Code No: R22A1201 **R22**

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

II B.Tech II Semester Regular Examinations, June 2024

Automata and Compiler Design

(IT, CS&IT, CSE-AIML & B.Tech-AIML)

Roll No										

Time: 3 hours **Max. Marks: 60**

Note: This question paper contains two parts A and B
Part A is compulsory which carries 10 marks and Answer all questions.
Part B Consists of 5 SECTIONS (One SECTION for each UNIT). Answer **FIVE** Questions, Choosing ONE Question from each SECTION and each Question carries 10 marks.

***

		PART-A (10 Marks)	**BCLL**	**CO(s)**	**Marks**
		(Write all answers of this part at one place)			
1	A	Name the types of finite automata.	**L1**	**CO-I**	**[1M]**
	B	Identify the applications of automata.	**L2**	**CO-I**	**[1M]**
	C	Differentiate between L attribute and S attribute.	**L3**	**CO-II**	**[1M]**
	D	Relate CLR and LALR parsers.	**L3**	**CO-II**	**[1M]**
	E	List out the need for type checking.	**L1**	**CO-III**	**[1M]**
	F	Define ambiguous grammar.	**L1**	**CO-IIII**	**[1M]**
	G	Outline the symbol table.	**L2**	**CO-IV**	**[1M]**
	H	Define lifetime of the variable.	**L1**	**CO-IV**	**[1M]**
	I	Outline the L-attribute.	**L2**	**CO-V**	**[1M]**
	J	Mention the basic block of DAG.	**L1**	**CO-V**	**[1M]**
		PART-B (50 Marks)			
		SECTION-I			
2	A	Construct a DFA, which accepts set of all string over {a,b} which accepts even number of a's and even number of b's.	**L6**	**CO-I**	**[5M]**
	B	Differentiate NFA and DFA.	**L2**	**CO-I**	**[5M]**
		OR			
3	A	Construct regular expression string ends with 00 and convert into DFA			
	B	Explain How Finite automata are useful in the lexical analysis?	**L2**	**CO-I**	**[5M]**
		SECTION-II			
4	A	Construct LL(1) Parse Table for the grammar E->E+T/T,T->T*F/F,F->(E)|id and parse the string id+id*id.	**L3**	**CO-I**	**[5M]**
	B	Identify the rule to eliminate left recursion in a grammar. Prepare and eliminate the left recursion for the grammar. $S \rightarrow Aa \mid b$ $A \rightarrow Ac \mid Sd \mid \varepsilon$	**L2,L3**	**CO-II**	**[5M]**
		OR			
5	A	Explain the phases of a compiler.	**L2**	**CO-II**	**[5M]**
	B	Perform Shift Reduce Parsing for the input string (a,(a,a)) using the grammar. S->(L)|a L->L,S|S	**L4**	**CO-II**	**[5M]**

SECTION-III

6	A	Translate the expression (a+b)*(c+d)+(a+b+c) into the following: i. Triples ii. Indirect triples.	**L6**	**CO-III**	**[5M]**
	B	Discuss about the Chomsky hierarchy of languages.	**L1**	**CO-III**	**[5M]**
		OR			
7	A	Identify the relation between the recursive and context sensitive language.	**L2**	**CO-III**	**[5M]**
	B	Outline the type-checking rule for overloaded functions with example.	**L2**	**CO-III**	**[5M]**

SECTION-IV

8	A	List the features of copy restore linkage in passing arguments.	**L2**	**CO-IV**	**[5M]**
	B	Explain storage allocation strategies in detail.	**L2**	**CO-IV**	**[5M]**
		OR			
9	A	Discuss about the followings: i) Dead code Elimination ii) Code motion.	**L2**	**CO-IV**	**[5M]**
	B	What is peephole optimization? Explain with example.	**L1**	**CO-IV**	**[5M]**

SECTION-V

10	A	Explain the various object code forms in detail.	**L3**	**CO-V**	**[5M]**
	B	Explain in brief about Induction variable elimination.	**L2**	**CO-V**	**[5M]**
		OR			
11	A	Construct a DAG and write the sequence of instructions for the expression a + a * (b-c) + (b-c) * d	**L6**	**CO-V**	**[5M]**
	B	Explain machine dependent code generation and generic code generation.	**L1**	**CO-V**	**[5M]**

***