Computer Science Basics
Computer Science History

Fall Term 2025/2026

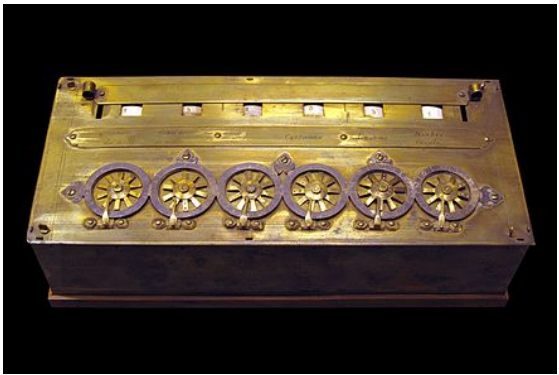
Emmanuel Benoist | BFH-TI

Computer Science History

- **Prehistory of Computer Science**
- **WWII: First computers**
- **Real computers**
- **Networks**
- **Conclusion**

Prehistory of Computer Science

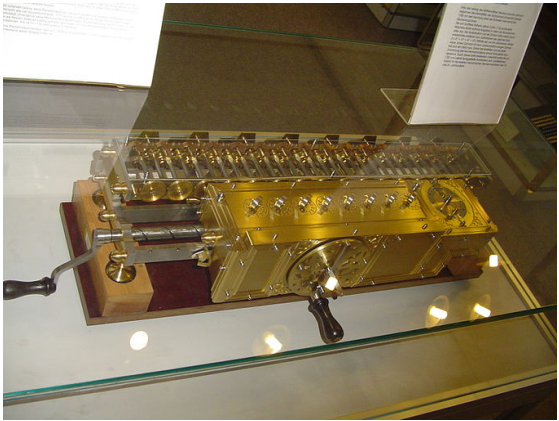
Blaise Pascal's Pascaline - 1645



Designed a machine to add and subtract two numbers

- Multiplication and division are done through repeated addition or subtraction
- Was not a mass success only a few were built

Leibniz's Stepped reckoner - 1694



Perform all four arithmetic operations

- Was never built in series
- Inputs 16 digits and 8 digits.
- add or subtract an 8-digit number to / from a 16-digit number
- multiply two 8-digit numbers to get a 16-digit result
- divide a 16-digit number by an 8-digit divisor

Leibniz invents binary numbers

All integer number can be represented as a binary number

- 0, 1, 10, 11, 100, 101, 110, 111, 1000,1001, 1010, ...

Can be represented using

- a ball (present or not)
- a punched card (like for Babbage machine)
- a relay that is manipulated by electricity
- a vacuum tube
- transistors.

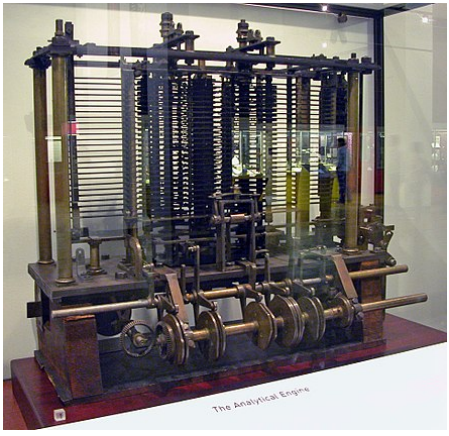
Analytical Engine

Designed by Charles Babbage

- General-Purpose computer
- First description in 1837,
- Construction was not finished at his death in 1871
- Was partly constructed by his son (1910)

Principles

- Worked with a steam powered machine
- Input was Formulae and Data: punched cards.
- It uses a store (“Memory”) of 1000 numbers of 40 decimal digits (ca. 16kB).



Trial model of a part of the Analytical Engine, built by Babbage, as displayed at the Science Museum

Analytical Engine II

First program designed by Ada Lovelace

- Program for computing Bernoulli numbers with Babbage's Machine.
- Similar to Assembly language
- Conditional branching, looping are possible

Analytical Engine III

Two types of punched cards used to program the machine.

- Foreground: 'operational cards', for inputting instructions;
- Background: 'variable cards', for inputting data

Unfortunately fell into historical obscurity !

