Introduction to Turing Machines and History of Computation

Meenakshi D'Souza

International Institute of Information Technology
Bangalore.

Term II 2022-23

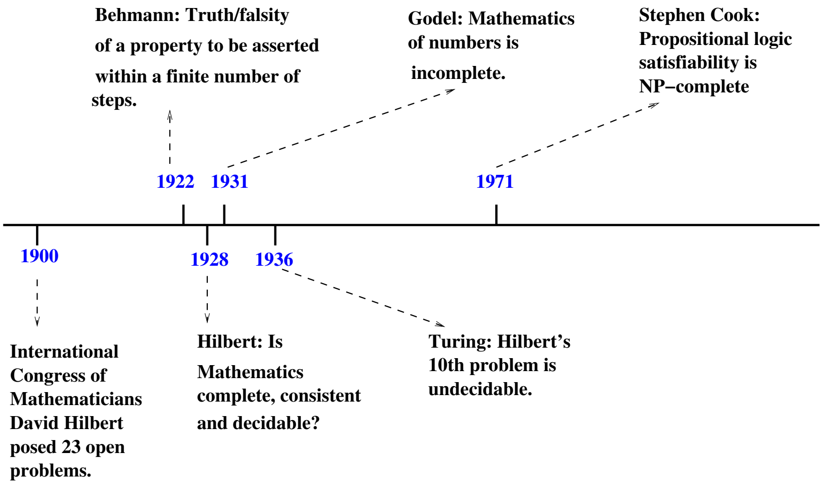
Outline

- Introduction
- Our popular mathematicians
- Turing machines and Computability
- Complexity Theory

Mathematics of Computer Science

- We all know that programs do a lot now.
- How to understand the Mathematics of programming?
  - Can every question/problem be answered/solved by programming?
  - Are there are difficult problems? What is difficulty?
  - Can we claim that one programming language can solve more problems than another?
- What is the History of the Mathematics of Computer Science?

Timeline: Theory of Computation



David Hilbert

- David Hilbert, a German Mathematician.
- Seminal work: Logic, functional analysis, axiomatization of geometry.
- Believed that there is no ignorabimus in Mathematics:
  - Every mathematical problem should have a solution.
  - The point is to know one way or the other: The solution could be a proof that the original problem is impossible.

Hilbert's problems

- Hilbert posed 23 open problems for the century in the 1900 International Congress of Mathematicians.
- Two problems relevant for us:
  - Problem 2: Finististic proof of the consistency of the axioms of arithmetic.
  - Problem 10: To devise a process according to which it can be determined in a finite number of operations whether the equation is solvable in rational integers.

Alonzo Church

- Alonzo Church, an American Mathematician.
- Seminal work: Mathematical logic, λ-Calculus, Church-Turing thesis.
- Had very influential set of students.

Students of Hilbert and Church

- David Hilbert: Kurt Gödel
- Alonzo Church: Alan Turing, Stephen Kleene, Raymond Smullyan, Dana Scott, Michael Rabin...

Kurt Gödel

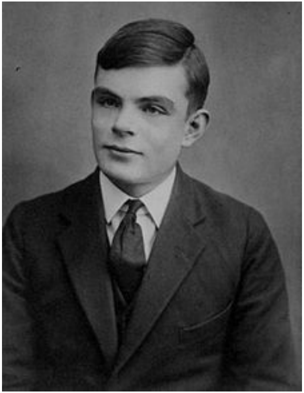
- Kurt Gödel, an Austrian and later, American Mathematician and philosopher.
- Seminal work: Mathematical logic, Incompleteness theorem, set theory.
- Proved that Mathematics of numbers is incomplete: Answered Hilbert's second problem in the negative.

Gödel's work: Incompleteness theorem

- In 1931, Gödel essentially constructed a formula that claims that it is unprovable in a given formal system.
- If it were provable, it would be false, which contradicts the idea that in a consistent system, provable statements are always true.
- Thus there will always be at least one true but unprovable statement.

