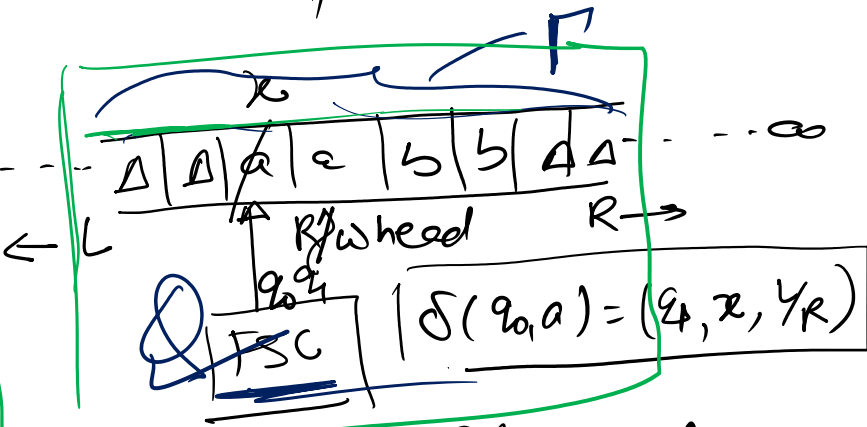
Turing Machine(TM):- Turing m/c is a mathematical model of Abstract Computa

Alan Turing 1936- → Automata

$\Sigma \subset \Gamma$

Formal Definition of TM:- 7 tuples



$TM = (Q, \Sigma, \delta, \Gamma, q_0, \Delta, F)$

- Q: finite non-empty set of states
- Σ: I/p alphabet (finite)
- δ: Transition function
- Γ: finite set of Tape symbol
- q_0: Initial state
- Δ: Blank symbol, $\Delta \notin \Sigma$, $\Delta \in \Gamma$
- F: set of final states, $F \subseteq Q$

$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

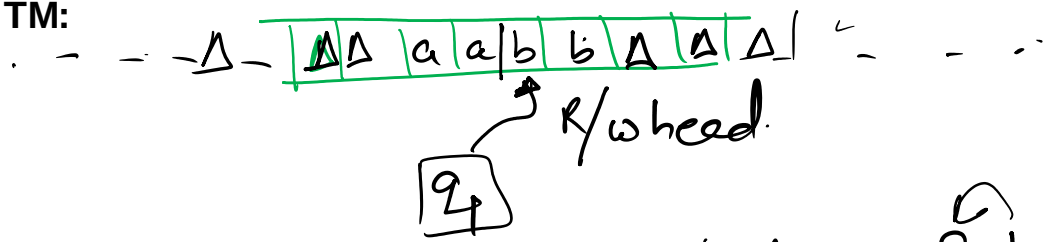
Blank = Δ

b → B

B

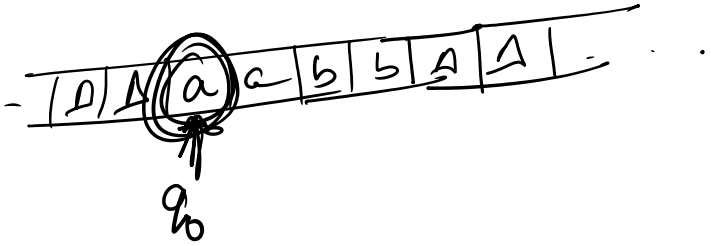
$Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

Instantaneous Description of TM:



$\Delta\Delta aaq_1bb\Delta\Delta$

Initial ID: If $M = (Q, \Sigma, \delta, \nabla, q_0, \Delta, F)$ is a TM Then the initial ID is $\Delta\Delta\Delta\Delta q_0 w \Delta\Delta\ldots$ $w \in \Sigma^*$



Acceptance by TM: a string $\omega \in \Sigma^*$ is said to be Accepted by TM, if it start at initial ID and terminate at some final state i.e.

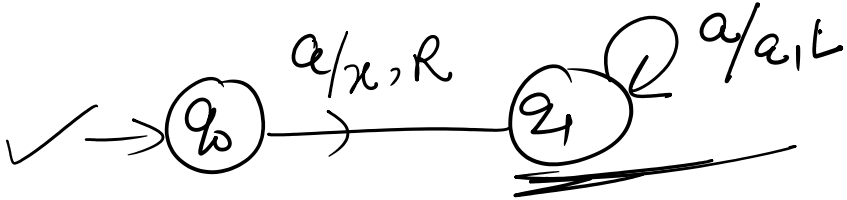
$$L(TM) = \{\omega \mid \omega \in \Sigma^* \mid q_0 \omega \vdash^* \underbrace{\alpha_1 q_i \alpha_2}_{\text{Halt/STOP}}, \alpha_1, \alpha_2 \in \Gamma^*$$

- if $q_i \in F$ ✓ Accepted. Halt with final state (Accepted)
- if $q_i \notin F$ ✓ Rejected. Halt with non final state (Rejected)

Representation of TM:

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

1) Graphical:



$f(n) = 2^n$

2) Table:

	a
$\rightarrow q_0$	$x R q_1$
q_1	$a L q_1$

TM

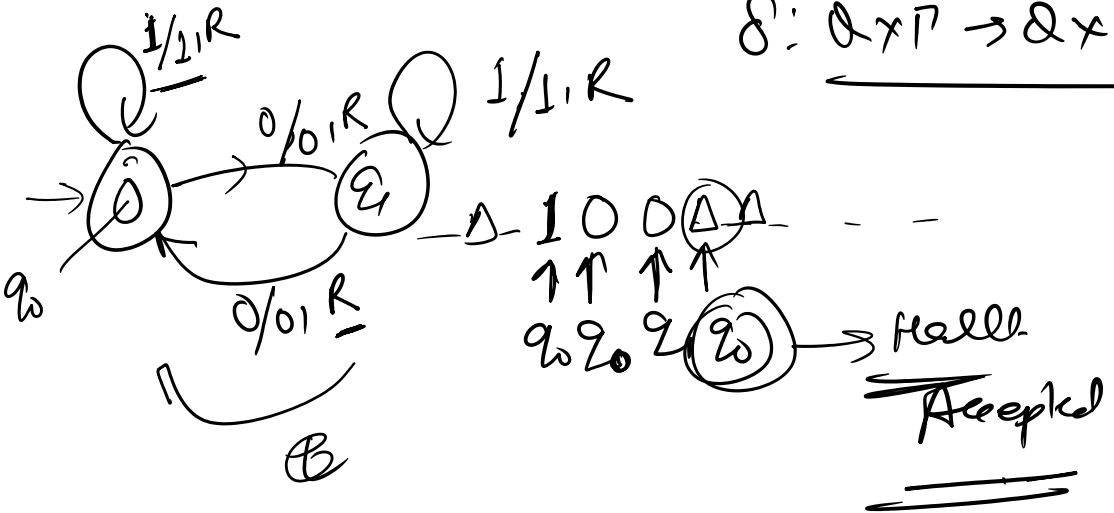
- Acceptor (It accept something)
- Transducer (It perform some function)

