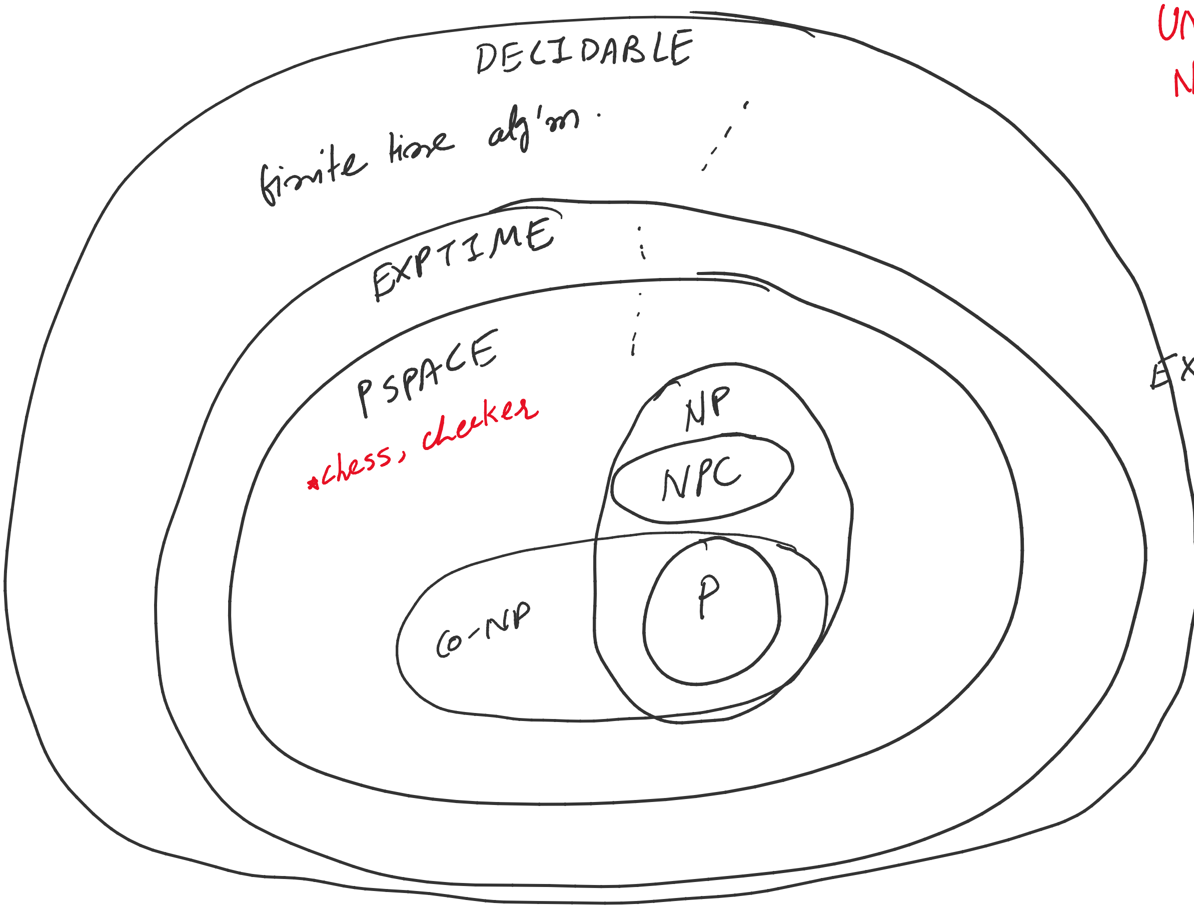
Thursday, April 25, 2024 10:54 AM

Midterm 2:
Mean, Median ~ 70!

Beyond-NP.

* Halting Problem.
UNDECIDABLE
No finite time
Alg'm!



EXPTIME ≠ P

UN-DECIDABILITY

→ TM ≡ Alg'm ≡ Computer Program

→ Problem A. (Decision) L(A) = { i/p x / Ans is YES for x }

→ ✓Problem ... $\underline{L(A)}$ = { i/p ... }
Decision

e.g. A = Independent set $\quad L(I.S.) = \{\langle G, K\rangle \mid G \text{ has I.S. of size} \geq K\}$

> Informal:
> ✓Problem A is decidable if ∃ finite time alg'm to solve it. o.w. un-decidable.
> Decision

Def: L is decidable, iff $\underline{\exists \text{ TM } M}$ (≡ Alg'm ≡ C.P.)

