15-251
Great Theoretical Ideas in Computer Science

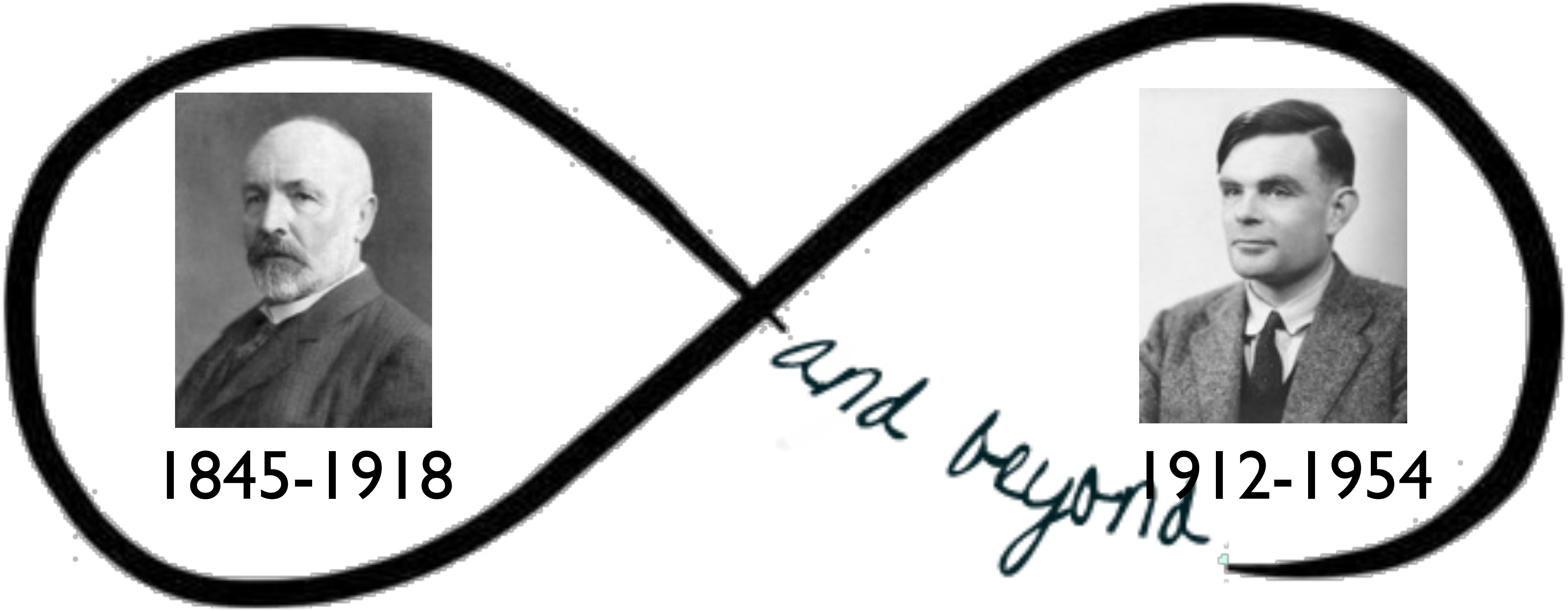
Uncountability and Uncomputability

January 29th, 2015

Our heros for this lecture

father of set theory

father of computer science



Uncountability

Uncomputability

Our heros for this lecture

father of set theory

father of computer science

Example 3: Set theory

Question:
How 'complete' are those 9 axioms?
(ZFC)

Answer based on 100 years of experience:
Amazingly complete!
Almost all true statements about **math** (GORM) can be deduced from them.

In particular, everything we will prove in 15-251!

Uncountability

Uncomputability

Infinity in mathematics

Pre-Cantor:

"Infinity is nothing more than a figure of speech which helps us talk about limits.
The notion of a completed infinity doesn't belong in mathematics"

- Carl Friedrich Gauss

Post-Cantor:

Infinite sets are mathematical objects just like finite sets.

Some of Cantor's contributions

> The study of infinite sets

> Explicit definition and use of 1-to-1 correspondence

- This is the right way to compare the cardinality of sets

> There are different levels of infinity.

- There are infinitely many infinities.

> $|\mathbb{N}| < |\mathbb{R}|$ even though they are both infinite.

> $|\mathbb{N}| = |\mathbb{Z}|$ even though $\mathbb{N} \subsetneq \mathbb{Z}$.

> The diagonal argument.

Reaction to Cantor's ideas

Most of the ideas of Cantorian set theory should be banished from mathematics once and for all!

- Henri Poincaré

Reaction to Cantor's ideas

I don't know what predominates
in Cantor's theory -
philosophy or theology.

- Leopold Kronecker

Reaction to Cantor's ideas

Scientific charlatan.

- Leopold Kronecker

Reaction to Cantor's ideas

Corrupter of youth.

- Leopold Kronecker

Reaction to Cantor's ideas

Wrong.

- Ludwig Wittgenstein

Reaction to Cantor's ideas

Utter non-sense.

- Ludwig Wittgenstein

Reaction to Cantor's ideas

Laughable.

- Ludwig Wittgenstein

Reaction to Cantor's ideas

No one should expel us from the Paradise that Cantor has created.

- David Hilbert

Reaction to Cantor's ideas

If one person can see it as a paradise,
why should not another see it as a joke?

- Ludwig Wittgenstein

How do we count a finite set?

$A = \{\text{apple, orange, banana, melon}\}$

What does $|A| = 4$ mean?

There is a 1-to-1 correspondence between

A and $\{1, 2, 3, 4\}$

$$\text{apple} \longleftrightarrow 1$$
$$\text{orange} \longleftrightarrow 2$$
$$\text{banana} \longleftrightarrow 3$$
$$\text{melon} \longleftrightarrow 4$$

How do we count a finite set?

$A = \{\text{apple, orange, banana, melon}\}$

$B = \{200, 300, 400, 500\}$

What does $|A| = |B|$ mean?

$$\begin{array}{lcccc} \text{apple} & \longleftrightarrow & 1 & \longleftrightarrow & 500 \\ \text{orange} & \longleftrightarrow & 2 & \longleftrightarrow & 200 \\ \text{banana} & \longleftrightarrow & 3 & \longleftrightarrow & 300 \\ \text{melon} & \longleftrightarrow & 4 & \longleftrightarrow & 400 \end{array}$$

How do we count a finite set?

$$A = \{\text{apple, orange, banana, melon}\}$$

$$B = \{200, 300, 400, 500\}$$

What does $|A| = |B|$ mean?

$$\text{apple} \longleftrightarrow 500$$

$$\text{orange} \longleftrightarrow 200$$

$$\text{banana} \longleftrightarrow 300$$

$$\text{melon} \longleftrightarrow 400$$

$|A| = |B|$ iff there is a 1-to-1 correspondence between A and B.

