Algorithms and calculators

Giuseppe Lipari

`http://feanor.sssup.it/~lipari`

Scuola Superiore Sant'Anna – Pisa

January 19, 2010

Outline

1. Algorithms
2. Early calculators
3. von Neumann Model
4. Credits

Outline

1. Algorithms
2. Early calculators
3. von Neumann Model
4. Credits

Something to think about

> *“Computer science is no more about computers than astronomy is about telescopes”*

– Edsger Dijkstra

What else might it be about?

Algorithms, programs, processes

- Algorithm:
  - It is a logical procedure thought to solve a certain problem
  - It is informally specified as a sequence of elementary steps that an *execution machine* must follow to solve the problem
  - it is not necessarily expressed in a formal programming language!
- Program:
  - It is the implementation of an algorithm in a programming language, that can be executed by an autonomous machine (calculator)
  - It can be executed several times, every time with different inputs
- Process:
  - An instance of a program that, given a set of input values, produces a set of outputs

Algorithm

- Given a *computational* problem, it is necessary to find a *procedure*, consisting of a *finite set of simple steps* that will produce the solution of the problem.
- Such a procedure is called “Algorithm” in honor of arab mathematician Mohammed ibn-Musa al-Khuwarizmi (VIII century AC)

Figure: al-Khuwarizmi

Examples:

- How to prepare a coffe
- How to buy a coffe from the vending machine
- How to calculate the square root of a number

Calculators

- An algorithm needs a machine to execute it
- Machine here is intended in the abstract sense
  - It can also be a human being, or group of people
  - However, it is important that the algorithm it is *described* so that the machine can execute it without further instructions, or wrong interpretation of what to do
  - Therefore, the steps must be simple, and precisely described

Coffe time!

- Example: in the description of the algorithm to prepare a coffe:
  - we must specify how much coffe to put, so that the *machine* cannot be wrong in preparing it
  - If the *machine* is a calculator (a *stupid* machine!), then we must tell it *exactly* how much coffe to put
  - If the machine is *smart*, we can be less precise, for example, put “coffee” until the machine is full

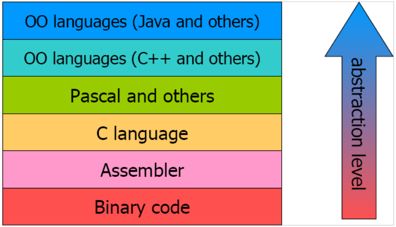
Programs

- In this course, we are interested in describing an algorithm so that a *computer* can understand and execute it
- How to communicate with a computer?

- We need to use a language that the computer can understand
  - A programming language is not so much different than any human language
  - The main difference is that the interpretation of a *sentence* expressed in a programming language must be unambigous
  - Human languages instead allow plenty of ambiguities!

Languages

- But the computer only *understands* two symbols: 0 and 1!
    - Then, every language must be *coded* in binary
    - However, coding in binary is tedious and prone to errors
    - No problem: we can translate from a *high level language* (close to human communication) to a *low level language* (coded in binary, and suitable to computers)



OO languages (Java and others)

OO languages (C++ and others)

Pascal and others

C language

Assembler

Binary code

abstraction level

Outline

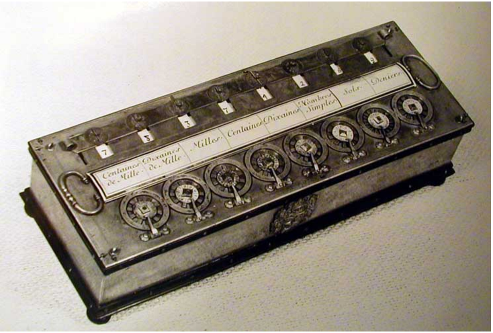
1 Algorithms

2 Early calculators

3 von Neumann Model

4 Credits

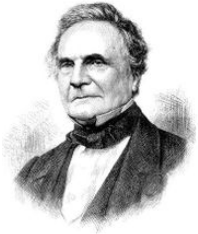
Pascaline



- This machine was invented by Blaise Pascal
- Could be used to add integer numbers by *dialing* the numbers

