CS & IT ENGINEERING

Theory of Computation

Decidability

Lecture No. 2

By- DEVA Sir

TOPICS TO BE COVERED

01 Membership

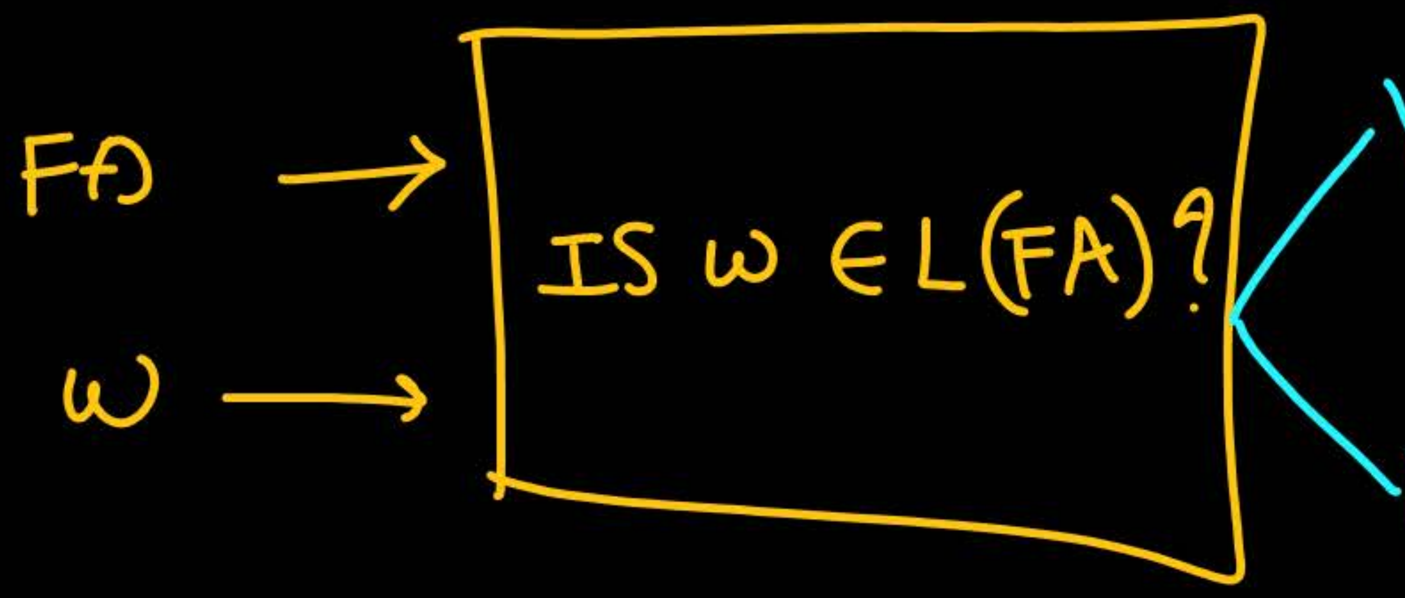
02 Problems

03 Languages

04 Reducibility

05

Membership for FA/Reg lang/RgExp/RG



Logic exist

Yes : $w \in L(FA)$: FA halts at final

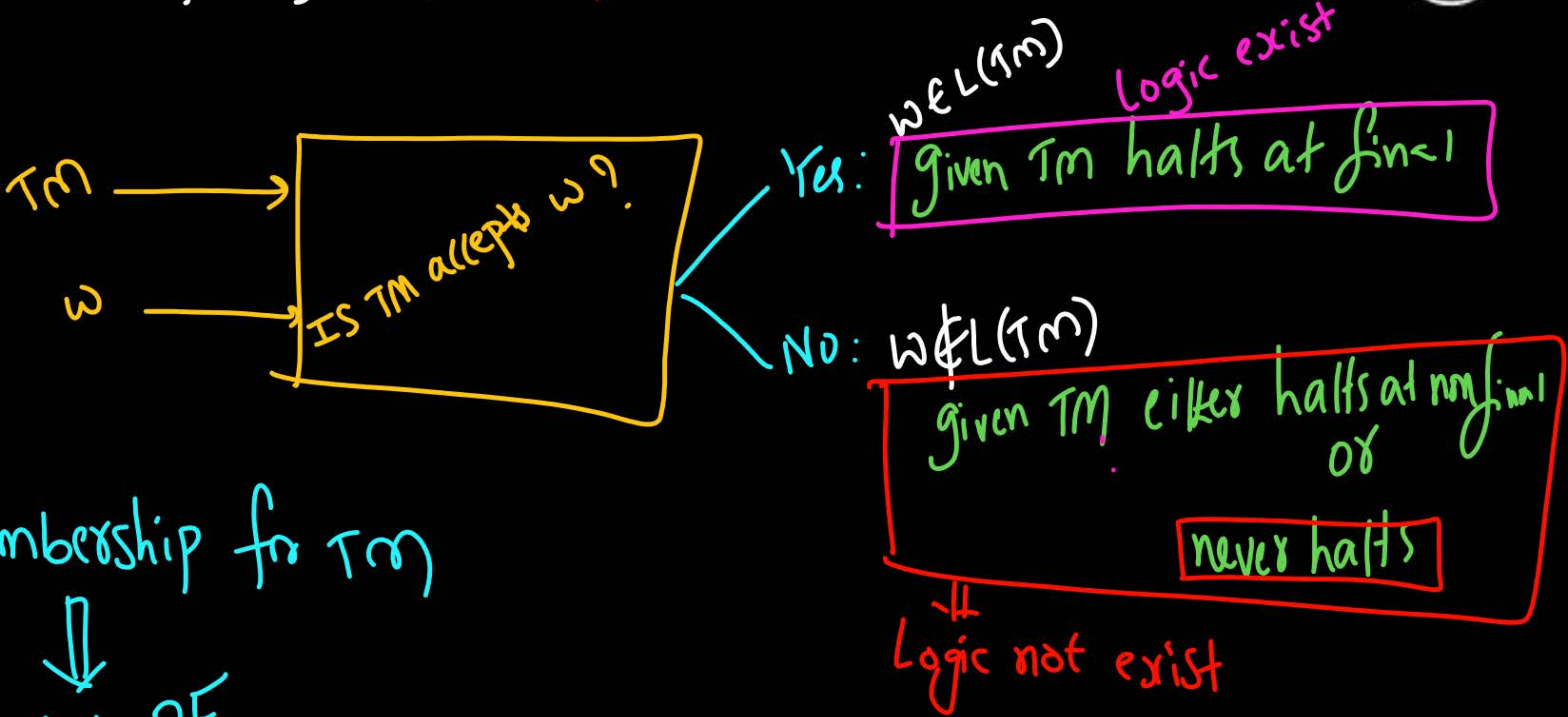
No : $w \notin L(FA)$: FA halts at non final

