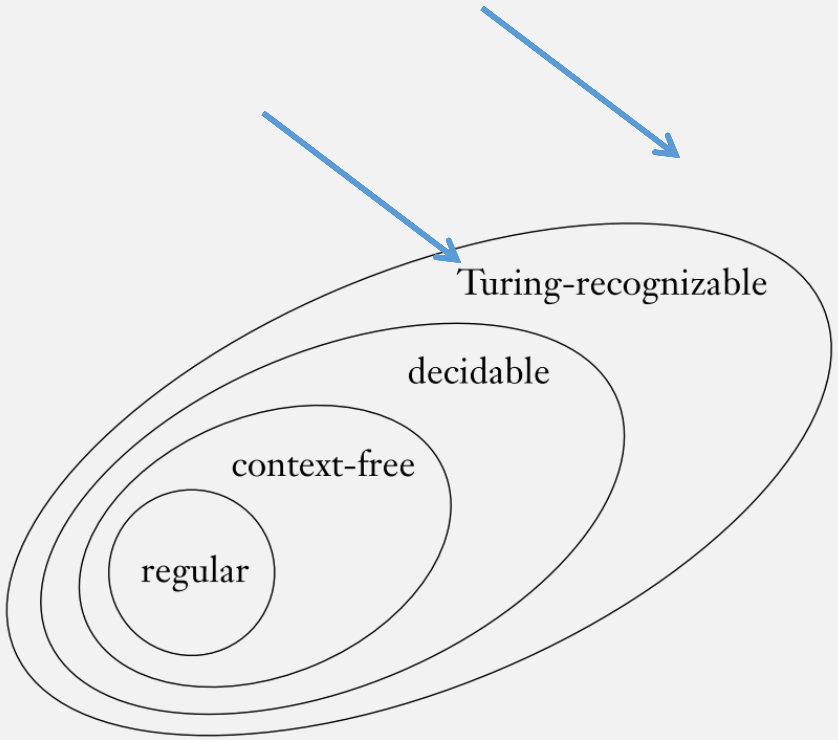
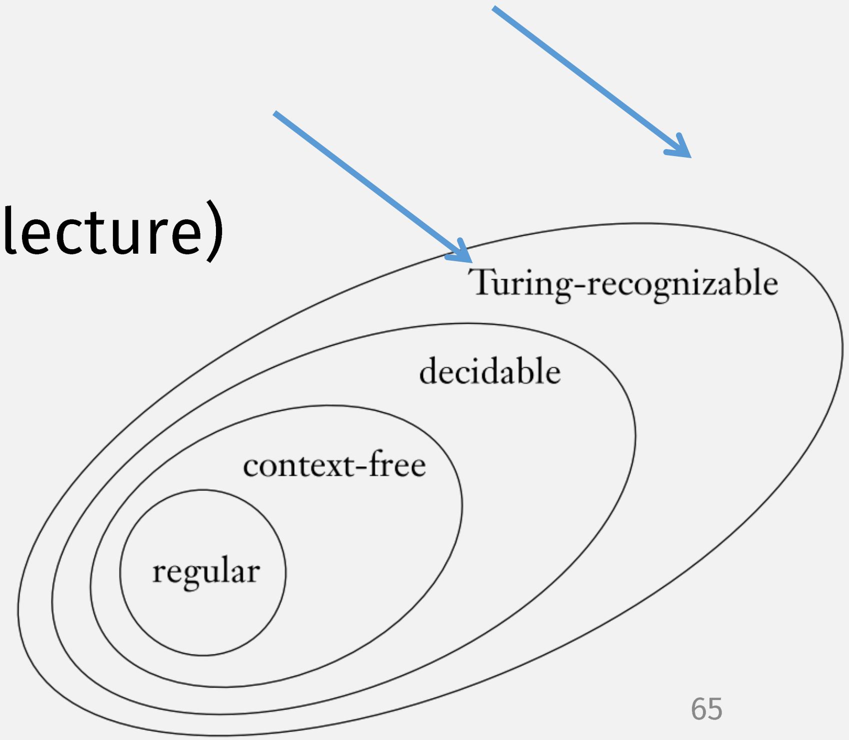
Undecidability

