23CS31T3 -AUTOMATA THEORY AND COMPILER DESIGN

Course Category:	Professional core	**Credits:**	3
Course Type:	Theory	**Lecture-Tutorial-Practical:**	3-0-0
Prerequisite:	• Knowledge in Automata Theory And Compiler Design	**Sessional Evaluation:** **Univ. Exam Evaluation:** **Total Marks:**	30 70 100
Course Objectives:	**Students undergoing this course are expected:**		
	• Able to understand the concept of abstract machines, construct FA, Regular Expressions for the regular languages and equivalent FSMs. • Able to construct pushdown automata equivalent to Context free Grammars, construct Turing Machines and understand undecidability. • Emphasize the concepts learnt in phases of compiler, lexical analyser and Top-down parser. • Able to understand the concepts of Bottom-up parser, Intermediate Code Generation. • Able to understand the concepts of Code optimizer and Code Generation.		

Course Outcomes:	**Upon successful completion of the course, the students will be able to:**	
	CO1	Demonstrate knowledge on Automata Theory, Regular Expression and Analyze and Design of finite automata, and prove equivalence of various finite automata.
	CO2	Demonstrate knowledge on context free grammar, Analyze and design of PDA and TM.
	CO3	Understand the basic concept of compiler design, and its different phases which will be helpful to construct new tools like LEX, YACC, etc.
	CO4	Ability to implement semantic rules into a parser that performs attribution while parsing and apply error detection and correction methods.
	CO5	Apply the code optimization techniques to improve the space and time complexity of programs while programming and Ability to design a compiler.
Course Content:	**Unit-I** **Introduction to Automata and Regular Expressions** Introduction, Alphabets, Strings and Languages, Chomsky Hierarchy, Automata and Grammars, Regular Grammar and Language, Finite Automata, Deterministic finite Automata (DFA), Nondeterministic finite Automata (NFA), Equivalence of NFA and DFA, Minimization of Finite Automata, Regular Expressions, Converting Regular Grammar and Expression into Finite Automata, Pumping lemma for regular sets, Closure properties of regular sets (Without proof).	

	UNIT-II **Context Free Grammars and Pushdown Automata** Context Free Language, Context Free Grammar, Derivation and Parse tree, Ambiguity, Simplification of CFG's, Chomsky Normal Form, Greibach Normal Form, Push Down Automat (PDA), Design of PDA, Equivalence of PDA and CFL/CFG **UNIT-III** **Turing Machines and Introduction to Compilers** Turing Machine, TM Model, Language acceptance, Design of Turing Machine, Compilers, Phases of Compiler, The role of Lexical Analyzer, Input Buffering. **UNIT-IV** **Parsers and Intermediate Code Generation** Parser, Top-Down parsers: Recursive Descent Parsers, Predictive Parsers Bottom–up Parsers: Shift-Reduce Parsing, LR parsers, Intermediate Code Generation: Three address codes. **UNIT-V** **Code Optimization and Code Generation** Code Optimization: Peephole optimization, Basic blocks and flow graphs, DAG, Principles of Source Code Optimization, Code Generation: Issues in Design of Code Generation, Simple Code Generator.
Text Books & References Books:	**TEXTBOOKS:** 1. Introduction to Automata theory languages and Computation, Hopcroft H.E. and Ullman Jeffrey.D, 3/e, 2006, Pearson Education, New Delhi, India. 2. Mishra K L P and Chandrasekaran N, ―Theory of Computer Science - Automata, Languages and Computation‖, 2/e, 2007, PHI, New Delhi, India. 3. Compilers: Principles, Techniques, and Tools, Updated 2e July 2023 Alfred V. Aho , Monica S. Lam, Ravi Sethi , Jeffrey D. Ullman , Sorav Bansal **REFERENCE BOOKS:** 1. Introduction to Languages and Theory of Computation, John C Martin, 1/e, 2009, Tata McGraw Hill Education, Hyderabad, India. 2. Introduction to Theory of Computation, Sipser, 2/e, 2005, Thomson, Australia. 3. Compiler Construction: Principles And Practice, Kenneth C. Louden, Thomson/ Delmar Cengage Learning, 2006. 4. Lex &yacc, Doug Brown, John Levine and Tony Mason, 2 nd Edition, O'reilly Media 5. Engineering a compiler, Keith Cooper and Linda Torczon, 2 nd Edition, Morgan Kaufmann, 2011.