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 \( w \),

\[
  w \in A \Leftrightarrow f(w) \in B.
\]

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

**Theorem.** \( E_{TM} \) is undecidable.

**Proof.** Given an input \( <M, w> \) we construct a machine \( M_w \) that accepts a nonempty language iff \( M \) accepts \( w \):

\[
  M_w = \text{"On input } x:\n  \begin{align*}
  1. \text{ If } x \neq w, \text{ reject.} \\
  2. \text{ If } x = w, \text{ run } M \text{ on input } w \text{ and accept if } M \text{ does."
  }
  \end{align*}
\]

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

\[
  S = \text{"On input } <M, w>:\n  \begin{align*}
  1. \text{ Use the description of } M \text{ and } w \text{ to construct } M_w. \\
  2. \text{ Run } R \text{ on input } <M_w>. \\
  3. \text{ If } R \text{ accepts, reject. If } R \text{ rejects, accept."
  }
  \end{align*}
\]

Problem Reduction

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

$A_{TM} = \{ <M, w> | M$ is a TM and $M$ accepts $w \}$  
$\leq_m$  
$HALT_{TM} = \{ <M, w> | M$ is a TM and $M$ halts on input $w \}$

$A_{TM} \leq_m HALT_{TM}$

$F = \text{“On input } <M, w>:\text{“}$

1. Construct the machine $M_w = \text{“On input } x:$
   - Run $M$ on $x.$
   - If $M$ accepts, accept.
   - If $M$ rejects, loop.
2. Output $<M_w, w>.$
**Solvable, Half-Solvable, Out-to-Lunch**

**Theorem.** The set $EQ_{TM} = \{ <M_1, M_2> \mid L(M_1) = L(M_2) \}$ is Out-to-Lunch.

**Proof.** We show $A_{TM} \leq_m EQ_{TM}$. Why does this help?

1. Construct the following two machines:
   - $M_1 = "\text{On any input:}"
     - 1. Accept."
   - $M_2 = "\text{On any input } x:
     - 1. Ignore } x \text{ and run } M \text{ on } w.
     - If it accepts, accept."

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

**EQ_{TM} is not Turing-recognizable**

**Theorem.** The set $EQ_{TM}$ is neither Turing-recognizable nor co-Turing-recognizable.

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

- $G =$ "On input $<M, w>$:
  1. Construct the following two machines:
     - $M_1 =$ "On any input:
       - 1. Accept."
     - $M_2 =$ "On any input $x$:
       - 1. Ignore $x$ and run $M$ on $w$.
       - If it accepts, accept."
  2. Output $<M_1, M_2>$."
F = "On input <M, w>:  
1. Construct the following two machines:  
   M₁ = "On any input:  
      1. Reject."  
   M₂ = "On any input x:  
      1. Ignore x and run M on w.  
      If it accepts, accept."  
2. Output <M₁, M₂>.

Exercises

1. Show that A_TM is not mapping reducible to E_TM. (Hint: Use the fact that A_TM is not Turing-recognizable whereas \overline{E_TM} is Turing-recognizable.)

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