Introduction to Theory of Computation

Lecture 1

Strathmore
UNIVERSITY

Lecture Outline

- By the end of this chapter, the student should be able to:
  - Define Theory of Computing
  - Describe importance of studying Theory of Computing
  - Differentiate areas of Theory of Computing
  - Describe historical sources of Theory of Computing

Strathmore UNIVERSITY

Computer Technology Today

Computer Technology Today

Strathmore UNIVERSITY

- Computers are changing our world and the way people think
  - Software, hardware is improving, run-time speeds, memory capacity increasing
- **New models** appear on the market almost every month
- Some televisions are taking on the functionality of what we call computers, blurring the distinction of what is and what is not a computer

Computer Technology Today (Cont)

Strathmore UNIVERSITY

- *How do you decide which computer to buy* when the sales people themselves often don't really understand the differences between the machines?
- Sales people usually characterize the machines based on what software they run or confuse the consumer with buzzwords like 500 megahertz, 16-meg RAM, multipurpose ports, and so on.

Theory of Computer Technology

Strathmore UNIVERSITY

- While *model* numbers, *buzzwords* and software change, the **theory is fixed**
- Theory can be used to recognize computation (or computing) in all its **forms** and **disguises**
- For example, the ubiquitous **vending machines** that dispense products from canned soft drinks and juices, to snacks are all really a type of computer called an **automaton**

Theory of Computer Technology (Cont)

Strathmore UNIVERSITY

- This same type of computer is at the heart of the **search engines** such as Google search, Bing, *Duckduckgo, Baidu*, Wikipedia, Yandex among others
- It is also the same type of computer which won the game of *chess in 1997* against the grandmaster Garry Kasparov, *AlphaGo* (2016 against Lee Sedol) and *Alphastar* (winning Starcraft in 2019)

Strathmore UNIVERSITY

WHAT IS COMPUTING?

Computing

Strathmore UNIVERSITY

1. *Computing* is the movement, and ***alteration*** during transit, of data
2. *Computing* is a type of ***calculation*** or use of computers in information processing
3. *Computing* is a **process** following a well defined *model* and expressed as an algorithm

Computing: Example I

Strathmore UNIVERSITY

- Consider a **vending machine** dispensing soft drinks.
- The input to the computing, the original data, is the coins you put in the slot and buttons(s) you press to select the product
- The coins are moved to a holding area and some electronic electronic impulses are generated
- The coins are **altered** into electronic impulses that moves through *wires* or *circuits*

Computing: Example I (Cont)

Strathmore UNIVERSITY

- The data now contains information about how many coins you entered and which denominations
- These impulses are **transformed** into signals that trigger the gate that lets the product you selected move to the area of the machine where you can collect it

Computing: Example I(cont)

Strathmore UNIVERSITY

- So, the **impulses** encoding your selection and the coins entered are changed into **a product**, and perhaps some change in another area of the machine
- This is **computing**

Computing : Example II

Strathmore UNIVERSITY

- When you use an automatic teller machine (ATM), another sequence of events happens that looks like computing
- You insert your card and then use a keypad to input more data
- Sometimes the keypad is organized around a display and you must make several selections
- As the display changes, so does the meaning of each button

Computing: Example II (Cont)

Strathmore UNIVERSITY

- Nonetheless, you are still inputting data to the ATM machine
- After a while, you get some cash as output and maybe a transaction receipt as well
- So, according to our working definition, **computing** has taken place

Computing: Example II (Cont)

Strathmore UNIVERSITY

- A more detailed analysis reveals even more computing
- After you have entered all the data, such as pin codes and transaction type, the bank machine *contacts another computer* holding your account information

Computing: Examples II (Cont)

Strathmore UNIVERSITY

- This connection may be complicated depending on how far away the bank machine you are using is from the location of the computer holding the needed information
- It is not unusual for such connections to cross state and even country borders (when not hosted locally)

Computing: Example III (Cont)

Strathmore UNIVERSITY

- The computer with the account information does what is typically considered a computing and sends some information, like an authorization number, back to the bank machine (ATM) you are using
- Few data are being transferred between the computer (remote) and the bank machine, but the *amount of withdrawal* and *account number* information is ***transformed*** into an authorization to dispense cash

Computing: Example III

Strathmore UNIVERSITY

- Another computing that has layers and layers happens every time you use a computer i.e desktop, mobile phone or a laptop
- At the outermost level, the keystrokes, touchscreen or mouse movements are transformed into the **display** you see on device's screen
- At the next level, the *keystrokes activate* some program which runs on the computer

