DOI:10.1145/2908733

Jack Copeland, Eli Dresner, Diane Proudfoot, and Oron Shagrir

Viewpoint
Time to Reinspect the Foundations?

Clarifying the logico-mathematical foundations of computer science.

THE THEORY OF computability was launched in the 1930s, by a group of logicians who proposed new characterizations of the ancient idea of an algorithmic process. The most prominent of these iconoclasts were Kurt Gödel, Alonzo Church, and Alan Turing. The theoretical and philosophical work that they carried out in the 1930s laid the foundations for the computer revolution, and this revolution in turn fueled the fantastic expansion of scientific knowledge in the late 20th and early 21st centuries. Thanks in large part to these groundbreaking logico-mathematical investigations, unimagined number-crunching power was soon boosting all fields of scientific enquiry. (For an account of other early contributors to the emergence of the computer age see Copeland and Sommaruga.[9])

The motivation of these three revolutionary thinkers was not to pioneer the disciplines now known as theoretical and applied computer science, although with hindsight this is indeed what they did. Nor was their objective to design electronic digital computers, although Turing did go on to do so. The founding fathers of computability would have thought of themselves as working in a most abstract field, far from practical computing. They sought to clarify and define the limits of human computability in order to resolve open questions at the foundations of mathematics.

The 1930s revolution was a critical moment in the history of science: ideas devised at that time have become cornerstones of current science and technology. Since then many diverse computational paradigms have blossomed, and still others are the object of current theoretical enquiry: massively parallel and distributed computing, quantum computing, real-time interactive asynchronous computing, hypercomputing, nano-computing, DNA computing, neuron-like computing, computing over the reals, and computing involving quantum random-number generators. The list goes on ... (for example, see Cooper et al.,[3] and recent issues of *Natural Computing* and *International Journal of Unconventional Computing*). Few of these forms of computation were even envisaged at the time of the 1930s analysis of computability—and yet the ideas forged then are still typically regarded as constituting the very basis of computing.

So here's the elephant in the room. Do the concepts introduced by the 1930s pioneers provide a logico-mathematical foundation for what we call computing today, or do we need to overhaul the foundations in order to fit the

CREDIT TK

21st century? Much work has been devoted in recent years to analysis of the foundations and theoretical bounds of computing. However, the results of this diverse work—carried out by computer scientists, mathematicians, and philosophers—does not so far form a unified and coherent picture. It is time for the reexamination of the logico-mathematical foundations of computing to move center stage. Not that we should necessarily expect an entirely unified picture to emerge from these investigations, since it is possible that there is no common thread uniting the very different styles of computation countenanced today. Perhaps investigation will conclude that nothing more than 'family resemblance' links them.

A question pressed by foundational revisionists is: How are the bounds of computability, as delineated by the 1930s pioneers—bounds that theoretical computer science has by and large simply inherited and enshrined in the textbooks—related to the bounds of *physical* computing? The famous Church-Turing thesis delineates the bounds of computability in terms of the action of a Turing machine. But could there be mathematical functions that are physically computable and yet not Turing-machine computable? Various physical processes have been proposed that allegedly allow for computation beyond Turing-machine computability, appealing to physical scenarios that involve, for example, special or general relativity[11,14,16,17,19]; see Davis[10] for a critique of the whole idea of super-Turing computation. The possibility of super-Turing computation or hypercomputation is not ruled out by the Church-Turing thesis, once the latter is understood as originally intended, namely as an analysis of the bounds of what is computable by an idealized *human* computer—a mathematical clerk who works by rote with paper and pencil.[5–7,21–23] The limits of a human computer don't necessarily set the limits of every conceivable physical process (for example, see Pour El and Richards[18]), nor the limits of every conceivable machine. So it is an open question whether the Turing-machine model of computation captures the physical, or even the logical, limits of machine computability.

