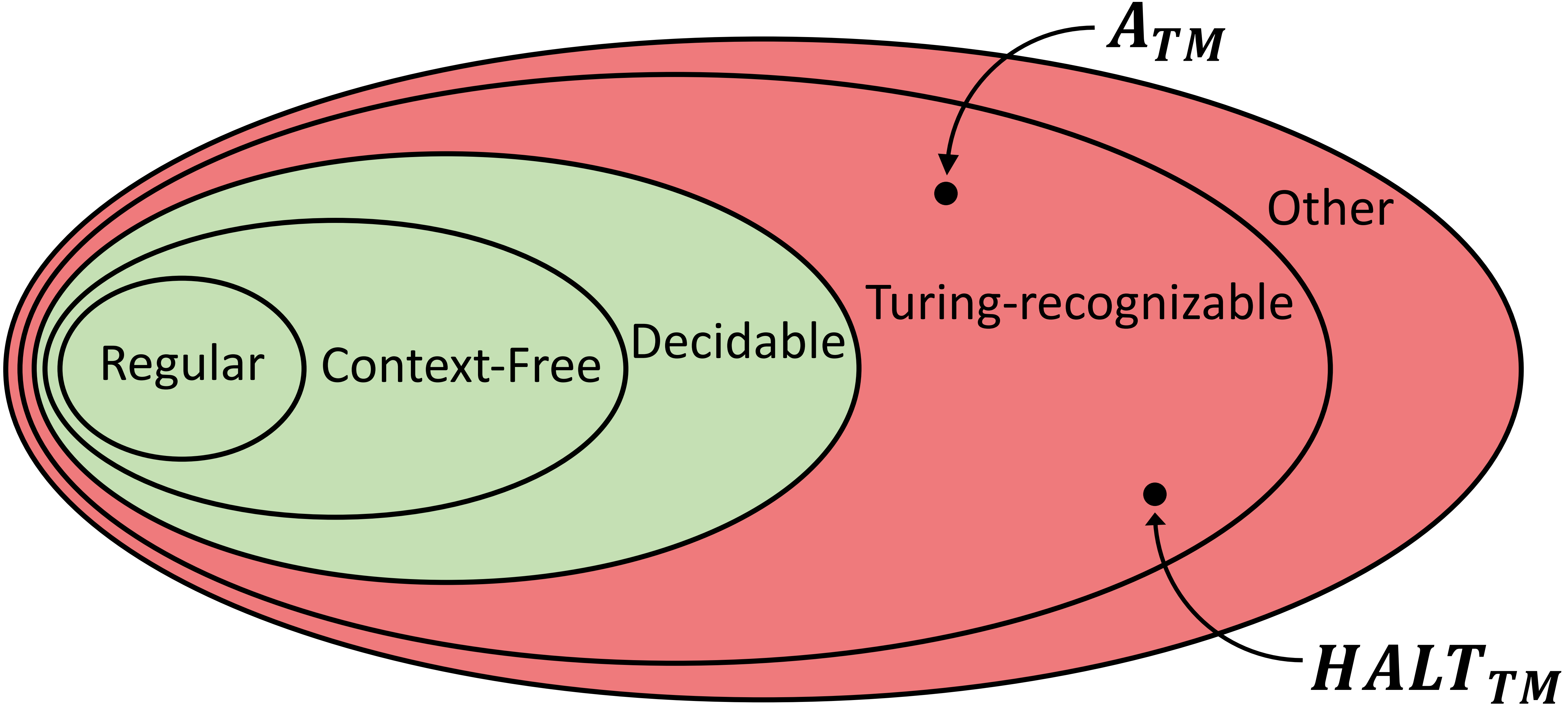
Undecidability

CSCI 338

Computability Hierarchy



A_{TM}

Claim: $A_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M accepts ω} is undecidable.

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

What happens with $D(\langle D \rangle)$? $D(\langle D \rangle) = \begin{cases} \text{accept, if } D \text{ does not accept } \langle D \rangle \\ \text{reject, if } D \text{ accepts } \langle D \rangle \end{cases}$

D accepts $\langle D \rangle$, so long as D does not accept $\langle D \rangle$.

$\Rightarrow$ TM D cannot exist $\Longrightarrow$ TM H cannot exist $\Longrightarrow$ A_{TM} is undecidable

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M halts on ω} is undecidable.

Proof: Suppose $HALT_{TM}$ is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega \rangle$

1. Run H on $\langle M, \omega \rangle$.
2. If H rejects, <u>reject</u> (i.e. M does not halt on ω).
3. If H accepts, run M on ω until it halts.
4. If M accepts, <u>accept</u>. If M rejects, <u>reject</u>.