Computing: Example III (Cont)

Strathmore UNIVERSITY

- One level deeper, the program is instructing the computer to get data from memory and move them into the processor where they are transformed into commands for the display
- This is **computing** and it exists in various forms

Strathmore UNIVERSITY

THEORY OF COMPUTING

Historical Sources of Theory of Computing

Strathmore UNIVERSITY

- Works performed by mathematicians such as Leibnitz and Pascal, trying to ***mechanize calculus and logical deduction***.
- Work performed by logicians such as Kurt Gödel, Alfred Tarsky, ***Alan Turing***, ***Alonzo Church***, ***A. A. Markov***, toward the development of a **formal concept of algorithm**

Historical Sources of Theory of Computing (Cont)

Strathmore UNIVERSITY

- **Alan Turing** is regarded as the father of computer [science]
- All computers we use today are **TURING COMPLETE**
- *Turing completes* is an ability of a machine to simulate any **turing machine**
- A ***Turing machine*** is a hypothetical machine thought of by the mathematician Alan Turing in 1936.
- Despite its simplicity, the machine can ***simulate ANY computer algorithm***, no matter how **complicated** it is!

Historical Sources of Theory of Computing (Cont)

Strathmore UNIVERSITY

- **Alonzo Church** designed **Lambda calculus**
- Lambda calculus is a model of computation, invented by Church in the early 1930's.
- Lambda calculus involves studying computation with **functions** giving definition to what is computable
- Lambda calculus and Turing machines are ***equivalent***, in the sense that any function that can be defined using one can be defined using the other

Historical Sources of Theory of Computing (Cont)

Strathmore UNIVERSITY

- Lambda calculus formed the basis for **functional** programming languages - involves binding everything in pure mathematical functions style and the focus is on **WHAT** we want to achieve) such as **LISP, Haskell, Scala**
- Turing machines formed the basis of **imperative** programming languages - involves specification of **HOW** the solution is to be achieved by the programmer) such as **Java, C#, Go, C**

Theory of Computing

Strathmore UNIVERSITY

- The revolutionary impact of a technology (Computing Technology) on our society does not necessarily facilitate the appreciation of the **intellectual contents** of the theory underlying it (Theory of Computing)
- Typically, *people are so overwhelmed by the wonders of the technology that they fail to wonder about the* ***theory underlying*** *it.*

Theory of Computing (Cont)

Strathmore UNIVERSITY

- Specifically, people tend not to think of computing in ***general terms*** but rather in the ***concrete terms*** in which they have lastly encountered it.
- Consequently, the intellectual contents of the Theory of Computing is rarely communicated and rarely understood (by non-specialists)

What is Theory of Computing

Strathmore UNIVERSITY

1. A branch of computer science or mathematics that deals with **whether** or **how** *efficiently* problems can be solved on *a* **model of computing** (an algorithm)
2. Theory of computing is the study of **efficient** computation through models of **computational processes** and their *limits*.

Scope of Theory of Computation

Strathmore UNIVERSITY

- Three traditionally central areas of the theory of computation are
  - Automata,
  - Computability, and
  - Complexity
- The above central areas are linked by a key fundamental question - ***"What are the fundamental capabilities and limitations of computers?"***

Scope of Theory of Computation (Cont)

Strathmore UNIVERSITY

- This question goes back to the 1930s when mathematical logicians first began to explore the meaning of computation
- Technological advances since that time have greatly increased our ability to compute and have brought this question out of the **realm of theory** into the world of **practical concern**

Scope of Theory of Computation (Cont)

Strathmore UNIVERSITY

- In each of the three areas—automata, computability, and complexity—the stated question is interpreted differently, and the answers vary according to the interpretation

Strathmore UNIVERSITY

COMPLEXITY THEORY

Complexity Theory

Strathmore UNIVERSITY

- Computer problems come in different varieties; some are **easy**, and some are **hard**
- For example, the **sorting** problem is an easy one
- Say that you need to arrange a list of numbers in ascending order
- Even a **small computer** can sort a million numbers rather quickly

Complexity Theory (Cont)

Strathmore UNIVERSITY

- Compare **sorting** problem and a **scheduling problem**.
- Say that you must find a schedule of classes for the entire university to satisfy some reasonable constraints, such as that no two classes take place in the same room at the same time.
- The scheduling problem seems to be much harder than the sorting problem.
- How do we know that ***SORTING*** is easier than ***SCHEDULING?***

Complexity Theory (Cont)

Strathmore UNIVERSITY

- If you have just a thousand classes, finding the **BEST** schedule may require centuries, even with a supercomputer

"What makes some problems computationally hard and others easy?"

