

Code	Category	L	T	P	C	I.M	E.M.	Exam
B20AM2203	PC	3	--	--	3	30	70	3 Hrs.
AUTOMATA THEORY AND COMPILER DESIGN								
(For AIML)								
Course Objectives:								
1.	To learn design of Automata's as Acceptors.							
2.	To learn fundamentals of Regular and Context Free Grammars and Languages							
3.	To study the various phases in the design of a compiler							
4.	To study the design of top-down, bottom-up parsers							
5.	To learn to develop algorithms for code optimization and machine code for a target machine.							
Course Outcomes: By the end of the course students will be able to								
S.No	Outcome							Knowledge Level
1	Construction of DFA, NFA, NFA with ϵ -Transitions and output generating machines, regular expressions							K3
2	Construct Regular grammars and CFG, derive strings.							K3
3	Demonstrate the different phases of compilation process.							K3
4	Construct LR parsers for CFG and summarize Syntax directed translation schemes, intermediated code							K3
5	Demonstrate different code optimization techniques code generation techniques							K3
SYLLABUS								
UNIT-I (10 Hrs)	Finite Automata and Regular expressions							
	Introduction to DFA and NFA, Acceptance of a Strings by DFA, Acceptance of a Strings by NFA, Design of DFAs, Design of NFAs, Conversion of NFA to DFA (Theorem and problems) Introduction to NFA with ϵ -Transitions, Conversion of NFA with ϵ -Transitions to DFA, Minimization of DFA algorithm and state minimization of DFA problems, Design of Mealy and Moore machines, Applications and Limitations of Finite Automata. Introduction to Regular Expressions, Regular Sets, Identity Rules, Conversion of Regular Expression to NFA with ϵ -Transitions, Applications of Regular Expressions.							

UNIT-II (10 Hrs)	Grammars and Formal languages Chomsky Hierarchy, Regular Grammar, Left-Linear Grammar, Right-Linear Grammar, Conversion of Finite Automata to Regular Grammars and Regular Grammars to Finite Automata, Context Free Grammar, Construction of CFGs for Languages, Determining language of the grammar. Leftmost and Rightmost Derivations, Parse Trees, Ambiguous Grammars, Elimination of left recursion and left factoring, Pumping Lemma of Regular Languages (Statement and problems), Applications of pumping lemma, Closure Properties of regular languages, Pumping Lemma for CFL (Statement and Problems),
-----------------------------	--

	Applications of pumping lemma for CFL, Closure Properties of CFL, Applications of Context Free Grammars.
UNIT-III (10 Hrs)	<p>Lexical analysis and Syntax analysis: Introduction to Language Processing, Structure of a Compiler, Compiler-Construction Tools.</p> <p>Lexical Analysis: The Role of Lexical Analysis, Specification of Tokens, Recognitions of Tokens, The Lexical Analyzer Generator LEX.</p> <p>Syntax Analysis: The Role of a Parser, Top down and Bottom-up Parsing, Shift Reduce Parser. Predictive Parser.</p>
UNIT-IV (10 Hrs)	<p>Powerful parsers and Intermediate code: Simple LR Parser, Canonical LR parser and LALR Parser, The Parser Generator YACC. Syntax-Directed Translation: Syntax Directed Definitions Inherited and Synthesized Attributes, Evaluating an SDD at Nodes of Parse Tree.</p> <p>Intermediate Code Generation: Three-Address Code, Quadruples, Triples, DAG, Type Checking (Rules for Type Checking and Type Conversion).</p>
UNIT-V (10 Hrs)	<p>Code Optimization and Code Generation: Code Optimization: The Principal Sources of Optimization, Introduction to Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Dead code elimination, constant propagation, Global Data-Flow Analysis.</p> <p>Code Generation: A Simple Code Generator, Code Generation from DAG, Peephole Optimization, Symbol Table and error detection in different phases. Storage Organization, Stack Allocation of Space.</p>
Text Books:	
1.	Introduction to Automata Theory, Languages and Computation, J. E. Hopcroft, R. Motwani and J. D. Ullman, 3 rd Edition, Pearson, 2008
2.	Theory of Computer Science-Automata, Languages and Computation, K. L. P. Mishra and N. Chandrasekharan, 3 rd Edition, PHI, 2007
3.	Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson, Pearson Education India; 2nd edition, 2013
4.	Compiler Construction- Principles and Practice, Kenneth C Loudon, Cengage Learning, 1 st Jan, 2009
Reference Books:	
1.	Elements of Theory of Computation, Lewis H.P. & Papadimition C.H., Pearson /PHI, 2 nd Edition, 2006
2.	Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press, 1998
Web Resources:	
1.	https://nptel.ac.in/courses/106/104/106104028/
2.	https://nptel.ac.in/courses/106/104/106104123/



SRKR
ENGINEERING COLLEGE
AUTONOMOUS