FORMAL LANGUAGE & AUTOMATA THEORY ()

Teaching Scheme	Examination Scheme						
Credits:	CIE	HA	SCE	ESE	PR/OR	TW	TOTAL
Lecture's/Week(L): Hrs/week Practical/Week(P): Hrs/week Tutorial/Week(T): Hrs/week	20	20	20	40	--	–	100

Prerequisites: Basic Mathematics, Discrete Structure

Course Objectives:

1.Study the foundational mathematical concepts and formal models of finite automata—including DFA, NFA, Moore and Mealy machines
2.Construct, simplify, and convert regular expressions and finite automata
3.Study the representation of grammars, classify them, analyze context-free grammars through derivations and parse trees.
4.Study the foundations and computational capabilities of Push Down Automata and Turing Machines.

Course Outcomes:

After studying this course, students will be able to

1.To understand the foundational mathematical concepts and formal models of finite automata—including DFA, NFA, Moore and Mealy machines
2. To develop the ability to construct, simplify, and convert regular expressions and finite automata
3. To understand the definition and representation of grammars, classify them, analyze context-free grammars through derivations and parse trees
4. To understand the theoretical foundations and computational capabilities of Push Down Automata and Turing Machines

Unit I - Theory of Automata — **6 Hrs.**

Basic Mathematical Objects: Sets, Logic, functions, Relations, Strings. Definition of Finite Automata (FA), Description of FA, Transition Systems, Acceptability of a String by a FA, Deterministic and Non deterministic FA, Equivalence of DFA and NFA, FA with output: Moore and Mealy machines-Definitions, Models, inter-conversion, minimization of FA. Applications of Finite Automata.

Unit II - Regular Expressions (RE) and Languages — **6 Hrs.**

Recursive definition of regular expression, regular set, identities of regular expressions, Conversion of NFA With epsilon moves to DFA, Conversion-RE to DFA , Conversion-DFA to RE, Equivalence of R.E, Equivalence of FA, Arden's theorem, Pumping lemma for regular languages, Closure properties of regular languages, Applications of R.E

Unit III - Grammar — **6 Hrs.**

Grammar- Definition, representation of grammar, Chomsky hierarchy, Context Free Grammar- Definition, Derivation, sentential form, parse tree, inference, derivation, parse tree, ambiguity in grammar and language, Simplifications of context free Grammar-Eliminating unit productions, useless symbols, and Null-productions, Normal Forms for CFG- Chomsky normal form.

Unit IV - Push Down Automata & Turing machines — **6 Hrs.**

Push Down Automata- Definition, Notation, acceptance by final state, acceptance by empty stack, Equivalence of PDA and CFG- Grammar to PDA , Deterministic PDA and Non Deterministic PDA, Turing machine Model, Representation of Turing machine, Language acceptability by Turing machine, Design of Turing machine, Types of TM, Halting Problem

Textbooks:

1. Mishra K., Chandrasekaran N., 'Theory of Computer Science (Automata, Languages and Computation)", Second Edition, Prentice Hall of India

2. John C Martin. "Introduction to Language and Theory of Computation", Third edition, Tata McGraw- Hill.	
Reference Books:	
1. Hopcroft J., Motwani R., Ullman J., "Introduction to Automata Theory, Languages and Computations", Third edition, Pearson Education Asia.	
List of Assignments	
1	Construct a DFA over $\Sigma = \{0,1\}$ that accepts all strings ending with “01”. Check if the string "1101" is accepted.
2	Design an NFA for strings over $\Sigma = \{a,b\}$ that contain “ab” as a substring. Convert the NFA to an equivalent DFA.
3	Design a Moore machine that outputs 1 for input ‘1’, and 0 otherwise. Convert it to a Mealy machine. Minimize a given DFA using the partitioning method.
4	Write a RE for the language of strings over {a, b} that start and end with ‘a’. Use RE identities to simplify (a + b)* a (a + b)*.
5	Convert an ε-NFA to DFA using subset construction. Convert RE = (a + b)*ab to an equivalent DFA.
6	Use Arden’s theorem to solve R = aR + b. Prove $\{a^n b^n \mid n \geq 0\}$ is not regular using Pumping Lemma.
7	Create a CFG for balanced parentheses. Show leftmost and rightmost derivations for “(())”. Identify if grammar is ambiguous.
8	Simplify the CFG: $S \rightarrow AB \mid \varepsilon$, $A \rightarrow aA \mid \varepsilon$, $B \rightarrow b$
9	• Design a PDA to accept $\{a^n b^n \mid n \geq 0\}$ using empty stack method. • Convert the CFG: $S \rightarrow aSb \mid \varepsilon$ to an equivalent PDA.
10	Design a TM to accept strings with equal number of a’s and b’s. Explain the concept of Halting Problem with a simple example.