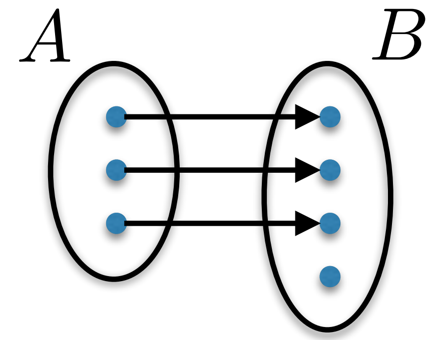
3 important types of functions

injective, 1-to-1

$f : A \to B$ is injective if

$a \neq a' \implies f(a) \neq f(a')$

$A \hookrightarrow B$

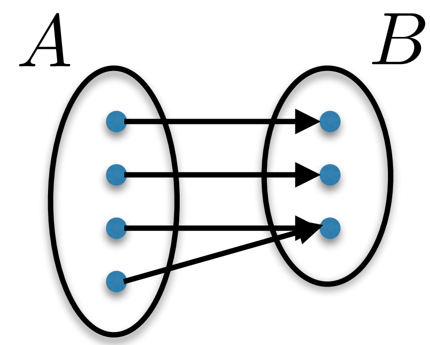


surjective, onto

$f : A \to B$ is surjective if

$\forall b \in B, \exists a \in A \text{ s.t. } f(a) = b$

$A \twoheadrightarrow B$

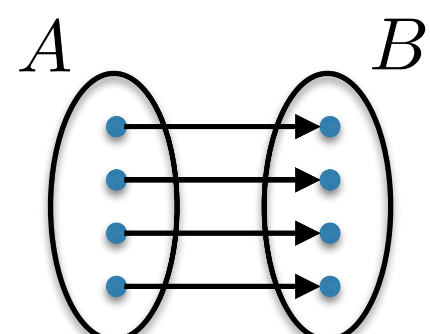


bijective, 1-to-1 correspondence

$f : A \to B$ is bijective if

f is injective and surjective

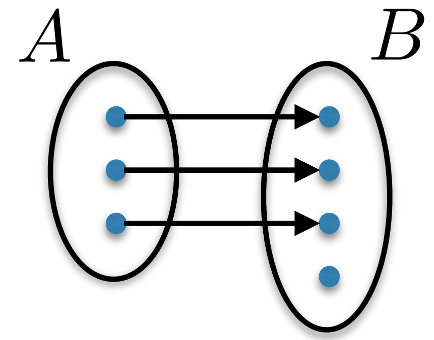
$A \leftrightarrow B$



Comparing the cardinality of finite sets

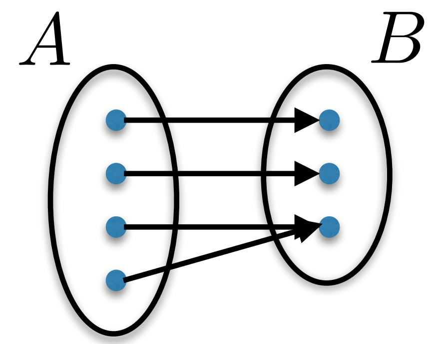
$|A| \leq |B|$

$A \hookrightarrow B$



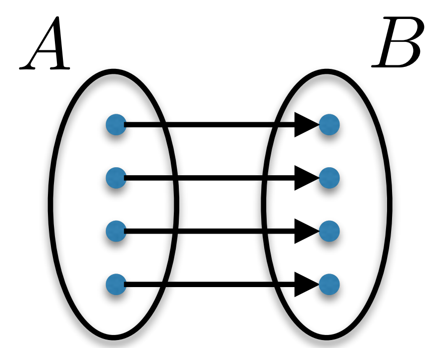
$|A| \geq |B|$

$A \twoheadrightarrow B$



$|A| = |B|$

$A \leftrightarrow B$



Sanity checks

$$|A| \leq |B| \text{ iff } |B| \geq |A|$$

$$A \hookrightarrow B \text{ iff } B \twoheadrightarrow A$$

$$|A| = |B| \text{ iff } |A| \leq |B| \text{ and } |A| \geq |B|$$

$$A \leftrightarrow B \text{ iff } A \hookrightarrow B \text{ and } A \twoheadrightarrow B$$

$$A \leftrightarrow B \text{ iff } A \hookrightarrow B \text{ and } B \hookrightarrow A$$

$$\text{If } |A| \leq |B| \text{ and } |B| \leq |C| \text{ then } |A| \leq |C|$$

$$\text{If } A \hookrightarrow B \text{ and } B \hookrightarrow C \text{ then } A \hookrightarrow C$$

One more definition

$$|A| < |B|$$

not $|A| \geq |B|$

There is no surjection from A to B.

There is no injection from B to A.

There is an injection from A to B,
but there is no bijection between A and B.

These are the **right** definitions for infinite sets as well!

All is OK with infinite sets

$$|A| \leq |B| \text{ iff } |A| \leq |B|$$

$$A \hookrightarrow B \text{ iff } B \twoheadrightarrow A$$

$$|A| = |B| \text{ iff } |A| \leq |B| \text{ and } |B| \leq |A|$$

$$A \leftrightarrow B \text{ iff } A \hookrightarrow B \text{ and } A \twoheadrightarrow B$$

$$A \leftrightarrow B \text{ iff } A \hookrightarrow B \text{ and } B \hookrightarrow A$$

Cantor
Schröder
Bernstein

$$\text{If } |A| \leq |B| \text{ and } |B| \leq |C| \text{ then } |A| \leq |C|$$

$$\text{If } A \hookrightarrow B \text{ and } B \hookrightarrow C \text{ then } A \hookrightarrow C$$

Let me show you some interesting consequences.

Examples of equal size sets

$$|\mathbb{N}| = |\mathbb{Z}|$$

$$\mathbb{N} = \{0, 1, 2, 3, 4, \ldots\}$$

$$\mathbb{Z} = \{\ldots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \ldots\}$$

$$\begin{array}{ccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \ldots \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \\ 0, & 1, & -1, & 2, & -2, & 3, & -3, & 4, & -4, & \ldots \end{array}$$

$$f(n) = (-1)^{n+1} \left\lceil \frac{n}{2} \right\rceil$$

List the integers so that *eventually* every number is reached.

Examples of equal size sets

$$|\mathbb{N}| = |\mathbb{Z}|$$

Does this make any sense? $\mathbb{N} \subsetneq \mathbb{Z}$

$A \subsetneq B \implies |A| < |B|?$ Surely $|\mathbb{N}| < |\mathbb{Z}|$.

Does renaming the elements of a set change its size?

Let's rename the elements of $\mathbb{Z}$:

$$\{\ldots, \text{banana}, \text{ apple}, \text{ melon}, \text{ orange}, \text{ mango}, \ldots\}$$

Let's call this set F. How can you justify $|\mathbb{N}| < |F|$?

Bijection is nothing more than renaming.

Examples of equal size sets

$$|\mathbb{N}| = |S|$$

$$\mathbb{N} = \{0, 1, 2, 3, 4, \ldots\}$$

$$S = \{0, 1, 4, 9, 16, \ldots\}$$

$$f(n) = n^2$$

Examples of equal size sets

$$|\mathbb{N}| = |P|$$

$$\mathbb{N} = \{0, 1, 2, 3, 4, \ldots\}$$

$$P = \{2, 3, 5, 7, 11, \ldots\}$$

$$f(n) = n\text{'th prime number.}$$

Countable sets

$$|\mathbb{N}| = |A|$$

if:

A is infinite,

and you can list the elements as $a_0, a_1, a_2, \ldots$

$(a_i \neq a_j \text{ for } i \neq j)$

in a well-defined way.

Definition:

A is **countably infinite** if $|\mathbb{N}| = |A|$.

A is **countable** if A is finite or $|\mathbb{N}| = |A|$.

Countable sets

Definition:

A is **countably infinite** if $|\mathbb{N}| = |A|$.

A is **countable** if A is finite or $|\mathbb{N}| = |A|$.

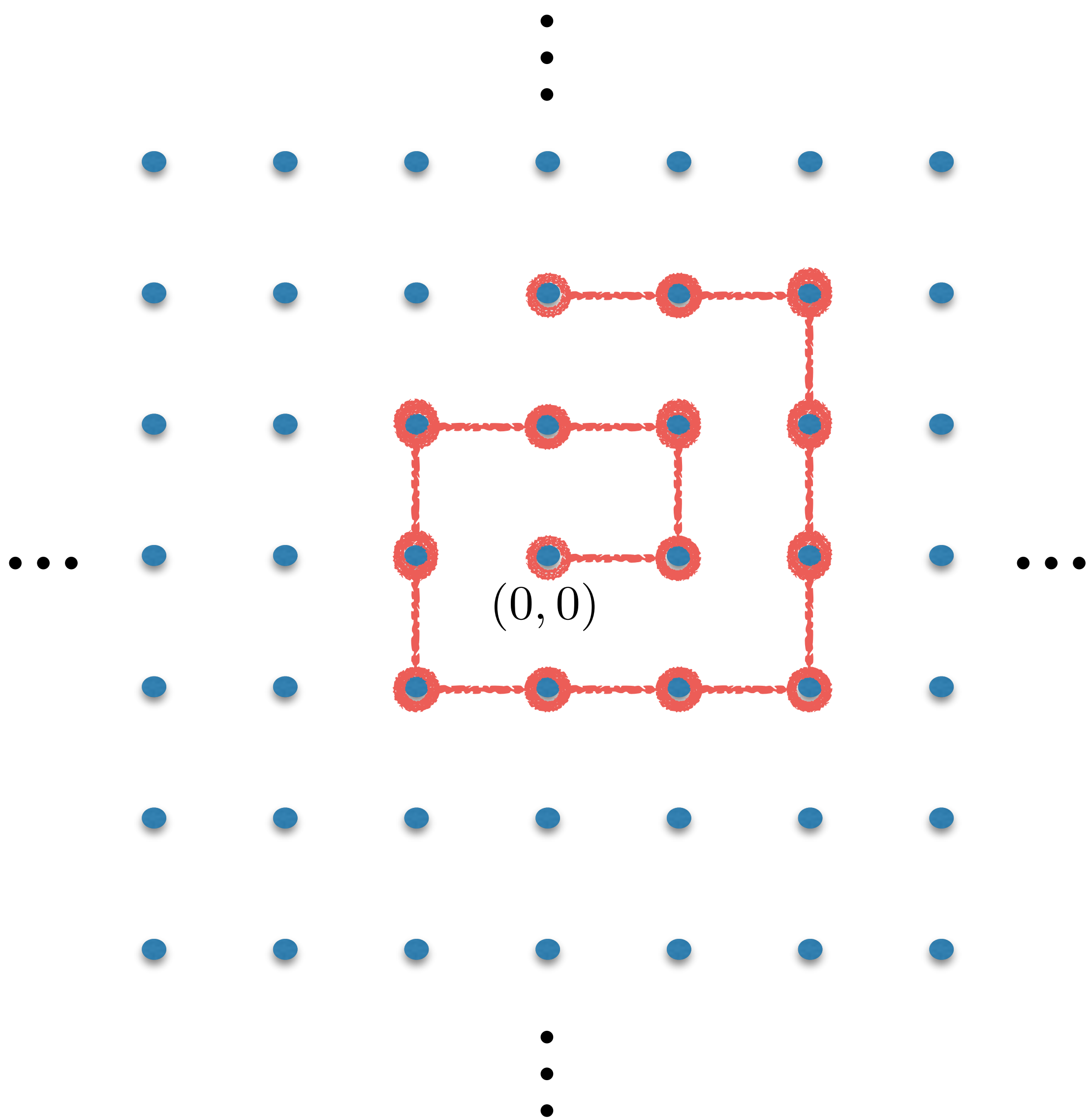
What if A is infinite, but $|A| < |\mathbb{N}|$?

