



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



NAAC
NATIONAL ASSESSMENT AND
ACCREDITATION COUNCIL



HANDOUT

II Year II Semester

DEPARTMENT OF CSE-AI&ML

ACADEMIC YEAR 2025-2026



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 CSE-AI&ML

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

(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 VISION

To develop competent engineers in the domain of Artificial Intelligence & Machine Learning for note worthy contributions to the society.

DEPARTMENT MISSION

- DM1:** To strengthen academic collaborations for better exposure.
- DM2:** Promote professional environment to imbibe ethical values and entrepreneurial skills
- DM3:** Encourage research & development by emphasizing on innovation
- DM4:** Facilitate Industry-Institute collaboration for mutual benefit

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

- PEO1:** Generate contributions towards advancements in Artificial Intelligence and Machine Learning
- PEO2:** Promote design, research and implementation of products through strong communication skills, leadership and entrepreneurial skills.
- PEO3:** Apply basic principles and practices of AIML to successfully complete Software related projects to meet customer business objectives and/or productively engage in research.

Program Outcomes(POs)

PO1	Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design / Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Program Educational Objectives(PEO's)

PEO1	PEO1: Higher studies: Pursue innovation oriented higher education with high degree of research.
PEO2	Professional Career: Engage in best practices to get employed in Top notch companies.
PEO3	Domain Knowledge: Demonstrated professionalism to analyse and design the data science solutions.
PEO4	Lifelong Learning: Continual Learning with ethical principles to contribute to the human well-being and society.

Program Specific Outcomes(PSO's)

PSO1	Register mathematical methodology to crack problems using suitable data structures.
PSO2	Competence to design and develop software for web based and mobiles androids under real world environment.
PSO3	Skill to design the algorithms for machine learning, data compression can be used in different applications.

COURSE OUTCOMES (CO's)

Academic Year: 2025-2026

Class : II YEAR-II SEM.

Course Name : AUTOMATA THEORY AND COMPILER DESIGN

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

C222.1	• Able to employ finite state machines for modeling and solving computing problems.
C222.2	• Able to design context free grammars for formal languages.
C222.3	• Able to distinguish between decidability and undecidability.
C222.4	• Able to Understand The role of Turing Machine
C222.5	• Demonstrate the knowledge of patterns, tokens & regular expressions for lexical analysis.
C222.6	• Acquire skills in using lex tool and design LR parsers

COURSE ARTICULATION MATRIX

Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C222.1	3	2	-	-	-	-	-	-	-	-	-	2	3	3	-
C222.2	3	3	2	-	-	-	-	-	-	-	-	-	3	2	-
C222.3	3	2	3	3	-	-	-	-	-	-	-	2	3	2	-
C222.4	3	2	2	3	-	-	-	-	-	-	-	2	3	3	-
C222.5	3	-	2	2	3	-	-	-	-	-	-	2	3	2	-
C222.6	3	-	-	-	3	-	-	-	-	-	-	-	3	3	-
C222	3	1.5	1.5	1.3	1	-	-	-	-	-	-	1.3	3	2.5	-

INSTITUTION ACADEMIC CALENDAR



SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY
(An Autonomous Institution under UGC, New Delhi)
Recognized under 2(f) and 12(B) of UGC Act 1956
NBA & NAAC Accredited, Approved by AICTE and Permanently affiliated to JNTUH
Sheriguda (V), Ibrahimpatnam, R.R. Dist, Hyderabad - 501 510

D4

ANNEXURE -III

BR-22

Lr.No.SICET/AUTO/DAE/IInd B.Tech Academic Calendar/30/2025

B.TECH II-YEAR I-SEM & II-SEM ACADEMIC CALENDAR **(FOR ACADEMIC YEAR : 2025-26)**

Academic Calendar for B.Tech – IInd Year Students (2024 - 25 Batch), BR-22 Regulation.

