4CS4-6: Theory of Computation (Objectives, Scope and Outcomes)

Lecture 01: Jan. 08, 2020

Prof. K.R. Chowdhary *: Professor of CS*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

1.1 Introduction

Computer science has emerged as a new field of science as well as engineering, with host of areas, like - data structures, operating systems, networking, artificial intelligence, and number of languages. The subject of theory of computation attempts to establish a theory to account for all these phenomena in a similar way that classical dynamics attempts to explain the motion of objects. In contrast to the technical and applied areas of computer science, theoretical computer science is strongly aimed to fundamental questions about the existence of algorithmic solutions, physical limits of computing, methodologies of algorithm design, etc.

Let us first attempt to answer the question - "what is computer science?" Usually, it is difficult to provide an exact and complete definition of a scientific discipline. A commonly accepted definition is: computer science is the science of *algorithmic processing, representation, storage and transmission of information.* This definition presents *information* and *algorithm* as the main objects investigated in computer science. However, it neglects to properly reveal the nature and methodology of computer science.

Another question regarding the substance of computer science is - to which scientific discipline does computer science belong? Is it *meta-science* such as mathematics and philosophy, a science, or Engineering? The answer is that computer science cannot be uniquely assumed to any of these disciplines. Computer science includes aspects of mathematics, and natural sciences as well as Engineering. Similar to philosophy and mathematics, computer science investigates general categories such as: *determinism, non-determinism, randomness, information, truth, untruth, complexity, language, proofs, communication, approximation, algorithms, simulation*, etc., and attributes to the understanding of these categories.

Computers are designed for processing information, which can be as simple as computing the distance between two cities, and sometimes as complex as weather prediction. The study of the subject of theory of computation provides an insight into the characteristics of these computations. Such insight can be aimed to: (i) predict the complexity of desired computations, (ii) choosing the appropriate approach of solution, and (iii) developing tools that facilitate the design. Study of computations can reveal that there are problems that cannot be solved, as well as that there are problems, which requires arbitrarily large amount of CPU (Central Processing Unit) time, say millions of years, and may require infinitely large memory. These revelations, though sounds discouraging, but can act as useful warnings against such problems.

1.2 Significance of Theory

Any *scientific theory* is primarily concerned with representing the real world entities in the form of abstract mathematics. The representation should facilitate the discovery of mathematical relationships in the form of equations or laws. Thus, in the case of classical dynamics the real-world entities include the distance, speed, force, and acceleration. The distance is usually represented as a function of time, and speed as derivative of distance function. The basic mathematical relations governing them are the Newton's laws of motion, and the laws of gravitation.

Every theory increases our understanding of, and systematizes our knowledge about the subject. Theory also leads to advances of practical. It is the second advantage, which is of immediate use to us. The theory of computer, which we will study, will be of immediate use in writing of programs to construct compliers, assemblers, operating systems, etc., as well as understanding of existing high-level languages and construction of new languages.

To each theory, there is corresponding *meta-theory*, which is concerned with analyzing theory itself. For example, precisely defining the concepts used in the theory and in discovering the limitations of the theory. The meta-theory for the theory of computer science is mathematics, which formalizes the notion of sets, proofs, and theorems, and attempt to discover their limitations. Typically, a theory is concerned with establishing useful positive results, where as the meta-theory consists largely of results often of a negative nature, which define and delimit the subject matter of the theory.

Two major goals of meta-computer science are:

1. to define the term computable precisely, and
2. given such a definition, to discover what (if any) are the theoretical limitations of computability, thus delimiting the subject matter and scope of the computer science.

A computer scientist has an intuitive grasp of notion of computable in the form of an *algorithm* or *program*. To be precise, for a computer scientist, the most obvious way to define something as computable is to represent it in the form of a program written in a typical programming language. Such a suggestion is not valid for a meta-computer scientist, because for that the solution should be sufficiently simple so that it can be verified with the help of mathematical proofs.

Along the same lines of discussions, following quote is important:

> Law, like engineering, changes fast. The so-called "practical" facts soon become obsolete. The only knowledge of permanent valeue – in law as elsewhere – is theoretical knowledge. Theoretical knowledge, critical judgment and the discipline of learning are the only aspect of legal education which makes the individual readily adaptable to changing problems.
>
> – William O. Douglass

The quote by Douglas is an indication of the importance of theoretical knowledge to support his view.