No such set exists!

So really A is countable if $|A| \leq |\mathbb{N}|$.

Countable?

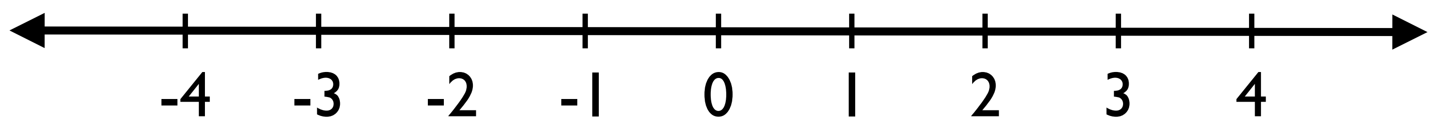
$$|\mathbb{N}| = |\mathbb{Z} \times \mathbb{Z}|?$$



$(0,0)$
$(1,0)$
$(1,1)$
$(0,1)$
$(-1,1)$
$(-1,0)$
$(-1,-1)$
$(0,-1)$
$(1,-1)$
$(2,-1)$
$(2,0)$
$(2,1)$
$(2,2)$
$(1,2)$
$(0,2)$

Countable?

$$|\mathbb{N}| = |\mathbb{Q}|?$$



Between **any** two rational numbers, there is another one.

Can't just list them in the order they appear on the line.

Any rational number can be written as a fraction $\dfrac{a}{b}$.

$\mathbb{Z} \times \mathbb{Z} \twoheadrightarrow \mathbb{Q}$ (map (a, b) to $\dfrac{a}{b}$)

$$\implies |\mathbb{Q}| \leq |\mathbb{Z} \times \mathbb{Z}| = |\mathbb{N}|$$

Clearly $|\mathbb{N}| \leq |\mathbb{Q}|$. So $|\mathbb{N}| = |\mathbb{Q}|$.

Countable?

$$|\mathbb{N}| = |\{0,1\}^*|?$$

$\{0,1\}^*$ = the set of finite length binary strings.

ε

0

1

$00, 01, 10, 11$

$000, 001, 010, 011, 100, 101, 110, 111$

$\ldots$

Countable?

$$|\mathbb{N}| = |\Sigma^*|?$$

Σ^* = the set of finite length words over Σ.

Same idea.

CS method to show a set A is countable $(|A| \leq |\mathbb{N}|)$:

Show $|A| \leq |\Sigma^*|$

i.e. $\Sigma^* \twoheadrightarrow A$

CS method in action

Is $\mathbb{Q}[x]$ countable?

$\mathbb{Q}[x]$ = polynomials with rational coefficients.

Take $\Sigma = \{0, 1, \ldots, 9, x, +, -, *, /, \hat{}\}$

Every polynomial can be described by a finite string over Σ.

e.g. $x\hat{}3 - 1/4x\hat{}2 + 6x - 22/7$

So $\Sigma^* \twoheadrightarrow \mathbb{Q}[x]$

Seems like every set is countable…

Nope!

That would be boring!

Cantor's Theorem

Theorem: For any non-empty set A,

$$|A| < |\mathcal{P}(A)|.$$

$$S = \{1, 2, 3\}$$

$$\mathcal{P}(S) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}\}$$

$$|\mathcal{P}(S)| = 2^{|S|}$$

$$\mathcal{P}(S) \leftrightarrow \{0,1\}^{|S|}$$

↓

binary strings of length |S|

$$S = \{1, 2, 3\}$$

$$1\ 0\ 1 \longleftrightarrow \{1, 3\}$$

$$0\ 0\ 0 \longleftrightarrow \emptyset$$

Cantor's Theorem

Theorem: For any non-empty set A,

$$|A| < |\mathcal{P}(A)|.$$

So:

$$|\mathbb{N}| < |\mathcal{P}(\mathbb{N})|.$$

$$|\mathbb{N}| < |\mathcal{P}(\mathbb{N})| < |\mathcal{P}(\mathcal{P}(\mathbb{N}))| < |\mathcal{P}(\mathcal{P}(\mathcal{P}(\mathbb{N})))| < \cdots$$

(an infinity of infinities)

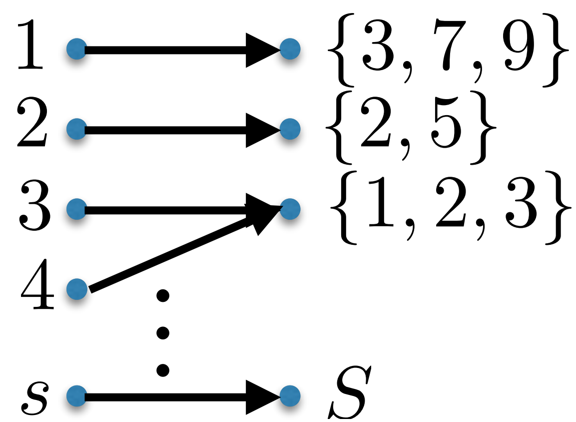
Proof by diagonalization

Assume $|\mathcal{P}(A)| \leq |A|$ for some set A.

So $A \twoheadrightarrow \mathcal{P}(A)$. Let f be such a surjection.

Example

$\mathbb{N} \twoheadrightarrow \mathcal{P}(\mathbb{N})$



$S = \{1, 4, \ldots\}$

Define $S = \{a \in A : a \notin f(a)\} \in \mathcal{P}(A)$.

Since f is onto, $\exists s \in A$ s.t. $f(s) = S$.

But this leads to a contradiction:

if $s \notin S$ then $s \in S$

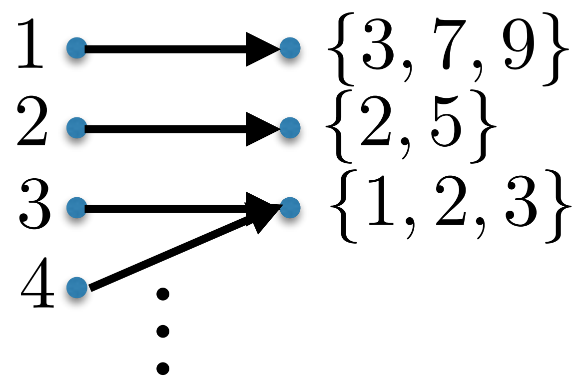
if $s \in S$ then $s \notin S$ $\square$

Why is this called a diagonalization argument?

