Sri Indu
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH

Estd.2001

HANDOUT

III Year-II Semester

AUTOMATA THEORY AND COMPILER DESIGN

DEPARTMENT OF AI&DS

ACADEMIC YEAR 2025-26

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY
(An Autonomous Institution under UGC, New Delhi)
(Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi and Accredited by NBA, NAAC)
Sheriguda Village, Ibrahimpatnam Mandal, Ranga Reddy Dist. – 501 510
DEPARTMENT OF AI&DS

HANDOUT- INDEX

S. No	Contents
1	Vision, Mission, PEOs, POs, PSOs & Cos
2	Institution Academic Calendar
3	Department Academic Calendar
4	Subject wise
i)	Syllabus Copy
ii)	Lesson Plan
iii)	Question Bank
iv)	End Examination Questions (Model Paper)
v)	Mid-1 & Mid-2 Questions (Model Paper)

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

INSTITUTION VISION

To be a premier Institution in Engineering & Technology and Management with competency, values and social consciousness.

INSTITUTION MISSION

IM1 Provide high quality academic programs, training activities and research facilities.

IM2 Promote Continuous Industry-Institute Interaction for Employability, Entrepreneurship, Leadership and Research aptitude among stakeholders.

IM3 Contribute to the Economical and technological development of the region, state and nation.

DEPARTMENT VISION

To be a recognized knowledge center in the field of AI&DS with self -motivated, employable engineers to society.

DEPARTMENT MISSION

The Department has following Missions:

DM1 To offer high quality student centric education in AI&DS

DM2 To provide a conducive environment towards innovation and skills.

DM3 To involve in activities that provide social and professional solutions.

DM4 To impart training on emerging technologies namely cloud computing and IOT with involvement of stake holders.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: **Higher Studies:** Graduates with an ability to apply knowledge of Basic sciences and programming skills in their career and higher education.

PEO2: **Lifelong Learning:** Graduates with an ability to adopt new technologies for ever changing IT industry needs through Self-Study, Critical thinking and Problem solving skills.

PEO3: **Professional skills:** Graduates will be ready to work in projects related to complex problems involving multi-disciplinary projects with effective analytical skills.

PEO4: **Engineering Citizenship:** Graduates with an ability to communicate well and exhibit social, technical and ethical responsibility in process or product.

PROGRAM OUTCOMES (POs) & PROGRAM SPECIFIC OUTCOMES (PSOs)

PO	Description
PO 1	**Engineering Knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
PO 2	**Problem Analysis:** Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
PO 3	**Design/Development of Solutions:** Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
PO 4	**Conduct Investigations of Complex Problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
PO 5	**Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
PO 6	**The Engineer and The World:** Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
PO 7	**Ethics:** Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
PO 8	**Individual and Collaborative Team work:** Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
PO 9	**Communication:** Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences
PO 10	**Project Management and Finance:** Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one’s own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
PO 11	**Life-Long Learning:** Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)
Program Specific Outcomes	
PSO 1	**Software Development:** To apply the knowledge of Software Engineering, Data Communication, Web Technology and Operating Systems for building IOT and Cloud Computing applications.
PSO 2	**Industrial Skills Ability:** Design, develop and test software systems for world-wide network of computers toprovide solutions to real world problems.

PSO 3	**Project implementation:** Analyze and recommend the appropriate IT Infrastructure required for the implementationof a project.

COURSE OUTCOMES (CO's)

Academic Year: 2025-26

**YEAR/SEM : ** III -II

Course Name : AUTOMATA THEORY AND COMPILER DESIGN COURSE

**Sub Code : ** R22CSM2212

At the end of the course student will be able to:

Course Outcomes	Statements
C321.1	Able to employ finite state machines for modeling and solving computing problems
C321.2	Able to design context free grammars for formal languages.
C321.3	Able to distinguish between decidability and un decidability.
C321.4	Demonstrate the knowledge of patterns, tokens & regular expressions for lexical analysis
C321.5	Acquire skills in using lex tool and design LR parsers
C321.6	Understand intermediate code generation and run-time environment

COURSE ARTICULATION MATRIX

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PS01	PSO2	PSO 3
C321.1	2	-	1	3	3	-	-	-	-	-	-	-	-	1	-
C321.2	1	3	3	2	1	-	-	-	-	-	-	-	-	-	-
C321.3	1	3	3	1	1	-	-	-	-	-	-	-	-	-	-
C321.4	1	1	1	3	3	-	-	-	-	-	-	-	-	-	-
C321.5	3	-	2	1	3	-	-	-	-	-	-	-	-	1	-
C321.6	3	1	2	1	3	-	-	-	-	-	-	-	1	1	1
C321	1.83	2	2	1.83	2.33	-	-	-	-	-	-	-	1	1	1

\

AUTOMATA THEORY AND COMPILER DESIGN

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

B.Tech. - III Year – II Semester

L	T	P	C
3	0	0	3

(R22CSM2212) AUTOMATA THEORY AND COMPILER DESIGN

Course Objectives

- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- To understand deterministic and non-deterministic machines and the differences between decidability and undecidability.
- Introduce the major concepts of language translation and compiler design and impart the knowledge of practical skills necessary for constructing a compiler.
- Topics include phases of compiler, parsing, syntax directed translation, type checking use of symbol tables, intermediate code generation

Course Outcomes

- Able to employ finite state machines for modeling and solving computing problems.
- Able to design context free grammars for formal languages.
- Able to distinguish between decidability and undecidability.
- Demonstrate the knowledge of patterns, tokens & regular expressions for lexical analysis.
- Acquire skills in using lex tool and design LR parsers

UNIT – I :

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.

Nondeterministic Finite Automata: Formal Definition, an application, Text Search, Finite Automatawith Epsilon-Transitions.

Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA, Conversion of NFA with €-transitions to NFA without €-transitions. Conversion of NFA to DFA

UNIT - II

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages: Statement of the pumping lemma, Applications of the Pumping Lemma.

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Parse Trees, Ambiguity in Grammars and Languages.

UNIT - III

Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA and CFG's, Acceptance by final state

Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine

Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines

UNIT - IV

Introduction: The structure of a compiler,

Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical- Analyzer Generator Lex,

Syntax Analysis: Introduction, Context-Free Grammars, Writing a Grammar, Top-Down Parsing, Bottom- Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers

UNIT - V

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's.

Intermediate-Code Generation: Variants of Syntax Trees, Three-Address Code

Run-Time Environments: Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management

TEXT BOOKS:

1. Introduction to Automata Theory, Languages, and Computation, 3rd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, 2nd Edition, PHI.

REFERENCE BOOKS:

1. Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, 2nd Edition, Pearson.
2. Introduction to Formal languages Automata Theory and Computation, Kamala Krithivasan, Rama R, Pearson.
3. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
4. lex & yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly Compiler Construction, Kenneth C. Louden, Thomson. Course Technology.

	SRI INDU COLLEGE OF ENGG &TECH (Regulation :R22) Department of IT,AI&DS			
	Sub. Code & Title	R22CSM2212 Automata Theory and Compiler Design		
	Academic Year: 2025-26		Year/Sem./Section	III-II
	Faculty Name & Designation		P. Swathi ,Assistant Professor P.Chaithanya, Assistant Professor	

LESSONPLAN

(L1-Remembering L2-Underst, ding,L3-Applying,L4-Analyzing,L5-Evaluation,L6-Creating)

Unit/ Item No.	Topic (s)	Book Reference	Page (s) From	Page (s) To	Teaching methodogy	Proposed no of periods	Actual Dateof Handled	CO/RBT
	UNIT-I							
I	Preliminary concepts					12		
1.1	Introduction to finite automata	T1	1	1	Black board	1		CO1/L1
1.2	Structural representation,automata and complexity	T1	1	5	Black board	1		CO1/L2
1.3	Automata theory concepts - alphabets,strings,languages	T1	28	31	Black Board	1		CO1/L2
1.4	An application of NFA	T1	68	69	Black board	1		CO2/L4
1.5	Text search	T1	70	72	Black board	1		CO2/L4
1.6	DFA problems	T1	55	59	Black board	2		CO2/L4
1.7	NFA,NFA problems	T1	45	48	Black board	2		CO2/L4
1.8	FA with epsilon transitions	T1	72	74	Black board	2		CO2/L4
1.9	Convert NFA with Ɛ to without Ɛ	T1	75	77	Black board	1		CO2/L4
1.10	Convert NFA to DFA	T1	77	79	Black board	1		CO2/L4
	Review	Signature of the HOD/Coordinator						