I – Semester

S.No.	EVENT	PERIOD	DURATION
1.	Commencement of class work.	01.08.2025 (Monday)	
2.	1 st Spell of Instructions for covering First Two and a half Units.	01.08.2025 – 07.10.2025	9 Weeks
3.	I Mid Term Examinations.	08.10.2025 – 10.10.2025	3 Days
4.	Submission of I Mid Term Examination Marks.	16.10.2025	
5.	2 nd Spell of Instructions for Remaining Two and a half Units.	13.10.2025 – 13.12.2025	9 Weeks
6.	II Mid Term Examinations.	15.12.2025 – 17.12.2025	3 Days
7.	Preparation Holidays and Practical Examinations	18.12.2025 – 27.12.2025	1 Week
8.	Submission of II Mid Term Examination Marks.	23.12.2025	
9.	I Semester End Examinations.	29.12.2025 – 10.01.2026	2 Weeks
Commencement of Class-Work for II B.Tech - II Semester 19.01.2026			

Note: No of Working/instructional days : 92

II – Semester

S.No.	EVENT	PERIOD	DURATION
1.	Commencement of class work.	19.01.2026 (Monday)	
2.	1 st Spell of Instructions for covering First Two and a half Units.	19.01.2026 – 14.03.2026	8 Weeks
3.	I Mid Term Examinations.	16.03.2026 – 18.03.2026	3 Days
4.	Submission of I Mid Term Examination Marks.	24.03.2026	
5.	2 nd Spell of Instructions for Remaining Two and a half Units.	20.03.2026 – 09.05.2026	7 Weeks
6.	Continuation of remaining 2 nd Spell of Instructions	25.05.2026 – 10.06.2026	2 Weeks
7.	II Mid Term Examinations.	11.06.2026 – 13.06.2026	3 Days
8.	Preparation Holidays and Project Evaluation.	15.06.2026 – 20.06.2026	1 Week
9.	Submission of II Mid Term Examination Marks.	18.06.2026	
10.	II Semester End Examinations	22.06.2026 – 04.07.2026	2 Weeks

Note: No of Working/instructional days : 92


ACE

Controller of Examination
Sri Indu College of Engineering & Technology
(An Autonomous Institution Under UGC)
Sheriguda (V), Ibrahimpatnam, R.R. Dist-501510

Sign: 

Dr. M.V.S.S. Giridhar

Prof. of CEA; JNTUH Nominee

(Copy to DAP&E and Copy to all the Dept. Heads)


CE

DIRECTOR
Sri Indu College of Engineering & Technology
(An Autonomous Institution Under UGC)
Sheriguda (V), Ibrahimpatnam, R.R. Dist-501510

Sign: 

Dr. T. Venu Gopal

Prof. of CSE; JNTUH Nominee


DIRECTOR


PRINCIPAL

PRINCIPAL
Sri Indu College of Engineering & Technology
(An Autonomous Institution Under UGC)
Sheriguda (V), Ibrahimpatnam, R.R. Dist-501510

Sign: 

Dr. D. Ramesh

Prof. of CSE; JNTUH Nominee

DEPARTMENT ACADEMIC CALENDAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI & ML) DEPARTMENT CALENDAR - 2025-2026 (SEMESTER-II)

