Theory of Computation [CSE-331]

Programme: B.Tech. (CSE)	Year: 3rd	Semester: 6th
Course: Core	Credits: 03	Hours: 40

Course Context and Overview (100 words):

The objective of this course is to introduce students to this fundamental area of computer science which enables students to focus on the study of abstract models of computation. These abstract models allow the students to assess via formal reasoning what could be achieved through computing when they are using it to solve problems in science and engineering. The course exposes students to the computability theory, as well as to the complexity theory. The goal is to allow them to answer fundamental questions about problems, such as whether they can or not be computed, and if they can, how efficiently.

Two fundamental questions about any problem are:

- Can it be solved using a given abstract machine? (Computability)
- How much time and space are required to solve it? (Complexity).

We explore these questions by developing abstract models of computing machines and reasoning about what they can and cannot compute efficiently.

Prerequisites Courses:

A. **Design & Analysis of Algorithms,**

B. **DMS**

Course outcomes (COs):

On completion of this course, the students will have the ability to:
CO1: To explain the basic concepts of deterministic and non-deterministic finite automata, regular language, context-free language, Turing machines, Church's thesis, halting problem, computability and complexity.
C02: To construct pushdown automata and the equivalent context free grammars.
C03: To recognize the power and limitation of a computer.
C04: To construct Turing machines and Post machines, and prove the equivalence of languages described by Turing machines and Post machines.
CO5: To elaborate with arguments about what problem can be solved by developing abstract models of computing machines and how much time and space are required to solve it.

Course Topics:

Topics	**Lecture Hours**	
UNIT - I 1. **Topic**		

1.1 Finite state Automata - Non deterministic and deterministic FSA, NFSA with ε- moves.	03	10
1.2 Regular Expressions - Equivalence of regular expression and FSA. Pumping lemma, closure properties and decidability	04	
1.3 Myhill - Nerode theorem and minimization Grammars - Production systems - Right linear grammar and Finite state automata.	03	
UNIT - II **2. Topic**		10
1. Pushdown automata - Acceptance by empty store and final state - Equivalence between pushdown automata and context-free grammars.	04	
2. Deterministic pushdown automata. Context-Free Grammar (CFG) – Parse Trees – Ambiguity in grammars and languages	03	
2.3 Normal forms for CFG – Pumping Lemma for CFL – Closure Properties of CFL & DCFL.	03	
UNIT - III **3. Topic**		10
1. Turing Machines - Techniques for Turing machine construction - Generalized and restricted versions equivalent to the basic model	04	
2. Universal Turing Machine - Recursively enumerable sets and recursive sets - Computable functions	03	
3. Time space complexity measures - context sensitive languages and linear bound automata.	03	
UNIT - IV **4. Topic**		10
1. Decidability; Post's correspondence problem; Rice's theorem; decidability of membership, emptiness and equivalence problems of languages.	03	
2. Time and tape complexity measures of Turing machines; Random access machines; the classes P and NP	03	
4.3 NP-Completeness; satisfiability and Cook's theorem; Polynomial reduction and some NP-complete problems.	04	

Textbook references (IEEE format):

Reference Books:

1. Introduction to Automata Theory, Languages and Computation by JE Hopcroft, R Motwani, JD Ullman
2. Elements of the Theory of Computation by Harry R. Lewis and Christos H. Papadimitriou
3. An Introduction to Formal Languages and Automata by Peter Linz.

Additional Resources (NPTEL, MIT Video Lectures, Web resources etc.):

Evaluation Methods:

Item	Weightage
Quiz1	35%
Quiz2	
Assignments (~3)	
Midterm	25%
Final Examination	40%

Prepared By: M. TANVEER
Last Update: 13/04/2015

Updated By: Sakthi Balan M
Updated on: 28/07/2017