10B11CI513: Theory of Computation

Course Credit: 4
Semester: V

Introduction

- Theory of Computation (TOC) course is to introduce the fundamental mathematical and computational principles that are the foundation of computer science. These include Topics such as Turing machines, Automata, grammars and formal languages, decidability, halting problem, the P = NP question and NP-Completeness reductions.
- A secondary objective is to address students' misconceptions that Theory of Computation is disconnected from state-of-the-art applications and today's open problems; once they complete a theory course, however, they understand that the fundamental theorems and proofs of theory provide the foundation material for all specialties of computer science. For example, TOC introduces conceptual tools that practitioner's use in compiler and programming language.

- A final and related objective of a TOC course is to prepare students to be either well-rounded practitioners or potential candidates for computer science graduate programs. Determining whether a TOC course prepares students for graduate school is straightforward: approximately 1/3 of the problems on the GATE test for computer science require an understanding of TOC.

Course Objectives (Post-conditions)

Knowledge objectives:

1. You will broaden your knowledge of the fundamental mathematical and computational principles that are the foundation of computer science
2. To understand the concept of Deterministic Finite Automata and Non-Deterministic Finite Automata

3. To understand how to minimize the states,
usage Moore and Mealy Machine
4. To understand how to use the context free grammars in languages and how to derive parse trees and solve ambiguity problems
5. To understand Normal forms for Context Free Grammar's Chomsky and Greibach Normal Forms
6. To understand the Push Down Automaton algorithm
7. To understand how the push down automata will accept arbitrary context free languages.
To understand the properties of CFG
To understand the determinism and parsing.
To understand different parsing methodologies
8. To understand the basic concepts of Turing Machine.

To understand the configuration of Turing Machine.
To understand the computing with the Turing Machine.
9.To understand multiple tapes, two way infinite tape concepts.
- To understand the real computers random access memories working.
- To understand the concept of non-deterministic Turing machines.

10.To understand the computational power of languages.
- To understand numerical functions applied to Turing machines.
- To understand numerical functions applied to Turing machines.
- To understand various mathematical models applied to Turing machines.
- To understand the concept of halting problem.
- To understand undecidable problems about Turing machines and grammars.
- To understand the properties of recursive languages.
- To understand the concept of polynomial decidable.

11. To understand Boolean satisfiability

Application objectives:
Design a compiler from scratch using concepts taught in the class.

Expected Student Background (Preconditions)

Graduate level courses in algorithms and complexity of algorithms are desirable.

Topics Outline:

S NO	Topics	Hrs
1	Introduction &Motivation, Infinite Sets, Closures, Alphabets, Languages & Representation	4
2	Deterministic finite automata, Non-Deterministic finite automata, Closure Properties & Equivalences, Regularity, State Minimization, Moore and Mealy Machine	7
3	Context Free Grammars, Parse Trees & Ambiguity, Chomsky and Greibach Normal forms, Push Down Automata, Equivalence of PDA and CFG, Properties of context free languages, Determinism & Parsing DCFG, Top-down & Bottom-up Parsing	10
4	Turing Machine-Introduction, Notations, Recursive and Recursively Enumerable Language, Extensions of Turing machines, Non-deterministic Turing machines, Primitive Recursive Functions, Mu-recursive functions	10
5	Church-Turing Thesis & Universal Turing machines, Halting problem, Undecidable problems, Properties of Recursive languages	5
6	The Complexity Class P, Satisfiability, The Complexity Class NP, NP Completeness and	7

	Reducibility NP complete problems, Cook's Theorem, NP Complete Problems	
	TOTAL	43

References

1. Elements of the Theory of Computation. Harry Lewis, Christos Papadimitriou, Second Edition, Pearson Education,1998.

2. Introduction to Automata Theory, Languages and Computation. John Hopcroft, Rajeev Motwani, and Jeffrey Ullman. Second Edition. Pearson Education, 2001.

3. Formal Languages and Automata- Peter Linz , Narosa Pub. 4th edn.

4. Theory of computer Science: Automata, Language and Computation - KLP Mishra N Chandra Sekhran - PHI, 3rd edn.

Evaluation Scheme:

S.No	Examination	Marks
1	T-1	15
2	T-2	25
3	T-3	35
4	*Internal Marks	25

*Internal Marks Breakdown:

Assignments 9 marks (3x3)

Quizzes 12 marks (3x4)

Regularity 4 Marks