Proof:

$A_{TM} = \{\langle M, w\rangle : M \text{ is a TM and } w \in L(M)\}$

Goal Show: A_{TM} is undecidable.

Proof by contradiction:

Assume A_{TM} is decidable.

Then let

$H = $ "On input $\langle M, w\rangle$, where M is a TM and w is a string:

1. Run D on input $\langle M, w\rangle$.
2. If D accepts, accept.
3. If D rejects, reject."

Claim: H decides A_{TM}.

(Exercise left out.)

Thm (Decidable Recognizable Languages Theorem):

Then need a computable function S such that on input $\langle \langle M\rangle, \langle w\rangle\rangle$, where M is a TM and w is a string

outputs a string $\langle x\rangle$ where M is a TM, w is a string, and $\langle x\rangle$ is the input to M that M accepts on input $\langle M, w\rangle$.

Proof:

$S =$ "On input $\langle M, w\rangle$:

1. Let $x = $ "$\langle M, w\rangle$ accepts w"

$\langle x\rangle = S(\langle M\rangle, \langle w\rangle)$

Lecture 2026-3-18

Some proofs of Undecidability

Def:

$HALT_{TM} = \{\langle M, w\rangle : M \text{ is a TM and } M \text{ halts on } w\}$

Thm: $HALT_{TM}$ is undecidable.

Proof: Assume otherwise.

Let

$H =$ "On input $\langle M, w\rangle$ where M is a TM and w is a string:

1. Let $x = $ "$\langle M, w\rangle$ accepts w"
2. If $D(\langle M, x\rangle) = $ accept, accept.
3. If $D(\langle M, x\rangle) = $ reject, reject."

Claim: H decides A_{TM}.

For any TM M & string w:

$\langle M, w\rangle \in A_{TM}$ iff M accepts w

$\Rightarrow H$ accepts $\langle M, w\rangle$ iff M accepts w

$\Rightarrow L(H) = \Sigma^*$

$\Rightarrow L(H)$ is decidable

$\Rightarrow D(\langle M, x\rangle) \in \{\text{accept}, \text{reject}\}$

$\langle M, w\rangle \notin A_{TM}$ iff M does not accept w

$\Rightarrow H$ does not accept $\langle M, w\rangle$

$\Rightarrow L(H) = \emptyset$

$\Rightarrow L(H)$ is decidable

$\Rightarrow D(\langle M, x\rangle) \in \{\text{accept}, \text{reject}\}$

$\Rightarrow H$ rejects $\langle M, w\rangle$

So H decides A_{TM} &

$HALT_{TM}$ is decidable.

Def:

$HALT_{TM} = \{\langle M, w\rangle : M \text{ halts on } w\}$

Thm: $HALT_{TM}$ is undecidable.

Proof: Assume otherwise. Let D be a decider for $HALT_{TM}$. Then D satisfies the following properties:

1. L is a language
2. Then L is a computable function

$f(\langle w\rangle) = \langle x\rangle$

such that

$\forall w \in \Sigma^*$, L has output x on input w if $w \in L$

Then L is a decidable language.

Proof:

$D =$ "On input w:

1. Let $x = f(w)$
2. Run L on input x.
3. If L accepts, accept.
4. If L rejects, reject."

Claim: D decides L.

Proof: Let $w \in \Sigma^*$.

Then D halts on input w iff L halts on input x iff L accepts x iff D accepts w.

possible ID's is

$$\underbrace{|\Gamma|^s}_{\substack{\#\text{ tape}\\ \text{contents}\\ \text{before}\\ \text{step } 0\ldots(s-1)}} \times \underbrace{|Q|}_{\substack{\#\text{ of}\\ \text{possible}\\ \text{states}}} \times \underbrace{s}_{\substack{\#\text{ possible}\\ \text{head positions}\\ (0\ldots(s-1))}}$$

$\therefore$ If M loops without scanning cell s, it must repeat an ID (Pigeonhole principle).

$\therefore$ (c) occurs.

$\therefore D$ is a decider

$\therefore D$ decides $L(M)$. $\boxtimes$

Reducibility

Def: Let $A, B \subseteq \Sigma^*$ languages. We say that A mapping reduces to B ($A \leq_m B$) if there exists a computable function $f : \Sigma^* \to \Sigma^*$ such that, for all $w \in \Sigma^*$

$w \in A \iff f(w) \in B$.

Such an f is called a mapping reduction from A to B ($f : A \leq_m B$).

Thm: Let $A, B \subseteq \Sigma^*$ such that $A \leq_m B$. Then

1. If B is decidable, then A is decidable.
2. If B is Turing, then A is Turing.

Proof: Let $f : A \leq_m B$ mapping reduce (m-reduce) A to B. Let M be a TM such that $B = L(M)$. Then let

$N :=$ "On input w:

1. Let $x := f(w)$
2. Run M on input x."

[Finish next time]