Logic exist

Similarly,

Membership problem is decidable for FA/DPDA/PDA/LBA/HTM.

Membership for TM :⇒ SDUD



Non membership for TM

⇓

Not RE

problems:

① IS FA accepts ε ? ⇒ Membership for FA ⇒ Decidable

~~A)~~ Recursive
~~B)~~ RE
c) Undecidable

② IS PDA accepts ε ? ⇒ D

③ IS TM accepts ε ? ⇒ SDUD

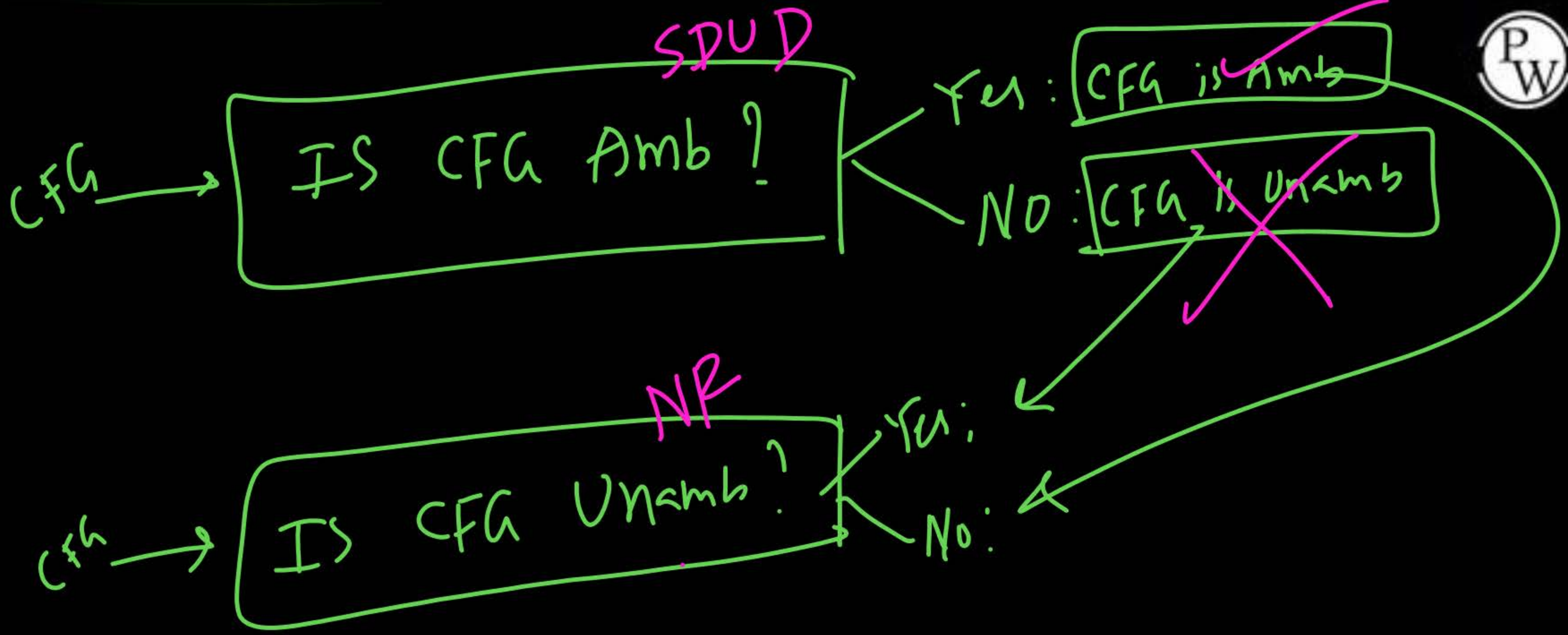
④ IS $FA_1 \cong FA_2$? ⇒ D

UD ⑤ IS CFG Ambiguous ? ⇒ SDUD

- Yes : $\overbrace{\text{CFG is Amb}}^{\text{some string, >1 PT}}$ ⇒ logic exist
- No : $\underbrace{\text{CFG is Unamb}}_{\text{every string, 1 PT}}$ ⇒ logic not exist

⑥ IS grammar CFG ? ⇒ Decidable

UD ⑦ IS CFG Unambigous ? ⇒ Not RE ⟵ ~~⑤~~


SDUD
CFG
IS CFG Amb ?
Yes : CFG is Amb
NO : CFG is Unamb
NR
CFG
IS CFG Unamb ?
Yes :
No :

What is given?

