FORMAL LANGUAGES AND AUTOMATA THEORY	
CSE 225	**Credits: 3**
Instruction**:3** Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Prerequisites:

- The students are expected to have a strong background in the fundamentals of discrete mathematics (symbolic logic, set, induction, number theory, summation, series, combinatory, graph, recursion, basic proof techniques, etc.), algorithms and data structures.
- Some knowledge of programming languages, programming, and computer architecture will be helpful.

Course Objectives:

The course should enable the students:

- To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.
- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- Classify machines by their power to recognize languages.
- Employ finite state machines to solve problems in computing.
- To understand deterministic and non-deterministic machines.
- To understand the differences between decidability and undecidability.

Course Outcomes:

the end of the course, the student will be able to:	
1.	Acquire a fundamental understanding of the core concepts in automata theory, construct DFAand NFA. Ability to transform between equivalent finite automata, Construct Epsilon-NFA and transform between equivalent finite automata
2.	Understand the power and the limitations of regular expressions and design regular expressions. Compute transformation between finite automata and regular expressions
3.	Describe and construct Context Free Grammar and Pushdown Automata, transformation between them
4.	Construct and analyze the use and properties of Turing machines performing simple tasks, with recent trends and applications in the area of finite state machines
5.	Understand the concepts of recursively enumerable languages and undecidability problem.

Mapping of Course Outcomes with Program Outcomes:

S.NO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	3	3	2	3	1	1	0	1	1	3	0	2	3	1
CO 2	2	2	3	2	0	0	0	1	1	2	0	2	2	1
CO 3	2	2	3	2	0	0	0	1	1	2	0	2	3	1
CO 4	2	3	3	2	0	0	0	1	1	2	0	3	2	1
CO5	2	3	3	3	0	0	0	1	1	2	0	3	3	2

SYLLABUS

UNIT-I: **15 periods**

FINITE AUTOMATA (FA): Introduction, Deterministic Finite Automata (DFA) -Formal definition, simpler notations (state transition diagram, transition table), language of a DFA.

NONDETERMINISTIC FINITE AUTOMATA (NFA)- Definition of NFA, language of an NFA, Equivalence of Deterministic and Nondeterministic 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.

LEARNING OUTCOMES:

1. An ability to design grammars and automata (recognizers) for different language classes.
2. An ability to prove and disprove theorems establishing key properties of formal languages and automata.

UNIT-II: **12 periods**

REGULAR EXPRESSIONS (RE): Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA are to Regular Expressions, Converting Regular Expressions to Automata, and applications of Regular Expressions.

REGULAR GRAMMARS: Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular -Pumping lemma, applications, and Closure properties of regular languages.

LEARNING OUTCOMES:

1. Design Finite Automata's for different Regular Expressions and Languages.
2. Understand the Pumping lemma for proving that languages are not regular.

UNIT-III: **12 periods**

CONTEXT FREE GRAMMER (CFG): Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG's, Minimization of CFG's, CNF, GNF, Pumping Lemma for CFL's, Enumeration of Properties of CFL (Proof's omitted).

LEARNING OUTCOMES:

1. To construct context free grammar for various languages.
2. Describe the language accepted by automata or generated by a regular expression or a context-free grammar.

UNIT-IV: **12 periods**

PUSHDOWN AUTOMATA: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA.

TURING MACHINES (TM): Formal definition and behavior, Languages of a TM, TM as accepters and as a computer of integer functions, Types of TMs.

LEARNING OUTCOMES:

1. To solve various problems of applying normal form techniques, push down automata and Turing Machines.
2. Construct a pushdown automaton for a given context-free language
3. Construct a total Turing machine deciding a given problem

UNIT-V: **12 periods**

RECURSIVE AND RECURSIVELY ENUMERABLE LANGUAGES (REL): Properties of recursive and recursively enumerable languages, Universal Turing machine, The Halting problem, Undecidable problems about TMs. Context sensitive language and linear bounded automata (LBA), Chomsky hierarchy, Decidability, Post's correspondence problem (PCP), undecidability of PCP.

LEARNING OUTCOMES:

1. Understand the basic results on computability, including undecidable problems such as the halting and Post correspondence problems, and their significance.
2. Define the various categories of languages and grammars in the Chomsky hierarchy.

TEXT BOOKS

1. "Introduction to Automata Theory Languages and Computation". Hopcroft H.E. and Ullman J. D. Pearson Education.

REFERENCE BOOKS

1. Theory of Computer Science – Automata languages and computation -Mishra and Chandrashekaran, 2nd edition, PHI.
2. Introduction to languages and the Theory of Computation, John C Martin, TMH, 4th edition.

ONLINE WEB RESOURCES

1. https://nptel.ac.in/courses/111103016/

Prepared By: Dr.K. Suresh, Mr.G.Sathar