SUNDAY		Dec-25			1		1							Jun-26		
MONDAY	1				2		2					1				
TUESDAY	2				3	IYRCRT TRAINING	3	IYRCRT TRAINING		Apr-26			2	IYRTRAINING		Jul-26
WEDNESDAY	3			Jan-26	4		4	HOLIDAY	1				3		1	II-II&III-II SEM ENDEXAMS
THURSDAY	4		1	NEWYEAR HOLIDAY	5		5		2	IYRTRAINING		May-26	4		2	II-II&III-II SEM ENDEXAMS
FRIDAY	5	AICLUBMEETING	2		6	IYRCRT TRAINING	6	IYRCRT TRAINING	3	GOODFRIDAY HOLIDAY	1	IV-IIMID II EXAMS	5	IYRTRAINING	3	II-II&III-II SEM ENDEXAMS
SATURDAY	6	DEPARTMENT MEETING	3		7	AICLUBMEETING	7		4	AICLUBMEETING	2	IV-IIMID II EXAMS	6		4	II-II&III-II SEM ENDEXAMS
SUNDAY	7	HOLIDAY	4	HOLIDAY	8	HOLIDAY	8	HOLIDAY	5	HOLIDAY	3	HOLIDAY	7	HOLIDAY	5	HOLIDAY
MONDAY	8		5		9		9		6		4	IV-II PREPARATION HOLIDAYS	8		6	
TUESDAY	9		6		10	II YR CRTTRAINING	10	II YR CRTTRAINING	7	IYRTRAINING	5	IV-II PREPARATION HOLIDAYS	9		7	
WEDNESDAY	10		7		11		11		8		6	IV-II PREPARATION HOLIDAYS	10		8	
THURSDAY	11		8		12		12		9		7	SUBMISSION OF MID-IIMARKS	11	II-II&III-IIMIDIIEXAM	9	
FRIDAY	12		9		13	II YR CRTTRAINING	13	II YR CRTTRAINING	10	IYRTRAINING	8	IV-II PREPARATION HOLIDAYS	12	II-II&III-IIMIDIIEXAM	10	
SATURDAY	13		10	DEPARTMENTMEETING	14		14		11	DEPARTMENTMEETING	9	IV-II PREPARATION HOLIDAYS	13	II-II&III-IIMIDIIEXAM	11	DEPARTMENTMEETING
SUNDAY	14	HOLIDAY	11	HOLIDAY	15	HOLIDAY	15	HOLIDAY	12	HOLIDAY	10	HOLIDAY	14	HOLIDAY	12	HOLIDAY
MONDAY	15	IV-IICOMMENCEMENTWORK	12		16	IV-IIMIDIIEXAM	16	II&III-IIMIDIIEXAM	13		11	IV-IISEMEXAMS	15	II-II&III-IIPREPHOILDAYS&PROJECT EVALUATION	13	
TUESDAY	16		13	PONGALHOLIDAYS	17	IV-II MID IEXAM&IYRTRAINING	17	II&III-IIMIDIIEXAM,IYRTRAINING	14	DR.AMBEDKAR JAYANTIHOLIDAY	12	IV-IISEMEXAMS	16	II-II&III-IIPREPHOILDAYS&PROJECT EVALUATION	14	
WEDNESDAY	17		14	PONGALHOLIDAYS	18	IV-IIMIDIIEXAM	18	II&III-IIMIDIIEXAM	15		13	IV-IISEMEXAMS	17	II-II&III-IIPREPHOILDAYS&PROJECT EVALUATION	15	
THURSDAY	18		15	PONGALHOLIDAYS	19	IV-II2NDSPELL	19		16		14	IV-IISEMEXAMS	18	II-II&III-IIPREPHOILDAYS&PROJECT EVALUATION	16	
FRIDAY	19		16	PONGALHOLIDAYS	20	II YR CRTTRAINING	20	II-II&III-IISPELLINSTRUCTIONS	17	IYRTRAINING	15	IV-IISEMEXAMS,	19	II-II&III-IIPREPHOILDAYS&PROJECT EVALUATION	17	
SATURDAY	20		17	PONGALHOLIDAYS	21		21		18		16	IV-IISEMEXAMS	20	II-II&III-IIPREPHOILDAYS&PROJECT EVALUATION	18	
SUNDAY	21	HOLIDAY	18	HOLIDAY	22	HOLIDAY	22	HOLIDAY	19	HOLIDAY	17	HOLIDAY	21	HOLIDAY	19	HOLIDAY
MONDAY	22		19	II&III-IICOMMENCEMENTWORK	23		23		20		18		22	II-II&III-IISEMEXAMS	20	

