PVP 23

PRASAD V. POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY

(Autonomous)

Kanuru, Vijayawada-520007

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)

III B Tech – I Semester

Automata Theory & Compiler Design

Course Code	23AM4501E	**Year**	III	**Semester**	I
Course Category	PEC	**Branch**	CSE (AI&ML)	**Course Type**	Theory
Credits	3	**L-T-P**	3-0-0	**Prerequisites**	Discrete Mathematics
Continuous Internal Evaluation	30	**Semester End Evaluation**	70	**Total Marks**	100

Course Outcomes		
Upon Successful completion of course, the student will be able to		
CO1	Describe automata, formal languages, Turing machines, and compiler phases to understand the foundations of computation and compiler design	**L2**
CO2	Apply regular expressions and context-free grammars to define and analyze programming language construct	**L3**
CO3	Apply lexical and syntax analysis techniques to generate intermediate code and manage runtime environments in compiler design.	**L3**
CO4	Analyze the behavior of Turing machines, syntax-directed translation, and runtime environment management to understand language processing and program execution.	**L4**

Contribution of course outcomes towards achievement of program outcomes & Strength of correlations (3: Substantial,2: Moderate,1: Slight)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	P010	PO11	PSO1	PSO2
CO1	2												
CO2	3												
CO3	3												
CO4		3									2		

PRASAD V. POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY

(Autonomous)
Kanuru, Vijayawada-520007

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)

III B Tech – I Semester

Syllabus		
Unit No	**Contents**	**Mapped CO**
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 €-transitions to NFA without €-transitions. Conversion of NFA to DFA.	CO1
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.	CO1, CO2, CO4
III	**Push Down Automata:** Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's 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.	CO1, CO2, CO4
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	CO1, CO3, CO4
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	CO1, CO3, CO4

Learning Resources
1. Introduction to Automata Theory, Languages, and Computation by John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, 3rd Edition, 2006, Pearson Education.
2. Compilers: Principles, Techniques and Tools by Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, 2nd Edition, 2006, Pearson.
3. Theory of Computer Science – Automata, Languages and Computation by K.L.P. Mishra and N. Chandrasekaran, 2nd Edition, 2001, PHI Learning.

PRASAD V. POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY

(Autonomous)
Kanuru, Vijayawada-520007

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)

III B Tech – I Semester

References

1. Introduction to Formal Languages, Automata Theory and Computation by Kamala Krithivasan and Rama R, 1st Edition, 2009, Pearson Education.
2. Introduction to Languages and the Theory of Computation by John C. Martin, 4th Edition, 2010, McGraw-Hill (TMH).
3. Lex & Yacc by John R. Levine, Tony Mason, and Doug Brown, 2nd Edition, 1992, O'Reilly Media.
4. Compiler Construction by Kenneth C. Louden, 1st Edition, 1997, Course Technology, Thomson Learning.

E-Recourses and other Digital Material

1. https://nptel.ac.in/courses/106/106106049
2. https://www.youtube.com/playlist?list=PL85CF9F4A047C7BF7