W9L1 thinking about undecidability

Monday, March 30, 2020 9:01

FOR CLARITY,
Consider a language L.

(1) Is L decidable?

(2) How would you design a TM to decide L?
Design it. Give the algorithm.

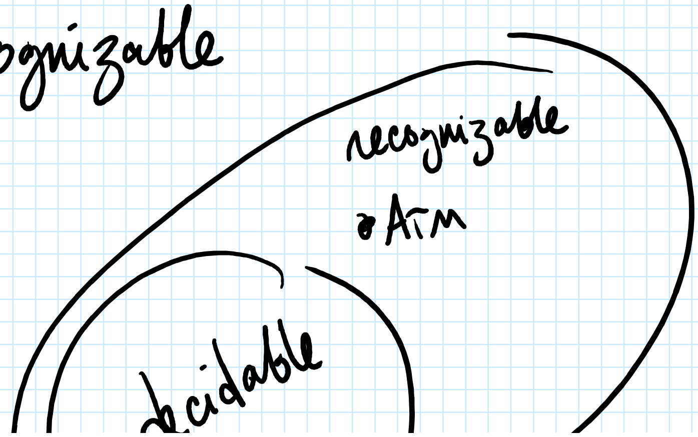
These questions are different!

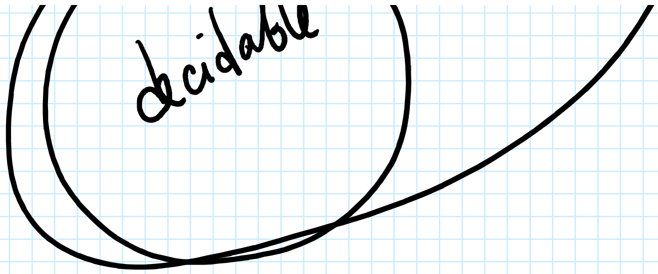
If you can give a TM in Q2, then Q1 is YES.
Sometimes we can prove Q1 is YES
even though we don't have enough info to answer Q2.

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

A_{TM} is <u>not</u> decidable

A_{TM} is recognizable





If M is a TM, define $L(M) = \{ w \mid M \text{ accepts } w \}$.
This is the lang recognized by M.

---

list of all TMs	all Turing-recognizable langs.	all co-recognizable langs
$\langle M_1 \rangle$	$L(M_1)$	$\overline{L(M_1)}$
$\langle M_2 \rangle$	$L(M_2)$	
$\langle M_3 \rangle$	$L(M_3)$	$\overline{L(M_3)}$
$\vdots$		
countable ✓	countable (might contain duplicates)	also countable ✓

<u>Claim</u>: $\overline{A_{TM}}$ is not decidable.

<u>Proof</u>: (by contradiction)
S'pose M decides $\overline{A_{TM}}$.

N = "On input $\langle R, w\rangle$:

1. Run M on $\langle R, w\rangle$.
2. Do the opposite."

N is a decider for A_{TM}.

$\Rightarrow\Leftarrow$ We know A_{TM} is undecidable!