Wed March 31, 2021

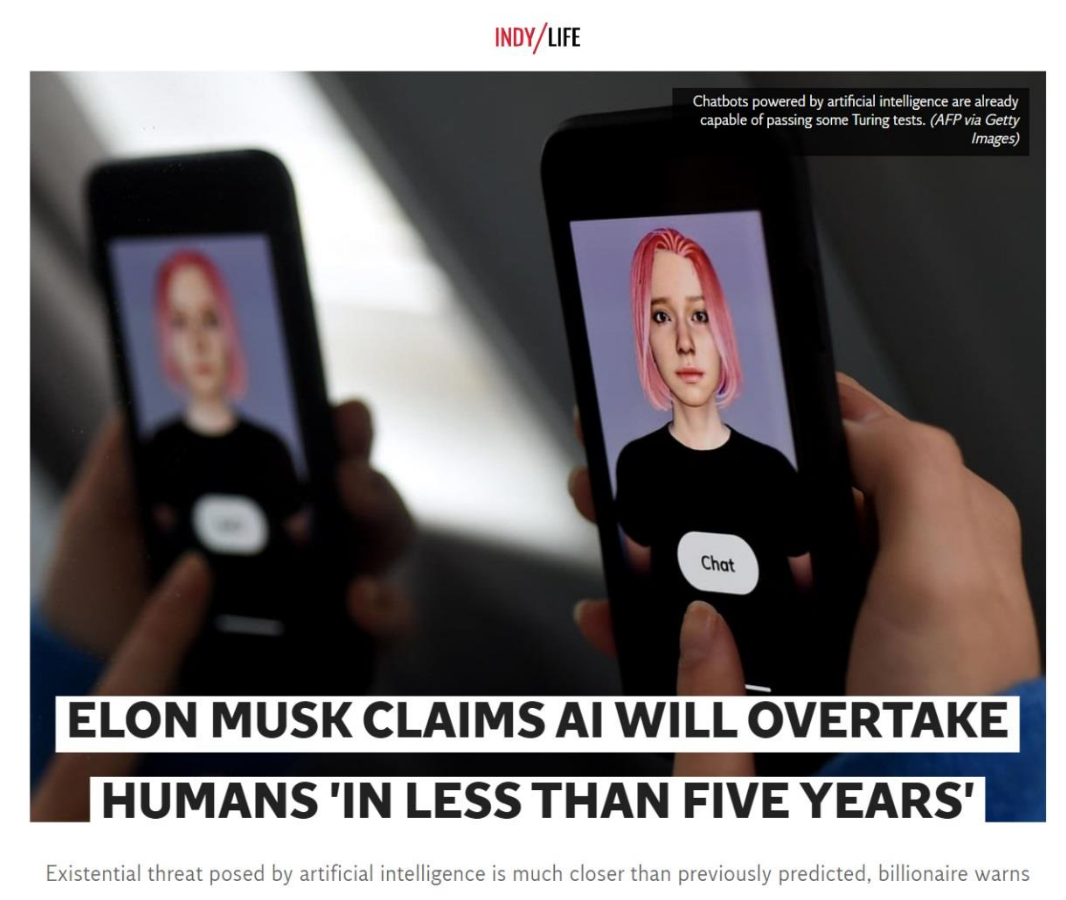


Announcements

- HW 7 due Sun 4/4 11:59pm EST
    - Remember to use your “library” of theorems
- HW 8 posted
    - due Sun 4/11 11:59pm EST
    - Covers Ch 4-5 material (starting with today’s lecture)