⇓

Yes : meaning

NO : meaning

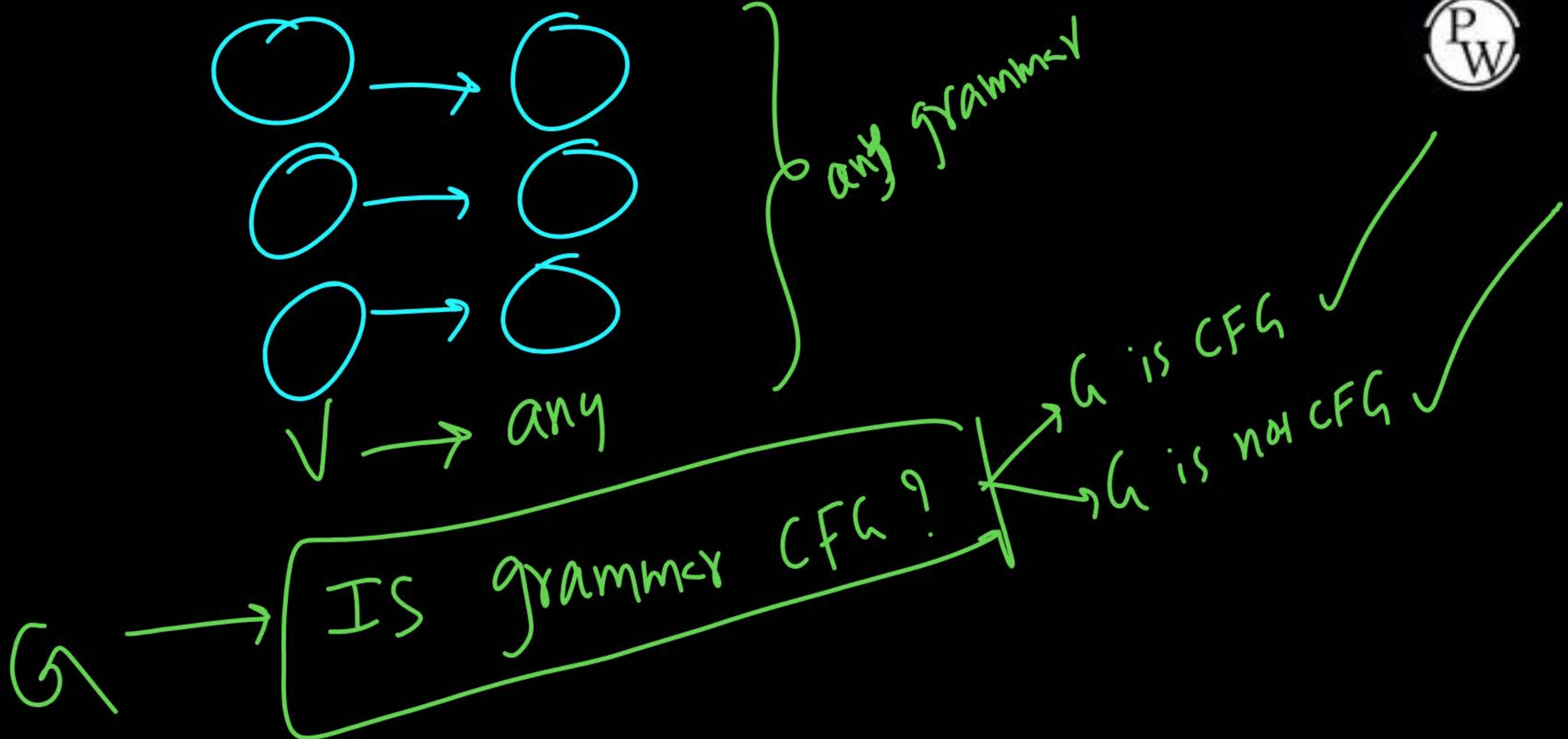
⇓

If answer Yes : check logic exist or not

If Yes has logic, then check for whether No has logic or not

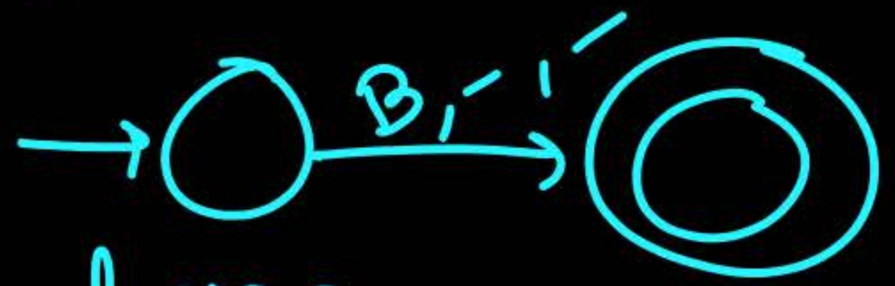
⇓

✓


any grammer
V → any
G
Is grammer CFG ?
G is CFG ✓
G is not CFG ✓

I) Can we design Tm that accepts ε ?

