TM

Exercise #1

We have a fixed DFA A. $L(A)$ over $\Sigma = \{0,1\}$

$\underbrace{L(A) \neq \emptyset ;}_{\text{redundant}}$ $L(A) \neq \Sigma^*$. $L(A)$ not finite

$\rightarrow |\emptyset| = 0 \rightarrow$ finite

It is possible that $\varepsilon \in L(A)$

$$P = \{ L \mid L \in RE, \quad \dots \quad L \cap L(A) = \emptyset \}$$

not a language (elements are sets/languages, not strings!)

we call P a class of languages

switch to language

$$L_P = \{ enc(M) \mid L(M) \in P \}$$

language!

a) Prove that $L_\emptyset$ not in REC, using Rice's theorem

left as a homework [deadline Sept. 5th]

b) Prove that $L_\emptyset$ is not in RE. Difficult!

$L_e \notin RE$ then: $L_e \leq_m L_\emptyset$

don't do !!

too difficult

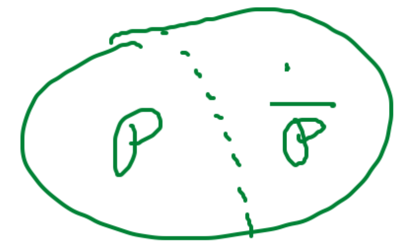
intuition: L_e is very different from $L_\emptyset$ ....

reduction will be very much involved : not good way of attacking problem

<u>Alternative solution</u> : look into $L_{\overline{P}} = \overline{L_P}$

$$\overline{P} = \{ L \mid L \in RE, \ L \cap L(A) \neq \emptyset \}$$ complement of P

RE:



if we could prove $L_{\overline{P}}$ is in RE,
then because $L_{\overline{P}}$ is not in REC (thm)
it follows that L_P not in RE (thm)

we prove <u>this</u>

<u>question</u> → L_P $\qquad \overline{L_P} = L_{\overline{P}}$ ← this is RE \ REC

outside of RE

- $\overline{L_\emptyset} = L_{\bar{\emptyset}} = \{ enc(M) \mid L(M) \cap L(A) \neq \emptyset \}$

We prove $\overline{L_\emptyset}$ is in <u>RE</u>

$\rightarrow$ specify TM M, s.t. $L(M) = \overline{L_\emptyset}$

does not need to halt for string not in $\overline{L_\emptyset}$

<u>TM</u>:

