

Code: 23IT3503

III B.Tech I Semester - Supplementary Examinations - APRIL 2026**AUTOMATA THEORY & COMPILER DESIGN
(INFORMATION TECHNOLOGY)**

Duration: 3 hours

Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.

2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.

3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.

4. All parts of Question paper must be answered in one place.

PART – A

1.a)	Draw NFA for the regular expression $c^*(a+b)$.
1.b)	Write the applications of Regular Expressions.
1.c)	Explain the Turing Machine model with its components and working principle.
1.d)	Write closure properties for Context Free Languages.
1.e)	Outline the significance of the Symbol table in the compilation.
1.f)	Describe a Parse tree with a suitable example and explain its role in syntax analysis.
1.g)	Draw the functional diagram of Shift Reduce parsing.
1.h)	What are the applications of Syntax Directed Translation?
1.i)	Describe Abstract Syntax Tree.
1.j)	Explain how a Directed Acyclic Graph (DAG) is constructed and its significance in compiler optimization.

PART – B

			Max. Marks
UNIT-I			
2	a)	Construct NFA for the regular expression $(11+0)^*(00+1)^*$.	5 M
	b)	Obtain a DFA to accept strings of a's and b's having an odd number of a's and even number of b's.	5 M
OR			
3	a)	Derive the regular expression for set of all strings beginning with 10 and ending with 101.	5 M
	b)	A language is regular if and only if some nondeterministic finite automaton recognizes It. Justify your answer with example?	5 M
UNIT-II			
4	a)	Define Parse Tree. Discuss use of Parse Tree in identifying Ambiguity in CFG.	5 M
	b)	What is PDA? Construct a PDA for $\{a^n b^n / n \geq 1\}$ by final state.	5 M
OR			
5	a)	Discuss the equivalences of PDA's and CFG's.	5 M
	b)	Define left linear grammar and Construct left linear grammar for $(0+1)^*00(0+1)^*$.	5 M
UNIT-III			
6	a)	Illustrate different phases of a Compiler with a neat diagram.	5 M
	b)	Define LEX tool. Explain in detail different sections of LEX Program.	5 M

OR			
7	a)	What is Left Recursion? Remove left recursion from the following grammar: $S \rightarrow Aa \mid b$ $A \rightarrow Ac \mid Sd \mid e$	5 M
	b)	Construct LL(1) parsing table for the following grammar: $E \rightarrow E+T \mid T$ $T \rightarrow T * F \mid F$ $F \rightarrow (E) \mid id$	5 M
UNIT-IV			
8	Construct LALR parsing table for the following grammar: $S \rightarrow CC$ $C \rightarrow cC \mid d$		10 M
OR			
9	a)	Elaborate on Parser Generator.	5 M
	b)	Define Syntax Directed Translation. Write an Syntax Directed Translation scheme for converting Infix expression into Postfix form.	5 M
UNIT-V			
10	a)	Define Three Address Codes. Construct various Three Address code representations for the expression. $x + - y * (- y + z)$.	5 M
	b)	What is Type Expression? Write type expression for two-dimensional array integers (i.e. an array of arrays) whose rows are indexed from 0 to 9 and	5 M

		whose columns are indexed from 1 to 10.	
OR			
11	a)	Elaborate Peephole optimization technique.	5 M
	b)	Give the stepwise implementation of the Code Generation algorithm and generate code for the following expression? $x=(a-b)+(a+c).$	5 M