

## PHILOSOPHY OF COMPUTER SCIENCE

The *Philosophy of Computer Science* is concerned with philosophical issues that arise from reflection upon the nature and practice of the academic discipline of computer science. It is a meta-discipline that employs the tools of philosophical analysis to uncover the fundamental assumptions that underpin the discipline. In this sense, it parallels the philosophies of mathematics and physics in being a parasitic philosophical discipline. But while physics and mathematics are well established disciplines, with broad agreement concerning their subject matter, computer science is new with its very nature unfixed and unclear. Below we indicate a few of the central questions.

- What is the nature of the Church-Turing thesis? Is it definitional or empirical in content?
- How is a programming language defined and fixed? What role does a semantic definition play? Does it have to be a formal abstract definition? What conceptual issues underpin the notions of compositionality and full abstraction?
- What is a specification? How does it differ from a definition? What is the difference between a specification and a program
- What does it mean to say that a program/software is correct?
- What is the role of the philosophical concept(s) of information in the philosophy of computer science?
- What is abstraction in computer science? Is there just one form of it? How is it related to abstraction in mathematics? How are the notions of specification, abstraction and implementation connected?
- Is there a distinctive form of reasoning that might be called computational reasoning? How, if at all, does it differ from mathematical reasoning?
- What kind of activity is academic computer science? Is it science, engineering or mathematics? Is it best seen as the science of processes and computations?
- What is the role of mathematics in computer science? What is the role of experimentation? Are the standards of experimentation and justification as rigorous as they are in the other sciences?
- What kinds of things are programs? Are they abstract or physical? What is it for two programs to be equal?
- How is the computational turn influencing the methodology of mainstream science? Are computational experiments standing proxy for physical ones?
- What are the central questions in the philosophy of artificial intelligence? What are the conceptual limits of computational simulation? What is the significance of the Turing test?
- Aside from the role of specification in software engineering are there any philosophical questions that arise in the industrial and commercial applications of computer science? Are the only other ones ethical and legal?
- Are Wittgenstein's views on rule following central to our understanding of computation?

Some of these questions may not be philosophically substantial, and there are many other candidates, but discovering the central issues is a good part of the research task. Much contemporary work operates in the other direction being an attempt to apply

computational ideas to philosophy itself—so called *computational philosophy*. Finally, the study is different to the philosophy of information both in terms of its central subject matter and its overall objectives. The later is much broader in its ambitions and seeks to make some notion of information a central concept of philosophy.