S is a decider for A_{TM}, which is a contradiction.

$\therefore HALT_{TM}$ is undecidable.

Undecidability Proof Blueprint

Claim: $\boldsymbol{New\_Problem}$ is undecidable.

Proof:

$\boldsymbol{New\_Problem}$: Problem we are trying to show is undecidable.

Undecidability Proof Blueprint

Claim: $\boldsymbol{New\_Problem}$ is undecidable.

Proof: Suppose $\boldsymbol{New\_Problem}$ is decidable and let H be its decider.

$\boldsymbol{New\_Problem}$: Problem we are trying to show is undecidable.

Undecidability Proof Blueprint

Claim: $\boldsymbol{New\_Problem}$ is undecidable.

Proof: Suppose $\boldsymbol{New\_Problem}$ is decidable and let H be its decider.

Build a TM S that decides $\boldsymbol{Old\_Problem}$:

$\boldsymbol{New\_Problem}$: Problem we are trying to show is undecidable.
$\boldsymbol{Old\_Problem}$: Problem we already know to be undecidable.

Undecidability Proof Blueprint

Claim: $\boldsymbol{New\_Problem}$ is undecidable.

Proof: Suppose $\boldsymbol{New\_Problem}$ is decidable and let H be its decider.

Build a TM S that decides $\boldsymbol{Old\_Problem}$:

S = on input $\langle ? \rangle$

1. …

$\vdots$

$\boldsymbol{New\_Problem}$: Problem we are trying to show is undecidable.
$\boldsymbol{Old\_Problem}$: Problem we already know to be undecidable.

Undecidability Proof Blueprint

Claim: $\boldsymbol{New\_Problem}$ is undecidable.

Proof: Suppose $\boldsymbol{New\_Problem}$ is decidable and let H be its decider.

Build a TM S that decides $\boldsymbol{Old\_Problem}$:

S = on input $\langle ? \rangle$

1. …

⋮

Input depends on the input to $\boldsymbol{Old\_Problem}$.

$\boldsymbol{New\_Problem}$: Problem we are trying to show is undecidable.
$\boldsymbol{Old\_Problem}$: Problem we already know to be undecidable.

Undecidability Proof Blueprint

Claim: $\boldsymbol{New\_Problem}$ is undecidable.

Proof: Suppose $\boldsymbol{New\_Problem}$ is decidable and let H be its decider.

Build a TM S that decides $\boldsymbol{Old\_Problem}$:

S = on input $\langle ? \rangle$

1. …

⋮

Input depends on the input to $\boldsymbol{Old\_Problem}$.

S is a decider for $\boldsymbol{Old\_Problem}$, which is a contradiction.

$\therefore$ $\boldsymbol{New\_Problem}$ is undecidable.

$\boldsymbol{New\_Problem}$: Problem we are trying to show is undecidable.
$\boldsymbol{Old\_Problem}$: Problem we already know to be undecidable.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof:

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1.

To show E_{TM} is undecidable, use it to decide A_{TM}.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1.

$\boxed{\text{To show } E_{TM} \text{ is undecidable, use it to decide } A_{TM}.}$

$$L(?\,) \neq \emptyset \quad \Longleftrightarrow \quad N \text{ accepts } \omega$$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:

$\boldsymbol{N}$ **accepts** $\boldsymbol{\omega \Longleftrightarrow L(M_2) \neq \emptyset}$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:

   $\boldsymbol{N}$ **accepts** $\boldsymbol{\omega \Leftrightarrow L(M_2) \neq \emptyset}$

2. Run H on $\langle M_2 \rangle$.
3. ?

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:

   $\boldsymbol{N}$ **accepts** $\boldsymbol{\omega \Leftrightarrow L(M_2) \neq \emptyset}$

2. Run H on $\langle M_2 \rangle$.
3. If H accepts, ?.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M \text{ is a TM and } L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:

   $\boldsymbol{N}$ **accepts** $\boldsymbol{\omega \Leftrightarrow L(M_2) \neq \emptyset}$

2. Run H on $\langle M_2 \rangle$.
3. If H accepts, ?.

If H accepts M_2, then…?

E_{TM}

Claim: $E_{TM} = \{\langle M\rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_2 on input $\langle x\rangle$:

   $\boldsymbol{N}$ **accepts** $\boldsymbol{\omega \Leftrightarrow L(M_2) \neq \emptyset}$

2. Run H on $\langle M_2\rangle$.
3. If H accepts, ?.

If $\boldsymbol{H}$ accepts $\boldsymbol{M_2}$, then $\boldsymbol{L(M_2) = \emptyset}$, which means…?

