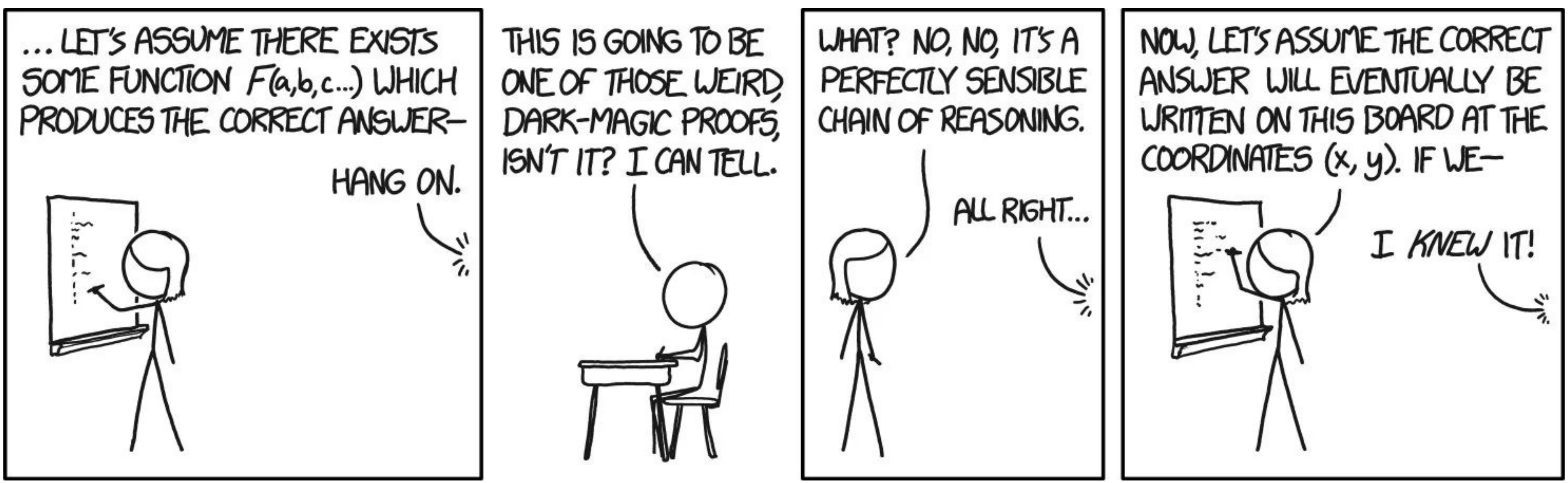
Undecidability
+
Diagonalization



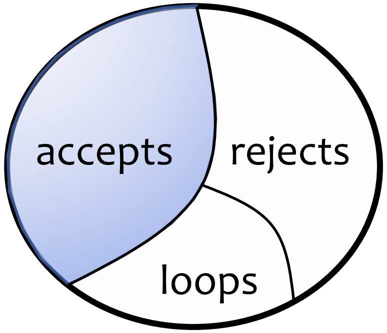
Extra Midterm Review Sessions

- Daphne: Thursday 2/22 6-8pm LMBE 1130
  - Topic: Turing Reductions and Dynamic Programming
- Eric K: Monday 3/4 6-8pm BBB 1670
  - Topic: past exams

Review: Undecidability

Question: What are the possible outcomes of a TM **M**?

Answer: **M** either (i) accepts, (ii) rejects, *or* (iii) it *"loops" (forever)*



Definition: A Turing Machine **M** ***decides*** a language L if it:

1. *accepts* every string in L, and
2. *rejects* every string not in L

(and never loops forever)

A *language* L is ***decidable*** if there is a TM that decides L.

Otherwise L is ***undecidable***.

Review: Church–Turing Thesis

"Any natural notion of being 'algorithmically computable' is captured by Turing Machines."

This is not a theorem,

but everyone seems to believe it.

We will assume it's true and proceed from there.

Anything that can be computed by Python, C++, a quantum computer, LaTex, pseudocode, etc. can be computed by a Turing Machine.

This statement we know is true

Question: To prove that a language L is decidable, must we design a TM?

Question: If a language L is decidable, then must L $\cup$ {ε} be decidable?

Existence of Undecidable Languages

To show that there **exists** an undecidable language (i.e. a language that no TM decides), we would like to show that:

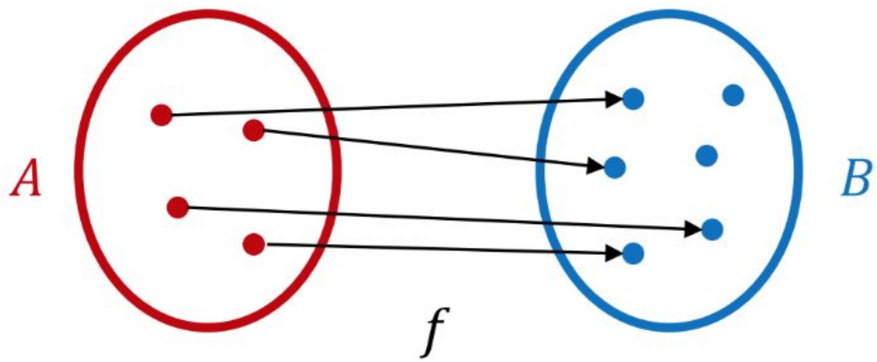
total # TMs < total # languages.

