GUJARAT TECHNOLOGICAL UNIVERSITY

BACHELOR OF ENGINEERING SYLLABUS

Subject Code : 3164203

Subject Name : Theory of Computation

WEF Academic Year :	2020-21
Semester :	6
Category of the Course :	Professional Elective - II

Prerequisite :

Calculus, Data Structures and Algorithms, Set Theory.

Rationale :

- Theory of computation teaches how efficiently problems can be solved on a model of computation. The main thrust is to identify the limitations of the computers through formalizing computation (by introducing several models including Turing Machines) and applying mathematical techniques to the formal models obtained. It is also necessary to learn the ways in which computer can be made to think. Finite state machines can help in natural language processing which is an emerging area.

Course Scheme :

Teaching Scheme			Total Credits	Assessment Pattern and Marks				Total Marks
				Theory		Practical		
L	T	PR	C	ESE (E)	PA(M)	ESE (V)	PA (I)	
03	00	02	04	70	30	30	20	150

Course Content :

Sr. No.	Course Content	Total Hours	Weightage (%)
1	**Review of Mathematical Theory :** Sets, Functions, Logical statements, Proofs, Relations, Languages, Principal of Mathematical Induction, Strong Principle, Recursive Definitions, Structural Induction.	04	10%
2	**Regular Languages and Finite Automata :** Regular Expressions, Regular Languages, Application of Finite Automata, Automata with output - Moore machine & Mealy machine, Finite Automata, Memory requirement in a recognizer, Definitions, union- intersection and complement of regular languages, Non Deterministic Finite Automata, Conversion from NFA to FA, ^ - Non Deterministic Finite Automata, Conversion of NFA- ^ to NFA, Kleene's Theorem, Minimization of Finite automata, Regular And Non Regular Languages – pumping lemma.	12	25%

3	**Context free grammar (CFG) :** Definitions and Examples, Unions Concatenations And Kleene's of Context free language, Regular Grammar for Regular Language, Derivations and Ambiguity , Unambiguous CFG and Algebraic Expressions, BacosNaur Form (BNF), Normal Form – CNF.	08	15%
4	**Pushdown Automata, CFL And NCFL :** Definitions, Deterministic PDA, Equivalence of CFG and PDA & Conversion, Pumping lemma for CFL, Intersections and Complements of CFL, Non-CFL.	08	15%
5	**Turing Machine (TM) :** TM Definition, Model Of Computation, Turing Machine as Language Acceptor, TM that Compute Partial Function, Church Turning Thesis, Combining TM, Variations Of TM, Non Deterministic TM, Universal TM, Recursively and Enumerable Languages, Context sensitive languages and Chomsky hierarchy.	08	15%
6	**Computable Functions** : Partial - Total - Constant Functions, Primitive Recursive Functions, Bounded Mineralization, Regular function, Recursive Functions, Quantification, Minimalization, and μ-Recursive Functions, All Computable Functions Are μ-Recursive.	04	10%
7	**Un-decidability :** A Language That Can't Be Accepted, and a Problem That Can't Be Decided, Non Recursive Enumerable (RE) Language – Undecidable Problem with RE – Undecidable Problems about TM – Undecidable Problems Involving Context-Free Languages, Post's Correspondence Problem, The Class P and NP.	04	10%

Suggested Specification table with Marks (Theory) :

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
15	25	25	5	0	0

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)

Note : This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Suggested list of practical :

1. Write a C program that accepts a regular expression as input and constructs a corresponding NFA. The program should be able to handle the basic operators like union, concatenation, and closure.
2. Write a C program that accepts a string as input and uses a DFA to determine if the string is in a given regular language. The program should print "Accepted" if the string is in the language, and "Rejected" otherwise.
3. Write a C program that implements a Moore machine. The program should accept a set of states, input alphabet, output alphabet, transition function, and initial state, and should be able to simulate the machine on a given input string.
4. Write a C program that implements a Mealy machine. The program should accept a set of states, input alphabet, output alphabet, transition function, and initial state, and should be able to simulate the machine on a given input string.
5. Write a C program that accepts a regular language as input and constructs a corresponding DFA. The program should use the subset construction algorithm to convert the NFA to a DFA.
6. Write a C program that accepts an NFA as input and converts it to an equivalent DFA using the powerset construction algorithm.
7. Write a C program that uses Kleene's theorem to generate a regular expression for a given regular language.
8. Write a C program that uses the pumping lemma to prove that a given language is not regular.
9. Write a C program that accepts a CFG as input and constructs a corresponding parse tree for a given input string. The program should be able to handle productions with multiple symbols on the right-hand side.
10. Write a C program that accepts a CFG as input and constructs a corresponding Chomsky normal form (CNF) grammar. The program should be able to handle nonterminal symbols with multiple productions.
11. Write a C program that accepts a regular expression as input and constructs a corresponding regular grammar. The program should be able to handle the basic operators like union, concatenation, and closure.
12. Write a C program that accepts a context-free grammar as input and determines if it is ambiguous. If the grammar is ambiguous, the program should print out a string that has two or more parse trees. If the grammar is unambiguous, the program should print out the unique parse tree for a given input string.
13. Write a C program that accepts a context-free grammar in Backus-Naur Form (BNF) and constructs a corresponding parse table for a given input string. The program should be able to handle left-recursive and ambiguous grammars
14. Write a C program that accepts a pushdown automaton (PDA) as input and determines if it is deterministic. The program should be able to handle multiple transitions from a single state on the same input symbol.
15. Write a C program that accepts a context-free grammar (CFG) as input and constructs a corresponding PDA. The program should be able to handle nonterminal symbols with multiple productions.
16. Write a C program that accepts a context-free language (CFL) as input and uses the pumping lemma to determine if it is a CFL. The program should be able to handle grammars with nullable productions and productions with more than one nonterminal symbol.
17. Write a C program that accepts two CFLs as input and constructs a PDA for their intersection. The program should be able to handle grammars with nullable productions and productions with more than one nonterminal symbol.

18. Write a program that takes a description of a Turing Machine (TM) as input and simulates the execution of the TM on a given input string.

Reference Books :

1. Introduction to Languages and the Theory of Computation, 4th by John Martin, Tata Mc Graw Hill
2. An introduction to automata theory and formal languages By Adesh K. Pandey, Publisher: S.K. Kataria& Sons
3. Introduction to computer theory By Deniel I. Cohen , Joh Wiley & Sons, Inc
4. Computation: Finite and Infinite By Marvin L. Minsky Prentice-Hall
5. Compiler Design By Alfred V Aho, Addison Weslley
6. Introduction to the Theory of Computation By Michael Sipser
7. Automata Theory, Languages, and Computation By John Hopcroft, Rajeev Motowani, and Jeffrey Ullman

Course Outcome :

Sr. No.	CO statement	Marks % weightage
CO1	Use the concepts and techniques of discrete mathematics for theoretical computer science.	10%
CO2	Identify different formal languages and their relationship.	25%
CO3	Classify and construct grammars for different languages and vice-versa.	25%
CO4	Build finite automata, push down automata and turing machine.	25%
CO5	Analyze various concepts of un-decidability and Computable Function and Discuss analytically and intuitively for problem-solving situation.	15%

List of Open Source Software/learning website :

1. http://en.wikipedia.org/wiki/Theory_of_computation
2. http://meru.cecs.missouri.edu/courses/cecs341/tc.html
3. https://www.geeksforgeeks.org/introduction-of-theory-of-computation/
4. http://www.vssut.ac.in/lecture_notes/lecture1428551440.pdf
5. https://nptel.ac.in/courses/106/104/106104028/

****