L is Turing recognizable if \( \exists a \) a TM \( M \) such that \( L(M) = L \)

\[ L(M) = \{ w \mid M \text{ accepts } w \} \]

A language \( L \) is recursively enumerable if there exists an enumerator \( E \) that prints it out, i.e., for all \( x \in L \), \( E \) will eventually print out \( x \) (repetitions allowed)

An enumerator is a TM that ignores its input and proceeds to write down strings separated by ";"

Theorem: \( L \) is Turing recognizable if and only if recursively enumerable

Proof: Let \( E \) be the enumerator. Consider \( M \):

On input \( w \)

1. Simulate \( E \)
2. If \( E \) prints \( w \), accept
3. If \( E \) halts (w/o printing \( w \)) reject
Then $L(M) = \emptyset$ because

we never print it out, eventually, then $M$ accepts

we never print it out so $M$ never accepts

$\Rightarrow$ Suppose $L$ is Turing recognizable

Let $M_0$ be the TM that recognizes it. Consider the following simulator

"On waking up,

time steps = 10

for every string $w$ of length up to $l$

run $M_0$ on input $w$ for $t$ steps

If it accepts, print out $w$"
Analysis

Suppose \( w \in L \) then \( M \) accepts \( w \) at time \( t_w \).

Each iteration of the loop takes a finite amount of time, so after \( \log |w| \) iterations, 
\[ l \geq |w| \]

After \( \log |w| \) iterations, \( t \geq |w| \).

When \( E \) runs \( M \) on input \( w \) for \( t \) steps, \( M \) accepts, so \( E \) prints \( w \) at the output.

Suppose \( w \notin L \) Then \( M \) will never accept \( w \), so \( E \) will never print \( w \) at the output.

Lemma

If \( L \) and \( L^c \) are both T-rec, then \( L \) is decidable.

Corollary

If \( L \) is T-rec and undecidable, then \( L^c \) is not T-rec.

Recall

A function \( f \) is computable if \( \exists \) a TM \( M \) that \( M \) has an input \( x \), \( M \) halts and \( f(x) \) is written on its tape when it halts.
Definition: Let $A, B$ be languages.

$A \leq_m B$ if there exists a computable function $f : \Sigma^* \to \Sigma^*$ such that $w \in A \iff f(w) \in B$.

Such a function is called a reduction from $A$ to $B$.

Then let $A, B$ be languages. If $A \leq_m B$ and $B$ is decidable, then $A$ is decidable.

If let $f$ be the reduction from $A$ to $B$.

Let $M_B$ be the decider for $B$.

Then $M_A$ is a decider for $A$:

$M_A = \text{"On input } x \text{, compute } y = f(x) \text{, Run } M_B \text{ on input } y \text{ Accept if it accepts, Reject otherwise.}"

Corollary: If $A, B$ languages, s.t. $A \leq_m B$ and $A$ is undecidable then $B$ is undecidable.

Then let $A, B$ be languages. If $A \leq_m B$ and $B$ is T-en, then $A$ is T-en.
Proof. Let $f$ be the reduction from $A$ to $B$

let $M_B$ be a TM that recognizes $B$

then $M_A$ (same as before) recognizes $A$

Analysis: Suppose we $A$, then by def of a

reduction $f(w) \in B$, then $M_B$ accepts it

therefore $M_A$ accepts $w$

Suppose $w \notin A$. Then $f(w) \notin B$ then $M_B$

does not accept $f(w)$, so $M_A$ does not

accept either

Corollary: If $A, B$ lang., s.t. $A \leq_M B$ and

$A$ is not $T$-rec., then $B$ is not $T$-rec

$A, B$ languages, $A \leq_M B$ and $A$ is decidable, then

we don't know anything about $B$

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

Proof that $E_{Tm}$ is undecidable

Suppose $E_{Tm}$ decidable, then ATM decidable

as follows: let $DE$ be the decider of $E_{Tm}$. Consider $DA$
$D_A := \text{"On input } \langle M, w \rangle$,

construct $\langle M' \rangle$ where

$M' = \text{"On input } x$

Run $M$ on $w$

if it accepts, accept

else reject

Run $D_x(\langle M' \rangle)$ if it accepts, reject; else accept.

$A_{TM} \leq_{m} E_{TM}$

Therefore $E_{TM}$ is not Turing rec.

Is $E_{TM}$ Turing rec? (Yes)

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

$EQ_{TM}$ is not rec.

$EQ_{TM}$ is not non-reducible.

$W_{S}: E_{TM} \leq_{m} EQ_{TM}$

$f(\langle M \rangle) = \begin{cases} \langle M_1, M_2 \rangle \text{ s.t. } L(M_1) \geq L(M_2) \\ \text{if } L(M) = \emptyset \end{cases}$

\begin{align*}
\{ \langle M_1, M_2 \rangle \text{ s.t. } L(M_1) \neq L(M_2) \\
\text{if } L(M) \neq \emptyset
\end{align*}
\[ f(\langle M \rangle) = \langle M_1, M_2 \rangle \text{ where } M_1 \text{ rejects everything} \]
\[ M_2 = M \]

wts: \( EQ_{TM} \) not Turing-rec

sufficient to show \( ATM \leq m\overline{EQ_{TM}} \)
equivalently \( ATM \leq m EQ_{TM} \)

\[ f(\langle M, w \rangle) = \begin{cases} 
\langle M_1, M_2 \rangle \text{ where } L(M_1) = L(M_2) \\
\langle M_1, M_2 \rangle \text{ where } L(M_1) \neq L(M_2) 
\end{cases} \]

if \( M \) accepts

\( EQ_{TM} \) is not T-rec
\( EQ_{TM} \) is not T-rec