Are you a Mathematician or a Computer Scientist?

Celina de Figueiredo
Prof. Titular do PESC e Pesquisadora CNPq 1A

26/Junho 11h
Sala H-324b Prédio do CT/UFRJ
Evento Híbrido @pesc-coppe
O Problema do Milênio sobre Intratabilidade Computacional

Celina Miraglia Herrera de Figueiredo
Mathematician wins Turing award for harnessing randomness

Wigderson started exploring the relationship between randomness and computers in the 1980s, before the internet existed, attracted to ideas he worked on by intellectual curiosity, rather than how they might be used.

One of the unexpected ways in which his ideas are now widely used was on zero-knowledge proofs, which detail ways of verifying information without revealing the information itself.

read Quanta Magazine
watch Zero Knowledge Proof
Abel prize celebrates union of Mathematics and Computer Science

Two pioneers of the theory of computation have won one of the most prestigious honours in mathematics.

Since the advent of computers in the twentieth century, the emphasis in research has changed from ‘can an algorithm solve this problem?’ to ‘can an algorithm, at least in principle, solve this problem on an actual computer and in a reasonable time?’

read Abel interview 2021
Today is more difficult to distinguish pure and applied math

Maths → Comput
László Lovász (1948, Budapest) grew up a talented child competing at solving hard problems Early inspiration from Paul Erdos, prolific mathematician of the modern era, who focused on the mathematics of discrete objects Interested in basic research as well as in its applications, worked as a full-time researcher at Microsoft for seven years in between academic positions

Comput → Maths
Avi Wigderson (1956, Haifa) studied in Israel and the United States and held various academic positions before moving to the IAS in 1999, where he is ever since. Contributed to practically all areas of computer science, in which he attacked any problem with whatever mathematical tools he could find, even from distant fields of study
Laureates since 2003 in DM and TCS

2012 Endre Szemerédi – fundamental contributions to discrete math and theoretical computer science

2021 László Lovász and Avi Wigderson – foundational contributions to theoretical computer science and discrete math, and their role in shaping them into central fields of modern mathematics


The Fields Medal is awarded since 1936 up to four mathematicians under 40 years at the International Mathematical Union Congress, every four years
Laureates since 1966 in theoretical computer science

1974 Donald Knuth – contributions to the analysis of algorithms

1982 Stephen Cook – understanding the complexity of computation

1985 Richard M. Karp – contributions to the theory of algorithms, polynomial-time computability and NP-completeness

1986 Robert Tarjan – design and analysis of algorithms and data structures
The Millennium Prize Problems

David Hilbert:
23 problems
Paris in 1900

Clay Mathematics Institute:
7 prize problems
Paris in 2000

P versus NP problem has no associated mathematician

watch Vijaya Ramachandran
The question is whether or not, for all problems for which an algorithm can verify a given solution quickly (in polynomial time), an algorithm can also find that solution quickly.

Avi Wigderson expects P not equal NP

Donald Knuth expects P equal NP

*Clique graph gadget: a catwalk for variable $u_i$*

RS-family of $G_1$ must contain either the false triangles in (a) or the true triangles in (b). All bold triangles must belong to the RS-family.

*“The complexity of clique graph recognition”*  

watch Donald Knuth: P=NP
Hilbert’s two-part dream:
Everything that is true in Mathematics is provable
Everything that is provable can be automatically computed

1931 Godel proved that no matter how hard you try, your set of axioms will always be incomplete, they will not be sufficient to prove all true facts

1936 Turing introduced his Turing machine and proved the unsolvability of the halting problem

1940s–50s Turing and von Neumann played a major role in early development of computers

KURT GÖDEL’S LETTER TO JOHN VON NEUMANN - 1956
Princeton, 20 March 1956
Dear Mr. von Neumann:
With the greatest sorrow I have learned of your illness. The news came to me as quite unexpected. Morgenstern already last summer told me of a bout of weakness you once had, but at that time he thought that this was not of any greater significance. As I hear, in the last months you have undergone a radical treatment and I am happy that this treatment was successful as desired, and that you are now doing better. I hope and wish for you that your condition will soon improve even more and that the newest medical discoveries, if possible, will lead to a complete recovery.

Since you now, as I hear, are feeling stronger, I would like to allow myself to write you about a mathematical problem, of which your opinion would very much interest me: One can obviously easily construct a Turing machine, which for every formula \( F \) in first order predicate logic and every natural number \( n \), allows one to decide if there is a proof of \( F \) of length \( n \) (length = number of symbols). Let \( \Psi(F, n) \) be the number of steps the machine requires for this and let \( \psi(n) = \max F \Psi(F, n) \). The question is how fast \( \psi(n) \) grows for an optimal machine. One can show that \( \psi(n) \geq k \cdot n \). If there really were a machine with \( \psi(n) \sim k \cdot n \) (or even \( \sim k \cdot n^2 \)), this would have consequences of the greatest importance. Namely, it would obviously mean that in spite of the undecidability of the Entscheidungsproblem, the mental work of a mathematician concerning Yes-or-No questions could be completely replaced by a machine. After all, one would simply have to choose the natural number \( n \) so large that when the machine does not deliver a result, it makes no sense to think more about the problem. Now it seems to me, however, to be completely within the realm of possibility that \( \psi(n) \) grows that slowly. Since it seems that \( \psi(n) \geq k \cdot n \) is the only estimation which one can obtain by a generalization of the proof of the undecidability of the Entscheidungsproblem and after all \( \psi(n) \sim k \cdot n \) (or \( \sim k \cdot n^2 \)) only means that the number of steps as opposed to trial and error can be reduced from \( N \) to \( \log N \) (or \( (\log N)^2 \)). However, such strong reductions appear in other finite problems, for example in the computation of the quadratic residue symbol using repeated application of the law of reciprocity. It would be interesting to know, for instance, the situation concerning the determination of primality of a number and how strongly in general the number of steps in finite combinatorial problems can be reduced with respect to simple exhaustive search.

