FORMAL LANGUAGES AND AUTOMATA THEORY

Course Code: IT 401 **Credit Units: 03**
Total Hours: 45

Course Objective:
Students will able to understand the formal mathematical models of computation along with their relationships with formal languages. In particular, they will learn regular languages and context free languages which are crucial to understand how compilers and programming languages are built. Also students will learn that not all problems are solvable by computers, and some problems do not admit efficient algorithms. Throughout this course, students will strengthen their rigorous mathematical reasoning skills.

Course Contents:

Module I: Finite Automata and Regular Languages: (11 Hours)
Introduction- Basic Mathematical Notation and techniques- Finite State systems – Basic Definitions – Finite Automaton – DFA & NDFA – Finite Automaton with €- moves – Regular Languages- Regular Expression – Equivalence of NFA and DFA – Equivalence of NDFA"s with and without €-moves – Equivalence of finite Automaton and regular expressions –Minimization of DFA- – Pumping Lemma for Regular sets – Problems based on Pumping Lemma.

Module II: Grammars: (11 Hours)
Grammar Introduction– Types of Grammar – Context Free Grammars and Languages– Derivations and Languages – Ambiguity- Relationship between derivation and derivation trees – Simplification of CFG – Elimination of Useless symbols – Unit productions – Null productions – Greibach Normal form – Chomsky normal form – Problems related to CNF and GNF. Chomsky hierarchy of languages.

Module III: Pushdown Automata (7 Hours)
Pushdown Automata- Definitions – Moves – Instantaneous descriptions – Deterministic pushdown automata – Equivalence of Pushdown automata and CFL – pumping lemma for CFL – problems based on pumping Lemma. Linear Bounded Automata (LBA).

Module IV: Turing Machines: (7 Hours)
The Turing Machine Model, Language acceptability of Turing Machine, Design of TM, Variation of TM, Universal TM, Church's Machine. Context sensitive language and linear bounded automata (LBA), Chomsky hierarchy, Decidability, Post's correspondence problem (PCP), undecidability of PCP

Module V: Introduction to compiler (9 Hours)
Compilers Analysis of source Program, The Phases of a compiler, The tasks of a compiler, Analysis of the Source Program, Phases and Passes in compilers, Cousins of the compiler, The Grouping of phases, Compiler - construction tools. Lexical Analysis - The role of Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, A Language for Specifying Lexical Analyzer, Review of Regular Expressions, Finite State Machines, Finite Automata based, Pattern Matching. Specification and recognition of tokens, a language for specifying lexical analyser

Course Outcomes:

At the end of this course, students will be able to do the following:

- Students will demonstrate knowledge of basic mathematical models of computation and describe how they relate to formal languages.
- Students will understand that there are limitations on what computers can do, and learn examples of unsolvable problems.
- Students will learn that certain problems do not admit efficient algorithms, and identify such problems.
- Students will learn basic concepts of compiler.

Examination Scheme:

Components	**A**	**CT**	**S/V/Q/HA**	**ESE**
Weightage (%)	5	15	10	70

A: Attendance, CT: Class Test, S/V/Q/HA: Seminar/Viva/Quiz/ Home Assignment, ESE: End Semester Examination.

Text & References:

Text:

- Hopcroft and Ullman, "Introduction to Automata Theory, languages and computation", Addision Wesley.
- "An introduction to formal languages and Automata (2nd ed)" by Peter Linz, D. C. Health and Company.
- Alfread V. Aho, Ravi Sethi & J.D. Ullman, "Compiler Design", Addison Wesley

References:

- "Introduction to theory of computation (2nd Ed)" by Michael sipser.
- Mishra & Chandrashekharan, "Theory of Computer Sciences", PHI.
- Zavi Kohavi, "Switching and finite Automata Theory "
- Ullman, Principles of Compiler Design, Narosa publications.