From Newton and Turing to cyber-physical systems: Exploring some fundamental problems in theoretical computer science

Joël Ouaknine*
Max Planck Institute for Software Systems, Saarland Informatics Campus, Germany

18 November 2017

> *Thinking in terms of* ***programs*** *rather than equations opens up a new kind of science.*
> Stephen Wolfram, 2002

> *The deepest, and most fundamental and consequential, open problem in Mathematics today is not about geometry or whole numbers: it is about* ***computation****.*
> Christos H. Papadimitriou, 2007

In 1937, a young Englishman by the name of Alan M. Turing published a paper with the obscure title *"On computable numbers, with an application to the Entscheidungsproblem"* in the Proceedings of the London Mathematical Society. In doing so, he arguably laid the mathematical foundations of modern computer science. Turing's seminal contribution was to show that the famous Entscheidungsproblem, formulated by the great German mathematician David Hilbert several years earlier, could not be solved: more precisely, Turing proved (in modern parlance) that the problem of determining whether a given computer program halts could not be done algorithmically—in other words that the famous *Halting Problem* is *undecidable*.

Although seemingly, at the time, a rather esoteric concern, the Halting Problem (and related questions) have dramatically gained in importance and relevance in more contemporary times. Fast forward to the 21st Century: nowadays, it is widely acknowledged that enabling engineers, programmers, and researchers to automatically verify and certify the correctness of the computer systems that they design is one of the Grand Challenges of computer science. In increasingly many instances, it is absolutely critical that the software governing various aspects of our daily lives (such as that running on an aircraft controller, for example) behave exactly as intended, lest catastrophic consequences ensue.

The modern field of *computer-aided formal verification* is concerned, broadly speaking, with ensuring that computer systems function as they were intended to. Initially focusing mainly on hardware, formal verification has in recent years undergone a paradigm shift: driven by the demands for *software* verification in increasingly varied and complex application domains, a central and pressing challenge has been to deal with system features that are *parametric*, *unbounded*, *quantitative*, *continuous*, or otherwise modelled as having an *infinite* number of states (unlike hardware, which is typically viewed as being finite-state).

One of the early successes of formal verification for software was Microsoft Research's Terminator tool and its successor T2, used to detect liveness bugs (such as non-termination of loops, leading to a hung computer, also known colloquially as the "Blue Screen of Death") in dozens of Windows device drivers. *In effect, Microsoft researchers sought to solve the Halting Problem, which Turing had shown decades earlier to be an impossible task!* In practice, by employing clever heuristics, deep mathematics, and state-of-the-art algorithms, the researchers managed to effectively handle a substantial proportion of the 'unsolvable' problem instances that they tackled, ultimately leading to a marked improvement in software quality (indeed, Windows doesn't crash or hang nearly as often as it used to!).

Returning to the fundamentals, however, raises the question of what classes of infinite-state programs can, at least in principle, be fully handled and analysed algorithmically. In the *Foundations of Algorithmic Verification* Group, we are attacking this challenge by viewing computer programs abstractly as *dynamical systems*, and we seek to design exact algorithms enabling one to fully analyse the behaviour of such systems. In particular, we are presently tackling a range of central algorithmic problems from verification, synthesis, performance, and control for linear dynamical systems, drawing among others on tools from number theory, Diophantine geometry, and algebraic geometry, with the

[*] Also affiliated with the Department of Computer Science, Oxford University, UK, and supported by ERC grant AVS-ISS (648701).

overarching goal of offering a systematic *exact computational treatment* of various important classes of dynamical systems and other fundamental models used in mathematics, computer science, and the quantitative sciences. Some of our achivements include several decidability and hardness results for linear recurrence sequences, which can be used to model simple loops in computer programs, answering a number of longstanding open questions in the mathematics and computer science literature; see, for instance, our survey [3]. It is worth noting, nevertheless, that even for the very basic class of so-called "simple linear loops", the question of whether the Halting Problem is decidable or not remains open to this day, a state of affairs described as "faintly outrageous" a few years ago by the mathematician Terence Tao [4].

In more recent work, we have attacked the so-called *Zero Problem* for linear differential equations, i.e., the question of determining algorithmically whether the unique solution to a given linear differential equation has a zero or not. Such equations, which go back as far as Newton, are ubiquitous in mathematics, physics, and engineering; they are also particularly useful to model *cyber-physical systems*, i.e., digital systems that evolve in and interact with a continuous environment. We were astounded to discover that the Zero Problem was not known to be either decidable (i.e., algorithmically solvable) or undecidable! In other words, it was—and still is!—an open problem as to whether one can algorithmically determine if a given linear differential equation has a zero or not (although of course in practice an answer can often be obtained using approximation techniques from numerical analysis).

Last year, we published two papers on this topic [1, 2] in which we obtained several important partial results: if one is interested in the existence of a zero over a *bounded* time interval, then it is possible to determine this algorithmically, provided that a certain hypothesis from the mathematical field of number theory, known as *Schanuel's Conjecture*, is true. (Schanuel's Conjecture is a far-reaching hypothesis which asserts among other things the irrationality and transcendence of numbers such as $(e + \pi)$—which are extremely deep open questions in mathematics.) We were also able to partially account for the fact that the Zero Problem has hitherto remained open in full generality: indeed, if one were able to solve it in dimension 9 (or higher), then in turn this would enable one to solve various longstanding hard open problems from a field of mathematics known as Diophantine approximation. In doing so, we therefore exhibited surprising and unexpected—at least to us!—connections between the modelling and analysis of cyber-physical systems and seemingly completely unrelated deep mathematical theories dealing with questions about whole numbers.

In summary, modern theoretical computer science brings fresh challenges and novel perspectives on old and deep problems in mathematics, and this synergy is leading to new and exciting scientific insights and advances; we are delighted to be able to contribute to this enterprise.

References

[1] Ventsislav Chonev, Joël Ouaknine, and James Worrell. On recurrent reachability for continuous linear dynamical systems. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 515–524. ACM, 2016.

[2] Ventsislav Chonev, Joël Ouaknine, and James Worrell. On the Skolem Problem for continuous linear dynamical systems. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 100:1–100:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[3] Joël Ouaknine and James Worrell. On linear recurrence sequences and loop termination. *SIGLOG News*, 2(2):4–13, 2015.

[4] Terence Tao. *Structure and randomness: pages from year one of a mathematical blog.* American Mathematical Society, 2008.