Proof by diagonalization

Example

$\mathbb{N} \twoheadrightarrow \mathcal{P}(\mathbb{N})$



$S = \{1, 4, \ldots\}$

	1	2	3	4	5	⋯
f(1)	0	0	1	0	0	
f(2)	0	1	0	0	1	
f(3)	1	1	1	0	0	⋯
f(4)	1	1	1	0	0	
f(5)	0	0	0	1	1	
⋮			⋮			
f(s)= S	1	0	0	1	0	⋯

S is defined so that
S cannot equal any f(a)

Uncountable sets

So $|\mathcal{P}(\mathbb{N})| > |\mathbb{N}|$.

Definition:

A set is A **uncountable** if it is not countable,

i.e. $|A| > |\mathbb{N}|$.

Some examples: $\mathcal{P}(\mathbb{N}), \mathcal{P}(\mathcal{P}(\mathbb{N})), \ldots$

Uncountable sets

Let $\{0,1\}^\infty$ be the set of binary strings of infinite length.

{0,1,2,3,4,5,6,7,8,9, … }

0000000000 … $\longleftrightarrow \emptyset$

1111111111 … $\longleftrightarrow \mathbb{N}$

1010101010 … $\longleftrightarrow \{\text{even natural numbers}\}$

⋮

$\{0,1\}^\infty$ is uncountable, i.e. $|\{0,1\}^\infty| > |\mathbb{N}|$

because $\{0,1\}^\infty \leftrightarrow \mathcal{P}(\mathbb{N})$. (just like $\{0,1\}^{|S|} \leftrightarrow \mathcal{P}(S)$)

(Recall $\{0,1\}^*$ *is* countable.)

Uncountable sets

Let $\{0,1\}^\infty$ be the set of binary strings of infinite length.

$\{0,1\}^\infty$ is uncountable, i.e. $|\{0,1\}^\infty| > |\mathbb{N}|$

<u>Direct diagonal proof:</u> Suppose $|\{0,1\}^\infty| \leq |\mathbb{N}|$

$$\mathbb{N} \twoheadrightarrow \{0,1\}^\infty$$

1	**0**	0	1	0	0	$\cdots$
2	0	**1**	0	0	1	$\cdots$
3	1	1	**1**	0	0	$\cdots$
4	1	1	1	**0**	0	$\cdots$
5	0	0	0	1	**1**	$\cdots$
$\vdots$			$\vdots$			

1 0 0 1 0 $\cdots$ —> cannot appear in the list

Uncountable sets

$\mathbb{R}$ is uncountable. In fact $(0,1)$ is uncountable.

exercise

Appreciating the diagonalization argument

If you want to appreciate something,
try to break it…

Exercise:

Why doesn't the diagonalization argument work for

$\mathbb{N}$, $\{0,1\}^*$, a countable subset of $\{0,1\}^\infty$?

Before we end this section:

Is there a set S such that

$$|\mathbb{N}| < |S| < |\mathcal{P}(\mathbb{N})|?$$

Continuum Hypothesis:
No such set exists.

(Hilbert's 1st problem)

Applications to Computer Science

Most problems are uncomputable

Just count!

For any TM M, $\langle M \rangle \in \Sigma^*$

So $\{M : M \text{ is a TM}\}$ is countable.

How about the set of all computational problems?

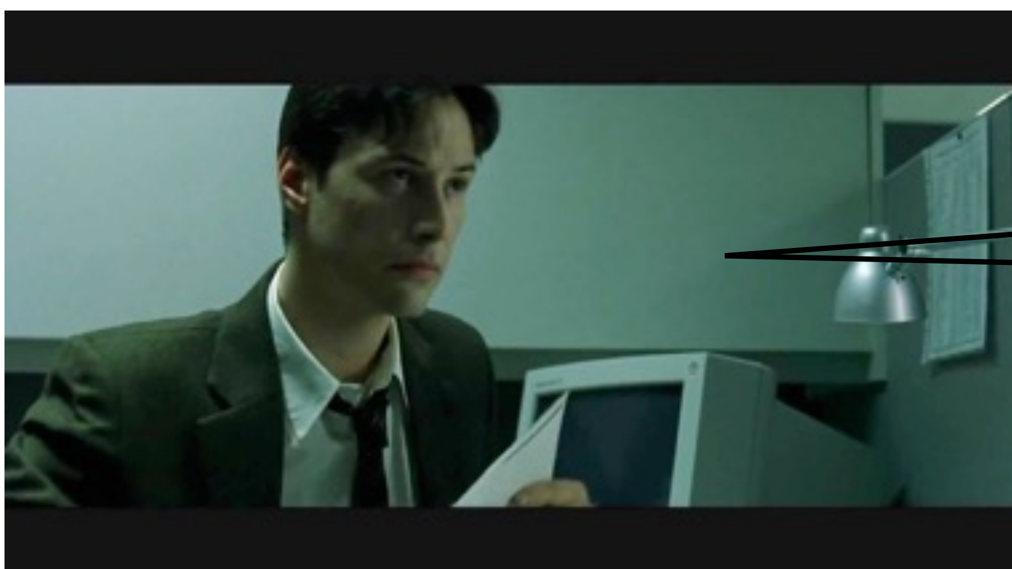
$$\{L : L \subseteq \Sigma^*\} = \mathcal{P}(\Sigma^*)$$ is uncountable.

Maybe all uncomputable problems are uninteresting ?

Working at Matrix Inc.

Debugging Trinity's code is taking too much time.

I think she keeps writing infinite loops.

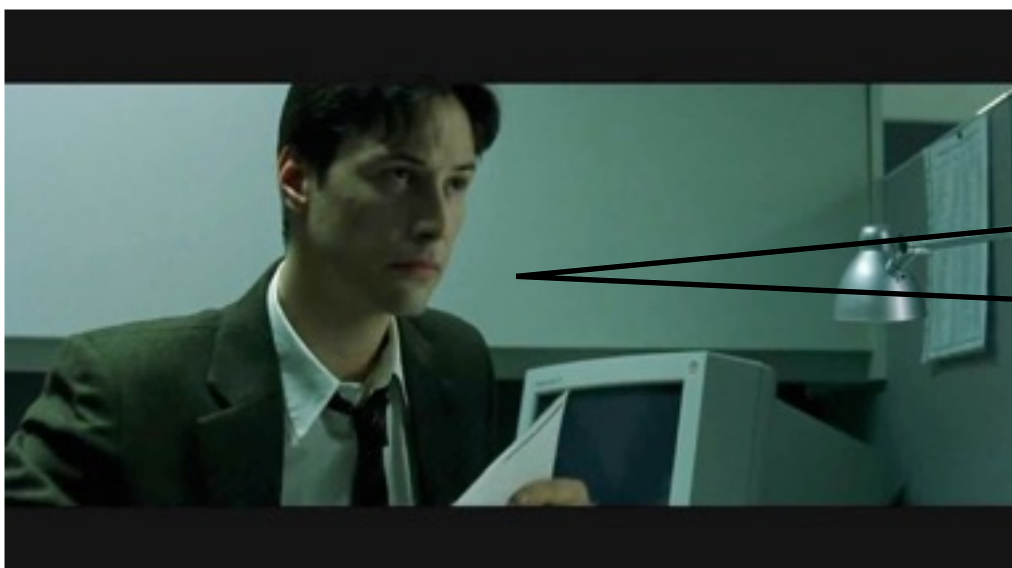


I'm the one.
I can fix anything.

Working at Matrix Inc.

Debugging Trinity's code is taking too much time.

I think she keeps writing infinite loops.

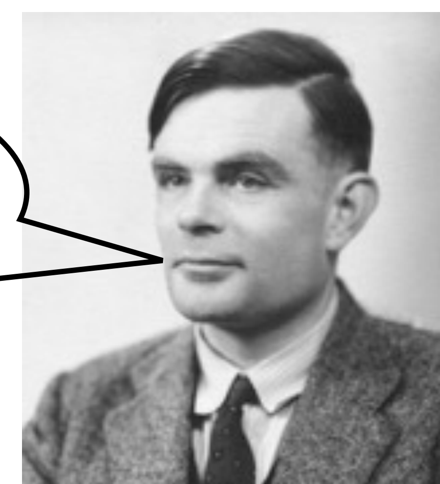


I'll first write a program that checks for infinite loops.

Halting Analyzer Program

How do you write such a program?

Dude, you might be the “One”,
but this is impossible!