WWII: First computers

The Bomb I

Enigma

- German used Enigma machines to encrypt messages of the navy and army

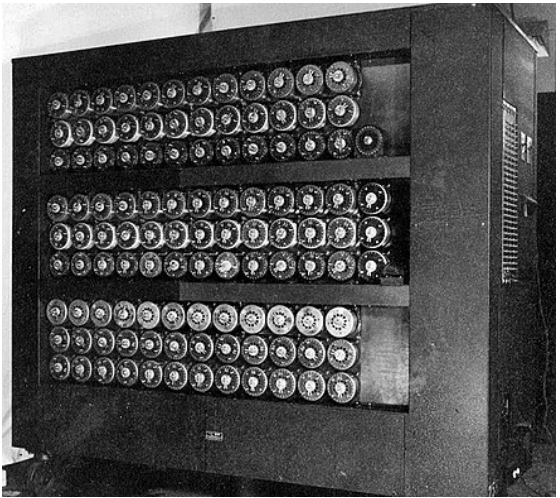
The Bomb II

Marian Rejewski (Poland) developed a machine for hacking Enigma (before WWII)

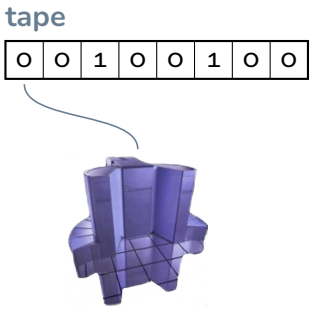
- German made the system more complex
- Alan Turing adapted the “Bombe” at Bletchley Park.

Designed specifically to crack Enigma codes

- English knew how Enigma works.
- They needed to find a new key every day.
- Specifically designed to “Brute Force” Enigma
- Not programable to do something else.



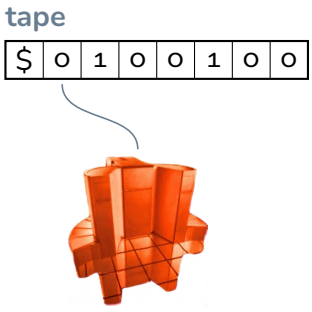
Turing Machines : a theoretical representation of computing



Characteristics

- Input fills fixed-size tape
- Input may be modified
- Tape alphabet is larger than input alphabet
- Machine can be in a finite set of states.
- Program is a set of rules :
  (symbol on tape, state) → (new symbol on tape, new state, movement on tape)

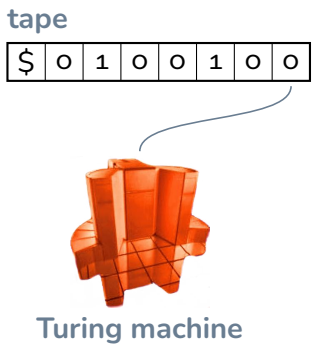
Turing Machines : a theoretical representation of computing



Characteristics

- Input fills fixed-size tape
- Input may be modified
- Tape alphabet is larger than input alphabet
- Machine can be in a finite set of states.
- Program is a set of rules :
  (symbol on tape, state) → (new symbol on tape, new state, movement on tape)

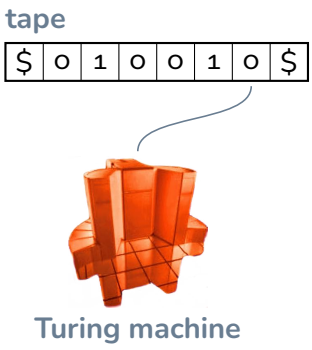
Turing Machines : a theoretical representation of computing



Characteristics

- Input fills fixed-size tape
- Input may be modified
- Tape alphabet is larger than input alphabet
- Machine can be in a finite set of states.
- Program is a set of rules :
  (symbol on tape, state) → (new symbol on tape, new state, movement on tape)

Turing Machines : a theoretical representation of computing



Characteristics

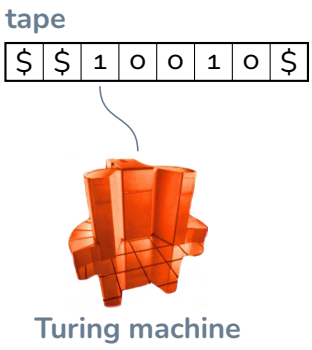
- Input fills fixed-size tape
- Input may be modified
- Tape alphabet is larger than input alphabet
- Machine can be in a finite set of states.
- Program is a set of rules :
  (symbol on tape, state) → (new symbol on tape, new state, movement on tape)