I do not know if you have heard that “Post’s problem”, whether there are degrees of unsolvability among problems of the form \( (\exists y) \phi(x, y) \), where \( \phi \) is recursive, has been solved in the positive sense by a very young man by the name of Richard Friedberg. The solution is very elegant. Unfortunately, Friedberg does not intend to study mathematics, but rather medicine (apparently under the influence of his father). By the way, what do you think of the attempts to build the foundations of analysis on ramified type theory, which have recently gained momentum? You are probably aware that Paul Lorenzen has pushed ahead with this approach to the theory of Lebesgue measure. However, I believe that in important parts of analysis non-eliminable impredicative proof methods do appear.

I would be very happy to hear something from you personally. Please let me know if there is something that I can do for you. With my best greetings and wishes, as well to your wife,

Sincerely yours,
Kurt Godel

P.S. I heartily congratulate you on the award that the American government has given to you.
1971 Stephen Cook – SAT NP-complete and polynomial-time reduction

1972 Richard Karp – Reducibility among combinatorial problems

equivalent classic unsolved problems

either each has polynomial algorithm or none does
Knuth’s terminology

Problem at least as difficult to solve in polynomial time as those of Cook–Karp class NP

Knuth wrote to 30 people: Herculean, Formidable or Arduous?

The winning write-in vote is NP-hard put forward by several people at Bell Labs

The Guide is 40 years old

“Despite that 23 years have passed since its publication, I consider Garey and Johnson the single most important book on my office bookshelf. Every computer scientist should have this book on their shelves as well. NP-completeness is the single most important concept to come out of theoretical computer science and no book covers it as well as Garey and Johnson.”

Advances in algorithms, machine learning, and hardware can help tackle many NP-hard problems once thought impossible.

Fifty Years of P vs. NP and the Possibility of the Impossible
Combinatorics is a branch of mathematics, plays crucial role in computer science, since digital computers manipulate discrete, finite objects

Combinatorial methods give analytical tools for computer algorithms worst-case and expected performance

Concrete Mathematics = CONtinuous and disCRETE mathematics

a complement to abstract mathematics
Theoretical Computer Science

Studies the power and limitations of computing

TCS two complementary sub-disciplines:

algorithm design develops efficient methods for computational problems

computational complexity shows limitations on efficiency of algorithms

discrete mathematics and TCS are allied fields: graphs, strings, permutations are central to TCS

Computing technology is made possible by algorithms, understanding the principles of powerful and efficient algorithms deepens our understanding of computer science, and also of the laws of nature
Graph Theory

Teoria Computacional de Grafos
Jayme Luiz Szwarcfiter, 2018

Graph theory is the mathematics of connectivity: covering, matching, packing, cuts, routing, independence.

Graphs and other combinatorial objects lead to algorithms for graph-theoretic problems, with application in computing.
Randomized Algorithms

Computers are deterministic: set of instructions of algorithm applied to input determines its computation and output.

The world we live in is full of random events that lack predictability, or a well-defined pattern.

Computer scientists allow algorithms to make random choices to improve their efficiency.

A randomized algorithm flips coins to compute a solution that is correct with high probability.
Las Vegas Quicksort:
correct answer
expected time

Monte Carlo Primality Test:
expected answer
deterministic time

Pseudoprime \((n)\)
1. If \(\text{MODULAR-EXponentiation}(2, n - 1, n) \neq 1 \pmod{n}\)
2. Return \text{COMPOSITE} // definitely
3. Else return \text{PRIME} // we hope!
Avi revolutionized our understanding of the role of randomness in computation.

every randomized polynomial time algorithm can be efficiently derandomized, made fully deterministic

trade-off between hardness versus randomness:

if there exists a hard enough problem, then randomness can be simulated by efficient deterministic algorithms; conversely, efficient deterministic algorithms even for specific problems with known randomized algorithms would imply that there must exist such a hard problem.
I am both a mathematician and a computer scientist

I study the mathematical foundations of computing

I prove theorems to understand computation, computational processes also in nature

Could a Nobel go to innovations of computing applied to a natural science?

My three decades in this field have been a continuous joyride, with fun problems, brilliant researchers, and many students, postdocs, and collaborators who have become close friends

I’m lucky to be part of a dynamic community
O Problema do Milênio sobre Intratabilidade Computacional

Celina Miraglia Herrera de Figueiredo