The Turing machine has also traditionally held a central place in complexity theory, with the so-called 'complexity-theoretic' Church-Turing[2] or 'extended' Church-Turing thesis[1] asserting that there is always at most a polynomial difference between the time complexity of any reasonable model of computation and that of a probabilistic Turing machine. (A deterministic version of the thesis is also known as the Cobham-Edmonds thesis.[13]) This traditional picture is today under threat, when some propose counterexamples to the extended Church-Turing thesis. Bernstein and Vazirani[2] gave what they called 'the first formal evidence' that quantum Turing machines violate the extended Church-Turing thesis. Shor's quantum algorithm for prime factorization is arguably another counterexample.[20] All this is highly controversial, but once again the signs are that we should take a systematic look at the foundations.

The most fundamental question of all is, of course, what *is* computation?

The most fundamental question of all is, of course: what *is* computation? Some will argue that the Turing-machine model gives an adequate answer to this question. But, given the enormous diversity in the many species of computation actually in use or under theoretical consideration, does the Turing-machine model still capture the nature of computation? For example, what about parallel asynchronous computations? Or interactive, process-based computations?[15,24] Now that such diversity is on the table, it may be that the Turing-machine model no longer nails the essence of computation. If so, is there any sufficiently flexible and general model to replace it? Some of the authors of this Viewpoint would bet on Turing's model, while the others would not. Either way this question is central.

Turning to the mysterious computer in our skulls … just as the limits of an idealized human clerk working by rote do not necessarily dictate the limits of computing hardware, nor do they necessarily dictate the limits of the human brain. For example, do human beings qua creative mathematicians carry out brain-based computations that transcend the limits of human beings qua mathematical clerks? The question is not merely philosophical: it seems highly likely that the brain will prove to be a rich source of models for new computing technologies. This gives us yet more reason to return to the writings of the 1930s pioneers, since both Gödel and Turing appear to have held that mathematical thinking possesses features going beyond the classical Turing-machine model of computation (see Copeland and Shagrir[8]).

In 2000, the first International Hypercomputation Workshop was held at University College, London. Now there is a growing community of hundreds of researchers in this and allied fields, communicating results at conferences (for example, 'Computability in Europe,' 'Theory and Applications of Computing,' and 'Unconventional Computing') and in journals (for example, *Applied Mathematics and Computation*, *Theoretical Computer Science*, *Computability*, and *Minds and Machines*). We hope this is only the beginning. If it was important to clarify the logico-mathematical foundations of computing in the 1930s, when computer science was no more than a gleam in Turing's eye, how much more important it seems today, when computing technology is diversifying and mutating at an unprecedented rate. Via the great pioneers of electronic computing (such as Freddie Williams, Tom Kilburn, Harry Huskey, Jay Forrester, John von Neumann, Julian Bigelow, and of course Turing himself) the 1930s analysis of computation led to the modern computing era. Who knows where a 21st-century overhaul of that classical analysis might lead.

