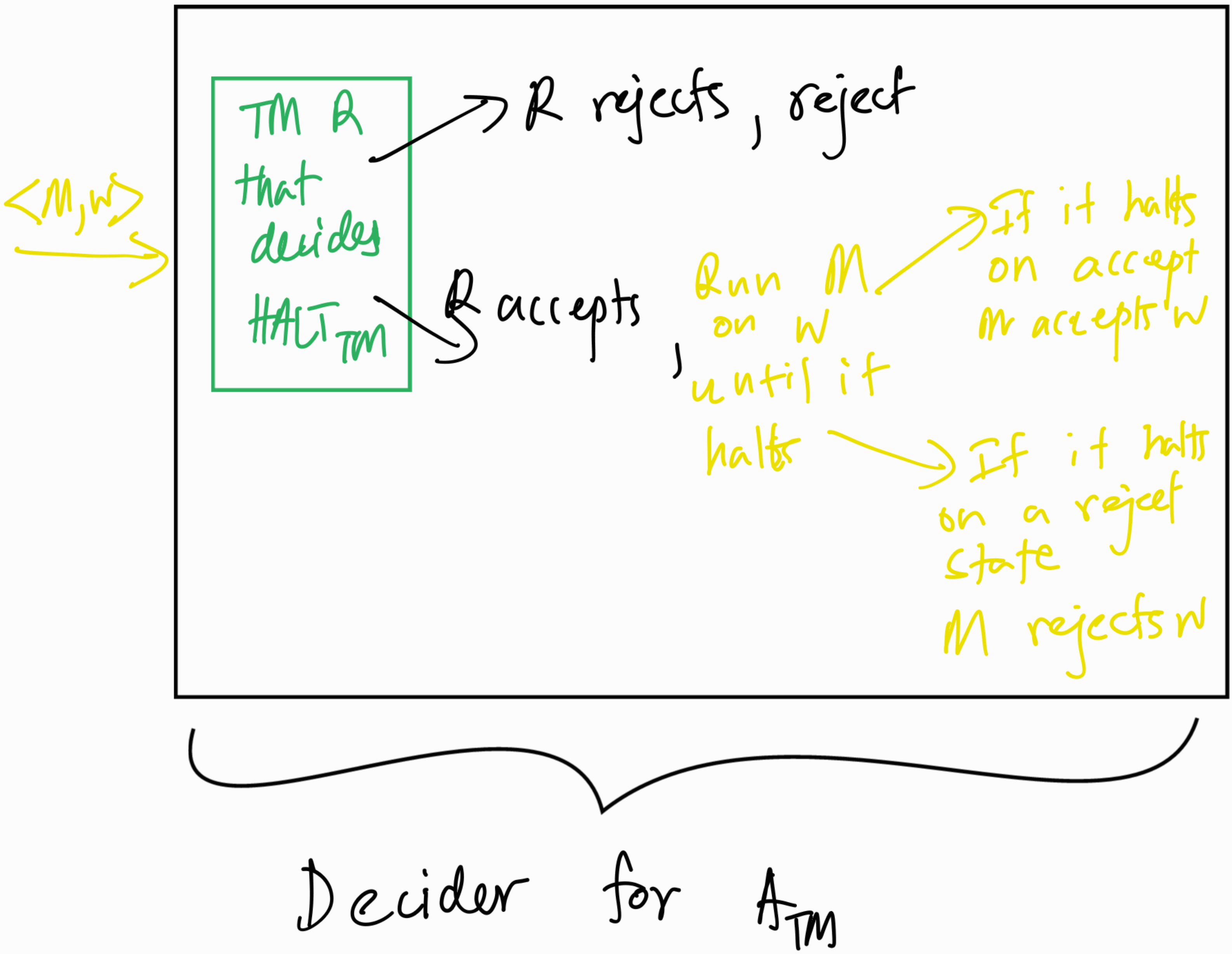
Recap

In the last lecture we proved that $HALT_{TM}$ is undecidable using reduction.

Recap:

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



Let's write the proper proof!