Turing Machines : a theoretical representation of computing

Characteristics

- Input fills fixed-size tape
- Input may be modified
- Tape alphabet is larger than input alphabet
- Machine can be in a finite set of states.
- Program is a set of rules :
  (symbol on tape, state) → (new symbol on tape, new state, movement on tape)

Turing Machines : a theoretical representation of computing



Characteristics

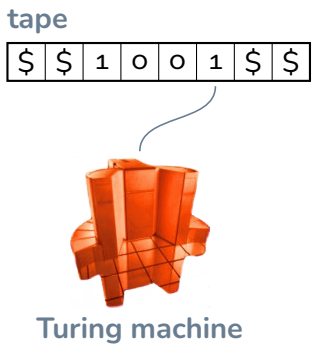
- Input fills fixed-size tape
- Input may be modified
- Tape alphabet is larger than input alphabet
- Machine can be in a finite set of states.
- Program is a set of rules :
  (symbol on tape, state) → (new symbol on tape, new state, movement on tape)

Turing Machines : a theoretical representation of computing

Characteristics

- Input fills fixed-size tape
- Input may be modified
- Tape alphabet is larger than input alphabet
- Machine can be in a finite set of states.
- Program is a set of rules :
  (symbol on tape, state) → (new symbol on tape, new state, movement on tape)

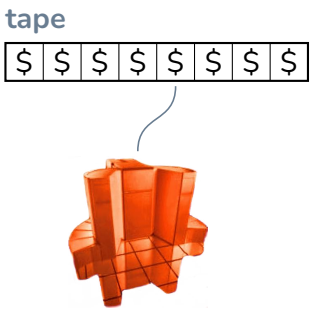
Turing Machines : a theoretical representation of computing



Characteristics

- Input fills fixed-size tape
- Input may be modified
- Tape alphabet is larger than input alphabet
- Machine can be in a finite set of states.
- Program is a set of rules :
  (symbol on tape, state) → (new symbol on tape, new state, movement on tape)

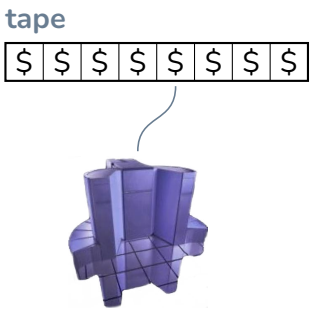
Turing Machines : a theoretical representation of computing



Characteristics

- Input fills fixed-size tape
- Input may be modified
- Tape alphabet is larger than input alphabet
- Machine can be in a finite set of states.
- Program is a set of rules :
  (symbol on tape, state) → (new symbol on tape, new state, movement on tape)

Turing Machines : a theoretical representation of computing



Characteristics

- Input fills fixed-size tape
- Input may be modified
- Tape alphabet is larger than input alphabet
- Machine can be in a finite set of states.
- Program is a set of rules :
  (symbol on tape, state) → (new symbol on tape, new state, movement on tape)

Alan Turing: the Turing Machines

Turing Machine

- Alan Turing 1936
- Is still the algorithmic model used nowadays,
- Used for the definition of problem classes like P, NP, coNP, NP-Complete
- Is used to compare powers of languages :
  "The language X is Turing Complete" means it has the same strength as a Turing Machine (and hence as any other programming language).

Definition of a TM

- One Band with data, can be read and written
- A set of sates (the machine can be in various states)
- A set of transformation (state + tape $\rightarrow$ state2+tape2 + movement)

Turing Church thesis

- Any computation can be done using a Turing Machine

Z3

First programable computer in the world

- Developed by Konrad Zuse for the German Nazis
- Goal: solve Wing flutter problems (Dynamic aeroelasticity).

Z3 was constructed in 1941 and destructed in december 1943

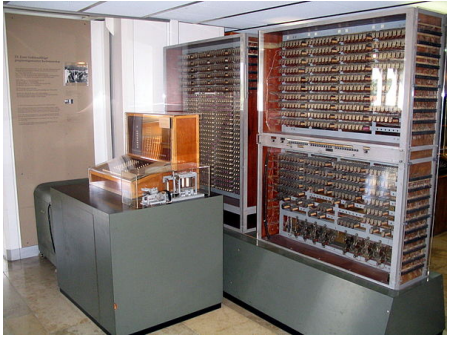
- It was destroyed in an allied bombardement of Berlin.