References

1. Aharonov, D. and Vazirani, U.V. Is quantum mechanics falsifiable? A computational perspective on the foundations of quantum mechanics. In B.J. Copeland, C. Posy, and O. Shagrir, Eds., *Computability: Gödel, Turing, Church and Beyond*, MIT Press, Cambridge, MA, 2013.
2. Bernstein, E. and Vazirani, U.V. Quantum complexity theory. *STOC '93 Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing* (1993), 11–20.
3. Cooper, B., Lowe, B. and Sorbi, A., Eds. *New Computational Paradigms: Changing Conceptions of What Is Computable*. Springer, 2008.
4. Copeland, B. J. 1997. The Church-Turing Thesis. *The Stanford Encyclopedia of Philosophy*. E.N. Zalta, Ed.; http://plato.stanford.edu/entries/church-turing/.
5. Copeland, B.J. Narrow versus wide mechanism. *The Journal of Philosophy 97*, (2000), 5–32.
6. Copeland, B.J. Hypercomputation: Philosophical issues. *Theoretical Computer Science 317* (2004), 251–267.
7. Copeland, B.J. and Proudfoot, D. Alan Turing's forgotten ideas in computer science. *Scientific American 280*, (1999), 76–81.
8. Copeland, B.J. and Shagrir, O. Turing versus Gödel on computability and the mind. In B.J. Copeland, C. Posy, and O. Shagrir, Eds. *Computability: Gödel, Turing, Church and Beyond*, MIT Press, Cambridge, MA, 2013.
9. Copeland, B.J. and Sommaruga, G. The stored-program universal computer: Did Zuse anticipate Turing and von Neumann? In G. Sommaruga and T. Strahm, Eds., *Turing's Revolution*, Birkhauser, Basel, 2006.
10. Davis, M. The myth of hypercomputation. In C. Teuscher, Ed., *Alan Turing: Life and Legacy of a Great Thinker*. Springer Verlag, Berlin, 2004, 195–212.
11. Etesi, G. and Németi, I. Non-Turing computations via Malament-Hogarth space-times. *International Journal of Theoretical Physics 41*, (2002), 341–370.
12. Gandy, R. Church's Thesis and principles for mechanisms. In J. Barwise, D. Kaplan, H.J. Keisler, P. Suppes, and A.S. Troelstra, Eds., *The Kleene Symposium*, North-Holland, Amsterdam, 1980.
13. Goldreich, O. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.
14. Hogarth, M.L. Does general relativity allow an observer to view an eternity in a finite time? *Foundations of Physics Letters 5*, (1992), 173–181.
15. Milner, R. Elements of interaction: Turing Award lecture. *Commun. ACM 36*, 1 (Jan. 1993), 78–89.
16. Németi, I. and Andréka, H. Can general relativistic computers break the Turing barrier? In A. Beckmann, U. Berger, B. Löwe, and J.V. Tucker, Eds., *Logical Approaches to Computational Barriers*, Springer-Verlag, Berlin, 2006, 398–412.
17. Pitowsky, I. The physical Church thesis and physical computational complexity. *Iyyun 39* (1990), 81–99.
18. Pour-El, M. and Richards, I. The wave equation with computable initial data such that its unique solution is not computable. *Advances in Mathematics 39*, (1981), 215–239.
19. Shagrir, O. and Pitowsky, I. Physical hypercomputation and the Church-Turing Thesis. Special issue on hypercomputation. *Minds and Machines 13*, 1 (2003), 87–101.
20. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing 26*, 5 (1997), 1484–1509.
21. Sieg, W. Calculations by man and machine: Conceptual analysis. In W. Sieg, R. Sommer, and C. Talcott, Eds., *Reflections on the Foundations of Mathematics*, Association for Symbolic Logic, Natick, MA, 2002, 390-409.
22. Sieg, W. On computability. In A. Irvine, Ed., *Handbook of the Philosophy of Mathematics*, Elsevier, 2009.
23. Stannett, M. X-machines and the halting problem: Building a super-Turing machine. *Formal Aspects of Computing 2*, (1990), 331–341.
24. Wegner, P. and Goldin, D. Computation beyond Turing machines. *Commun. ACM 46*, 4 (Apr. 2003), 100–102.

Jack Copeland (jack.copeland@canterbury.ac.nz) is a professor of Philosophy at the University of Canterbury, New Zealand, where he is also the director of the Turing Archive for the History of Computing.

Eli Dresner (dresner@post.tau.ac.il) is a member of the Gershon H. Gordon Faculty of Social Sciences at Tel Aviv University.

Diane Proudfoot (diane.proudfoot@canterbury.ac.nz) is an assistant professor in the School of Humanities and Creative Arts at the University of Canterbury, New Zealand.

Oron Shagrir (shagrir@cc.huji.ac.il) is a professor of Philosophy and Cognitive Science at the Hebrew University of Jerusalem.