But both of these quantities are **infinite** so what does that even mean?

To infinity and beyond!

203 Review: Functions and Set Cardinality

* A (total) function $f: A \to B$ maps each element $a \in A$ to an element $f(a) \in B$.
* A function is ***1-to-1 (injective)*** if each element $a \in A$ is mapped to a <u>*different*</u> element $f(a) \in B$.
  * Formally: $\forall a_1, a_2 \in A.\ a_1 \neq a_2 \implies f(a_1) \neq f(a_2)$

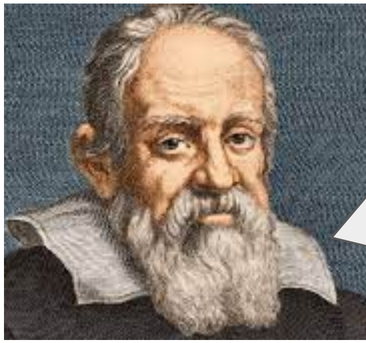


"cardinality of A"

* If a 1-to-1 function $f: A \to B$ exists, then we say $|A| \leq |B|$.

Warning: properties of "≤" for finite values do not necessarily apply to infinite values

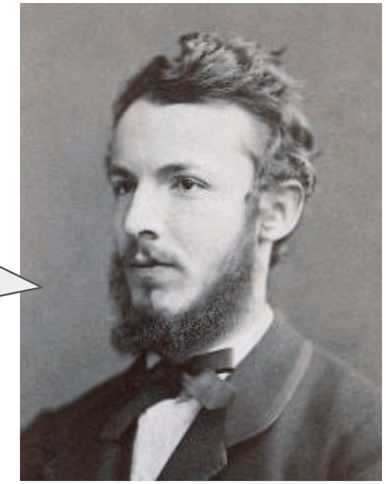
203 Review: Countability



The attributes “equal,” “greater,” and “less,” are not applicable to infinite, but only to finite, quantities.

Galileo (1638)

Actually yes they are



Georg Cantor (1895)

* **Definition:** A set S is ***countable*** if it is “no larger than” the naturals $\mathbb{N} = \{0,1,2,\ldots\}$, i.e. $|S| \leq |\mathbb{N}|$.
* **Equivalently :** S is countable if there exists a 1-to-1 (injective) function $f: S \to \mathbb{N}$.
* We can also show S is countable by demonstrating how to list all the elements in S such that each element $s \in S$ appears *somewhere* on the list. Why?

Is ___ a countable set?

The integers from 1 to 10?

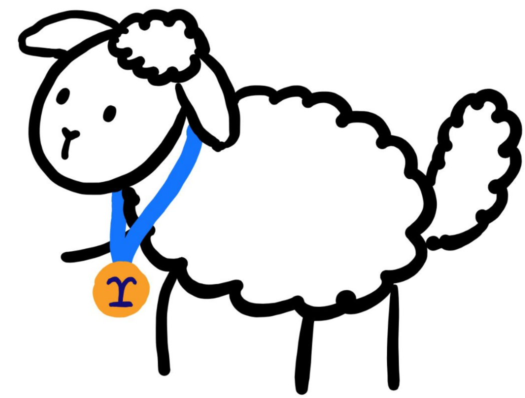
The even numbers?

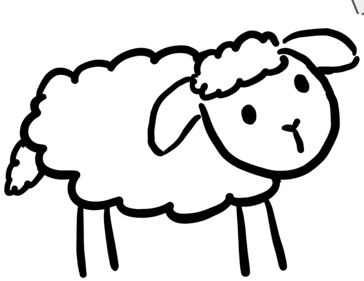
The integers?

The rationals?

The reals in (0,1)?

Natural Number	Real Number in (0,1)
1	.1
2	.2
3	.3
…	…
11	.11
12	.12
…	…





Diagonalization

Suppose for contradiction we could write a list of all the reals in (0,1).

Why doesn't the same argument work to show that the integers are uncountable?

Kronecker

I don't know what predominates in Cantor's theory – philosophy or theology, but I am sure that there is no mathematics there.

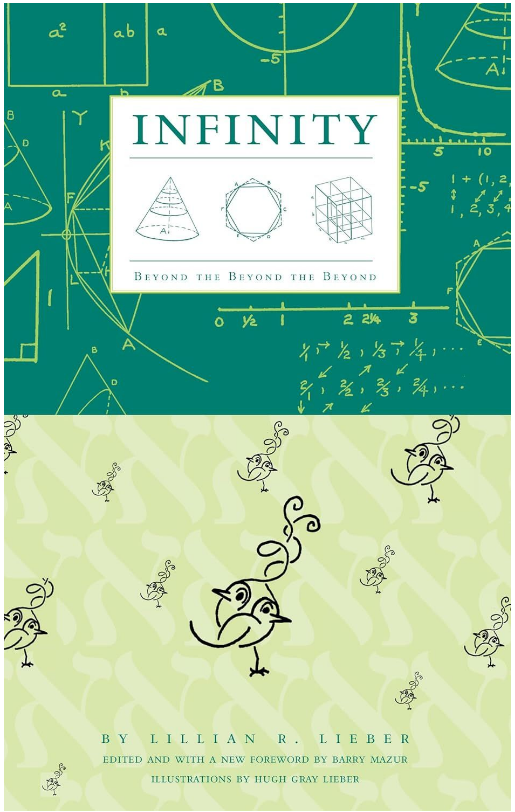
I believe, and hope, that a future generation will laugh at this hocus pocus.

