R.T.M. Nagpur University, Nagpur
FOUR YEAR B.E. COURSE
(Revised Curriculum as per AICTE Model Curriculum)
Computer Science and Engineering
B.E. Second Year Proposed Scheme

Fourth Semester:-

SN	Subject	Teaching Scheme			Evaluation Scheme			Credits	Category
		L	T	P	CA	UE	Total		
1	Discrete Mathematics and Graph Theory	03	-	-	30	70	100	03	PCC-CS
2	Data Structure and Program Design	03	01	-	30	70	100	04	PCC-CS
3	Database Managements Systems	03	01	-	30	70	100	04	PCC-CS
4	Computer Networks	03	-	-	30	70	100	03	PCC-CS
5	Theory of Computation	03	01	-	30	70	100	04	PCC-CS
6	System Programming	03	-	-	30	70	100	03	PCC-CS
7	Data Structure and Program Design-Lab	-	-	02	25	25	50	01	PCC-CS
8	Database Managements Systems-Lab	-	-	02	25	25	50	01	PCC-CS
9	Computer Workshop-II (Python)	-	-	02	25	25	50	01	PCC-CS
10	Constitution of India (Audit Course)	02	-	-	-	-	-	Audit	MC
	Total	20	03	06			750	24	

RASHTRASANT TUKADOJI MAHARAJ NAGPUR UNIVERSITY, NAGPUR

FOUR YEAR BACHELOR OF ENGINEERING (B.E.) DEGREE COURSE

SEMESTER: FOURTH (C.B.S.C)

BRANCH: COMPUTER SCIENCE AND ENGINEERING

Subject : Theory of Computation **Subject Code:** BECSE405T

Load	Credit	Total Marks	Sessional Marks	University Marks	Total
3 hrs (Theory)	03	100	30	70	100

Aim: The main motivation behind developing Theory of Computation was to develop methods to describe and analyze the dynamic behavior of discrete systems.

Prerequisite(s): Basics of Discrete Mathematics

Course Objective/Learning Objective:

1	To discuss the Chomsky classification of formal language with discussion on grammar and automata for regular, context-free, context sensitive and unrestricted language.
2	Understand the basic properties of Turing machines and computing with Turing machines.
3	To discuss the notion of decidability.
4	To compute Ackerman function and analyze recursively and non-recursively enumerable language

Course Outcome:

At the end of this course Student are able to:

CO1	Design finite automata and its minimization along with Moore and Mealy machines.
CO2	Apply regular expression and create grammar for the same.
CO3	Deal with context free grammar and various normal forms of CFGs.
CO4	Create Push Down Automata for the given CFG and inter-conversion of the same.
CO5	Create Turning Machine for the grammar and Deal with Recursive and Recursively Enumerable Languages.

UNIT I:

Finite Automata (FA): Basic Terminology and Definitions, Chomsky hierarchy, Deterministic Finite Automata , language of a DFA. Nondeterministic Finite Automata, Equivalence of Deterministic and Non-deterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machines) and Inter conversion.

UNIT II:

Regular Grammars (RG): Definition, regular grammars and FA, Conversion. Proving languages to be non-regular, Pumping lemma, applications, Closure properties of regular languages.

Regular Expressions (RE): Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions, Converting from DFA's to Regular Expressions, Converting Regular Expressions to Automata, applications of Regular Expressions.

UNIT III:

Context Free Grammar (CFG): Definition, Parse Tress, Derivation Trees, Rightmost and Leftmost derivations of Strings and Conversions. Ambiguity in CFGs, Minimization of CFGs, Normal forms for CFG, Pumping Lemma for CFLs.

Unit –IV:

Push down Automata (PDA): Definition, Model, Non-determinism, acceptance by two methods and their equivalence, conversion of PDA to CFG, CFG to PDAs, closure and decision properties of CFLs.

UNIT V:

Turing Machines (TM) : Formal definition and behavior, Languages of a TM, TM as acceptor, TM as transducers, Variations of TM, Linear Bounded Automata, TM as computer of function. Properties of recursive and recursively enumerable languages, Recursively enumerable set, Undesirability, Decidability and solvability, Post correspondence Problem, Primitive recursive functions, Ackerman function

Textbooks:

- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory Languages and Computation, 3rd edition, Pearson Education.
- Michael Sipser, Introduction to the Theory of Computation, 3rd edition, Cengage Learning.
- Peter Linz,An Introduction to Formal Languages and Automata, 5th Edition, Malloy, Inc.
- Vivek Kulkarni, Theory of Computation, Oxford University Press, ISBN-13: 978-0-19-808458-7.
- Theory of Computation - O.G. Kakde ,University Science Press

Reference books:

- K. L. P Mishra, N. Chandrashekaran , Theory of Computer Science-Automata Languages and Computation, 2nd edition, Prentice Hall of India, India.
- John C Martin, Introduction to languages and the Theory of Computation, TMH
- Daniel I.A. Cohen, John Wiley, Introduction to Computer Theory.
- P.K. Srimani, Nasir S, A Text book on Automata Theory, Cambridge University Press.
- Kamala Krithivasan, Rama R, Introduction to Formal languages Automata Theory and Computation Pearson.