Unit/ Item No.	Topic (s)	Book Reference	Page (s) From	Page (s) To	Teaching methodogy	Proposed no of periods	Actual Dateof Handled	CO/RBT
	UNIT-II							
II	Regular expressions					15		
2.1	Finite automata and regular expressions	T1	83	96	Black board	1		CO3/L3
2.2	Applications and algebroic laws of RE	T1	108	117	Black board	1		CO3/L3
2.3	Conversion of FA to RE	T1	101	106	Black board	2		CO3/L3
2.4	Pumping lemma and its applications	T1	126	127	Black board	1		CO3/L3
2.5	CFG-LMD,RMD	T1	169	175	Black board	2		CO4/L4
2.6	Parse Tree	T1	181	188	Black board	1		CO4/L5
2.7	Ambiguity	T1	205	212	Black board	1		CO4/L5

Unit/ Item No.	Topic (s)	Book Reference	Page (s) From	Page (s) To	Teaching Methodology	Proposed no of periods	Actual Dateof Handled	CO/RBT
	UNIT-III							
III	Push down automata and Turing Machine					13		
3.1	PDA definition and languages	T1	219	224	Black board	1		CO6/L6
3.2	Equivalence of PDA and CFG	T1	237	239	Black board	1		CO6/L6
3.3	Turing machine def,language	T1	327	330	Black board	2		CO6/L6
3.4	Turing machine Problem	T1	330	335	PPT	1		CO6/L6

3.5	Undecidability in RE	T1	373	374	Black board	1		CO6/L6
3.6	Undecidability in Turing machine	T1	374	375	Black board	2		CO6/L6
	Review				**Signature of HOD/Coorinator**			

SRI INDU COLLEGE OF ENGG &TECH (Regulation :R22) Department of IT, AI&DS		
Sub. Code & Title	**R22CSM2212 Automata and Compiler Design**	
Academic Year: 2025-26	**Year/Sem./Section**	**III-II**
Faculty Name & Designation	**P.Swathi ,Assistant Professor** **P.Chaithanya, Assistant Professor**	

			UNIT-IV					
IV	**Compilers**					**20**		
4.1	Compiler structure	R1	4	12	Black board	1		CO5/L1
4.2	Lexical analyzer role	R1	109	116	Black board	1		CO5/L1
4.3	Input buffering,token recognization	R1	374	375	Black board	1		CO5/L1
4.4	LEX	R1	176	177	Black board	1		CO5/L1
4.5	Syntax analysis grammer	R1	191	197	PPT	2		CO5/L2
4.6	Top down parsing	R1	217	231	Black board	1		CO5/L2
4.7	Bottom up parsing	R1	233	240	Black board	1		CO5/L1
4.8	Simple LR parsing	R1	241	257	Black board	1		CO5/L2
4.9	Powerful LR parsers	R1	250	277	Black board	2		CO5/L2
	Review	**Signature of the HOD/Coordinator**						
			UNIT-V					
V	**Syntax directed Translation**					**20**		
5.1	SDT definition and evaluation	R1	304	309	Black board	1		CO5/L6
5.2	SDT schemes	R1	310	317	Black board	1		CO5/L6
5.3	Implementing L attributed SDT	R1	337	352	Black board	1		CO5/L6
5.4	Varients of syntax trees	R1	358	362	Black board	1		CO5/L6
5.5	Three address code	R1	363	373	PPT	2		CO5/L6
5.6	Stacks	R1	430	451	Black board	1		CO5/L6
5.7	Heaps	R1	452	463	Black board	1		CO5/L6
	Review		**Signature of the HOD/Coordinator**					

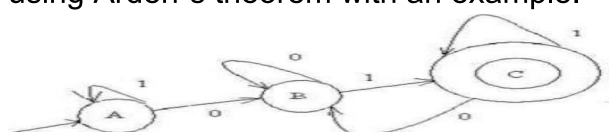
Web links

W1: https://www.javatpoint.com/theory-of-automata
W2: https://en.wikipedia.org/wiki/Automata_theory
W3: https://www.geeksforgeeks.org/introduction-of-compiler-design/
W4: https://www.javatpoint.com/compiler-tutorial
W5: https://www.tutorialspoint.com/compiler_design/index.htm

MID-I ASSIGNMENT

AUTOMATA THEORY AND COMPILER DESIGN (R22CSM2212)

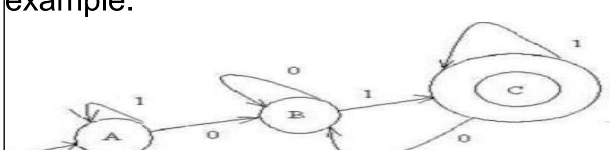
BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO’S(1-5))	BT LEVEL	COURSE OUTCOMES
	1	Construct a DFA to accept set of all strings ending with 010. Define language over an alphabet ∑ = { 0,1} and write for the above DFA .	III	CO1
	2	Differentiate Moore and Melay Machine with an example.	IV	CO1
1	3	Define DFA and write formal definition of DFA with an example	III	CO1
	4	Explain about algebraic laws of Regular Expression	II	CO2
	5	Show that the following CFG ambiguous. S→ iCtS | iCtSeS | a C →b	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO’S(6-11))	BT LEVEL	COURSE OUTCOMES
	1	Construct NFA for (0 + 1)*(00 + 11)(0 + 1)* and Convert to DFA.	III	CO1
	2	Differentiate between DFA and NFA with an example	IV	CO1
2	3	Convert given Finite Automata to Regular Expression using Arden‟s theorem with an example. 	V	CO2
	4	Define Regular Expression. List out the applications of Regular Expression	II	CO2
	5	Show that the following grammar is ambiguous with respect to the string aaabbabbba. S→aB | bA A→ aS| bAA|a B→ bS | aBB | b	V	CO3

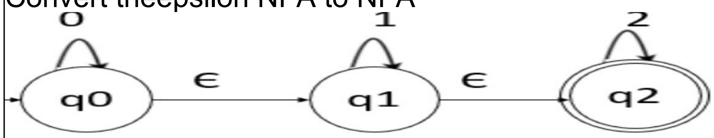
BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO’S(12-16))	BT LEVEL	COURSE OUTCOMES
	1	Construct a DFA to accept the language L={w|w is of even length and begins with 01}	III	CO1
	2	Explain NFA with suitable example	V	CO1
3	3	Discuss about the closure properties of regular languages.	II	CO2
	4	What is pumping lemma explain its applications.	III	CO2

	5	Construct Leftmost Derivation. , Rightmost Derivation,Derivation Tree for the following grammar $G = (V, T, P, S)$ with$N=\{E\}$, $S = E$, $T = \{id,+,*(,)\}$ E ☾ E+E E ☾ E*E E ☾ (E) E ☾ id Obtain id+id*id in right most derivation,left most derivation	5	CO3

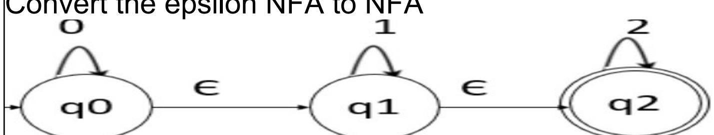
BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(17-21))	BT LEVEL	COURSE OUTCOMES
4	1	Design DFA for the following languages shown below∑ = { a,b} a .L={w/ w does not contain the substring ab} b .L={w/ w contains neither the substring ab nor ba} c.L={w/ w is any string that doesn"t contain exactly two a} d.L={w/ w is any string except a and b}	II	CO1
	2	Explain state elimination method with example	IV	CO1
	3	Explain about algebraic laws of Regular Expression.	II	CO2
	4	What is parse tree construct parse tree for the following E ☾ E+E E ☾ E*E E ☾ (E) E ☾ id Obtain id+id*id	II	CO2
	5	Show that the following CFG ambiguous. S→ iCtS | iCtSeS | a C →b	V	CO3

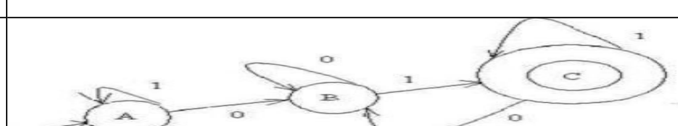
BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(22-26))	BT LEVEL	COURSE OUTCOMES
5	1	Differentiate between DFA and NFA with an example	IV	CO1
	2	Define push down automata and its language	III	CO1
	3	Convert given Finite Automata to Regular Expression using Arden"s theorem with an example. 	V	CO2
	4	Define Regular Expression. List out the applications of Regular Expression	II	CO2
	5	What is ambiguity in grammer explain with suitable example	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(27-32))	BT LEVEL	COURSE OUTCOMES
	1	Explain NFA with epsilon moves with example	IV	CO1

