Undecidability
CSCI 338

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.

If N accepts ω, $L(M_2) = \Sigma^*$, H will reject, and S will accept. If N does not accept ω, $L(M_2) = \emptyset$, H will accept, and S will reject. Thus, S decides A_{TM}, which is a contradiction, so E_{TM} is undecidable.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ 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>.

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

If N accepts ω, then M_1 and M_2 have the same language (Σ^*) and S accepts. If N does not accept ω, then they have different languages and S rejects. Thus, S decides A_{TM}, which is a contradiction, so EQ_{TM} is undecidable.

$REGULAR_{TM}$

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

Proof:

?

$REGULAR_{TM}$

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

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

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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.

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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.

Plan: Build a TM whose language is regular if N accepts ω and not regular if N does not accept ω.

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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 $\langle x\rangle$:

?

$L(M_2)$ is regular

$\Updownarrow$

N accepts ω

Plan: Build a TM whose language is regular if N accepts ω and not regular if N does not accept ω.

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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 $x \in \{ \quad ??? \quad \}$, <u>accept</u>.
   2. If $x \notin \{ \quad ??? \quad \}$, run N on ω and <u>accept</u> if N does.

$L(M_2)$ is regular

$\Updownarrow$

N accepts ω

Plan: Build a TM whose language is regular if N accepts ω and not regular if N does not accept ω.

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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 $x \in \{0^n 1^n : n \geq 0\}$, <u>accept</u>.
    2. If $x \notin \{0^n 1^n : n \geq 0\}$, run N on ω and <u>accept</u> if N does.

$L(M_2)$ is regular

$\Updownarrow$

N accepts ω

Plan: Build a TM whose language is regular if N accepts ω and not regular if N does not accept ω.

$REGULAR_{TM}$

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

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

Build a TM S that decides A_{TM}:

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

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

1. Construct TM M_2 on input $\langle x \rangle$:
   1. If $x \in \{0^n 1^n : n \geq 0\}$, <u>accept</u>.
   2. If $x \notin \{0^n 1^n : n \geq 0\}$, run N on ω and <u>accept</u> if N does.

Plan: Build a TM whose language is regular if N accepts ω and not regular if N does not accept ω.

$REGULAR_{TM}$

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

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

Build a TM S that decides A_{TM}:

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

$\boldsymbol{L(M_2) = 0^n 1^n}$ **or** $\boldsymbol{\Sigma^*}$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. If $x \in \{0^n 1^n : n \geq 0\}$, <u>accept</u>.
   2. If $x \notin \{0^n 1^n : n \geq 0\}$, run N on ω and <u>accept</u> if N does.

Plan: Build a TM whose language is regular if N accepts ω and not regular if N does not accept ω.

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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 $x \in \{0^n 1^n : n \geq 0\}$, <u>accept</u>.
   2. If $x \notin \{0^n 1^n : n \geq 0\}$, run N on ω and <u>accept</u> if N does.
2. ?

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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 $x \in \{0^n 1^n : n \geq 0\}$, <u>accept</u>.
   2. If $x \notin \{0^n 1^n : n \geq 0\}$, run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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 $x \in \{0^n 1^n : n \geq 0\}$, <u>accept</u>.
   2. If $x \notin \{0^n 1^n : n \geq 0\}$, run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>accept</u>. If H rejects, <u>reject</u>.

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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 $x \in \{0^n 1^n : n \geq 0\}$, <u>accept</u>.
   2. If $x \notin \{0^n 1^n : n \geq 0\}$, run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>accept</u>. If H rejects, <u>reject</u>.

If N accepts ω, … S accepts.

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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 $x \in \{0^n 1^n : n \geq 0\}$, <u>accept</u>.
   2. If $x \notin \{0^n 1^n : n \geq 0\}$, run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>accept</u>. If H rejects, <u>reject</u>.

If N accepts ω, $L(M_2) = \Sigma^*$ (regular) and S accepts.

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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 $x \in \{0^n 1^n : n \geq 0\}$, <u>accept</u>.
   2. If $x \notin \{0^n 1^n : n \geq 0\}$, run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2\rangle$.
3. If H accepts, <u>accept</u>. If H rejects, <u>reject</u>.

If N accepts ω, $L(M_2) = \Sigma^*$ (regular) and S accepts. If N does not accept ω, … S rejects.

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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 $x \in \{0^n 1^n : n \geq 0\}$, <u>accept</u>.
   2. If $x \notin \{0^n 1^n : n \geq 0\}$, run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>accept</u>. If H rejects, <u>reject</u>.

If N accepts ω, $L(M_2) = \Sigma^*$ (regular) and S accepts. If N does not accept ω, $L(M_2) = \{0^n 1^n : n \geq 0\}$ (not regular) and S rejects.

$REGULAR_{TM}$

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

Proof: Suppose $REGULAR_{TM}$ 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 $x \in \{0^n 1^n : n \geq 0\}$, <u>accept</u>.
    2. If $x \notin \{0^n 1^n : n \geq 0\}$, run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2\rangle$.
3. If H accepts, <u>accept</u>. If H rejects, <u>reject</u>.

If N accepts ω, $L(M_2) = \Sigma^*$ (regular) and S accepts. If N does not accept ω, $L(M_2) = \{0^n 1^n : n \geq 0\}$ (not regular) and S rejects. Therefore, S is a decider for A_{TM}, which is a contradiction, so $REGULAR_{TM}$ is undecidable.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle: A, B$ are TMs and $L(A) = L(B)\}$ 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.
3. Run H on $\langle M_1, M_2\rangle$.
4. If H accepts, accept. If H rejects, reject.

If N accepts ω, then M_1 and M_2 have the same language (Σ^*) and S accepts. If N does not accept ω, then they have different languages and S rejects. Thus, S decides A_{TM}, which is a contradiction, so EQ_{TM} is undecidable.

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

1.

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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 language is empty?

Plan: ?

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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 language is empty?

Plan: Make a TM with an empty language and use H to compare it to input to E_{TM}.

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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

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

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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

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

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

1. Construct TM M_2 on input $\langle x\rangle$:
   1. reject.
2. Run H on $\langle P, M_2\rangle$.

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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

If $L(P) = \emptyset$, … S will accept. If $L(P) \neq \emptyset$, … S will reject.

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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

If $L(P) = \emptyset$, M_2 and P will have the same language (since $L(M_2) = \emptyset$) and S will accept. If $L(P) \neq \emptyset$, … S will reject.

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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

If $L(P) = \emptyset$, M_2 and P will have the same language (since $L(M_2) = \emptyset$) and S will accept. If $L(P) \neq \emptyset$, M_2 and P will not have the same language and S will reject.

EQ_{TM}

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

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

Build a TM S that decides E_{TM}:

S = on input $\langle P\rangle$

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

If $L(P) = \emptyset$, M_2 and P will have the same language (since $L(M_2) = \emptyset$) and S will accept. If $L(P) \neq \emptyset$, M_2 and P will not have the same language and S will reject. Therefore, S is a decider for E_{TM}, which is a contradiction, so EQ_{TM} is undecidable.