Ques : Design a TM for

$$f : \Sigma_0^* \to \Sigma_0^*$$

Where $f(w) = w w^R$

i.e. $s, \triangleright \sqcup ab \underline{\sqcup} \vdash h, \triangleright \sqcup abba\underline{\sqcup}$

Let f be any function from Σ_0^* to Σ_0^*

$$\Sigma_0 = \Sigma - \{\sqcup, \triangleright\}$$

We say that, M Computes function f, if

$$\forall w \in \Sigma_0^* \quad M(w) = f(w)$$

i.e, M halts on input w → yes / → no $H = \{y, n\}$

A function f is called recursive, if there is a TM M that computes f.

recursive function

Recursive Language

Turing decidable Machine

$w \in L$
$w \notin L$ → halt

Recursive Enumerable Languages

Let $M = (K, \Sigma, \delta, s, H)$ be a TM,

let $\Sigma_0 \subseteq \Sigma - \{\sqcup, \triangleright\}$ be an alphabet,

and $L \subseteq \Sigma_0^*$ be a language.

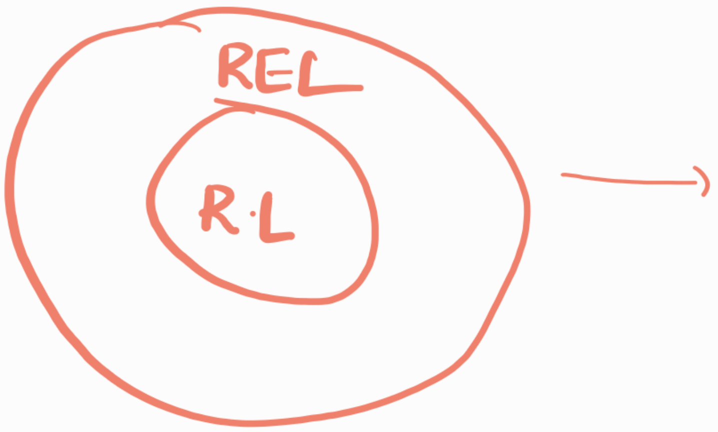
We say that, M <u>semidecides</u> $\mathcal{L}$ if for any string $w \in \Sigma_0^*$ the following is true:

(1) $\boxed{w \in \mathcal{L} \text{ iff } M \text{ halt on } w.}$

(2) $w \notin L$ M never enter a halting state
i.e machine continues its computation indefinitely

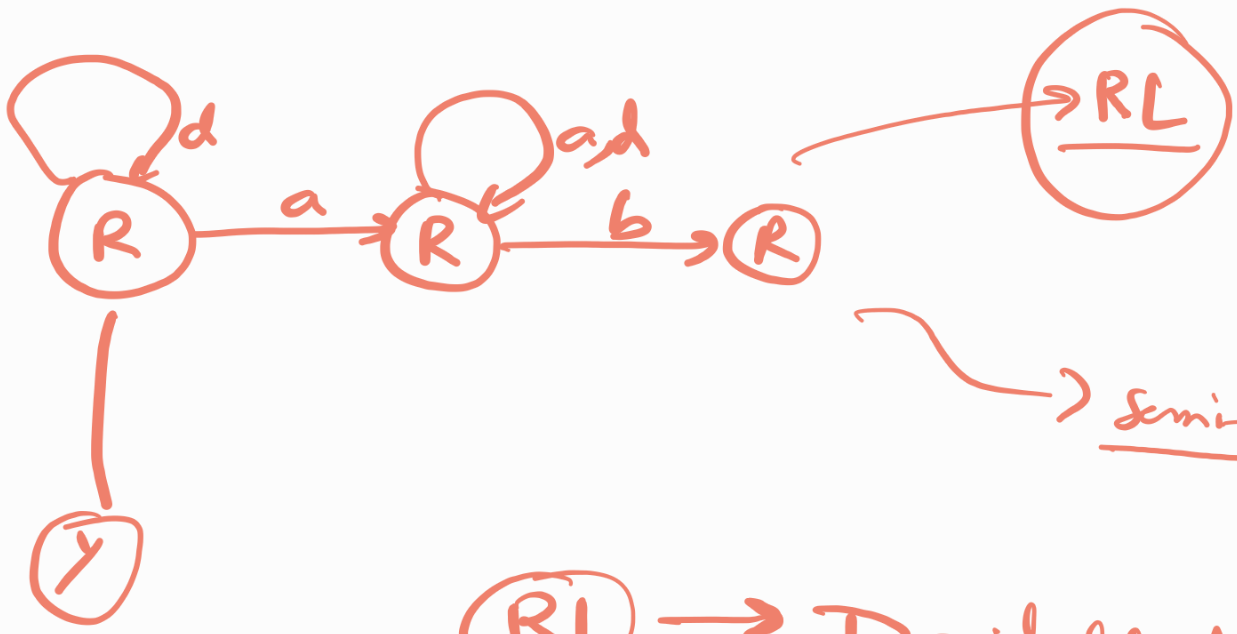
$\mathcal{L}$ is <u>Recursive Enumerable Language</u>

