

Semester: VI				
Programme : B.Sc. Computer Science (Hons)				
Course : THEORY OF COMPUTATION				
Paper code: C3CS230611T				Credits: 4
Hours/week : Theory: 4				
Category: Core/MDC/SEC/VAC : Core				
Theory / Practical / Composite : Theory				
No of Modules : 2				
<p>Course Overview: This course provides a rigorous foundation in the mathematical theory of computation, exploring the relationships between formal languages, automata, and computational models. Students investigate how abstract machines (finite automata, pushdown automata, Turing machines) recognize language classes of increasing complexity (regular, context-free, context-sensitive, recursively enumerable) within the Chomsky hierarchy. The curriculum emphasizes mathematical precision for proving language properties, algorithmic design for automata construction, and theoretical limits of computation—essential for understanding compiler design, natural language processing, computational complexity, and decidability.</p>				
Course Outcome:				
1. Explain fundamental concepts of automata theory, formal languages, and mathematical preliminaries, and apply them to model computational problems using alphabets, strings, and language operations.				
2. Design deterministic/non-deterministic finite automata (DFA/NFA) and regular expressions and demonstrate their equivalence for recognizing regular languages through conversion techniques and pumping lemma analysis.				
3. Construct context-free grammars (CFGs) and pushdown automata (PDA) and transform grammars into normal forms (CNF/GNF) to analyze context-free language properties and parsing strategies.				
4. Analyze Turing machines as computational models, design solutions for computationally complex problems, and evaluate their limitations (e.g., halting problem) for decidability and complexity classification.				
5. Compare language classes within the Chomsky hierarchy and assess their relationships to computational models for solving problems in compiler design, NLP, and AI.				
Prerequisites:				
<ul style="list-style-type: none"> • Discrete Structures & Graph Algorithms (sets, relations, logic, induction) • Basic programming proficiency • Mathematical maturity (comfort with formal proofs and abstract reasoning) 				
SYLLABUS				
UNIT/Module	CONTENT	HOURS or NUMBER OF CLASSES	CO Mapping	COGNITIVE LEVEL
Group A				
I.	Introduction to Theory of Automata: Mathematical Preliminaries and notation, Basic concepts of Languages, Grammars and Automata, Some Applications	9	CO1	Understand, Apply (K2, K3)

II.	Languages: Alphabets, string, language, Basic Operations on language, Concatenation, Kleene's Star, Kleene's theorem.	9	CO1	Understand, Apply (K2, K3)
III.	Finite Automata and Regular Languages: Regular Expressions, Transition Graphs, Deterministic Finite automata (DFA) and nondeterministic finite automata (NFA), NFA to DFA Conversion, Regular languages and their relationship with finite automata, Pumping lemma and closure properties of regular languages.	8	CO2	Apply, Analyse (K3, K4)
Group B				
IV.	Context free languages: Chomsky hierarchy of languages, Context free grammars, parse trees, ambiguities in grammars and languages, Pushdown automata (Deterministic and Nondeterministic), Properties of context free languages, Chomsky Normal Form(CNF), Greibach Normal Form(GNF).	10	CO3 CO5	Apply, Analyse, Evaluate (K3, K4, K5)
V.	Turing Macines and Models of Computations context sensitive language, Turing Machine as a model of computation, Designing Turing, Non deterministic Turing machine, halting problem.	10	CO5	Analyse, Evaluate (K4, K5)
VI.	Other Turing Models: Linear bounded automata, Universal Turing Machine	6	CO5	Analyse, Evaluate (K4, K5)
Text Books				
1. P. Linz, An Introduction to Formal Language and Automata 4th edition Publication: Jones Bartlett, 2006				
2. Hopcroft, Aho, Ullman, Introduction to Automata theory, Language & Computation–3rd Edition, Pearson Education. 2006				
3. Lewis & Papadimitriou, Elements of the theory of computation, PHI 1997.				
Evaluation				
Theory CIA: 25 Attendance: 5 Semester Exam: 70				
Paper Structure for Theory Semester Exam Module: Answer 5 out of 7 of 7 marks each in Group A Answer 5 out of 7 of 7 marks each in Group B				

Course outcomes (COs) and Cognitive Level Mapping

COs	CO Description	Cognitive levels
CO1	Explain fundamental concepts of automata theory, formal languages, and mathematical preliminaries, and apply them to model computational problems using alphabets, strings, and language operations.	Understand, Apply (K2, K3)
CO2	Design deterministic/non-deterministic finite automata (DFA/NFA) and regular expressions and demonstrate their equivalence for recognizing regular languages through conversion techniques and pumping lemma analysis.	Apply, Analyse (K3, K4)
CO3	Construct context-free grammars (CFGs) and pushdown automata (PDA), and transform grammars into normal forms (CNF/GNF) to analyze context-free language properties and parsing strategies.	Analyse, Evaluate (K4, K5)
CO4	Analyze Turing machines as computational models, design solutions for computationally complex problems, and evaluate their limitations (e.g., halting problem) for decidability and complexity classification.	Analyse, Evaluate (K4, K5)
CO5	Compare language classes within the Chomsky hierarchy and assess their relationships to computational models for solving problems in compiler design, NLP, and AI.	Analyse, Evaluate (K4, K5)