$L = \{ww\}$

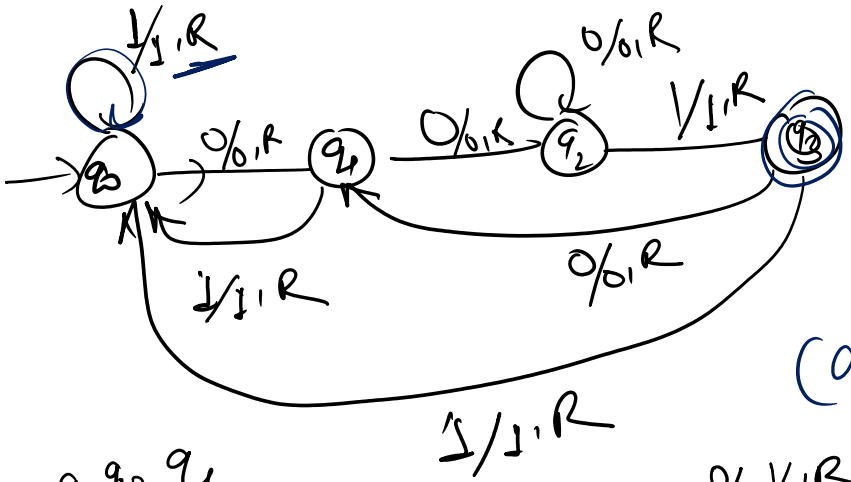
TM Designing:

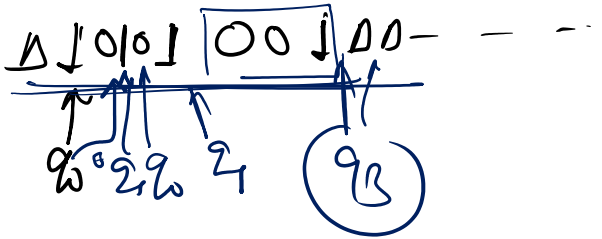
Ex: Design a TM over $\{0,1\}$ that Accept all strings having even no of 0's

$\delta: Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

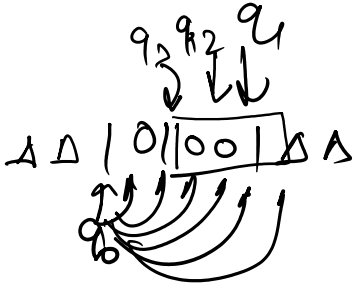


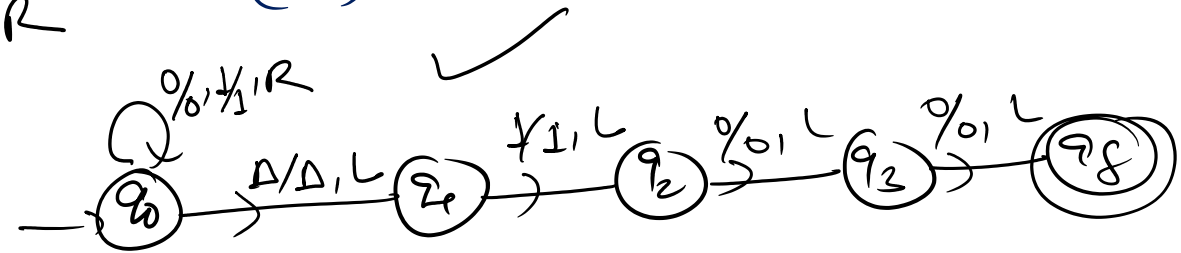
Ex: Design a TM over {0,1} That Accept all strings ending with 001





(or)

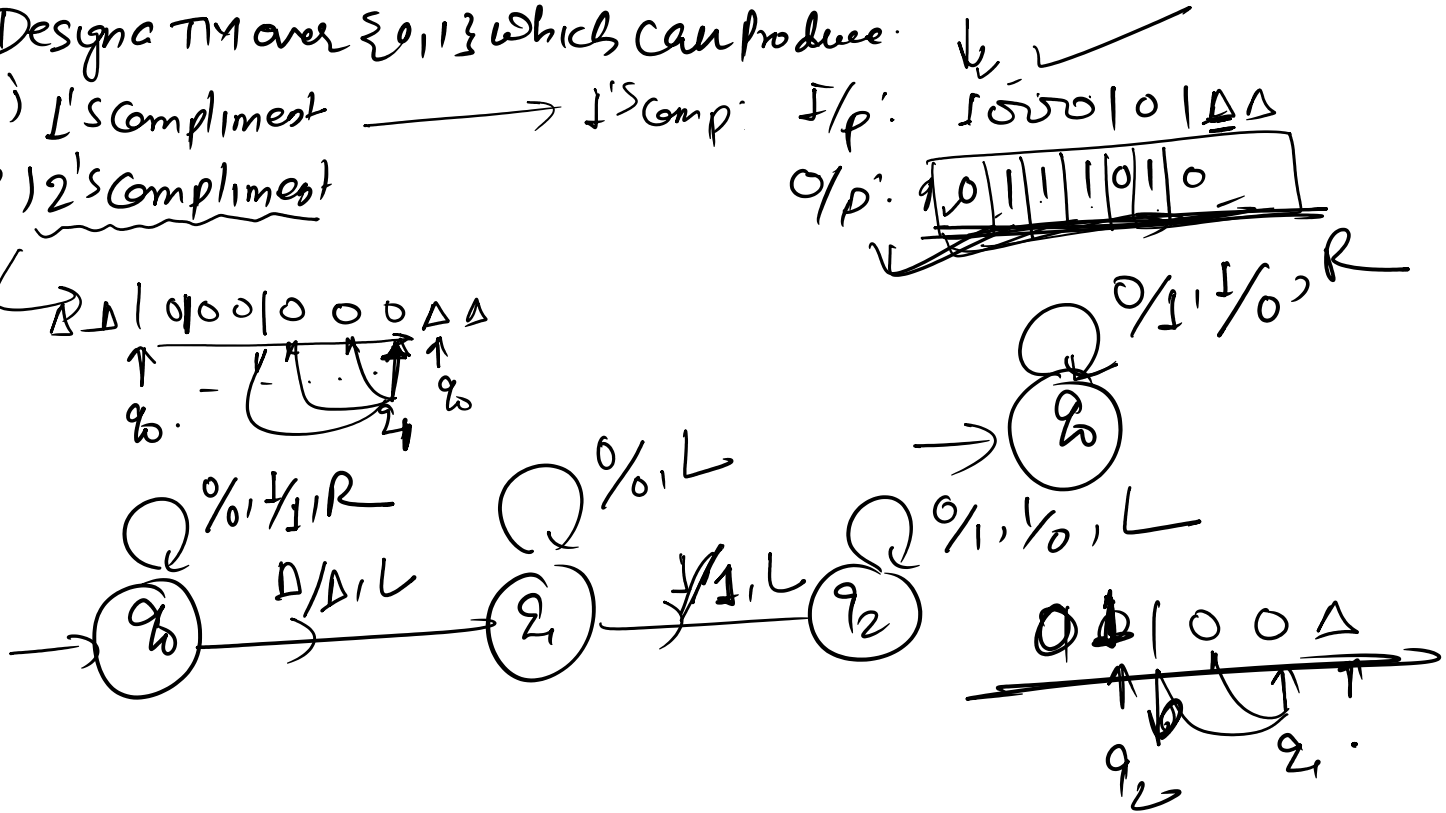




Ex: Design TM over $\{0,1\}$ which can produce:

