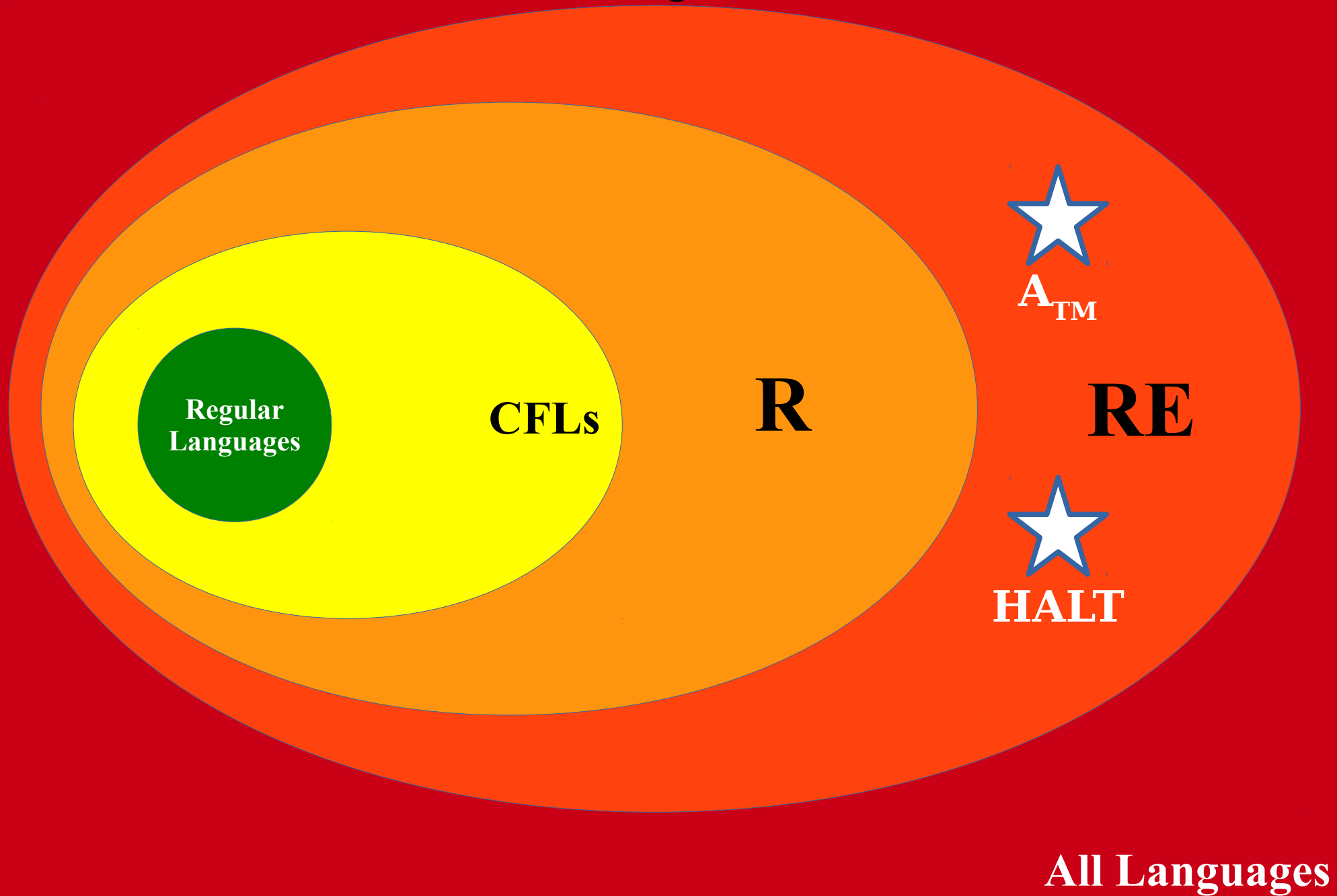


The Class RE

The Story So Far



Unsolvable Problems

Finding Non-**RE** Languages

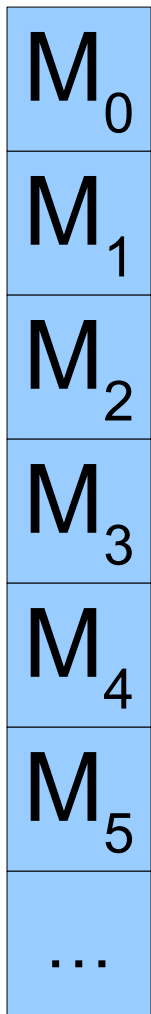
Finding Non-**RE** Languages

- Remember **RE** but non-**R** (undecidable) languages are those where we can reliably identify strings in the language, but cannot readily identify strings that are *not* in the language.
- Non-**RE** languages will be those where we cannot even readily identify strings that *are* in the language!
- How might we find an example of a non-**RE** language?