⇓



We can design Tm that accept ε

II) Can we verify arbitrary TM that accepts ε ?

→ Yes ⇒ logic ✓

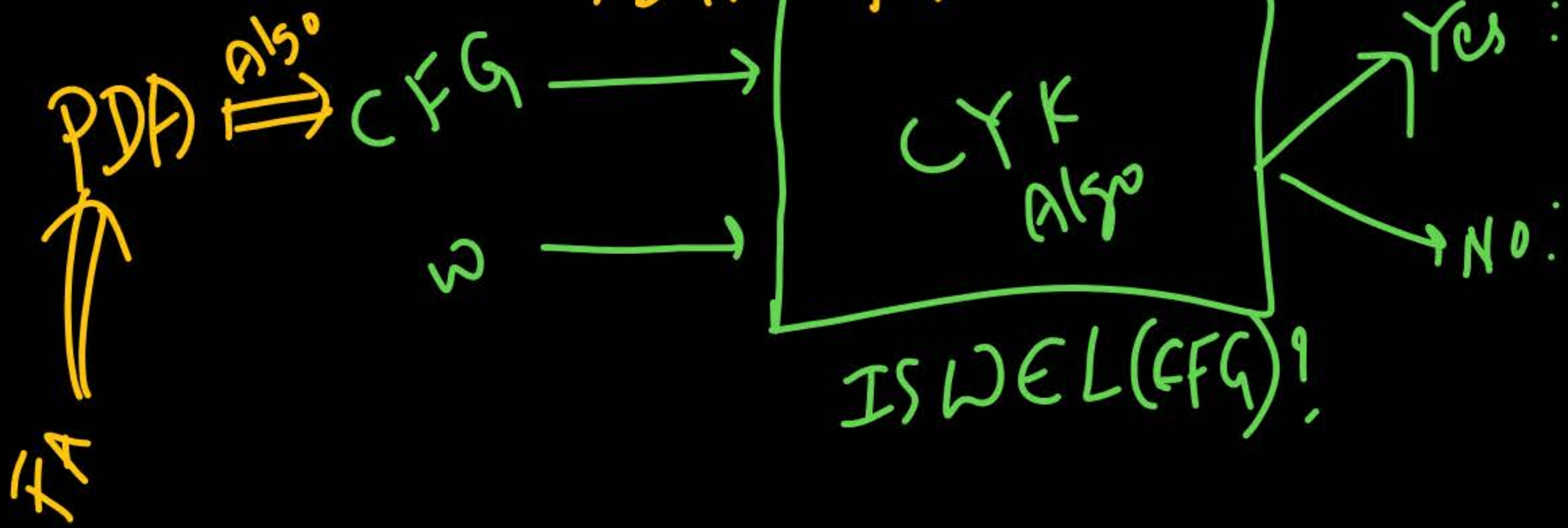
→ No ⇒ logic ✗

↓

Undecidable & Semidecidable

CYK Algo [Membership Algo]

- → $O(n^3)$, Dynamic programming
- → Bottom up parsing



Languages:

① $\{ FA \mid \underbrace{FA \text{ accepts } \varepsilon}\ \} \rightarrow D$

$L = \{ FA_1, FA_2, \ldots \}$

$\overline{L} = \{ FA_1', FA_2', \ldots \}$

$\underbrace{\text{IS FA accepts } \varepsilon}_{\text{membership}}$?

$\Downarrow$

D.

② $\{\langle FA, w\rangle \mid FA \text{ accepts } w\}$? ⇒ Decidable

③ $\{FA\#w \mid FA \text{ accepts } w\}$

④ $\{TM \mid TM \text{ accepts } \boxed{\varepsilon}\}$ Is Tm accepts ε ? ⇒ SDUD

Tm	#	w

⑤ $\{Tm \mid Tm \text{ accepts } \boxed{w}\}$

⑥ $\{Tm \mid Tm \text{ accepts some string}\}$ ⇒ SDUD

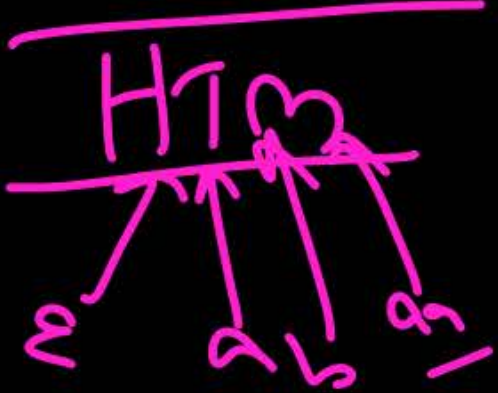
⑦ $\{Tm \mid Tm \text{ accepts only } w\}$

⑥ Is TM accepts some string ? ⇒ SDUD

Is Tm accepts something ?