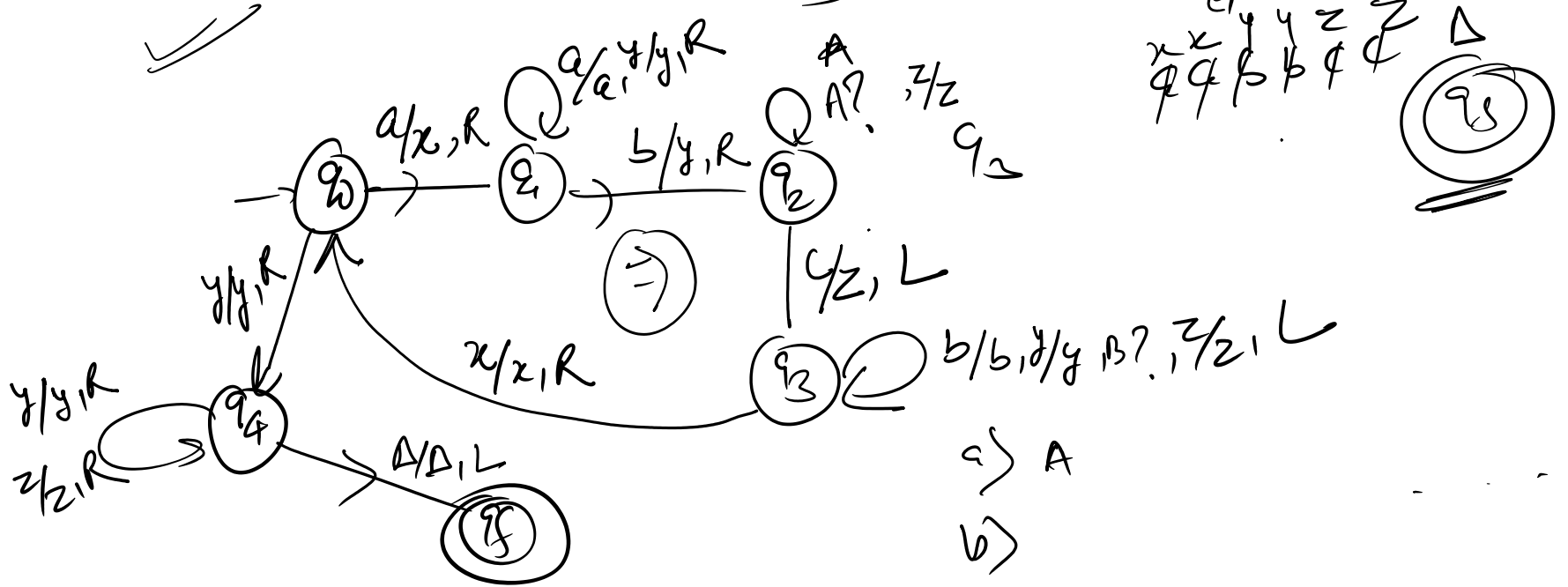
(i) 1's complement

(ii) 2's complement



$L = \{a^n b^n c^n \mid n \geq 1\}$ Design TM

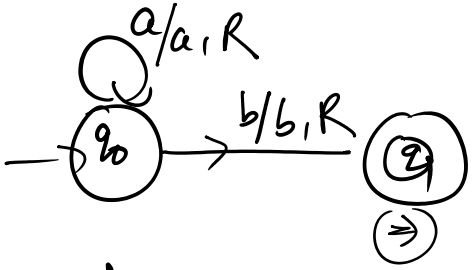
$= \{abc, a^2b^2c^2, a^3b^3c^3 \ldots\}$



a) A

b)

Ex:



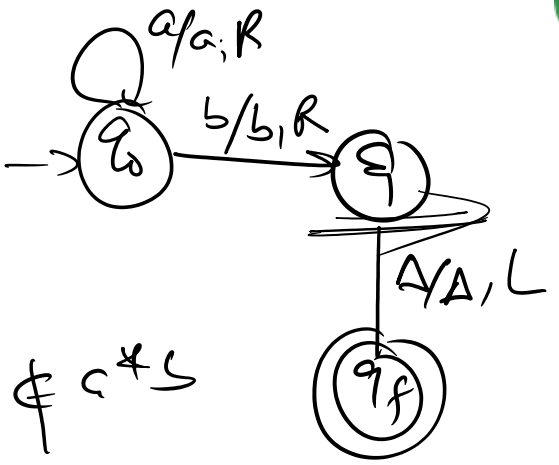
A) a^*b ✗

B) $a^*b(a+b)^*$

C) ~~a^*b~~ neither A or B

D) None.

