Course Code	Course Title
COMP 3721	**Theory of Computation**

Course Description

This course is an introduction to the foundation of computation, and aims at answering some of the most fundamental questions in computer science: What is an algorithm? What can and cannot be computed at all? What can and cannot be computed efficiently? The topics covered include set theory and countability, formal languages, finite automata and regular languages, pushdown automata and context-free languages, Turing machines, undecidability, P and NP, NP-completeness.

List of Topics

0. Introduction to course
1. Sets, Relations, and Functions
2. Languages and Regular Expressions
3. Countability and uncountability
4. Deterministic Finite Automata
5. Nondeterministic Finite Automata
6. DFA = NFA = regular expression
7. Properties of Regular Languages
8. Pumping Theorem for R. L.
9. Context-free languages
10. Pushdown automata
11. PA = Context-free grammars
12. Closure Properties of CFLs
13. Pumping Theorem for CFLs
14. Turing Machines
15. Computing with TM
16. TM Extensions and Church-Turing Thesis
17. Closure properties of Recursive L
18. Universal Turing Machines
19. The Halting Problem
20. Undecidable Problems
21. P and NP
22. NP-Completeness
23. Approximation Algorithms

Textbooks

- Elements of the Theory of Computation, Second Edition, by Harry R. Lewis and Christos H. Papadimitriou, Prentice-Hall International Inc., 1999.

Reference books

- Theory of Computing: A Gentle Introduction, by Efim Kinber, Carl Smith, Prentice Hall, 2001.
- Introduction to the Theory of Computation, Second Edition, by Michael Sipser, PWS Publishing Company, 1996.

- Introduction to Automata Theory, Languages, and Computation. by John E. Hopcroft and Jeffrey D. Ullman, Addison-Wesley, 2001

Grading Scheme

- Assignments: 5% x 4 = 20%
- Midterm: 30%
- Final: 50%

Course Intended Learning Outcomes

On successful completion of this course, students are expected to be able to:

1. Understand the equivalence of decision problems and language recognition problems and understand the limitations of different classes of computational machine (finite automata, pushdown automata, Turing machine) in recognizing different classes of language.
2. Prove the equivalence of DFA, NFA, regular expressions and the equivalence of PDA and CFG.
3. Design finite automata and write regular expressions for regular languages.
4. Write context-free grammar (CFG) for context-free languages, especially for expressions occurring in programming languages.
5. Design pushdown automata for context-free languages. Given a context-free grammar, construct a PDA that accepts the language generated by the given grammar.
6. Given a non-deterministic finite automaton (NFA), convert it to a deterministic finite automaton (DFA).
7. Apply pumping theorem to prove that a language is not regular or not context free.
8. Understand the Church-Turing thesis and the unsolvability of the halting problem and apply reduction technique to prove the undecidability of decision problems, in particular language recognition problems.
9. Understand the concepts of P, NP, and NP-completeness and use reductions to prove that a given problem is NP-complete.