Theory of Computation

Course Title: Theory of Computation   **Full Marks:** 60 + 20 + 20
Course No: CSC262   **Pass Marks:** 24 + 8 + 8
Nature of the Course: Theory + Lab   **Credit Hrs:** 3
Semester: IV

Course Description: This course presents a study of Finite State Machines and their languages. It covers the details of finite state automata, regular expressions, context free grammars. More, the course includes design of the Push-down automata and Turing Machines. The course also includes basics of undecidabilty and intractability.

Course Objectives: The main objective of the course is to introduce concepts of the models of computation and formal language approach to computation. The general objectives of this course are to, introduce concepts in automata theory and theory of computation, design different finite state machines and grammars and recognizers for different formal languages, identify different formal language classes and their relationships, determine the decidability and intractability of computational problems.

Course Contents:
Unit I: Basic Foundations (3 Hrs.)
    1.1. Review of Set Theory, Logic, Functions, Proofs
    1.2. Automata, Computability and Complexity: Complexity Theory, Computability Theory, Automata Theory
    1.3. Basic concepts of Automata Theory: Alphabets, Power of Alphabet, Kleen Closure Alphabet, Positive Closure of Alphabet, Strings, Empty String, Substring of a string, Concatenation of strings, Languages, Empty Language

Unit II: Introduction to Finite Automata (8 Hrs.)
    2.1 Introduction to Finite Automata, Introduction of Finite State Machine
    2.2 Deterministic Finite Automata (DFA), Notations for DFA, Language of DFA, Extended Transition Function of DFA Non-Deterministic Finite Automaton (NFA), Notations for NFA, Language of NFA, Extended Transition
    2.3 Equivalence of DFA and NFA, Subset-Construction
    2.4 Method for reduction of NFA to DFA, Theorems for equivalence of Language accepted by DFA and NFA
    2.5 Finite Automaton with Epsilon Transition (ε - NFA), Notations for ε - NFA, Epsilon Closure of a State, Extended Transition Function of ε – NFA, Removing Epsilon Transition using the concept of Epsilon Closure, Equivalence of NFA and ε –NFA, Equivalence of DFA and ε – NFA
    2.6 Finite State Machines with output: Moore machine and Mealy Machines

Unit III: Regular Expressions (6 Hrs.)
    3.1 Regular Expressions, Regular Operators, Regular Languages and their applications, Algebraic Rules for Regular Expressions

3.2 Equivalence of Regular Expression and Finite Automata, Reduction of Regular Expression to ε – NFA, Conversion of DFA to Regular Expression

3.3 Properties of Regular Languages, Pumping Lemma, Application of Pumping Lemma, Closure Properties of Regular Languages over (Union, Intersection, Complement) Minimization of Finite State Machines: Table Filling Algorithm

Unit IV: Context Free Grammar (9 Hrs.)

4.1 Introduction to Context Free Grammar (CFG), Components of CFG, Use of CFG, Context Free Language (CFL)

4.2 Types of derivations: Bottomup and Topdown approach, Leftmost and Rightmost, Language of a grammar

4.3 Parse tree and its construction, Ambiguous grammar, Use of parse tree to show ambiguity in grammar

4.4 Regular Grammars: Right Linear and Left Linear, Equivalence of regular grammar and finite automata

4.5 Simplification of CFG: Removal of Useless symbols, Nullable Symbols, and Unit Productions, Chomsky Normal Form (CNF), Greibach Normal Form (GNF), Backus-Naur Form (BNF)

4.6 Context Sensitive Grammar, Chomsky Hierarchy Pumping Lemma for CFL, Application of Pumping Lemma, Closure Properties of CFL

Unit V: Push Down Automata (7 Hrs.)

5.1 Introduction to Push Down Automata (PDA), Representation of PDA, Operations of PDA, Move of a PDA, Instantaneous Description for PDA

5.2 Deterministic PDA, Non Deterministic PDA, Acceptance of strings by PDA, Language of PDA

5.3 Construction of PDA by Final State , Construction of PDA by Empty Stack,

5.4 Conversion of PDA by Final State to PDA accepting by Empty Stack and vice-versa, Conversion of CFG to PDA, Conversion of PDA to CFG

Unit VI: Turing Machines (10 Hrs.)

6.1 Introduction to Turing Machines (TM), Notations of Turing Machine, Language of a Turing Machine, Instantaneous Description for Turing Machine, Acceptance of a string by a Turing Machines

6.2 Turing Machine as a Language Recognizer, Turing Machine as a Computing Function, Turing Machine with Storage in its State, Turing Machine as a enumerator of stings of a language, Turing Machine as Subroutine

6.3 Turing Machine with Multiple Tracks, Turing Machine with Multiple Tapes, Equivalence of Multitape-TM and Multitrack-TM, Non-Deterministic Turing Machines, Restricted Turing Machines: With Semi-infinite Tape, Multistack Machines, Counter Machines

6.4 Curch Turing Thesis, Universal Turing Machine, Turing Machine and Computers, Encoding of Turing Machine, Enumerating Binary Strings, Codes of Turing Machine, Universal Turing Machine for encoding of Turing Machine

Unit VII: Undecidability and Intractability (5 Hrs.)

    7.1 Computational Complexity, Time and Space complexity of A Turing Machine, Intractability

    7.2 Complexity Classes, Problem and its types: Absract, Decision, Optimization

    7.3 Reducibility, Turing Reducible, Circuit Satisfiability, Cook's Theorem,

    7.4 Undecidability, Undecidable Problems: Post's Correspondence Problem, Halting Problem and its proof, Undecidable Problem about Turing Machines

Laboratory Works:

The laboratory work consists of design and implementation of finite state machines like DFA, NFA, PDA, and Turing Machine. Students are highly recommended to construct Tokenizers/ Lexers over/for some language. Students are advised to use regex and Perl (for using regular expressions), or any other higher level language for the laboratory works.

Text Books:

1.   John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Pearson - Addison-Wesley.

Reference Books:

1. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, 2nd Edition, Prentice Hall.
2. Michael Sipser, Introduction to the Theory of Computation, 3rd Edition, Thomson Course Technology
3. Efim Kinber, Carl Smith, Theory of Computing: A Gentle introduction, Prentice- Hall.
4. John Martin, Introduction to Languages and the Theory of Computation, 3rd Edition, Tata McGraw Hill.
5. Kenneth H. Rosen, Discrete Mathematics and its Applications to Computers Science, WCB/Mc-Graw Hill.