a^*b



~~aab~~

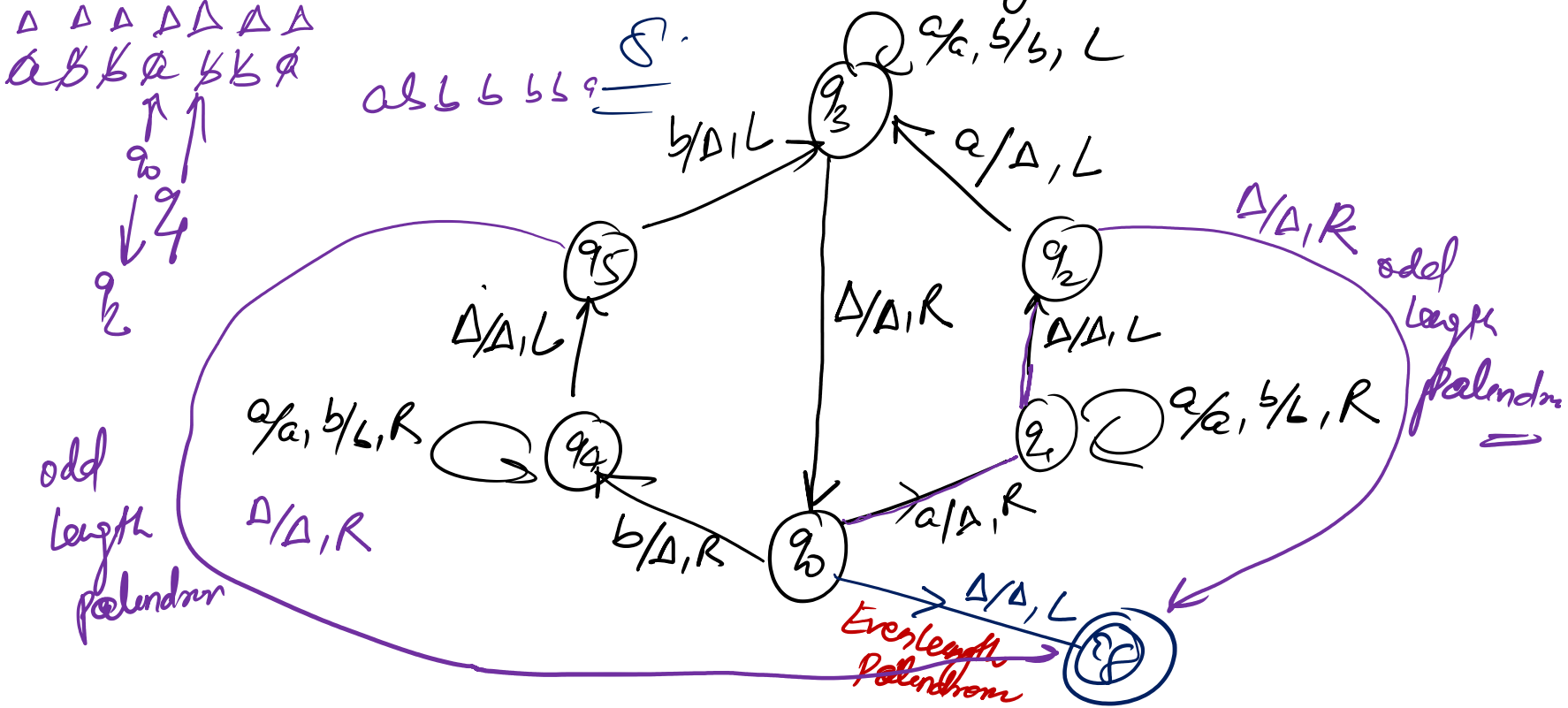
abab $\notin a^*b$

q0 q0 q1

q1

a^*b

$L = \{\omega \mid \omega \in (a,b)^*$ and ω is a Palindrom stng $\} \Rightarrow$ NDCFL



$L = \{ww \mid w \in (a,b)^*\}$

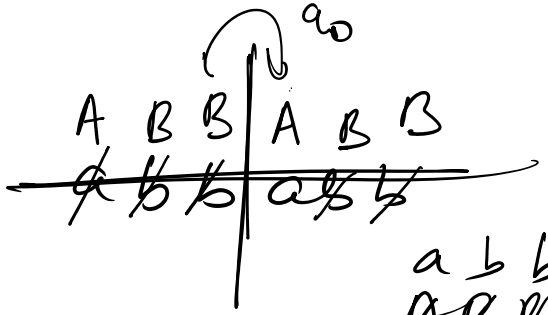
$ww \Rightarrow$ CSL

$a^n b^m c^n d^m$

↳ CSL

$a \to A$

$b \to B$



a b b

A B B A B B

q_0

a b A B B

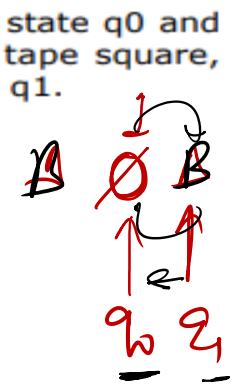
Q: A single tape Turing Machine M has two states q0 and q1, of which q0 is the starting state. The tape alphabet of M is {0, 1, B} and its input alphabet is {0, 1}. The symbol B is the blank symbol used to indicate end of an input string. The transition function of M is described in the following table.

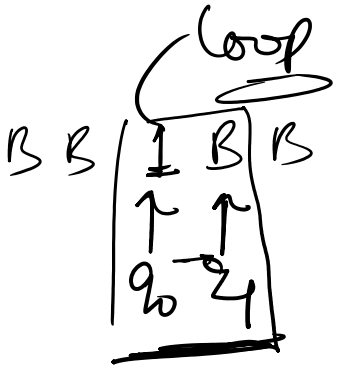
GATE 2003

	0	1	B
q0	q1, 1, R	q1, 1, R	Halt
q1	q1, 1, R	q0, 1, L	q0, B, L

The table is interpreted as illustrated below.

The entry (q1, 1, R) in row q0 and column 1 signifies that if M is in state q0 and reads 1 on the current tape square, then it writes 1 on the same tape square, moves its tape head one position to the right and transitions to state q1.





Which of the following statements is true about M?

(A) M does not halt on any string in $(0+1)^+$

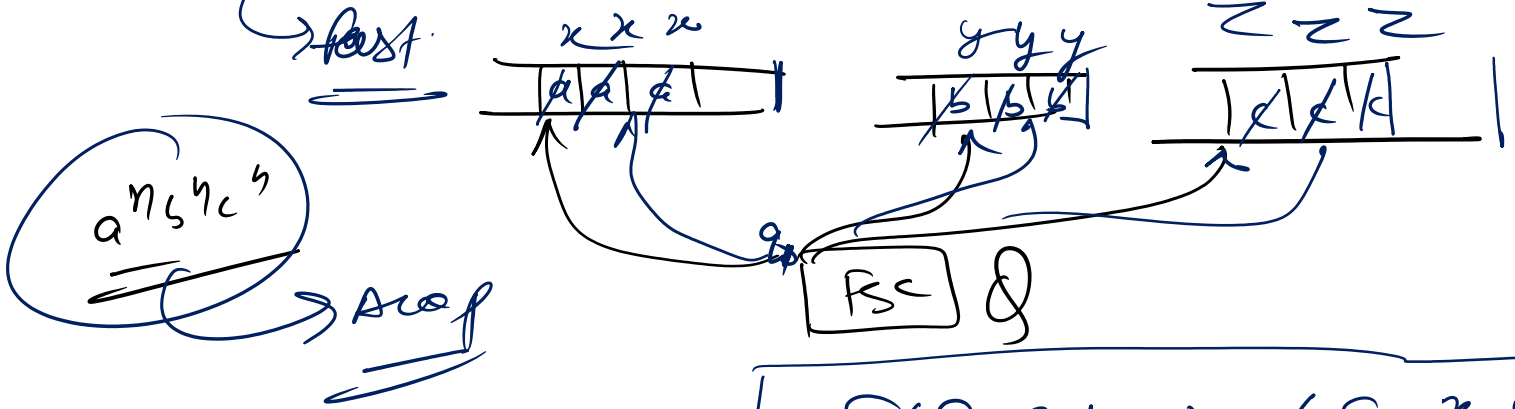
(B) M does not halt on any string in $(00+1)^*$

(C) M halts on all strings ending in a 0

(D) M halts on all strings ending in a 1

Variants of TM: -

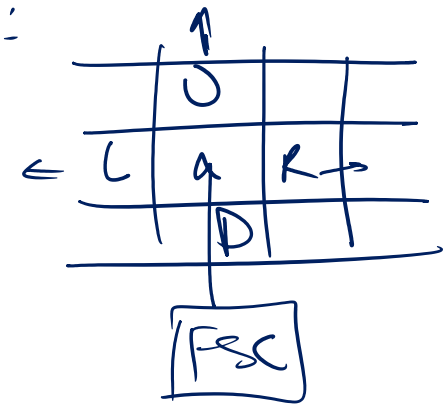
1) multitape TM :- $\delta : Q \times \Gamma^n \rightarrow Q \times \Gamma^n \times \{L, R\}^n$



$\delta(q_0, a, b, c) = (q_1, x, y, z, R, R, R)$

$P(\text{multitape TM}) = P(\text{SDT})$

2) multidemensional TM:



