# Foundations of Computer Science

Cezar Ionescu
cezar.ionescu@th-deg.de

WS 2023-24

# Outline

# Administratrivia

# Standard procedure

- Weekly classes, usually lecture in the first block (9:45-11:15) and exercises in the second block (11:30-13:00)

# Assessment

- Mid-term: pass or fail, needed for admission to the written exam
  - *Take home exam* (homework), released on the 14th of December, due on the 21st
    - No lecture on the 21st of December!
    - You must register for the mid-term!
- Written exam:
  - 90 min, **in presence**
  - *similar* exercises to the problem sheets

# Contact

- preferred method via email, **please use your THD account**
  - emails sent from other accounts might go unanswered
- office hours by appointment

# Introduction

# Computer Science

- **Computer Science**: The branch of knowledge concerned with the construction, programming, operation, and use of computers

Oxford English Dictionary (requires subscription)

# Computer vs. Computing

**Edsger Wybe Dijkstra (1930-2002)**

*Computer science is no more about computers than astronomy is about telescopes.*

# Computer vs. Computing

Cambridge vs. Oxford

- Cambridge: Computer Laboratory (1937)
- Oxford: Computing Laboratory (1956-2011)
  - but since 2011: Department of Computer Science

# Alan Turing

**Alan Turing (1912-1954)**

- 1936: Turing machine (*On Computable Numbers, with an Application to the Entscheidungsproblem*)
- 1943: Colossus, first programmable, electronic, digital computer,
  - Could decrypt in hours encoded messages (but not by Enigma!) that took humans days or even weeks.
- 1949: Program correctness (*Checking a Large Routine*)
- 1950: Turing Test (*Computing Machinery and Intelligence*)

# An economist's point of view

*What did the semiconductor revolution reduce the cost of? In a word: arithmetic.*

*Before semiconductors, "computers" were humans who were employed to do arithmetic problems. Digital computers made arithmetic inexpensive, which eventually resulted in thousands of new applications for everything from data storage to word processing to photography.*

Ajay Agrawal, Joshua S. Gans, and Avi Goldfarb

https://sloanreview.mit.edu/article/what-to-expect-from-artificial-intelligence/

# Foundations

- **Foundations**: (plural). [translating German *grundlagen*, etc.] The underlying principles or logical basis (of a subject), esp. as a separate matter for study.

  Examples:

  - 1817 S. T. Coleridge Biogr. Lit. x. 194 I retired to a cottage in Somersetshire..and devoted my thoughts and studies to the foundations of religions and morals.
  - 1931 F. P. Ramsey (title) The foundations of mathematics.
  - 1941 R. Courant & H. E. Robbins What is Math.? ii. 87 Paradoxes like this have led Russell and others to a systematic study of the foundations of mathematics and logic.
  - 1944 H. Reichenbach (title) Philosophic foundations of quantum mechanics.

Oxford Dictionary of English 2019

https://www.oed.com/view/Entry/73932

# Foundations of Computer Science

- Foundations of *computer engineering*
  - hardware-oriented
  - German: *technische Informatik*
  - digital circuits
  - computer architecture
  - storage organization
  - network technology

  Could also include
  - binary decision diagrams
  - error handling, coding, encryption
  - programming
  - operating systems

# Foundations of Computer Science

- Foundations of *theoretical computer science*
  - software-oriented
  - German: *theoretische* Informatik
  - logic
  - computability
  - finite automata
  - formal languages
  - complexity theory

  Could also include
  - binary decision diagrams
  - error handling, coding, encryption
  - programming
  - operating systems

# Algorithms

- The concept of *algorithm* is central to computer science.
- From the OED:

***algorithm, n.***

1. Mathematics *and* Computing. *A procedure or set of rules used in calculation and problem-solving; (in later use spec.) a precisely defined set of mathematical or logical operations for the performance of a particular task.*

*Origin: A borrowing from Latin. Etymon: Latin algorithmus. Etymology: < post-classical Latin algorithmus (15th cent.), alteration of* algorismus *ALGORISM n., after ancient Greek ἀριθμός number (see ARITHMETIC n.).*

- Alternative etymology: from the name of the persian-arab mathematician Al-Hwarizmī (d. 846).