Languages, TMs, and TM Encodings

- What happens if we list off all Turing machines, looking at how those TMs behave when given other TM codes (as strings, so various $\langle M_x \rangle$ strings) as input?

M_0
M_1
M_2
M_3
M_4
M_5
...



All Turing machines, listed
in some order.

$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----

M_0

M_1

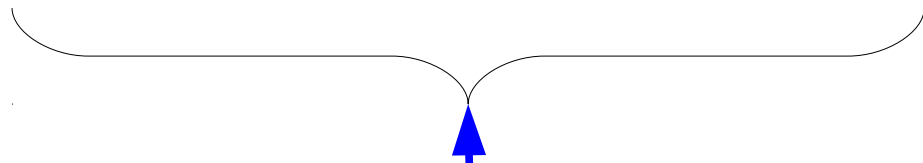
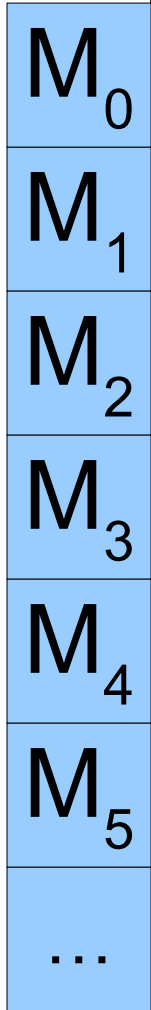
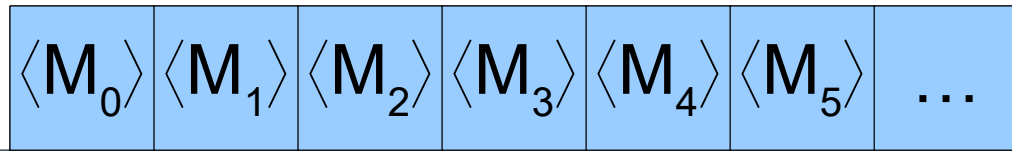
M_2

M_3

M_4

M_5

...



All files/strings of TMs' code, listed in the same order.

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1							
M_2							
M_3							
M_4							
M_5							
...							

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2							
M_3							
M_4							
M_5							
...							

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3							
M_4							
M_5							
...							

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc		
M_3							
M_4							
M_5							
...							

What is $\mathcal{L}(M_0)$?

- A. Σ^*
- B. $\{\langle M_0 \rangle, \langle M_3 \rangle, \langle M_4 \rangle, \dots\}$
- C. $\{\langle M_0 \rangle\}$
- D. $\{\langle M_0 \rangle, \langle M_1 \rangle, \langle M_2 \rangle, \langle M_3 \rangle, \dots\}$
- E. Something else.

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3							
M_4							
M_5							
...							

$\mathcal{L}(M_0) = \{ \langle M_0 \rangle, \langle M_3 \rangle, \langle M_4 \rangle, \dots \}$.
 And we can't see the rest of the table for M_2 , but it accepts everything so far, so it's at least possible that its language is $\mathcal{L}(M_2) = \Sigma^*$.

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc			
M_2	Acc	Acc	Acc	Acc			
M_3							
M_4							
M_5							
...							

Aside: we aren't really worrying about the existence of other strings that aren't TM code right now, but you could also think of it including those strings, so $\mathcal{L}(M_0) = \{\langle M_0 \rangle, \langle M_3 \rangle, \langle M_4 \rangle, \dots\} \cup \{w \mid w \text{ is a string that isn't a TM's code, and } M_0 \text{ accepts } w\}$

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4							
M_5							
...							

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5							
...							

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...							

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...							

Quick check:
 How many of the TMs on this chart so far do NOT accept their own code as a string?
 (Enter a number.)

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

What are we going to do next?

