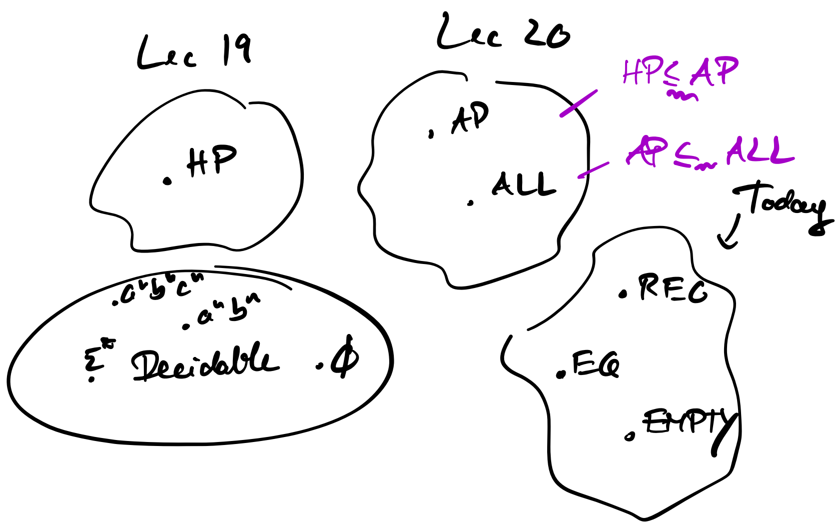
Comp 330 - Lecture 21 - Nov 14^{th}

IE : Tutto fa brodo
Everything makes broth / soup

Showing undecidability via REDUCTIONs (2)
+ Characterizing undecidability

Recall



Reductions D.P.s P & Q $\quad P \leq_m Q$
if I can solve Q, I can also solve P.

$P \leq_m Q$
Showing undecidability : P undecidable $\Rightarrow$ Q undecidable

Formally, $P \leq_m Q \iff \exists$ a mapping$^{\text{reduction}}$ function

f s.t. $\forall x \in \Sigma^*$ $\quad \underbrace{x \in L_P \Leftrightarrow f(x) \in L_Q}$

Must be computable

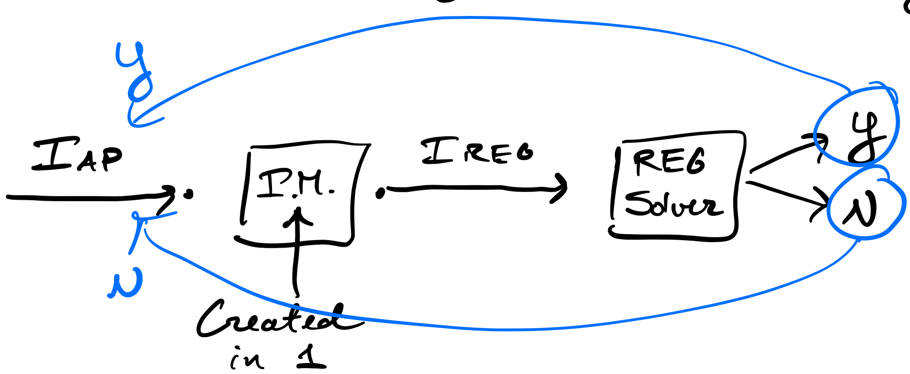
$\Downarrow$

1. $I_P \to I_Q$
2. $ANS(I_P) = Y \Leftrightarrow$ $\underbrace{ANS(I_Q)}_{\text{Answer of } I_Q} = Y$

Ex1 Show REG is undecidable by showing that $AP \leq_m REG$.

$REG(M)$: "Given a TM M, is $L(M)$ regular?" $\quad \Sigma = \{0,1\}$

1. Convert $I_{AP} = \langle M, x \rangle$ to $I_{REG} = \langle N \rangle$ $\Big\}$ Instance modification
2. $ANS(I_{AP}) = Y \Leftrightarrow ANS(I_{REG}) = Y$



AP Solver

f will always have the form

$v \in \{0,1\}^*$

$$f(v) = \begin{cases} I.M.(v) & \text{if } v = I \wedge P \\ \text{Invalid instance of the REG} & \text{otherwise} \end{cases}$$