An explicit uncomputable problem

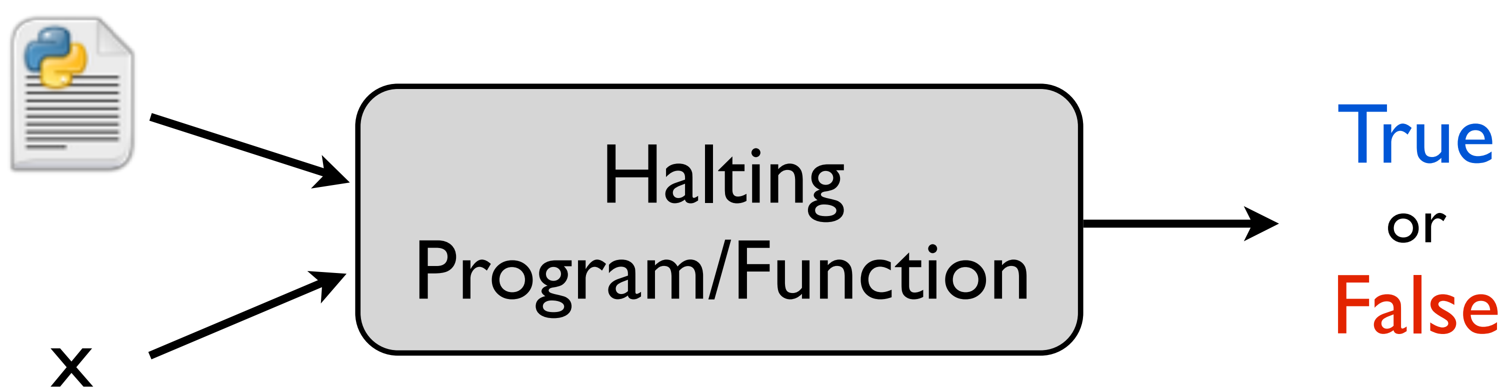
Theorem: The **halting problem** is uncomputable.

Proof by Python:

Halting Problem

Inputs: A Python program file.
An *input* to the program.

Outputs: True if the program halts for the given *input*.
False otherwise.



Halting problem is uncomputable

Assume such a program exists:

```
def halt(program, inputToProgram):
    # program and inputToProgram are both strings
    # Returns True if program halts when run with inputToProgram
    # as its input.

def turing(program):
    if (halt(program, program)):
        while True:
            pass   # a pass statement does nothing
    return None
```

What happens when you call turing(*turing*) ?

if halt(*turing*, *turing*) ----> turing doesn't terminate

if not halt(*turing*, *turing*) ----> turing terminates ☐

That was a diagonalization argument

```
def turing(program):
    if (halt(program, program)):
        while True:
            pass    # a pass statement does nothing
    return None
```

	$\langle f_1 \rangle$	$\langle f_2 \rangle$	$\langle f_3 \rangle$	$\langle f_4 \rangle$	$\cdots$
f_1	**∞**	∞	H	∞	
f_2	H	**H**	H	∞	
f_3	∞	∞	**H**	H	$\cdots$
f_4	∞	H	H	**∞**	
$\vdots$		$\vdots$			
turing	H	∞	∞	H	$\cdots$

Halting problem is uncomputable

Proof by a theoretical computer scientist:

$$\text{HALT} = \{\langle M, x \rangle : M \text{ halts on input } x\}$$

Suppose M_{HALT} decides HALT.

Consider the following TM (let's call it M_{TURING}):

M_{TURING}

Treat the input as $\langle M \rangle$ for some TM M.

Run M_{HALT} with input $\langle M, M \rangle$.

If it accepts, go into an infinite loop.

If it rejects, accept.

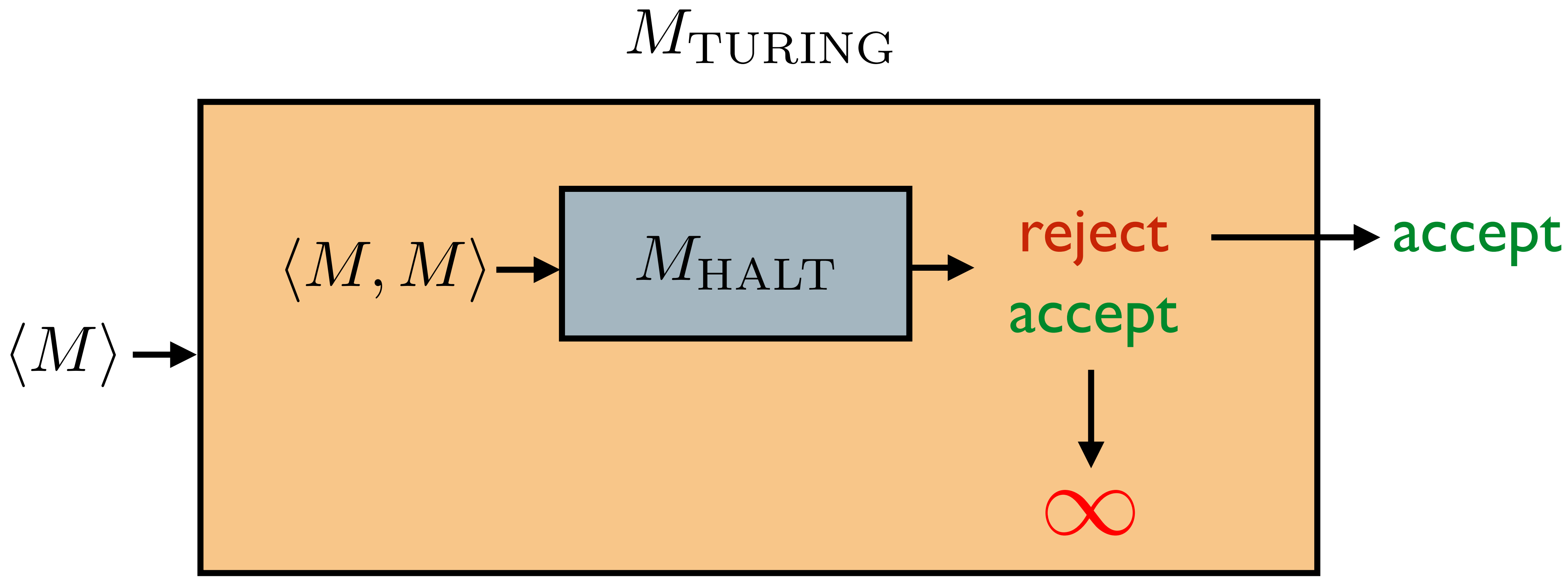
Halting problem is uncomputable

Proof by a theoretical computer scientist:

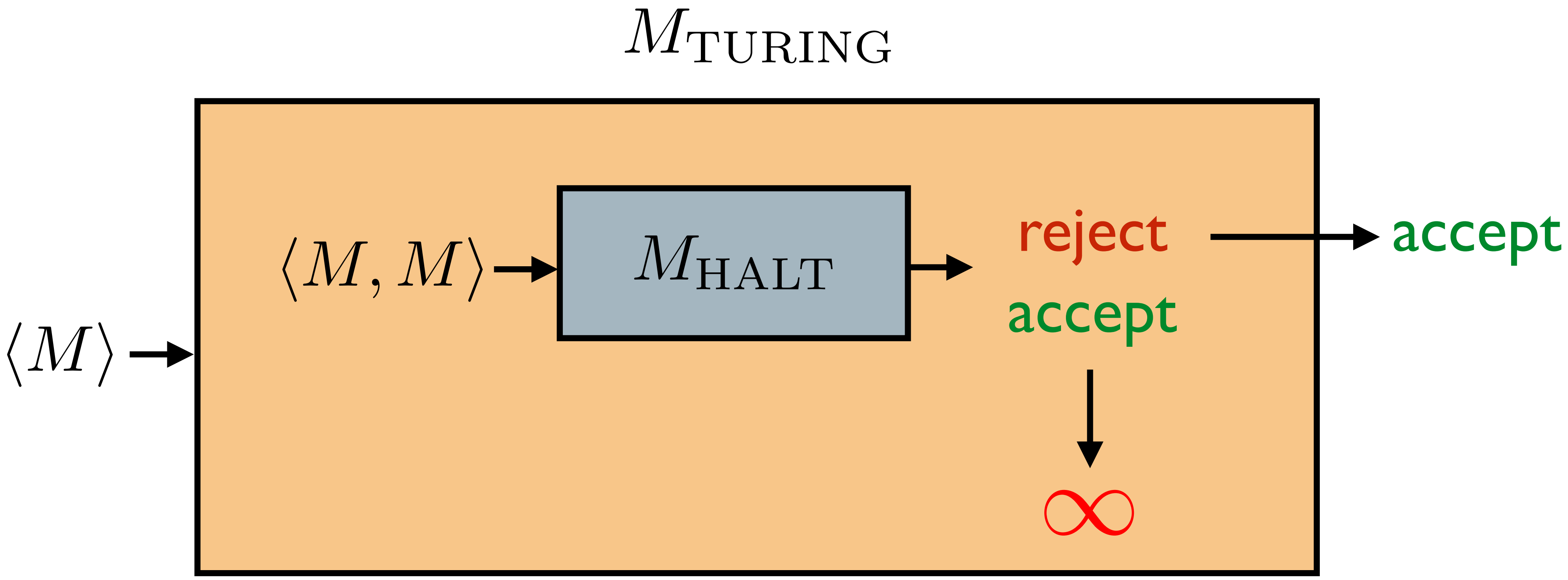
$$\text{HALT} = \{\langle M, x\rangle : M \text{ halts on input } x\}$$