Features

- Uses Binary system
- 22-bit word length
- Program code was stored punched on film
- Initial values entered manually

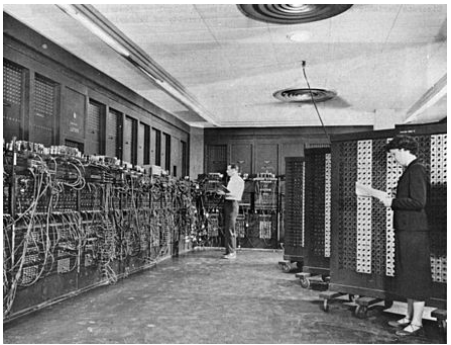
Limitations

- Does contain neither conditional branching, nor loops.
- It has been proven Turing-complete (1998).



Zuse Z3 replica on display at Deutsches Museum in Munich

ENIAC



Built by Americans in 1945

- First computer to be Turing-Complete,
- Can be programmed to solve any computing problem

Use Vacuum tubes

- Only electronic (no more mechanic)
- Information is stored binary

Problem

- Uses decimal numbers
- Must be cabled to change programs

Von Neumann architecture

Initially

- Program is hardcoded
- Data is represented by a status of the system

Improvement

- Program is written on a tape (film, punch cards, ...)
- Data is stored in memory

Von Neuman architecture

- Programs and Data can be stored in memory
- Both have the same representation
- Memory can be used for Data or Program!

Real computers

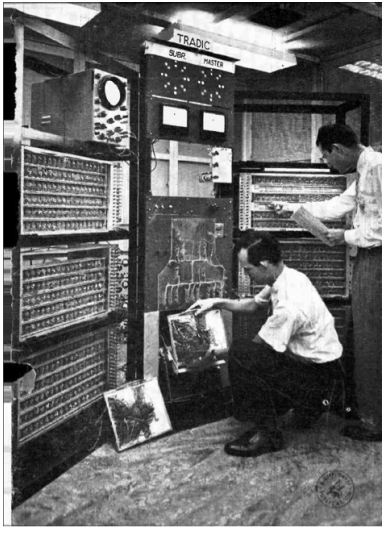
Univac

One big computer for big computation

Univac 1951

- 5000 tubes
- Memory = 1000 words of 12 bits
- 8333 addition / s or 555 multiplications / s
- 25 m^2

TRADIC: First computer with transistors



Used for memory

- Store information without vacuum tubes

Networks

Internet I

Local Area Networks (LAN)

- Computer in an organisation
- Connected together
- Infrastructures : star (one server in the middle), token ring (one ring, each host talks after the other), …
- Different protocols (Apple Talk, Windows for Workgroups 3.11, Add hoc networks for Mainfraims).

Connections between Networks

- Ad hoc connections (point to point)
- Interconnection of networks
  X25 in Europe (circuit switching)
  Arpanet in USA (packet switching)

Internet

- Packet Switching
- IP stack of protocols
- Interconnection of the different networks

Internet II

IP Protocol (Vinton Cerf)

- Every machine has an address
- Actually IP V4 (192.168.0.1) or IP V6 (2001:db8:3333:4444:CCCC:DDDD:EEEE:FFFF)

Communication between Networks

- Routing is done on the fly: no central point

Domain Name Service DNS

- Transforms a name www.benoist.ch into an IP address : 194.150.248.53

World Wide Web

1990 Tim Berners-Lee at CERN (Geneva)

- A protocol : HTTP (HyperText Transport Protocole)
- A format for HyperText documents : HTML (HyperText Markup Language)
- A browser to display the document and navigate using links

First browsers

- Text browsers
- Mosaic (1993) first popular graphical browser

Conclusion

Features of a computer

?

Features of a computer

What is a computer ?

- A computer can make calculations.
- Contains Data and a Program
- Data and Program are in memory (von Neuman Architecure).
- Computers are made of transistors.

How does it work?

- Programs must be started and managed by an Operating System (OS).
- Files must be stored on permanent devices (tapes, floppy, disks, flash memory, ...).

Bibliography

- **Wikipedia**
  - https://en.wikipedia.org/wiki/Pascal%27s_calculator
  - https://en.wikipedia.org/wiki/Leibniz_wheel
  - https://en.wikipedia.org/wiki/Analytical_Engine
  - https://en.wikipedia.org/wiki/Bombe
  - https://en.wikipedia.org/wiki/Z3_(computer)
  - https://en.wikipedia.org/wiki/ENIAC
- **History of Mainframes (french)**
  - http://zeriodes.fr/z_opportunites/