$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, U, D\}$

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, U, D\}$

P(multidemensional TM) = P(STM)

A DPDA with 2-STACK:

$$\delta : Q \times \{\Sigma \cup \epsilon\} \times \Gamma_1 \times \Gamma_2 \rightarrow Q \times \Gamma_1^* \times \Gamma_2^*$$

$$\delta : Q \times (\Sigma \cup \epsilon) \times \Gamma \rightarrow Q \times \Gamma^*$$

A NPDA with 2 STACK.

$$\delta : Q \times (\Sigma \cup \epsilon) \times \Gamma_1 \times \Gamma_2 \rightarrow 2^{Q \times \Gamma_1^* \times \Gamma_2^*}$$

$P(DPDA) \subsetneq P(NPDA)$

$$P(\text{DPDA with 2 STACK}) \equiv P(\text{NPDA with 2 STACK}) \equiv P(\text{STM})$$

Non Deterministic TM: (NDTM)

$$\delta : Q \times \Gamma \rightarrow 2^{Q \times P \times \{L, R\}}$$

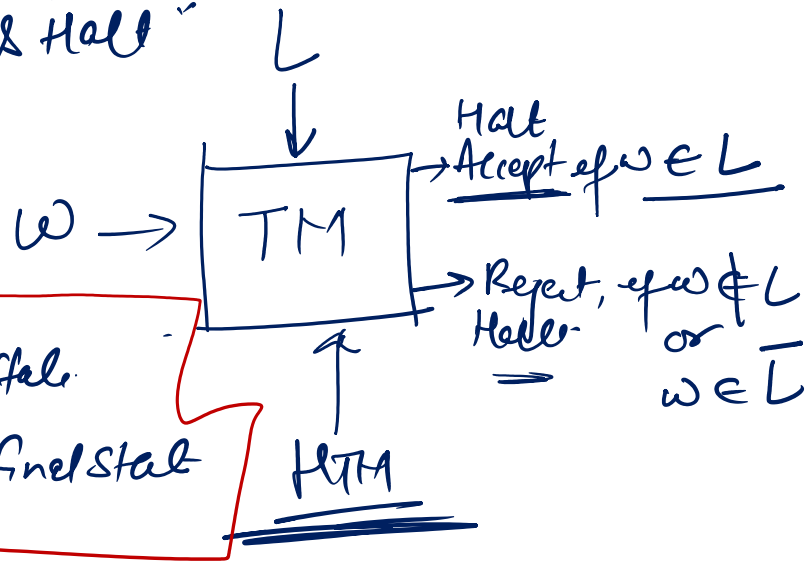
$$P(NDTM) \cong P(DTM)$$

Halting Turing Machine (HTM) :

After Processing the i/p string the HTM halt either in a final state or non final state

i.e on every i/p string, HTM always Halt

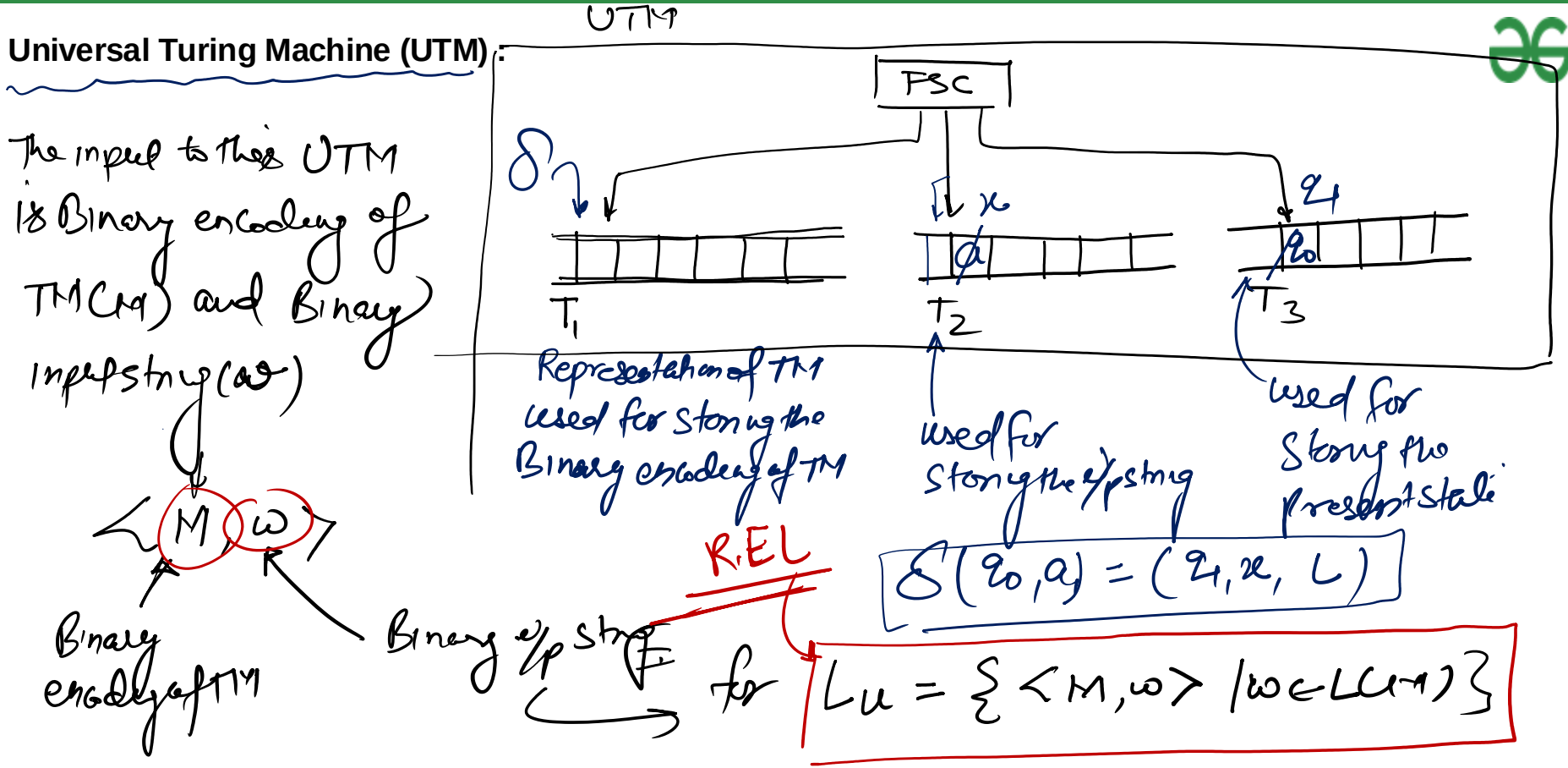
- HTM is used to Accept Recursive Languages.
- HTM power is lesser than STM



if $w \in L \Rightarrow$ HTM Halts in a final state.

if $w \notin L \Rightarrow$ HTM Halt in a non final state

Universal Turing Machine (UTM):



Binary encoding of TM:

