Course Outline: Computers and Computations

Dr. Neil Barton
neil.barton@uni-konstanz.de

Office Hours/Sprechstunde: By Arrangement/nach Vereinbarung

1 Course description and objectives

1.1 Course description

It is likely that right now you are reading this course description on some sort of computing device—a machine that, given certain inputs, is able to calculate and produce desired outputs. It's widely acknowledged that the idea of *computation* and the machines that implement them are enormously important for understanding contemporary life. But behind all this wizardry there are some really fundamental philosophical questions: What is a computation? Are there limits to what we can do with them? What kinds of responsibility are there when things go wrong? This course will explore these questions and others in detail. We'll start with the notion of computation as it appears in the work of Turing, and we'll discuss the Halting Problem and its implications. We'll then consider different versions of the Church-Turing thesis, and whether there are counterexamples. We'll then discuss some arguments relating computers to theories of mind, and consider the Lucas-Penrose argument that the mind is not a machine. In the second half of the course, we'll start by looking at ways of analysing time and space restrictions and the notion of a *feasible* computation. We'll provide a brief discussion of how quantum computing might figure into all of this, before finishing by examining the idea of ethical responsibility and computation.

1.2 Prerequisites

A first course in logic (or knowledge of the basics of propositional and predicate logic) will greatly help students follow discussion. Students from outside philosophy are very welcome to take the course.

1.3 Objectives

By the end of this course you should be able to:

(i) Understand how we formalise notions of computation (e.g. Turing computability).

(ii) Understand how different complexity classes (e.g. P and NP) are defined, and what their philosophical relevance is.

(iii) Understand and use central concepts in the philosophy of computation (e.g. the Church-Turing Thesis).

(iv) Use these concepts to outline various important positions.

(v) Show familiarity with texts in the philosophy of computation.

(vi) Evaluate these texts, positions, and ideas.

(vii) Construct your own rigorous arguments for specific positions.

2 Temporal and spatial location

- **Asynchronous part:** There will be a YouTube video available on my channel[1] under the 'Computers and Computations' playlist, by Saturday morning before the seminar. You should watch this before attending.
- **Synchronous part:** We will discuss the set text(s) and video in a seminar held via Big Blue Button on Mondays between 11:45 and 13:15 (I will make the Big Blue Button room available shortly before 11:45). Since you will already have watched a video (the 'lecture' portion), you are only required to attend between 12:00 and 13:00. I will, however, be online at 11:45 and leave at 13:15 so we can chat more informally and I can answer other questions.
- **Office Hour/Sprechstunde:** By Arrangement/nach Vereinbarung.

[1] See `https://www.youtube.com/channel/UCDsaup9L-AotqI-8M3NI9_w`.

3 Assessment

- MA students:

  Seminarschein 6 ECTS:

  - 60% One paper of 2000–2500 words.
  - 10% attendance/contribution in class. In order to get this extra 10%, you can miss at most two seminars (but it is otherwise ungraded).
  - 30% weekly questions: Between one and three *reasoned* (see below) questions of 200–500 words total. Again, you can miss at most two weeks of questions to get this extra credit, but the questions themselves are ungraded.

  Hausarbeitschein 6 ECTS

  - 90% One paper of 5000–6000 words.
  - 10% contribution and attendance (again, two possible missed classes are allowed for the extra credit).

- BA students

  Hausarbeitschein 6 ECTS

  - 60% One paper of 1500–2000 words.
  - 10% attendance/contribution in class. In order to get this extra 10%, you can miss at most two seminars (but it is otherwise ungraded).
  - 30% weekly questions: Between one and three *reasoned* (see below) questions of 200–500 words total. Again, you can miss at most two weeks of questions to get this extra credit, but the questions themselves are ungraded.

  Hausarbeit 6 ECTS

  - 90% One paper of 4000–5000 words
  - 10% contribution and attendance (again, two possible missed classes are allowed for the extra credit).

- BA/MA students can be assessed through either method inclusive (i.e. you can do both). Lehramtsstudierende can do *either* Seminarschein or Hausarbeit but *not* both.

- **Reasoned Questions:** The questions (at least one) each week are meant to be *reasoned* and should total between 200 and 500 words.

  - **Example 1.** (Bad question.) I don't understand what Turing means by computation?
  - **Example 2.** (Good question.) I am confused by what Turing means by computation. On the one hand, he seems to be talking about human computers, as in humans that are carrying out some symbolic manipulation(s). But on the other hand, he models this computation in terms of a machine that can scan cells and move on a tape. How do we know that the machine-based model is a good representation of what humans do? Is Turing just interested in one kind of computation, or are there different kinds at play?
  - **Important:** Submission of non-reasoned questions will count as a *missed* assignment.

