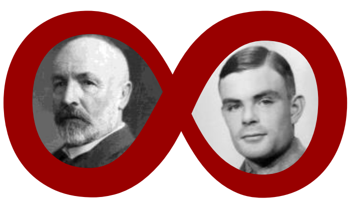
Great Theoretical Ideas in CS

Lecture 8:
Turing's Legacy: Undecidability

Anil Ada
Ariel Procaccia (this time)

Our Protagonists

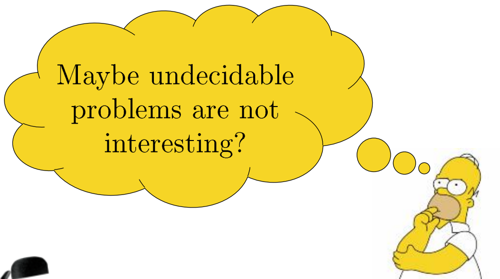


Georg Cantor
1845-1918
Father of set theory

Alan Turing
1912-1954
Father of CS

Decidable or Undecidable?

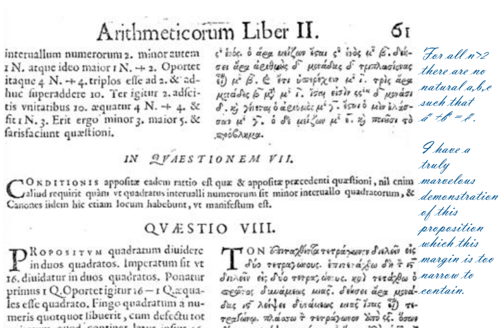
- Poll 1: Let Σ be a finite alphabet. Which of the following sets is countable?
  1. The set of decidable languages over Σ
  2. The set of all languages over Σ
  3. Both
  4. Neither



The Halting Problem

- Input: Program pseudocode, input to the program
- Output: True if the given program halts on the given input, false otherwise

Why is it interesting?


Arithmeticorum Liber II. 61

interuallum numerorum 2. minor autem 1 N. atque ideo maior 1 N. + 2. Oportet itaque 4 N. + 4. triplos esse ad 2. & adhuc superaddere 10. Ter igitur 2. adscitis vnitatibus 10. æquatur 4 N. + 4. & fit 1 N. 3. Erit ergo minor 3. maior 5. & satisfaciunt quæstioni.

IN QVAESTIONEM VII.

CONDITIONIS appositæ eadem ratio est quæ & appositæ præcedenti quæstioni, nil enim aliud requirit quàm vt quadratus interualli numerorum sit minor interuallo quadratorum, & Canones iidem hic etiam locum habebunt, vt manifestum est.

QVÆSTIO VIII.

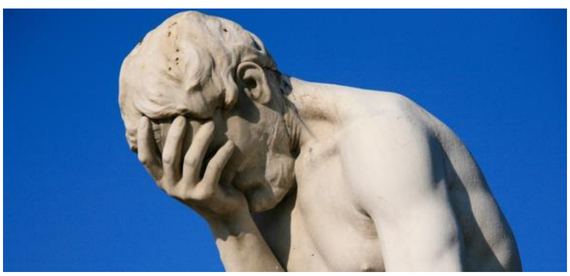
PROPOSITVM quadratum diuidere in duos quadratos. Imperatum sit vt 16. diuidatur in duos quadratos. Ponatur primus 1 Q. Oportet igitur 16 − 1 Q. æquales esse quadrato. Fingo quadratum a numeris quotquot libuerit, cum defectu tot vnitatum quod continet latus ipsius 16.

For all n>2 there are no natural a,b,c such that $a^n + b^n = c^n$.

I have a truly marvelous demonstration of this proposition which this margin is too narrow to contain.

```
FERMAT()
t ← 3
while true
   for all n ∈ {3,…,t} and x,y,z ∈ {1,…,t}
      if x^n + y^n = z^n then return (x,y,z,n)
   end for
   t ← t + 1
end while
```

Question: Does this program halt?



Theorem:

The Halting Problem is undecidable!

