UNDECIDABILITY

Recall: We saw how TMs can simulate other machines, including other TMs (via a universal Turing Machine)

Today: The Halting Problem is undecidable

$$L_{TM} = \{ \langle M \rangle \# w \mid M \text{ is a TM and } M \text{ accepts } w \}$$

$\langle M \rangle$ is the string description of a TM M.

So a machine which takes a string as input could, in theory, be made to run on its own description!

$$\langle M \rangle \# \langle M \rangle$$

E.g: Bootstrapping compilers

We will use this idea for diagonalization over the set of TMs that take a string as input

Claim: $\mathcal{L}_{TM}$ is undecidable

Proof: We prove this by contradiction. Assume $\mathcal{L}_{TM}$ is decidable.

Then, there is a machine H which decides $\mathcal{L}_{TM}$.

$$H(\langle M \rangle \# \omega) = \begin{cases} Y, & \text{if } M \text{ accepts } \omega \\ N, & \text{if } M \text{ rejects or loops on } \omega \end{cases}$$

We will construct a machine which uses H as a subroutine.

What happens if one runs M on input $\langle M \rangle$?

	$\langle M_1\rangle$	$\langle M_2\rangle$	$\langle M_3\rangle$	$\langle M_4\rangle$	$\langle M_5\rangle$	$\langle M_6\rangle$	$\cdots$
M_1	Y	–	N	Y	Y	Y	$\cdots$
M_2	–	N	N	Y	Y	Y	$\cdots$
M_3	Y	Y	–	–	–	N	$\cdots$
M_4	N	Y	Y	N	N	Y	$\cdots$
M_5	–	–	–	N	Y	N	$\cdots$
M_6	N	N	Y	–	–	–	$\cdots$
$\vdots$							

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$	$\langle M_6 \rangle$	. . . . . .
M_1	Y	N	N	Y	Y	Y	. . . . .
M_2	N	N	N	Y	Y	Y	. . . . . .
M_3	Y	Y	N	N	N	N	. . . . .
M_4	N	Y	Y	N	N	Y	. . . . .
M_5	N	N	N	N	Y	N	. . . . .
M_6	N	N	Y	N	N	N	. . . . . .
	. . . .						

$\mathcal{H}$

We will construct the following TM D.

D takes as input the string description of a TM M, and does the opposite of whatever H would do.

$$D(\langle M \rangle) = \begin{cases} Y, & \text{if } M \text{ does not accept } \langle M \rangle \\ N, & \text{if } M \text{ accepts } \langle M \rangle \end{cases}$$

D accepts $\langle M \rangle$ exactly when M does not accept $\langle M \rangle$.

D will operate on the string description of any TM.

What about on its own string description?

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$	$\langle M_6 \rangle$	$\cdots$
M_1	N	Y	Y	N	N	N	$\cdots$
M_2	Y	Y	Y	N	N	N	$\cdots$
M_3	N	N	Y	Y	Y	Y	$\cdots$
M_4	Y	N	N	Y	Y	N	$\cdots$
M_5	Y	Y	Y	Y	N	Y	$\cdots$
M_6	Y	Y	N	Y	Y	Y	$\cdots$
$\vdots$							