Suppose M_{HALT} decides HALT.

Consider the following TM (let's call it M_{TURING}):



Halting problem is uncomputable



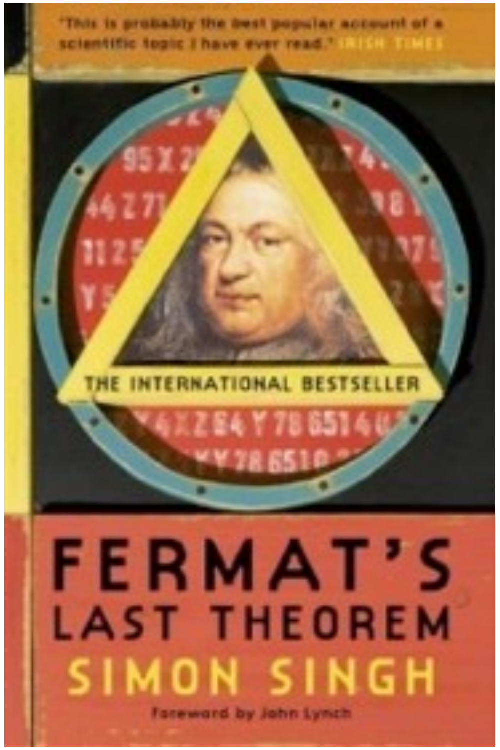
What happens when $\langle M_{\text{TURING}} \rangle$ is input to M_{TURING}?

So what?

- No debugger program.

- Consider the following program:

```
def fermat():
    t = 3
    while (True):
        for n in xrange(3, t+1):
            for x in xrange(1, t+1):
                for y in xrange(1, t+1):
                    for z in xrange(1, t+1):
                        if (x**n + y**n == z**n): return (x, y, z, n)
        t += 1
```



Question: Does this program halt?

So what?

- Reductions to other interesting problems
(show other interesting problems are as hard as the halting problem)

Entscheidungsproblem

Is there a finitary procedure to determine the validity of a given logical expression?

e.g. $\neg\exists x, y, z, n \in \mathbb{N} : (n \geq 3) \wedge (x^n + y^n = z^n)$

(Mechanization of mathematics)

Hilbert's 10th Problem

Is there a program to determine if a given multivariate polynomial with integral coefficients has an integral solution?

So what?

Different laws of physics ----->

Different computational devices ----->

Every problem computable (?)

Can you come up with sensible laws of physics
such that the Halting Problem becomes computable?

Let's show some other uncomputable problems.

Reduction

A central concept used to compare the "difficulty" of problems.

will differ based on context

Now we are interested in decidability vs undecidability (computability vs uncomputability)

Want to define: $A \leq B$

B is at least as hard as A (with respect to decidability).

i.e., B decidable $\implies$ A decidable

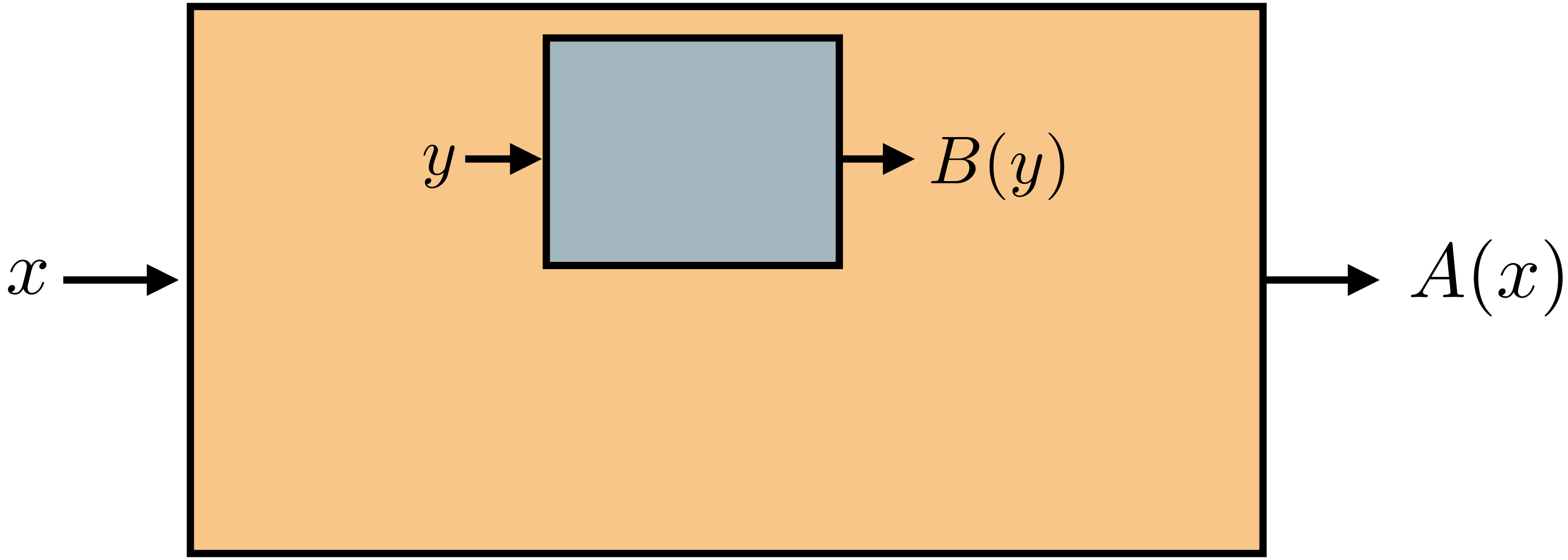
A undecidable $\implies$ B undecidable

Reduction

Definition:

$A \leq_T B$ (A reduces to B):

if it is possible to decide A
using an algorithm for deciding B as a subroutine.

