So far: \( A \text{TM} \) undecidable, \( T\text{-rec} \)
\( A \text{TM} \) not \( T\text{-rec} \)
Other undecidable problems:
\( \text{HALT}_{\text{TM}} \)
\( E_{\text{TM}} \)
\( EQ_{\text{TM}} \)

**Computable functions**

**Mapping reduction:**

**Definition:**
\( A \leq_m B \) iff there is a computable function \( f: \Sigma^* \rightarrow \Sigma^* \)
\( \text{such that } y \in \Sigma^* \Rightarrow w \in A \iff f(w) \in B \)

**Theorem:**
Suppose that \( A \leq_m B \):
- If \( B \) is decidable, then \( A \) is decidable.
- If \( A \) is undecidable, then \( B \) is undecidable.
- If \( B \) is \( T\text{-rec} \), then \( A \) is \( T\text{-rec} \).
- If \( A \) is not \( T\text{-rec} \), then \( B \) is not \( T\text{-rec} \).

**Correctness of Mapping Reduction**

Then \( A \leq_m B \iff \overline{A} \leq_m \overline{B} \)

**Figure:**

See picture
Thus \( A \leq_m B \) and \( B \leq_m C \implies A \leq_m C \)

**Proof**

Let \( f \) be a reduction given by \( TM M_f \)

showing that \( w \in A \iff f(w) \in B \)

Let \( g \) be a reduction given by \( TM M_g \)

showing that \( w \in B \iff g(f(w)) \in C \)

let \( w \in A \). \( w \in A \iff f(w) \in B \iff g(f(w)) \in C \)

\( g \circ f \) is a reduction showing \( A \leq_m C \)

Thus Neither \( EQ^m \) nor \( \overline{EQ^m} \) is \( T\text{-}rec \)

**Proof (a)** \( EQ^m \) is not \( T\text{-}rec \):

**Claim:** \( A_{TM} \leq_m EQ^m \)

Want \( s \) s.t. \( < M, w > \xrightarrow{c} < M_1, M_2 > \) and \( < M, w > \in A_{TM} \iff L(M_1) = L(M_2) \)
i.e. $M$ does not accept $w$ $\iff L(M_w) = L(M_{\#})$

Idea: $\langle M, w \rangle \mapsto \langle M_w, M_{\#} \rangle$

Clearly, $f$ is computable

Clearly, $f$ is computable

where

- $M_w$ is the TM that ignores its input and runs $M$ on input $w$
- $M_{\#}$ is a simple TM that always rejects

Now

$L(M_w) = \begin{cases} \emptyset & \text{if } M \text{ accepts } w \\ \Sigma^* & \text{if } M \text{ does not accept } w \end{cases}$

$\text{Correctness: } L(M_{\#}) = L(M_w)$ $\iff M$ does not accept $w$.

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

Claim: $\overline{A_{TM}} \leq_m \overline{EQ_{TM}}$

i.e. $A_{TM} \leq_m EQ_{TM}$

Want $f$ with

$\langle M, w \rangle \mapsto \langle M_1, M_2 \rangle$

$s + \text{M accepts } w$ $\iff L(M_1) = L(M_2)$

Similar idea: $\langle M, w \rangle \mapsto \langle M_w, M_{\#*} \rangle$

Clearly, $f$ is computable

where $M_{\#*}$ is a TM with input alphabet $\Sigma \cup \{\#\}$ that always accepts.
\[ L(M_{\text{rec}}) = \Sigma^* \]

and \[ L(M_w) = \Sigma^* \Rightarrow L(M_{\text{rec}}) \]

if \( M \) accepts \( w \).

\[ \text{Reduction is correct} \]

\[ \text{EQTM} \]

A much broader class of properties that are undecidable.

\[ P_{\text{TM}} = \{ <M> : M \text{ is a TM s.t. } L(M) \text{ has property } \mathcal{P} \} \]

Examples of properties \( \mathcal{P} \) = "empty" or "regular".