Proof (by Pseudocode)

- Suppose that there exists a procedure `HALT`(*program*, *input*)
- Consider the program:

```
Turing(program)
if HALT(program, program) then
   loop forever
else
   return true
```

- What is the output of `Halt(Turing,Turing)`?
  - If `Halt(Turing,Turing)` then `Turing(Turing)` doesn't halt
  - If not `Halt(Turing,Turing)` then `Turing(Turing)` halts

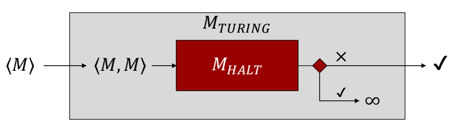
Proof (More Formal)

- HALT = $\{\langle M, x\rangle : M$ is a TM that halts on $x\}$
- Suppose the TM M_{HALT} decides HALT
- Consider the following TM M_{TURING}

Treat the input as $\langle M\rangle$ for a TM M
Run M_{HALT} with input $\langle M, M\rangle$
If it accepts, go into an infinite loop
If it rejects, accept (i.e., halt)

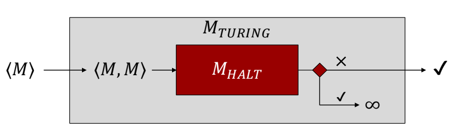
Proof (More Formal)

- HALT = $\{\langle M, x\rangle : M$ is a TM that halts on $x\}$
- Suppose the TM M_{HALT} decides HALT
- Consider the following TM M_{TURING}

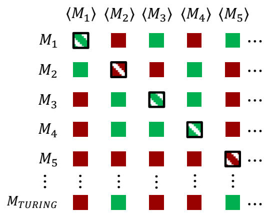


Proof (More Formal)

What happens when $\langle M_{TURING}\rangle$ is given as input to M_{TURING}?



Diagonalization Redux

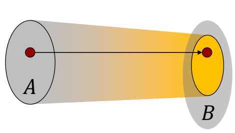


This is nothing but a diagonalization argument!

Is there a way to show other languages are undecidable?

Reductions

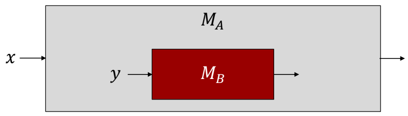
- We want to define $A \leq B$ to mean that B is at least as hard as A



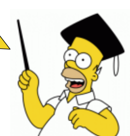
- That is:
  - B decidable $\Longrightarrow$ A decidable
  - A undecidable $\Longrightarrow$ B undecidable

Reductions

- Terminology: Let A and B be two languages, we say that A reduces to B, and write $A \leq B$, if it is possible to decide A using a TM that decides B as a subroutine



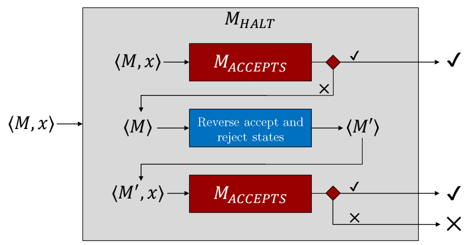
To show that problem B is undecidable, we just need to show that HALT $\leq B$



Example: Accepts

- ACCEPTS $= \{\langle M, x\rangle : M$ is a TM that accepts $x\}$
- This means:
  - $\langle M, x\rangle \in$ ACCEPTS $\Longrightarrow x$ leads to an accept state in M
  - $\langle M, x\rangle \notin$ ACCEPTS $\Longrightarrow x$ leads to a reject state or M does not halt
- Theorem: ACCEPTS is undecidable

Proof (by Illustration)

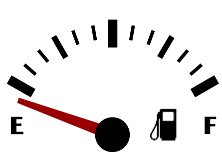


Proof (More Formal)