E_{TM}

Claim: $E_{TM} = \{\langle M\rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_2 on input $\langle x\rangle$:

   N accepts $\omega \Leftrightarrow L(M_2) \neq \emptyset$

2. Run H on $\langle M_2\rangle$.
3. If H accepts, ?.

If H accepts M_2, then $L(M_2) = \emptyset$, which means that N does not accept ω.

E_{TM}

Claim: $E_{TM} = \{\langle M\rangle: M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_2 on input $\langle x\rangle$:

   $\boldsymbol{N}$ **accepts** $\boldsymbol{\omega \Leftrightarrow L(M_2) \neq \emptyset}$

2. Run H on $\langle M_2\rangle$.
3. If H accepts, reject.

If $\boldsymbol{H}$ accepts $\boldsymbol{M_2}$, then $\boldsymbol{L(M_2) = \emptyset}$, which means that $\boldsymbol{N}$ does not accept $\boldsymbol{\omega}$.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:

   $\boldsymbol{N}$ **accepts** $\boldsymbol{\omega \Leftrightarrow L(M_2) \neq \emptyset}$

2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject.

If $\boldsymbol{H}$ rejects $\boldsymbol{M_2}$, then $\boldsymbol{L(M_2) \neq \emptyset}$, which means that $\boldsymbol{N}$ does accept $\boldsymbol{\omega}$.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:

   N accepts $\omega \Leftrightarrow L(M_2) \neq \emptyset$

2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

If H rejects M_2, then $L(M_2) \neq \emptyset$, which means that N does accept ω.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

$\boldsymbol{N}$ **accepts** $\boldsymbol{\omega}$

$\Updownarrow$

$\boldsymbol{L(M_2) \neq \emptyset}$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. If $x \neq \omega$, <u>reject</u>.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

N accepts ω

$$\Updownarrow$$

$$\boldsymbol{L(M_2) \neq \emptyset}$$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. If $x \neq \omega$, <u>reject</u>.
    2. If $x = \omega$ run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

$\boldsymbol{N}$ **accepts** $\boldsymbol{\omega}$

$\Updownarrow$

$\boldsymbol{L(M_2) \neq \emptyset}$

E_{TM}

Claim: $E_{TM} = \{\langle M\rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_2 on input $\langle x\rangle$:
   1. If $x \neq \omega$, reject.
   2. If $x = \omega$ run N on ω and accept if N does.
2. Run H on $\langle M_2\rangle$.
3. If H accepts, reject. If H rejects, accept.

$\boldsymbol{N}$ **accepts** $\boldsymbol{\omega}$

$\Updownarrow$

$\boldsymbol{L(M_2) \neq \emptyset}$

$\boldsymbol{L(M_2) = ??}$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. If $x \neq \omega$, reject.
   2. If $x = \omega$ run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

N accepts ω

$$\boldsymbol{\Updownarrow}$$

$$\boldsymbol{L(M_2) \neq \emptyset}$$

$\boldsymbol{L(M_2) = \{\omega\}}$ **or** $\boldsymbol{\emptyset}$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M \text{ is a TM and } L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. Run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

$$\boldsymbol{N} \textbf{ accepts } \boldsymbol{\omega} \Updownarrow \boldsymbol{L(M_2) \neq \emptyset}$$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. Run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

$\boldsymbol{N}$ **accepts** $\boldsymbol{\omega}$

$\Updownarrow$

$\boldsymbol{L(M_2) \neq \emptyset}$

$\boldsymbol{L(M_2) = ??}$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. Run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

N accepts ω

$\Updownarrow$

$\boldsymbol{L(M_2) \neq \emptyset}$

$\boldsymbol{L(M_2) = \Sigma^*}$ **or** $\boldsymbol{\emptyset}$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. Run N on ω and accept if N does **not**.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

***N* accepts $\boldsymbol{\omega}$**

$\Updownarrow$

$\boldsymbol{L(M_2) \neq \emptyset}$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. Run N on ω and accept if N does **not**.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

$$\boldsymbol{N \text{ accepts } \omega} \Updownarrow \boldsymbol{L(M_2) = \emptyset}$$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. Run N on ω and accept if N does **not**.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

N accepts ω

$$\Updownarrow$$

$$\boldsymbol{L(M_2) = \emptyset}$$

$$\boldsymbol{L(M_2) = \emptyset \text{ or } \Sigma^*}$$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. Run N on ω and accept if N does **not**.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, accept. If H rejects, reject.

