End of last lecture: \( \text{EQ}_{\text{TM}} \) is undecidable

**P.F.** Let \( \text{DEQ} \) be a decider for \( \text{EQ}_{\text{TM}} = \{<M_1, M_2> | L(M_1) = L(CA_M)\} \).

Construct a decider \( \text{DA} \) for \( A_{\text{TM}} \):

On input \(<M, w>\):

\[
f(<M, w>) = L(M_1, M_2) \quad \text{where}
\]

\[
f(<M, w>) = L(M_1, M_2) \quad \text{where}
\]

- \( M_1 \) always accepts
- \( M_2 \) runs on \( M \) on \( w \), accepts if \( M \) accepts

Run \( \text{DEQ} \) on input \(<M_1, M_2>\), do as \( \text{DEQ} \) does

**Idea:** If \( M \) accept \( w \), \( M_1 \) and \( M_2 \) are equivalent.

**Analysis:**

- If \(<M, w> \notin A_{\text{TM}} \), \( M_1 \) and \( M_2 \) are equivalent, so \( \text{DEQ} \) accepts \(<M_1, M_2>\), so \( \text{DA} \) accepts.

- If \(<M, w> \in A_{\text{TM}} \), \( M_1 \) and \( M_2 \) aren't equivalent, so \( \text{DEQ} \) rejects \(<M_1, M_2>\), so \( \text{DA} \) rejects.

**Def:** (5.17 in Sipser): A function \( f: \Sigma^* \rightarrow \Sigma^* \) is computable if there exists a TM \( M \) that, on any input \( x \in \Sigma^* \), halts with \( x \) on its tape.

**Def:** (5.20): Let \( A, B \) be languages. We say \( A \) is mapping-reducible to \( B \), or \( A \leq m B \), if there exists some computable function \( f \) s.t. \( \forall x, x \in A \iff f(x) \in B \).

In our example, we have \( A_{\text{TM}} \leq m \text{EQ}_{\text{TM}} \).
pf of claim: Given f:

1. On input <M, w>,
   Output <M', M_2>
   where M' = "On input a, accept"
   M_2 = "On input a, Run M on w. If it accepts, accept. Else, reject."

Analysis: On previous page.

Thm (5.22): If A \leq^m B and B is decidable, then so is A.

pf: Let M_B be a decider for B.

Let M_A be as follows: function s.t. x \in A if f(x) \in B
M_A = "On input x, Compute y = f(x) \(\text{using TM } F_n \text{ that must exist because } A \leq^m B \)."

Run M_B on y.
Accept if M accepts, reject otherwise.

We claim M_A is a decider for A.
Suppose x \in A. Then y = f(x) \in B by the definition of a mapping reduction. M_B accepts y, so M_A accepts x.
Suppose x \notin A. Then y = f(x) \notin B since A \leq^m B.
Thus, M_B rejects y and so M_A rejects x.

(5.23) Corollary: If A \leq^m B and A is undecidable, then so is B.

pf: Suppose B were decidable. Then by the theorem, A is decidable. Contradiction.

Q: What if A \leq^m B and B is undecidable?
A: We can't say anything about A.

Consider the language Decider_{TM} = \{ <M> | TM M is a decider \}.
Want to show: A_{TM} \leq^m Decider_{TM}
Here is a computable function $f$:

"On input $<M, w>$
Output $<M', w'>$ where
$M' = " On input $x$, 
. Run $M$ on $w$, else if it accepts
   Enter an infinite loop otherwise"

To compute $f$, we modify $M'$: immediately write $w$ on the tape, change the reject state.

$\Rightarrow f$ is computable.

Correctness: Given $<M, w> \in \mathbb{A}_m$, $M'$ always accepts,
so $<M'> \in \mathbb{D}_m$.
Given $<M, w> \notin \mathbb{A}_m$, $M'$ always loops, so $<M'> \notin \mathbb{D}_m$.

By the corollary, $\mathbb{D}_m$ is undecidable.

Note: $A \equiv_m B$ does not imply $B \equiv_m A$.

$A_m \equiv_m \mathbb{A}_m$:

"On input $<M, w>$
Output $<M', w'>$ where
$M' = " On input $x$, 
. Run $M$ on $w$, if it accepts accept
   Else, loop forever"

$\mathbb{H}_m \equiv_m A_m$:

"On input $<M, w>$
Output $<M', w'>$ where
$M' = " On input $x$, 
. Run $M$ on $w$, if it halts accept"

w' = w"
Let's try to relax the conditions on the theorem.

**Thm (5.29):** If $A \subseteq B$ and $B$ is Turing-recognizable, then so is $A$.

**f.** Let $M_3$ be a TM s.t. $L(M_3) = B$.

Let $M_4$ be as follows:

- $M_4$: on input $x$,
  - compute $f(x) = y$.
  - run $M_3$ on input $y$.
  - accept if it accepts,
  - reject otherwise.

$M_4$ recognizes $A$: $x \in A \iff y \in B$, since $f$ is a mapping reduction.

Then $M_4$ accepts $y$, so $M_4$ accepts $y \in B$, since $f$ is a mapping reduction.

Then $M_3$ either rejects, so $M_4$ rejects $x$, or loops, so $M_4$ loops and does not accept $x$.

**Corollary (5.29):** If $A \subseteq B$ and $A$ is not Turing-recognizable, then no TM is $B$.

**f.** Suppose $B$ were recognizable. Then by theorem, so is $A$. Contradiction.

**Claim:** $A_{TM}^{c} \leq_{T} \text{Decider}_{TM}$.

**f.** Reduction $f$: on input $(M, w)$ where $M' = "$ on input $x$,

- interpret $x$ as an integer $k$,
- run $M'$ on $w$ for $k$ steps,
- if it accepts, loop forever,
- else accept.

**Analysis:** $f$ is computable: Think of $M'$ as a multi-tape TM.

**Correctness:** If $(M, w) \geq A_{TM}^{c}$, $M'$ accepts on all inputs.

Thus, $M'$ is a decider, so $(M, w) \geq \text{Decider}_{TM}$.

If $(M, w) \geq A_{TM}^{c}$, $L(M) \geq e_{A_{TM}}$.

Thus, $M$ accepts on some finite number of steps.

By the corollary, $\text{Decider}_{TM}$ is not Turing-recognizable.
Claim: $A_{tm} \leq_m \text{EQ}_{tm}$. 

 Pf: Reduction $F$:

 On input $\langle M, w \rangle$, output $\langle M_1, M_2 \rangle$ where

 $M_1$: "On input $x$, reject"

 $M_2$: "On input $x$,

 Run $M$ on $x$, do as $M$ does."

 Analysis: $F$ is computable, since we've seen things like it before.

 Correctness: $\langle M, w \rangle \in A_{tm}$ $\iff$ $L(M) = \emptyset = L(M_1)$ $\iff$ $\langle M_1, M_2 \rangle \in \text{EQ}_{tm}$ $\iff$ $\langle M, w \rangle \notin A_{tm}$ $\iff$ $L(M) \neq \emptyset = L(M_2)$ $\iff$ $\langle M_1, M_2 \rangle \notin \text{EQ}_{tm}$ $\iff$ $E_{Q_{tm}}$ is not Turing recognizable.

 Final Thm: If $A \leq_m B$, $A^c \leq_m B^c$.

 Pf: Use the same $F$:

 $A_{tm} \leq_m \text{Decider}_{tm}$ $\iff$ $A_{tm} \leq_m \text{Decider}_{tm}^c$ $\iff$ $\text{Decider}_{tm}$ not Turing recognizable.

 Same for $E_{Q_{tm}}$. 