- This is the central question of complexity theory
- *We don't know the answer* to that question, though it has been intensively researched for decades now

Complexity Theory (Cont)

Strathmore UNIVERSITY

- Researchers have discovered an elegant scheme for ***classifying*** problems according to their ***computational difficulty.***
- It is analogous to the *periodic table for classifying elements according to their chemical properties.*
- Using this scheme, we can show by giving **evidence** (in terms of **TIME** & **SPACE**) that certain problems are computationally hard, even if we are **unable to prove** that they are

Complexity Theory (Cont)

Strathmore UNIVERSITY

- One applied area that has been affected directly by complexity theory is the ***ancient field of cryptography***
- In most fields, an *easy computational problem* is preferable to a hard one because easy ones are cheaper to solve
- Cryptography is unusual because it specifically requires *computational problems that are **hard**, rather than **easy***

Complexity Theory (Cont)

Strathmore UNIVERSITY

- ***Secret codes*** should be hard to break without the secret key or password.
- Complexity theory has pointed cryptographers in the direction of computationally hard problems around which they have designed revolutionary new codes

Strathmore UNIVERSITY

COMPUTABILITY THEORY

Computability Theory

Strathmore UNIVERSITY

- Certain basic problems cannot be solved by computers.
- One example of this phenomenon is the problem of determining whether a mathematical statement is true or false
- This task is easy for mathematicians
- It seems like a natural solution by computer because it lies strictly within the realm of mathematics - ***But no computer algorithm can perform this task.***

Computability Theory

Strathmore UNIVERSITY

- The theories of computability and complexity are closely related.
- In **complexity theory,** the objective is to classify problems as easy ones and hard ones; whereas in **computability theory,** the classification of problems is by those that are solvable and those that are not.
- Computability theory introduces several of the concepts used in complexity theory

Strathmore UNIVERSITY

AUTOMATA THEORY

Automata Theory

Strathmore UNIVERSITY

- Automata theory deals with the definitions and properties of mathematical **models of computation**
- It deals with the logic of computation with respect to simple machines, referred to as **automata**.
- Through automata, computer scientists are able to understand how machines compute functions and solve problems and more importantly, what it means for a function to be defined as **computable**

Automata Theory (Cont)

Strathmore UNIVERSITY

- Automata plays key role in several applied areas of computer science
- One model, called the **finite automaton**, is used in
  - Text processing (Natural Language Processing)
  - Compilers and
  - Hardware design
- Another model, called the **Context-free Grammar,** is used in programming languages and artificial intelligence

Automata Theory (Cont)

Strathmore UNIVERSITY

- The theories of **computability** and **complexity** require *a precise definition of a computer.*
- Automata theory allows practice with **formal** definitions of computation

Strathmore UNIVERSITY

Why study Theory of Computing?

Why Theory of Computing

Strathmore UNIVERSITY

- Any scientific theory is primarily concerned with representing in the abstract realm of mathematics the real-world entities that comprise its subject matter.
- The representation should facilitate the **discovery of mathematical relationships** in the form of equations or laws

Why Theory of Computing (Cont)

Strathmore UNIVERSITY

- The relationships discovered within the theory can be **interpreted in terms of real-world entities** to predict or explain observed real-world phenomena
- Alternatively one may be concerned to extend the **range of applicability of the theory** to accommodate further observations
- Any theory **increases our understanding** of, and systematizes our knowledge about, the subject domain

Why Theory of Computing (Cont)

Strathmore UNIVERSITY

- But a more immediate reason for constructing theories derives from the fact that it is usually the case that the theory also **leads to significant practical advances –** So it is with the theory of computer science
- Theory of computing lays a **strong foundation for a lot of *abstract areas*** *of computer science*
- It also teaches us about the elementary ways in which a computer **can be made to think**

Why Theory of Computing (Cont)

Strathmore UNIVERSITY

- There is a great deal of work that was made possible in the area of **Natural Language Processing Processing** that involved building **Finite State** machines also known as **Finite State Automata**
- **Regular expressions** can be beautifully represented using **Non-deterministic Finite Automata**
- **Any algorithm** can be expressed in the form of a **finite state machine** and can serve as a really helpful visual representation of the same

Strathmore
UNIVERSITY

Questions