<table>
<tr><td colspan="4">Devi Ahilya University, Indore, India
Institute of Engineering & Technology</td><td colspan="3">III Year B.E. (Information Technology (Full Time)</td></tr>
<tr><td>Subject Code & Name</td><td colspan="6">Instructions Hours per Semester & Credits</td></tr>
<tr><td>5RIPC1
Theory of Computation</td><td colspan="2">Classroom Instruction (CI)</td><td>Lab Instruction (LI)</td><td>Term Work (TW) and Self Learning (SL)</td><td>Total no. of Hours Per semeste</td><td>Total Credits (Total Hours/30)</td></tr>
<tr><td rowspan="2">Duration of Theory Paper: 3 Hours</td><td>L</td><td>T</td><td>P</td><td>TW+SL</td><td rowspan="2">90</td><td rowspan="2">3</td></tr>
<tr><td>20</td><td>10</td><td>0</td><td>60</td></tr>
</table>

Prerequisite:

Students should have a background in discrete mathematics, data structures, and programming languages.

Course Objective:

The course is designed

1. To understand the mathematical modeling
2. To analyse the mathematical structures like automata.
3. To demonstrate a familiarity with grammar, languages and automata.
4. To apply important logical design paradigms and methods of analysis.
5. To differentiate polynomial and nonpolynomial problems.

Learning Objectives:

- To provide a formal connection between algorithmic problem solving and the theory of languages and automata and develop them into a mathematical (abstract) view towards algorithmic design and in general computation itself. The course should in addition clarify the practical view towards the applications of these ideas in the engineering part as well.

Course Outcome:

Students earned credits will develop ability to

CO.No.	CO	PO
CO1	Ability to write a problem's solution using different ways.	PO-1, PO2, PO3
CO2	To Analyse the languages using a variety of solutions	PO-1, PO-2
CO3	To apply problem solving using grammar and automata theory	PO-1, PO2, PO3
CO4	Apply important design paradigms and methods of analysis	PO2, PO-3, PO4
CO5	To have mathematic and logical thinking for real world problems	PO2, PO3, PO4
CO6	To apply regular and context free grammars rules for grammar of programming languages.	PO6, PO9
CO7	Identify and differentiate polynomial and nonpolynomial problems	PO4

CO-PO Relationship

CO	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12
CO1	2	2	2									
CO2	2	2										
CO3	2	3	3									
CO4		3	3	3								
CO5		2	2	3								
CO6						3			3			
CO7				3								

* CO (rows) mention nil/very small/insignificant contribution to the PO(column)
1→ relevant and small significance 2 → medium or moderate and 3 →strong

COURSE CONTENTS

UNIT-I

Finite Automata and Regular Languages: Motivation for studying theory of computation, Notion of formal languages and grammars, Kleene's Closure, Regular Expressions and Regular languages, closure properties of regular languages, Finite Automata. Finite Automata with output: Mealy and Moore machines, applications.

UNIT-II

Nondeterminism and Minimization: Nondeterministic Finite Automata, Acceptance condition. Kleene's Theorem, Myhill-Nerode relations, Minimization Algorithm, Non-Regular languages, Pumping Lemma for regular languages.

UNIT-III

Grammars and Context-Free Languages: Grammars and Chomsky Hierarchy, Context-Free Grammars, Context-Free Languages (CFLs), Inherent Ambiguity of CFLs, closure properties of CFLs, Eliminating useless symbols; null-productions; and unit productions, Chomsky Normal Form, Greibach Normal Form, Cock-Younger-Kasami (CYK) Algorithm, Applications to Parsing.

UNIT-IV

Pushdown Automata: Pushdown Automata (PDAs), PDAs vs CFLs. Deterministic PDAs and CFLs, applications, notion of acceptance for PDAs: acceptance by final states, and by empty stack; the equivalence of the two notions, Proof that CFGs generate the same class of languages that PDAs accept, Pumping Lemma for CFLs.

UNIT-V

Turing Machines and Computability: Introduction to Turing Machines, Configurations, Halting vs Looping, Turing computability, Nondeterministic, multitape and other versions of Turing machines. Church`s thesis, Universal Turing Machines, Linear Bounded Automata (LBAs) and context-sensitive languages, Recursive and Recursively enumerable languages, Undesirability of Halting Problem and unsolvable problems about Turing Machines, the diagonalization language and proof that it is not Recursively enumerable.

.

Learning Outcomes:

After completing the course, the student will be able to:

1) Model, compare and analyse different computational models.
2) Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.
3) Identify limitations of some computational models and possible methods of proving them.
4) Have an overview of how the theoretical study in this course is applicable to and engineering application like designing the compilers.
5) Have an understanding of the solvable and unsolvable problems and their computational behaviors.

Books Recommended:

[1]. Daniel I.A. Cohen, Introduction to Computer Theory, John Wiley, 1990

[2]. John C. Martin, Introduction to Languages and the Theory of Computation, 3/e Tata McGraw Hill, 2005 [3]. J.E. Hopcroft and J.D.Ullman, Introduction to Automata, Languages and Computation, Narosa Publishing

House, 1995

[4]. J.E. Hopcroft, Rajeev Motwani and J.D.Ullman, Introduction to Automata, Languages and Computation,

Pearson Education, Asia, 2002

[5]. H.R. Lewis and C.H. Papadimitrou, Elements of the Theory of Computation, Prentice Hall Inc., 1999 [6]. M. Sipser, Introduction to the Theory of Computation, Brooks/Cole Thomson Learning, 1996

[7]. Zohar Manna, Mathematical Theory of Computation, McGraw Hill, 1997