D

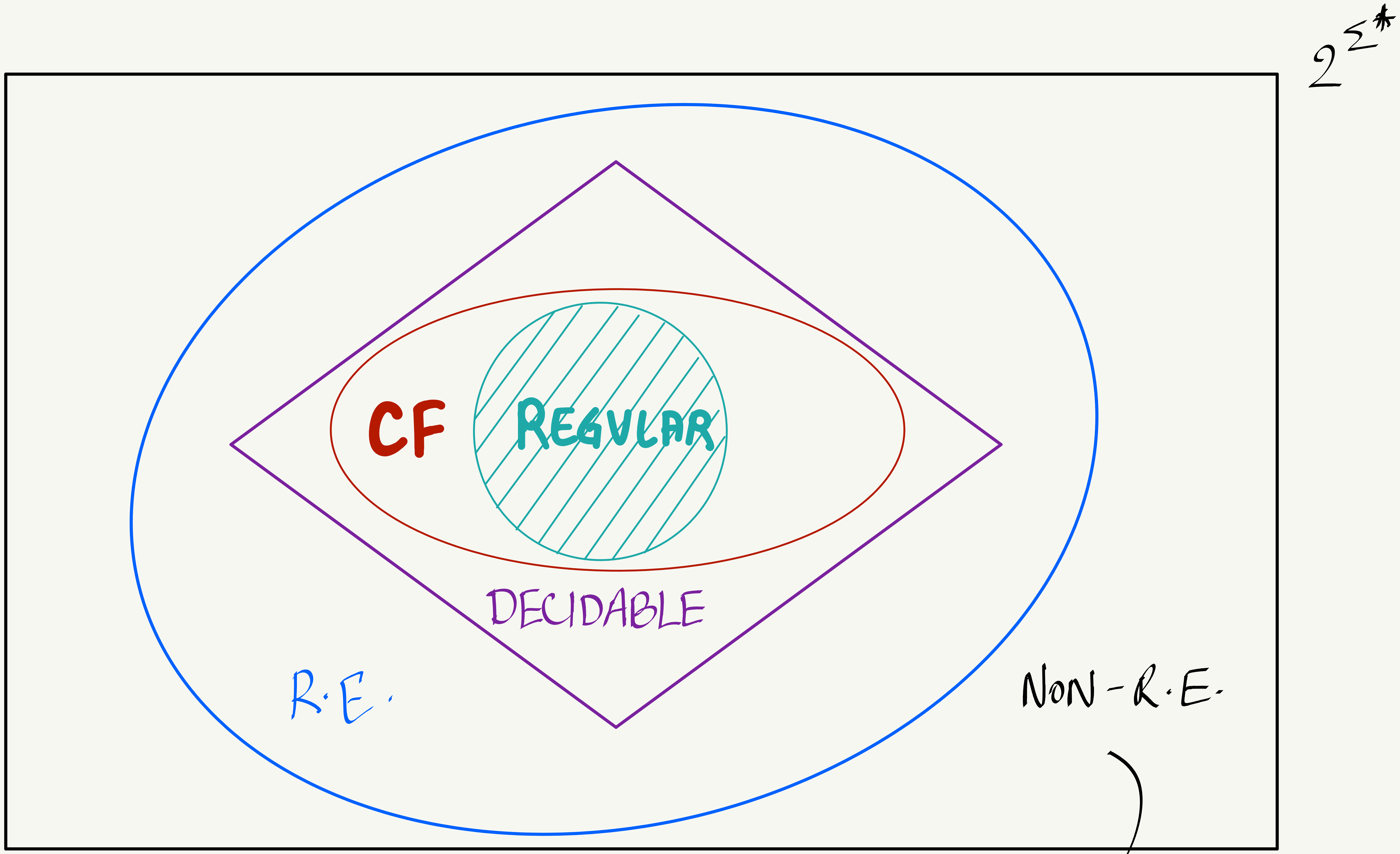
	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\langle M_5 \rangle$	$\langle M_6 \rangle$	$\cdots$	$\langle D \rangle$	
M_1	N	Y	Y	N	N	N	$\cdots$		
M_2	Y	Y	Y	N	N	N	$\cdots$		
M_3	N	N	Y	Y	Y	Y	$\cdots$		
M_4	Y	N	N	Y	Y	N	$\cdots$		
M_5	Y	Y	Y	Y	N	Y	$\cdots$		
M_6	Y	Y	N	Y	Y	Y	$\cdots$		
	$\vdots$								
D								?	

$$D(\langle D \rangle) = \begin{cases} Y, & \text{if } D \text{ does not accept } \langle D \rangle \\ N, & \text{if } D \text{ accepts } \langle D \rangle \end{cases}$$

This is an obvious contradiction,
so our assumption that H decides $\mathcal{L}_{TM}$ must be false!

So,

$\mathcal{L}_{TM} = \{ M\#\omega \mid M \text{ is a TM and } M \text{ accepts } \omega \}$
is undecidable.



How do we know that there is any language in this set at all?

$\mathcal{L}_{TM}$ is not decidable, but

$\mathcal{L}_{TM}$ is Turing-recognizable.

<u>Claim</u>: $\overline{\mathcal{L}_{TM}}$ is not Turing-recognizable

<u>Proof</u>: By contradiction.