

Course Title: Automata and Complexity Theory**Course Code: CoSc 3101****Credit Hrs.: 3 Hrs.****Prerequisite: None****Course Category: Compulsory****Year III Semester I**

Course Description This course aims to develop the theoretical foundations of computer science through study of mathematical and abstract models of computers and the theory of formal languages. It also, introduces some fundamental concepts in automata theory and formal languages including grammar, finite automaton, regular expression, formal language, push down automaton, and Turing machine. The importance of time and space complexities, various notations and theorems of Complexity theory.

Course Objectives By the end of this course, students will be able to:

On completion of this course students should be able to:

- introduce concepts in automata theory and theory of computation
- study the central concepts of automata theory
- acquire insights into the relationship among formal languages, formal grammars, and automata.
- identify different formal language classes and their relationships
- design grammars and recognizer for different formal languages
- explain Models of Computation, resources (time and space), algorithms, computability, and complexity.
- understand Complexity classes, P/NP/PSPACE, reductions, hardness, completeness, hierarchy, relationships between complexity classes.
- learn Randomized computation and complexity; Logical characterizations, incompleteness and approximately.

Course Outline**Chapter 1: Introduction to Basic Concepts(4hrs)**

- 1.1. Alphabets and strings
- 1.2 Languages and Grammars
- 1.3. Automata
 - 1.3.1. Finite automata, Deterministic and Non-deterministic finite automata

Chapter 2: Regular Expression and Regular languages (4hrs)

- 2.1. Regular expressions
- 2.2. Connection between regular expression and regular languages
- 2.3. Regular grammar
- 2.4. Pumping lemma and non-regular language grammars

Chapter 3: Context free languages (6hrs)

- 3.1. Context free grammars
- 3.2. Sentential forms
- 3.3. Derivation tree or parse tree
 - 3.3.1. Left most and right most derivations

- 3.4. Parsing and ambiguity
- 3.5. Simplification of context free grammar
 - 3.5.1. Methods for transforming grammars
 - 3.5.2. Chomsky's hierarchy of grammars

Chapter 4. Push down automata * (4hrs)

- 4.1. Non-deterministic pushdown automata
- 4.2. Push down automata and context free languages
- 4.3. Deterministic push down automata
- 4.4. Deterministic context free languages

Chapter 5: Turing machines (4hrs)

- 5.1. Standard TM
- 5.2. Construction of TM
- 5.3. Turing Decidable and Turing Acceptable
- 5.4. Undecidable problems

Chapter 6: Computability (4hrs)

- 6.1. Recursive functions
- 6.2. Recursive languages and recursive Enumerable languages

Chapter 7: Computational complexity (6hrs)

- 7.1. Big-O notations
- 7.2. Class P vs class NP
- 7.3. Polynomial time reduction and NP-complete problems
- 7.4. Cook's Theorem

Teaching - Learning methods: Three contact hours of lectures. Students do home assignments.

Assessment method:

Quizzes 20%, Assignments 30%, Test 10%, Final Exam 40%. Total 100%

Teaching materials Required Text: An Introduction to Formal Languages and Automata, Third Edition, Peter Linz, 2001

Reference books:

- Introduction to Automata Theory, Languages, and Computation by Hopcroft, Ullman and Motwani
- An Introduction to Formal Language Theory that Integrates Experimentation and Proof Allen Stoughton, 2004.
- Complexity Theory: A Modern Approach Sanjeev Arora and Boaz Bara