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

Acc	Acc	Acc	No	Acc	No	...
-----	-----	-----	----	-----	----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

Flip all "accept" to "no" and vice-versa

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

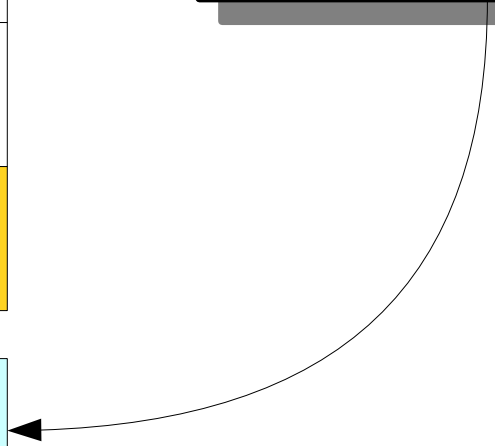
	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

What TM has this behavior?



	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

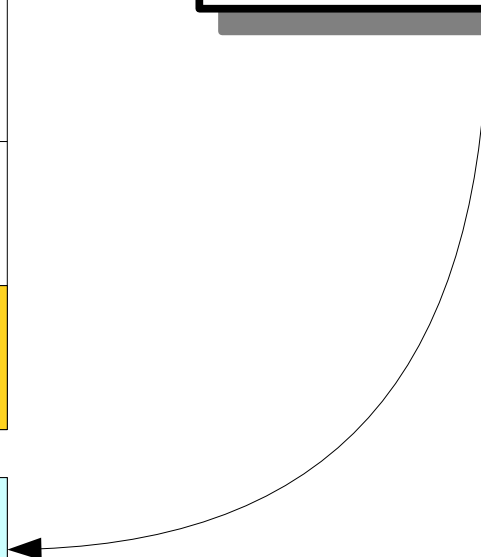
	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

No TM has this behavior!



	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

“The language of all TMs that do not accept their own description.”

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

$\{ \langle M \rangle \mid M \text{ is a TM that does not accept } \langle M \rangle \}$

	$\langle M_0 \rangle$	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$...
M_0	Acc	No	No	Acc	Acc	No	...
M_1	Acc	Acc	Acc	Acc	Acc	Acc	...
M_2	Acc	Acc	Acc	Acc	Acc	Acc	...
M_3	No	Acc	Acc	No	Acc	Acc	...
M_4	Acc	No	Acc	No	Acc	No	...
M_5	No	No	Acc	Acc	No	No	...
...

No	No	No	Acc	No	Acc	...
----	----	----	-----	----	-----	-----

$\{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$

Diagonalization Revisited

- The ***diagonalization language***, which we denote L_D , is defined as

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

- That is, L_D is the set of descriptions of Turing machines that do not accept themselves.

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$.

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

Because $\mathcal{L}(R) = L_D$, we know that a string belongs to one set if and only if it belongs to the other.

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

From the definition of L_D , we see that $\langle M \rangle \in L_D$ iff $\langle M \rangle \notin \mathcal{L}(M)$.

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

From the definition of L_D , we see that $\langle M \rangle \in L_D$ iff $\langle M \rangle \notin \mathcal{L}(M)$. Combining this with statement (1) tells us that

$$\langle M \rangle \notin \mathcal{L}(M) \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (2)$$

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

From the definition of L_D , we see that $\langle M \rangle \in L_D$ iff $\langle M \rangle \notin \mathcal{L}(M)$. Combining this with statement (1) tells us that

$$\langle M \rangle \notin \mathcal{L}(M) \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (2)$$

We've replaced the left-hand side of this biconditional with an equivalent statement.

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

From the definition of L_D , we see that $\langle M \rangle \in L_D$ iff $\langle M \rangle \notin \mathcal{L}(M)$. Combining this with statement (1) tells us that

$$\langle M \rangle \notin \mathcal{L}(M) \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (2)$$

Since our choice of M was arbitrary, we see that statement (2) holds for any TM M .

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

From the definition of L_D , we see that $\langle M \rangle \in L_D$ iff $\langle M \rangle \notin \mathcal{L}(M)$. Combining this with statement (1) tells us that

$$\langle M \rangle \notin \mathcal{L}(M) \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (2)$$

Since our choice of M was arbitrary, we see that statement (2) holds for any TM M .

A nice consequence of a universally-quantified statement is that it should work in all cases.

