Philosophy of Computer Science:

PoC Meets AI and Law

(Roundtable Discussion)

Organized by

Rainhard Bengez, TU Muenchen, Germany (bengez@cvl-a.tum.de)

Raymond Turner, University of Essex, UK (turnr@essex.ac.uk)

Any Congress delegate is welcome to attend and should just turn up at the time, space permitting. The event will consist of short invited talks followed by a chaired discussion.

(The event is in part thematically related to, and will follow consecutively from, the Symposium on History and Philosophy of Programming. Both are part of the AISB/IACAP 2012 World Congress, http://events.cs.bham.ac.uk/turing12/)

TOPICS:

Computer science is one of the most influential disciplines of the present day. It provides a substantial body of new intellectual material that concerns itself with the nature of algorithms and computations, the computational representation of abstract concepts, the study of computational complexity, the design, definition and implementation of programming languages, the nature of specification, the study of processes, agents and states, etc. etc. But it underpins more than this. It has introduced whole new areas of mathematical study. In particular, new areas of logic such as substructural and dynamic logic have been developed as a result of the analysis of computational notions. It has also modified the practice mainstream science. And this influence goes beyond the use of computational tools. The computational turn has infected the way we think about fundamental science. In physics, chemistry, biology, psychology, economics, and cognitive science, not only are computers used to do complex computations but they are impinging upon our very notion of experiment. All of this is in addition to the impact on our everyday existence: every aspect of our lives has been changed by the digital revolution which has its intellectual roots in 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. Indeed, the philosophy of computer science is in its infancy. Much contemporary work operates in the other direction being an attempt to apply computational ideas to philosophy itself–so called *computational philosophy*. It is also not to be identified with the philosophy of artificial intelligence which only forms a part of the philosophy of computer science. 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.

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? Does it apply to physical machines? What is a physical computation?
- 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 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?
- Is computer science a genuine science; is it the science of processes and computations?
- What kinds of things are programs? Are they abstract or physical?
- 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?
- 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.