FORMAL LANGUAGES AND AUTOMATA

Course code	PCC-CSE-305G				
Category	Professional Core Course				
Course title	Formal Languages & Automata				
Scheme and Credits	L	T	P	Credits	Semester 5
	3	0		3	
Class work	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Course Objectives:

- To understand basic concepts of formal languages and automata theory.
- To study the types of Automata i.e. NFA, DFA, NFA with ϵ-transition and their interconversion methods and importance.
- To Study formal languages of different kinds, such as regular and context-free languages. Understand the concept of grammar and its types. Removal of ambiguity and reduced form and Normal forms of grammar.
- To develop the concepts and design of higher-level automata to accept the language not accepted by finite automata such as PDA &Turing machine.
- To study the various properties of turing machine and their designing.

Note: Examiner will set nine questions in total. Question one will be compulsory. Question one will have 6 parts of 2.5 marks each from all units and remaining eight questions of 15 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit 1:

Finite Automata: Introduction: Set, Power Set, Super Set, Alphabet, languages and grammars, productions and derivation, Deterministic finite automata (DFA), Non-Deterministic finite automata (NDFA), Equivalence of DFA and NDFA, Conversion of NFA to DFA , minimization of finite automata, Finite automata with ϵ- moves, Acceptability of a string by a finite Automata.

Introduction to Machines: Properties and limitations of Finite Automata, Mealy and Moore Machines, Equivalence of Mealy and Moore machines.

Unit 2:

Regular Expression: State and prove Arden's Method, Regular Expressions, Recursive definition of regular expression, Regular expression conversion to Finite Automata and vice versa.

Properties of regular languages: Regular language, pumping lemma for regular sets/languages, Application of regular languages.

Unit 3:

Grammars: Chomsky hierarchy of languages, Relation between different types of grammars, Context-free grammar, Derivation tree / Parse tree, Ambiguity in regular grammar and their removal, Reduced Forms: Removal of useless symbols, null and unit productions, Normal Form: Chomsky Normal form(CNF) and Greibach Normal Form(GNF),

Push Down Automata: Introduction to PDA, Deterministic and Non-Deterministic PDA, Design of PDA: Transition table, Transition diagram and acceptability of strings by designed PDA, Pushdown automata (PDA) and equivalence with CFG.

Unit 4:

Turing machines: The basic model for Turing machines (TM), Deterministic and Non-Deterministic Turing machines and their equivalence, Design of Turing Machines: Transition table, Transition diagram and acceptability of strings by designed turing machine. Variants of Turing machines, Halting problem of Turing machine, PCP Problem of Turing Machine, Linear Bounded Automata, TMs as enumerators.

Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.

Suggested books:

1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage Learning.

Suggested reference books

1. K. L. P Mishra, N. Chandrashekaran (2003), Theory of Computer Science-Automata Languages and Computation, 2nd edition, Prentice Hall of India, India.
2. Raymond Greenlaw, H. James Hoover, Fundamentals of the Theory of Computation, Principles and Practice, Morgan Kaufmann, 1998.
3. John C. Martin: Introduction to Languages and Automata Theory, 3rd edition, Tata Mcgraw-Hill, 2007

Course Outcomes:

- To use basic concepts of formal languages of finite automata techniques.
- To Design Finite Automata’s for different Regular Expressions and Languages.
- To Construct context free grammar for various languages.
- To solve various problems of applying normal form techniques, push down automata and Turing Machines.