

Student Association for the Philosophy of Computation at Berkeley

Jongmin Jerome Baek, Alexandra Tran, Nihar Dalal
Faculty Advisors: John DeNero & Seth Yalcin

March 2017

1 Why A New Student Association?

Computer science is no more about computers than astronomy is about telescopes. (Edsger Dijkstra)

When the word “computer science” is uttered, few people think of philosophy or poetry. Similarly, few people know that the “father of computer science”, Alan Turing, was also a philosopher, or that the “mother of computer science”, Ada Lovelace, was also a poet. Amongst the glamour and wads of cash at nearby Silicon Valley, the history of computer science is rarely discussed, and the motivations of major thinkers frequently forgotten. This is a shame, because what motivated these thinkers were in fact not money nor glamour but some of the most universal, basic, visceral questions such as

I propose to consider the question, can machines think? (Alan Turing)

[I will have] the most *harmoniously* disciplined troops; – consisting of vast *numbers*, and marching in irresistible power to the sound of *Music*. Is not this very mysterious?...But then, *what* are these *Numbers*? There is a riddle – (Ada Lovelace, at her deathbed)

The neglect of philosophical motivations is doubly a shame because, decades since its founders’ deaths, computer science has steadily advanced its light, and we are now in a much better position to start answering these questions! Some ideas from computer science, such as uncomputability, P vs. NP, and quantum computing, have slowly seeped into such diverse areas of thought as child development, language, evolution, culture, epistemology, metaphysics, morality, and on and on. But the idea that computer science has philosophical roots and implications has been slow to be smuggled into the cultural zeitgeist.

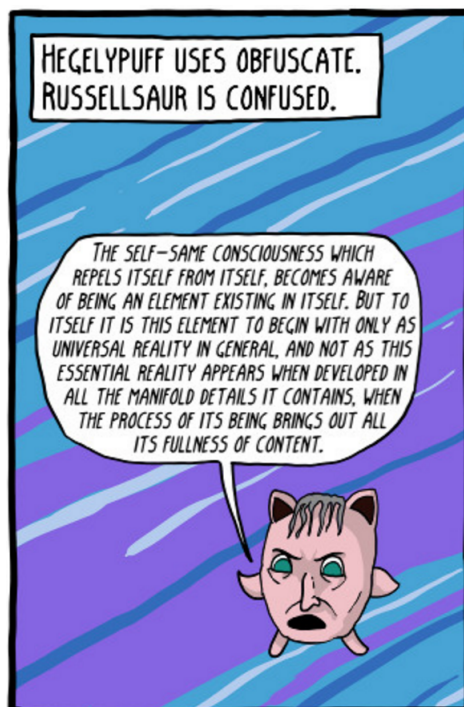
We lament not so much the lack of discussion on the philosophy of computation, but the lack of *awareness* that such discussions are possible. Because of the lack of awareness, computer science is instead frequently viewed as materialistic, elitist, and a means to an end. We want to foster an academic culture where a sizable portion of students are actively talking about the philosophy of computation, so that computer science becomes more of an end in itself. We believe such a culture will not only provide a major motivation to study computer science for many more people, but also make the department more welcoming to people of diverse backgrounds, diverse viewpoints, and diverse areas of focus and competence.

2 Why Philosophy of Computation?

“Computer science” is a bit of a misnomer; maybe it should be called “quantitative epistemology.” It’s sort of the study of the capacity of finite beings such as ourselves to learn mathematical truths.

(Scott Aaronson, Berkeley PhD 2004)

But what’s so special about the philosophy of computation, and why should anyone care? After all, philosophy is abstruse and elitist, and has nothing to do with our lives anyway.



We believe that computation as a lens for asking philosophical questions is unique in the aspect that it is quantitative, formal, and exact. So anyone with a training in computer science and computational thinking can easily grasp its ideas. In fact, many important ideas in theoretical computer science, like P vs. NP, are so intuitive that someone with absolutely no knowledge in the subject could understand its ideas. Moreover, because the lens is mathematical and thus universal, diverse cultural viewpoints can be respected and understood – a major problem in philosophy, currently plagued by a glaring diversity problem. Another compelling reason is that, probably sometime soon, *everybody* will have to deal with the philosophical implications of computation: some seventy years since Turing posed his question, *Can machines think?*, an answer is now approaching us at a seemingly relentless pace. And then there’s that fun part, of course, that the computational lens leads to completely unexpected (potential) answers to our philosophical questions, answers that threaten to turn our intuitions about the world completely upside down.

3 Structure of the Association

The basic idea is to read an essay or a few chapters of a book about things like P vs. NP, Gödel's incompleteness theorem, or how to make artificial intelligence that doesn't try to destroy the human race, and then meet for an hour or two each week to discuss what we read, passionately and respectfully. We would be happy to host researchers and professors from various institutions to come and speak. Motivated members may write essays and come up with new theories, and will have a platform to present their ideas.

4 Tentative Topics and Readings

4.1 Background reading

The Information: a History, a Theory, a Flood, James Gleick
Logicomix: An Epic Search for Truth, Doxiadis & Papadimitriou

4.2 Computational Complexity

Why Philosophers Should Care About Computational Complexity, Scott Aaronson

4.3 Quantum Computation

Quantum Computing Since Democritus, Scott Aaronson
The Ghost in the Quantum Turing Machine, Scott Aaronson

4.4 Turing, Gödel, self-reference and uncomputability

Gödel, Escher, Bach, Douglas Hofstadter

4.5 Strong vs. Weak AI, AI ethics, computational morality

Minds, Brains, and Programs, John Searle
Superintelligence, Nick Bostrom
Universal Knowledge-Seeking Agents, Laurent Orseau

4.6 Algorithmic information theory

The Discovery of Algorithmic Probability, Ray Solomonoff
Probably Approximately Correct, Leslie Valiant

4.7 Evolution, memes, and memology

From Bacteria to Bach and Back, Daniel Dennett

4.8 Computational culture, metaphor, and child development

The Development of Thought, Jean Piaget
Metaphors We Live By, George Lakoff
The Analects, Confucius