Accept = YES
Reject = NO.

s.t.

$x \in L \Rightarrow M$ accepts $\underline{x}$

$x \notin L \Rightarrow M$ rejects x.

Otherwise, L is un-decidable.

(Informal) Halting Problem: Given an Alg'm check if it halts on every i/p.

(Formal) TM-Halt = $\{\langle M\rangle \mid M$ halts on every i/p$\}$

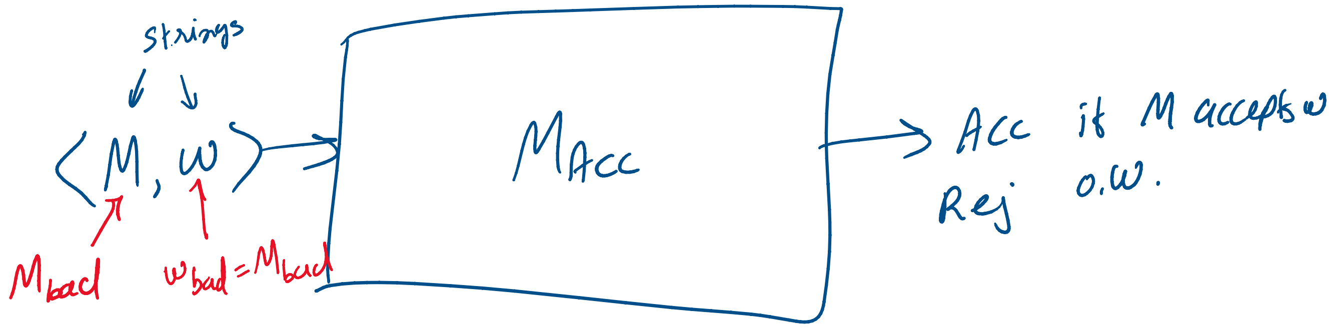
TM-ACC = $\{(\underline{M}, \underline{w}) \mid M$ accepts $w\}$.

(1936): TM-ACC is $\underline{\text{undecidable}}$.

Turing's Thm (1936): TM-ACC is undecidable.

> Informal: Given C.P. M & it's i/p w
> $\nexists$ Alg'm to check if M accepts w.

Pf: To the contrary, Suppose TM-ACC is decidable by TM M_{ACC}.



Construct a counter example $\langle M_{bad}, w_{bad} \rangle$:

☆ "$M_{bad}(x)$ {

(x = M_{bad})

1. M = description of a machine given by x. ($M = M_{bad}$)

2. If $M_{Acc}(M, \langle M \rangle)$ = Accept then o/p Reject (M = "M_{bad}", $\langle M \rangle$ = "M_{bad}")

3. Else o/p Accept.

}"

☆ w_{bad} = description of M_{bad}. $= \langle M_{bad} \rangle$

case I: $M_{Acc}(M_{bad}, \langle M_{bad} \rangle)$ = Accept.

case I: $M_{ACC}(M_{bad}, \langle M_{bad} \rangle) = \text{Accept.}$

$\langle M_{bad} \rangle = w_{bad}$

Then M_{bad} will reject i/p $w_{bad} = \langle M_{bad} \rangle$.

WRONG!

case II: $M_{ACC}(M_{bad}, \langle M_{bad} \rangle) = \text{Reject.}$

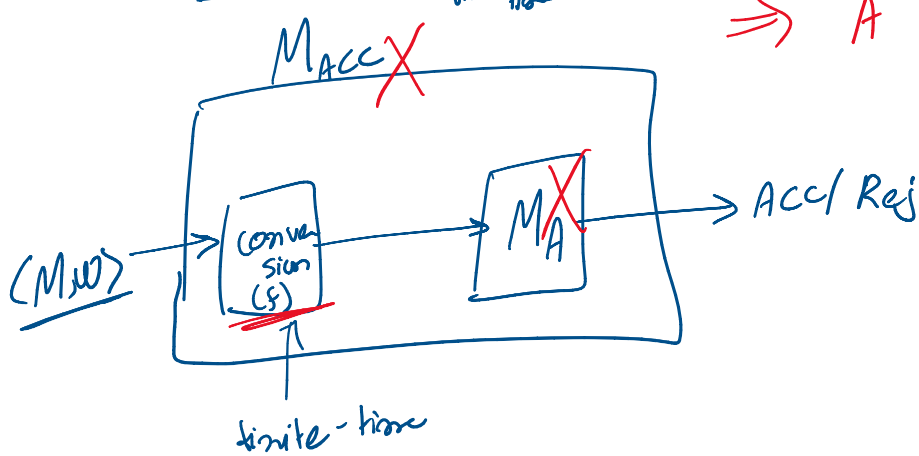
$\langle M_{bad} \rangle = w_{bad}$

Then M_{bad} will Accept i/p w_{bad}

WRONG!