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

From the definition of L_D , we see that $\langle M \rangle \in L_D$ iff $\langle M \rangle \notin \mathcal{L}(M)$. Combining this with statement (1) tells us that

$$\langle M \rangle \notin \mathcal{L}(M) \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (2)$$

Since our choice of M was arbitrary, we see that statement (2) holds for any TM M . In particular, this means that statement (2) holds for the TM R , which tells us that

$$\langle R \rangle \notin \mathcal{L}(R) \text{ iff } \langle R \rangle \in \mathcal{L}(R). \quad (3)$$

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

From the definition of L_D , we see that $\langle M \rangle \in L_D$ iff $\langle M \rangle \notin \mathcal{L}(M)$. Combining this with statement (1) tells us that

$$\langle M \rangle \notin \mathcal{L}(M) \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (2)$$

Since our choice of M was arbitrary, we see that statement (2) holds for any TM M . In particular, this means that statement (2) holds for the TM R , which tells us that

$$\langle R \rangle \notin \mathcal{L}(R) \text{ iff } \langle R \rangle \in \mathcal{L}(R). \quad (3)$$

This is clearly impossible.

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

From the definition of L_D , we see that $\langle M \rangle \in L_D$ iff $\langle M \rangle \notin \mathcal{L}(M)$. Combining this with statement (1) tells us that

$$\langle M \rangle \notin \mathcal{L}(M) \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (2)$$

Since our choice of M was arbitrary, we see that statement (2) holds for any TM M . In particular, this means that statement (2) holds for the TM R , which tells us that

$$\langle R \rangle \notin \mathcal{L}(R) \text{ iff } \langle R \rangle \in \mathcal{L}(R). \quad (3)$$

This is clearly impossible. We have reached a contradiction, so our assumption must have been wrong.

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

From the definition of L_D , we see that $\langle M \rangle \in L_D$ iff $\langle M \rangle \notin \mathcal{L}(M)$. Combining this with statement (1) tells us that

$$\langle M \rangle \notin \mathcal{L}(M) \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (2)$$

Since our choice of M was arbitrary, we see that statement (2) holds for any TM M . In particular, this means that statement (2) holds for the TM R , which tells us that

$$\langle R \rangle \notin \mathcal{L}(R) \text{ iff } \langle R \rangle \in \mathcal{L}(R). \quad (3)$$

This is clearly impossible. We have reached a contradiction, so our assumption must have been wrong. Thus $L_D \notin \mathbf{RE}$.

$$L_D = \{ \langle M \rangle \mid M \text{ is a TM and } \langle M \rangle \notin \mathcal{L}(M) \}$$

Theorem: $L_D \notin \mathbf{RE}$.

Proof: By contradiction; assume that $L_D \in \mathbf{RE}$. Then there must be some recognizer R such that $\mathcal{L}(R) = L_D$.

Let M be an arbitrary TM. Since $\mathcal{L}(R) = L_D$, we know that

$$\langle M \rangle \in L_D \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (1)$$