① $\delta(q_0, a) = (q_1, x, L)$

② $\delta(q_1, b) = (q_2, y, R)$

If start & end with 111, separator for two 11

$q_0 \to 0$
$q_1 \to 00$
$q_2 \to 000$

$a \to 0$
$b \to 00$
$x \to 000$
$y \to 0000$

$L \to 0$
$R \to 00$

111 0 10 100 1000 1 0 11 00 100 1000 10000 100 111

$\delta(q_0, a) = (q_1, x, L), \delta(q_1, b) = (q_2, y, R)$

Church Thesis:

Turing m/c

- As an Acceptor: Every logically recognise languages is Accepted by TM
- As a Transducer: Every logically Computable function can be Computed by TM

Recursive Language: A language L is said to be Recursive iff there exist a HTM to accept L

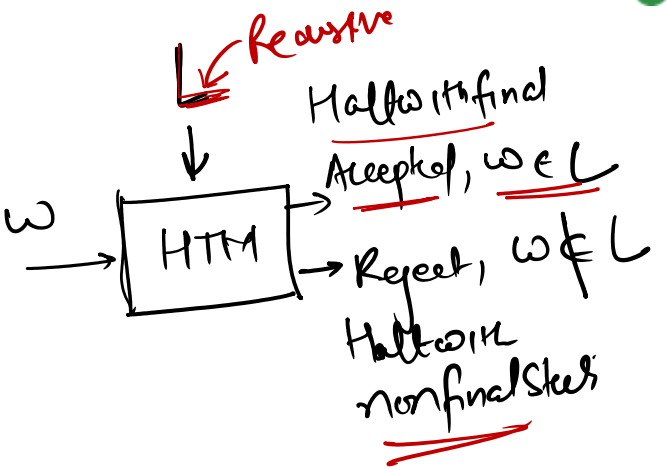
$L \in Rec \Rightarrow \exists HTM$

$\exists HTM \Rightarrow L \in Rec$

$L \in \sim Rec \Rightarrow \sim \exists HTM$

$\sim \exists HTM \Rightarrow L \in \sim Rec$

Rec = Rec



→ Turing Decidable or Decidable language.

Recursive Enumerable Language: A language L is said to be Recursive Enumerable.

Language (REL) iff ∃ a TM to Accept L

L ∈ RE ⇒ ∃ a TM

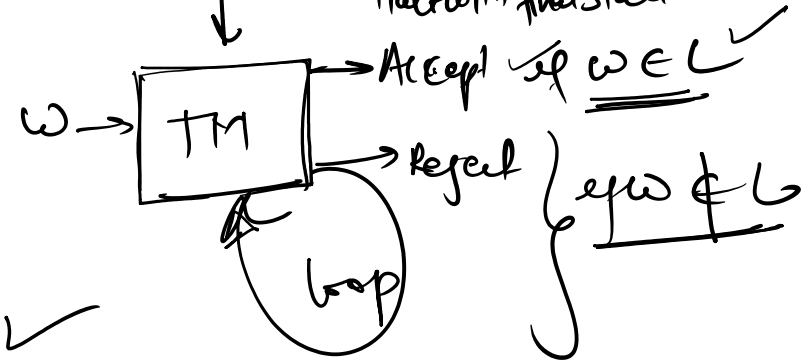
∃ a TM ⇒ L ∈ RE

L ∈ ∼RE ⇒ ∀ a TM

∼∃ TM ⇒ L ∈ ∼RE

L Recursive Enumerable language

Halting final state



Turing Recognizable language.

Turing Enumerable language.

Semidecidable language.

Undecidable language

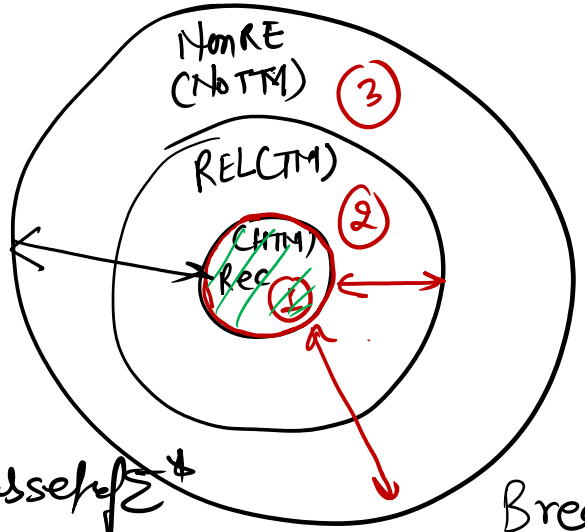
1 → Rec (Decidable)

1+2 → REL (Semidecidable)

2 → RELbutnotRec (Undecidable)

2+3 → Not Rec (Undecidable)
(REL+NotREL)

3 → NotREL (Undecidable)



$1+2+3 = 2^{\Sigma^*}$

REL + NotREL = 2^{Σ^*}

→ Set of all sets of Σ^*
or
Set of all languages

→ Undecidable.

Break 15 minit

1:35 Continu

Closure Properties of Languages:

Example: L_1 = Regular, L_2 = DCFL, L_3 = REL but not Recursive. Which of the following statement FALSE?

a) $L_1 \cap L_2$ is DCFL

b) $L_3 \cap L_1$ is Recursive

c) $L_1 \cup L_2$ is CFL

d) $L_1 \cap L_2 \cap L_3$ is R.EL

Regular ∩ DCFL ∩ REL

↑ DCFL ∩ REL

↑ CFL ∩ REL

REL ∩ REL => REL

Operations	RL	CFL	DCFL	CSL	REC	REL
Union	✓	✓	×	✓	✓	✓
Intersection	✓	×	×	✓	✓	✓
Set Difference	✓	×	×	✓	✓	×
Complementation	✓	×	✓	✓	✓	×
Intersection with Regular	✓	✓	✓	✓	✓	✓
Concatenation	✓	✓	×	✓	✓	✓
Kleen Closure	✓	✓	×	✓	✓	✓
Kleen Plus	✓	✓	×	✓	✓	✓
Reversal	✓	✓	×	✓	✓	✓
Homomorphism	✓	✓	×	×	×	✓
ε-free Homomorphism	✓	✓	×	✓	✓	✓
Inverse Homomorphism	✓	✓	✓	✓	✓	✓
Substitution	✓	✓	×	×	×	✓
ε-free Substitution	✓	✓	×	✓	✓	✓
Quotient with Regular	✓	✓	✓	×	×	✓

Closure Properties of Languages:

Example: L_1 = Recursive and L_2, L_3 = REL but not Recursive. Which of the following statement is not necessarily TRUE?

a) $L_1 - L_2$ is R.EL

b) $L_1 - L_3$ is R.EL

c) $L_2 \cap L_3$ is R.EL True

d) $L_2 \cup L_3$ is R.EL True

a) $L_1 - L_2 = L_1 \cap \overline{L_2}$ ⇒ Rec ∩ Rec ⇒ not true

Rec ∩ Rel

Rel ∩ Rel

$L_1 - L_3$ ⇒ not R

L_1

Operations	RL	CFL	DCFL	CSL	REC	REL
Union	✓	✓	×	✓	✓	✓
Intersection	✓	×	×	✓	✓	✓
Set Difference	✓	×	×	✓	✓	×
Complementation	✓	×	✓	✓	✓	×
Intersection with Regular	✓	✓	✓	✓	✓	✓
Concatenation	✓	✓	×	✓	✓	✓
Kleen Closure	✓	✓	×	✓	✓	✓
Kleen Plus	✓	✓	×	✓	✓	✓
Reversal	✓	✓	×	✓	✓	✓
Homomorphism	✓	✓	×	×	×	✓
ε-free Homomorphism	✓	✓	×	✓	✓	✓
Inverse Homomorphism	✓	✓	✓	✓	✓	✓
Substitution	✓	✓	×	×	×	✓
ε-free Substitution	✓	✓	×	✓	✓	✓
Quotient with Regular	✓	✓	✓	×	×	✓

Closure Properties of Languages:

Example: Consider the following types of language. Which of the following is/are TRUE?