**Rice's Theorem**

Unless \( \mathcal{P} \) is "trivial" 

\( P_{\text{TM}} \) is undecidable.

We prove this in a special case first. 

\( \text{REGULAR}_{\text{TM}} = \{ <M> : L(M) \text{ is regular} \} \)
Thus REGULAR_TM is undecidable

Proof. We prove this by showing

Claim: $A_{TM} \leq_m \text{REGULAR}_{TM}$

We want $\langle M, w \rangle \overset{f}{\mapsto} \langle M' \rangle$

Goal: $M$ accepts $w \iff L(M')$ is regular

More specific: $M$ accepts $w \implies L(M') = \Sigma^*$

M doesn't accept $\implies L(M') = \{0^n1^n \mid n \geq 0\}$

Codebook of $M'$:

On input $x$:
- if $x$ is of form $0^n1^n$ for some $n$ then accept
- otherwise, run $M$ on input $w$ & accept iff $M$ does

Clearly $f$ is computable

Correctness: If $M$ accepts $w$ then $M'$ accepts every string $M$ doesn't accept $w$ run $M'$ accept strings of the form $0^n1^n$.

$\therefore L(M')$ is regular $\iff M$ accepts $w$
Rice's Theorem:

**Definition**

A property \( P \) is **non-trivial** iff there is some TM \( M_1 \) s.t. \( L(M_1) \) has property \( P \) and there is some TM \( M_0 \) s.t. \( L(M_0) \) does not have property \( P \).

**Examples of trivial properties:**
- \( L(M) \) is T-reducible (always has \( P \))
- \( L(M) \) is not T-reducible (never has \( P \))

**Proof of Rice's Theorem**

**Case 1:** \( \Sigma^* \) has property \( P \)

We use TM \( M_0 \) s.t. \( L(M_0) \) does not have property \( P \)

**Claim**

ATM \( \leq_m P_{TM} \)

**Want s.t.**

\(<M, w> \xrightarrow{f} <M_0>\)

\(M \) accepts \( w \) \iff \( L(M_0) \) has property \( P \)

**Design goal:** \( L(M_0) = \begin{cases} \Sigma^* & \text{if} \ M \text{ accepts } w \\ L(M_0) & \text{if} \ M \text{ does not accept } w \end{cases} \)

Want to use similar idea as for REGULAR_TM
Design of $M_p$ (Attempt 1)

On input $x$
- Run $M_o$ on input $x$
- If $M_o$ accepts then accept
- Else run $M$ on input $w$ and accept if $M$ does

Problem: $M_o$ may not halt (unlike test for $0^n1^n$)

We want $x$ to be accepted if
- ($M_o$ accepts $x$ or $M$ accepts $w$)

Actual Design for $M_p$:

On input $x$:
- Run $M_o$ on input $x$
- Run $M$ on input $w$ in parallel one step at a time
- If either accepts then accept

If $M$ accepts $w$ then $L(M_p) = \Sigma^*$ has property $P$ and if not, then $L(M_p) \neq L(M_o)$ hence

$.\ A_{TM} \leq_{P} P_{TM}$ so $P_{TM}$ is undecidable.

Case 2: $\Sigma^*$ does not have property $P$

We prove that $A_{TM} \leq_{m} P_{TM}$

$.\ P_{TM}$ is undecidable which means that $P_{TM}$ is undecidable.
We use $\overline{P}$, the complement of property $P$
\[ L(M) \text{ has property } P \]
\[ L(M) \text{ does not have property } \overline{P}. \]
On the other hand, $S^x$ has property $P$.

To show $A^m \leq_m \overline{P}^m$

we use $M$, in place of $M_0$ and the same proof idea to get $M_0^\overline{P}$.

\[ \overline{P}^m \text{ is undecidable and so } P^m \text{ is undecidable as required.} \]

We give an alternative proof in the notes.