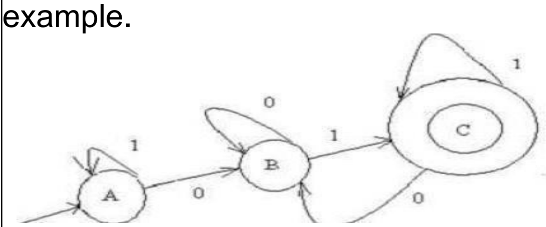
	2	Explain Arden's method with an example	III	CO1
	3	Explain about algebraic laws of Regular Expression	II	CO2
6	4	Convert theepsilon NFA to NFA 	III	CO2
	5	Show that the following grammar is ambiguous with respect to the string aaabbabbba. S→aB | bA A→ aS| bAA|a B→ bS | aBB | b	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(33-37))	BT LEVEL	COURSE OUTCOMES
	1	Differentiate between DFA and NFA with an example.	IV	CO1
	2	Equivalence of CFG and PDA	V	CO1
	3	Explain the pumping lemma and its application with an example.	II	CO2
7	4	Write short notes on 1.DFA 2.NFA 3.NFA epsilon	III	CO2
	5	What is ambiguity in grammer explain with suitable example	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(38-42))	BT LEVEL	COURSE OUTCOMES
	1	Differentiate between DFA and NFA with an example	IV	CO1
	2	Construct a DFA to accept set of all strings ending with 010. Define language over an alphabet ∑ = { 0,1} and write for the above DFA .	III	CO1
8	3	Define Regular Expression. List out the applications of Regular Expression	II	CO2
	4	Convert the epsilon NFA to NFA 	II	CO2
	5	Show that the following grammar is ambiguous with respect to the string aaabbabbba. S→aB | bA A→ aS| bAA|a B→ bS | aBB | b	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(43-47))	BT LEVEL	COURSE OUTCOMES
9	1	 Convert given Finite Automata to Regular Expression using Arden"s theorem with an	III	CO1

		example.		
	2	Differentiate between DFA and NFA with an example	IV	CO1
	3	Define Regular Expression. List out the applications of Regular Expression	II	CO2
	4	Explain about algebraic laws of Regular Expression	II	CO2
	5	What is pumping lemma explain its applications.	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(48-53))	BT LEVEL	COURSE OUTCOMES
10	1	Construct a DFA to accept the language L={w|w is of even length and begins with 01}	III	CO1
	2	Explain PDA acceptance by final state	III	CO1
	3	Explain about algebraic laws of Regular Expression.	II	CO2
	4	Convert given Finite Automata to Regular Expression using Arden"s theorem with an example. 	V	CO2
	5	Construct Leftmost Derivation. , Rightmost Derivation,Derivation Tree for the following grammar *G* = (*V*, *T*, *P*, *S*) with*N*={*E*}, *S* =*E*, *T* ={*id*,+,*(,)} E → E+E E → E*E E → (E) E → id Obtain id+id*id in right most derivation,left most derivation	V	C03

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(54-58))	BT LEVEL	COURSE OUTCOMES
11	1	Differentiate NFA and DFA with an example.	IV	CO1
	2	Convert the epsilon NFA to NFA explain with example	III	CO1
	3	Explain about algebraic laws of Regular Expression	II	CO2
	4	Explain state elimination method with example	III	CO2
	5	Show that the following grammar is ambiguous with respect to the string aaabbabbba. S→aB | bA A→ aS| bAA|a B→ bS | aBB | b	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(59-63))	BT LEVEL	COURSE OUTCOMES
12	1	Construct a DFA to accept set of all strings ending with 010. Define language over an alphabet ∑ = { 0,1} and write for the above DFA .	III	CO1
	2	.Define Regular Expression. List out the applications of Regular Expression	IV	CO1
	3	Define DFA and write formal definition of DFA with an example	III	CO2
	4	Explain about algebraic laws of Regular Expression.	II	CO2
	5	Explain PDA acceptance by final state	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(L1-L3))	BT LEVEL	COURSE OUTCOMES
13	1	Explain about algebraic laws of Regular Expression	III	CO1
	2	Explain state elimination method with example	IV	CO1
	3	Define Regular Expression. List out the applications of Regular Expression	II	CO2
	4	Define pumping lemma and write its applications	III	CO2
	5	Show that the following grammar is ambiguous with respect to the string aaabbabbba. $S \rightarrow aB \mid bA \quad A \rightarrow aS \mid bAA \mid a \quad B \rightarrow bS \mid aBB \mid b$	V	CO3

MID-II ASSIGNMENT

AUTOMATA THEORY AND COMPILER DESIGN (R22CSM2212)

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(1-5))	BT LEVEL	COURSE OUTCOMES
1	1	Explain structure of a compiler with neat diagram	III	CO1
	2	Explain the following A) compiler B)Linker and loader	IV	CO1
	3	Explain LL(1) parser with an example	III	CO1
	4	Explain LR(0) parser with an example	II	CO2
	5	Define and describe Syntax Directed Definition	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(6-11))	BT LEVEL	COURSE OUTCOMES
2	1	What is Lexical Analysis?Explain the role of Lexical Analyzer	III	CO1
	2	Explain LL(1) parser with an example	IV	CO1
	3	Define and describe context free grammar with an example	V	CO2

	4	What is parser?Explain different types of parser	II	CO2
	5	What is SDD?Describe the evaluation order for SDD	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(12-16))	BT LEVEL	COURSE OUTCOMES
	1	Explain input buffering in Lexical Analysis	III	CO1
	2	Explain LL(1) parser with an example	V	CO1
	3	What is derivation?Explain different types of derivation	II	CO2
3	4	Define SDD and write SDD schema	III	*CO3*
	5	Explain about Lexical Analyzer Generator LEX	5	C03

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(17-21))	BT LEVEL	COURSE OUTCOMES
	1	Explain LL(1) parser with an example	II	CO1
	2	Explain design of Lexical Analyzer Generator	IV	CO1
4	3	What is back tracking explain with an example	II	CO2
	4	Explain LR parsing with suitable example	II	CO2
	5	Define and describe Syntax Directed Definition	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(22-26))	BT LEVE L	COURSE OUTCOMES
	1	Explain structure of a compiler with neat diagram	IV	CO1
	2	Explain turing machine with example	III	CO1
5	3	Define and discuss about top down parser	V	CO2
	4	Explain LL(1) parser with an example	II	CO2
	5	What is SDD?Describe the evaluation order for SDD	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(27-32))	BT LEVEL	COURSE OUTCOMES
	1	Explain the following A)Turing machine B)Undecidability	IV	CO1
	2	Explain the structure of a compiler	III	CO1
	3	Explain top down parser	II	CO2
6	4	Explain SLR(1) parser with example	III	CO2
	5	Define SDD and write SDD schema	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(33-37))	BT LEVEL	COURSE OUTCOMES
7	1	What is Lexical Analysis?Explain the role of Lexical Analyze	IV	CO1
	2	Explain the following A) Compiler B) interpreter C) Assembler	V	CO1
	3	Explain CLR parser with example	II	CO2
	4	What is parse tree?Draw parse tree for given grammar and input string is a+b*c E-->E+T E-->E*T E-->T T-->a/b/c	III	CO2
	5	Define SDD and write SDD schema	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(38-42))	BT LEVEL	COURSE OUTCOMES
8	1	Explain input buffering in Lexical Analysis	IV	CO1
	2	Define and describe undecidability	III	CO1
	3	Explain LR parser with an example	II	CO2
	4	What is ambiguity?The following grammar is ambiguous or not input string a+b*c E-->E+T E-->E*T E-->T T-->a/b/c	II	CO2
	5	Define and describe Syntax Directed Definition	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(43-47))	BT LEVEL	COURSE OUTCOMES
9	1	Define and describe Syntax Directed Definition	III	CO1
	2	Explain design of Lexical Analyzer Generator	IV	CO1
	3	Explain LALR(1) parser with an exaple	II	CO2
	4	Explain turing machine with example	II	CO2
	5	Explain the following A) compiler B)Linker and loader	V	CO3

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(48-53))	BT LEVEL	COURSE OUTCOMES

10	1	Explain about Lexical Analyzer Generator LEX	III	CO1
	2	What is derivation?Explain different types of derivation	III	CO1
	3	Explain SLR parser with example	II	CO2
	4	Define and describe Syntax Directed Definition	V	C03
	5	Explain turing machine with example	V	CO2

BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(54-58))	BT LEVEL	COURSE OUTCOMES
11	1	Explain structure of a compiler with neat diagram	IV	CO1
	2	Explain undecidability with example	III	CO1
	3	Explain about run time allocation	II	CO2
	4	Explain LL(1) parser with an example	III	CO2
	5	What is SDD?Describe the evaluation order for SDD	V	CO3
BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(59-64))	BT LEVEL	COURSE OUTCOMES
12	1	Explain input buffering in Lexical Analysis	III	CO1
	2	What is stack and explain stack allocation	IV	CO1
	3	Explain SLR(1) parser with example	III	CO2
	4	What is parse tree?Draw parse tree for given grammar and input string is a+b*c F-->E+T E-->E*T F-->T T-->a/b/c	II	CO2
	5	Define SDD and write SDD schema	V	CO3