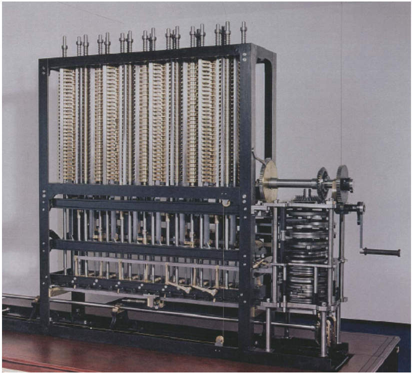
Charles Babbage

- Numerical tables were calculated by humans called ’computers’.
- Babbage saw the high error rate of the people computing the tables



- Babbage wanted to calculate the tables mechanically, removing all human error. He began in 1822 with what he called the difference engine, made to compute values of polynomial functions.
- The first difference engine needed around 25,000 parts of a combined weight of fifteen tons standing eight feet high. Although he received much funding for the project, he did not complete it.

Babbage – II



- Babbage started designing a different, more complex machine called the Analytical Engine, which could be programmed using punch cards, an idea unheard of in his time.
- Several features subsequently used in modern computers, including sequential control, branching, and looping

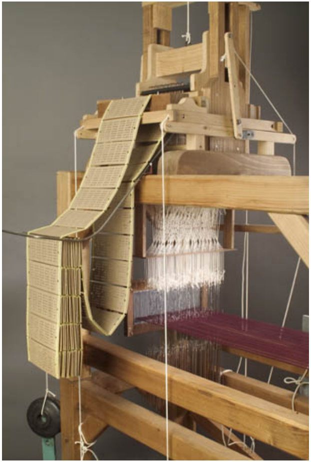
Lady Ada Lovelace

- Ada Lovelace, (Augusta Ada King, Countess of Lovelace (December 10, 1815 – November 27, 1852))
- Ada was the only legitimate child of the poet Lord Byron and his wife, Annabella Milbanke

- She was an impressive mathematician and one of the few who understood Babbage's vision,
- She created a program for the Analytical Engine.
- Based on this work, Ada is now credited as being the first computer programmer and, in 1979, a contemporary programming language was named Ada in her honour.

A strage kind of computer

Jacquard Loom

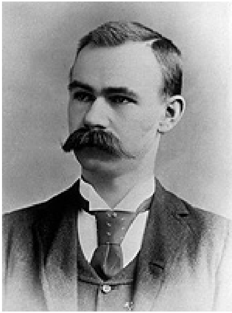


This machine takes instructions by using *punched cards* (like early computers), to produce weavings:

Note the repetitive nature of the task

Hermann Hollerith

- American engineer Herman Hollerith developed a substantial business in punch card punching, sorting and tabulating machines, based on his patents, which were used in the US census quite early.

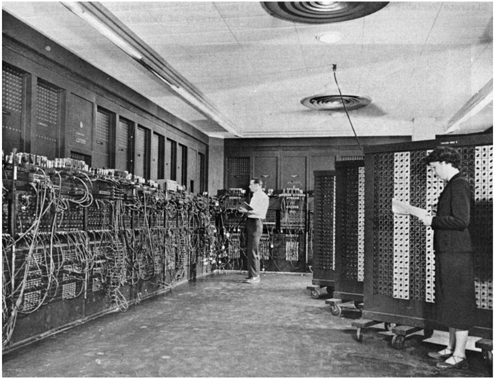


- His company, the Tabulating Machine Company, became International Business Machines (IBM), still the largest corporation in computing.
- However, true computing as Babbage envisioned it did not become practical for a century.

