

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

CSE-Artificial Intelligence and Machine Learning/ Artificial Intelligence and Machine Learning, VI-Semester

AL-601 Theory of Computation

COURSE OBJECTIVE

- To understand computability, decidability, and complexity through problem solving.
- To analyze and design abstract model of computation & formal languages
- To understand and conduct mathematical proofs for computation and algorithms.

COURSE OUTCOMES

After completion of this course, the students would be able to:

- CO1.explain the basic concepts of switching and finite automata theory & languages.
- CO2.relate practical problems to languages, automata, computability and complexity.
- CO3.construct abstract models of computing and check their power to recognize the languages.
- CO4.analyse the grammar, its types, simplification and normal form.
- CO5.interpret rigorously formal mathematical methods to prove properties of languages, grammars and automata.
- CO6.develop an overview of how automata theory, languages and computation are applicable in engineering application.

Unit-I

Introduction of Automata Theory: Examples of automata machines, Finite Automata as a language acceptor and translator, Moore machines and mealy machines, composite machine, Conversion from Mealy to Moore and vice versa.

Unit-II

Types of Finite Automata: Non Deterministic Finite Automata (N DFA), Deterministic finite automata machines, conversion of N DFA to DFA, minimization of automata machines, regular expression, Arden's theorem. Meaning of union, intersection, concatenation and closure, 2 way DFA.

Unit-III

Grammars: Types of grammar, context sensitive grammar, and context free grammar, regular grammar. Derivation trees, ambiguity in grammar, simplification of context free grammar, conversion of grammar to automata machine and vice versa, Chomsky hierarchy of grammar, killing null and unit productions. Chomsky normal form and Greibach normal form.

Unit-IV

Push down Automata: example of PDA, deterministic and non-deterministic PDA, conversion of PDA into context free grammar and vice versa, CFG equivalent to PDA, Petrinet model.

Unit-V

Turing Machine: Techniques for construction. Universal Turing machine Multitape, multihead and multidimensional Turing machine, N-P complete problems. Decidability and Recursively Enumerable Languages, decidability, decidable languages, undecidable languages, Halting problem of Turing machine & the post correspondence problem.

RECOMMENDED BOOKS

- Introduction to Automata Theory Language & Computation, Hopcroft& Ullman, Narosa Publication.
- Element of the Theory Computation, Lewis &Christors, Pearson.
- Theory of Computation, Chandrasekhar & Mishra, PHI.
- Theory of Computation, Wood, Harper & Row.

- Introduction to Computing Theory, Daniel I-A Cohen, Wiley.

LIST OF EXPERIMENTS

1. Design a Program for creating machine that accepts three consecutive one.
2. Design a Program for creating machine that accepts the string always ending with 101.
3. Design a Program for Mode 3 Machine
4. Design a program for accepting decimal number divisible by 2.
5. Design a program for creating a machine which accepts string having equal no. of 1's and 0's.
6. Design a program for creating a machine which count number of 1's and 0's in a given string.
7. Design a Program to find 2's complement of a given binary number.
8. Design a Program which will increment the given binary number by 1.
9. Design a Program to convert NDFA to DFA.
10. Design a Program to create PDA machine that accept the well-formed parenthesis.
11. Design a PDA to accept WCWR where w is any string and WR is reverse of that string and C is a Special symbol.
12. Design a Turing machine that's accepts the following language $a^n b^n c^n$ where $n > 0$

COURSE OUTCOMES

After completion of this course, the students would be able to:

- CO1: judge various computational models.
- CO2: construct abstract models of computing.
- CO3: justify the power of abstract models in computing to recognize the languages.
- CO4: demonstrate analytical thinking and intuition for problem solving in the related areas.
- CO5: discuss the limitations of computation in problem solving.
- CO6: follow set of rules for syntax verification.