Wittgenstein

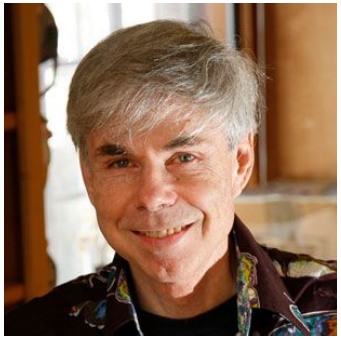
A Bag of Reals

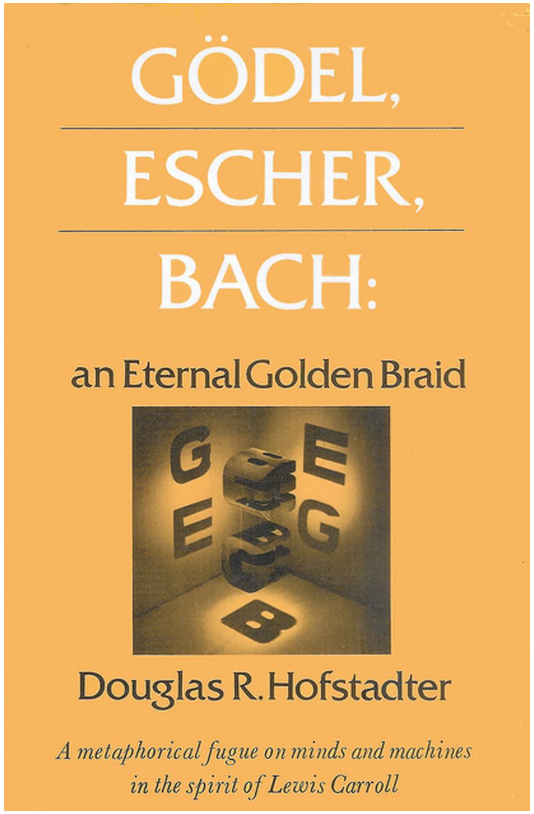
Book Recommendations about Infinities

Mark Brehob:



Chris Peikert:





Is ___ a countable set?

The set of TMs?

(any TM can be represented by a finite-length binary string)

The set of possible TM inputs from {0,1}*?

(i.e. the set of all finite-length binary strings)

The set of languages over the alphabet {0,1}?

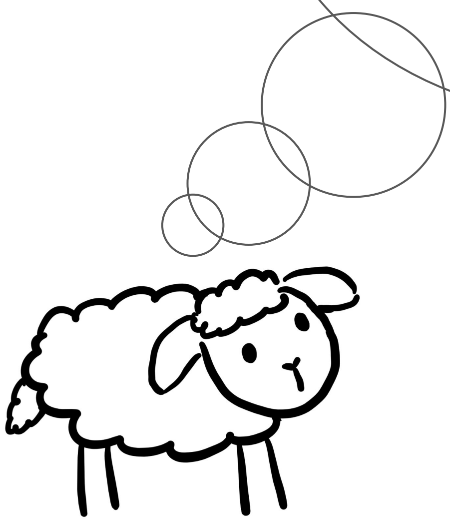
(i.e. the set of all infinite-length binary strings. Why?)

Formal Definition of a Turing Machine

* A Turing machine is a 7-tuple:

$$M = \langle Q, \Gamma, \Sigma, \delta, q_{start}, q_{accept}, q_{reject} \rangle$$

  * Q = set of **states**
  * Σ = the **input** alphabet (typically {0,1} but not always)
  * $\perp$ = the **blank symbol**
  * Γ = the **tape alphabet** where generally $\Gamma = \Sigma \cup \{\perp\}$
  * $q_{start} \in Q$, = the **initial state**
  * $F = \{q_{accept}, q_{reject}\} \subseteq Q$, = the set of **final states** (one accepting state and one rejecting state)
  * $\delta: (Q \backslash F) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ = the **transition function**



Existence of Undecidable Languages

Task: Convince yourself that we can use diagonalization to construct an undecidable language.

{0,1}* (i.e. TM inputs) →

TMs ↓

	s1	s2	s3	s4	s5	s6	...
M1	1	0	0	1	1	0	...
M2	0	0	1	0	0	0	...
M3	1	1	1	1	1	1	...
M4	0	0	0	0	0	0	...
M5	1	0	1	0	0	0	...
...	...	...	...	...	...	...	...

A Bag of Languages