if ↳ there is a TM that semi-decides $\mathcal{L}$



REL : Recursive Enumerable Language

RL : Recursive Language



Semi-decidable

RL → Decidable Machine

{y, n}

w∈L → y   n ← w∉L

REL → Semidecidable Machine

{y}

w∈L → y

M halts

w∉L

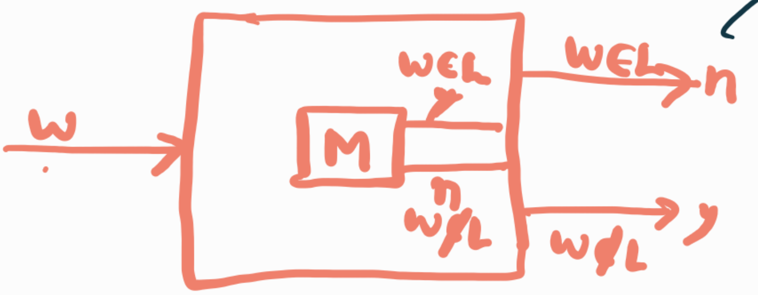
Ques : Prove that if L is recursive Language, then its Complement $\bar{L}$ is also recursive i.e. Recursive Language are closed under complementation

Ans : Let $M = (K, \Sigma, \delta, s, \{y, n\})$ is a TM that decides L

Then, there must exist a TM

$$M' = (K, \Sigma, \delta', s, \{n, y\})$$

that must also decides L



Complement

M & M' are identical, except that it reverses the roles of y and n

$$\delta'(q, a) = \begin{cases} n & \text{if } \delta(q,a) = y, \\ y & \text{if } \delta(q,a) = n, \\ \delta(q,a) & \text{otherwise} \end{cases}$$