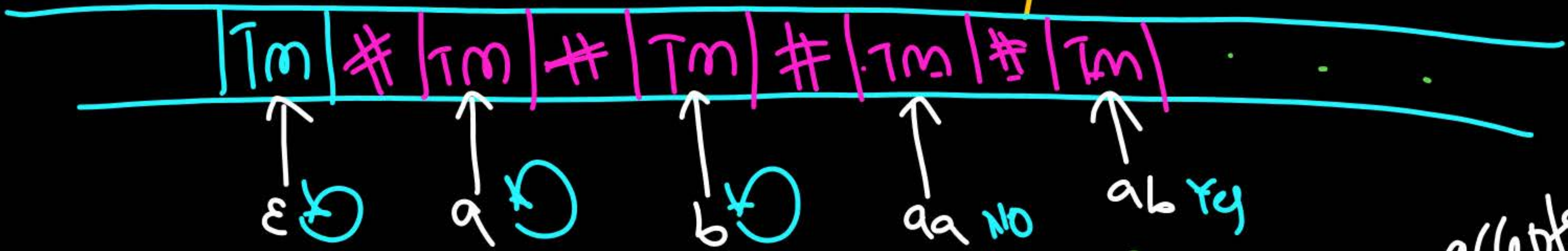
Is $L(Tm) \neq \phi$?

Non emptiness



Yes : Tm accepts some string
Logic exist

No : Tm accepts nothing
Logic not exist



atleast one Tm halts at final ⇒ Tm accepts something

⑦ Is Tm accepts only w ?

Is $L(Tm) = \{w\}$?

↓

NR

Yes : Tm is accepting only w

No : $L(Tm) \neq \{w\}$

Tm not only accepts w

<u>Yes</u> : $L(Tm) = \{w\}$

→ $\underbrace{\text{Tm should accept } w \text{ and also should not accept everyone other than } w}_{\text{logic not exist}}$

No : $L(Tm) \neq \{w\}$

→ $\underbrace{\text{Tm may accept } \phi}_{\text{not logic exist}}$ or accepts someone other than w

⑧ $\{ \text{Tm} \mid \underbrace{\text{Tm accepts 3 length string}} \} \Rightarrow$ SDUD

⑨ $\{ \text{Tm} \mid \text{Tm reaches state } q \} \Rightarrow$ SDUD

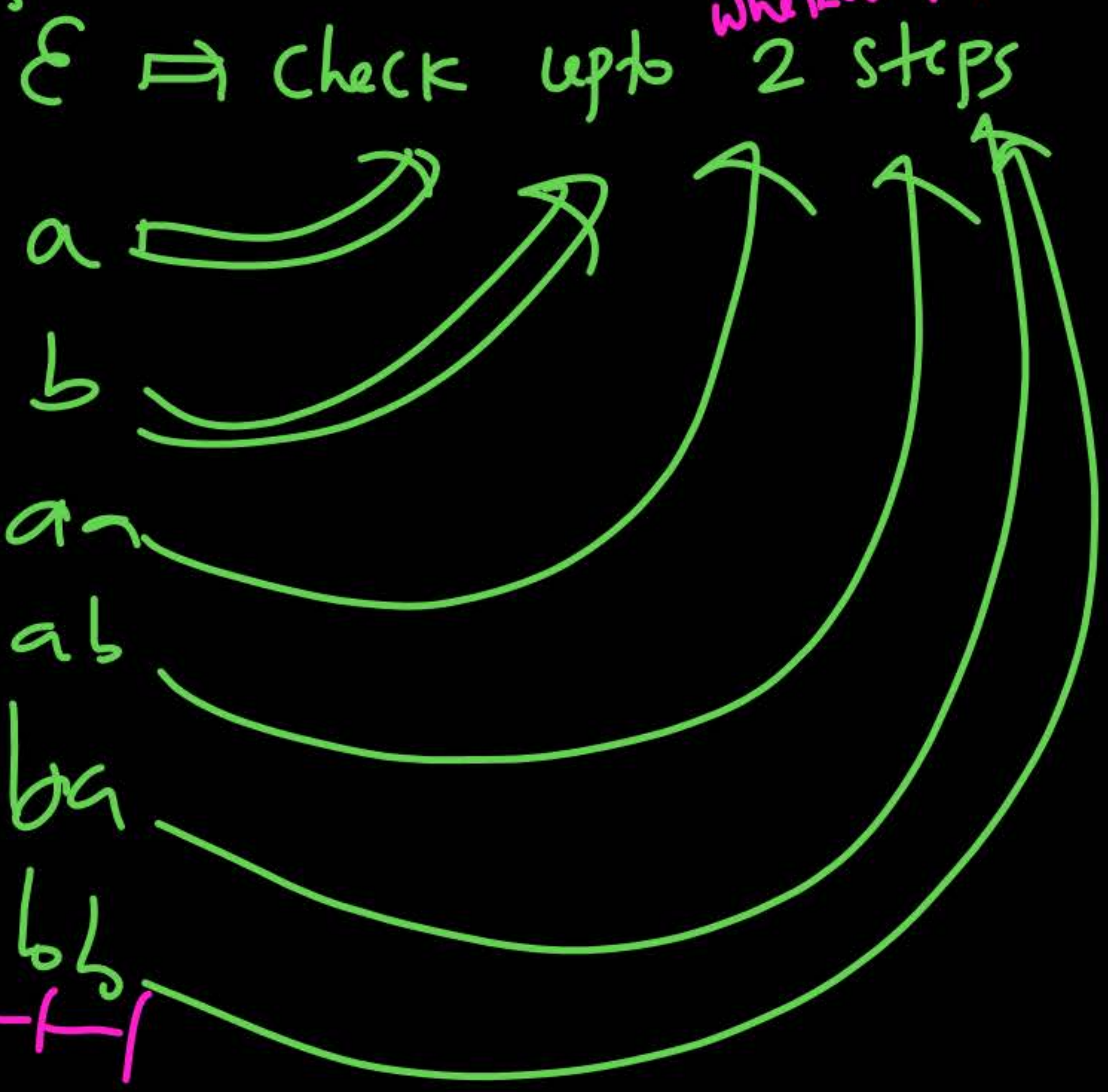
⑩ $\{ \text{Tm} \mid \text{Tm reaches state } q \; \underbrace{\text{within 5 steps}} \} \Rightarrow$ Decidable

Is Tm reaches state q within 2 steps (moves) ?