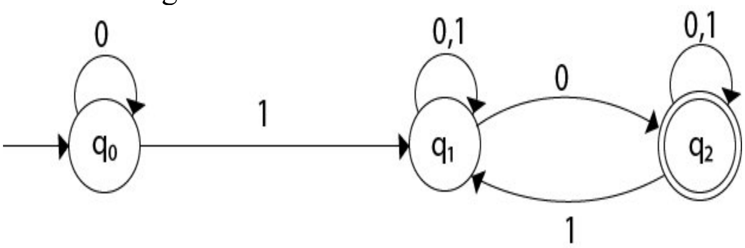
BATCH NO	SET NO	ASSIGNMENT QUESTIONS(ROLL NO'S(L1-L3))	BT LEVEL	COURSE OUTCOMES
13	1	Explain structure of a compiler with neat diagram	III	CO1
	2	Explain the following A) compiler B)Linker and loader	IV	CO1
	3	Explain turing machine with example	II	CO2
	4	Explain LL(1) parser with an example	III	CO2
	5	What is SDD?Describe the evaluation order for SDD	V	CO3

	SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY **Question Bank** **(Regulation: R22)** **Department of IT,AI&DS**			**Prepared on 05/01/2026**
	Subject Code & Title	**R22CSM2212 – AUTOMATA THEORY AND COMPILER DESIGN**		
	Academic Year: 2025-2026	**Year/Sem/Section**	**III/II**	
	Faculty Name & Designation	**P.SWATHI, Assistant Professor** **P.Chaithanya,Assistant Professor**		

UNIT – I			
OBJECTIVE QUESTIONS			
SL.NO	**MULTIPLE CHOOSE QUESTIONS**	**BT Level**	**Course Outcome**
1C – 1	An Automata is (A) plural form of Automaton (B) abstract computing device (C) abstract model (D) all of the above	**I**	**CO1**

1C – 2	Which of the following is powerful (A) Turing machine (B) NFA (C) DFA (D) B and C	**I**	**CO1**
1C – 3	Finite Automata are a useful model for (A) hardware (B) software (C) lexical analyzer (D) all of the above	**I**	**CO1**
1C – 4	Finite Automata defines (A) infinite states (B) finite states (C) initial state (D) final state	**I**	**CO1**
1C – 5	Which of the following represents Alphabet? (A) $\sum$ (B) Q (C) A (D) ε	**I**	**CO1**
1C – 6	Which of the following is not true? (A) NFA is not capable as DFA (B) NFA and DFA are equal capable (C) we can convert NFA to DFA (D) finite Automata accepts Regular languages	**I**	**CO1**
1C – 7	NFA Represented by (A) 6 Tuples (B) 3 Tuples (C) 5 Tuples (D) 7 Tuples	**I**	**CO1**
1C – 8	The value of $\lvert 1101 \rvert$ is (A) 2 (B) 3 (C) 4 (D) 5	**II**	**CO1**
1C – 9	Which of the statements is incorrect? (A) Symbols on alphabet are infinite (B) Strings on alphabet are infinite (C) Terminal are denoted by small letters (D) Variables are also known as Non-terminals	**I**	**CO1**
1C – 10	Which of the following is used to represent Automata by graph? (A) table (B) Transition diagram (C) pie-chart (D) none	**I**	**CO1**
1C – 11	In Automata theory a problem is the question of deciding (A) complexity (B) membership in a language (C) string type (D) A and C	**II**	**CO1**
1C – 12	Which of the following is an application of DFA? (A) Search engine (B) paint (C) Text Search (D) A and C	**I**	**CO1**

1C – 13	A string is (A) set of symbols (B) infinite symbols (C) only one symbol (D) finite length sequence of symbols	**I**	**CO1**
1C – 14	Number of initial states in NDA (A) 4 (B) 2 (C) 1 (D) 0	**I**	**CO1**
1C – 15	Number of Final states in Automata (A)1 (B) 2 (C) 1 or more than 1 (D) infinite	**II**	**CO1**
FILL IN THE BLANKS			
1F – 1	Automata theory is the study of________________	**I**	**CO1**
1F – 2	Regular languages are accepted by______________ Automata	**II**	**CO1**
1F – 3	Expansion of NFA______________________	**I**	**CO1**
1F – 4	___________________language is a language over any alphabet	**III**	**CO1**
1F – 5	_______________ indicates empty string	**I**	**CO1**
1F - 6	Reason to study automata ________________________	**I**	**CO1**
1F – 7	The set of strings of length 2 over alphabet a,b_______	**I**	**CO1**
1F - 8	A string also known as______________	**I**	**CO1**
1F - 9	_________________________is a Tabular Representation of a function	**I**	**CO1**
1F – 10	___________has the power to be in several states at once	**II**	**CO1**
1F – 11	contributes nothing to the string along the path	**I**	**CO1**
1F – 12	--------------- number of states required to accept string ends with 01	**I**	**CO1**
1F – 13	Language of finite automata is type _____________	**I**	**CO1**
1F – 14	Finite Automata Require number of stacks_______	**I**	**CO1**
1F – 15	Regular Expression for all strings start with ab and ends with bba is______________	**V**	**CO1**
MATCH THE FOLLOWING			
1M – 1	1. Terminal a. B 2. Non-terminal b. a 3. Empty string c. q0 4. Empty language d. € 5. Initial state e. ɷ	**III**	**CO1**
1M – 2	1. Finite Automata a. No Epsilon transition 2. Automata b. Language Acceptor 3. NFA c. word 4. DFA d. Epsilon transition 5. String e. Abstract device	**III**	**CO1**
1M – 3	1. F a. finite set of input symbols 2. ∑ b. Subset of Q 3. δ c. initial state 4. qo d. transition function 5. Q e. set of final states	**I**	**CO1**
1M – 4	1. Alphabet a. set of strings 2. Language b. set of symbols 3. String c. list of symbols 4. Empty String d. xy	**I**	**CO1**

	5. Concatenation e. zero		
1M - 5	1. Type 0 a. Regular grammar 2. Type 3 b. More poweful than Automata 3. DFA c. Unrestricted grammar 4. NFA d. accepts Epsilon 5. TM e. does not accept Epsilon	**II**	**CO1**
	10 – MARKS QUESTIONS		
1D – 1	Define Finite Automata , and list out the uses of Finite Automata	**I**	**CO1**
1D – 2	Why do we study Automata Theory? Discuss about Automata Theory	**IV**	**CO1**
1D – 3	Elaborate notations that play an important role in the study of automata and their applications	**VI**	**CO1**
1D – 4	Discuss the following a) Grammars b) Regular Expressions	**II**	**CO1**
1D – 5	Analyze about the Automata and Complexity	**III**	**CO1**
1D – 6	Analyze the following a) Alphabet b) Strings c) Languages d) Grammar	**IV**	**CO1**
1D – 7	Explain NFA with an example	**II**	**CO1**
1D – 8	Explain about DFA	**V**	**CO1**
1D – 9	Compare DFA and NFA	**II**	**CO1**
1D – 10	What is Transition Diagram? Give an example	**I**	**CO1**
1D – 11	Discuss Finite Automata with Epsilon transitions	**III**	**CO1**
1D – 12	Write in detail about Applications of NFA	**I**	**CO1**
1D – 13	Discuss about how DFA Process Strings, in detail	**IV**	**CO1**
1D – 14	Evaluate the steps in converting NFA to DFA with Example	**V**	**CO1**
1D – 15	Explain about the central concepts of Finite Automata	**II**	**CO1**
1D – 16	Discuss about the Conversion of NFA with €-transitions to NFA without €-transitions	**IV**	**CO1**
1D – 17	Convert the given NFA to DFA 	**III**	**CO1**