- **Submission of questions:** Through ILIAS (there will be an Übung under the relevant Sitzung).
- **Submission of essay:** Via e-mail: neil.barton@uni-konstanz.de.

3.1 Important deadlines

Weekly: (Seminarschein only) Deadline for questions: 23:55, Friday before the seminar.

14. March 2021: Deadline for Seminarschein paper (23:55).

31. March 2021: Deadline for Hausarbeitschein paper (23:55).

4 Course Details

4.1 Materials

Selected papers and book chapters: Available either from the publisher, through JSTOR, and/or ILIAS.

Lecture notes: Available immediately after lectures on ILIAS.

Lecture videos: On my YouTube channel.[2] **Note:** Seminars are *not* recorded.

[2]See `https://www.youtube.com/channel/UCDsaup9L-AotqI-8M3NI9_w`

4.2 Lecture Schedule

Week 1. *2. Nov.* Introduction and outline of the course

Week 2. *9. Nov.* Historical accounts of computation

Week 3. *16. Nov.* Turing computability

Week 4. *23. Nov.* The Halting Problem

Week 5. *30. Nov.* Other models of computation

Week 6. *7. Dec.* Church-Turing Theses

Week 7. *14. Dec.* The Lucas-Penrose Argument

Week 8. *21. Dec.* Computational functionalism

24. December 2020 – 6. January 2021: No lectures.

Week 9. *11. Jan.* Computational Complexity: The Zoo and Feasibility

Week 10. *18. Jan.* Quantum Computation

Week 11. *25. Jan.* Ethics and computations: Who is responsible?

Week 12. *1. Feb.* Computing the standard model of arithmetic

Week 13. *8. Feb.* Digital and Analog Computation

4.3 Some Possible Essay Questions

1. To what extent was modern computability theory (i.e. post Church and Turing) anticipated by their predecessors?
2. Is every effective procedure Turing computable?
3. Are all models of computation fundamentally equivalent?
4. Explain the Halting Problem and critically examine one way in which it is significant.
5. Can the mind be modelled by a machine?
6. Is there a meaningful distinction between software and hardware?
7. Are computer programmes abstract or physical artefacts?

8. Are feasible computations exactly those that are computable in polynomial time?

9. Do notions of computation allow us to single out the standard model of arithmetic?

10. What is a quantum computer? Assess one way in which they are claimed to be significant.

11. Are software developers morally responsible for bugs in their software?

12. What is the difference between digital and analogue computation?

13. N.B. I will accept (and indeed encourage!) coming up with your own essay questions. However, you **must** have cleared your question with me at least 2 weeks in advance of essay submission.

4.4 Readings for each week

The course is based around a cluster of issues facing our philosophical and mathematical understanding of computations. This includes its historical development, as well as some themes that have arisen more recently.

I have marked the core reading for each week with an asterisk (this is the reading that you accomplish before class). I will post the core reading on ILIAS. Other readings are optional—you are certainly not required to read everything for every week. Nonetheless, if you wish to write a paper on the topic then the optional readings will give you additional background, should you want it. For the extra keen, further references are available in the course notes that I will make available on ILIAS after class each week.

Week 1. *2. Nov.* Introduction and outline of the course.

- (∗) This course outline.

Week 2. *9. Nov.* Historical accounts of computation

- (∗) O'Regan, Gerard (2012) *A Brief History of computing*. London: Springer. Ch 1 and Ch 2, until §2.4 (inclusive), pp. 1–47

Week 3. *16. Nov.* Turing computability

- (∗) Penrose, Roger (1989) *The Emperor's New Mind*. Oxford University Press. pp. 30–57.
- Dick, Stephanie. (2013). Machines Who Write. IEEE Annals of the History of Computing 35(2), 88-87.
- For the technical background: Boolos, G., Burgess, J., and Jeffrey, R. (2007). Computability and Logic (5th ed.). Cambridge: Cambridge University Press. Ch 3. pp.