Theorem 5.1

$HALT_{TM}$ is undecidable

Proof Idea: we will construct a decider for A_{TM} (which we know that is undecidable) using a decider for $HALT_{TM}$.

Assume $HALT_{TM}$ is decidable, and the turing machine R is that decider.

Let us construct Turing machine S that decides A_{TM}.

S = on input $\langle M, w \rangle$ where M is a TM and w is a string

1. Run TM R on $\langle M, w \rangle$
2. If R rejects $\langle M, w \rangle$ reject
3. If R accepts, simulate M on w until it halts.
4. If M halts on an accept state, accept
   If M halts on a reject state, reject.

Clearly TM S is a decider for A_{TM}. But A_{TM} is undecidable. Therefore, our initial assumption must be false.

$HALT_{TM}$ is undecidable.

$E_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset\}$

Theorem 5.2

E_{TM} is undecidable.

E_{TM} is basically the problem where we check a TM accepts nothing.

I am going to reduce A_{TM} to E_{TM}

$$A_{TM} \leq E_{TM}$$

$\langle M, w \rangle$ $\langle M, \rangle$

How can I do this?

Assume E_{TM} is decidable and TM R decides E_{TM}.

Idea: Run R on $\langle M \rangle$ and check whether $L(M) = \phi$, then reject.
But if $L(M) \neq \phi$, then I can not conclude whether M accept w or not.

Let's create the decider for A_{TM} which is the TM S.

M_1 = on input $\langle x \rangle$, where $\langle M, w \rangle$ is hard coded inside this

1. If $x \neq w$, reject

2. If $x = w$,
   run M on w and accept x if M accepts w.

M accepts $w \Rightarrow L(M_1) = \{w\}$

M does not accept $w \Rightarrow L(M_1) = \phi$

$S =$ on input $\langle M, w \rangle$, where M is a TM and w is a string

1. construct special TM M_1 that encodes $\langle M, w \rangle$ inside it.

2. Run R on $\langle M_1 \rangle$

3. If R accepts, reject
   If R rejects, accept

Note that S is a decider for A_{TM}, but A_{TM} is undecidable.

$\therefore$ E_{TM} is undecidable.

$FINITE_{TM} = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is finite} \}$

$A_{TM} \leq FINITE_{TM}$

$\langle M, w \rangle \qquad \langle M \rangle$

Assume $FINITE_{TM}$ is decidable, and TM R decides it.

Now we construct TM S that is a decider for A_{TM}.

M_1 = on input $\langle x \rangle$, havd code $\langle M, w \rangle$ inside it

1. Run M on w, accept if M accepts w.

M accept $w \Rightarrow L(M_1) = \Sigma^*$

M does not accept $w \Rightarrow L(M_1) = \emptyset$

S = on input <M, w>, where M is a TM and w is a string

1. construct a special TM M_1 that encodes <M, w> in its construction.
2. Run R on $\langle M_1 \rangle$
3. If R accepts, reject
4. If R rejects, accept

S is a decider for A_{TM},
but A_{TM} is undecidable.
$\therefore$ our initial assumption is incorrect.
$\therefore$ $FINITE_{TM}$ is undecidable.

$2\text{-DIFF-TM} = \{ \langle M \rangle \mid M$ is a TM and $L(M)$ contains two different strings of same length $\}$

$$A_{TM} \leq 2\text{-DIFF-TM}$$

$$\langle M, w \rangle \qquad \langle M \rangle$$

construct special TM M_1 that encodes $\langle M, w \rangle$

M_1 = " on input $\langle x \rangle$

1. If $x = aa$ or $x = bb$
   Run M on w and accept if M accepts w.
2. Else
   reject.

M accepts $w \Rightarrow L(M_1) = \{aa, bb\}$

M does not accept $w \Rightarrow L(M_1) = \emptyset$

Assume 2-DIFF-TM is decidable and TM R decides it.

Then construct TM S that decides A_{TM} as follows:

1. construct M_1
2. Run R on $\langle M_1 \rangle$
3. If R accepts; accepts

   If R rejects, reject.

clearly S is a decider for A_{TM}.
contradiction.