Warning: AI is Taking Over Soon

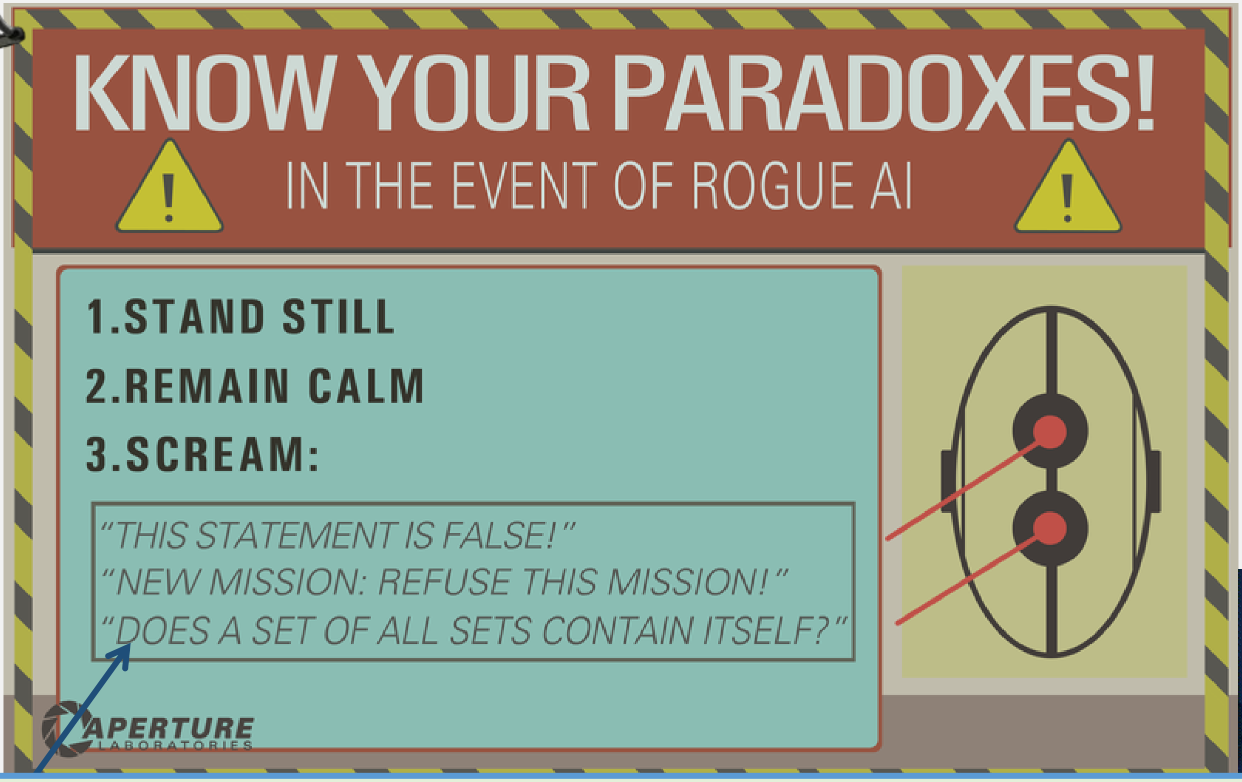

INDY/LIFE

Chatbots powered by artificial intelligence are already capable of passing some Turing tests. *(AFP via Getty Images)*

ELON MUSK CLAIMS AI WILL OVERTAKE HUMANS 'IN LESS THAN FIVE YEARS'

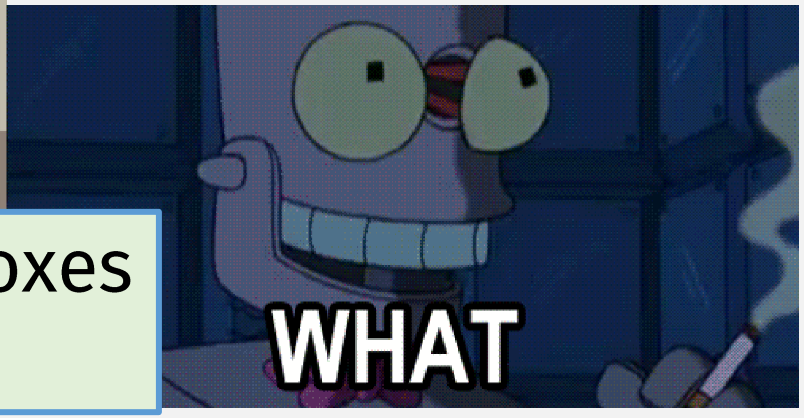
Existential threat posed by artificial intelligence is much closer than previously predicted, billionaire warns

There's Hope (If You Pay Attention Today)



Bertrand Russell's Paradox (1901)

<u>Today</u>: A method for creating paradoxes (used by Russell and others)



Recap: Decidability of Regular and CFLs

- $A_{\text{DFA}} = \{\langle B, w\rangle \mid B \text{ is a DFA that accepts input string } w\}$ Decidable
- $A_{\text{NFA}} = \{\langle B, w\rangle \mid B \text{ is an NFA that accepts input string } w\}$ Decidable
- $A_{\text{REX}} = \{\langle R, w\rangle \mid R \text{ is a regular expression that generates string } w\}$ Decidable
- $E_{\text{DFA}} = \{\langle A\rangle \mid A \text{ is a DFA and } L(A) = \emptyset\}$ Decidable
- $EQ_{\text{DFA}} = \{\langle A, B\rangle \mid A \text{ and } B \text{ are DFAs and } L(A) = L(B)\}$ Decidable
- $A_{\text{CFG}} = \{\langle G, w\rangle \mid G \text{ is a CFG that generates string } w\}$ Decidable
- $E_{\text{CFG}} = \{\langle G\rangle \mid G \text{ is a CFG and } L(G) = \emptyset\}$ Decidable
- $EQ_{\text{CFG}} = \{\langle G, H\rangle \mid G \text{ and } H \text{ are CFGs and } L(G) = L(H)\}$ *Undecidable?*
- $A_{\text{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$ *Undecidable?*

<u>Thm</u>: A_{TM} is Turing-recognizable

$$A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$$

U = "On input $\langle M, w\rangle$, where M is a TM and w is a string:

1. Simulate M on input w.
2. If M ever enters its accept state, *accept*; if M ever enters its reject state, *reject*."

- U = "run" function for TMs
  - Computer that can simulate other computers
  - i.e., "The Universal Turing Machine"
  - <u>Problem</u>: U loops when M loops

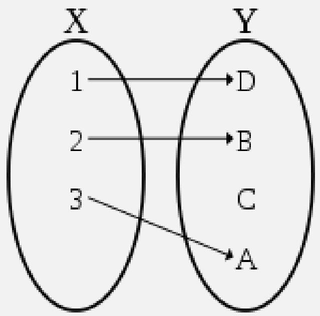
Thm: A_{TM} is undecidable

$A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$

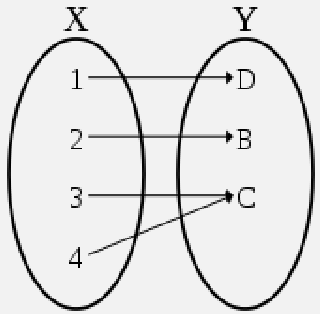
- ???