input : $enc(M)$

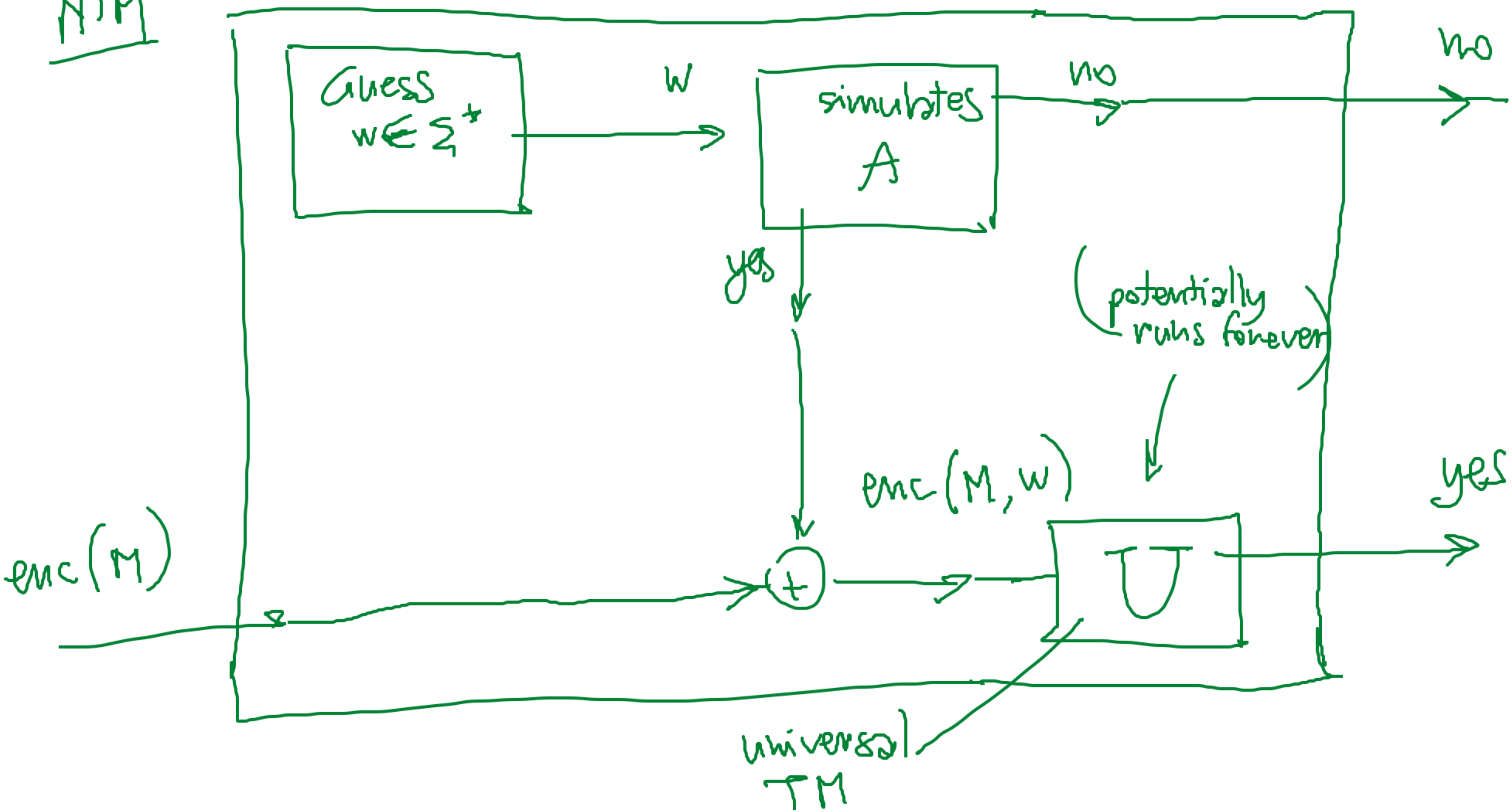
output : yes, if $L(M) \in \overline{L_\emptyset}$

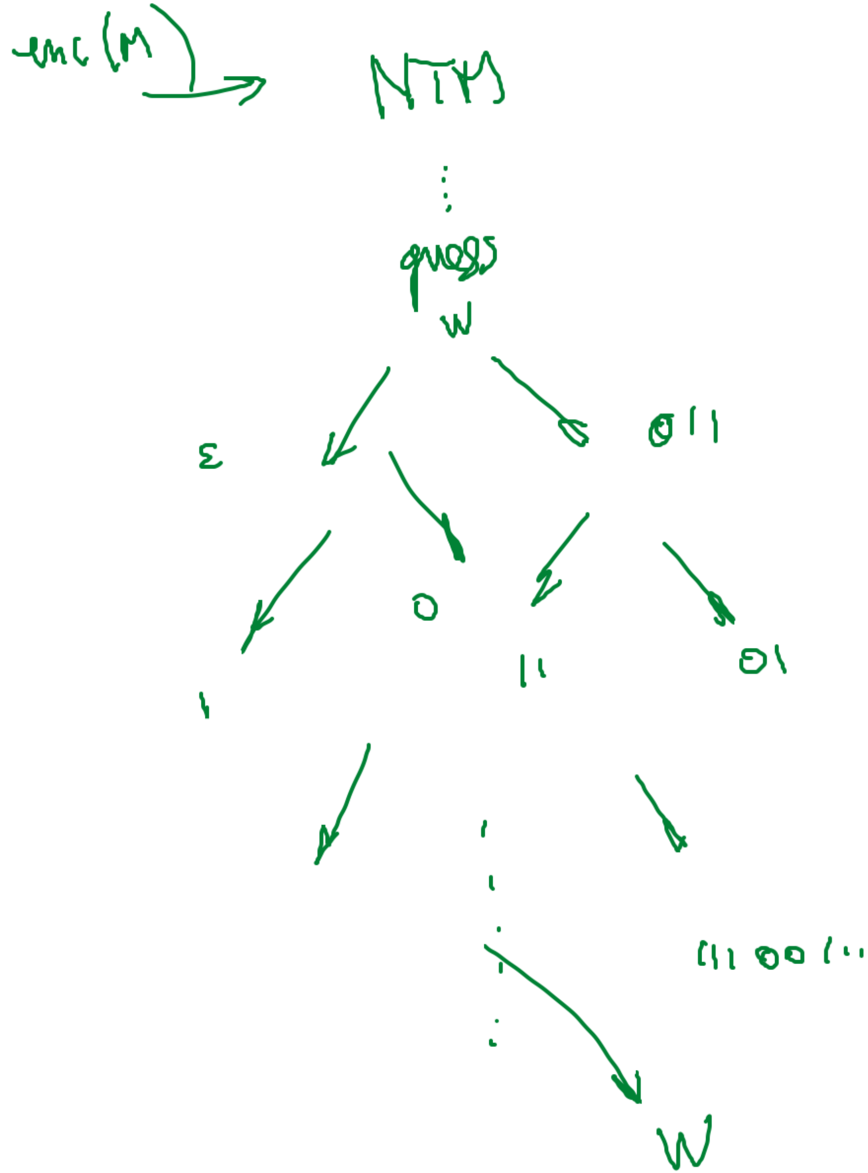
we can answer <u>no</u> or else compute <u>forever</u> otherwise