Outline

1. Algorithms
2. Early calculators
3. von Neumann Model
4. Credits

ENIAC (1946)



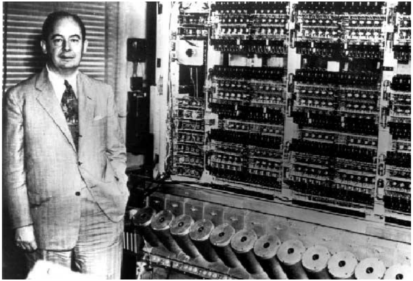
- ENIAC, short for Electronic Numerical Integrator And Computer was the first general-purpose electronic computer.
- ENIAC contained 17,468 vacuum tubes, 7,200 crystal diodes, 1,500 relays, 70,000 resistors, 10,000 capacitors and around 5 million hand-soldered joints. It weighed 27 tons.
- The basic machine cycle was 200 microseconds

ENIAC Architecture

- It was digital computer capable of being reprogrammed to solve a full range of computing problems.
- ENIAC was designed to calculate artillery firing tables for the U.S. Army's Ballistic Research Laboratory, but its first use was in calculations for the hydrogen bomb.
- ENIAC was not able to *store* a program in memory
    - Reprogramming was done by setting switches and re-wiring the machine
    - It could take up to 3 weeks!
    - Most of the programming was done by six women, now in the history of computers

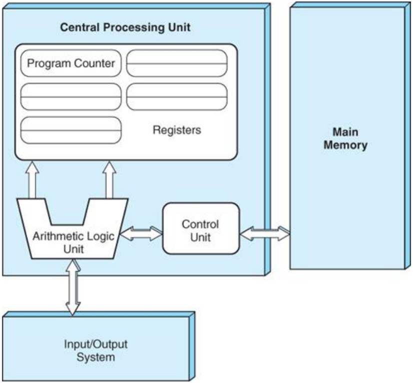
EDVAC

- Mathematician John von Neumann worked at ENIAC, and later proposed the general model known as von Neuman architecture, first implemented on EDVAC, and still used today



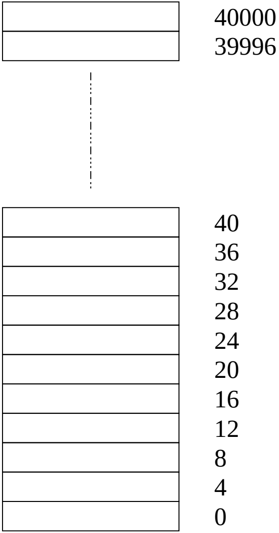
von Neuman Architecture

- von Neuman proposed to store the program in memory, together with data
- It performs a cycle where:
  - the processor first *loads* instructions from memory (fetching)
  - decodes them
  - if necessary loads the operands (data)
  - performs the operation
  - if necessary, stores results in memory



Memory

- The memory is organized as a sequential list of binary words
- In 32-bit architectures, a word corresponds to 4 bytes (32 bits)
- A processor can read one word at every cycle



Instruction set architecture

- The processor has an instruction set (i.e. a set of pre-defined commands, coded in binary)
    - This is hard-wired on the processor: different processors (Intel, Motorola, ARM, etc.) will have different instruction sets
    - They cannot communicate with each other
- A program is a set of binary words, each one codes a command, or its operands
- Here is an example of program, described in a symbolic language called assembly, which has a 1-1 relationship with the machine code

```
LD  R0, 0x5AF85C42
ADD R0, R1
```

Programs and data

- In the von Neumann machine, programs can be treated as data
  - they can be automatically produced by other programs
  - they can be stored on files on the hard disk, and later they can be loaded in memory
  - they can self-modify!
  - They can be modifies by other programs (program updates, virus, ...)

Outline

1. Algorithms
2. Early calculators
3. von Neumann Model
4. Credits

Credits

- Thanks to Michael Tobis for the historical slides and pictures
  - `http://webpages.cs.luc.edu/~mt/CS150/M1.html`