	UNIT – II		
	OBJECTIVE QUESTIONS		
SL.NO	**MULTIPLE CHOOSE QUESTIONS**	**BT Level**	**Course Outcome**
2C – 1	Which of the following operations is NOT supported in regular expressions? a) Concatenation b) Exponentiation c) Union	**I**	**CO2**

	d) Kleene star		
2C – 2	Which of the following regular expressions represents the language of all strings over {0,1} that do not contain '11' as a substring? a) (0+1)* b) (0+1)11(0+1) c) (0+1)(0+11) d) (01)*	**I**	**CO2**
2C – 3	Which of the following regular expressions represents an empty language? a) ∅ b) ε c) (0+1)* d) (0+1)+	**I**	**CO2**
2C – 4	Which operation in regular expressions is used to represent alternatives? a) Concatenation b) Kleene star c) Union d) Intersection	**II**	**CO2**
2C – 5	What does the regular expression "a|b" represent? a) Either 'a' or 'b' b) String containing 'a' followed by 'b' c) String containing 'a' and 'b' d) String containing 'a' or 'b'	**I**	**CO2**
2C – 6	What does the regular expression "a*" represent? a) Strings containing only 'a' b) Strings containing at least one 'a' c) Strings containing zero or more 'a's d) Strings containing exactly one 'a'	**II**	**CO2**
2C – 7	What does a context-free grammar consist of? a) Terminals, production rules, start symbol b) Terminals, non-terminals, start symbol c) Terminals, production rules, regular expressions d) Terminals, non-terminals, regular expressions	**I**	**CO2**
2C – 8	Which notation is commonly used to represent non-terminals in a context-free grammar? a) Uppercase letters b) Lowercase letters c) Alphanumeric characters d) Symbols enclosed in square brackets	**I**	**CO2**
2C – 9	What does the start symbol represent in a context-free grammar? a) The symbol used to begin each production rule b) The first non-terminal symbol in the grammar c) The initial state in parsing d) The symbol used to terminate each production rule	**I**	**CO2**
2C – 10	Which of the following describes the language generated by a context-free grammar? a) Regular languages b) Context-sensitive languages c) Context-free languages d) Turing machines	**III**	**CO2**
2C – 11	What is the purpose of using recursion in context-free grammars? a) To generate infinitely long strings b) To represent looping constructs c) To allow non-terminals to expand into themselves d) To ensure termination of parsing	**I**	**CO2**

2C – 12	Which of the following is NOT a valid operation in context-free grammars? a) Union b) Concatenation c) Kleene closure d) Intersection	**I**	**CO2**
2C – 13	In a context-free grammar, what does ε represent? a) Empty string b) End of input c) Start of input d) Epsilon transition	**II**	**CO2**
2C – 14	Which of the following is used to denote the production of multiple alternatives in a context-free grammar? a) Curly braces {} b) Square brackets [] c) Angle brackets <> d) Pipe symbol |	**II**	**CO2**
2C – 15	Context-free grammars are commonly associated with which type of automaton? a) Deterministic Finite Automaton (DFA) b) Non-deterministic Finite Automaton (NFA) c) Pushdown Automaton (PDA) d) Turing Machine	**II**	**CO2**
	FILL IN THE BLANKS		
2F – 1	The regular expression _______ represents the language of all strings over {0,1} that start with '1'.	**I**	**CO2**
2F – 2	The regular expression _______ denotes zero or more occurrences of the preceding character 'a'.	**II**	**CO2**
2F – 3	The regular expression _______ matches strings containing any number of 'a's followed by any number of 'b's.	**I**	**CO2**
2F – 4	The regular expression _______ denotes one or more occurrences of the preceding character 'c'.	**I**	**CO2**
2F – 5	A grammar is ________________ used to generate strings in a language. It consists of a finite set of symbols	**I**	**CO2**
2F – 6	A language is _________________ over some alphabet. It can be finite or infinite.	**II**	**CO2**
2F – 7	Regular languages can be recognized by _____________.	**I**	**CO2**
2F – 8	The _____________ lemma is a fundamental result in the theory of regular languages.	**I**	**CO2**
2F – 9	Every regular language can be expressed by a _____________ expression.	**I**	**CO2**
2F – 10	The union, concatenation, and Kleene star operations are closed under _____________.	**I**	**CO2**
2F – 11	Terminals are symbols that appear in the _____________ strings of the language.	**I**	**CO2**
2F – 12	Non-terminals are symbols that can be replaced by groups of _____________ according to the rules of the grammar.	**III**	**CO2**
2F – 13	Productions (or rules) specify how _____________ can be replaced by combinations of terminals and other non-terminals.	**I**	**CO2**
2F – 14	Context-free grammars are used to describe _____________ languages.	**III**	**CO2**
2F – 15	A context-free grammar consists of a set of ________________	**I**	**CO2**
	MATCH THE FOLLOWING		

2M – 1	1. (0+1)0 2. (00+11) 3. (0+1)(0+1) 4. 01 a. Matches strings containing an even number of '0's and an even number of '1's. b. Matches strings containing '0's followed by '1's. c. Matches strings containing alternating '0's and '1's. d. Matches strings ending with '0'.	**II**	**CO2**
2M – 2	1. V a. Start Symbol 2. T b. Set of Variables 3. P c. Terminal 4. S d. Production Rules	**I**	**CO2**
2M – 3	1.Recursive enumerable languages a. Finite Automata 2. Context Sensitive Grammar b. PDA 3.Context free Grammar c. LBA 4.Regular Grammar d. Turing Machine	**III**	**CO2**
2M – 4	1. Stack DS a. Gives an infinite set of all possible strings of any length 2. Queue DS b. PDA 3. Pumping Lemma c. Used to prove that given language is not regular 4. Kleen Colsure d. LBA	**IV**	**CO2**
2M – 5	1. Identity for union a. Epsilon 2. Identity for concatenation b. Theta 3. Idempotent law c. L+M=M+L 4. Commutative law d. L Union L	**III**	**CO2**
	10 – MARKS QUESTIONS		
2D – 1	What is Regular Expression? Discuss Finite automata with Regular Expression	**I**	**CO2**
2D – 2	List and Explain Applications of Regular Expressions	**III**	**CO2**
2D – 3	What is the Transition Diagram and strings for the given Regular Expression 01*+10*	**V**	**CO2**
2D – 4	Evaluate the Operators of Regular Expressions with example	**V**	**CO2**
2D – 5	How Regular Expression approach is different from Finite Automata for describing languages	**IV**	**CO2**
2D – 6	Explain Relation ship between Finite Automata and Regular Expression with Diagram	**II**	**CO2**
2D – 7	Convert the given Regular Expression 1*01(0+11)* into DFA Transition Diagram	**I**	**CO2**
2D – 8	List and Explain Applications of Regular expressions	**I**	**CO2**
2D – 9	What are the algebraic laws that are used for Regular Expressions	**II**	**CO2**
2D - 10	Evaluate the steps in conversion of finite automata to Regular Expression.	**V**	**CO2**
2D – 11	Explain the types of Grammar in detail with examples.	**VI**	**CO2**
2D – 12	What do you mean by Pumping Lemma? What are the Applications of Pumping Lemma.	**I**	**CO2**
2D – 13	Create Context Free Grammar. Explain the components of CFG	**VI**	**CO2**
2D – 14	Explain Leftmost and Rightmost Derivations of CFG with Examples	**VI**	**CO2**
2D – 15	What is Parse Tree? How to construct Parse Tree	**I**	**CO2**
2D – 16	Evaluate Ambiguity in Grammar? How to remove Ambiguity for the Grammar	**V**	**CO2**

UNIT – III			
OBJECTIVE QUESTIONS			
SL.NO	**MULTIPLE CHOOSE QUESTIONS**	**BT Level**	**Course Outcome**
3C - 1	Which of the following is NOT a component of a pushdown automaton (PDA)? a) Finite control b) Input tape c) Stack d) Transition function	**I**	**CO3**
3C - 2	What is the main difference between a finite automaton (FA) and a pushdown automaton (PDA)? a) FA has an input tape, while PDA doesn't. b) PDA has a stack, while FA doesn't. c) FA can recognize context-free languages, while PDA can't. d) PDA can recognize regular languages, while FA can't.	**II**	**CO3**
3C - 3	In a PDA, the stack allows for: a) LIFO (Last-In-First-Out) operations b) FIFO (First-In-First-Out) operations c) Both LIFO and FIFO operations d) Random access operations	**I**	**CO3**
3C - 4	Which of the following describes the type of languages recognized by pushdown automata? a) Regular languages b) Context-free languages c) Context-sensitive languages d) Unrestricted languages	**I**	**CO3**
3C - 5	What is the role of the stack in a pushdown automaton? a) To store the input symbols. b) To store intermediate results during computation. c) To represent the state transitions. d) To handle memory management.	**II**	**CO3**
3C - 6	Which of the following describes the type of languages recognized by pushdown automata? a) Regular languages b) Context-free languages c) Context-sensitive languages d) Unrestricted languages	**I**	**CO3**
3C - 7	Which of the following operations can a pushdown automaton perform? a) Reading input symbols b) Pushing symbols onto the stack c) Popping symbols from the stack d) All of the above	**I**	**CO3**
3C - 8	Turing machines are capable of recognizing which type of languages? a) Regular languages b) Context-free languages c) Context-sensitive languages d) Recursively enumerable languages	**II**	**CO3**
3C - 9	In a Turing machine, the tape is initially:	**I**	**CO4**

