

## FORMAL LANGUAGES AND AUTOMATA THEORY

<b>B.Tech. III Year I Semester</b>								
<b>Course Code</b>	<b>Category</b>	<b>Hours / Week</b>			<b>Credits</b>	<b>Maximum Marks</b>		
<b>CS3101PC</b>	<b>Core</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>	<b>CIA</b>	<b>SEE</b>	<b>Total</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>	<b>25</b>	<b>75</b>	<b>100</b>
<b>Contact classes: 60</b>	<b>Tutorial Classes : NIL</b>	<b>Practical classes : NIL</b>			<b>Total Classes :60</b>			
<b>Prerequisite: A course on "Discrete Mathematics"</b>								

### Course Objectives

- 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

- Able to understand the concept of abstract machines and their power to recognize the languages.
- Able to employ finite state machines for modeling and solving computing problems.
- Able to design context free grammars for formal languages.
- Able to distinguish between decidability and undecidability.
- Able to gain proficiency with mathematical tools and formal methods.

### MODULE- I

**Introduction to Finite Automata:** Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory–Alphabets, Strings, Languages, Problems.

**Nondeterministic Finite Automata:** Formal Definition, an application, Text Search, Finite Automata with Epsilon-Transitions.

**Deterministic Finite Automata:** Definition of DFA, How a DFA Process Strings, The language of DFA, Conversion of NFA with  $\epsilon$ -transitions to NFA without  $\epsilon$ -transitions.

Conversion of NFA to DFA, Moore and Melay machines.

## **MODULE- II**

**Regular Expressions:** Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

**Pumping Lemma for Regular Languages:** Statement of the pumping lemma, Applications of the Pumping Lemma.

**Closure Properties of Regular Languages:** Closure properties of Regular languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

## **MODULE- III**

**Context-Free Grammars:** Definition of Context- Free Grammars, Derivations Using a Grammar, Left most and Right most Derivations, the Language of a Grammar, Sentential Forms, Parse Trees, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.

**Push Down Automata:** Definition of the Push down Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state, Acceptance by empty stack, Deterministic Pushdown Automata. From CFG to PDA, From PDA to CFG.

## **MODULE- IV**

**Normal Forms for Context-Free Grammars:** Eliminating useless symbols, Eliminating  $\epsilon$ -Productions. Chomsky Normal form, Griebach Normal form.

**Pumping Lemma for Context-Free Languages:** Statement of pumping lemma, Applications.

**Closure Properties of Context-Free Languages:** Closure properties of CFL's, Decision Properties of CFL's

**Turing Machines:** Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine

## **MODULE- V**

**Types of Turing machine:** Turing machines and halting

**Undecidability:** Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Recursive languages, Properties of recursive languages, Post's Correspondence Problem, Modified Post Correspondence problem, Other Undecidable Problems, Counter machines.

**TEXT BOOKS:**

1. Introduction to Automata Theory, Languages and Computation, 3<sup>rd</sup> Edition, John E.Hopcroft, Rajeev Motwani, Jeffrey D.Ullman, Pearson Education.
2. Theory of Computer Science–Automata languages and computation, Mishra and Chandrashekar, 2<sup>nd</sup> edition, PHI.

**REFERENCE BOOKS:**

1. Introduction to Languages and the Theory of Computation, John C Martin, TMH.
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. A Textbook on Automata Theory, P.K.Srimani, NasirS.F.B, Cambridge University Press.
4. Introduction to the Theory of Computation, Michael Sipser, 3<sup>rd</sup>edition, Cengage Learning.
5. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, Pearson.