We use DFA A inside out TM !!

most convenient approach: use <u>NTM</u> ! We later convert into <u>TM</u>

<u>NTM</u>





if $L(M) \cap L(A) \neq \emptyset$

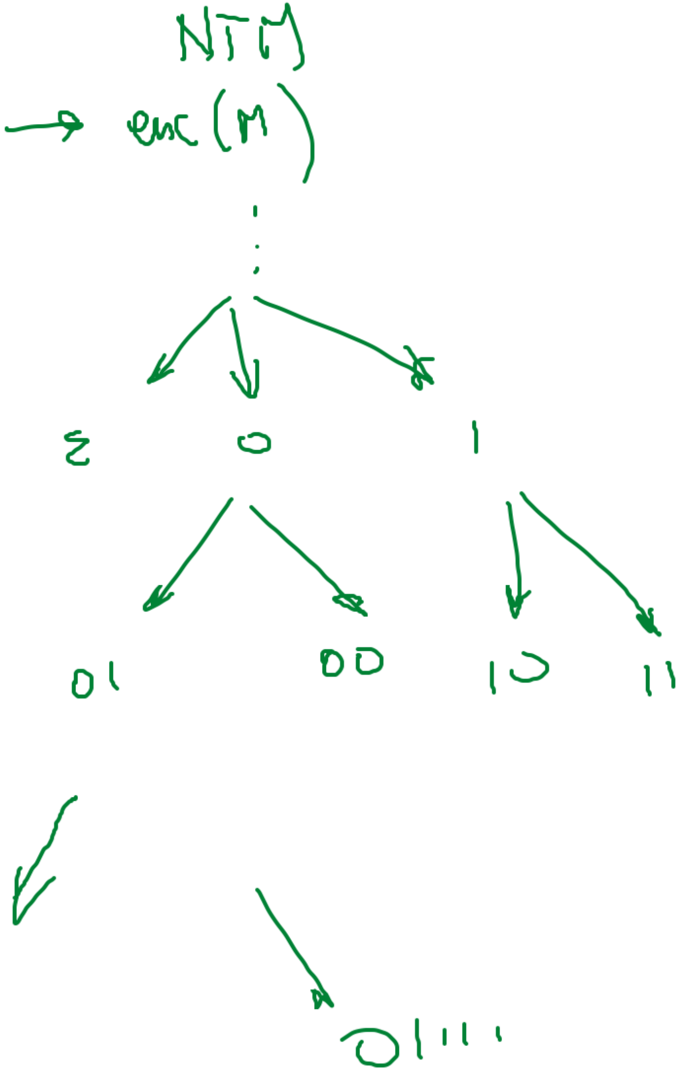
then there exists $w \in \Sigma^*$

such that $w \in L(A)$

and $w \in L(M)$

then the w-computation

for NTM will answer <u>yes</u>



if $L(M) \cap L(A) = \emptyset$ then no string w belongs to both $L(A)$ and $L(M)$