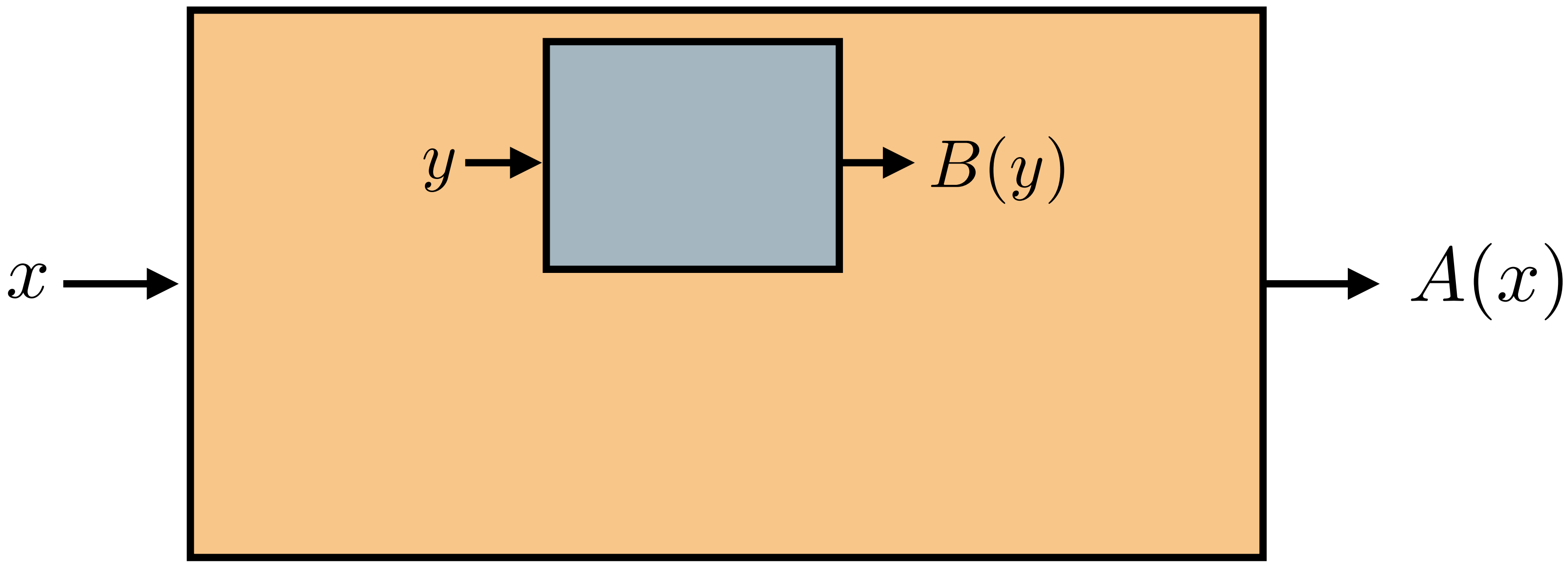


Reduction

If $A \leq_T B$ (A reduces to B) :

B decidable $\Longrightarrow$ A decidable

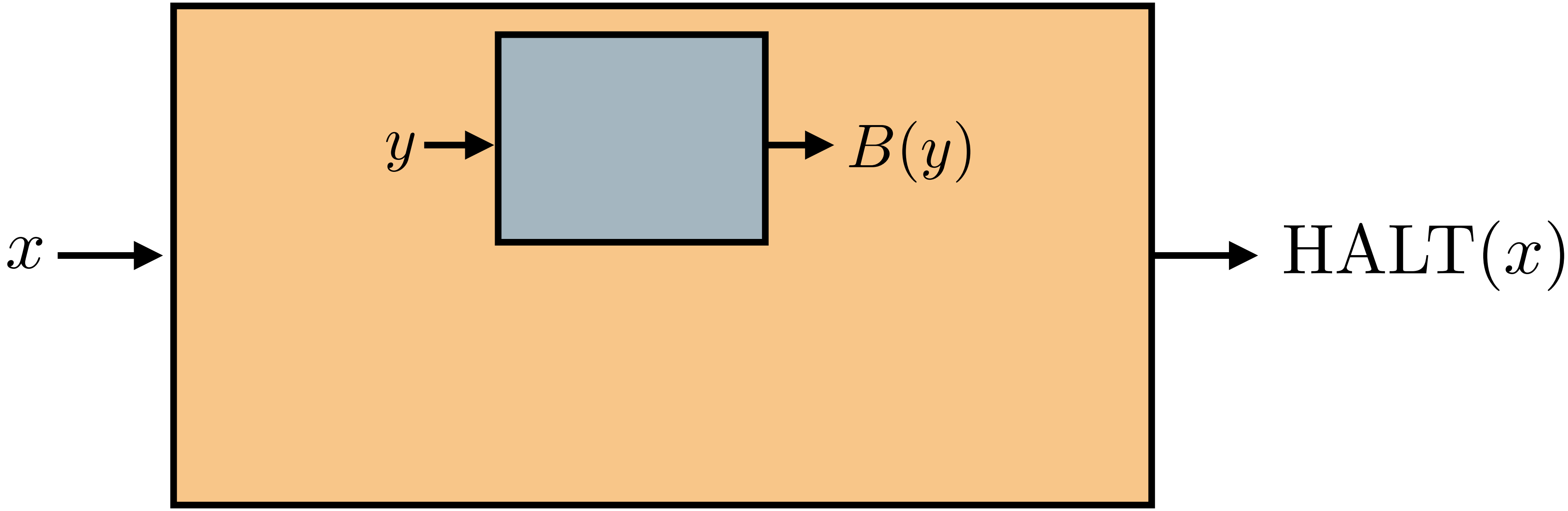
A undecidable $\Longrightarrow$ B undecidable



Reduction

If $\text{HALT} \leq_T B$ (HALT reduces to B) :

B is **not** decidable.



Example 1: ACCEPTS

Theorem:

$\text{ACCEPTS} = \{\langle M, x \rangle : M \text{ is a TM that accepts } x\}$

is **undecidable**.

$\langle M, x \rangle$ is in the language $\Longrightarrow$

x leads to an accept state in M.

$\langle M, x \rangle$ is not in the language $\Longrightarrow$

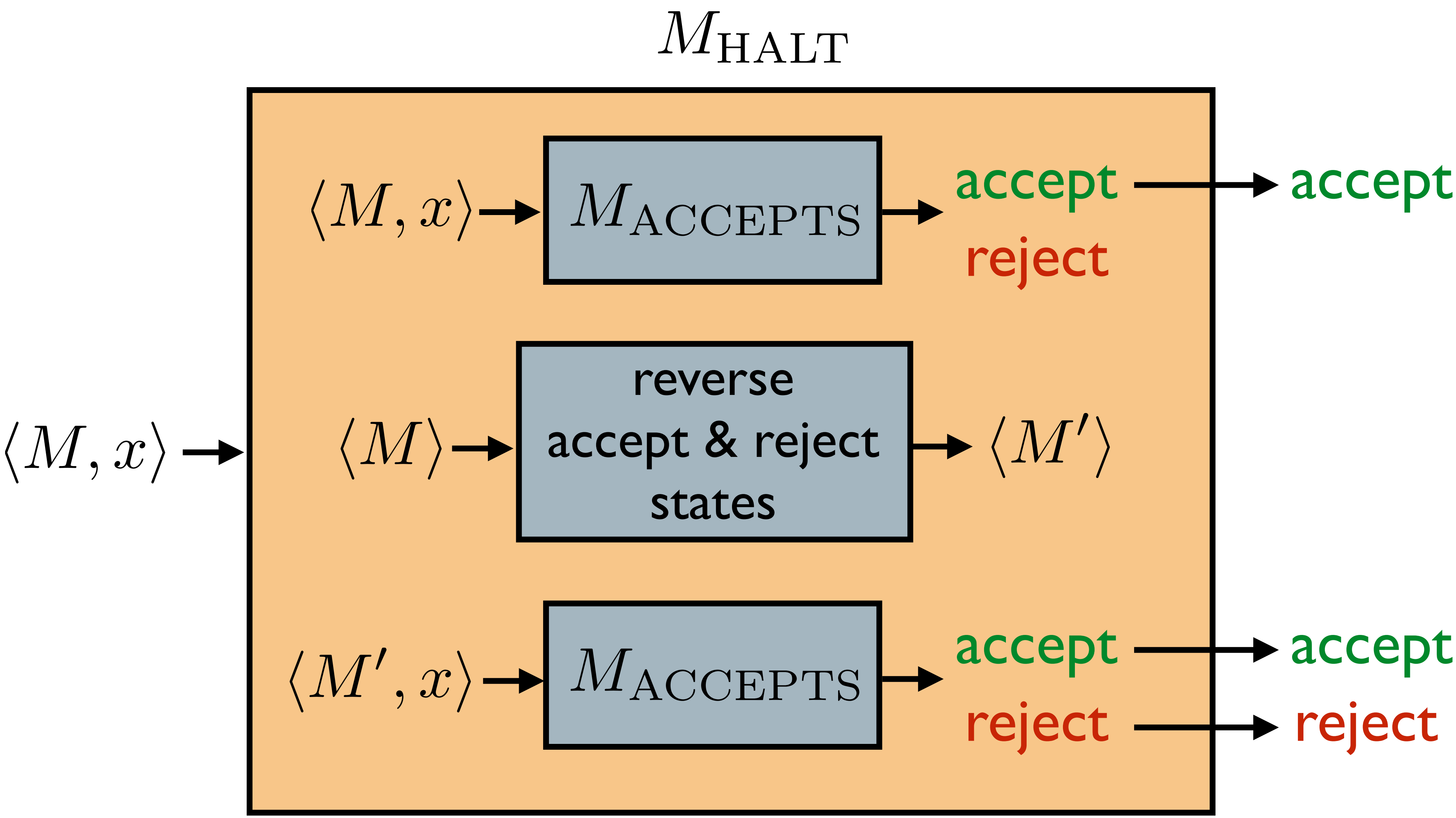
x leads to a reject state, **or** M loops forever.

