Search Wikipedia Search

Philosophy of computer science

<u>▼A</u> 17 languages <u>∨</u>

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

Not to be confused with Computational philosophy.

The **philosophy of computer science** is concerned with the philosophical questions that arise within the study of computer science. There is still no common understanding of the content, aims, focus, or topics of the philosophy of computer science, [1] despite some attempts to develop a philosophy of computer science like the philosophy of physics or the philosophy of mathematics. Due to the abstract nature of computer programs and the technological ambitions of computer science, many of the conceptual questions of the philosophy of computer science are also comparable to the philosophy of science, philosophy of mathematics, and the philosophy of technology. [2]

Overview [edit]

Many of the central philosophical questions of computer science are centered on the logical, ethical, methodological, ontological and epistemological issues that concern it. [3] Some of these questions may include:

- What is computation?
- Does the Church-Turing thesis capture the mathematical notion of an effective method in logic and mathematics?^{[4][5]}
- What are the philosophical consequences of the P vs NP problem?
- What is information?

Church—Turing thesis [edit]

The Church–Turing thesis and its variations are central to the theory of computation. Since, as an informal notion, the concept of effective calculability does not have a formal definition, the thesis, although it has near-universal acceptance, cannot be formally proven. The implications of this thesis is also of philosophical concern. Philosophers have interpreted the Church–Turing thesis as having implications for the philosophy of mind.^{[6][7]}

P versus NP problem [edit]

The P versus NP problem is an unsolved problem in computer science and mathematics. It asks whether every problem whose solution can be verified in polynomial time (and so defined to belong to the class NP) can also be solved in polynomial time (and so defined to belong to the class P). Most computer scientists believe that $P \neq NP$. [8][9] Apart from the reason that after decades of studying these problems no one has been able to find a polynomial-time algorithm for any of more than 3000 important known NP-complete problems, philosophical reasons that concern its implications may have motivated this belief.

For instance, according to Scott Aaronson, the American computer scientist then at MIT:

If **P** = **NP**, then the world would be a profoundly different place than we usually assume it to be. There would be no special value in "creative leaps", no fundamental gap between solving a problem and recognizing the solution once it's found. Everyone who could appreciate a symphony would be Mozart; everyone who could follow a step-by-step argument would be Gauss. [10]

See also [edit]

- Computer-assisted proof: Philosophical objections
- Philosophy of artificial intelligence
- Philosophy of information
- Philosophy of mathematics
- Philosophy of science
- Philosophy of technology

$References \ \ [\ {\tt edit}\]$

- 1. ^ Tedre, Matti (2014). The Science of Computing: Shaping a Discipline. Chapman Hall.
- 2. ^ Turner, Raymond; Angius, Nicola (2020), "The Philosophy of Computer Science" , in Zalta, Edward N. (ed.), *The Stanford Encyclopedia of Philosophy* (Spring 2020 ed.), Metaphysics Research Lab, Stanford University, retrieved 2020-05-21
- 3. ^ Turner, Raymond (January 2008). "The Philosophy of Computer Science" ∠". Journal of Applied Logic. 6 (4): 459. doi:10.1016/j.jal.2008.09.006 ∠". hdl:2434/807648 ∂ via ResearchGate.
- 4. ^ Copeland, B. Jack. "The Church-Turing Thesis" ☑. Stanford Encyclopedia of Philosophy.
- machines?" 년.

5. ^ Hodges, Andrew. "Did Church and Turing have a thesis about

- Copeland, B. Jack (November 10, 2017). "The Church-Turing Thesis" ∠. In Zalta, Edward N. (ed.). Stanford Encyclopedia of Philosophy.
- 7. ^ For a good place to encounter original papers see Chalmers, David J., ed. (2002). *Philosophy of Mind: Classical and Contemporary Readings*. New York: Oxford University Press. ISBN 978-0-19-514581-6. OCLC 610918145 ...
- 8. ^ William I. Gasarch (June 2002). "The P=?NP poll" (PDF). SIGACT News. 33 (2): 34–47. CiteSeerX 10.1.1.172.1005 (a). doi:10.1145/564585.564599 . S2CID 36828694 . Retrieved 26 September 2018.
- 9. ^ Rosenberger, Jack (May 2012). "P vs. NP poll results" ☑. Communications of the ACM. **55** (5): 10.
- 10. ^ "Shtetl-Optimized » Blog Archive » Reasons to believe" ☑. 4 September 2006. Retrieved 2021-09-16.

Further reading [edit]

- Matti Tedre (2014). The Science of Computing: Shaping a Discipline. Chapman Hall.
 Scott Agronson, "Why Philosophers Should Care About Computational Complexity of Agronson."
- Scott Aaronson. "Why Philosophers Should Care About Computational Complexity ". In Computability: Gödel, Turing, Church, and beyond.
- Timothy Colburn. Philosophy and Computer Science. Explorations in Philosophy. M.E. Sharpe, 1999. ISBN 1-56324-991-X.
 A.K. Dewdney, New Turing Omnibus: 66 Excursions in Computer Science.
- A.K. Dewdney. New Turing Omnibus: 66 Excursions in Computer Science
 A.K. Dewdney. New Turing Omnibus: 66 Excursions in Computer Science
- Luciano Floridi (editor). The Blackwell Guide to the Philosophy of Computing and Information, 2004.
- Luciano Floridi (editor). Philosophy of Computing and Information: 5 Questions. Automatic Press, 2008.
 Luciano Floridi (editor). Philosophy and Computing: An Introduction. Routledge, 1999.
- Luciano Floridi. *Philosophy and Computing: An Introduction*, Routledge, 1999.
- Christian Jongeneel. The informatical worldview, an inquiry into the methodology of computer science.
 Jan van Leeuwen. "Towards a philosophy of the information and computing sciences", NIAS Newsletter 42, 2009.
- Jan van Leeuwen. "Iowards a philosophy of the information and computing sciences", NIAS Newsletter 42, 2009
 Moschovakis V (2001) What is an algorithm? In Enquist B, and Schmid W, editors Mathematics unlimited 20
- Moschovakis, Y. (2001). What is an algorithm? In Enquist, B. and Schmid, W., editors, Mathematics unlimited 2001 and beyond, pages 919–936. Springer.
- Alexander Ollongren, Jaap van den Herik. Filosofie van de informatica. London and New York: Routledge, 1999. ISBN 0-415-19749-X

Tedre, Matti (2014), The Science of Computing: Shaping a Discipline ☑, CRC Press, ISBN 9781482217698 Taylor and Francis.

• Ray Turner and Nicola Angius. "The Philosophy of Computer Science ". Stanford Encyclopedia of Philosophy.

Matti Tedre (2011). Computing as a Science: A Survey of Competing Viewpoints

∴ Minds & Machines 21, 3, 361–387.

• Ray Turner. Computational Artefacts-Towards a Philosophy of Computer Science. Springer. [1] 🗹

External links [edit]

V • T • E

- The International Association for Computing and Philosophy
- Philosophy of Computing and Information at Philosophy of Computation at Parkalaua

 Philosophy of Computation at Parkalaua
- Philosophy of Computation at Berkeley ∠
 Rapaport, William J. (2020-07-27), "Philosophy of Computation at Berkeley ∠
- Rapaport, William J. (2020-07-27). "Philosophy of Computer Science (draft version)" (PDF). Archived from the original (PDF) on 2021-10-26.

Computer science

Philosophy of science [show]

Category: Philosophy of computer science

This page was last edited on 4 September 2025, at 13:43 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike 4.0 License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.



show