- We will show that HALT $\leq$ ACCEPTS
- Let $M_{ACCEPTS}$ be a TM that decides ACCEPTS
- Here is a TM that decides HALT:
  - On input $\langle M, x\rangle$ run $M_{ACCEPTS}(\langle M, x\rangle)$
  - If it accepts, accept
  - Reverse the accept and reverse states of M, call it M'
  - Run $M_{ACCEPTS}(\langle M', x\rangle)$
  - If it accepts, accept, and reject otherwise
- Argue that:
  - If $\langle M, x\rangle \in$ HALT then the machine accepts it
  - If $\langle M, x\rangle \notin$ HALT then the machine rejects it ■

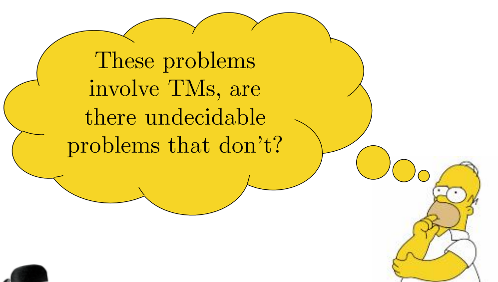
Example: EMPTY

- EMPTY $= \{\langle M\rangle : M$ is a TM that accepts nothing$\}$
- Theorem: EMPTY is undecidable



Proof

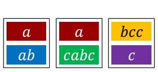
- We will show that ACCEPTS $\leq$ EMPTY
- Given $\langle M, x \rangle$, construct a TM M_x that, given y, runs $M(x)$ and returns its output
- The machine $M_{ACCEPTS}$ constructs M_x, runs $M_{EMPTY}(\langle M_x \rangle)$, and flips its output
- Two cases:
  - M accepts $x \Longrightarrow L(M_x) = \Sigma^* \Longrightarrow M_{EMPTY}$ rejects $\langle M_x \rangle$
  - M rejects x or doesn't halt on $x \Longrightarrow L(M_x) = \emptyset$ $\Longrightarrow M_{EMPTY}$ accepts $\langle M_x \rangle$ ■



Post's Correspondence Problem

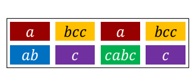
Input

A finite collection of "dominoes" with strings written on each half



Output

Accept if copies of the dominoes can be arranged so that the strings match

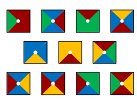


Undecidable! Proved in 1946 by Post

Wang Tiles

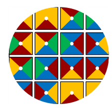
Input

A finite collection of "Wang tiles" (squares) with colored edges



Output

Accept if the infinite plane can be tiled using tiles with matching sides



Undecidable! Proved in 1966 by Berger

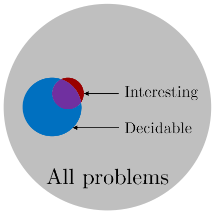
Big Undecidable Problems

- Entscheidungsproblem:
  - Pronunciation: https://youtu.be/RG2uPLG5K48
  - Can a first-order-logic formula be derived from given axioms?
  - Example: $\neg\exists x, y, z, n \in \mathbb{N}: (n \geq 3) \wedge (x^n + y^n = z^n)$
  - Formulated by Hilbert in 1928, proved undecidable by Turing in 1936 (and, independently, by Church)
- Hilbert's 10th Problem (Diophantine equations):
  - Does a given multivariate polynomial with integer coefficients have an integer root?
  - Example: $3x^2 - 2xy - y^2z - 7 = 0$ $(x = 1, y = 2, z = -2)$
  - One of 23 open problems on Hilbert's famous 1900 list
  - Proved undecidable by Matiyasevich in 1970

Decidable or Undecidable?

- Poll 2: Which of the following problems is decidable?
  1. $\mathsf{EQ} = \{\langle M, M' \rangle : M, M' \text{ TMs}, L(M) = L(M')\}$
  2. $\mathsf{GRAVITON} = \emptyset$ if gravitons exist, $\{1\}$ otherwise
  3. Both
  4. Neither

Interesting vs. Decidable



So what next?

Summary

- Terminology and concepts:
  - HALT, ACCEPTS, EMPTY
  - Reductions between computational problems
- Theorems:
  - Most problems are undecidable
  - HALT, ACCEPTS, EMPTY are undecidable
- Big ideas:
  - Exploring the limits of computation via reductions