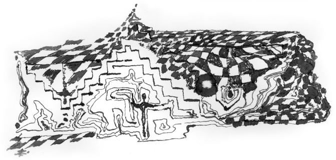
Chapter 1: Origin of Computer Science

Algorithmic Adventures

From Knowledge to Magic

Book by Juraj Hromkovič, ETH Zurich
Slides by Tom Verhoeff, TU Eindhoven

Quotation from Preface



Die Wissenschaft ist innerlich eine Einheit.
Die Aufteilung in einzelne Gebiete
ist nicht durch die Natur der Dinge bedingt,
sondern insbesondere durch die Schranken
der menschlichen Fähigkeiten in dem Erkenntnisprozess.

Max Planck

Quotation

By living always with enthusiasm,
being interested in everything that seems inaccessible,
one becomes greater by striving constantly upwards.
The sense of life is creativity,
and creativity itself is unlimited.

Maxim Gorky

What Computer Science Is and Is Not

- Look at the Informatics bookshelves in a book store

  Mostly about how to use computers and specific programs

- Look at the Informatics curriculum in secondary education

  Mostly about how to use computers and specific programs

- What is a Science?

  Not a collection of discoveries, not a collection of applications

Foundations of Science?

- How is a science founded?
- What are the fundamental building blocks of a science?

Foundations of Science

- Definitions, axioms
- Logical reasoning, calculation (in broad sense), proof

Interlude on Logical Reasoning: Implication, Direct Proof

Truth table for implication, denoted by $\Rightarrow$:

A	B	$A \Rightarrow B$	
true	true	true	
true	false	false	$\leftarrow$ ONLY HERE false
false	true	true	
false	false	true	

A **direct proof** is based on the *transitive property* of $\Rightarrow$:

If $A \Rightarrow B$ and $B \Rightarrow C$, then $A \Rightarrow C$

Interlude on Logical Reasoning: Implication, Transitivity

A	B	C	$A \Rightarrow B$	$B \Rightarrow C$	$A \Rightarrow C$
true	true	true	**true**	**true**	true
true	true	false	true	false	false
true	false	true	false	true	true
true	false	false	false	true	false
false	true	true	**true**	**true**	true
false	true	false	true	false	true
false	false	true	**true**	**true**	true
false	false	false	**true**	**true**	true

If both $A \Rightarrow B$ and $B \Rightarrow C$ hold (bold **true**), then so does $A \Rightarrow C$.

Interlude on Logical Reasoning: Indirect Proof

Truth table for negation, denoted by $\overline{}$:

A	$\overline{A}$
true	false
false	true

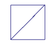
An **indirect proof** is based on the equivalence of

$D \Rightarrow Z$ and $\overline{Z} \Rightarrow \overline{D}$

and in particular also the equivalence of

Z and true $\Rightarrow Z$ and $\overline{Z} \Rightarrow$ false

Example of Indirect Proof

The diagonal of the unit square has length $\sqrt{2}$ (Pythagoras). 

Theorem $\sqrt{2}$ is not a rational number (fraction of integers)

Proof Assume the contrary: $\sqrt{2} = \frac{m}{n}$, for integers m, n that have no common factors. We then calculate (for appropriate k and ℓ):

$2 = (m/n)^2$	m^2 is even	$2n^2 = (2k)^2$	n^2 is even
$2 = m^2/n^2$	m is even	$2n^2 = 4k^2$	n is even
$2n^2 = m^2$	$m = 2k$	$n^2 = 2k^2$	$n = 2\ell$

Thus, m and n have common factor 2, contradicting the assumption.

Consequently, $\sqrt{2}$ is not a fraction of integers. Q.E.D.

Origin of (Theoretical) Computer Science

Hilbert's 23 problems (1900):

1. Continuum hypothesis
2. Consistency of the axioms of arithmetic
8. Riemann hypothesis

Hilbert's research program for Mathematics (1920):

- Formulate finite collection of axioms for all of mathematics
- Formulate a method for deciding the correctness of any statement

Requires: Formalization of the notion of an 'effective method'

The notion of a "method"

A **method** for solving a problem (a task) describes an effective path that leads to the problem solution.

This description must consist of a sequence of instructions that everybody can perform.

One does not need to understand *why a method works* and *how it was discovered* in order to be able to apply it for solving given problem instances.

Example: Solve quadratic equations of the form $x^2 + 2px + q = 0$

Method: $x_1 = -p + \sqrt{p^2 - q}$ and $x_2 = -p - \sqrt{p^2 - q}$

Algorithm

A mathematical method can be executed in an **automatic** way.

An effective solution method is called an **algorithm**.

Algorithm derives from (the Latin form of) the name of the Persian mathematician **al-Khwarizmi** (circa 780–850 AD).

The word *algebra* derives from the word *al-jabr*, one of the operations that al-Khwarizmi used to solve quadratic equations.

Hilbert's Program and Its Failure

Hilbert strove to build a perfect theory of mathematics, in which one has a method for verifying the correctness of all statements expressible in terms of this mathematics.

In 1931, Kurt Gödel proved by mathematical arguments:

(a) There does not exist any complete, "reasonable" mathematical theory. In each consistent and sufficiently "rich" mathematical theory one can formulate statements, whose truthfulness cannot be verified inside this theory. To prove the truthfulness of such theorems, one must add new axioms and so build a new, even richer theory.

(b) A method (algorithm) for automatically proving mathematical theorems does not exist.

(Computer) Science after Gödel

Before Gödel, nobody saw any reason to give an exact definition of the notion of a method.

Such a definition was not needed, because people only presented methods for solving *particular* problems.

When one wants to prove the nonexistence of an algorithm (of a method) for solving a given problem, then one needs to know exactly (in the sense of a rigorous mathematical definition) what an algorithm is and what it is not.

Proving the *nonexistence* of an object is impossible if the object has not been exactly specified.

First Formal Definition of Algorithm

Alan Turing gave the first formal definition of an algorithm in 1936: *Turing Machines*.

Later, further definitions followed (Church, Kleene, Post, . . .).

These definitions use very different mathematical approaches and formalisms.

All reasonable attempts to create a definition of the notion of an algorithm led to the same meaning of this term: they all specify the same classes of algorithmically (un)solvable problems.

Turing's definition is viewed as the first axiom of computer science.

Questions

Consider a scientific discipline of your choice:

- Identify where/how the notion of *information* plays a role
- Give an example
- Identify where/how the notion of *algorithm* plays a role
- Give an example

Summary

- Fundamental activity in science: create notions and methods
- Computer Science: introduce formal notion of 'algorithm'
- Allows investigation of computability (algorithmic solvability)
- Allows investigation of computational complexity