Kinds of Functions (a fn maps DOMAIN -> RANGE)

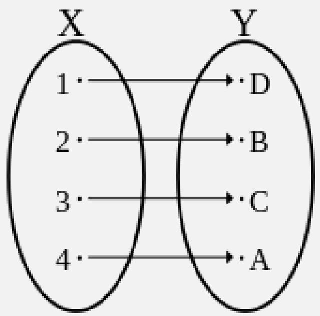
- **Injective**
  - A.k.a., “one-to-one”
  - Every element in DOMAIN has a unique mapping
  - How to remember:
    - DOMAIN is mapped “in” to the RANGE



- **Surjective**
  - A.k.a., “onto”
  - Every element in RANGE is mapped to
  - How to remember:
    - “Sur” = “over” (eg, survey); DOMAIN is mapped “over” the RANGE



- **Bijective**
  - A.k.a., “correspondence” or “one-to-one correspondence”
  - Is both injective and surjective
  - Unique pairing of every element in DOMAIN and RANGE



Countability

- A set is “countable” if it is:
  - Finite
  - Or, there exists a bijection between the set and the natural numbers
    - This set is then considered to have the same size as the set of natural numbers
    - This is called “countably infinite”

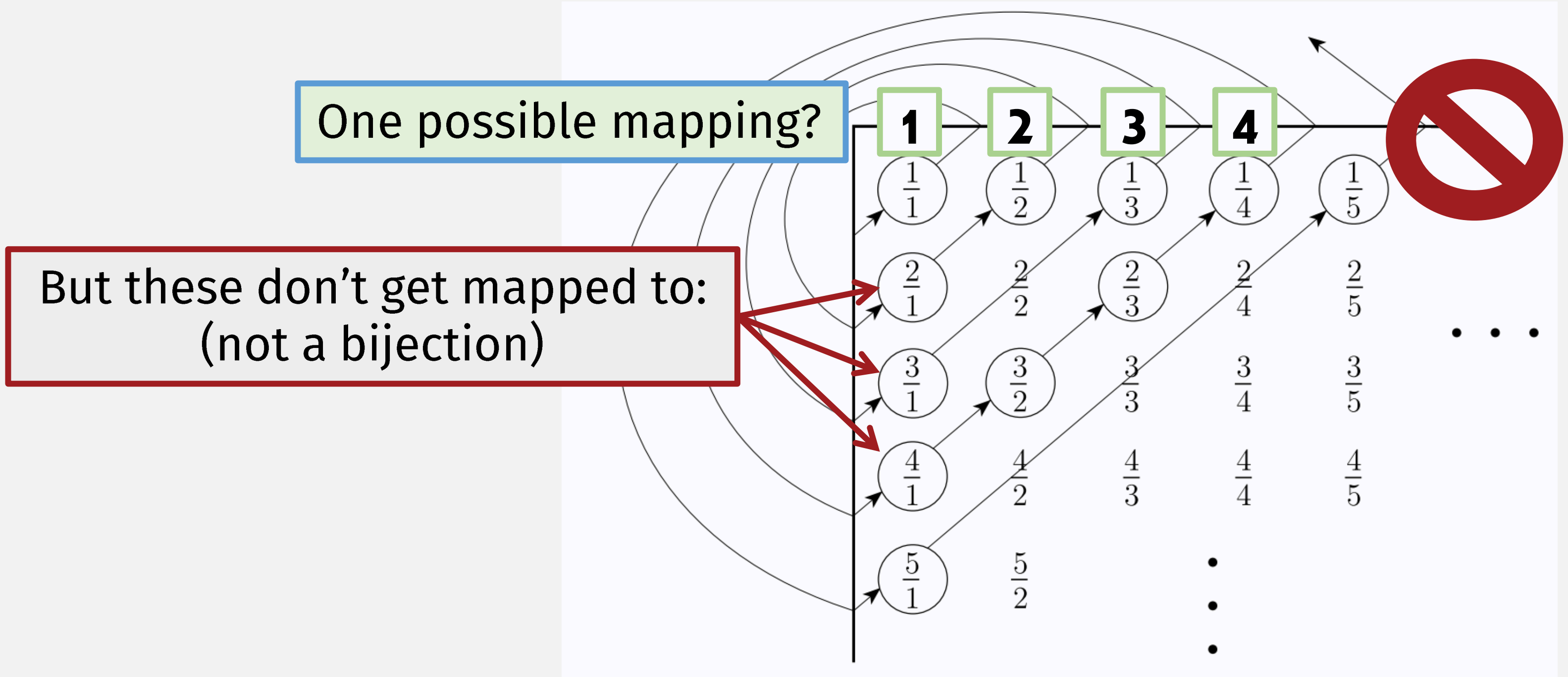
Exercise: Which set is larger?