CONTRADICTION to M_{ACC} deciding TM-ACC ∎

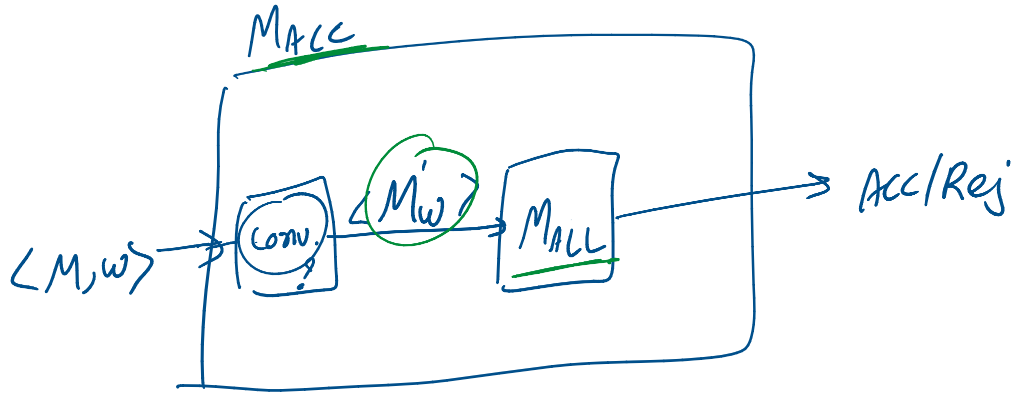
TM-ACC $\leq_{\text{finite-time}}$ A $\Rightarrow$ A is undecidable.



Ex1: TM-ACC-ALL $\{ \langle M \rangle \mid M$ accept all i/p $(L(M) = \Sigma^*) \}$

Claim TM-ACC $\leq$ TM-ACC-ALL $\Rightarrow$ TM-ACC-ALL is undecidable.

Pf :



★ Conversion func (M, w) :

1. Construct " $M'_w(x)$ {

   1. If $M(w)$ = accept then o/p accept

   2. Else o/p reject

   }"

★ Correctness pf :

M_{Acc} Accepts $\langle M, w \rangle$

$\Leftrightarrow$ M_{ALL} " $\langle M'_w \rangle$

$\Leftrightarrow$ $L(M'_w) = \Sigma^*$

$\Leftrightarrow$ M Accepts w.

Ex 2 : TM-ACC-SOME = $\{\langle M \rangle \mid M$ accepts some i/p$\}$

Same proof as EX1.

Ex 3 : TM-ACC-NONE = $\{\langle M \rangle \mid M$ does not accept any i/p$\}$

Ex 3: TM-ACC-NONE = { ⟨M⟩ | ... any I/P }

Use: closed under complements.

Ex 4:

$$\text{TM-Equiv} = \{ \langle M_1, M_2 \rangle \mid L(M_1) = L(M_2) \}$$

where $L(M_1) = \{x \mid M_1 \text{ accepts } x\}$ and $L(M_2) = \{x \mid M_2 \text{ accepts } x\}$

Pf: TM-ACC-ALL $\leq$ TM-Equiv.

TPT: $\langle M \rangle \longrightarrow \langle M_1, M_2 \rangle$

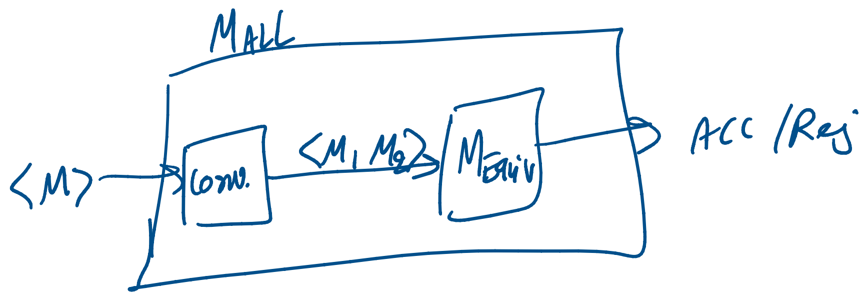
s.t. $L(M) = \Sigma^* \iff L(M_1) = L(M_2)$.