$M'(w) = y$ iff $M(w) = n$ are therefore

M' decides $\overline{L}$

Recursive Languages are closed under

Complementation

$L \longrightarrow$ Recursive

$\overline{L} \longrightarrow$ Recursive

Complementation

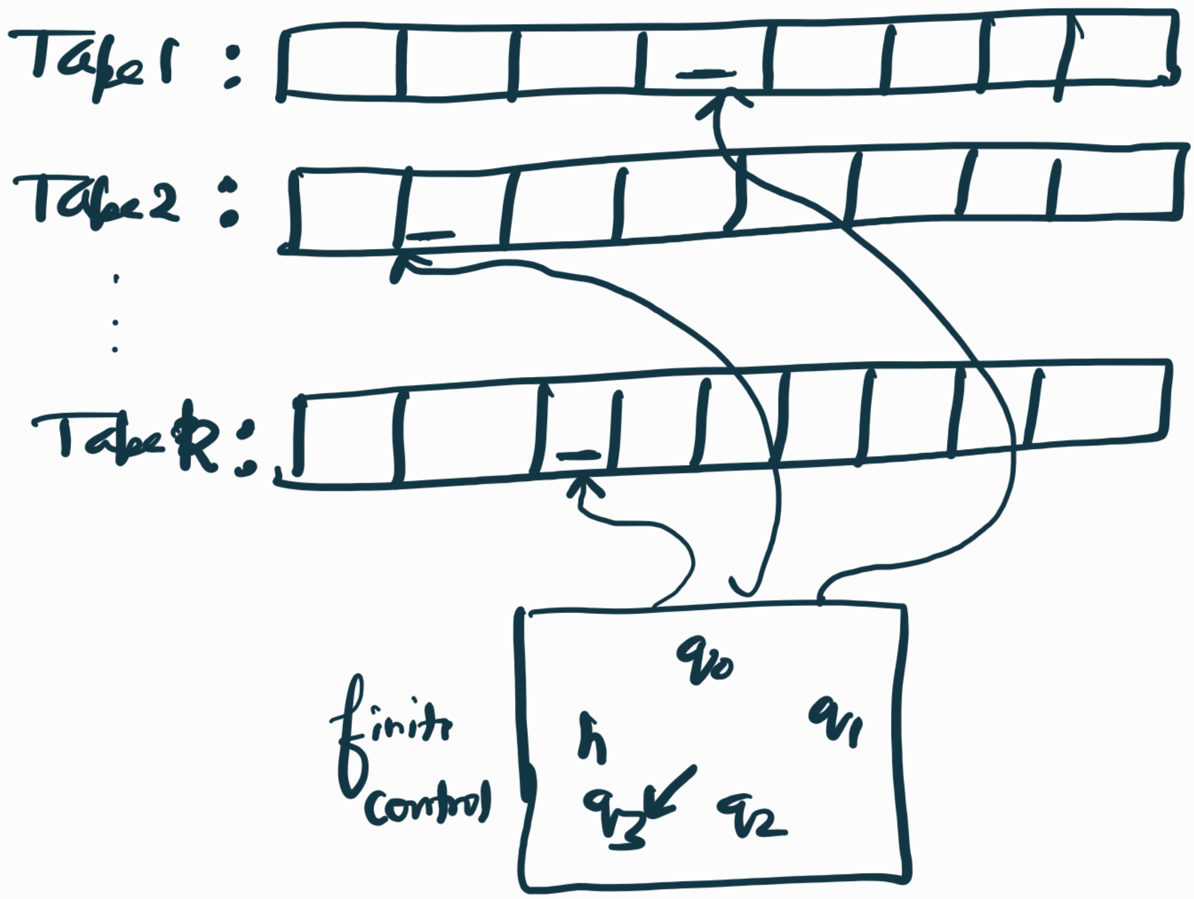
TM with Multiple Tapes

Set of States

Multi-tape TM

Let $M = (K, \Sigma, \delta, s, H)$ be a k-tape TM. A configuration of M is a member of