- The set of:
  - Natural numbers, or
  - Even numbers?
- They are the **same** size! Both are countably infinite
  - Bijection:

n	$f(n) = 2n$
1	2
2	4
3	6
$\vdots$	$\vdots$

Exercise: Which set is larger?

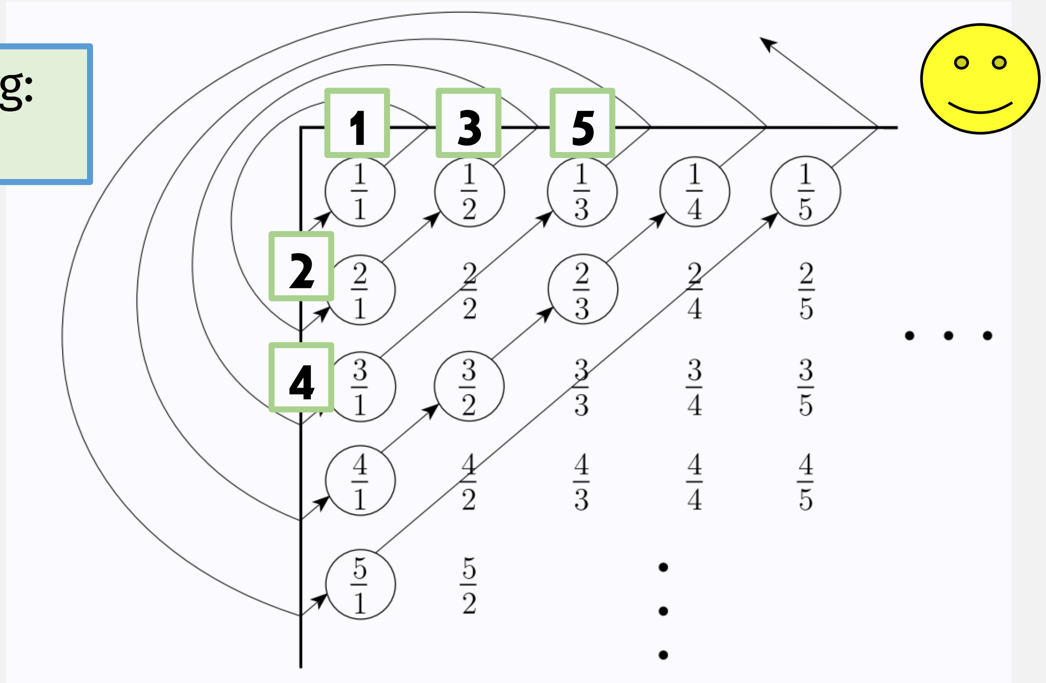
- The set of:
  - Natural numbers $\mathcal{N}$, or
  - Positive rational numbers? $\mathcal{Q} = \{\frac{m}{n} | m, n \in \mathcal{N}\}$
- They are the **same** size! Both are countably infinite



Exercise: Which set is larger?

- The set of:
  - Natural numbers $\mathcal{N}$, or
  - Positive rational numbers? $\mathcal{Q} = \{\frac{m}{n} | m, n \in \mathcal{N}\}$
- They are the **same** size! Both are countably infinite

Another mapping:
(is a bijection)



Exercise: Which set is larger?

- The set of:
  - Natural numbers, or $\mathcal{N}$
  - Real numbers? $\mathcal{R}$
- There are **more** real numbers. It is uncountably infinite.

This is called **“diagonalization”**

Proof, by contradiction:

- Assume a bijection between natural and real numbers exists.
  - This means that every real number should get mapped to.
- But we show that in any given mapping, … e.g.:

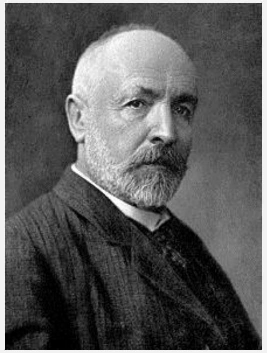
n	$f(n)$
1	$3.\underline{1}4159\ldots$
2	$55.5\underline{5}555\ldots$
3	$0.12\underline{3}45\ldots$
4	$0.500\underline{0}0\ldots$
$\vdots$	$\vdots$

  - Some real number is not mapped to …
  - E.g., any number that has different digits at each position:

$$x = 0.4641\ \ldots$$

  - This number is cannot included in mapping
  - Contradiction!

Georg Cantor



- Invented set theory
- Came up with countable infinity in 1873
- And uncountability:
  - And how to show uncountability with “diagonalization” technique

A formative day for Georg Cantor.

