INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

COMPUTER SCIENCE AND ENGINEERING

COURSE DESCRIPTION FORM

Course Title	**FORMAL LANGUAGES AND AUTOMATA THEORY**			
Course Code	A40509			
Regulation	R15 – JNTUH			
Course Structure	Lectures	Tutorials	Practicals	Credits
	4	1	-	4
Course Coordinator	Dr. P Rajendra Prasad, Professor			
Team of Instructors	Ms. N Mamatha, Assistant Professor Ms. M Sandhya Rani, Assistant Professor Ms. K Rashmi, Assistant Professor			

I. COURSE OVERVIEW:

Formal languages and automata theory deals with the concepts of automata, formal languages, grammar, computability and decidability. The reasons to study Formal Languages and Automata Theory are Automata Theory provides a simple, elegant view of the complex machine that we call a computer. Automata Theory possesses a high degree of permanence and stability, in contrast with the ever-changing paradigms of the technology, development, and management of computer systems. Further, parts of the Automata theory have direct bearing on practice, such as Automata on circuit design, compiler design, and search algorithms; Formal Languages and Grammars on compiler design; and Complexity on cryptography and optimization problems in manufacturing, business, and management. Last, but not least, research-oriented students will make good use of the Automata theory studied in this course.

II. PREREQUISITES:

Level	Credits	Periods / Week	Prerequisites
UG	4	5	Discrete mathematics, data structures and algorithms

III. COURSE ASSESSMENT METHODS:

Marks Distribution:

Session Marks	**University End Exam Marks**	**Total Marks**
There shall be 2 mid term examinations. Each midterm examination consists of subjective type and Objective type tests. The subjective test is for 10 marks, with duration of 1 hour. Subjective test of each semester shall contain 4 questions; the student has to answer 2 questions, each carrying 5 marks. The objective type test is for 10 marks with duration of 20 minutes. It consists of 10 Multiple choice and 10 objective type questions, the	75	100

student has to answer all the questions and each carries half mark. First mid term examination shall be conducted for the first two and half units of syllabus and second midterm examination shall be conducted for the remaining portion. Five marks are earmarked for assignments. There shall be two assignments in every theory course. Marks shall considering the average of two assignments in each course.		

IV. EVALUATION SCHEME:

S. No	Component	Duration	Marks
1	I Mid Examination	90 minutes	20
2	I Assignment	-	05
3	II Mid Examination	90 minutes	20
4	II Assignment	-	05
5	External Examination	3 hours	75

V. COURSE OBJECTIVES:

I. **Understand** an overview of the theoretical foundations of computer science from the perspective of formal languages
II. **Illustrate** finite state machines to solve problems in computing
III. **Understand** the hierarchy of problems arising in the computer sciences.
IV. **Understand** Regular grammars, context free grammar.
V. **Construct** the model of Push down Automata, Turing Machines.

VI. COURSE OUTCOMES :

1. **Describe** the concept of abstract machines and their power to recognize the languages.
2. **Illustrate** DFA & NFA problems.
3. **Illustrate** FA with outputs.
4. **Describe** the languages & grammars relationship among them with the help of Chomsky hierarchy.
5. **Understand** the pre – requisites to the course compiler or advanced compiler design.
6. **Apply** finite state machines to solve problems in computing.
7. **Understand** Regular grammars, languages and ability to construct grammars for specific tasks.
8. **Understand** context-free grammars, languages, derivations and parse trees.
9. **Apply** the knowledge to construct context-free grammars for specific tasks.
10. **Design** the model of Push down Automata.
11. **Understand** CFL and PDA conversions.
12. **Design** the model of Turing Machine.
13. **Understand** the principles of computability and complexity including decision problems, halting problems and basic complexity classes such as P and NP.

VII. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes		Level	Proficiency assessed by
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	H	Lectures, Assignments
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.	S	Lectures, Assignments, Exercises
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.	H	Lectures, Assignments, Exercises
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.	N	--
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.	H	Assignments, Exercises
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.	N	--
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.	N	--
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	N	--
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	N	--
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.	N	--
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.	N	--
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	S	Assignments, Exercises

N= None **S= Supportive** **H = Highly Related**

VIII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes		Level	Proficiency assessed by
PSO1	**Professional Skills:** The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer-based systems of varying complexity.	H	Lectures, Assignments
PSO2	**Problem-solving Skills:** The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.	S	Projects
PSO3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.	S	Guest Lectures

N - None S - Supportive H - Highly Related

IX. SYLLABUS:

UNIT I:

Fundamentals:

Strings, Alphabets, Language, Operations, Finite state machine, definitions, finite automaton model, acceptance of strings, and languages, deterministic finite automation and non deterministic finite automaton, transition diagrams and language recognizers.

