Mapping Reducibility
$E_{TM} = \{ \langle M \rangle \mid M \text{ is a TM} \& \mathcal{L}(M) = \emptyset \}$

**Theorem.** $E_{TM}$ is undecidable.

**Proof.** Given an input $\langle M, w \rangle$ we construct a machine $M_w$ that accepts a nonempty language iff $M$ accepts $w$:

$M_w = \text{“On input } x$: 
1. If $x \neq w$, reject.
2. If $x = w$, run $M$ on input $w$ and accept if $M$ does.”

Suppose TM $R$ decides $E_{TM}$ and establish a contradiction by creating a decider $S$ of $A_{TM}$:

$S = \text{“On input } \langle M, w \rangle$: 
1. Use the description of $M$ and $w$ to construct $M_w$.
2. Run $R$ on input $\langle M_w \rangle$.
3. If $R$ accepts, reject. If $R$ rejects, accept.”
Computable Functions
Computable Functions

**Definition.** A function $f : \Sigma^* \rightarrow \Sigma^*$ is a *computable function* if some Turing machine $M$, on every input $w$, halts with just $f(w)$ on its tape.

**Example.** The *increment function*

$$inc^+ : \{1\}^* \rightarrow \{1\}^*$$

is Turing computable.
Machine Transformers

\[ F = \text{“On input } < M > \text{:} \]

1. Construct the machine

\[ M_\infty = \text{“On input } x \text{:} \]
   1. Run \( M \) on \( x \).
   2. If \( M \) accepts, \textit{accept}.
   3. If \( M \) rejects, loop.

2. Output \( < M_\infty > \)."
Mapping Reducibility

**Definition.** Language $A$ is *mapping reducible* to language $B$, written $A \leq_m B$, if there is a computable function $f : \Sigma^* \rightarrow \Sigma^*$, where for every $w$,

$$w \in A \iff f(w) \in B.$$
Problem Reduction

**Theorem.** If \( A \leq_m B \) and \( B \) is decidable, then \( A \) is decidable.
The Contrapositive is Also Useful

Theorem. If $A \leq_m B$ and $B$ is decidable, then $A$ is decidable.

Corollary. If $A \leq_m B$ and $A$ is undecidable, then $B$ is undecidable.
Similarly ...

**Theorem.** If \( A \leq_m B \) and \( B \) is Turing-recognizable, then \( A \) is Turing-recognizable.

**Corollary.** If \( A \leq_m B \) and \( A \) is not Turing-recognizable, then \( B \) is not Turing-recognizable.
A Familiar Mapping Reduction

\[ A_{TM} = \{ <M, w> \mid M \text{ is a TM and } M \text{ accepts } w \} \]

\[ \leq_{m} \]

\[ HALT_{TM} = \{ <M, w> \mid M \text{ is a TM and } M \text{ halts on input } w \} \]
\[ A_{TM} \leq_m \text{HALT}_{TM} \]

\( F = \) “On input \( <M, w> \):

1. Construct the machine

   \( M_\infty = \) “On input \( x \):
   
   1. Run \( M \) on \( x \).
   
   2. If \( M \) accepts, accept.
   
   3. If \( M \) rejects, loop.

2. Output \( <M_\infty, w> \).”
Solvable, Half-Solvable, Out-to-Lunch
\[ \text{EQ}_{TM} = \{ \langle M_1, M_2 \rangle \mid L(M_1) = L(M_2) \} \]
is Out-to-Lunch

**Theorem.** \( \text{EQ}_{TM} \) is neither Turing-recognizable nor co-Turing-recognizable.

**Proof.** We show \( A_{TM} \leq_m \text{EQ}_{TM} \). Why does this help?
$A_{TM} \leq_m EQ_{TM}$

$G = \text{"On input } <M, w>:\$

1. Construct the following two machines:
   $M_1 = \text{"On any input:} $
      
   1. Accept."

   $M_2 = \text{"On any input } x:$
      
   1. Ignore $x$ and run $M$ on $w$. If it accepts, accept."

2. Output $<M_1, M_2>$."

\[
\begin{array}{c}
\node[state,fill=white] (A) at (0,0) {$A_{TM}$};
\node[state,fill=white] (B) at (2,0) {$EQ_{TM}$};
\node[state,fill=white] (C) at (-2,0) {$A_{\overline{TM}}$};
\node[state,fill=white] (D) at (2,-2) {$\overline{EQ_{TM}}$};
\end{array}
\]

\[
\begin{array}{c}
\node[state,fill=white] (A) at (0,0) {$A_{TM}$};
\node[state,fill=white] (B) at (2,0) {$EQ_{TM}$};
\node[state,fill=white] (C) at (-2,0) {$A_{\overline{TM}}$};
\node[state,fill=white] (D) at (2,-2) {$\overline{EQ_{TM}}$};
\end{array}
\]
**Theorem.** \( \overline{EQ_{TM}} \) is not Turing-recognizable.

**Proof.** We show \( A_{TM} \leq_m \overline{EQ_{TM}} \).

\[
\begin{align*}
\text{EQ}_{TM} & \text{ is not Turing-recognizable} \\
\text{EQ}_{TM} & \text{ is neither Turing-recognizable nor co-Turing-recognizable.}
\end{align*}
\]
\( A_{TM} \leq_m E_{Q_{TM}} \)

\[ F = \text{"On input } \langle M, w \rangle \text{:} \]

1. Construct the following two machines:
   
   \( M_1 = \text{"On any input:} \]
   
   1. \( \text{Reject.} \)

   \( M_2 = \text{"On any input } x \text{:} \]
   
   1. \( \text{Ignore } x \text{ and run } M \text{ on } w. \)
      
   If it accepts, \( \text{accept.} \)

2. Output \( \langle M_1, M_2 \rangle \)."
1. Show that $A_{TM}$ is not mapping reducible to $E_{TM}$.
   (Hint: Use the fact that $A_{TM}$ is not Turing-recognizable whereas $E_{TM}$ is Turing-recognizable.)

2. Show that if $P$ is Turing-recognizable and $P \leq_m \overline{P}$, then $P$ is decidable.