TUESDAY	23		20		24	SUBMISSION OF IV-II MID IMARKS, II YR TRAINING	24	SUBMISSION OF III- IIMIDIMARKS	21	IIYR TRAINING	19	IIYR TRAINING	23	II-II&III- IISEM ENDE XAMS	21	
---------	----	--	----	--	----	---	----	--------------------------------------	----	---------------	----	---------------	----	----------------------------------	----	--

WEDNESDAY	24		21		25		25		22		20		24	II-II&III-II SEM ENDEXAMS	22	
THURSDAY	25	CHRISTMAS HOLIDAY	22		26		26		23		21		25	II-II&III-II SEM ENDEXAMS	23	
FRIDAY	26	BOXINGDAY HOLIDAY	23	AICLUBMEETING	27	IIYR CRT TRAINING	27	IIYR TRAINING	24	IIYR TRAINING	22	IIYR TRAINING	26	MUHARRAM HOLIDAY	24	
SATURDAY	27	DEPARTMENT MEETING	24		28	DEPARTMENT MEETING	28		25		23		27	II-II&III-II SEM ENDEXAMS	25	
SUNDAY	28	HOLIDAY	25	HOLIDAY		HOLIDAY	29	HOLIDAY	26	HOLIDAY	24	HOLIDAY	28	HOLIDAY	26	HOLIDAY
MONDAY	29		26	REPUBLICDAY			30		27		25		29	II-II&III-II SEM ENDEXAMS	27	
TUESDAY	30		27	II&IIIYR CRT TRAINING			31	IIYR TRAINING	28	IIYR TRAINING	26	IIYR TRAINING	30	II-II&III-II SEM ENDEXAMS	28	
WEDNESDAY	31	AICLUBMEETING	28	II&IIIYR CRT TRAINING					29	AICLUBMEETING	27	BAKRIDHOLIDAY			29	
THURSDAY			29	II&IIIYR CRT TRAINING					30	IV-IIMIDIEXAMS	28				30	
FRIDAY			30	II&IIIYR CRT TRAINING							29	IIYR TRAINING			31	
SATURDAY			31	II&IIIYR CRT TRAINING							30					
SUNDAY		HOLIDAY		HOLIDAY		HOLIDAY		HOLIDAY		HOLIDAY	31	HOLIDAY		HOLIDAY		HOLIDAY

HOD

PRINCIPAL

AUTOMATA THEORY AND COMPILER DESIGN

SYLLABUS

BR22 – B.TECH. – CSE(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY (An Autonomous Institution under UGC, New Delhi)

B.Tech. - II 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 Automata with Epsilon-Transitions.

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

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

Page 82

BR22 – B.TECH. – CSE(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

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 Chandrashekarán, 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. Loudon, Thomson. Course Technology.



SRI INDU COLLEGE OF ENGINEERING & ECHNOLOGY
LESSON PLAN
(Regulation: R22)
Department of AI&ML

**Prepared on
2026**

Sub. Code & Title	(R22CSM2212) AUTOMATA THEORY AND COMPILER DESIGN		
Academic Year: 2025-2026	Year/Sem./Section	II/II/A&B	
Faculty Name & Designation	Ch PraveenReddy, Assistant Professor		

Unit/ Item No.	Topic (s)	Book Reference	Page (s)		Teaching Methodology	Propose d No. of Periods	Actual Date of Handled	CO/RBT
			From	To				
UNIT-I								
1.1	Introduction to Finite Automata	T1	2	3	Black Board	01		CO1,TB1
1.2	Structural Representations,	T1	4	5	Black Board	01		CO1,TB1
1.3	Automata and Complexity	T1	5	5	Black Board	01		CO1,TB1
1.4	the CentralConcepts of Automata Theory	T1	27	28	Chart	01		CO1,TB1
1.5	Alphabets	T1	28	28	Black Board	01		CO1,TB1
1.6	Strings	T1	28	28	Black Board	01		CO1,TB1
1.7	Languages	T1	29	30	Black Board	01		CO1,TB1
1.8	Problems	T1	30	32	Black Board	01		CO1,TB1
1.9	Non Deterministic Finite automata: Formal Definition	T1	53	54	Black Board	01		CO1.TB1
1.10	application, Text Search	T1	64	64	Black Board	01		CO1.TB1
1.11	Finite Automata with Epsilon-Transitions.	T1	68	70	PPT	01		CO1.TB1
1.12	Deterministic Finite Automata: Definition of DFA	T1	43	43	Black Board	01		
1.13	How a DFA Process Strings The Language of DFA	T1	44	50	Black Board	01		CO1,BTL 2