Finite automata:

NFA with ε transitions – Significance, acceptance of languages. Conversions and Equivalence : Equivalence between NFA with and without ε – transitions. NFA to DFA conversion, minimization of FSM, equivalence between two FSM's, Finite Automata with output – Moore and Melay machines.

UNIT II:

Regular Languages:

Regular sets, regular expressions, identify rules, Constructing finite Automata for a given regular expressions, Conversion of Finite Automata to Regular expressions. Pumping lemma of regular sets, closure properties of regular sets(proofs not required)

Grammar Formalism:

Regular grammars – right linear and left linear grammars, equivalence between regular linear grammar and FA, inter conversion, Context free grammar , derivation trees, sentential forms. Right most and left most derivation of strings.

UNIT III:

Context Free Grammars :

Ambiguity in context free grammars. Minimization of Context Free Grammars. Chomsky normal form, Greiback normal form, Pumping Lemma for Context Free Languages. Enumeration of properties of CFL (proofs omitted).

Push Down Automata:

Push down automata, definition , model, acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence. Equivalence of CFL and PDA, interconversion.(Proofs not required). Introduction to DCFL and DPDA.

UNIT IV:
Turing Machine :
Turing Machine, definition, model , design of TM, Computable functions, recursively enumerable languages. Church's hypothesis , counter machine , types of Turing machines(proofs not required).linear bounded automata and context sensitive language.

UNIT V:
Computability Theory :
Chomsky hierarchy of languages, decidability of problems, Universal Turing Machine, undecidability of posts. Correspondence problem, Turing reducibility, Definition of P and NP problems, NP complete and NP hard problems.

TEXT BOOKS:
1. "Introduction to Automata Theory Languages and Computation".Hopcroft H.E. and Ullman J.D. Pearson Education.
2. Introduction to Theory of Computation –Sipser 2nd edition Thomson

REFERENCES
1. Theory of Computation – Vivek Kulkarni - OXFORD
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. Introduction to languages and the Theory of Computation ,John C Martin, TMH
4. Theory of Computer Science – Automata languages and computation – Mishra and Chandrashekaran, 2nd edition, PHI
5. "Elements of Theory of Computation", Lewis H.P. & Papadimition C.H. Pearson/PHI

X. COURSE PLAN:

The course plan is meant as a guideline. There may probably be changes.

Lecture No.	Topics to be covered	Course Learning Outcomes	Reference
1 – 2	**Fundamentals :** Strings, Alphabet, Language, Operations	**Understand** Fundaments	T1: 1.1-1.2 R1: 1.2 – 1.3
3	Finite state machine, definitions, finite automaton model, acceptance of strings, and languages	**Understand** Finite state machine	T1:2.1 -2.2 R1: 2.2 – 2.3
4 – 5	Deterministic finite automaton.	**Illustrate** Deterministic finite automaton	T1:2.2 R1: 2.4
6 -7	Non deterministic finite automaton	**Illustrate** Non deterministic finite automaton	T1:2.3 R1: 2.5
8	Transition diagrams and Language recognizers	**Illustrate** Transition diagrams	T1:2.2 R1: 2.2 -2.3
9 – 10	**Finite Automata :** Non deterministic finite automaton with ε transitions - Significance, acceptance of languages.	**Illustrate** Non deterministic finite automaton with ε transitions	T1:2.4 R1: 2.7
11	Conversions and Equivalence : Equivalence between NFA with and without ε transitions	**Illustrate** NFA with ε to NFA conversions	T1:2.4 R1: 2.8
12 – 13	NFA to DFA conversion	**Illustrate** NFA to DFA conversion	T1:2.3 R1: 2.6.1
14 – 15	Minimization of Finite state machine	**Understand** Minimization of Finite state machine	T1:3.4 R1: 2.6.2 , 2.13
16	Equivalence between two Finite state machine's	**Illustrate** Equivalence between two Finite state machine's	T1:3.4 R1: 2.12
17 – 18	Finite Automata with output- Moore and Melay machines	**Illustrate** Finite Automata with output machines.	T1:2.7 R1: 2.10
19	**Regular Languages :** Regular sets, regular expressions, identity rules,	**Understand** Regular languages	T1:2.5 R1: 3.2 -3.3
20 – 21	Constructing finite Automata for a given regular	**Illustrate** Constructing Finite	T1:2.5