$\langle M, x \rangle \in \text{HALT}$ if x leads to an accept or reject state.

Example 1: ACCEPTS

$$\text{ACCEPTS} = \{\langle M, x\rangle : M \text{ is a TM that accepts } x\}$$

Proof: (by picture)



Example 1: ACCEPTS

$\text{ACCEPTS} = \{\langle M, x \rangle : M \text{ is a TM that accepts } x\}$

Proof:

We will show $\text{HALT} \leq_T \text{ACCEPTS}$.

Let M_{ACCEPTS} be a TM that decides ACCEPTS.

Here is a TM that decides HALT:

On input $\langle M, x \rangle$, run $M_{\text{ACCEPTS}}(\langle M, x \rangle)$.

If it accepts, accept.

Reverse the accept and rejects states of M. Call it M'.

Run $M_{\text{ACCEPTS}}(\langle M', x \rangle)$.

If it accepts (M rejects x), accept.

Reject.

Reductions are transitive

If $A \leq_T B$ and $B \leq_T C$, then $A \leq_T C$.

(follows directly from the definition)

Example 2: EMPTY

Theorem:

$\text{EMPTY} = \{\langle M \rangle : M \text{ is a TM that accepts no strings}\}$

is **undecidable**.

Suffices to show $\text{ACCEPTS} \leq_T \text{EMPTY}$

since we showed $\text{HALT} \leq_T \text{ACCEPTS}$.

exercise or recitation or homework

Example 3: REG

Theorem:

$\text{REG} = \{\langle M \rangle : M \text{ is a TM and } L(M) \text{ is regular}\}$ is **undecidable**.

exercise or recitation or homework

Interesting Observation

To show a negative result (that there is no algorithm)

we are showing a positive result (that there is a reduction)

Undecidable problems not involving Turing Machines

Entscheidungsproblem

Determining the validity of a given FOL sentence.

e.g. $\neg \exists x, y, z, n \in \mathbb{N} : (n \geq 3) \wedge (x^n + y^n = z^n)$

Undecidable!

Proved in 1936 by Turing.


230 A. M. TURING [Nov. 12,

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable numbers, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

In §§ 9, 10 I give some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable. In particular, I show that certain large classes of numbers are computable. They include, for instance, the real parts of all algebraic numbers, the real parts of the zeros of the Bessel functions, the numbers π, e, etc. The computable numbers do not, however, include all definable numbers, and an example is given of a definable number which is not computable.

Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable. In § 8 I examine certain arguments which would seem to prove the contrary. By the correct application of one of these arguments, conclusions are reached which are superficially similar to those of Gödel†. These results

† Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I", *Monatshefte Math. Phys.*, 38 (1931), 173–198.

Hilbert's 10th Problem

Determining if a given multivariate polynomial with integral coefficients has an integer root.

e.g. $5xy^2z + 8yz^3 + 100x^{99}$

Undecidable!

Proved in 1970 by Matiyasevich-Robinson-Davis-Putnam.

Does it have a **real** root? **Decidable!**

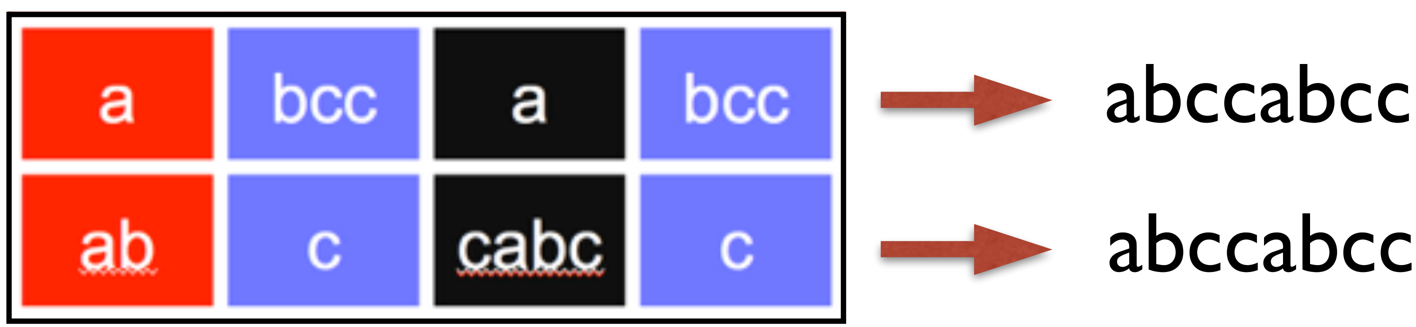
Proved in 1951 by Tarski.

Does it have a **rational** root? **No one knows!**

Post's Correspondence Problem

Input: A finite collection of "dominoes",
having strings written on each half.

Output: Accept if it is possible to match the strings.



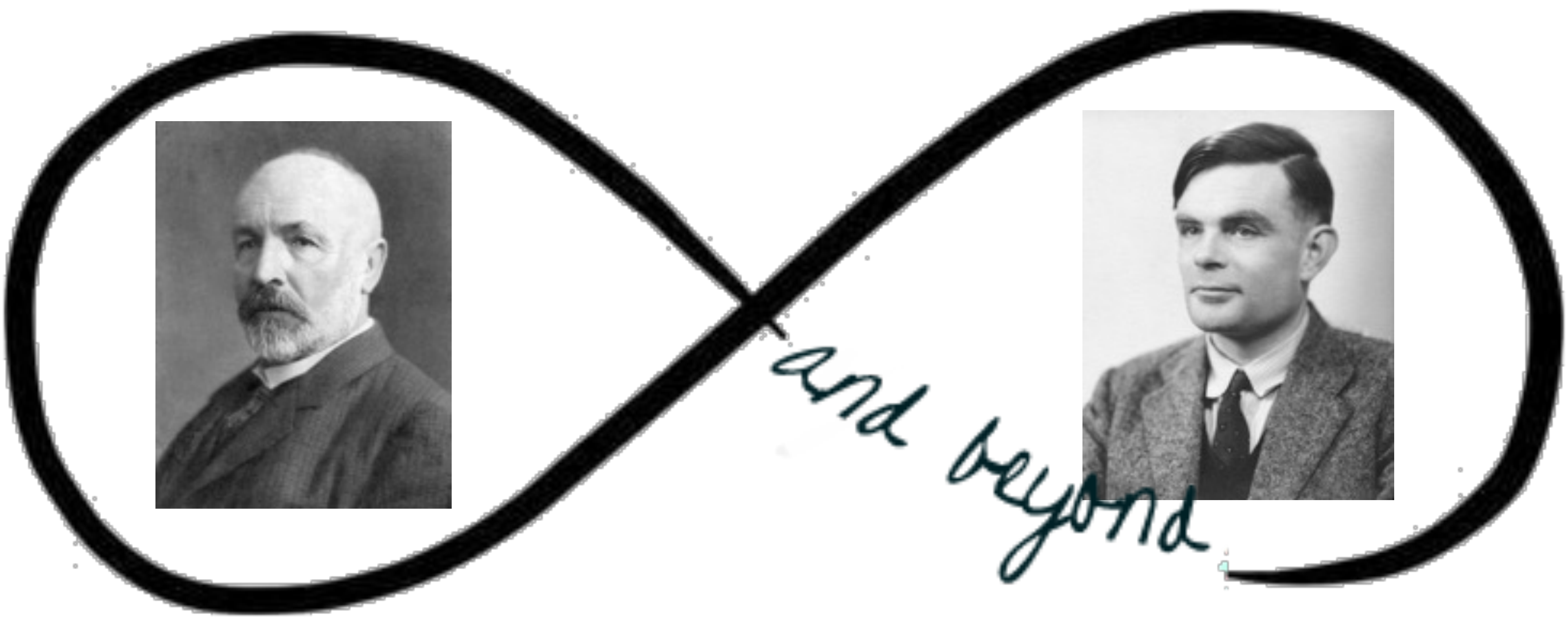
Undecidable!

Proved in 1946 by Post.

Most problems are undecidable.

Some very interesting problems undecidable.

But most interesting problems are decidable.


and beyond