Literature

Introduction to Automata Theory, Languages, and Computation.
J.E. Hopcraft, J.D. Ullman. Addison Wesley, 1979. Chap. 7, 8, 12, 13.
2nd edition, 2001, with R. Motwani: Chapter 8-10.

Background:

- *Elements of the Theory of Computation.* 2nd edition. H.R. Lewis, C.H. Papadimitriou. Prentice-Hall. 1998.
- *Introduction to Languages and the Theory of Computation*
  J. Martin, McGraw-Hill, 3rd edition 2003. Parts IV - VI.
- *Computability and Complexity: From a programming perspective.*
  Neil D. Jones, MIT Press. 1996. Chapter 4-8, 15, 16, 22, 25-27
- *A Programming Approach to Computability.* A.J. Kfoury, R.N. Moll and M.A. Arbib. Springer, 1982.

What we will study

- **§I Computability**

  A formal and intuitive description of computation.

  Is everything computable? undecidable problems

- **§II Complexity**

  A formal "measure" of how efficient algorithms/programs are.

  What is practically feasible? intractable problems

§I.1 Computability - History

David Hilbert

(1862-1943, Königsberg/Kaliningrad - Göttingen)

Brilliant mathematician (geometry) who claimed that arithmetic (mathematics) is

- consistent (no contradiction can be derived)
- complete (every true formula is provable)
- decidable (there is an algorithm to detect true formulas)
  a.k.a *Entscheidungsproblem* (cf. 23 Hilbert problems)

Hilbert was a formalist and thought that there was one fundamental system for all mathematics.

I.1 Computability: History

Kurt Gödel

(1906-1978, Brünn/Brno - Princeton)

- 1931 young Gödel proved the famous Incompleteness Theorem: that arithmetic cannot be consistent and complete at the same time
- Result: Earthquake in Mathematics
  partly smashed Hilbert's programme
- $\boxed{\text{provability} \neq \text{truth}}$
- Trick: Self reference

§I.1 Computability: History

Basic Idea of Gödel's Incompleteness Theorem

- 2 main ideas: If a formal system T (proof system) contains arithmetic one can define **inside** T
  1. the provability of the n-th formula of T as predicate on $\mathbb{N}$
  2. one can define a formula stating: "I am not provable" which is true but cannot be provable in a sound proof system.
- Self reference is most important:
  - cf. ancient liar paradox: "Everything I say is a lie." $\lightning$
  - cf. Barber's paradox: "I shave the men in this street that do not shave themselves (and no others)." $\lightning$

§I.1 Computability: History

Alan Mathison Turing
(1912-1954, London - Wilmslow/Cheshire)

Turing wanted to describe formally the notion of “computation” so he

- invented an **abstract machine**, the Turing Machine (TM), i.e. a formalization of a “computer” (1936 computer=human)
- invented the Universal TM, a **universally programmable** TM
- proved with it *Undecidability of the Halting Problem* in parallel with Alonzo Church (1903-1995 Washington)
  $\Longrightarrow$ *Entscheidungsproblem* solved
  $\Longrightarrow$ Hilbert's program smashed for good