Always halts

I) we will check only upto 2 length strings.

II) on every string, run Tm upto 2 steps only



ε ⇒ Check upto 2 steps whether q is reached

a

b

aa

ab

ba

bb

Yes : For some string, (upto 2 length string) q is reachable within 2 steps

No : For every string, (upto 2 len) q is not reachable upto 2 steps

IS Tm accepts 3 length string ?.

Logic exist

Yes

Tm accepting 3 length string.

NO

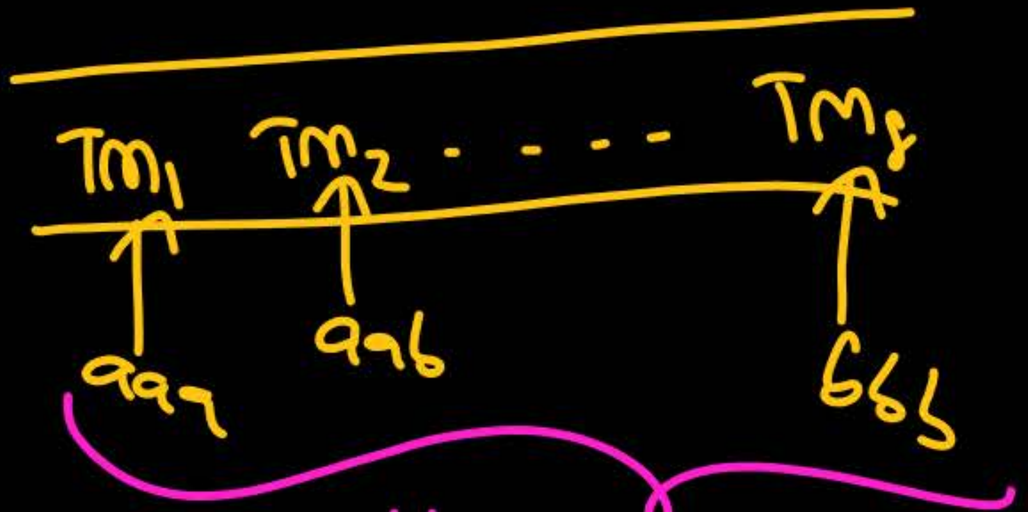
Tm doesn't accept 3 length string

logic not exist

$\Sigma=\{a,b\}$

⇓

$\underbrace{x\ y\ z}_{8 \text{ string}}$



atleast one Tm halts at final

D 11 $L = a^* b^*$

D 12 Finite Language

D 13 Regular Language

D 14 DCFL

D 15 CFL

D 16 CSL

D 17 Recursive language

18 REL → D or SDUD

19 REL but not recursive → SDUD

20 Not REL → NR

21 Not Recursive → NR or SDUD

22 Not Reg language → D, or SDUD, or NR

23 Not CFL → D, or SDUD, or NR

24 $a^n b^n$ → D

D

SDUD

NR

***

SDUD

(25) SDUD Set of (all) finite languages

(26) Set of (all) regular languages

(27) Set of DCFLs

(28) Set of CFLs

(29) Set of CSLs

(30) Set of Decidable languages

(31) Set of RELs

(32) Set of SDUDs

$\overline{(32)}$ = NR

NR (33) Set of not RELs

(34) Set of infinite languages NR

(35) $\underbrace{\underbrace{\{L \mid L \text{ is Decidable}\}}_{\text{Set of all decidables}} \cup \underbrace{\{L \mid L \text{ is NR}\}}_{\text{Set of all NRs}}}$

SDUD $L = \{F_1, F_2, F_3, F_4, F_5, \ldots\}$

↓ every valid hai logic

$F_i \in L \Rightarrow$ logic exist

$NF_i \notin L \Rightarrow$ logic not exist

NR $\overline{L} = \{NF_1, NF_2, NF_3, \ldots\}$ → logic not exist

NF_1: Reg but nonfi.

↓ logic exist

NF_3: (Not RE)