1. $\langle M, x \rangle \rightarrow \langle N \rangle$

M accepts x $\iff$ $L(N)$ is regular

$N :=$ On input w

→ If $w = a^n b^n$ then accept

Erase w & load x

Run M on x

If M accepts x then accept

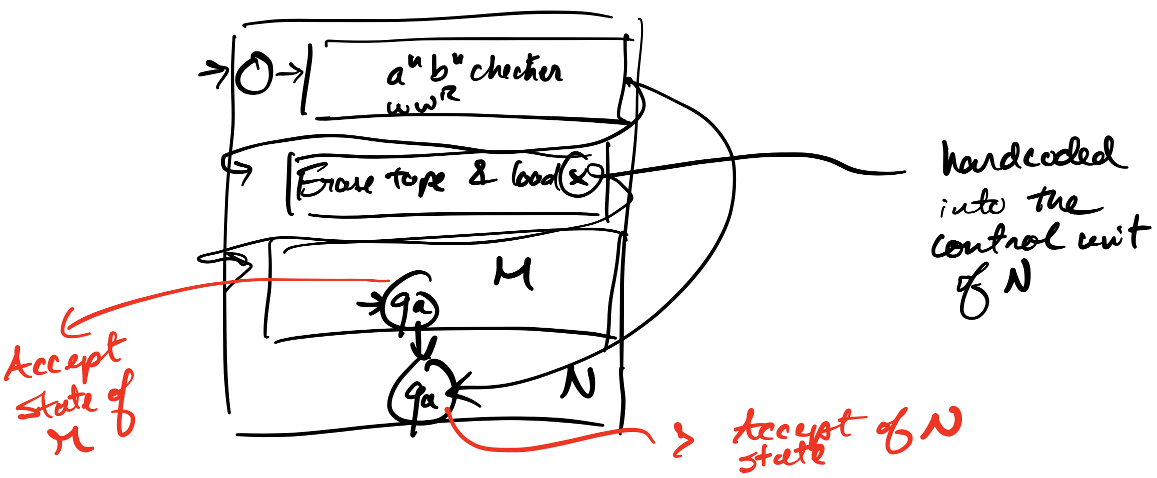
M accepts x → $L(N) = \Sigma^*$ is regular

M does NOT accept x → ~~$L(N) = \emptyset$~~ is also regular

$L(N) = \{a^n b^n\}$

$\boxed{\text{Typo in lecture}}$

The instance modification of $\langle M, x \rangle$ to $\langle N \rangle$ was computable.



$ANS(I_{AP}) = Y$ ↓

2. M accepts x ⇒ Regardless of w, N accepts $L(N) = \Sigma^*$
⇒ $ANS(I_{REG}) = Y$

$ANS(I_{AP}) = N$

↳ M does not accept x ⇒ $L(N) = \{a^n b^n\}$ which is not regular
∴ $ANS(I_{REG}) = N$

Expectation $X \leq_m Y$ for A6 / Final

1. Give the IM and argue that it's computable.
2. $ANS(X) = \text{Yes} \iff ANS(Y) = \text{Yes}$.

<u>Exercises</u> Reduce from HP or AP

1. "Given a TM M, is $0^{330} \in L(M)$?"
2. ——"——, is $L(M) \neq \emptyset$?"
3. ——"——, is $L(M)$ is context-free?"

<u>Ex 2</u> Show that the following DP is undecidable

EQ(M_1, M_2): "Given TMs M_1 & M_2, is $L(M_1) = L(M_2)$?"

$ALL \leq_m EQ$.

1. Convert $I_{ALL} = \langle M \rangle$ to $I_{EQ} = \langle M_1, M_2 \rangle$

2. $ANS(I_{ALL}) = \text{Yes} \iff ANS(I_{EQ}) = \text{Yes}$.

