QP Code:CS2T05-1 **(Pages:3)**

Reg No : ..... Name : .....

MAR ATHANASIUS COLLEGE OF ENGINEERING, KOTHAMANGALAM

(Govt. Aided Autonomous Institution)

Fourth Semester B.Tech. Degree Examination, May, 2026

Course Code: B24CS2T05

Course Name: FORMAL LANGAUGES AND AUTOMATA THEORY

Max. Marks: 100 Duration:3 Hours

PART - A: 30 Marks (10 Qns x 3 Marks)

Answer all questions

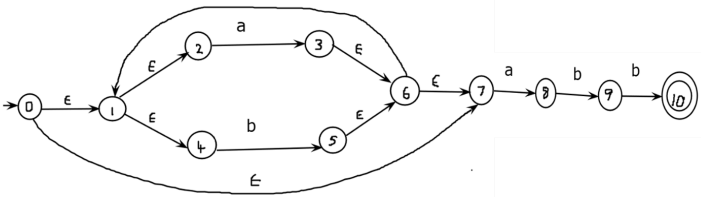
1. Define a Deterministic Finite Automata formally.
2. Write a regular expression for the language over $\Sigma=\{0,1\}$ containing all strings with at least one 1.
3. Briefly explain the method of converting a DFA into a regular expression using Arden's theorem.
4. Construct a DFA for strings over {0,1} divisible by 3 (binary number interpretation) and minimize the DFA.
5. State and explain any three closure properties of Context Free Languages
6. Remove useless symbols and epsilon productions from $S \rightarrow AB$, $A \rightarrow aA \mid \varepsilon$, $B \rightarrow C$, $C \rightarrow \varepsilon$.
7. Describe the various ways of acceptance of a string in PDA.
8. Show the trivial conversion from any given DPDA to an equivalent NPDA
9. When do you say a problem is decidable. Give example.
10. Give an overview of recursively enumerable language.

PART - B: 70 Marks (5 Qns x 14 Marks)

Answer any One full question from each module.

Module 1

11. (a)Convert the below epsilon-NFA to DFA (10 marks)



(b)Identify the language accepted by the finite automata. (4 marks)

OR

12. (a) Explain how finite automata are used in text search and keyword recognition (e.g., searching a keyword in a document). (5 marks)

(b) Construct a Mealy machine to recognize the keyword “aba” in a stream of characters and output 1 when detected. Draw the state diagram and write the transition table. (9 marks)

Module 2

13. (a) Convert the following finite automaton into an equivalent Left Linear Grammar: (7 marks)

$\delta(q_0, a) \rightarrow q_1$

$\delta(q_1, b) \rightarrow q_2$

$\delta(q_2, a) \rightarrow q_2$

(q_0 is start state and q_2 is final state)

(b) Convert the given right linear grammar to left linear grammar (7 marks)

A->aB

B->aB|bB|e

OR

14. a) State and explain the Pumping Lemma for Regular Languages. (7 marks)

b) Using the Pumping Lemma, prove that the language

$L = \{ a^n b^n \mid n \geq 0 \}$ is not regular. (7 marks)

Module 3

15. (a) Design a CFG for $L = \{w\, w^R \mid w \in \{a,b\}^*\}$. (7 marks)

(b)Apply leftmost and rightmost derivations for "abba". (7 marks)

OR

16. (a) Discuss techniques to remove ambiguity: precedence/associativity enforcement, grammar rewriting, auxiliary non-terminals. (5 marks)

(b) Resolve ambiguity in the grammar S → S + S | S * S | id to make * higher precedence and both left-associative. (5 marks)

(c) Provide parse trees for id + id * id before and after removal. (4 marks)

Module 4

17. (a) Construct CSG for L = $\{a^n b^m c^n \mid n,m \geq 1\}$. (7 marks)

(b) Design corresponding LBA and simulate on "aabbcc". (7 marks)

OR

18. (a) Construct aCFG to generate L(M) where M: ({p, q}, {0, 1}, {X, Z_0},δ,q,Z_0, ø) where δ is defined as follows:

δ (q, 0,Z_0)= (q,XZ_0)

δ (q,0, X) = (q, XX)

δ (q, l, X) = (p, Ɛ)

δ (p, l, X) = (p, Ɛ)

δ(p, Ɛ, X) = (p, Ɛ)

δ (p, Ɛ, Z0) = (p, Ɛ) (8 marks)

(b) Simulate any string using the CFG constructed. (6 marks)

Module 5

19. (a)Define a Turing Machine formally. (4 marks)

(b)Design a Turing Machine that performs binary addition of two numbers separated by a symbol #, Explain the procedure with an example . (10 marks)

OR

20. What is a Universal Turing Machine? Explain its construction and significance. How does it simulate any arbitrary Turing Machine?

****