	a) Empty b) Contains the input string c) Contains only the start symbol d) Contains random symbols		
3C - 10	The set of states in a Turing machine includes: a) Start state only b) Accept state only c) Reject state only d) Start, accept, and reject states	**I**	**CO4**
3C - 11	What is the Turing machine's response when it halts in an accept state? a) Accepts the input string b) Rejects the input string c) Continues computation d) None of the above	**IV**	**CO4**
3C - 12	In a non-deterministic Turing machine (NTM), a transition can have: a) Exactly one possible outcome b) Multiple possible outcomes c) No outcome d) A different alphabet	**I**	**CO4**
3C - 13	The concept of Turing machines is fundamental in understanding: a) Computability theory b) Artificial intelligence c) Data structures d) Digital circuits	**II**	**CO4**
3C – 14	Which of the following problems is known to be undecidable ? a) Sorting a list of integers b) Determining whether a given context-free grammar generates an infinite language c) Finding the shortest path in a weighted graph d) Solving systems of linear equations	**I**	**CO4**
3C - 15	Undecidability implies that there are problems for which: a) No solution exists b) Only approximate solutions exist c) No algorithm can guarantee a correct solution for all instances d) Only deterministic solutions exist	**IV**	**CO4**
FILL IN THE BLANKS			
3F – 1	In a pushdown automaton (PDA), the finite control, input tape, and stack collectively form the _______.	**I**	**CO3**
3F - 2	Pushdown automata are capable of recognizing languages beyond those recognized by _______.	**IV**	**CO3**
3F – 3	The stack in a pushdown automaton (PDA) allows for _______ operations.	**I**	**CO3**
3F – 4	The type of languages recognized by pushdown automata are known as _______ languages.	**IV**	**CO3**
3F – 5	An ε-transition in a pushdown automaton allows the machine to read an input symbol without _______ any input.	**I**	**CO3**
3F – 6	A pushdown automaton can perform operations such as reading input symbols, pushing symbols onto the stack, and _______ symbols from the stack	**I**	**CO3**
3F – 7	In a Turing machine, the ___________ is the main control unit responsible for determining the machine's behavior based on its	**IV**	**CO3**

	current state and the symbol read from the tape.		
3F – 8	The tape in a Turing machine is divided into ___________ cells, where each cell can hold a single symbol from the tape alphabet.	**I**	**CO3**
3F – 9	The function of the ___________ is to transition the Turing machine from one state to another based on the current state and symbol read from the tape.	**II**	**CO4**
3F – 10	Turing machines are central to the study of ___________ theory, which deals with the fundamental capabilities and limitations of computational devices.	**I**	**CO4**
3F – 11	The set of states in a Turing machine includes a ___________ state, which indicates the initial state of the machine.	**I**	**CO4**
3F – 12	Undecidability refers to the property of a problem for which no algorithm exists to determine a __________ answer.	**I**	**CO4**
3F – 13	The concept of undecidability was formally established by the proof of the __________ theorem.	**I**	**CO4**
3F – 14	Undecidabilty is a fundamental limitation of __________.	**V**	**CO4**
3F – 15	The concept of undecidability challenges the notion that all problems can be solved with sufficient __________ resources.	**I**	**CO4**
	MATCH THE FOLLOWING		
3M – 1	1. Push a. add a symbol onto the top of the stack. 2. Pop b. removes the symbol from the top of the stack 3. Read c. examine the current symbol on the input tape. 4. Accept d. State reached when the stack is empty.	**I**	**CO3**
3M – 2	1. Finite Control a. Stores symbols 2. Stack b. Determines the next move. 3. Input Tape c. Controls the behavior 4. Transition Functioned. Allows for the storage of symbols	**I**	**CO3**
3M – 3	1. Read a. Examines the current symbol on the tape 2. Write b. Modifies the symbol currently under the tape head 3. Move c. Changes the position of the tape head 4. Transition d. Determines the next move	**II**	**CO3**
3M – 4	1. Non-determinism a. it can have multiple possible transitions 2. Determinism b. it can recognize and simulate any other Turing machine. 3. Unboundedness c. tape being infinite in length. 4. Universality d. uniquely determined	**I**	**CO4**
3M – 5	1.RE a. Universal language 2. Recursive language b. Un-decidable problems 3.Non-recursive language c. Turing Machine 4.Lu d. Decidable Problems	**I**	**CO4**
	10 – MARKS QUESTIONS		
3D – 1	Evaluate Push Down Automata and Explain each component of PDA	**V**	**CO3**
3D – 2	What are the different Approaches by which a language can be accepted by PDA	**II**	**CO3**
3D – 3	How to Construct PDA From the given Grammar	**II**	**CO3**
3D – 4	Convert the Grammar Sà 0AA Aà 0S/1S/0 into PDA that accepts the same language by empty stack	**I**	**CO3**
3D – 5	How to Evaluate Turing Machine .Describe the Turing Machine with Diagram	**V**	**CO3**
3D – 6	Explain each notation of Turing Machine. give some examples of problems that can and cannot be solved by a Turing machine?	**II**	**CO3**

3D – 7	Discuss the concept of computability and its relationship to Turing machines.	**II**	**CO3**
3D – 8	Discuss the halting problem for Turing machines. Explain why it is undecidable, and explore its significance in theoretical computer science.	**IV**	**CO3**
3D – 9	Explore the connection between Turing machines and formal language theory. Discuss how Turing machines are used to define and recognize formal languages.	**II**	**CO4**
3D – 10	Design a Turing machine to compute a specific function or solve a particular problem. Explain the machine's states, transition function, and tape alphabet, and analyze its computational complexity.	**I**	**CO4**
3D – 11	Explain the concept of undecidability in the theory of computation.	**I**	**CO4**
3D – 12	Provide examples of undecidable problems and discuss the implications of undecidability for computer science.	**II**	**CO4**
3D – 13	Explain Undecidable problems about Turing machines	**II**	**CO4**
3D – 14	Define the following a) Undecidability b) Undecidable Problems	**II**	**CO4**
3D – 15	What are the Classes of Languages that are accepted by Turing Machine	**II**	**CO4**

UNIT IV			
OBJECTIVE QUESTIONS			
SL.NO		**BT Level**	**Course Outcome**
4C – 1	What is the primary function of a compiler? a) Interpretation of source code b) Execution of machine code c) Translation of high-level code to machine code d) Debugging of programs	II	**CO5**
4C – 2	Which phase of the compiler translates the source code into intermediate code? a) Lexical Analysis b) Syntax Analysis c) Semantic Analysis d) Intermediate Code Generation	II	**CO5**
4C – 3	Which phase of the compiler checks for syntax errors and constructs a parse tree? a) Lexical Analysis b) Syntax Analysis c) Semantic Analysis d) Code Generation	I	**CO5**
4C – 4	Which phase of compilation involves associating attributes with syntactic constructs? a) Lexical Analysis b) Syntax Analysis c) Semantic Analysis d) Intermediate Code Generation	I	**CO5**
4C – 5	Types of Parsers a) 2 b) 3 c) 4 d) 5	I	**CO5**
4C – 6	Which of the following is true about L-Attributed Syntax-Directed Definitions? a) They cannot handle inherited attributes b) They can handle both inherited and synthesized attributes c) They can only handle synthesized attributes d) They are used exclusively for code optimization	I	**CO5**
4C – 7	Which of the following is parser(s) a) Top down Parser b) Bottom up parser c) Universal Parser d) All of the above	I	**CO5**

4C – 8	Function of Symbol table is a) Error handling b) Maintain Record of each variable c) Both a and b d) None	II	**CO5**
4C – 9	Which of the following comes under synthesis phase a) Semantic Analysis b) Code generation c) Lexical Analysis d) Syntax Analysis	I	**CO5**
4C – 10	Which phase of compilation involves associating attributes with syntactic constructs? a) Lexical Analysis b) Syntax Analysis c) Semantic Analysis d) Intermediate Code Generation	I	**CO5**
4C – 11	What is the primary role of semantic actions in Syntax-Directed Translation? a) To generate intermediate code b) To specify attribute evaluation rules c) To construct parse trees d) To perform code optimization	I	**CO5**
4C – 12	Function of Lexical Analyzer is a) Read input of Source code b) Type checking c) Code conversion d) optimization	I	**CO5**
4C – 13	Context Free Grammar accepted by which automaton a) Turing Machine b) LBA c) PDA d) Finite Automaton	II	**CO5**
4C – 14	Which of the following is/are Compiler Construction tool? a) Parser generator b) Scanner generator c) Code generator d) All of the above	II	**CO5**
4C – 15	A Grammar consists of a) Terminals b) Non-terminals c) Start symbol, Productions d) All of the above	I	**CO5**
	FILL IN THE BLANKS		
4F – 1	1. The primary function of a compiler is to translate high-level code into _____ code.	**I**	**CO5**
4F – 3	The role of the _____ is to identify tokens and lexemes in the source code.	**II**	**CO5**
4F – 4	Context-Free Grammars are primarily used in the _____ phase of the compiler	**II**	**CO5**
4F – 5	Semantic Analysis is the phase of the compiler responsible for checking for _____ errors.	**I**	**CO5**

