Theory of Computation

Introduction

From high level

- From a high level what this course is about?
  - Given a set, say S.
  - Let $A \subseteq S$.
  - Both S and A are well defined.
  - We are given an element $x \in S$, and asked to find whether this x is in A or not.
  - That's all !

Surprises !!

- Surprising things
  - This is related to decision problems.
    - S is set all face images. A is set of images of a particular person. {Face verification}
    - S is set of all graphs. A is set of graphs with Hamiltonian cycle.
  - Sometimes this is an unsolvable problem. ??
  - Sometimes this is an easy task, sometimes quite a difficult one.
    - Recall, $O(n^2)$ is time consuming than $O(n)$ algorithm.

We want general enough set

- Set of strings over some alphabet like {0,1}.
- For example set of strings that end with a 0,
  {0, 00, 10, 000, 010, 100, 110, …}
- Each string in the set can be seen as a positive binary number and let the set be the set of prime numbers.
  - Given a number (binary string of 0s and 1s) you want to find whether this is in the set (prime) or not (not a prime).

Why strings are chosen?

- Any data element like number or image or any thing can be represented as a string.
  - Can we say DNA code represents a human being?
- Even a method which solves a problem can be represented as a string.
- So strings over an alphabet gives us power to represent the things… that is we have *languages* of strings to represent the things.

What will you learn from this course?

- How to define a computer? **Automata theory**
- Are there problems that a computer cannot solve? If so, can we find one such problem? **Computability theory**
- For problems that a computer can solve, some problems are easy (e.g., sorting) and some are difficult (e.g., time-table scheduling). Any systematic way to classify problems? **Complexity theory**

Text Books

Introduction to Automata Theory, Languages, and Computation, by J. Hopcroft, R. Motwani, and J. Ullman.

Introduction to the Theory of Computation (2nd Edition), by Michael Sipser

Reference

Computational Complexity, by C. Papadimitiou

Evaluation

- Quiz/Assign: 20 marks
- Mid1: 20 marks
- Mid2: 25 marks
- Endsem: 35 marks