

Theory of Automata and Formal Languages

Course Code: CS-352

Credit Hours: 3-0

Course Description

This is a foundational course in computer science. The purpose of this course is to ask very fundamental questions

about the very nature of computation:

1. What is a computation?
2. What is the exact definition of an algorithm?
3. Are there any problems that cannot be computationally solved?
4. What is the relation of automata and formal languages with formal languages?
5. Can we identify problems that can be solved in principle (given a lot of resources) but cannot be solved in practice?

Text Book:

1. Michael Sipser, "Introduction to the Theory of Computation", 3rd Ed., Cengage Learning, 2013

Reference Book:

1. J.E. Hopcroft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computation", 2nd Ed., Addison-Wesley 2001.
2. Elaine A. Rich "Automata, Computability and Complexity: Theory and Applications", Prentice Hall, 2013

Prerequisites

None

ASSESSMENT SYSTEM FOR THEORY

Quizzes	10%
Assignments	10%
Mid Terms	30%
Project	10%
ESE	40%

Teaching Plan

Week No	Topics	Learning Outcomes
1	Introduction	Introducing deterministic finite automata, designing a DFA
2-8	Automata-I	Regular languages, regular expressions, non-deterministic finite automata, equivalence of NFAs and DFAs, generalized non-deterministic finite automata, pumping lemma for regular expressions, push down automata, context free grammars
9	MID TERM IN WEEK 9	
10-17	Automata-II	Equivalence of PDAs and CFGs, Designing CFGs according to the language demanded by scenario, pumping lemma for CFLs, limitations of PDAs, Turing machine decidability and infinite search space
18	FINAL TERM IN WEEK 18	