Diagonalization with Turing Machines

Diagonal: Result of Giving a TM its own Encoding as Input

All TM Encodings

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$	$\langle D \rangle$	$\cdots$
M_1	*accept*	*reject*	*accept*	*reject*		*accept*	
M_2	*accept*	*accept*	*accept*	*accept*	$\cdots$	*accept*	$\cdots$
M_3	*reject*	*reject*	*reject*	*reject*		*reject*	
M_4	*accept*	*accept*	*reject*	*reject*		*accept*	
$\vdots$			$\vdots$		$\ddots$		
D	*reject*	*reject*	*accept*	*accept*		?	
$\vdots$			$\vdots$				

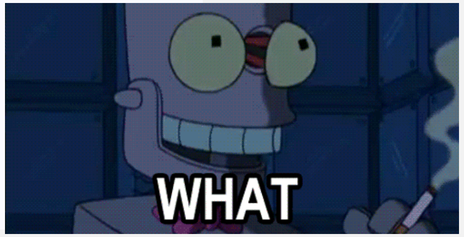
opposites

All TMs

Try to construct "opposite" TM

What should happen here?

TM D can't exist!



Thm: A_{TM} is undecidable

$$A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$$

Proof by contradiction:

- Assume A_{TM} is decidable. Then there exists a decider:

$$H(\langle M, w\rangle) = \begin{cases} accept & \text{if } M \text{ accepts } w \\ reject & \text{if } M \text{ does not accept } w \end{cases}$$

- If H exists, then we can create:

$D =$ "On input $\langle M\rangle$, where M is a TM:

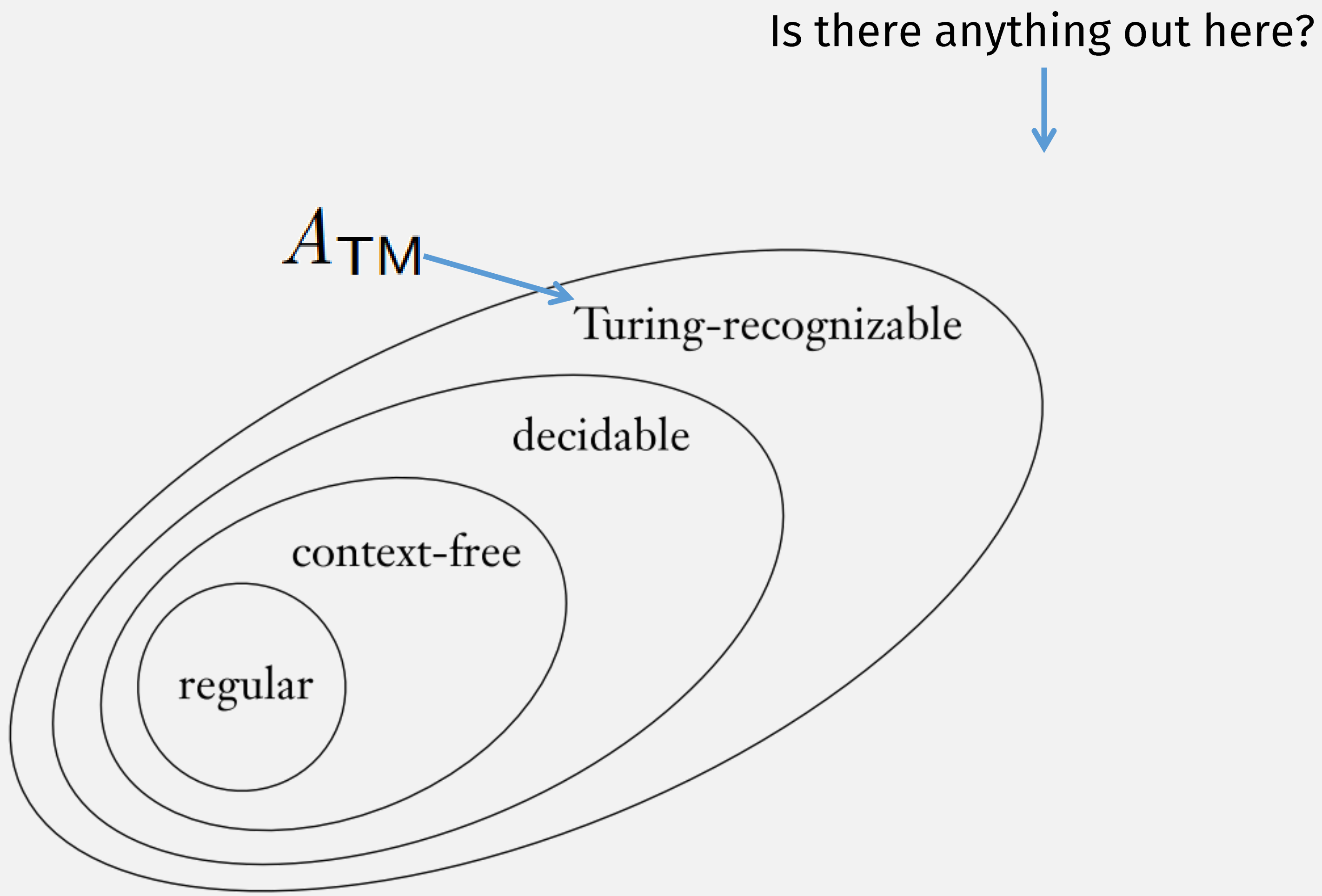
1. Run H on input $\langle M, \langle M\rangle\rangle$.