1.14	Conversion of NFA with ϵ -transitions to NFA without ϵ -transitions.	W1	11	15	Black Board	01		CO1,BTL 5
1.15	Conversion of NFA to DFA	R1	152	156	Black Board	01		CO1,
	Review	Signature of the HOD/Coordinator						

	SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY			Prepared on 2026
	LESSON PLAN			
	(Regulation: R22)			
	Department of AI&ML			
	Sub. Code & Title	(R22CSM2212) AUTOMATA THEORY AND COMPILER DESIGN		
	Academic Year :2025-2026	Year/Sem./Section	II/II A&B	
	Faculty Name & Designation	Ch PraveenReddy, Assistant Professor		

Unit/ Item No.	Topic (s)	Book Reference	Page (s)		Teaching Methodology	Proposed No. of Periods	Actual Date of Handled	CO/RB T
			From	To				
UNIT-II								
2.1	Regular Expressions: Finite Automata and Regular Expressions,	T1	79	94	Black Board	01		CO2,RB2
2.2	Applications of Regular Expressions,	T1	100	104	Black Board	02		CO2,RB1
2.3	Algebraic Laws for Regular Expressions,	T1	106	112	Black Board	01		CO2, RB1
2.4	Conversion of Finite Automata to Regular Expressions.	T1	94	96	Black Board	01		CO2,RB1
2.5	Pumping Lemma for Regular Languages: Statement of the pumping lemma,	T1	118	119	Black Board	01		CO2,RB2
2.6	Applications of the Pumping Lemma.	T1	119	120	Black Board	02		CO2,RB2
2.7	Context-Free Grammars: Definition of Context-Free Grammars	T1	157	159	Black Board	01		CO2,RB2

2.8	Derivations Using a Grammar,	T1	161	163	Black Board	04		CO2,RB5
2.9	Leftmost and Rightmost Derivations,	T1	163	164	Black Board	01		CO2,RB3
2.10	the Language of a Grammar,	T1	165	166	Black Board	02		CO2,RB3
2.11	Parse Trees,	T1	169	170	Black Board	01		CO2,RB2
2.12	Ambiguity in Grammars and Languages.	T1	192	194	Black Board	01		CO2,RB1



SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY
LESSON PLAN
(Regulation: R22)
Department of AI&ML

**Prepared
on
2026**

Sub. Code & Title

(R22CSM2212) AUTOMATA THEORY AND COMPILER DESIGN

Academic Year: 2025-2026

Year/Sem./Section | II/II A&B

Faculty Name & Designation

Ch PraveenReddy, Assistant Professor

Unit/ Item No.	Topic (s)	Book Reference	Page (s)		Teaching Methodology	Proposed No. of Periods	Actual Date of Handled	CO/RBT
			From	To				
UNIT-III								
3.1	Push Down Automata: Definition of the Pushdown Automaton, , ,	T1	205	207	Black Board	01		CO3,RB1
3.2	the Languages of a PDA	T1	214	214	Black Board	01		CO3,RB3
3.3	Equivalence ofPDA and CFG"s	T1	223	224	Black Board	01		CO3,RB2
3.4	Acceptance by final state	T1	214	215	Black Board	01		CO3,RB1
3.5	Turing Machines: Introduction to Turing Machine,	T1	287	288	Black Board	02		CO3,RB4
3.6	Formal Description,	T1	298	298	Black Board	01		CO3,RB1
3.7	Instantaneous description,	T1	299	300	Black Board	01		CO3,RB2
3.8	The language of a Turing machine	T1	306	306	Black Board	01		CO3,RB3
3.9	Undecidability: Undecidability, A Language that is Not Recursively Enumerable,	T1	343	344	Black Board	02		CO3, RB4
3.10	An Undecidable Problem That is RE,	T1	349	349	Black Board	01		CO3,RB3
3.11	Undecidable Problems about Turing Machines	T1	358	358	Black Board	01		CO3,RB4
	Review	Signature of the HOD/Coordinator						



SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY
LESSON PLAN
(Regulation: R22)
Department of AI&ML

**Prepared
on
2026**

Sub. Code & Title

(R22CSM2212) AUTOMATA THEORY AND COMPILER DESIGN

Academic Year: 2025-2026

Year/Sem./Section II/II A&B

Faculty Name & Designation

Ch PraveenReddy, Assistant Professor

Unit/ Item No.	Topic (s)	Book Reference	Page (s)		Teaching Methodology	Proposed No. of Periods	Actual Date of Handled	CO/RBT
			From	To				
UNIT-IV					10			
4.1	Introduction: The structure of a compiler,	R1	4	4	Black Board	01		CO4,RB2
4.2	Lexical Analysis: The Role of the Lexical Analyzer,	R1	109	110	Black Board	01		CO4,RB1
4.3	Input Buffering,	R1	115	116	Black Board	01		CO4,RB2
4.4	Recognition of Tokens,	R1	128	129	Black Board	02		CO4,RB2
4.5	The Lexical-Analyzer Generator Lex,	R1	140	345	Black Board	01		CO4,RB4
4.6	Syntax Analysis: Introduction, Context-Free Grammars, ,	R1	191	205	Black Board	01		CO4,RB2
4.7	Writing a Grammar	R1	209	214	Black Board	01		CO4,RB5
4.8	Top-Down Parsing,Bottom-Up Parsing,	R1	217	238	Black Board	02		CO4,RB3
4.9	Introduction to LR Parsing:	R1	241	248	Black Board	01		CO5,RB4
4.10	Simple LR,	R1	241	241	Black Board	01		CO5,RB1
4.11	More Powerful LR Parsers	R1	259	275	Black Board	01		CO5,RB4
	Review	Signature of the HOD/Coordinator						

	SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY			Prepared On 2026
	LESSON PLAN			
	(Regulation: R22)			
	Department of AI&ML			
	Sub. Code & Title	(R22CSM2212) AUTOMATA THEORY AND COMPILER DESIGN		
	Academic Year: 2025-2026	Year/Sem./Section	II/II A&B	
	Faculty Name & Designation	Ch PraveenReddy, Assistant Professor		

Unit/ Item No.	Topic (s)	Book Reference	Page (s)		Teaching Methodology	Proposed No. of Periods	Actual Date of Handled	CO/RBT
			From	To				
UNIT-V10								
5.1	Syntax-Directed Translation: Syntax-Directed Definitions, ,	R1	303	304	Black Board	01		CO5,RB2
5.2	Evaluation Orders for SDD's	R1	310	314	Black Board	01		CO6,RB1
5.3	Syntax-Directed Translation Schemes,	R1	324	331	Black Board	01		CO5,RB3
5.4	Implementing L-Attributed SDD's.	R1	337	340	Black Board	02		CO6,RB3
5.5	Intermediate-Code Generation: Variants of Syntax Trees,	R1	357	358	Black Board	01		CO5,RB3
5.6	Three-Address Code	R1	363	363	Black Board	01		CO5,RB4
5.7	Run-Time Environments: Stack Allocation of Space,	R1	427	433	Black Board	02		CO5,RB2
5.8	Access to Nonlocal Data on the Stack,	R1	441	443	Black Board	01		CO1,RB3

5.9	HeapManagement	R1	452	455	Black Board	01		CO6,RB2
	Review	Signature of the HOD/Coordinator						

LIST OF TEXT BOOKS AND REFERENCES

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 Chandrashekar, 2nd Edition, PHI.