4F – 6	The process of identifying keywords and identifiers is performed during _____.	**II**	**CO5**
4F – 7	LR Parsing is more powerful than _____ parsing.	**II**	**CO5**
4F – 8	LALR stands for _____.	**I**	**CO5**
4F – 9	Type checking is performed during _____.	**II**	**CO5**
4F – 10	Evaluation orders for SDDs can vary, such as ____ or ____.	**V**	**CO5**
4F – 11	Three-Address Code typically consists of ____ operands.	**I**	**CO5**
4F – 12	Run-Time Environments manage memory and access to data during program execution, including ____.	**I**	**CO5**
4F – 13	Syntax-Directed Translation involves associating ____ with the productions of a grammar.	**I**	**CO5**
4F – 14	Flex is a tool commonly used to generate _____ analyzers.	**II**	**CO5**
4F – 15	LR Parsing is a type of _____ parsing.	**III**	**CO5**
	MATCH THE FOLLOWING		
4M – 1	1. Lexical Analysis — a. Parsing 2. Syntax Analysis — b. Type checking 3. Semantic Analysis — c. Three address code 4. Intermediate form — d. Scanning	**I**	**CO5**
4M – 2	1. Code Optimization — a. Maps to target Language 2. Code generation — b. improves the Intermediate code 3. Symbol table — c. done at every phase of compilation 4. Error Handling — d. Contains a record of each variable	**I**	**CO5**
4M - 3	1. Lexeme Patterns — a. Sequence of Characters 2. Token — b. Lexical Analyzer 3. Pattern — c. Name, Attribute 4. Lexeme — d. Description of the form	**II**	**CO5**
4M – 4	1. Top-down Parser — a. Read input of source code 2. Bottom- up Parser — b. LL parser 3. LALR — c. LR Parser 4. Lexical Analyzer — d. Look ahead LR	**II**	**CO5**
4M – 5	1. Parse Tree — a. Left to Right scanning of Input 2. Ambiguous Grammar — b. Picture of Derivation 3. LR parser — c. Parse generator 4. Yaac — d. Produce more than One Parse Tree	**III**	**CO5**
	10 – MARKS QUESTIONS		
4D – 1	Provide an overview of the structure of a compiler. What are the main components, and what role does each component play in the compilation process?	**II**	**CO5**

4D – 2	Explain the purpose and operation of the lexical analyzer (lexer) in a compiler.?	**II**	**CO5**
4D – 3	Describe the process of tokenization performed by the lexical analyzer	**III**	**CO5**
4D – 4	What are tokens, and how does the lexer identify and classify different types of tokens	**I**	**CO5**
4D – 5	Describe the following a) Context Free Grammar b) Symbol table	**II**	**CO5**
4D – 6	What is Compiler? What are the functions performed by Compiler	**III**	**CO5**
4D – 7	What is the Role of Lexical Analyzer in Lexical Analysis phase	**IV**	**CO5**
4D – 8	What is Parsing? Explain Top-down and Bottom-up parsing	**I**	**CO5**
4D – 9	What do you Mean by LR Parsing? Describe various LR parsers	**I**	**CO5**
4D – 10	Describe the following a) Input buffer b) Lex	**I**	**CO5**
4D – 11	Explain Syntax Analysis phase in Compilation	**I**	**CO5**
4D – 12	Briefly describe the structure of a compiler with Diagram	**I**	**CO5**
4D – 13	Analyze the role of LR parsing table in the parsing process, particularly The actions and goto function	**V**	**CO5**
4D – 14	Evaluate the process of constructing an LR parsing table and the Challenges involved in this process	**V**	**CO5**
4D – 15	Explain the primary concept of LR Parsing, in contrast to LL parsing	**I**	**CO5**

UNIT – V			
OBJECTIVE QUESTIONS			
SL.NO	**MULTIPLE CHOOSE QUESTIONS**	**BT Level**	**Course Outcome**
5C – 1	What is the primary purpose of Syntax-Directed Definitions (SDDs)? a) Generating machine code directly b) Describing the syntax of programming languages c) Associating attributes with syntactic constructs d) Optimizing intermediate code	**II**	**CO6**
5C – 2	Which of the following is NOT a type of attribute in Syntax-Directed Translation?	**I**	**CO6**

	a) Synthesized b) Inherited c) Semantic d) L-Attributed		
5C – 3	What are Syntax-Directed Translation Schemes (SDTS) used for? a) Parsing input strings b) Generating machine code c) Associating attributes with productions in a grammar d) Constructing parse trees	**I**	**CO6**
5C – 4	What is the main advantage of using intermediate code in a compiler? a) It simplifies the implementation of code optimization techniques b) It reduces the complexity of semantic analysis c) It allows for easier integration of external libraries d) It improves the performance of the generated machine code	**II**	**CO6**
5C – 5	What is the relationship between Syntax-Directed Definitions (SDDs) and context-free grammars? a) SDDs are more powerful than context-free grammars b) SDDs are a type of context-free grammar c) Context-free grammars are a type of SDD d) SDDs are unrelated to context-free grammars	**II**	**CO6**
5C – 6	What does a Syntax-Directed Translation Scheme (SDTS) specify? a) A set of rules for constructing parse trees b) A set of rules for associating attributes with grammar productions c) A set of rules for generating machine code d) A set of rules for lexical analysis	**II**	**CO6**
5C – 7	In Syntax-Directed Translation, what is the purpose of the evaluation order? a) To determine the order of attribute evaluation b) To determine the order of grammar rule application c) To determine the order of parse tree construction d) To determine the order of code optimization	**II**	**CO6**
5C – 8	What is the primary advantage of using Three-Address Code (TAC) over other intermediate representations? a) It is easier to generate from source code b) It is more compact and efficient c) It is easier to optimize d) It is easier to interpret	**I**	**CO6**
5C – 9	Which of the following is true about L-Attributed Syntax-Directed Definitions? a) They cannot handle inherited attributes b) They can handle both inherited and synthesized attributes c) They can only handle synthesized attributes d) They are used exclusively for code optimization	**II**	**CO6**
5C – 10	What is an inherited attribute in Syntax-Directed Translation? a) An attribute inherited from the parent node to its children b) An attribute inherited from the children to the parent node	**II**	**CO6**

	c) An attribute that is synthesized from the children d) An attribute that is synthesized from the parent		
5C - 11	In a run-time environment, what is the role of the heap? a) To store local variables and function parameters b) To store global variables and static data c) To manage memory dynamically allocated during program execution d) To maintain a record of function calls and return addresses	**I**	**CO6**
5C – 12	What is the difference between static and dynamic memory allocation? a) Static memory allocation is performed at compile time, while dynamic memory allocation is performed at run time b) Static memory allocation involves allocating memory on the stack, while dynamic memory allocation involves allocating memory on the heap c) Static memory allocation is more efficient than dynamic memory allocation d) Static memory allocation requires explicit deallocation, while dynamic memory allocation does not	**I**	**CO6**
5C – 13	What is the purpose of heap management in a run-time environment? a) To allocate memory for local variables b) To allocate memory for global variables c) To manage dynamically allocated memory d) To optimize stack space usage	**I**	**CO6**
5C – 14	Which of the following is NOT a common method for heap management? a) First-fit allocation b) Best-fit allocation c) Quick-fit allocation d) Stack-based allocation	**II**	**CO6**
5C – 15	What is a potential downside of heap-based memory allocation? a) It can lead to fragmentation of memory b) It requires more memory than stack allocation c) It is slower than stack allocation d) It can cause stack overflow errors	**II**	**CO6**
FILL IN THE BLANKS			
5F – 1	Syntax-Directed Translation involves associating ____ with the productions of a grammar.	**I**	**CO6**
5F – 2	Run-Time Environments manage memory and access to data during program execution, including ____.	**I**	**CO6**
5F – 3	Stack Allocation of Space involves allocating memory for ____.	**II**	**CO6**
5F – 4	Heap Management deals with allocating and deallocating memory for ____.	**I**	**CO6**
5F – 5	Syntax-Directed Translation Schemes are used to ____ semantic actions with productions.	**I**	**CO6**