2. Output the opposite of what H outputs. That is, if H accepts, *reject*; and if H rejects, *accept*."

From the previous slide

- But D does not exist! Contradiction! So assumption is false.

Turing Unrecognizable?



Thm: Some langs are not Turing-recognizable

Proof: requires 2 lemmas

- Lemma 1: The **set of all languages** is *uncountable*
  - Proof: Show there is a bijection with another uncountable set …
    - … The set of all infinite binary sequences

- Lemma 2: The **set of all TMs** is *countable*

- Therefore, some language is not recognized by a TM

Mapping a Language to a Binary Sequence

All Possible Strings	$\Sigma^* = \{$	ε,	0,	1,	00,	01,	10,	11,	000,	001,	$\cdots \}$
Some Language (subset of above)	$A = \{$		0,		00,	01,			000,	001,	$\cdots \}$
Its (unique) Binary Sequence	$\chi_A =$	0	1	0	1	1	0	0	1	1	$\cdots$

Each digit represents one possible string:
- 1 if lang has that string,
- 0 otherwise

Thm: Some langs are not Turing-recognizable

Proof: **requires 2 lemmas**

- Lemma 1: The **set of all languages** is *uncountable*
  - Proof: Show there is a bijection with another uncountable set …
    - … The set of all infinite binary sequences
    - ➢ Now just prove set of infinite binary sequences is uncountable (hw8)
- Lemma 2: The **set of all TMs** is *countable*
  - Because every TM M can be encoded as a string $<M>$
  - And set of all strings is countable
- Therefore, some language is not recognized by a TM ∎

Co-Turing-Recognizability

- A language is **co-Turing-recognizable** if …
- … it is the complement of a Turing-recognizable language.

Thm: Decidable ⇔ Recognizable & co-Recognizable

=> If a language is decidable, then it is recognizable and co-recognizable

- Decidable => Recognizable:
  - A decider is a recognizer, bc decidable langs are a subset of recognizable langs
- Decidable => Co-Recognizable:
  - To create co-decider (which is also a co-recognizer) from a decider …
  - … switch reject/accept of all inputs

<= If a language is recognizable and co-recognizable, then it is decidable

Thm: Decidable ⇔ Recognizable & co-Recognizable

=> If a language is decidable, then it is recognizable and co-recognizable

- Decidable => Recognizable:
  - A decider is a recognizer, bc decidable langs are a subset of recognizable langs
- Decidable => Co-Recognizable:
  - To create co-decider (which is also a co-recognizer) from a decider …
  - … switch reject/accept of all inputs

<= If a language is recognizable and co-recognizable, then it is decidable

- Let $M1$ = recognizer for the language,
- And $M2$ = recognizer for its complement
- Decider M:
  - Run 1 step on $M1$,
  - Run 1 step on $M2$,
  - Repeat, until one machine accepts. If it's $M1$, accept. If it's $M2$, reject
- One of $M1$ or $M2$ must accept and halt, so M halts and is a decider

A Turing-unrecognizable language

- We've proved:

