THEORY OF COMPUTATION

PROF. SUBRAHMANYAM KALYANASUNDARAM
Department of Computer Science Engineering
IIT Hyderabad

PRE-REQUISITES : Discrete Mathematics should have been completed. It is also desirable (not mandatory) that the students have done/are doing in parallel the Design and Analysis of Algorithms course.

INTENDED AUDIENCE : This is a core course BTech students in the Computer Science stream.

INDUSTRY SUPPORT : As this is a core BTech CS course, this is important for any industry that requires a BTech student who is strong in CS fundamentals

COURSE OUTLINE :
This course is an introduction to the theory of computation. We start with computability --- different models of computations and see what are the capabilities of each of these models. Each of these models try to model a computer. We see the classes of languages each of these models are capable of computing. After computability, we will see an introduction to complexity, where we classify computable languages on the basis of easiness/difficulty of computation. The student should be comfortable with basic proof techniques.

ABOUT INSTRUCTOR :
Prof. Subrahmanyam Kalyanasundaram did his Masters from Dept of ECE in IISc. After which, he did his Masters in Mathematics and Phd in Algorithms, Combinatorics and Optimization from Georgia Tech. He has been at the Department of Computer Science and Engineering, IIT Hyderabad since 2011, and is presently an Associate Professor. He is interested in almost all areas of theoretical computer science, and has most recently been working on combinatorics, graph theory and graph algorithms.

COURSE PLAN :
Week 1: Introduction to the course, DFAs, Regular Languages, NFAs, Equivalence of DFAs and NFAs.
Week 2: Closure properties, regular expressions, Pumping Lemma for regular languages
Week 3: Myhill-Nerode Theorem, DFA Minimization
Week 4: Context free grammar, Chomsky Normal Form, CYK Algorithm
Week 5: Closure properties of CFL, Pushdown Automata, Equivalence
Week 6: Pumping Lemma for CFLs. Turing Machines, Decidable (recursive) languages, Turing-Recognizable (recursively enumerable) languages.
Week 7: (i) Equivalence of NTM and DTM, Church Turing thesis, Algorithms, Decidable languages from regular and context-free languages
(ii) Halting Problem and undecidability. Reductions and other undecidable languages.
Week 8: Post Correspondence Problem (PCP) is undecidable, Rice’s theorem.
Week 9: Introduction to Complexity Theory. Asymptotic notation, Classes P and NP, Verifier model for NP.
Week 10: NP Completeness, Polynomial Time reductions, Cook-Levin Theorem
Week 11: NP Complete problems like Vertex Cover, Hamiltonian Path, Subset Sum, ILP
Week 12: Space Complexity, Relation with time bounded complexity classes, introduction to classes like L, NL, PSPACE and overview of results in space complexity