5F – 6	Run-Time Environments handle aspects like ____ and ____.	**II**	**CO6**
5F – 7	Stack Allocation of Space is efficient for ____ with a fixed lifetime.	**I**	**CO6**
5F – 8	In a syntax-directed definition, ____ are associated with grammar symbols.	**I**	**CO6**
5F - 9	Access to Nonlocal Data on the Stack requires efficient ____ mechanisms.	**I**	**CO6**
5F – 10	Three-Address Code facilitates ____ for various architectures.	**I**	**CO6**
5F – 11	Run-Time Environments ensure proper ____ and ____ of resources.	**I**	**CO6**
5F – 12	DAG stands for___________________________	**I**	**CO6**
5F – 13	In___________________ there is at most one operator on the right side of an Instruction	**I**	**CO6**
5F – 14	Type checking can take on two forms_________________and _____-	**I**	**CO6**
5F – 15	________________Environment deals with a variety of issues such as the layout and allocation of storage locations.	**II**	**CO6**
	MATCH THE FOLLOWING		
5M – 1	1. Heap a. Allocates / Deallocates space 2. Memory manager b. finds the space with in the heap 3. Garbage collection c. modifies the collection of objects in the heap. 4. Mutator d. portion of store	**III**	**CO6**
5M – 2	1.DAG a. op, arg1, arg2 2.Triples b. syntax trees 3.type checking c. data structure with named field 4. Record d. behavior of a program	**III**	**CO6**
5M – 3	1. SDD a. when two type expressions equals 2. Annotated parse tree b. depicts the flow of information 3. Dependency graph c. showing values of it's attributes 4. Type equivalence d. Context free Grammar	**II**	**CO6**
5M – 4	1. Three address code a. Memory management 2. Runtime management b. Optimization phase 3. SDD c. Associate Semantic actions 4. SDT d. Semantic actions	**II**	**CO6**
5M – 5	1. Local variable stored a. Run time 2. Activation Record b. Compile time 3. Static Memory allocation c. Data Structure holds information about current function call 4. Dynamic Memory allocation d. on the stack	**II**	**CO6**
	10 – MARKS QUESTIONS		
5D – 1	What is SDD? Explain the attributes of SDD	**II**	**CO6**
5D – 2	Illustrate the Dependency graph with Example	**III**	**CO6**

5D – 3	Describe the following a) S-attributed SDD b) L-attributed SDD	**II**	**CO6**
5D – 4	Define DAG. What is the corresponding DAG for the given Expression a+a*(b-c)+(b-c)*d	**II**	**CO6**
5D – 5	What is Three address code? Give an example	**II**	**CO6**
5D – 6	What is Heap? What are functions performed by Heap Manager	**II**	**CO6**
5D – 7	Explain the concept of Syntax directed Translation and how it differs from direct interpretation of the source code	**II**	**CO6**
5D – 8	Discuss the basic principles of Syntax directed translation	**II**	**CO6**
5D – 9	Compare and Contrast SDD and SDT	**II**	**CO6**
5D – 10	Explain about Syntax directed Translation Schemes	**V**	**CO6**
5D – 11	Explain the following a) Dependency graph b) S-attributed and L-attributed Definitions	**II**	**CO6**
5D – 12	Explain Storage organization of Run time Environment	**I**	**CO6**
5D – 13	Give the evaluation order of SDT with an example	**II**	**CO6**
5D – 14	Describe the following a) Activation trees b) Activation Records	**I**	**CO6**
5D – 15	illustrate Static versus Dynamic storage allocation of data and the Memory Hierarchy of Computer	**II**	**CO6**

R22CSM2212 – AUTOMATA THEORY AND COMPILER DESIGN

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

Recognized under 2(f) and 12(B) of UGC Act 1956

II B.Tech-II Semester–End Examinations (Model paper)

AUTOMATA THEORY AND COMPILER DESIGN

(CSE-AI&ML)

Duration: 3 Hrs **Max Marks: 60 M**

Answer All of the following questions		**Marks: 10Qx1M = 10M**		
		Marks	**Course Outcome**	**BT Level**
1	What is Automata Theory ?	1	CO1	I
2	Define Transition diagram ?	1	CO1	I
3	Discuss Regular Expression	1	CO2	VI
4	How Regular Expression approach is different from Finite Automata for describing languages	1	CO2	IV
5	Give some examples of problems that can be solved by a Turing machine?	1	CO3	II
6	Define Push Down Automata and List each component of PDA	1	CO4	I
7	What are tokens, and how does the lexer identify and classify different types of tokens	1	CO5	I
8	Explain Syntax Analysis phase in Compilation	1	CO5	V
9	What is SDD? Explain the attributes of SDD	1	CO6	I

10	Discuss the basic principles of Syntax directed translation	1	CO6	II

Section – A

Section – B

Answer any *FIVE* questions choosing at least one from each Unit 5Qx10M = 50M	Marks:		
	Marks	**Course Outcome**	**BT Level**
UNIT-I 11. Why do we study Automata Theory. Discuss about Automata Theory **(OR)** 12. Evaluate the steps in converting NFA to DFA with Example	10	CO1	VI, V
UNIT-II 13. What is Regular Expression? Discuss Finite automata with Regular Expression **(OR)** 14. What do you mean by Pumping Lemma? What are the Applications of Pumping Lemma.	10	CO2	I,III
UNIT – III 15. Convert the Grammar Sà 0AA Aà 0S/1S/0 into PDA that accepts the same language by empty stack **(OR)** 16. Provide examples of undecidable problems and discuss the implications of undecidability for computer science.	10	CO3 CO4	III,IV
UNIT – IV 17. Provide an overview of the structure of a compiler. What are the main components, and what role does each component play in the compilation process? **(OR)** 18. Evaluate the process of constructing an LR parsing table and the Challenges involved in this process	10	CO5	I,V
UNIT – V 19. Define DAG. What is the corresponding DAG for the given Expression? a+a*(b-c) +(b-c) *d **(OR)**	10	CO6	I,II

20. illustrate Static versus Dynamic storage allocation of data and the Memory Hierarchy of Computer			

S.NO	Bloom Taxonomy Level	No. Of. Questions	%	Name & Signature of Paper Setter
1	Fundamental Knowledge from level I & II	5	50	
2	Knowledge on application & Analysis from level III & IV	2	20	
3	Critical thinking and ability to design from level V & VI	3	30	

---------------------------***********--------------------------

BR-22 HT.No. 22U11A6906 D4

QC2472

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi) - Recognized under 2(f) and 12(B) of UGC Act 1956

III B.Tech. I Semester (REGULAR) End Examinations, JANUARY– 2025.

(R22CSM2212) AUTOMATA THEORY AND COMPILER DESIGN

07/01/2025 (For IoT) Day-4(FN)

Duration: 3 Hrs **Maximum Marks: 60M**

Blooms Taxonomy : (I-Remembering, II-Understanding, III-Applying, IV-Analyzing, V-Evaluating and VI-Creating)

Course Outcomes : CO

PART – A

Answer ALL the following questions. (10Qx1M=10M)

1. a)	How does an NFA differ from a DFA?	II	CO[illegible]
b)	How does an epsilon transition differ from a regular transition in an NFA?	III	CO1
c)	Define a Regular expression.	II	CO[illegible]
d)	Explain the concept of a Parse tree in CFGs.	I	CO[illegible]
e)	What type of language does a Pushdown Automaton accept?	I	C[illegible]
f)	What is a Recursively enumerable language?	II	CO3
g)	Mention the role of the front end in a compiler.	I	CO4
h)	How does a lexical analyzer recognize tokens?	II	CO4
i)	What is the difference between Synthesized and Inherited attributes?	III	CO5
j)	What is an Abstract syntax tree?	II	CO5

PART – B

Answer FIVE questions choosing at least one from each unit. (5Qx10M=50M)

UNIT-I

2a.	Describe the different components of finite automata, including states, transitions, and accept states, with a labeled diagram with examples.	III	CO1
	(OR)		
2b.	Define Deterministic Finite Automata (DFA) and explain how it processes strings with a simple example.	III	CO1

UNIT-II

3a.	Using the Pumping Lemma, demonstrate that the language $L= \{a^n b^n \mid n >= 0\}$ is not regular. Explain each step.	II	CO2
	(OR)		
3b.	Convert the given finite automaton to a regular expression and explain each step of your process.	V	CO2

$$Q = \{q_0, q_1, q_2\}, \quad \Sigma = \{a, b\}, \quad q_0 \to q_1 \xrightarrow{a} q_2, \quad q_2 \to q_0$$

UNIT-III

4a.	Construct a PDA for the language $L= \{a^n b^n \mid n \geq 1\}$, where the PDA accepts by final state. Clearly explain each step in the PDA's operation for an example string "aabb".	III	CO3
	(OR)		
4b.	Compare the operation of a deterministic Turing Machine with that of a Non-deterministic Turing Machine.	IV	CO3

P.T.O

UNIT-IV

5a. Compare and contrast recursive descent and predictive parsing as two forms of top-down parsing. IV CO4

(OR)

5b. Construct the CLR parsing table for the following grammar: V CO4

S→L=R | R
L→*R | id
R→L.

UNIT-V

6a. Write the quadruple, triple, indirect triple for the given expression. IV CO5

-(a*b) + (c + d)- (a + b + c + d).

(OR)

6b. Explain in detail how an L-attributed grammar can be converted into a translation scheme. V CO5

*****