Construct func(M):

① $M_1 = M$.

② $M_2 =$ " run(x) { accept x; } ".

$L(M_2) = \Sigma^*$



★ Correctness Pf: M_{ALL} Accepts ⟨M⟩

$\iff M_{Equiv}$ accepts $\langle M_1, M_2 \rangle$ (where $M_1 = M$)

$\iff L(M_1) = L(M_2)$

$\Leftrightarrow \quad L(M_1) = L(M_2)$

$\Leftrightarrow \quad L(M) = \Sigma^*$ $\quad (\because M_1 = M)$

Exs: $M_{Reg} = \{ \langle M \rangle \mid L(M)$ is ~~regular~~ languages $\}$

P

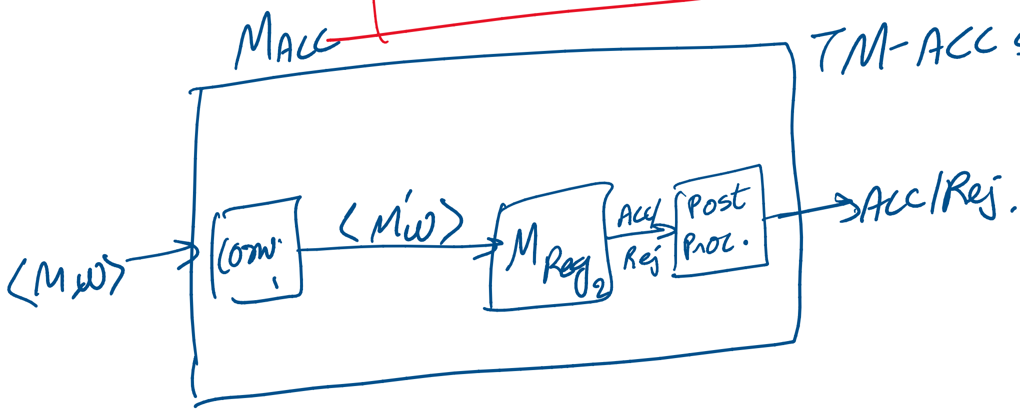
Undecidable.

$L = \phi$ = regular

~~$L = \{palindroms\}$ not regular~~

$TM\text{-}ACC \leq TM\text{-}Reg.$

Pf:



★ Conversion func $(\underline{M}, \underline{w})$:

Construct M'_w : "$M'_w(x)$ {

1. If $M(w)$ = accept then
2. If x is a palindrom then Accept; return;

3. Reject.

}"

Claim: If M accepts $w \Rightarrow L(M'_w) = \{x \mid x$ is a palindrom$\}$ ← not regular

Case I: If M accepts w ⇒ $L(M'_w) = \{x \mid x$ is a palindrome$\}$

Case II: M rej/does not halt on w ⇒ $L(M'_w) = \emptyset$ ← regular.

★ Post processing: If $M_{Reg}(M'_w)$ = Accept then o/p Reject
Else " Accept.

Correctness pf:

M_{ACC} Accepts $\langle M, w \rangle$

⟺ M_{Reg} Rejects $\langle M'_w \rangle$

⟺ $L(M'_w)$ is not regular

⟺ $L(M'_w) = \{x \mid x$ palindrom$\}$

⟺ M accepts w.

Rice's Thm (1950's): P : Any non-trivial property about languages.

Informally Non trivial ≡ ∃ decidable language w/ P
∃ " " w/o P.

Then, $TM_P = \{\langle M \rangle \mid L(M)$ satisfies P$\}$

is undecidable.

eg: D: |L| is 374.

Ex 6: P = $|L|$ is 374.

$\{\langle M\rangle \mid \langle M\rangle$ accepts 374 strings$\}$ is undecidable

★ What is decidable?

$\{\langle M\rangle \mid M$ halts in 2024 steps$\}$ is decidable

(just run M for 2024 steps)

★ Other un-decidable problems:

→ "Does there exist integer solution to a given polynomial?"

e.g. $x^7 + y^7 = z^7 + 10$.

(Hilbert's 10th Thm.)

→ $\{\langle G\rangle \mid L(G) = \Sigma^*$ for CFG $G\}$.