# Example

Given natural numbers `X > 0, Y > 0`.

```
x, y = X, Y
while x ≠ y:
  if x > y then x = x - y
  if y > x then y = y - x
print((x+y)/2)
```

What is the output?

# Example: Newton's method

Let $f : \mathbb{R} \to \mathbb{R}$ be a differentiable function with continuous derivative, and $x_0 \in \mathbb{R}$.

Then the sequence defined by

$$x_{n+1} = x_n - \frac{f(x_n)}{f' x_n}$$

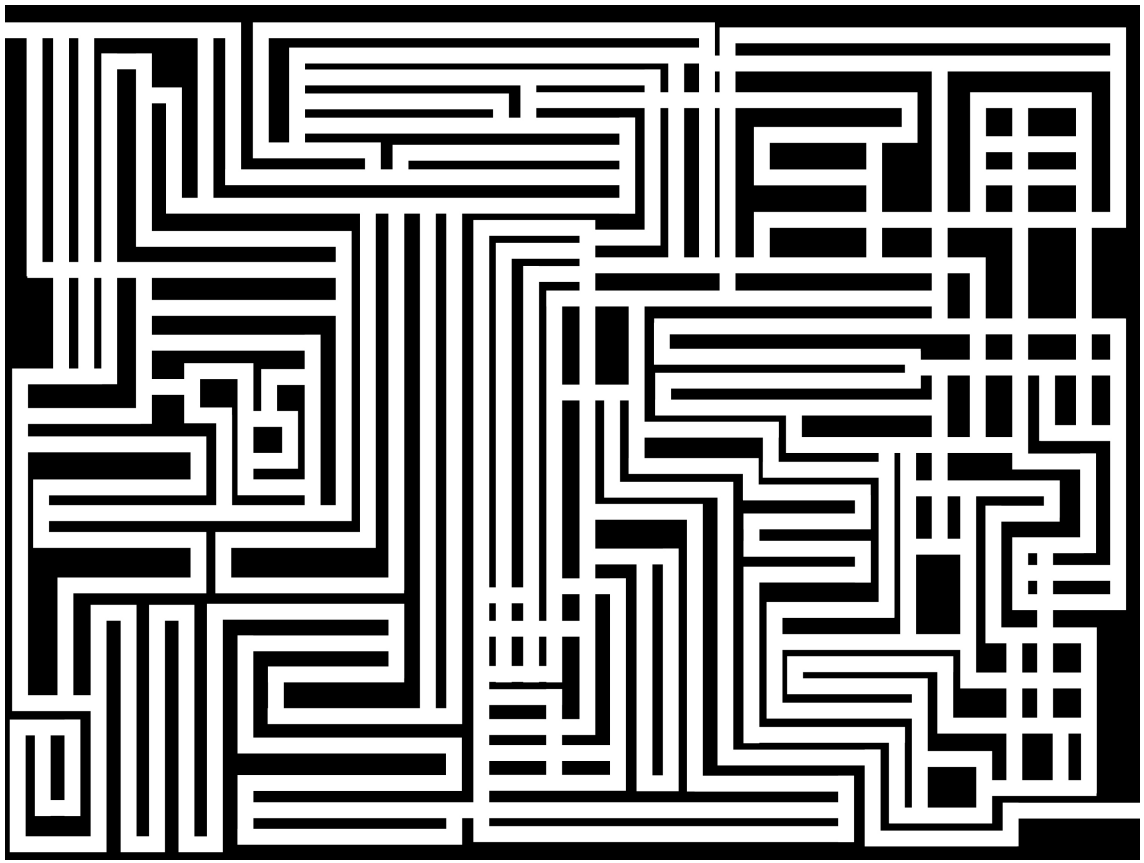
converges[1] to a **zero** of $f$.

Graphical example (Wikipedia)

[1] For an "appropriate" choice of $x_0$. Standard disclaimers apply.

# Example: Labyrinth

The classical algorithm for finding the way out of a maze: walk while keeping your (e.g., right) hand touching the wall. Follow the wall at any fork in the road. Sooner or later, you'll get to an exit[2].