Alan Turing

- Alan Turing, an English Computer Scientist, logician, cryptanalyst.
- Widely considered as the father of theoretical computer science and artificial intelligence.
- Most popular work: Turing machines.
- Highest award in Computer Science: Turing award.



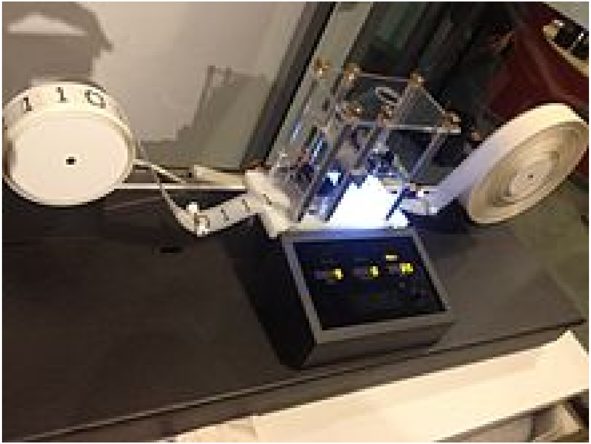
Hilbert's 10th problem

- 1900: Hilbert's 10th problem: Given a Diophantine equation with any number of unknown quantities and with rational integral coefficients, to devise a process according to which it can be determined in a finite number of operations whether the equation is solvable in rational integers.
- 1922: H. Behmann: A quite definite generally applicable prescription is required which will allow one to decide in a finite number of steps the truth or falsity of a given purely logical assertion ...
- 1928, International Congress again, David Hilbert: First, was Mathematics complete ... Second, was Mathematics consistent ... And thirdly, was Mathematics decidable?"

Turing's undecidability result

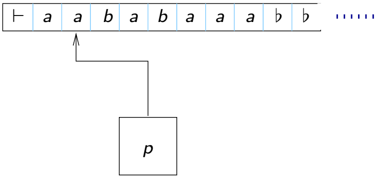
- Turing showed that there is no process (algorithm) to solve Hilbert's 10th problem.
- Devised Turing machines, an abstract machine that has operations to mimic modern day computer operations.
- Gave a mathematical proof based on diagnolization technique that there is no algorithm for Hilbert's tenth problem.

Turing machine



A Turing machine is a mathematical model of computation with states and transitions that describe how states change on reading letters from an input tape.

How a Turing machine works



- Finite control
- Tape infinite to the right
- Each step: In current state p, read current symbol under the tape head, say a: Change state to q, replace current symbol by b, and move head left or right.

$$(p, a) \to (q, b, L/R).$$

How a Turing machine works

- Special designated accept state t and reject state r. These states are assumed to be "sink" states.
- TM accepts its input by entering state t.
- TM rejects its input by entering state r.
- TM never falls off the left end of the tape (i.e it always moves right on seeing '$\vdash$').

Church-Turing thesis

- Church-Turing thesis: Turing machines represent an abstract model of all that is programmable in today's computers.
- All programming languages are equivalent, they all implement the same set of algorithms to solve problems.
- They only differ in their notations, data structures and other details.

Undecidable/Unsolvable problems

- Turing showed first that there is an unsolvable problem.
- There are now several other unsolvable problems:
  - Halting problem.
  - Membership problem.
  - Puzzles with dominoes, cards etc.
  - … many more such problems.
- Such problems can never be solved in their full generality using any modern day language, even on a super computer.

Some difficult problems

- Some problems can be solved, but, algorithms to solve them are not efficient.
- They are so inefficient that it is practically impossible to implement and use them on large problems.
- Mathematical characterization: NP-complete problems.
- 1971: Stephen Cook showed that propositional logic satisfiability is NP-complete.
- Many useful problems are difficult: Travelling salesperson problem, graph colouring problem, knapsack problem, prime factorization problem etc.
- We use intractable problems in a positive way too...

Conclusion

- Programs, even in the most sophisticated programming language and in the best super computer, cannot solve every problem computationally.
- Some problems are unsolvable.
- Some problems are solvable but intractable.