\[ E_{T_m} = \{ <M> \mid M \text{ is a TM and } L(M) = \emptyset \}. \]

The \( E_{T_m} \) is undecidable.

Given \( <M> \) and \( w \), define \( M_w \) to be a TM that behaves as follows:

- On input \( x \):
  - If \( x \neq w \) then reject
  - If \( x = w \) then simulate \( M \) on \( w \). \( M_w \) accepts \( w \) iff \( M \) accepts \( w \).

Observe that \( M_w \) is easy to construct, given \( <M> \) and \( w \).

Observe that

\[(<M>, w) \in \overline{A}_{T_m} \iff M \text{ does not accept } w \iff (M_w) \in E_{T_m}.\]

Suppose \( A \) is an alg. that decides \( E_{T_m} \). Then decide whether

\[(<M>, w) \in A_{T_m} \text{ as follows:} \]

- Input \( (M_w) \) to \( A \). If \( A \) answers that \( L(M_w) \) is empty, then \( M_w \) does not accept \( w \), so \( M \) does not accept \( w \), so \( (<M>, w) \in \overline{A}_{T_m}. \)

If \( A \) answers that \( L(M_w) \neq \emptyset \), then \( M \) accepts \( w \).

i.e., existence of \( A \) implies that \( A_{T_m} \) is decidable.

So, \( E_{T_m} \) is undecidable.
Today we showed that the following problem is undecidable:

instance. Turing machine $M$
question. Does $M$ accept $(M)$?

Then, we showed the following is undecidable:

instance. Turing machine $M$
and input word $x$
question. Does $M$ accept $x$?

Today we show the following is undecidable:

instance. Turing machine $M$
question. $L(M) = \Sigma^*$
10/24/2008

\[ P \vdash M \]

A mapping reducible to B if there is a computable \( f \) such that for all \( x \):

\[ x \in A \iff f(x) \in B \]

\[ A \leq_m B \]

**Theorem 5.22**

If \( A \leq_m B \) and \( B \) decidable, then \( A \) is decidable.

**Corollary 5.23**

If \( A \leq_m B \) and \( A \) undecidable, then \( B \) is undecidable.

**Application:** We show \( k \leq_m \text{HALT} \)

Therefore, \( \text{HALT} \) is undecidable.
\[ \text{All}_{\text{TM}} = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) = \Sigma^* \} \]

**Theorem.** \( k \leq_m \text{All}_{\text{TM}} \)

**Corollary.** \( \text{All}_{\text{TM}} \) is undecidable.

**Proof of Theorem.**

Want \( \langle M \rangle \in k \iff f(<M>) \in \text{All}_{\text{TM}} \)

For each \( \text{TM} M \), we define a \( \text{TM} D_M \) s.t.
\( \langle M \rangle \in k \iff L(D_M) = \Sigma^* \). Then take

\[ f(<M>) = <D_M > \text{ to obtain } \]
\[ \langle M \rangle \in k \iff L(D_M) = \Sigma^* \]
\[ \iff <D_M > \in \text{All}_{\text{TM}} \]
\[ \iff f(<M>) \in \text{All}_{\text{TM}}, \text{ and we're done.} \]

**The construction:***

Design \( D_M \) to behave as follows: input \( x \) to \( D_M \), where \( x \) is a word in \( M \)'s input alphabet; simulate \( M \) on \( <M> \);

if \( M \) accepts \( <M> \), then accept \( x \)

(Note that \( D_M \) is oblivious)
Observe that \( L(D_m) = \Sigma^* \) or
\( L(D_m) = \emptyset \), and
\( \langle M \rangle \in \mathcal{K} \Leftrightarrow L(D_m) = \Sigma^* \).

\[ f(\langle M \rangle) = \langle D_m \rangle. \]
\( \langle M \rangle \in \mathcal{K} \Leftrightarrow \langle D_m \rangle \in \text{ALL}_{TM} \)
Let $M_{\Sigma^*}$ be the TM that accepts every input word. So $L(M_{\Sigma^*}) = \Sigma^*$.

Define

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

Then observe that

$$\langle M \rangle \in \text{All}_{TM} \iff (\langle M \rangle, \langle M_{\Sigma^*} \rangle) \in EQ_{TM}.$$ 

Define computable function $f$ by

$$f(\langle M \rangle) = (\langle M \rangle, \langle M_{\Sigma^*} \rangle).$$

Then $\langle M \rangle \in \text{All}_{TM} \iff f(\langle M \rangle) \in EQ_{TM}.$

That is, $\text{All}_{TM} \leq_m EQ_{TM}$.

Therefore, since $\text{All}_{TM}$ is undecidable, it follows that $EQ_{TM}$ is undecidable.