LOGIC II

WHAT IS COMPUTATION?

Sina Hazratpour

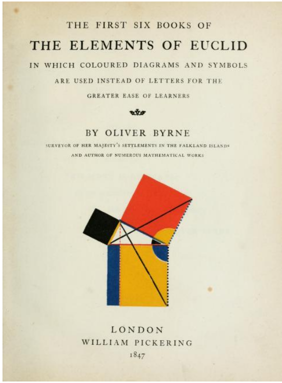
Stockholms Universitet

Fall 2025

History & Formal Models

Conception of Computation in Antiquity

- Informal conception of computation: **sequence of steps** performed according to some kind of **recipe**.
- This conception goes back to antiquity: The Greek Mathematician *Euclid* introduced compass-and-straightedge constructions as a proto-algorithmic process in his *Elements* (c. 300 BC).
- Euclid's algorithm for computing the greatest common divisor (gcd) of two numbers.



Conception of Computation in Middle Ages

- The word **algorithm** comes from the 9th century Persian mathematician *Khwarizmi* (ca. 825) who worked on arithmetic, **equations**, and word problems. His treatise *Hisab al-jabr w'al-muqabalah* gave us the word **algebra**.

- In his 'Book of Indian computation' he described algorithms on decimal numbers (Hindu–Arabic numerals) that could be carried out on a dust board.

History of Computing is History of Mechanization

- Blaise Pascal's calculating machine (1642): *Pascaline*. It could do arithemtic addition and subtraction.

- Leibniz's improved calculating machine (1671) : *Machina Arithmetica* could do all the basic four arithmetic operations.

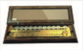
Zweispezies-Rechenmaschine von Blaise Pascal, 1926, Kopie des Originals im CNAM, Paris, FDM 9665, © Arithmeum

`Machina Arithmetica`

Images: Courtesy of Arithmeum

(`https://www.arithmeum.uni-bonn.de`).

Mechanical Computers (17th Century)

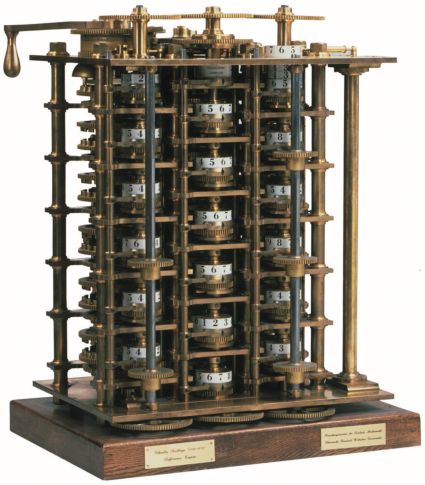
	Schickard	Schickard, Wilhelm	1623	Ein- bis Dreispeziesmaschine
	Rechenpfennig		1630	Einfaches Rechenhilfsmittel, Rechenpfer
	Rechenpfennig mit Rechenmeister	Hans Krauwinckel	1630	Einfaches Rechenhilfsmittel, Rechenpfer
	Pascaline	Pascal, Blaise	1642	Ein- bis Dreispeziesmaschine
	Pascaline (Historischer Nachbau)	Pascal, Blaise	1642	Ein- bis Dreispeziesmaschine
	Proportionalzirkel 17. Jahrhundert		1650	Ablesegerät mit Skalen, logarithmisch, Proportionalzirkel
	Morland Multipliziergerät	Morland, Sir Samuel	1664	Ein- bis Dreispeziesmaschine
	Morland Addiergerät	Morland, Sir Samuel	1666	Einfaches Rechenhilfsmittel
	Leibniz machina arithmetica	Leibniz, Gottfried Wilhelm	1671	Vierspeziesmaschine, Staffelwalze

Mechanical Computers (18th Century)

	Braun / Vayringe (Villingen-Schwenningen)	Braun, Anton	1727	Vierspeziesmaschine, Stellsegment
	Braun Sprossenradmaschine	Braun, Anton	1727	Vierspeziesmaschine, Sprossenrad
	Rotula Arithmetica von George Brown	G. Adams	1734	Ein- bis Dreispeziesmaschine
	Hahn I	Hahn, Philipp Matthäus	1774	Vierspeziesmaschine, Staffelwalze
	Stanhope Staffelwalzenmaschine	Stanhope, Charles Viscount Mahon, 3rd Earl of / James Bullock	1775	Vierspeziesmaschine, Staffelwalze
	Stanhope Stellsegmentmaschine	Stanhope, Charles Viscount Mahon, 3rd Earl of / James Bullock	1777	Vierspeziesmaschine, Stellsegment

Babbage’s Difference Engine (19th Century)

- Charles Babbage: Difference Engine (1822) and Analytical Engine (1837).
- Ada Lovelace (1815-1852) wrote early computer programs for this machine.
- Babbage’s Difference Engine (v.2) was designed to calculate polynomials up to the seventh degree, handling numbers up to 31 decimal digits in accuracy.



Babbage Difference Engine, 1821, FDM9114, © Arithmeum

Babbage’s Difference Engine

Computation in the 20th Century

- In the 1910s and 1920s century an increasing variety of mechanical computing devices were developed.
- Yet, most of these devices were limited to specific tasks, such as tabulating mathematical functions or performing basic arithmetic operations.
- In practice, most computations were still done by human "computers" following detailed instructions. World War I and II created huge demand for human computers, map grids, surveying aids, navigation tables and artillery tables, and code breaking.


4'×4' SUPERSONIC PRESSURE
TUNNEL
1236

What was lacking in all these developments up until this point was a **precise definition of what it means for a function to be computable**.