$M_1 := M$ $\qquad\qquad$ $M_2 :=$ On input w

1. Accept

$$ANS(I_{ALL}) = Y \Rightarrow L(M) = \Sigma^*$$

$$L(M_1) = L(M) = \Sigma^*$$

$$L(M_1) = L(M_2) = \Sigma^*$$

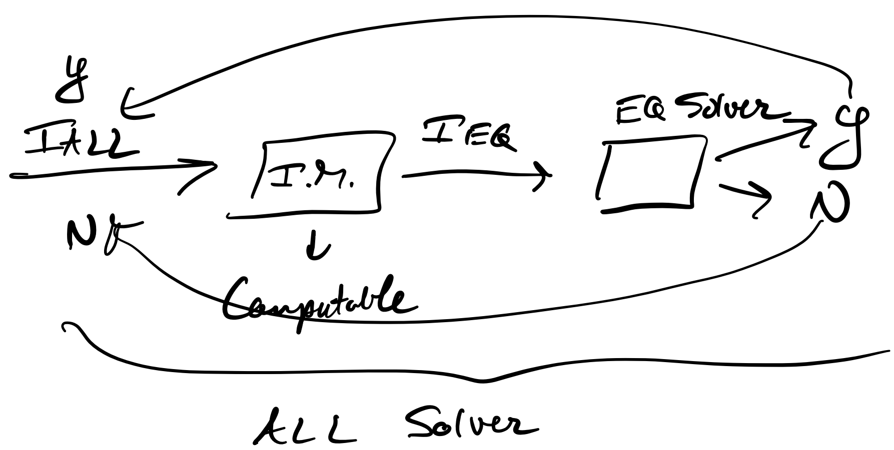
$$\Rightarrow ANS(I_{EQ}) = Y$$

$$ANS(I_{ALL}) = N \Rightarrow L(M) \neq \Sigma^*$$

$$L(M_1) = L(M) \neq \Sigma^*$$

$$L(M_1) \neq L(M_2) = \Sigma^*$$

$$\Rightarrow ANS(I_{EQ}) = N$$



ALL $\leq_m$ EQ since ALL is undecidable so is EQ

Remark 1 $\underbrace{AP \leq_m ALL}_{\text{Lec 20}}$ $\underbrace{ALL \leq_m EQ}_{\text{Now}}$

$\Rightarrow AP \leq_m EQ$

In general A, B, C $A \leq_m B$, $B \leq_m C$

Then $A \leq_m C$

Pf Omit. Composition of comp map reductions is a comp. map. reduction.

Remark 2 If L is decidable then so is $\overline{L}$.

Pf L is decidable $\Rightarrow \exists M$ s.t. $L(M) = L$ & M halts on every input

Create M' s.t. $L(M') = \overline{L}$

$q_a' = q_r$ $q_r' := q_a$

accept of M' reject of M

$x \in L(M') \iff \vdash s\, x \square \xrightarrow{*} \vdash u\, q_a'\, v \square$

$$\iff \vdash s\, x \sqcup \xrightarrow{*} \vdash u\, q_r\, v \sqcup$$

$$\iff x \notin L(M)$$

$$\iff x \in \overline{L(M)} \iff x \in \overline{L}. \qquad \square$$

Implication: If L is undecidable then so is $\overline{L}$.

Useful because in some cases for L_1, L_2 it is impossible to create f s.t. $L_1 \le_m L_2$ but it is possible for $\overline{L_1} \le_m L_2$.

Ex3 Show EMPTY is undecidable

EMPTY(M) : " Given a TM M, is $L(M) = \emptyset$?"

Tempting : $HP \not\le_m EMPTY$ → Impossible, we will see in Lec 22

HP: Not co-CE

EMPTY: is co-CE

$$\overline{HP} \le_m EMPTY$$

$\overline{HP}(M, w)$: " Given TM M & input string w, does M not halt on w?"
does M loop on w?"

1. $I_{\overline{HP}} = \langle M, x \rangle \rightarrow I_{EMPTY} = \langle N \rangle$

M loops on x $\iff L(N) = \phi$

Common error

N := On input w
1. Erase w & load x
2. Run M on x
3. If M halts on x then accept
3. If M loops on x then reject. → Common error

On input x

Overloading notation / Need to create a TM which accepts any input.

2. ↗ M loops on x ⇒ Regardless of w,
Y N does not accept,
$L(N) = \phi \rightarrow Y$.

N → M halts on x ⇒ Regardless of w, N accepts,
$L(N) = \Sigma^* \neq \phi \rightarrow N$

This inability to do $\overline{HP} \leq_m$ EMPTY is evidence that there is more subtlety to undecidability.

Characterizing undecidability

Recall A language L is Turing recognizable if $\exists$ a TM M s.t. $L(M) = L$. There is no guarantee that M halts if $x \notin L$.

Clearly $$L_{DEC} \subseteq L_{REC}$$

$$L_{DEC} \subsetneq L_{REC}$$

Proper subset

What's an example of a language which is undecidable but Turing recognizable?

computably enumerable

Claim L_{HP} is Turing recognizable

Intuition: If you have a yes-instance then you can check. $L(N) = L_{HP}$

Design a TM N s.t. $L(N) = L_{HP}$ as follows:

$N :=$ On input w

1. Check if $w = I_{HP} = \langle M, x \rangle$
2. If no, reject
3. Otherwise, run M on x

4. If M halts on x, then accept

by construction of N

<u>Claim</u> $L(N) = L_{HP}$ $\quad w \in L(N) \iff$

$w = \langle M, x \rangle$, M is a TM, x is an input string, M halts on x. $\iff$

$w \in L_{HP}$

What's the point? Further characterize undecidable ~~pro~~ problems

HP : Undecidable but you can check $\underbrace{\text{yes-instances}}_{\text{T.R. / C.E.}}$

EMPTY : Undecidable but you can check $\underbrace{\text{no-instances}}_{\text{co-T.R. / co-C.E.}}$

EQ : Undecidable but you can neither check a yes-instance $\underbrace{\text{nor a no-instance}}_{\text{neither C.E. nor co-C.E.}}$

Can you do $EMPTY \leq_m HP$? No! Lec 22.