↓ logic not exist

Reducibility:

$$A \leq B$$

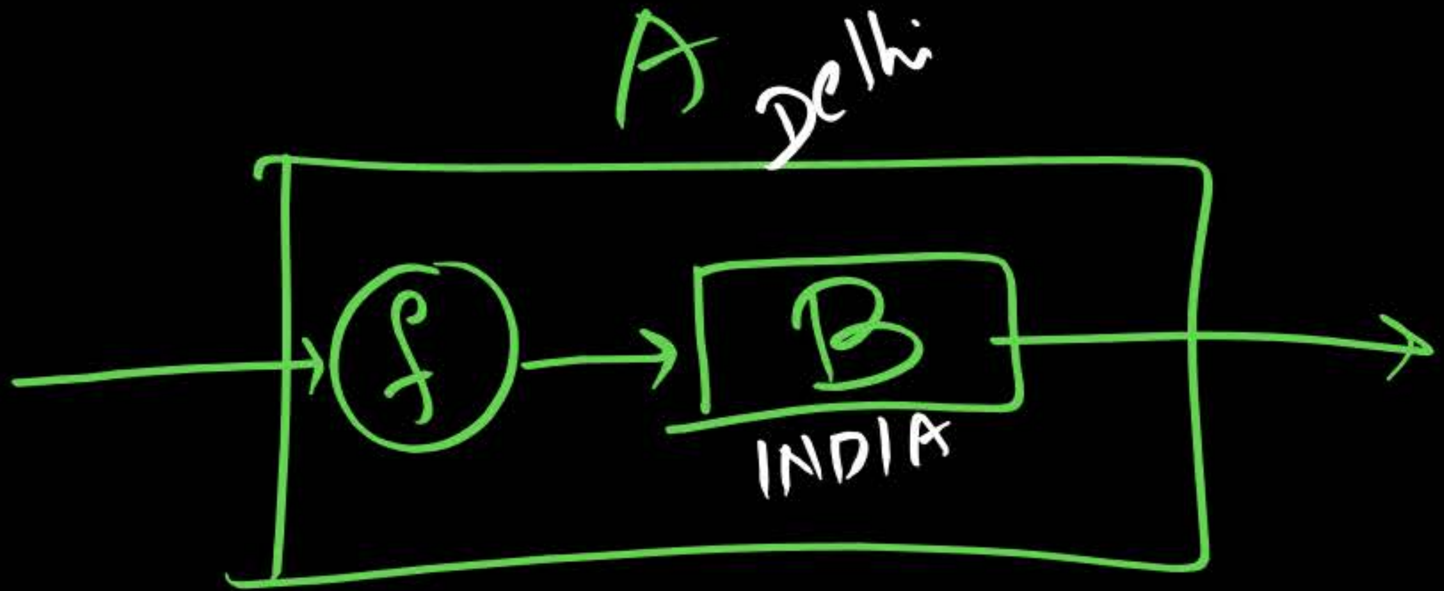
Delhi $\leq$ India

A is Reducibe to B

Every instance of A is mapped into some instance of B

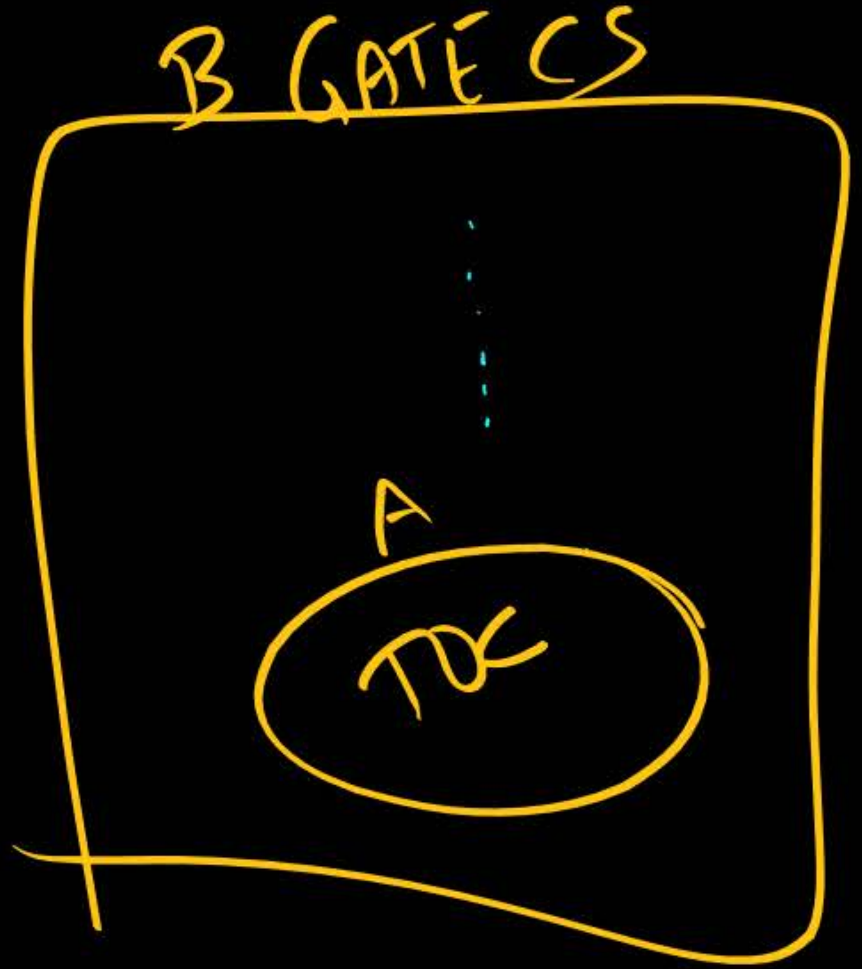
B is equal or harder than A

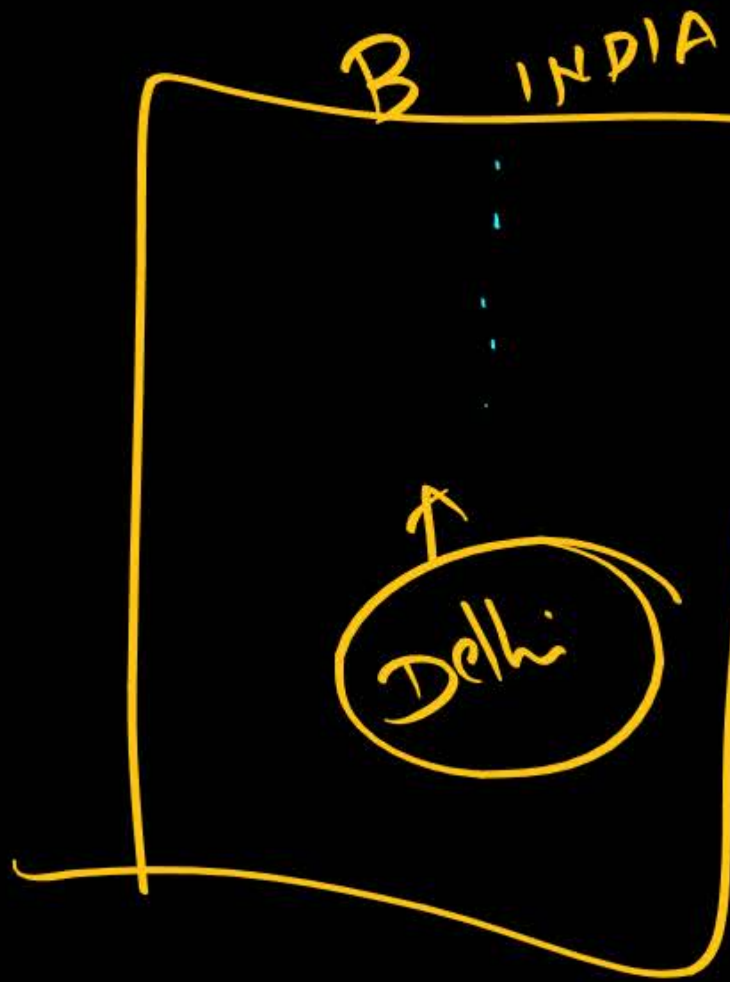
B is atleast as hard as A

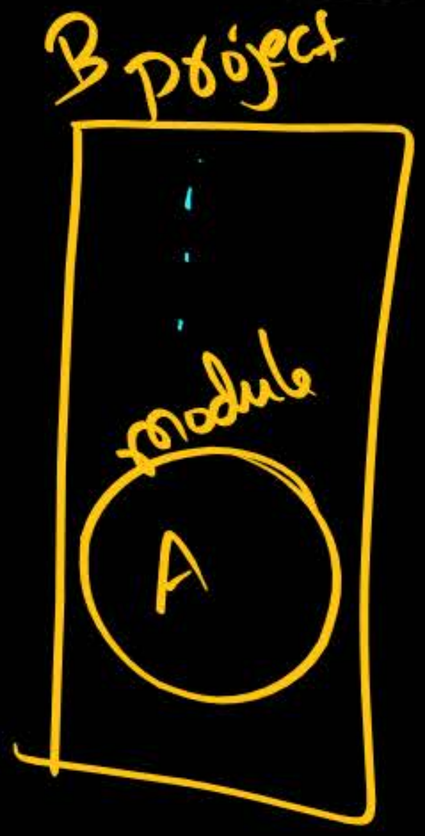


$A \propto_p B$ (A is polynomially reducible to B)

$A \leq_m B$ (A is many to one reducible to B)







$$A \le B$$

Let $\overset{\longleftarrow}{\underset{\longrightarrow}{A \leq B}}$.

1. If A is $\underbrace{\text{undecidable}}_{\text{no halting program}}$ then B is ____ Undecidable

2. If B is Decidable then A is ____ Decidable

3. If B is undecidable then A is ____ ? Decidable or undecidable

4. If A is Decidable then B is ____ ? Decidable or Undec

⑤ If B is REL then A is REL

⑥ If A is not REL then B is not REL

⑦ If A takes 100 days then B takes min 100 days

⑧ If B takes 100 days then A takes max 100 days

