CSE 211 (Theory of Computation)

Introduction

Dr. Muhammad Masroor Ali

Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

January 2024

Version: 1.1, Last modified: August 31, 2024

Automata, Computability, and Complexity

Sipser, 0.1, p-1

- Three traditionally central areas of the theory of computation:
  - automata,
  - computability, and
  - complexity.

- They are linked by the question:

What are the fundamental capabilities and limitations of computers?

- 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.
- Have brought this question out of the realm of theory into the world of practical concern.

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

- 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.

- Compare that to a scheduling problem.
- Say that you must find a schedule of classes for the entire university to satisfy some reasonable constraints.
- No two classes are to take place in the same room at the same time.

- The scheduling problem seems to be much harder than the sorting problem.
- If you have just a thousand classes, finding the best schedule may require centuries, even with a supercomputer.

Automata, Computability, and Complexity

Sipser, 0.1, p-2

What makes some problems computationally hard and others easy?

- This is the central question of complexity theory.
- Remarkably, we don't know the answer to it, though it has been intensively researched for over 40 years.
- Later, we explore this fascinating question and some of its ramifications.

- One important achievement of complexity theory thus far.
- 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 demonstrate a method for giving evidence that certain problems are computationally hard, even if we are unable to prove that they are.

Automata, Computability, and Complexity

Sipser, 0.1, p-2

- You have several options when you confront a problem that appears to be computationally hard.
- First, by understanding which aspect of the problem is at the root of the difficulty, you may be able to alter it so that the problem is more easily solvable.
- Second, you may be able to settle for less than a perfect solution to the problem.
- In certain cases, finding solutions that only approximate the perfect one is relatively easy.
- Third, some problems are hard only in the worst case situation, but easy most of the time.

- Depending on the application, you may be satisfied with a procedure that occasionally is slow but usually runs quickly.
- Finally, you may consider alternative types of computation, such as randomized computation, that can speed up certain tasks.

- 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.
- 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.

The RSA Algorithm

Cryptography and Network Security Principles and Practice, 7th Edition, William Stallings

- Developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT and first published in 1978.
- The Rivest-Shamir-Adleman (RSA) scheme has since that time reigned supreme as the most widely accepted and implemented general-purpose approach to public-key encryption.

- The public key is chosen using $n = pq$, where, p, q, two prime numbers are chosen privately.
- The private key is calculated using p and q.
- In order to break the algorithm, the attacker needs to factor n into its two prime factors.

Automata, Computability, and Complexity

Sipser, 0.1, p-3

- During the first half of the twentieth century, mathematicians such as Kurt Gödel, Alan Turing, and Alonzo Church discovered that 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 the bread and butter of mathematicians.
- It seems like a natural for solution by computer because it lies strictly within the realm of mathematics.
- But no computer algorithm can perform this task.

- Among the consequences of this profound result was the development of ideas concerning theoretical models of computers that eventually would help lead to the construction of actual computers.

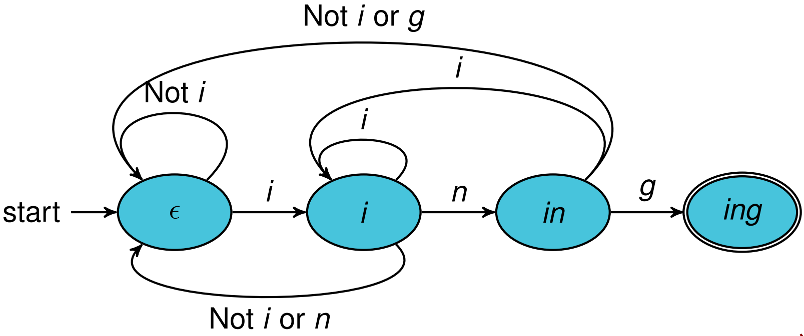
- The theories of computability and complexity are closely related.
- In complexity theory, the objective is to classify problems as easy ones and hard ones.
- Whereas in computability theory, the classification of problems is by those that are solvable and those that are not.
- Computability theory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

- Automata theory deals with the definitions and properties of mathematical models of computation.
- These models play a role in several applied areas of computer science.
- One model, called the finite automaton, is used in text processing, compilers, and hardware design.
- Another model, called the context-free grammar, is used in programming languages and artificial intelligence.

- Automata theory is an excellent place to begin the study of the theory of computation.
- The theories of computability and complexity require a precise definition of a computer.
- Automata theory allows practice with formal definitions of computation as it introduces concepts relevant to other non-theoretical areas of computer science.

- Recognizing strings ending in “ing”

- Recognizing strings ending in “ing”

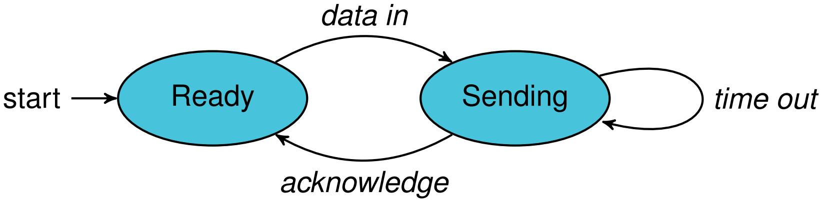


- Protocol for sending data.

Automata, a Motivating Example

Adapted from
http://infolab.stanford.edu/~ullman/ialc/spr10/slides/fa1.pdf

- Protocol for sending data.



End of Slides