  A_{TM} is Turing-recognizable

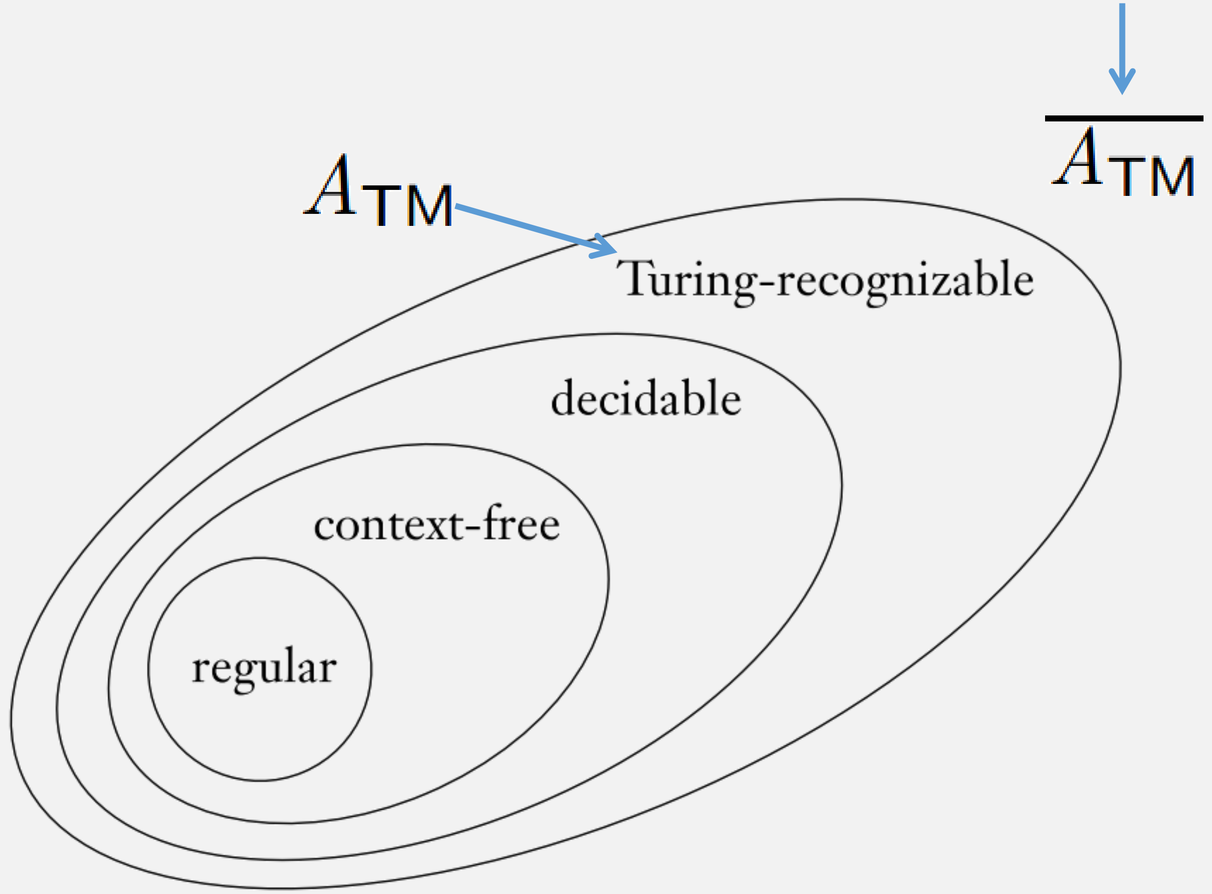
  A_{TM} is undecidable

- So:

  $\overline{A_{\mathsf{TM}}}$ is not Turing-recognizable

- Because: recognizable & co-recognizable implies decidable

Is there anything out here?



Next time: Easier Undecidability Proofs!

- We proved $A_{\mathsf{TM}} = \{\langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$ undecidable by …

- … showing that it's decider could be used to implement an impossible "D" decider.
  - This was hard (need diagonalization)

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$	$\langle D \rangle$
M_1	*accept*	*reject*	*accept*	*reject*		*accept*
M_2	*accept*	*accept*	*accept*	*accept*	$\cdots$	*accept*
M_3	*reject*	*reject*	*reject*	*reject*		*reject*
M_4	*accept*	*accept*	*reject*	*reject*		*accept*
$\vdots$			$\vdots$		$\ddots$	
D	*reject*	*reject*	*accept*	*accept*		?

- In other words, we **reduced** A_{TM} to the "D" problem.

- But now we can just reduce things to A_{TM}: much easier!

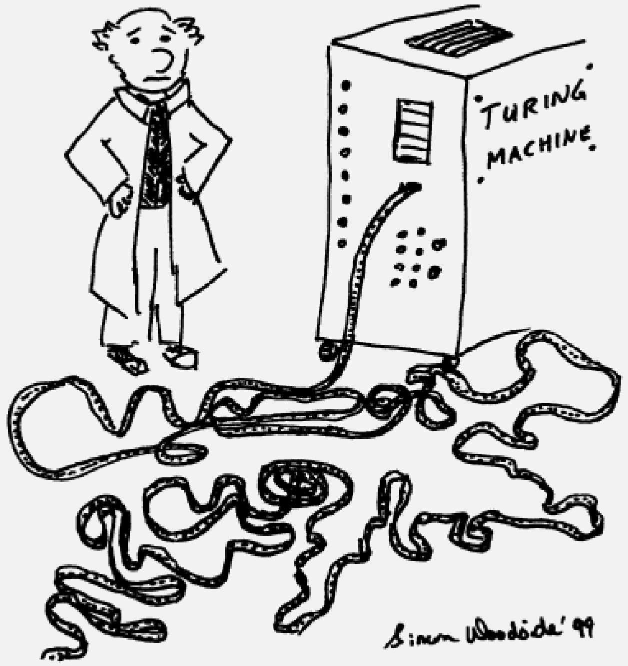
Next time: The Halting Problem

$HALT_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$

Thm: $HALT_{\mathsf{TM}}$ is undecidable

Proof, by contradiction:

- Assume $HALT_{\mathsf{TM}}$ has *decider* R
- Use it to create decider for A_{TM} :
  - …
- But A_{TM} is undecidable!



Turing solves the halting problem, only to discover that the REAL problem with his machine is what to do with all the tape.

Check-in Quiz 3/31

On gradescope