$$A \leq B$$

If $A \le B$ & $C \le B$

① If A is Decidable then B is $\underline{\ ?\ }$
C is $\underline{\ ?\ }$

② If B is Decidable then A is $\underline{\text{Decidable}}$
C is $\underline{\text{Decidable}}$

③ If C is Decidable then A is $\underline{\ ?\ }$
B is $\underline{\ ?\ }$

(I) If $A \leq B$ & $B \leq A$

then $A = B$

(II) If $A \leq B \leq C$ then

i) If C is Decidable then A is D
B is D

ii) If A is UD then B is $\underline{\text{UD}}$
C is $\underline{\text{UD}}$

I) L is SDUD $\Rightarrow$ $\bar{L}$ is NR

II) L is Set of all SDUDs $\Rightarrow$ $\bar{L}$ = Set of all Decidables

$\cup$

Set of all NRs

Extra class : [2-4hr]

- Concepts of CFG
  - Simplified CFG
  - Normal forms [CNF & GNF]
  - CYK Algo
- TM concepts
  - Types of TMs
  - Restrictions of TM
- Complexity Theory (P, NP, NPC, NPH) → not in GATE syllabus
- Decidability extra discussion
- GATE PYQs

Summary

→ problems & Languages

Next: extra class (next weekend or weekday)

THANK YOU GW SOLDIERS !