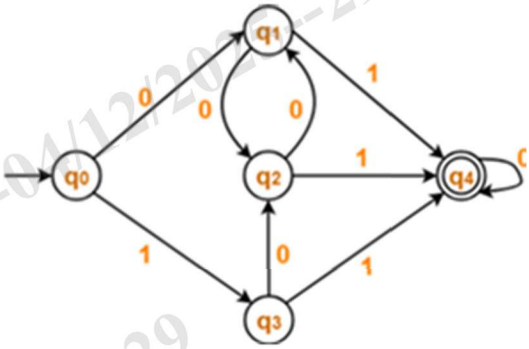
Q2 **Only Focused-Short Answer Type Questions- (Answer Any Eight out of Twelve)** **(6 x 8)**

a) Convert the following NFA to an equivalent DFA and draw the transition diagram:



b) Explain ε-closure with an example. Construct an NFA with ε transitions for the regular expression a(b+c)*.

c) Minimize the following DFA. Show all partition steps.



d) Convert the regular expression (ab + b)* to an NFA. Convert the obtained NFA to a DFA and draw the transition diagram.

e) Define pumping Leema for regular Languages. Use the Pumping Lemma to show that L = { $a^n b^n$ | n ≥ 0 } is *not* regular.

f) Convert the following left-linear grammar to right-linear grammar. Convert the right-linear grammar into an equivalent DFA:
S → Sa | Ab | a
A → Ab | ε

g) Convert the grammar
S → aSb | ab
into Chomsky Normal Form and Greibach Normal Form.

h) Construct a PDA with transition diagram that accepts $L = \{ ww^R \mid w \in \{a, b\} \}^*$ by empty stack.

i) Explain ambiguity in grammars. Show by derivations that the grammar
E → E+E | E*E | id
is ambiguous.

j) Design a Turing Machine to increment a binary number by 1. Draw the full transition diagram.

k) Explain mapping reducibility. Show that HALT_TM is reducible to ATM.

l) Explain with examples the complexity classes P, NP, and NP-Complete. Give one problem each in P, NP, and NP-Complete.

Part-III
Only Long Answer Type Questions (Answer Any Two out of Four)

Q3 Write a regular expression for the language: **(16)**

L = {w | w ∈ {0, 1}*, w = 〈n〉 , n ∈ N, n ≡ 4 mod 5}

In other words, L is the language of binary strings that encode natural numbers that have remainder 4 when divided by 5. For example, 1001 = 〈9〉 ∈ L, but 110 = 〈6〉 ∉ L Ignore leadning zeros (so the strings 00010 and 0010 are both encodings of the same natural number, 〈2〉.

Q4 **a)** Design a DPDA (deterministic PDA) for the language **(8)**
$L = \{ a^n b^n c^m \mid n \geq 1, m \geq 0 \}$
showing its full formal transition structure and transition diagram.

b) Convert the following grammar to Greibach Normal Form (GNF): **(8)**
$S \rightarrow AB \mid a$
$A \rightarrow aA \mid b$
$B \rightarrow bB \mid a$

Q5 **a)** Construct a Turing Machine to decide the language **(8)**
$L = \{ 0^n 1^n 0^n \mid n \geq 1 \}$
Draw the complete transition diagram and explain every tape movement.

b) Prove that Post Correspondence Problem (PCP) is undecidable using reduction from ATM. **(8)**

Q6 **a)** Describe a Turing Machine that recognizes the language **(8)**
$L = \{ w0w \mid w \in \{0,1\}^* \}$
Draw the state diagram.

b) Explain the Cook–Levin theorem. Using diagrams, reduce 3-SAT to CLIQUE showing each conversion step. **(8)**