$\boldsymbol{N}$ **accepts** $\boldsymbol{\omega}$

$\Updownarrow$

$\boldsymbol{L(M_2) = \emptyset}$

$\boldsymbol{L(M_2) = \emptyset}$ **or** $\boldsymbol{\Sigma^*}$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. If $x \neq \omega$, reject.
   2. If $x = \omega$ run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

$$\boldsymbol{N \text{ accepts } \omega \Updownarrow L(M_2) \neq \emptyset}$$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M \text{ is a TM and } L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. If $x \neq \omega$, reject.
   2. If $x = \omega$ run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

N accepts ω

$$\Updownarrow$$

$$\boldsymbol{L(M_2) \neq \emptyset}$$

E_{TM}

Claim: $E_{TM} = \{\langle M\rangle : M \text{ is a TM and } L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_2 on input $\langle x\rangle$:
   1. If $x \neq \omega$, reject.
   2. If $x = \omega$ run N on ω and accept if N does.
2. Run H on $\langle M_2\rangle$.
3. If H accepts, reject. If H rejects, accept.

N accepts ω

$\Updownarrow$

$\boldsymbol{L(M_2) \neq \emptyset}$

If N accepts ω, $L(M_2) = ?$

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. If $x \neq \omega$, reject.
   2. If $x = \omega$ run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

N accepts ω

$$\Updownarrow$$

$$\boldsymbol{L(M_2) \neq \emptyset}$$

If N accepts ω, $L(M_2) = \{\omega\}$, H will…?

E_{TM}

Claim: $E_{TM} = \{\langle M\rangle : M \text{ is a TM and } L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_2 on input $\langle x\rangle$:
    1. If $x \neq \omega$, reject.
    2. If $x = \omega$ run N on ω and accept if N does.
2. Run H on $\langle M_2\rangle$.
3. If H accepts, reject. If H rejects, accept.

N accepts ω

$$\Updownarrow$$

$$\boldsymbol{L(M_2) \neq \emptyset}$$

If N accepts ω, $L(M_2) = \{\omega\}$, H will reject, and S will…?

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. If $x \neq \omega$, <u>reject</u>.
   2. If $x = \omega$ run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

N accepts ω

$\Updownarrow$

$\boldsymbol{L(M_2) \neq \emptyset}$

If N accepts ω, $L(M_2) = \{\omega\}$, H will reject, and S will accept.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. If $x \neq \omega$, reject.
    2. If $x = \omega$ run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

N accepts ω

$$\Updownarrow$$

$$\boldsymbol{L(M_2) \neq \emptyset}$$

If N accepts ω, $L(M_2) = \{\omega\}$, H will reject, and S will accept. If N does not accept ω, $L(M_2) = ?$

E_{TM}

Claim: $E_{TM} = \{\langle M\rangle : M \text{ is a TM and } L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_2 on input $\langle x\rangle$:
   1. If $x \neq \omega$, <u>reject</u>.
   2. If $x = \omega$ run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2\rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

N accepts ω

$\Updownarrow$

$\boldsymbol{L(M_2) \neq \emptyset}$

If N accepts ω, $L(M_2) = \{\omega\}$, H will reject, and S will accept. If N does not accept ω, $L(M_2) = \emptyset$, H will…?

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. If $x \neq \omega$, reject.
    2. If $x = \omega$ run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

$\boxed{\begin{array}{c} \boldsymbol{N \text{ accepts } \omega} \\ \Updownarrow \\ \boldsymbol{L(M_2) \neq \emptyset} \end{array}}$

If N accepts ω, $L(M_2) = \{\omega\}$, H will reject, and S will accept. If N does not accept ω, $L(M_2) = \emptyset$, H will accept, and S will…?

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. If $x \neq \omega$, reject.
    2. If $x = \omega$ run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

$\boxed{\begin{array}{c} \boldsymbol{N} \textbf{ accepts } \boldsymbol{\omega} \\ \Updownarrow \\ \boldsymbol{L(M_2) \neq \emptyset} \end{array}}$

If N accepts ω, $L(M_2) = \{\omega\}$, H will reject, and S will accept. If N does not accept ω, $L(M_2) = \emptyset$, H will accept, and S will reject.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M \text{ is a TM and } L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. If $x \neq \omega$, <u>reject</u>.
   2. If $x = \omega$ run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

$$\boxed{\begin{array}{c} \boldsymbol{N} \textbf{ accepts } \boldsymbol{\omega} \\ \Updownarrow \\ \boldsymbol{L(M_2) \neq \emptyset} \end{array}}$$

