CSPC3001 THEORY OF COMPUTATION (3-0-0)

Course Objectives

The course aims to:

1. Introduce the mathematical foundations of computation through formal languages, grammars, automata, and Turing machines.
2. Develop students' ability to design abstract models of computation and analyze their capabilities and limitations.
3. Understand the concepts of decidability, reducibility, and computational complexity.
4. Foster mathematical rigor in designing and proving properties of computational models.
5. Explore the Chomsky hierarchy and the relationships between various computational classes.

Module-I: Fundamentals and Finite Automata (08 Hours)

Basic definitions: alphabets, strings, languages, and operations on languages. Introduction to finite automata: deterministic finite automata (DFA), nondeterministic finite automata (NFA), and transition diagrams. Language recognition, equivalence between DFA and NFA, and conversion of NFA to DFA. Epsilon (ε) transitions in NFA and their elimination. Minimization of finite state machines. Moore and Mealy machines, and conversion between them.

Module-II: Regular Languages and Expressions (08 Hours)

Regular expressions and identity rules. Construction of finite automata from regular expressions and vice versa. Regular grammars: right linear and left linear forms. Conversion between different regular representations (grammar, FA, and regex). Pumping lemma for regular languages and its application. Closure properties of regular languages.

Module III: Context-Free Grammars and Pushdown Automata (08 Hours)

Definition and examples of context-free grammars (CFGs), derivation trees, leftmost and rightmost derivations, ambiguity in CFGs. Simplification of CFGs, Chomsky and Greibach normal forms. Pumping lemma for context-free languages and its applications. Pushdown automata (PDA): definition, model, language acceptance by final state and empty stack. Equivalence between CFG and PDA. Introduction to deterministic and nondeterministic PDA, and comparison.

Module IV: Turing Machines and Computability (8 Hours)

Turing Machine (TM): formal definition, configurations, and design of TMs. Variants of TMs and their equivalence. Linear bounded automata (LBA) and context-sensitive languages. Computable functions, recursively enumerable and recursive languages. Church–Turing thesis. Decidability and undecidability, examples of undecidable problems. Reductions and mapping reducibility.

Module V: Complexity Theory and Chomsky Hierarchy (08 Hours)

Chomsky hierarchy: classification of languages and corresponding machines. Introduction to complexity theory: efficiency of computation, time and space complexity. Complexity classes P and NP, NP-completeness, and the P vs NP problem. Introduction to polynomial-time reductions and basic NP-complete problems. Language families and their relationships.

Course Outcomes

Upon successful completion of the course, students will be able to:

CO1. Explain the fundamental concepts of automata theory and formal languages.
CO2. Design finite automata, pushdown automata, and Turing machines for formal language recognition.
CO3. Apply regular expressions, context-free grammars, and normal forms in language design.
CO4. Analyze decidability and computability of languages and problems.
CO5. Classify languages and problems using the Chomsky hierarchy and complexity classes.
CO6. Prove formal properties of computational models using pumping lemmas and reduction techniques.

Text books

1. Introduction to Automata Theory, Languages, and Computation – Hopcroft H.E. and Ullman J.D., Pearson Education.
2. An Introduction to Formal Languages and Automata – Peter Linz, Narosa Publishing.
3. Introduction to the Theory of Computation – Michael Sipser, 2nd Edition, Thomson.

Reference Books

1. Introduction to Computer Theory – Daniel I.A. Cohen, John Wiley.
2. Introduction to Languages and the Theory of Computation – John C. Martin, TMH.
3. Elements of the Theory of Computation – Lewis H.P. & Papadimitriou C.H., Pearson.
4. Theory of Computer Science (Automata, Languages and Computation) – Mishra and Chandrashekaran, PHI.