Philosophical Issues of Computer Science

An Introduction

Instructor: Viola Schiaffonati

February, 25th 2019
It’s not about specific tools and techniques ...
What the course *is* about

*It’s about conceptual and historical tools, critical analysis, original development of ideas, widening of the given view, looking at usual problems with different eyes, ...*

*... it’s about philosophy*
Philosophical Issues of Computer Science

- **Critical analysis** of some **key concepts** in computer science and engineering
  - Presentation of the **conceptual tools** to be used in the analysis (mainly philosophical analysis)
  - Discussion of **key concepts** and issues of computer science and engineering
- **Final project**
The structure of the course

- What are science, technology and engineering?
  - Philosophical and historical analysis of science, technology, and engineering
  - The philosophy of computer science/computing

- What is computer science? Foundational issues
  - The Universal Computer: The Road from Leibniz to Turing
  - Philosophical Issues of Computation
  - Experimental Computer Science
  - Computer simulations and experiments

- The philosophy of AI
  - Machine and thought: from Turing Test to embodied cognition
  - The physical system hypothesis
  - Minds and brains: the Chinese room argument
  - Computational models of consciousness
  - Invited lecture (Federico Cabitza)

- Computer and information ethics

- Final project
  - Paper assignment and discussion
  - How to choose a (good) topic and to write a (good) paper
  - Papers’ supervisions
Philosophy as a tool

- Philosophy as an **analytic** and **critical** discipline (not the history of philosophy)
- Centrality of **critical analysis** and **rational enquire**
- Philosophy of x as the study of the **fundamental assumptions** and **main goals** of any discipline x
- **Philosophical questions** for conceptual clarification
What is computer science /informatics?

- **Science of computers** and related phenomena (Newell, Perlis, Simon 1967)
- **Study** (and not science!) of **algorithms** and related phenomena (Knuth 1974)
- **Empirical study** (‘artificial science’) of phenomena related to computers (Newell e Simon 1976; Simon 1996)
- **Natural science** of **procedures** (not of computers and algorithms) (Shapiro 2001)
- Study of **computational processes** (Colburn 2004)
- **Synthetic engineering** discipline (Brooks 1996)
- Study of **information** (Hartmanin e Lin 1992)
Objects and perspectives

- Different **objects**
  - Computer (**hardware**)  
  - Algorithms, processes, procedures (**software**)  
  - **Information**

- Different **perspectives**
  - Computer science as a **science** or as other (**study, corpus of knowledge, discipline, ...**)
  - **Different names** (computer science, computing science, informatics, computer engineering, information science, ...)
"Computer science is in part [1] a **scientific discipline** concerned with the empirical study of a class of phenomena [...], in part [2] a **mathematical discipline** concerned with the formal properties of certain classes of abstract structures, and in part [3] a **technological discipline** concerned with the cost-effective design and construction of commercially and socially valuable products"

(Wegner 1976)
“The discipline of computing is the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation, and application. The fundamental question underlying all of computing is, ‘What can be (efficiently) automated?’”

(Denning et al. 1989)

- Roots both in mathematics (analysis) and engineering (design)
- Discipline with a theory, an experimental method, and engineering (different from natural sciences that are separated from their engineering sciences)
- Constant exchange between the scientific paradigm and the engineering one
Uniqueness of computer science (Colburn 2004)

- Natural sciences deal with models to test hypotheses explaining phenomena.
- Computer science realizes and manipulates non-physical models.
  - Computer science is the study of computational processes.
  - Computational processes are different from other processes as they are studied without any reference to their physical nature.
- Computer science realizes computational models of processes in the form of programs.
  - These models can be tested by executing the program and observing its behavior.
  - These models can be analyzed from a pure abstract point of view.
“Computer science is an empirical discipline. We would have called it an experimental science, but like astronomy, economics, and geology, some of its unique forms of observation and experience do not fit a narrow stereotype of the experimental method. None the less, they are experiments. Each new machine that is built is an experiment. Actually constructing the machine poses a question to nature; and we listen for the answer by observing the machine in operation and analyzing it by all analytical and measurement means available. Each new program that is built is an experiment. It poses a question to nature, and its behavior offers clues to an answer. Neither machines nor programs are black boxes; they are artifacts that have been designed, both hardware and software, and we can open them up and look inside. We can relate their structure to their behavior and draw many lessons from a single experiment.” (Newell and Simon 1976)
In 1960s and 1970s computer science became distinct from mathematics. Mathematical, empirical, and engineering methods. Whether computer science has to be mostly considered as a mathematical discipline, a branch of engineering, or as a scientific discipline.
Programs are **mathematical entities**

Purely **deductive reasoning** provided by the **formal methods** of theoretical computer science

Original motivation for a mathematical analysis of computation coming from **mathematical logic**

- **Hilbert’s problem**: could there be an algorithm, a procedure, for deciding of an arbitrary sentence of the logic whether it is provable?
From proofs of programs’ correctness to **methods managing** the **complexity** of those systems and evaluating their **reliability** (1970s)

- Emphasis on how **to produce** artefacts
- Production of the phenomena to be studied, those concerning **computational artefacts**
Both **formal methods** and **empirical testing** used to evaluate the correctness of computational artefacts.

Cs understood as **scientific discipline** in that it makes use of both **deductive** and **inductive probabilistic reasoning** to examine computational artefacts.
Practical info

- [http://home.deib.polimi.it/schiaffo/TFI/](http://home.deib.polimi.it/schiaffo/TFI/)
- Slides and reading materials (syllabus)

<table>
<thead>
<tr>
<th>Date</th>
<th>Topics</th>
<th>Slides and Reading Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Thursday February 28th</td>
<td>Science, paradigms and scientific revolutions</td>
<td></td>
</tr>
<tr>
<td>Thursday April 26th</td>
<td>NO CLASS (HOLIDAYS)</td>
<td></td>
</tr>
<tr>
<td>14. Monday April 29th</td>
<td>AI and Ethics</td>
<td>&quot;What is Computer Ethics?&quot; (J. Moore)</td>
</tr>
<tr>
<td>15. Thursday May 2nd</td>
<td>Invited lecture by Federico Cabitza (Università degli Studi di Milano Bicocca)</td>
<td></td>
</tr>
<tr>
<td>16. Monday May 6th</td>
<td>Final project assignment and discussion</td>
<td></td>
</tr>
</tbody>
</table>
Grading will be on the following basis

- 50% final essay (English or Italian)
- 50% oral discussion on the papers published on the course web-page (English or Italian)

You have to deliver your essay at least one week before the oral exam

- **Exam June 24** (TBC) 2019 – paper due June 17 2019
- **Exam July 9** (TBC) 2019 – paper due July 2 2019
More on practical info

- No prerequisite required
- Bibliography
  - Scientific papers available on the course web page
- Timetable
  - Mo 10:15-12:15
  - Thur 10:15-12:15
