Introduction to Foundations of Computing

The basic theoretical framework of the computing approach to problem-solving
Information and information processing

• We do not have a good definition for information. Information is an intuitive (primitive) concept.
• Nonetheless, information processing is all around us (this is the information era!!!!)
• Indeed, the ingredients of an information processing event are easy to identify:
  – Input data
  – A processing method (usually an algorithm)
  – An output
The concept of problem

• What triggers an information processing event is usually a problem
  • The concept of problem is also intuitive. For our purposes, a problem is well-defined if:
    – It is formulated within a mathematical domain (the objects of the problem can be represented as mathematical objects), and
    – The question posed in the problem can be written as a logical predicate
Examples: well- and ill-formulated problems

• Well-formulated problem:
  – **Domain**: Finite lists of natural numbers and the set of all natural numbers
  – **Instance**: A list and a natural number
  – **Question**: Is the given number in the list?

• Ill-formulated:
  – **Domain**: All cities
  – **Instance**: Paris, Barcelona
  – **Question**: Is Paris more enjoyable than Barcelona?
Problem solving

• The problem solving activity has two main aspects:
  – **Formulation of the problem** (Domain, instance, question), and
  – **Design of method (also called procedure, algorithm, etc.)** for answering the question

• In experimental sciences problems are solved through experiments, in (applied) mathematics, through formulas and equations, etc.

• All these methods are, or include at their core, information processing events.
Computation

• Nowadays, **computation** refers to information processing events that can be represented mathematically. This includes from
  – Natural information processing: human thinking or biological information processes to
  – Artificial information processing: calculations, text processing, web searches, etc.

• An information processing event **cannot qualify as computation if it is not based in a model** that is:
  – Its abstract, formal description is well-understood, and
  – Can be expressed in an algorithm, protocol, network topology, etc.
A “human thinking” information processing event

What do you see?
Was it computation?

• The **input** for the “sunset image formation” phenomenon is **a set of pixels of different colors**

• The human brain **processes** the input and forms as an output: **a sunset**

• The model is unclear (there is more than one conjecture about how this happen) and therefore, this event is **natural information processing** but not (yet) a **computation**
Picture of a biological information processing event

The Central Dogma of Molecular Biology

- Replication: DNA duplicates
- Transcription: RNA synthesis
- Translation: Protein synthesis
Molecular biology information processing

• In this case, the input is a set of DNA sequences encoding a protein, the output, a protein. We do have a better idea of the events that form a protein:
  • the mRNA *transcribes* the information of the DNA and *carries* it to the ribosome, where
  • the information is *translated* into the actual components of the targeted protein

• The model leaves important gaps. So this *information processing event* is “closer” to be, but it is not yet a full-fledged computation
Example of an artificial information processing method

Given an item and a list of items

For each item in the list check to see if the given item matches it.

If it matches, return the location where it was found. End the calculation.

If the end of the list is reached return “the item is not in the list”

End the calculation
The **theory of computation** is a field of **Computer Science** that studies **models of computation**.

The field is divided into three major branches:

- **Formal models of computation**
- **Computability theory**, and
- **Complexity theory**

This division is not strict since both, **computability** and **complexity** theory **depend** on the formal model of computation.
But, what is a formal model?

• In general terms, a formal model is a mathematical description of a concept or phenomenon of interest.

• Based on an analysis of the relevant objects in a concept, or components of a phenomenon, a standard mathematical structure is defined for each object or component, and for their interactions.

• Not all details are captured in a formal model.
  • For simplicity, the model represents only the structures that are relevant for a particular study.
  • Different studies may require different models.
Main objects in the information processing phenomena

- Objects **representing** information (i.e. data)
- Objects **describing** information processing systems or mechanisms. These may be divided into objects describing systems for:
  - The **generation** of new information
  - The **recognition** of particular features in the information
Mathematical representation of objects in the information processing phenomena

• Mathematical objects for *representing information*
  – Alphabets, strings, languages
• Mathematical objects describing information processing systems for *the generation of new information*
  – String operations, regular algebras, grammars
• Mathematical objects describing information processing systems *for the recognition of particular features in the information*
  – Finite automata, pushdown automata, Turing machines
An important question:

• Given a well-formulated problem: *Is there an information processing method for solving it?*
  
  — This is arguably one of the oldest questions in all science and engineering branches.

• Given the capabilities of computer s (i.e. electronic information processing machines), in practice, the question is naturally interpreted as:

  *Is there a computation (i.e. computational method) for solving it?*
Computability

- In terms of information processing, the theory of computability attempts to answer the question of which problems admit a computational or algorithmic solution. This is, solution through an information processing method of those we have characterized as computation.

- A significant amount of well-formulated problems have been shown to have no algorithmic solution.
Estimating the complexity of computing a solution

• Some algorithmic solutions are too costly in terms of time (i.e. number of steps) or space (i.e. space needed for representing the information)

• The central problem in complexity theory is deciding whether a given problem has a computation that is “affordable” in terms of compute time and memory space
  – This is usually characterized as a time/space polynomial algorithm

• A famous and yet unsolved theoretical question is there is a P algorithm for every problem? (the so-called P=NP question)
CIIC 6005 Road Map

• **Background**: Logic and Mathematics elements
  – Review of set theory and Boolean logic
  – Overview of proof methods

• **Section 1**: Representation of information
  – Alphabets, strings and encodings
  – Languages and operations with languages
Section 2

• **Section 2**: The string generation problem
  – Regular operations
  – Regular expressions
  – Regular languages
  – Context-free grammars
  – Context free languages
  – Context-dependent grammars and languages
Section 3

- **Section 3:** The string recognition problem
  - Discrete systems
  - Modeling discrete systems as automata
  - Finite state automata
    - Recognition of regular languages
  - Pushdown automata
    - Recognition of context-free languages
  - Turing machines
    - Characterization of Turing-recognizable languages
Section 4

• **Section 4: Computability**
  – Decidable languages
  – The Halting problem
  – Undecidable languages
  – Existence of an infinite uncountable number of languages that are not Turing-recognizable
Section 5

- **Section 5: Complexity**
  - Measuring time complexity
  - Polynomial time reducibility
  - Complexity classes
  - Space complexity: Savitch’s theorem
  - The class PSPACE