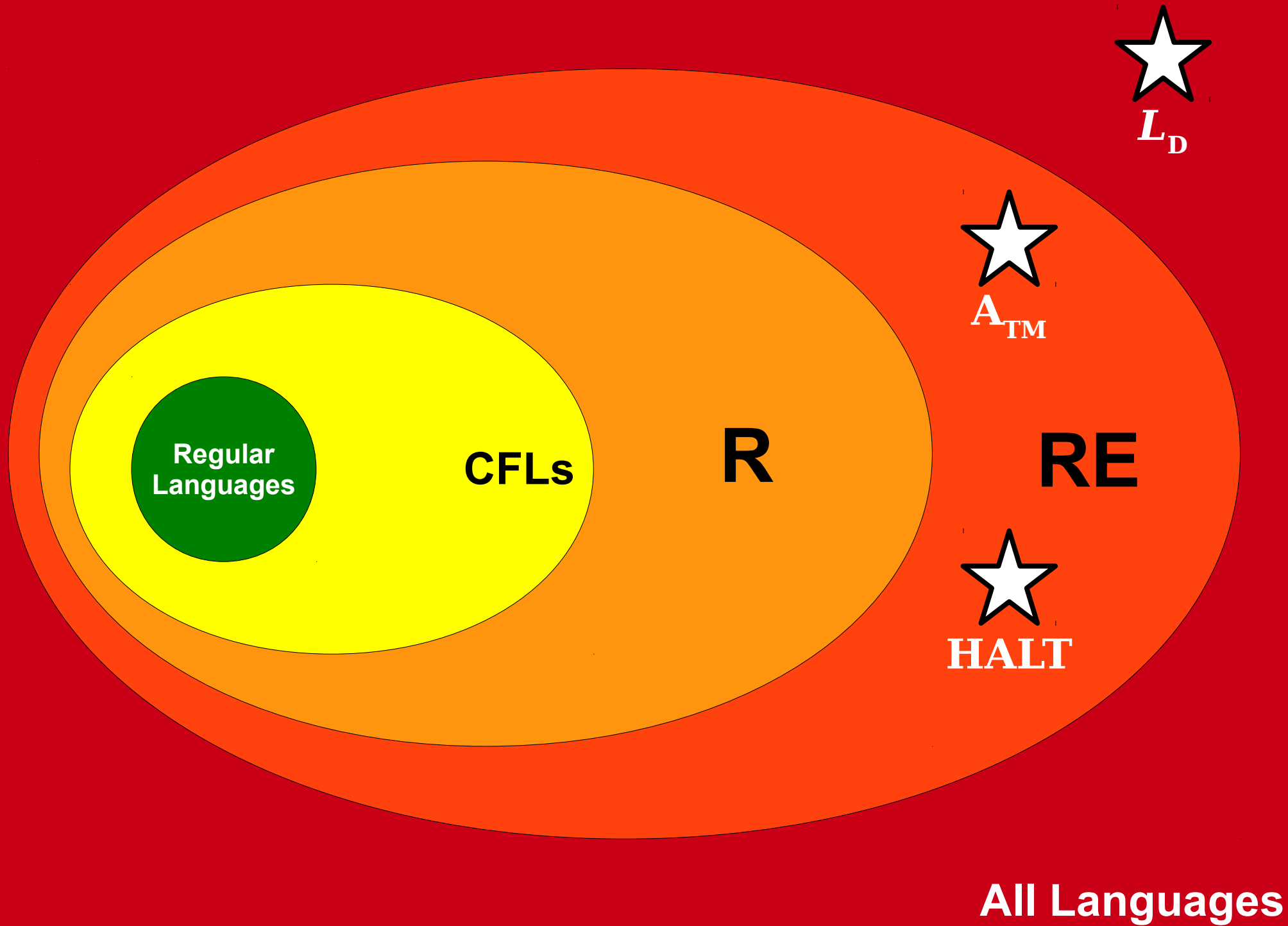
From the definition of L_D , we see that $\langle M \rangle \in L_D$ iff $\langle M \rangle \notin \mathcal{L}(M)$. Combining this with statement (1) tells us that

$$\langle M \rangle \notin \mathcal{L}(M) \text{ iff } \langle M \rangle \in \mathcal{L}(R). \quad (2)$$

Since our choice of M was arbitrary, we see that statement (2) holds for any TM M . In particular, this means that statement (2) holds for the TM R , which tells us that

$$\langle R \rangle \notin \mathcal{L}(R) \text{ iff } \langle R \rangle \in \mathcal{L}(R). \quad (3)$$

This is clearly impossible. We have reached a contradiction, so our assumption must have been wrong. Thus $L_D \notin \mathbf{RE}$. ■



What This Means

- On a deeper philosophical level, the fact that non-**RE** languages exist supports the following claim:

There are statements that are true but not provable.

- Intuitively, given any non-**RE** language, there will be some string in the language that *cannot* be proven to be in the language.
- This result can be formalized as a result called ***Gödel's incompleteness theorem***, one of the most important mathematical results of all time.
- Want to learn more? Take Phil 152 or CS154!

