L	T	P	C
4	1	0	5

Course Code: CSE211
Semester: III

FORMAL LANGUAGE & AUTOMATA THEORY

Course Objectives:

This course will help the learner to discuss different classes of formal languages in Chomsky hierarchy, their properties and to design the acceptor machines for those languages

UNIT - I 18 Periods

Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages
Regular languages and finite automata: Deterministic Finite Automata (DFA) - Nondeterministic Finite Automata (NFA) – Finite Automata with Epsilon Transitions – Regular Expressions - Finite Automata and Regular Expressions - Kleene’s theorem- Regular grammars and Equivalence with Finite Automata - Properties of Regular Languages: Proving Languages Not to Be Regular–Closure Properties of Regular Languages - Myhill-Nerode theorem -Minimization of Finite Automata

UNIT - II 19 Periods

Context-free languages and pushdown automata: Context-free grammars (CFG) – Parse Trees - Ambiguity in Grammars and Languages - nondeterministic pushdown automata (PDA) - Equivalence of PDAs and CFGs - Deterministic Pushdown Automata - Properties of Context Free Languages: Normal Forms for Context Free Grammars - Chomsky and Greibach normal forms - Pumping lemma for context-free languages - closure properties of CFLs

UNIT - III 19 Periods

Context-sensitive languages: Context-sensitive grammars (CSG) and languages - linear bounded automata and equivalence with CSG
Introduction to Turing machines: The Turing Machine (TM) - Church-Turing thesis Programming Techniques for Turing Machines – extensions to the Basic Turing Machine – Restricted Turing Machine - Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators

UNIT - IV 19 Periods

Undecidability: A Language That Is Not Recursively Enumerable (RE)- Diagonalization languages - An Undecidable Problem that Is RE - Universal Turing machine - undecidable problems about Turing Machines - Reductions between languages and Rice s theorem

Basic Introduction to Complexity: Intractable Problems - Introductory ideas on Time complexity of deterministic and nondeterministic Turing machines – Classed P and NP, An NP- complete Problem - Cook’s Theorem - Additional NP -Complete problems

TEXT BOOKS

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson, 3rd Edition, 2011.
2. Peter Linz, An Introduction to Formal Languages and Automata, Jones and Bartle Learning International, United Kingdom, 6th Edition, 2016.

REFERENCES

1. Harry R, Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson, 2015.
2. Michael Sipser, Introduction to the Theory of Computation, 2nd Edition, Thomson Course Technology, 2006.
3. John Martin, Introduction to Languages and the Theory of Computation, McGraw Hill Higher Education, 2007.
4. M. R. Garey and D. S. Johnson, Computers and Intractability, A Guide to the Theory of NP Completeness, W. H. Freeman & Co, 1990.

UNITWISE LEARNING OUTCOMES

Upon successful completion of each unit, the learner will be able to

Unit I	• List the properties of regular languages, design regular expressions and construct equivalent automata. • Identify and prove a language is regular or not
Unit II	• Design the context-free grammars for context-free languages, transform them into normal forms • Construct Push Down Automata for the equivalent context-free grammars • Examine the properties of CFL
Unit III	• Design Context Sensitive Grammars for Context Sensitive Languages • Design Turing Machine for simple and complex tasks • Describe different Turing Machine models
Unit IV	• Summarize Chomsky Hierarchy, and differentiate recursive &recursively enumerable languages • Describe concepts of computational complexity, unsolvable and undecidable problems.

COURSE LEARNING OUTCOMES

Upon successful completion of this course, the learner will be able to

- Design an appropriate automaton for a given language
- Construct a grammar for a given language
- Examine the equivalence of Automata and Grammars
- Employ Pumping Lemma to find whether a language is not regular or a language is not Context Free
- Examine the properties of Languages
- Select appropriate Turing Machines for a given problem
- Describe concepts of computational complexity, unsolvable and undecidable problems