Theory of Computation (CS F351)

Introduction.

BITS Pilani
Hyderabad Campus

Prof.R.Gururaj
CS&IS Dept.

Every discipline has its basic solid foundation.
Similarly for Computer Science the fundamental questions are-

- What is a computer?
- What can it do?

We study the concepts, models pertaining to Computers.
As part of this course we study fundamental properties of computers.
The ideas and models we examine are more mathematical in nature.
It is all about the theory behind the computations done by computers.

Want to know what this course is about?

The Theory of computation (Automata Theory) examines the questions-

- What is a computer?
- What can it do?
- How Computer languages are represented and recognized?

Automata Theory: Study of abstract devices.

The theory behind the Models represent computers.

Automaton is a machine that responds to encoded instructions.

Theory of Computation:

What can be computed; what a computer can compute what not etc.

Computation is simply a sequence of steps that can be performed by computer.

We work with abstract machines or models that represent computers.

Some of them are very basic and others are as powerful as real-world computers.

These simple machines allow us to introduce formalisms used in theory.

What is the common theory with today's computers?

Can we say anything about what computers can or cannot do that will still be true thirty years from now?

Purpose of this Course

We'll look at simplified, mathematical models for different kinds of computers.

What is an algorithm.

What can be computed and what cannot be.

Data in computers is stored as strings.
We study models that manipulate strings (Automata).
(Automaton is a machine that responds to encoded instructions)
How to represent languages (grammar).

We look at simplified, mathematical models for different kinds of computers.

They vary by how much memory they have and how they can access it.

What Models we study?

A *finite state machine* – This is a restricted model of an actual computer. Helps in recognizing languages (Regular Languages).

A *pushdown automaton* has an infinite amount of memory, but it can only access it by pushing an value onto its stack or popping a value off of the stack, and it only has one stack.

A *Turing machine (1930)* has an infinite amount of memory, organized as a long tape with a head that can scan back and forth on it. (named after Alan M. Turing and English mathematician)

Why these machines

- These models are chosen because they have simple descriptions.
- And for each kind of machine we can show that there are clearly defined classes of things that they can and cannot do.

Anything the biggest, fastest computer in the world can do, can be done by a Turing machine.

And there are things that a Turing machine cannot do.

This means that there are problems that no computer can solve!

These are known as "*undecidable*" problems.

Some problems have solutions in theory. But are not practical due to excessive time requirements.

We study TOC in mathematical way and specifically we deal with concepts related to –

- Sets,
- Relations and
- Functions.

What we gain from this course

innovate achieve lead

1. Understand abstract models that represent computers.
2. To have an understanding about the capabilities and limitations of a computer.
3. Assess the solvability of problems, with the help of theoretical principles we learn.
4. Theory gives you simpler and elegant side of computers. This helps in building more beautiful systems.
5. Developing new SW and know what they can do.
6. Studying this would expand your mind and train you in problem solving.
7. We gain knowledge to design a new programming languages.
8. Helps Designing compilers for computer languages.