Week 4. *23. Nov.* The Halting Problem

- (∗) Penrose, Roger (1989) *The Emperor's New Mind*. Oxford University Press. pp. 57–66.
- Hopcroft, John (1984). 'Turing Machines'. *Scientific American*, 250(5), 86-E9.
- Technical background: Boolos, G., Burgess, J., and Jeffrey, R. (2007). Computability and Logic (5th ed.). Cambridge: Cambridge University Press. Ch 4.
- Fun: Pullum, Geoffrey K. (2000) 'Scooping the Loop Snooper', *Mathematics Magazine*, October 2000, pp. 319-320. An improved version is available here: `http://www.lel.ed.ac.uk/~gpullum/loopsnoop.html`

Week 5. *30. Nov.* Other models of computation

- (∗) Boolos, G., Burgess, J., and Jeffrey, R. (2007). Computability and Logic (5th ed.). Cambridge: Cambridge University Press. Chs. 5–6, pp. 45–72.

Week 6. *7. Dec.* The Church-Turing Thesis and its relatives

- (∗) Copeland, B. Jack, "The Church-Turing Thesis", The Stanford Encyclopedia of Philosophy (Summer 2020 Edition), Edward N. Zalta (ed.), `https://plato.stanford.edu/archives/sum2020/entries/church-turing/`
- Shapiro, Stewart (2013) 'The Open Texture of Computability', in B. J. Copeland and C. J. Posy and O. Shagrir Eds *Computability: Turing, Gödel, Church, and Beyond*, MIT Press, pp. 153–181.

Week 7. *14. Dec.* The Lucas-Penrose Argument

- (∗) Shapiro, Stewart (2003). Mechanism, truth, and Penrose's new argument. *Journal of Philosophical Logic* 32 (1):19-42.

- Koellner, Peter (2018). On the Question of Whether the Mind Can Be Mechanized, I: From Gödel to Penrose. *Journal of Philosophy* 115 (7):337-360.
- Koellner, Peter (2018). On the Question of Whether the Mind Can Be Mechanized, II: Penrose's New Argument. *Journal of Philosophy* 115 (9):453-484.

Week 8. *21. Dec.* Computational functionalism

- (∗) Putnam, Hilary (1988) *Representation and Reality*, MIT Press. Ch. 5, pp. 73–89.
- Chalmers, David 'Does a rock implement every finite-state automaton?'. *Synthese* 108, 309–333 (1996).

24. December 2020 – 6. January 2021: No lectures.

Week 9. *11. Jan.* Computational Complexity: The Zoo and Feasibility

- (∗) Dean W. (2016) Squeezing Feasibility. In: Beckmann A., Bienvenu L., Jonoska N. (eds) Pursuit of the Universal. CiE 2016. Lecture Notes in Computer Science, vol 9709. Springer, Cham.
- (∗) hackerdashery. P vs. NP and the computational complexity zoo. `https://www.youtube.com/watch?v=`
- Walter Dean, Computational Complexity Theory and the Philosophy of Mathematics, Philosophia Mathematica, Volume 27, Issue 3, October 2019, pp. 381–439,
- Aaronson, Scott 'Why Philosophers Should Care About Computational Complexity' in J. Copeland, C. Posy, and O. Shagrir (eds.), *Computability: Turing, Gödel, Church, and Beyond*, pp. 261–328, Cambridge, Massachusetts: MIT Press.

Week 10. *18. Jan.* Quantum Computation

- (∗) Hagar, Amit and Michael Cuffaro, 'Quantum Computing', The Stanford Encyclopedia of Philosophy (Winter 2019 Edition), Edward N. Zalta (ed.), `https://plato.stanford.edu/archives/win2019/entries/qt-quantcomp/`.
- Aaronson, Scott (2013) *Quantum Computing Since Democritus*, Cambridge, England: Cambridge University Press. chs. 9, 10.

Week 11. *25. Jan.* Ethics and computations: Who is responsible?

- (∗) Edgar, Stacey (1997) *Morality and Machines: Perspectives on Computer Ethics*, Jones and Bartlett. Ch. 8. pp. 293–322.

Week 12. *1. Feb.* Computing the standard model of arithmetic

- (∗) Halbach, Volker and Horsten, Leon (2005) 'Computational Structuralism' *Philosophia Mathematica*, Volume 13, Issue 2, June 2005, Pages 174–186,
- Button, Tim and Smith, Peter (2012) 'The Philosophical Significance of Tennenbaum's Theorem', *Philosophia Mathematica*, Volume 20, Issue 1, February 2012, Pages 114–121,

Week 13. *8. Feb.* Digital and Analog Computation

- (∗) Papayannopoulos, P. (2020). Computing and modelling: Analog vs. Analogue. Studies in History and Philosophy of Science Part A, 83, 103–120.