L_1 : Regular, L_2 : CFL, L_3 : Recursive, L_4 : REL

i) $\overline{L_3} \cup L_4$ is R.EL True

ii) $L_2 \cup L_3$ is Recursive

iii) $L_1^* \cap L_2$ is CFL True

iv) $L_1 \cup L_2$ is CFL False

a) i only b) i & iii only c) i & iv only d) ~~i~~, ii & iii only

L_1 = Regular

L_2 = CFL

L_3 = Rec

L_4 = Rel

$\overline{L_3} \cup L_4$

↑ Rec ∪ Rel

Rel ∪ Rel = Rel

Operations	RL	CFL	DCFL	CSL	REC	REL
Union	✓	✓	×	✓	✓	✓
Intersection	✓	×	×	✓	✓	✓
Set Difference	✓	×	×	✓	✓	×
Complementation	✓	×	✓	✓	✓	×
Intersection with Regular	✓	✓	✓	✓	✓	✓
Concatenation	✓	✓	×	✓	✓	✓
Kleen Closure	✓	✓	×	✓	✓	✓
Kleen Plus	✓	✓	×	✓	✓	✓
Reversal	✓	✓	×	✓	✓	✓
Homomorphism	✓	✓	×	×	×	✓
ε-free Homomorphism	✓	✓	×	✓	✓	✓
Inverse Homomorphism	✓	✓	✓	✓	✓	✓
Substitution	✓	✓	×	×	×	✓
ε-free Substitution	✓	✓	×	✓	✓	✓
Quotient with Regular	✓	✓	✓	×	×	✓

L & $\bar{L}$ Theorem:

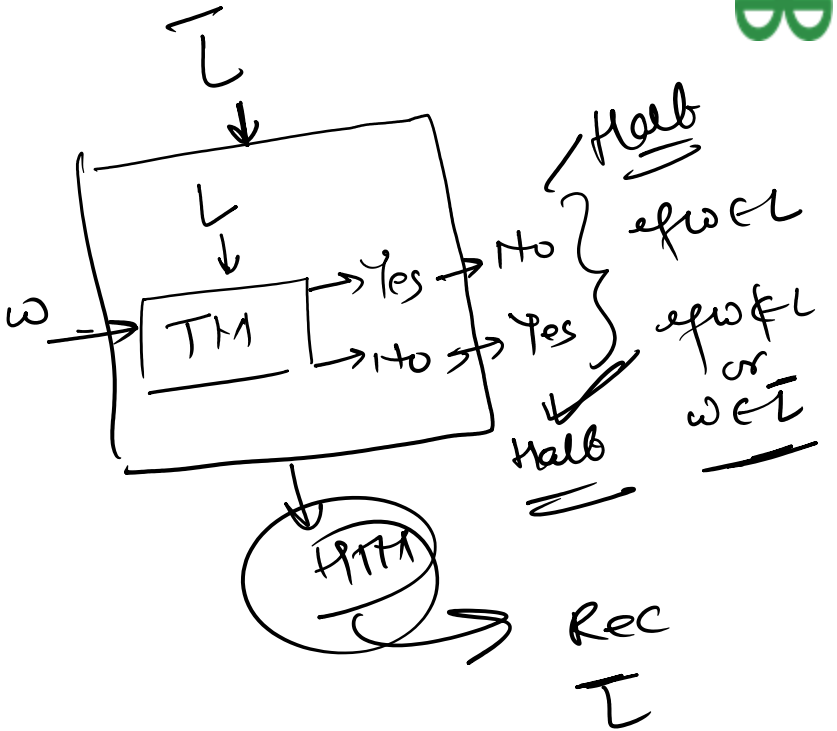
- If L is Rec ⇒ L is also RE
- If L is RE ⇒ L may or may not be Rec
- If L is Rec ⇒ $\bar{L}$ is also Rec
- If L is RE ⇒ $\bar{L}$ may or may not be RE
- If L & $\bar{L}$ both are RE ⇒ L is Rec and $\bar{L}$ is also Rec

- If L is Rec ⇒ Compliment of Rec is also Rec
- If L is RE ⇒ Compliment of RE is either Rec or Non RE

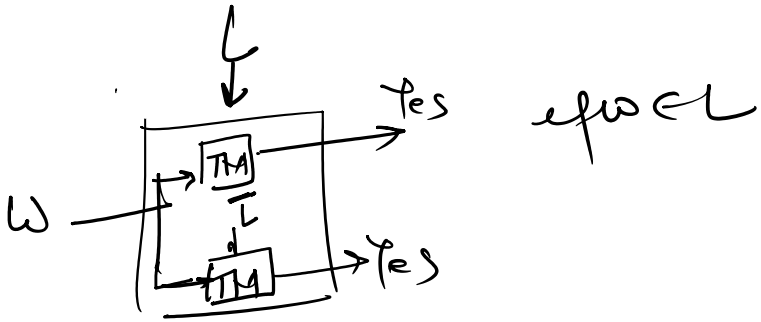
Operations	RL	CFL	DCFL	CSL	REC	REL
Union	✓	✓	×	✓	✓	✓
Intersection	✓	×	×	✓	✓	✓
Set Difference	✓	×	×	✓	✓	×
Complementation	✓	×	✓	✓	✓	×
Intersection with Regular	✓	✓	✓	✓	✓	✓
Concatenation	✓	✓	×	✓	✓	✓
Kleen Closure	✓	✓	×	✓	✓	✓
Kleen Plus	✓	✓	×	✓	✓	✓
Reversal	✓	✓	×	✓	✓	✓
Homomorphism	✓	✓	×	×	×	✓
ε-free Homomorphism	✓	✓	×	✓	✓	✓
Inverse Homomorphism	✓	✓	✓	✓	✓	✓
Substitution	✓	✓	×	×	×	✓
ε-free Substitution	✓	✓	×	✓	✓	✓
Quotient with Regular	✓	✓	✓	×	×	✓