REFERENCE BOOKS:

1. Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey 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. Loudon, Thomson. Course Technology.

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

	SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY			Prepared on 05/01/2026
	Question Bank			
	(Regulation: R22)			
	Department of AI&ML			
Subject Code & Title	R22CSM2212 – AUTOMATA THEORY AND COMPILER DESIGN			
Academic Year: 2025-2026	Year/Sem/Section	II/II/A&B		
Faculty Name & Designation	Ch PraveenReddy, 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	I	CO1

	(B) finite states (C) initial state (D) final state		
1C - 5	Which of the following represents Alphabet? (A) Σ (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 $ 1101 $ 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	II	CO1

	(C) 1 or more than 1 (D) infinite		
--	--------------------------------------	--	--

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

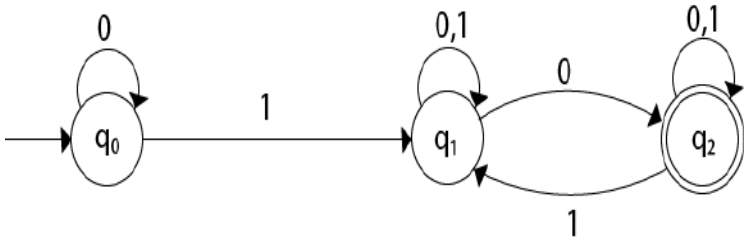
MATCH THE FOLLOWING

1M - 1	1. Terminal a. B 2. Non-terminal b. a 3. Empty string c. q0 4. Empty language d. ϵ 5. Initial state e. $\emptyset$	III	CO1
---------------	---	------------	------------

1M - 2	1. Finite Automata 2. Automata 3. NFA 4. DFA 5. String	a. No Epsilon transition b. Language Acceptor c. word d. Epsilon transition e. Abstract device	III	CO1
1M - 3	1. F 2. Σ 3. δ 4. q ₀ 5. Q	a. finite set of input symbols b. Subset of Q c. initial state d. transition function e. set of final states	I	CO1
1M - 4	1. Alphabet 2. Language 3. String 4. Empty String 5. Concatenation	a. set of strings b. set of symbols c. list of symbols d. xy e. zero	I	CO1
1M - 5	1. Type 0 2. Type 3 3. DFA 4. NFA 5. TM	a. Regular grammar b. More powerful than Automata c. Unrestricted grammar d. accepts Epsilon 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 ϵ -transitions to NFA without ϵ -transitions	IV	CO1
1D - 17	Convert the given NFA to DFA 	III	CO1

	SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY			Prepared on 05/01/2026
	Question Bank			
	(Regulation: R22)			
	Department of AI&ML			
	Subject Code & Title	R22CSM2212 – AUTOMATA THEORY AND COMPILER DESIGN		
	Academic Year: 2025-2026	Year/Sem/Section	II/II/A&B	
	Faculty Name & Designation	Ch PraveenReddy, Assistant Professor		

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 d) Kleene star	I	CO2
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) $\emptyset$ 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 { }	II	CO2

	b) Square brackets [] c) Angle brackets <> d) Pipe symbol		
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	II	CO2
---------------	---	-----------	------------

	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'.		
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 Closure 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. $L \cup L$	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

	SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY			Prepared on 05/01/2026
	Question Bank			
	(Regulation: R22)			
	Department of AI&ML			
	Subject Code & Title	R22CSM2212 – AUTOMATA THEORY AND COMPILER DESIGN		
	Academic Year: 2025-2026	Year/Sem/Section	II/II/A&B	
	Faculty Name & Designation	Ch PraveenReddy, Assistant Professor		

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	I	CO3

	b) Context-free languages c) Context-sensitive languages d) Unrestricted languages		
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: a) Empty b) Contains the input string c) Contains only the start symbol d) Contains random symbols	I	CO4
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 current state and the symbol read from the tape.	IV	CO3
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	I	CO4

	algorithm exists to determine a _____ answer.		
3F - 13	The concept of undecidability was formally established by the proof of the _____ theorem.	I	CO4
3F - 14	Undecidability 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 $S \rightarrow 0AA$ $A \rightarrow 0S/1S/0$ 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

	SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY			Prepared on 05/01/2026
	Question Bank (Regulation: R22) Department of AI&ML			
	Subject Code & Title		R22CSM2212 – AUTOMATA THEORY AND COMPILER DESIGN	
	Academic Year: 2025-2026		Year/Sem/Section	II/II/A&B
Faculty Name & Designation		Ch PraveenReddy, Assistant Professor		

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	I	CO5

	b) Bottom up parser c) Universal Parser d) All of the above		
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	II	CO5

	b) LBA c) PDA d) Finite Automaton		
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 2. Syntax Analysis 3. Semantic Analysis 4. Intermediate form	a. Parsing b. Type checking c. Three address code d. Scanning	I	CO5
4M - 2	1. Code Optimization 2. Code generation 3. Symbol table 4. Error Handling	a. Maps to target Language b. improves the Intermediate code c. done at every phase of compilation d. Contains a record of each variable	I	CO5
4M - 3	1. Lexeme Patterns 2. Token 3. Pattern 4. Lexeme	a. Sequence of Characters b. Lexical Analyzer c. Name, Attribute d. Description of the form	II	CO5
4M - 4	1. Top-down Parser 2. Bottom- up Parser 3. LALR 4. Lexical Analyzer	a. Read input of source code b. LL parser c. LR Parser d. Look ahead LR	II	CO5
4M - 5	1. Parse Tree 2. Ambiguous Grammar 3. LR parser 4. Yaac	a. Left to Right scanning of Input b. Picture of Derivation c. Parse generator 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	II	CO5

	a) Context Free Grammar b) Symbol table		
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



Subject Code & Title **R22CSM2212 – AUTOMATA THEORY AND COMPILER DESIGN**

Academic Year: 2025-2026

Year/Sem/Section

II/II/A&B

Faculty Name & Designation

Ch PraveenReddy, Assistant Professor

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? a) Synthesized b) Inherited c) Semantic d) L-Attributed	I	CO6
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	II	CO6

	c) Context-free grammars are a type of SDD d) SDDs are unrelated to context-free grammars		
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?	II	CO6

	a) An attribute inherited from the parent node to its children b) An attribute inherited from the children to the parent node 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	II	CO6

	d) Stack-based allocation		
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 2. Memory manager 3. Garbage collection 4. Mutator	a. Allocates / Deallocates space b. finds the space with in the heap c. modifies the collection of objects in the heap. d. portion of store	III	CO6
5M - 2	1.DAG 2.Triples 3.type checking 4. Record	a. op, arg1, arg2 b. syntax trees c. data structure with named field d. behavior of a program	III	CO6
5M - 3	1. SDD 2. Annotated parse tree 3. Dependency graph 4. Type equivalence	a. when two type expressions equals b. depicts the flow of information c. showing values of it's attributes d. Context free Grammar	II	CO6
5M - 4	1. Three address code 2. Runtime management 3. SDD 4. SDT	a. Memory management b. Optimization phase c. Associate Semantic actions d. Semantic actions	II	CO6
5M - 5	1. Local variable stored 2. Activation Record 3. Static Memory allocation 4. Dynamic Memory allocation	a. Run time b. Compile time c. Data Structure holds information about current function call 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

Section – A

Answer <u>All</u> 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 – B

Answer any <u>FIVE</u> questions choosing at least one from each Unit		Marks: 5Qx10M = 50M	
	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 $S \rightarrow 0AA$ $A \rightarrow 0S/1S/0$ 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	10	CO5	I,V

the Challenges involved in this process			
UNIT – V 19. Define DAG. What is the corresponding DAG for the given Expression? $a+a*(b-c) +(b-c) *d$ (OR) 20. illustrate Static versus Dynamic storage allocation of data and the Memory Hierarchy of Computer	10	CO6	I,II

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	

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