Lecture No.	Topics to be covered	Course Learning Outcomes	Reference
	expressions	Automata for given RE	R1: 3.4
22	Conversion of Finite Automata to Regular expressions.	**Illustrate** Conversion of finite automaton to Regular expressions	T1:2.5 R1: 3.4
23	Pumping lemma of regular sets	**Understand** Pumping lemma	T1:3.1 R1: 3.6
24	Closure properties of regular sets (proofs not required).	**Understand** Closure properties	T1:3.2 R1:3.5.2
25	**Grammar Formalism :** Regular grammars-right linear and left linear grammars	**Understand** Regular Grammars	T1:9.1 R1: 5.11.4
26 – 27	Equivalence between regular linear grammar and Finite Automata, inter conversion	**Illustrate** Equivalence of regular linear grammar & Finite Automata	T1:9.1 R1: 5.12 – 5.13
28	Context free grammar, derivation trees	**Understand** Context free grammar	T1:4.2 R1: 5.6 , 5.7
29	Sentential forms, Right most and leftmost derivation of strings	**Illustrate** Derivation of strings	T1:4.3 R1: 5.5
30	**Context Free Grammars:** Ambiguity in context free grammars.	**Understand** Ambiguity in context free grammars	T1:4.3 R1: 5.8
31	Minimization of Context Free Grammars.	**Understand** Minimization	T1:4.4 R1: 5.9
32	Chomsky normal form	**Understand** Chomsky normal form	T1:4.5 R1: 5.10.1
33	Greiback normal form	**Understand** Greiback normal form	T1:4.6 R1: 5.10.2
34	Pumping Lemma for Context Free Languages.	**Understand** Pumping Lemma	T1:6.1 R1: 5.14
35	Enumeration of properties of CFL (proofs omitted).	**Understand**Context Free Language properties	T1:6.2 R1: 6.8
36 -40	**Push Down Automata** Push down automata, definition, model, acceptance of CFL. Acceptance by final state and acceptance by empty state and its equivalence.	**Understand** Push down automata	T1:5.2 R1: 6.2 – 6.3 T1:5.2 R1: 6.5
40 – 41	Equivalence of CFL and PDA, inter conversion. (Proofs not required).	**Understand** Equivalences	T1:5.3 R1: 6.7
42	Introduction to DCFL and DPDA	**Illustrate** DCFL,DPDA	T1:10.1 – 10.2 R1: 6.6
43- 46	**Turing Machine :** Turing Machine, definition, Model, design of TM	**Understand** Turing machines	T1:7.2 R1: 4.2 – 4.3 T1:7.2 R1: 4.4 – 4.5
47 -49	Computable functions	**Illustrate** Computable functions	T1:7.3 R1: 4.6
50	Recursively enumerable languages.	**Understand** Recursively enumerable languages	T1:7.3 , 8.2 R1: 4.15
51	Church's hypothesis, counter machine	**Understand** Cchurch's hypothesis	T1:7.6 R1: 4.17
52	Types of Turing machines (proofs not required)	**Illustrate**Types of Turing machines	T1:7.5 R1: 4.8 – 4.12
53 – 54	**Computability Theory :** Chomsky hierarchy of languages, Linear bounded automata and context sensitive languages	**Understand** Chomsky hierarchy	T1:9.1 – 9.4 R1: 5.11 T1:9.3 R1: 4.20, 5.11.2
55 – 56	LR(0) grammar, decidability of, problems	**Illustrate** LR(0)	T1:10.6 R1: 7.5.1
57	Universal Turing Machine	**Understand** Universal Turing Machine	T1:8.3 R1: 4.9
58 – 60	Undecidability of posts. Correspondence problem,Turing reducibility, Definition of P and NP problems,NP complete and NP hard problems.	**Understand** PCP, P, NP, NP complete problems	T1:8.5| R1: 4.18 T1:13.1 T1:13.2

XI. MAPPING COURSE OBJECTIVES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES:

Course Objectives	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
I	S	H	S										H		
II		H										S	H		
III	H				H							S		S	S
IV	S	H											H	S	
V		S	H											S	S

S - Supportive **H - Highly Related**

XII. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES:

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	H	S												S	
2			H		S								H		
3			H		S								H		
4	H											S		S	S
5												S			S
6		H	S											S	
7	S	H											H	S	
8	S	H											H	S	
9		H	S											S	S
10		H	S										H	S	S
11	S	H											H	S	
12		S	H											S	
13	S				H									S	S

S - Supportive **H - Highly Related**

Prepared by: Dr. P Rajendra Prasad, Professor

HOD, CSE