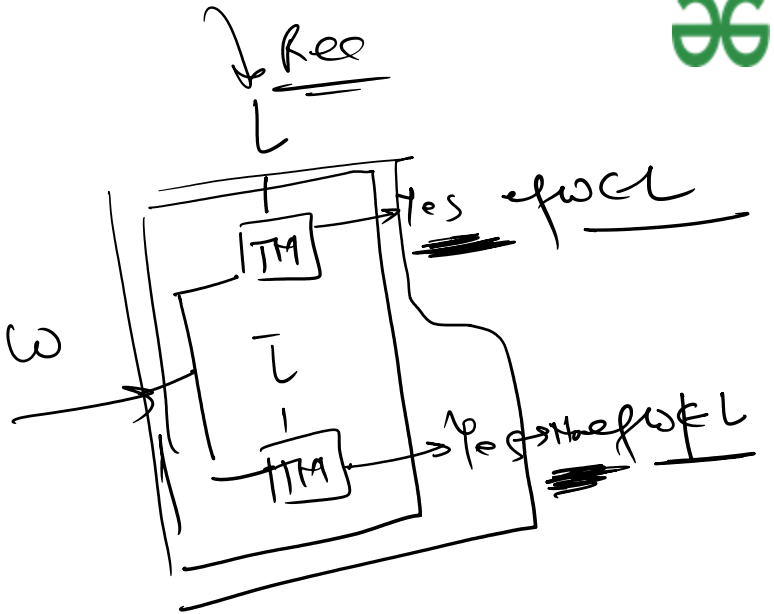
Closure Properties Proofs:

1) If L is recursive Then $\overline{L}$ is also Recursive

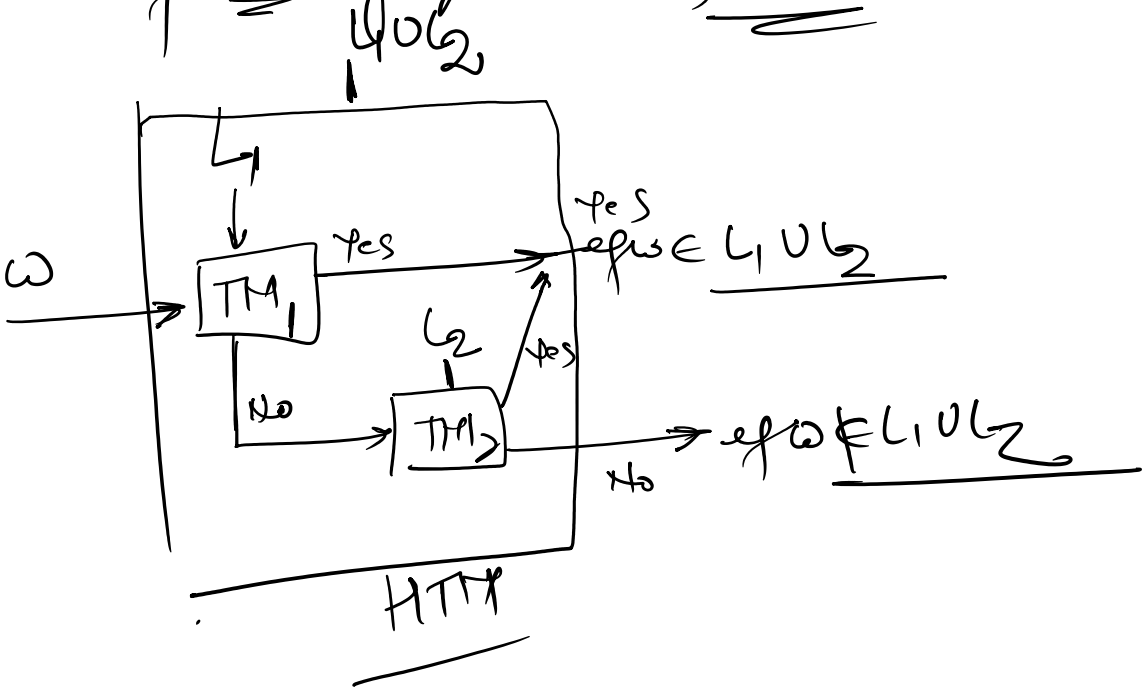


2) If L and $\bar{L}$ are Rel, Then L is Rec

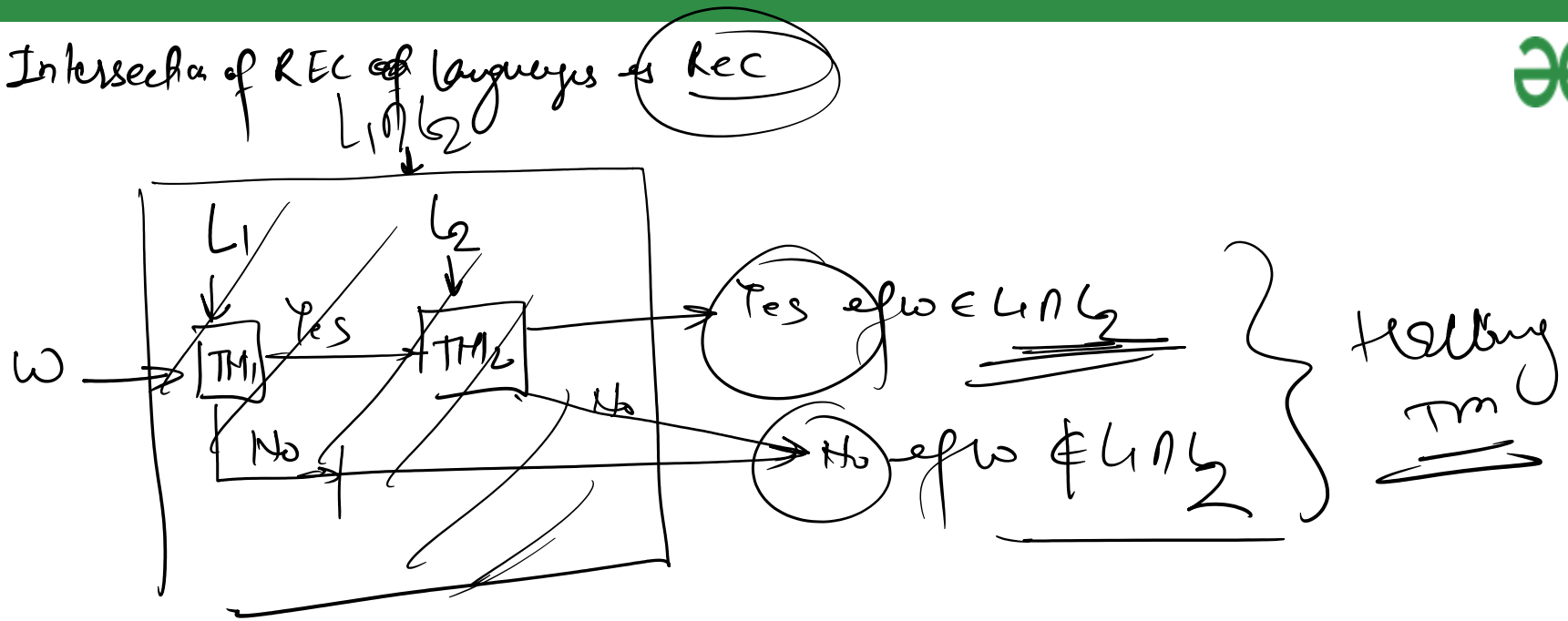




3) Union of Rec language is Rec $\Rightarrow$ Rec



4) Intersection of REC languages is REC



5) Union of two Rel is Rel