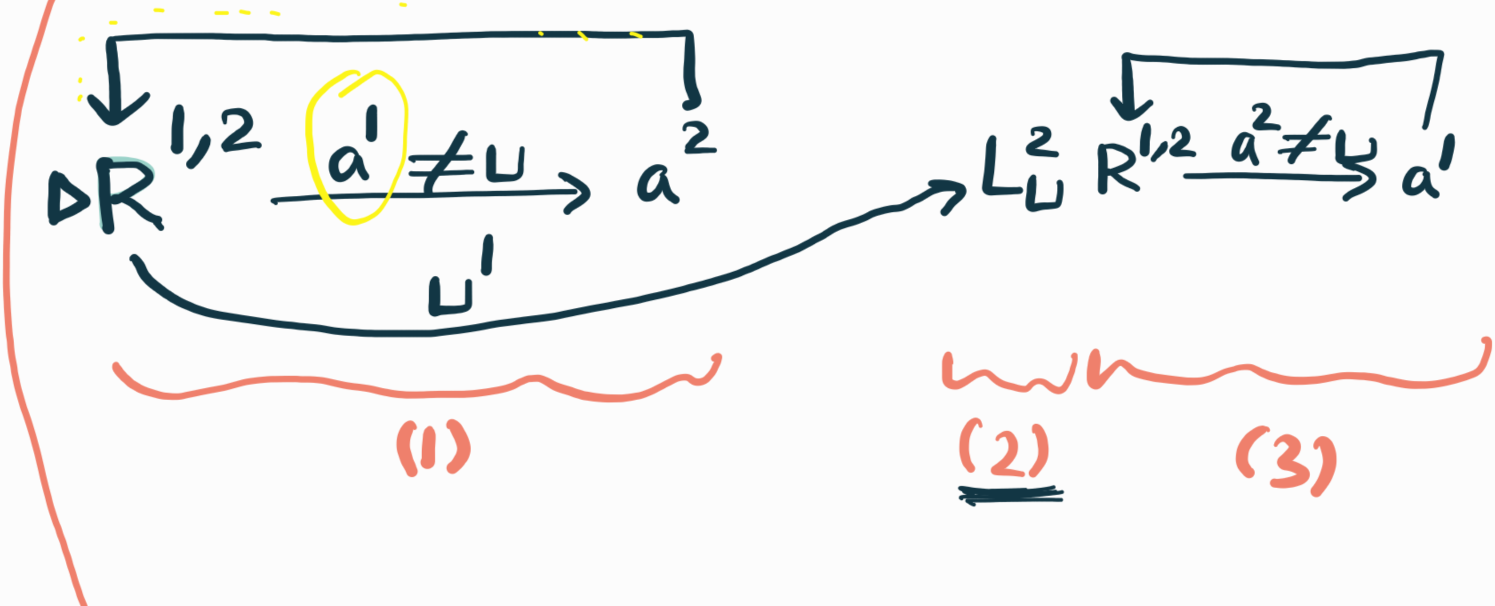
$$\underline{K} \times (\underline{\triangleright \Sigma^*} \times (\Sigma^*(\Sigma - \{\sqcup\} \cup \{\epsilon\}))^k$$



Ques: Design a 2-tape TM for the copying Machine to accomplish the task of transforming

$\triangleright \underline{\sqcup} w \sqcup$ into $\triangleright \sqcup w \sqcup w \underline{\sqcup}$

where $w \in \{a, b\}^*$



	1st Tape	2nd Tape	
	$\triangleright \underline{\sqcup} w \sqcup$	$\triangleright \underline{\sqcup}$	Initial configuration
(1)	$\triangleright \sqcup \underline{w} \sqcup$	$\triangleright \sqcup \underline{\sqcup}$	
	$\triangleright \sqcup \underline{w} \sqcup$	$\triangleright \sqcup \underline{w}$	
	$\triangleright \sqcup w \underline{\sqcup}$	$\triangleright \sqcup w \underline{\sqcup}$	
2)	$\triangleright \sqcup w \underline{\sqcup}$	$\triangleright \underline{\sqcup} w \sqcup$	
3)	$\triangleright \sqcup w \sqcup \underline{\sqcup}$	$\triangleright \sqcup \underline{w} \sqcup$	
	$\triangleright \sqcup w \sqcup \underline{w}$	$\triangleright \sqcup \underline{w} \sqcup$	
	$\triangleright \sqcup w \sqcup w \underline{\sqcup}$	$\triangleright \sqcup w \underline{\sqcup}$	

Copying Machine

$$\triangleright \underline{\sqcup} ab \sqcup \;\vdash\; \triangleright \sqcup ab \sqcup ab \underline{\sqcup}$$