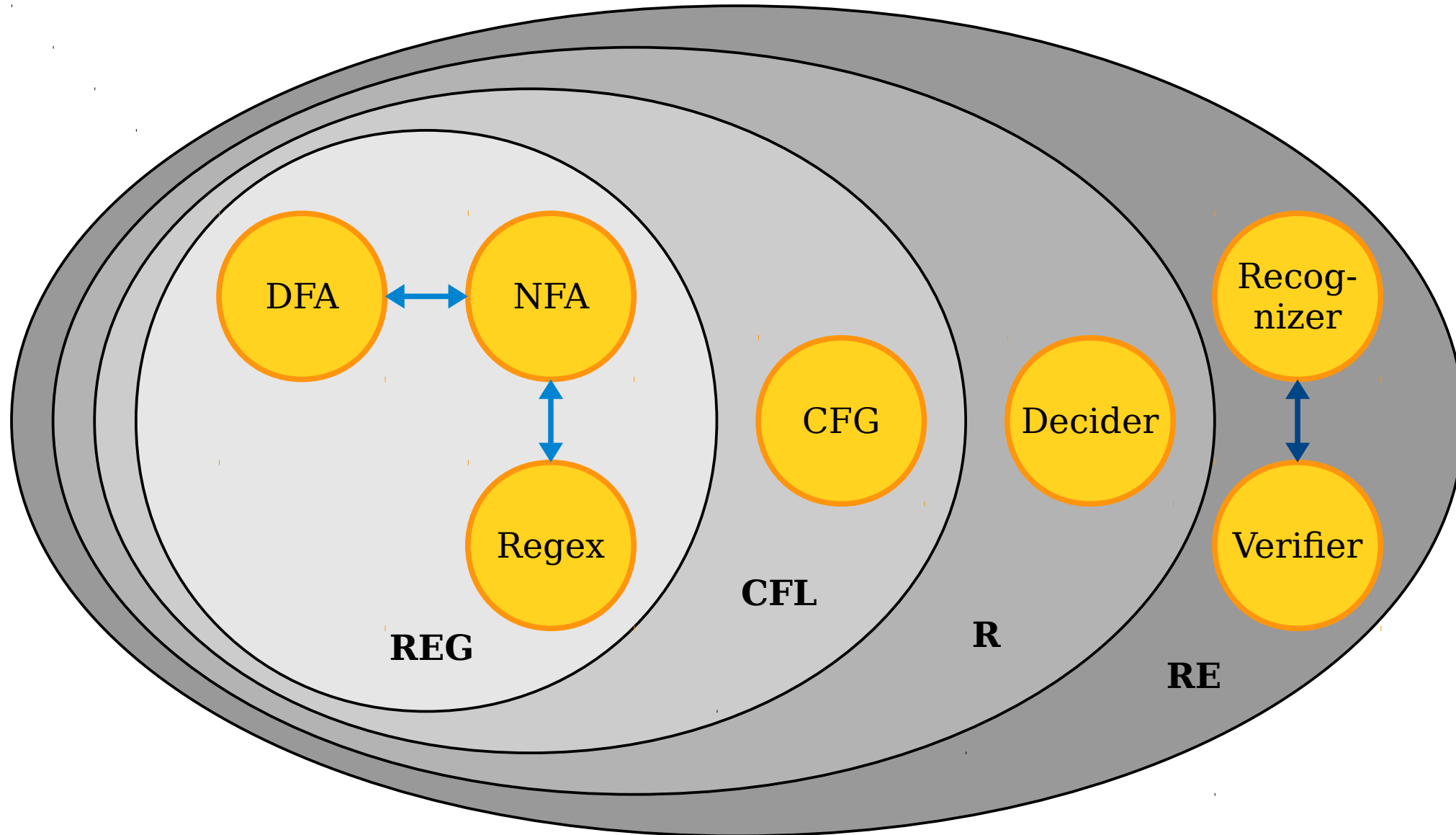
What This Means

- On a more philosophical note, you could interpret the previous result in the following way:

There are inherent limits about what mathematics can teach us.

- There's no automatic way to do math. There are true statements that we can't prove.
- That doesn't mean that mathematics is worthless. It just means that we need to temper our expectations about it.

The Big Picture



Up to this point:

“Can we solve this problem?”

(Computability Theory)

Up to this point:
“*Can we solve this problem?*”
(**Computability Theory**)



Up to this point:

“*Can we solve this problem?*”

(Computability Theory)

Starting today:

“Ok, even if we *can*, we need to consider whether the time/resources required actually make practical/feasible sense.”

(Complexity Theory)

Where We've Been

- The class **R** represents problems that can be solved by a computer.
- The class **RE** represents problems where “yes” answers can be verified by a computer.

Where We're Going

- The class **P** represents problems that can be solved *efficiently* by a computer.
- The class **NP** represents problems where “yes” answers can be verified *efficiently* by a computer.