If N accepts ω, $L(M_2) = \{\omega\}$, H will reject, and S will accept. If N does not accept ω, $L(M_2) = \emptyset$, H will accept, and S will reject. So, using H to decide if $L(M_2) = \emptyset$ determines if N accepts ω.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M \text{ is a TM and } L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. If $x \neq \omega$, reject.
    2. If $x = \omega$ run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

$$\boxed{\begin{array}{c} \boldsymbol{N} \textbf{ accepts } \boldsymbol{\omega} \\ \Updownarrow \\ \boldsymbol{L(M_2) \neq \emptyset} \end{array}}$$

If N accepts ω, $L(M_2) = \{\omega\}$, H will reject, and S will accept. If N does not accept ω, $L(M_2) = \emptyset$, H will accept, and S will reject. So, using H to decide if $L(M_2) = \emptyset$ determines if N accepts ω. $\Rightarrow$ S decides A_{TM}

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. If $x \neq \omega$, <u>reject</u>.
    2. If $x = \omega$ run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

N accepts ω

$$\Updownarrow$$

$$\boldsymbol{L(M_2) \neq \emptyset}$$

If N accepts ω, $L(M_2) = \{\omega\}$, H will reject, and S will accept. If N does not accept ω, $L(M_2) = \emptyset$, H will accept, and S will reject. So, using H to decide if $L(M_2) = \emptyset$ determines if N accepts ω. $\Rightarrow$ S decides $A_{TM} \Rightarrow E_{TM}$ is undecidable.

When in doubt use A_{TM}!!!

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N \rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof:

?

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides ???:

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides ???:

A_{TM}?

$HALT_{TM}$?

E_{TM}?

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides ???:

$\left\{ \begin{array}{l} \boldsymbol{A_{TM}?} \\ HALT_{TM}? \\ E_{TM}? \end{array} \right.$

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. ???

$$\boxed{\text{To show } EQ_{TM} \text{ is undecidable, use it to decide } A_{TM}.}$$

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. ???

We have a way (H) to test if two TMs have the same language.

How could we use that to test if a TM's accepts some input?

Plan: ?

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. ???

We have a way (H) to test if two TMs have the same language.

How could we use that to test if a TM's accepts some input?

Plan: Make two TMs that…?

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. ???

We have a way (H) to test if two TMs have the same language.

How could we use that to test if a TM's accepts some input?

Plan: Make two TMs that have the same language if and only if N accepts ω.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N \rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_1 on input $\langle x \rangle$:
    1. accept.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. accept.

$\boldsymbol{L(M_2) = ?}$

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$: $\boldsymbol{L(M_2) = \Sigma^*}$
    1. <u>accept.</u>

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. <u>accept.</u>
2. Construct TM M_2 on input $\langle y\rangle$:
   1. ?

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. <u>accept.</u>
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and <u>accept</u> if N does.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. accept.
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and accept if N does.

$$\boxed{\begin{array}{c} \boldsymbol{N} \textbf{ accepts } \boldsymbol{\omega} \\ \Updownarrow \\ \boldsymbol{L(M_2) = \Sigma^*} \end{array}}$$

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N \rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_1 on input $\langle x \rangle$:
   1. <u>accept.</u>
2. Construct TM M_2 on input $\langle y \rangle$:
   1. Run N on ω and <u>accept</u> if N does.
3. Run H on $\langle M_1, M_2 \rangle$.
4. If H accepts, <u>???</u>

$\boxed{\textbf{N accepts } \boldsymbol{\omega} \iff \boldsymbol{L(M_2) = \Sigma^*}}$

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. <u>accept.</u>
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and <u>accept</u> if N does.
3. Run H on $\langle M_1, M_2\rangle$.
4. If H accepts, <u>accept</u>. If H rejects, <u>reject</u>.

$$\boxed{\begin{gathered} \boldsymbol{N} \textbf{ accepts } \boldsymbol{\omega} \\ \Updownarrow \\ \boldsymbol{L(M_2) = \Sigma^*} \end{gathered}}$$

EQ_{TM}

Claim: $EQ_{TM} = \{\langle M, N\rangle : M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. <u>accept.</u>
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and <u>accept</u> if N does.
3. Run H on $\langle M_1, M_2\rangle$.
4. If H accepts, <u>accept</u>. If H rejects, <u>reject</u>.

$$\boldsymbol{N} \textbf{ accepts } \boldsymbol{\omega} \;\Updownarrow\; \boldsymbol{L(M_2) = \Sigma^*}$$

If N accepts ω, then M_1 and M_2 have the same language (Σ^*). If N does not accept ω, then they have different languages. Thus S decides A_{TM}. (bad!)