[2]Unless you happen to be walking along an "island", in which case you should switch to the opposite wall when you realize you're tracing your footsteps.

# Algorithm?

**George Pólya (1887-1985)**

- **First:** *Understand* the problem.
- **Second:** Obtain a *plan* of the solution.
- **Third:** *Carry out* your plan.
- **Fourth:** *Examine* the solution obtained.

# General Problem Solver

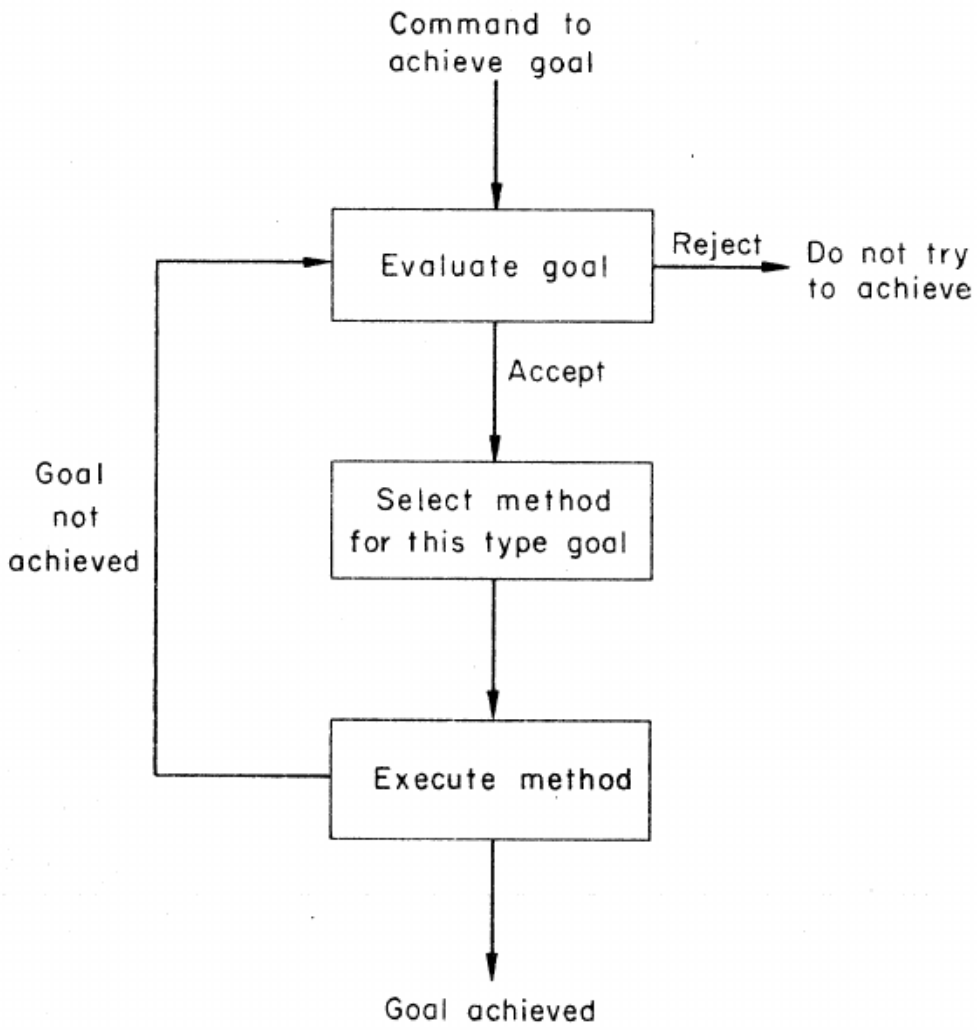



Fig. 1—Executive organization of GPS

Newell, Shaw, Simon **General Problem Solver** (1958)

# Questions

The central questions of computer science:

- What exactly is an algorithm? Are there (mathematical) problems for which no algorithm can be given?
- How are algorithms to be described? Implemented? Compared?
- What kind of machines can carry out algorithms? How can such machines be built? Do we need a machine for each algorithm, or is there a "universal" machine?
- Etc.

# Plan

Plan:

- binary arithmetic
- digital circuits
- formal languages
- finite automata
- computability

Other potential topics:

- programming languages
- logic