CS2204 FORMAL LANGUAGES AND AUTOMATA THEORY

Course objectives:

- To introduce the concepts in automata theory and theory of computation to design grammars and recognizers for different formal languages.
- To employ finite state machines to solve problems in computing.
- To introduce finite state machines, context free grammars and Turing Machines and their properties as the basis for the formal expressivity of computer languages for solving linguistic decision problems.
- To understand the concepts of tractability and decidability, the concepts of NP-completeness and NP-hard problem and also the challenges for Theoretical Computer Science and its contribution to other sciences.

Course outcomes:

- Ability to think analytically and intuitively for problem-solving situations in related areas of theory in computer science
- Ability to describe the language accepted by an automata or generated by a regular expression or a context-free grammar;
- Ability to Understand the functioning of Finite-State Machines, Deterministic Finite-State Automata, Nondeterministic Finite-State Automata and Pushdown Automata and Turing Machines.

SYLLABUS

Introduction to Grammars and Languages: Definitions of alphabet, strings, language, grammar, types of grammar, types of machines, generation of languages from grammar, construction of grammar from the given description of languages, Chomsky Hierarchy of languages.

Finite State Machine (FSM): Definition of finite state machine, Representation of FSMs. Classification of FSM's and their construction, Conversion from NFA to DFA, Elimination of ε – transitions from NFA, Equivalence of two FSM's, optimization of finite state machine (Equivalence theorem method and Table filling method), Finite state machine with output: Moore and Mealy machines. Applications of FSM.

Regular Expression and Languages: Regular Expression, Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Properties of Regular Languages: Pumping Lemma for regular Languages, Application of Pumping Lemma, Closure properties of Regular Languages, Decision properties of Regular Languages, Equivalence and Minimization of Automata.

Context Free Grammars and Languages: Context Free Grammars, Parse Trees, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages, Normal Forms, Pumping Lemma for CFL, Closure properties of CFL, Decision properties for CFL.

Push down Automata: Definition of push down automata, The Languages of a PDA, push down automata, Equivalence of PDA's and CFG's, push down automata to context free grammar, context free grammar to push down automata, Deterministic Pushdown Automata.

Turing Machines: The Definition of Turing Machine, Turing Machine Model, Representation of Turing Machines, Language Acceptability by Turing Machines, Design of Turing Machines, Description of Turing Machines, Techniques for TM Construction, Variants of Turing Machines, Turing Machines and Type 0 Grammars.

Undecidability: A Language That Is Not Recursively Enumerable, An Undecidable Problem That Is RE, Undecidable Problems About Turing Machines, Decidable & Undecidable Problems, Post Correspondence Problem.

Text books:

1. Introduction to automata theory, languages and computation, John.E.H.P croft/ Rajeev Motwani & JD Ullman—pearson education- III edition
2. Theory of computation, K.L.P.Mishra and N.Chandrasekhar, PHI

Reference Books:

1. Theory of computation, formal languages and automata theory, G P Saradhi Varma, B.Thirupathi Rao –Sci Tech publications.