then NTM will not accept in any of its w-computions

We conclude that $L(NTM) = \overline{L_\phi}$ as desired. Next convert NTM into TM M, and we have $L(M) = \overline{L_\phi}$; then $\overline{L_\phi} = L_{\overline{\phi}}$ is in RE

Therefore, L_ϕ cannot be in RE

possible <u>mistakes</u>

- if you get (a) wrong (you prove $L_\emptyset \in REC$)
  then (b) will wrong ($L_\emptyset \in RE$)

- student attempts <u>reduction</u> : you don't have <u>knowledge</u> of languages L s.t. $L \leq_m L_\emptyset$ and $L \notin RE$

  you know : L_e , L_d
  you will not be able to do the reduction (you provide <u>WRONG</u> reduction!)

- correct intuition about switching to $\overline{L_\emptyset}$
  but design wrong NTM

→ remember to convert NTM into TM
  (say something about this!!)

Exercise #2

polynomial time TM

class P ($\mathcal{P}$) ← notation from the book

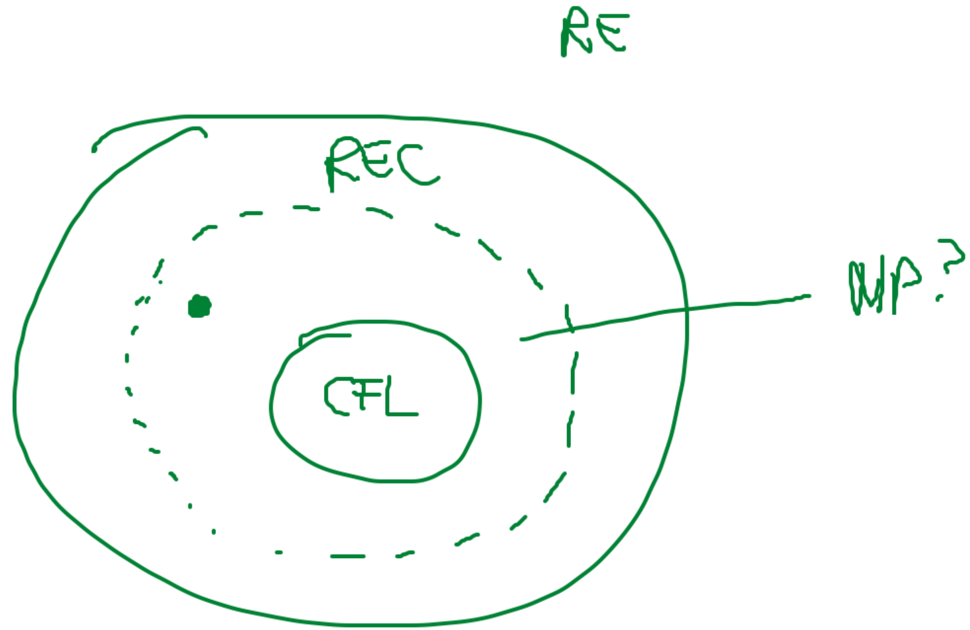
class NP ($\mathcal{NP}$)

<u>question</u>:

$$NP \overset{?}{\subseteq} REC$$

in other words:

does every language in NP belong to REC ??



NP provides yes/no answer in polynomial time

→ <u>always stops</u>

→ when converte into TM may take exp time!

proof: Let L be a language in NP

We prove L is in REC.

if L in NP, there exists NTM N such that $L(N) = L$, and N alway stops in poly time.

Convert N into TM (deterministic) M, so that $L(N) = L(M)$. This forces exponential blow up in computational time. Still M will alway stop! Therefore $L(M) = L$ is in REC.