Hilbert's Famous 23 Problems (1900)

- In 1900 at the International Congress of Mathematicians in Paris, mathematician David Hilbert presented a list of 23 unsolved problems.
- The tenth problem: determine whether a given Diophantine equation has integer solutions.
- A Diophantine equation is a polynomial equation with integer coefficients, seeking integer solutions.
- Hilbert asked for an **algorithm** to decide this for any such equation.

David Hilbert (1862–1943)

A Negative Response to Hilbert

- Much later in the century, this problem was solved in the negative by Yuri Matiyasevich in 1970.
- For this purpose, having a formal model of computability was essential.
- To show that something is computable you describe an algorithm meeting some formal criteria.
- In order to show that no computational procedure can solve Diophantine equations, one has characterize all possible computational procedures!
- It was crucial for Matiyasevich to use a formal model of computation.
- Luckily for him this was developed in the 1930s.

Young Gödel at Hilbert’s Lecture

„Wir dürfen nicht denen glauben, die heute mit philosophischer Miene und überlegenem Tone den Kulturuntergang prophezeien und sich in dem Ignorabimus gefallen. Für uns gibt es kein Ignorabimus und meiner Meinung nach auch für die Naturwissenschaft überhaupt nicht. Statt des törichten Ignorabimus heiße im Gegenteil unsere Losung: wir müssen wissen, wir werden wissen.“

Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I[1].

Von Kurt Gödel in Wien.

1.

Die Entwicklung der Mathematik in der Richtung zu größerer Exaktheit hat bekanntlich dazu geführt, daß weite Gebiete von ihr formalisiert wurden, in der Art, daß das Beweisen nach einigen wenigen mechanischen Regeln vollzogen werden kann. Die umfassendsten derzeit aufgestellten formalen Systeme sind das System der Principia Mathematica (PM)[2] einerseits, das Zermelo-Fraenkelsche (von J. v. Neumann weiter ausgebildete) Axiomensystem der Mengenlehre[3] andererseits. Diese beiden Systeme sind so weit, daß alle heute in der Mathematik angewendeten Beweismethoden in ihnen formalisiert, d. h. auf einige wenige Axiome und Schlußregeln zurückgeführt sind. Es liegt daher die Vermutung nahe, daß diese Axiome und Schlußregeln dazu ausreichen, alle mathematischen Fragen, die sich in den betreffenden Systemen überhaupt formal ausdrücken lassen, auch zu entscheiden. Im folgenden wird gezeigt, daß dies nicht der Fall ist, sondern daß es in den beiden angeführten Systemen sogar relativ einfache Probleme aus der Theorie der gewöhnlichen ganzen Zahlen gibt[4], die sich aus den Axiomen nicht

---

[1]) Vgl. die im Anzeiger der Akad. d. Wiss. in Wien (math.-naturw. Kl.) 1930, Nr. 19 erschienene Zusammenfassung der Resultate dieser Arbeit.

[2]) A. Whitehead und B. Russell, Principia Mathematica, 2. Aufl., Cambridge 1925. Zu den Axiomen des Systems PM rechnen wir insbesondere auch: Das Unendlichkeitsaxiom (in der Form: es gibt genau abzählbar viele Individuen), das Reduzibilitäts- und das Auswahlaxiom (für alle Typen).

[3]) Vgl. A. Fraenkel, Zehn Vorlesungen über die Grundlegung der Mengenlehre, Wissensch. u. Hyp. Bd. XXXI. J. v. Neumann, Die Axiomatisierung der Mengenlehre. Math. Zeitschr. 27, 1928. Journ. f. reine u. angew. Math. 154 (1925), 160 (1929). Wir bemerken, daß man zu den in der angeführten Literatur gegebenen mengentheoretischen Axiomen noch die Axiome und Schlußregeln des Logikkalküls hinzufügen muß, um die Formalisierung zu vollenden. — Die nachfolgenden Überlegungen gelten auch für die in den letzten Jahren von D. Hilbert und seinen Mitarbeitern aufgestellten formalen Systeme (soweit diese bisher vorliegen). Vgl. D. Hilbert, Math. Ann. 88, Abh. aus d. math. Sem. der Univ. Hamburg I (1922), VI (1928). P. Bernays, Math. Ann. 90. J. v. Neumann, Math. Zeitschr. 26 (1927). W. Ackermann, Math. Ann. 93.

Formal Models of Computation

- **Partial recursive functions** (Gödel, Herbrand - 1930s)
- Lambda calculus (Church - 1936)
- Turing machines (Turing - 1936)
- String rewriting systems (Post - 1940s)
- State register machines (Minsky - 1960s)
- Cellular automata (von Neumann - 1950s, Conway - 1970s)
- Nondeterministic Turing machines (1960s)
- Programmability in programming languages, like ML, C++, or Java.

Real World Computers

- Theory of computation predates modern computers by about a decade.
- 1944: IBM–Harvard Automatic Sequence Controlled Calculator (ASCC or Mark I).
- 1945: John von Neumann's design of the EDVAC.
- Subsequent: ENIAC, MANIAC, UNIVAC, and more.

Going back to Leibniz

- Leibniz in *De Arte Combinatoria* (1666) proposed the idea of a **universal language** (characteristica universalis) and a **calculus of thoughts** (calculus ratiocinator).
- "All reasoning is just a computation."
- This idea, a radical and prophetic idea in his time, influenced later thinkers like Boole, Frege, and Turing.

Quelle: Deutsche Fotothek