

December 2024
B.Tech (CE) - 5th Sem
Theory of Computation (PCC-CS-502)

Duration: 3 Hours

Max. Marks: 75

Instructions:

- It is compulsory to answer all the questions (1.5 marks each) of Part-A in short.
- Answer any four questions from Part-B in detail.
- Different sub-parts of a question are to be attempted adjacent to each other.

PART A (1.5 marks each)

- Q1** (a) Enumerate the strings of exactly length four from the regular expression $(1^*0 + 01^*)^*$ (1.5)
- (b) Define mathematically DFA and NDFA. (1.5)
- (c) Define mathematically DPDA and NDPA. (1.5)
- (d) Enumerate the strings of exactly length four for the given grammar $S \rightarrow aSb/bSb/a/b/\varepsilon$ (1.5)
- (e) Define mathematically Deterministic Turing Machine and Non-Deterministic TM. (1.5)
- (f) Differentiate between Compiler and Interpreter. (1.5)
- (g) Name all the phases of Compilers. (1.5)
- (h) What are the functions of lexical analyzer? (1.5)
- (i) What do you mean by Parser? (1.5)
- (j) Differentiate between syntax and semantic. (1.5)

PART B

- Q2** (a) Design the DFA for the acceptance of occurrences of even numbers of 1's and odd number's of 0's in any binary string. (10)
- (b) Design the DFA for the regular expression $(1^*0 + 01^*)^*$ (5)
- Q3** (a) State the Arden's theorem. (5)
- (b) Prove that $\{a^p \mid p \text{ is prime number and } a \in \Sigma\}$ is not regular. (10)

Q4 (a) Design a PDA for the language $M = \{0^n, 1^m, 2^p \mid n, m > 0, \text{ either } n = m \text{ or } n = p\}$ (10)

(b) Briefly discuss the functions of all the phases of a compiler. (5)

Q5 (a) Design a TM for the language $M = \{w\#w \mid w \in (a + b)^*\}$ (5)

(b) Explain the different types of Chomsky classification of formal languages. (10)

Q6 For the grammar:

(1) $E \rightarrow E + T$

(2) $E \rightarrow T$

(3) $T \rightarrow T * F$

(4) $T \rightarrow F$

(5) $F \rightarrow (E)$

(6) $F \rightarrow \text{id}$

Demonstrate the working of LR parser for the string $\text{id} * \text{id} + \text{id}$ (15)

STATE (E)	ACTION						GOTO		
	id	+	*	(	)	\$	E	T	F
0	S5			S4			1	2	3
1		S6				Accept			
2		R2	S7		R2	R2			
3		R4	R4		R4	R4			
4	S5			S4			8	2	3
5		R6	R6		R6	R6			
6	S5			S4				9	3
7	S5			S4					10
8		S6			S11				
9		R1	S7		R1	R1			
10		R3	R3		R3	R3			
11		R5	R5		R5	R5			

Q7 Write short notes on: (15)

1. Symbol Table
2. Intermediate Code Generation
3. Machine Code Generation and Optimization



pyqfort.com

Praxian