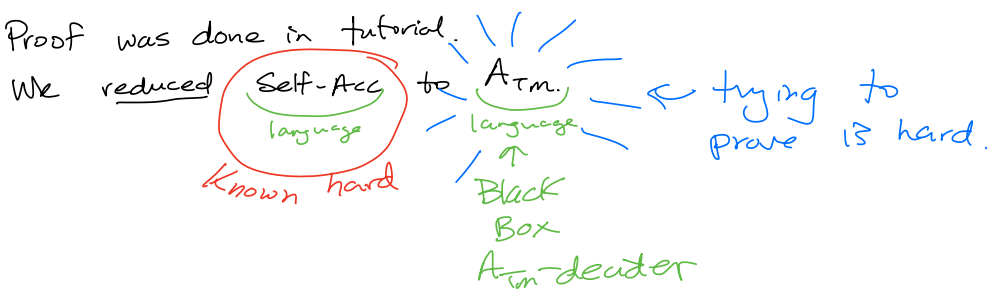


Thm 4.11 A_{TM} is undecidable.

Proof was done in tutorial.



Allows us to construct a Self-Acc decider that calls the Black-Box A_{TM} decider as a subroutine

Algorithm reductions are common.

Eg. To tell if a graph is Connected

- Call Shortest Paths
- check that every node has non- ∞ path length to every other node.

reduces Connectivity to shortest paths.

$$E_{TM} = \{ \langle M \rangle \mid M \text{ is a TM that accepts no strings} \}$$

Theorem: E_{TM} is undecidable.

Proof: We reduce A_{TM} to E_{TM} .

I.e. we assume (BWOC) that E_{TM} is decided by a TM, call it X (decides E_{TM}).

We construct a TM Y as follows: [idea: Y decides A_{TM}]

- $Y =$ "on input $\langle M, w \rangle$ where M is a TM, w a string:
- A_{TM} decider →
1. Construct a new TM, M_w , that works as follows:
 - M_w 1. Erase the input, and write w on the tape.
 - M_w 2. Run M on the input tape contents, w .

2. Run X on $\langle M_w \rangle$
3. If X accepts, REJECT.
If X rejects, ACCEPT

What does M_w do on all inputs if M accepts w ?

- M_w accepts all inputs hence $L(M_w)$ is not empty and X rejects. (+ vice versa).

∴ X is a decider for A_{TM} . $\Rightarrow \Leftarrow$.

∴ E_{TM} is undecidable. $\square$

" M "
 $\text{square}(x)$
 $\text{return}(x * x);$

" M_w " ($w=10$).
 $\text{square10}(x)$
 $\text{return square}(10);$

$$EQ_{TM} = \{ \langle M_1, M_2 \rangle \mid M_1, M_2 \text{ are TMs, } L(M_1) = L(M_2) \}$$

Theorem: EQ_{TM} is undecidable

Proof: We reduce the decidability of EQ_{TM} to the decidability of E_{TM}

Suppose $\exists$ a decider W for

We construct a decider T for as follows:

T = "on input $\langle M \rangle$, where M is a TM:

1. Construct
- 2.
- 3.

Since W is a decider for EQ_{TM} , then clearly T

- always halts
- accepts $\langle M \rangle$ iff M is equivalent to $M_\emptyset$ i.e. $L(M) = \emptyset$.

◦ T decides E_{TM}

$\Rightarrow \Leftarrow$

◦ W cannot exist, and EQ_{TM} is undecidable. $\square$

Let $NotA_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM that does not accept } w \}$

Note: $NotA_{TM} = \overline{A_{TM}} \cap \{ \langle M, w \rangle \mid M \text{ is a TM, } w \text{ is a string} \}$.

It is common to just refer to $NotA_{TM}$ as $\overline{A_{TM}}$, but is technically not correct.

Theorem 4.23 $NotA_{TM}$ is not recognizable

Proof: BWOC. Suppose $NotA_{TM}$ is recognized by some TM $\bar{A}$.

This tells us what the hard part of A_{TM} is ...
(though we already knew)

A_{TM} - YES! (ACCEPT) answers — we get those
 A_{TM} - NO! (REJECT or loop) answers — we don't get them all.