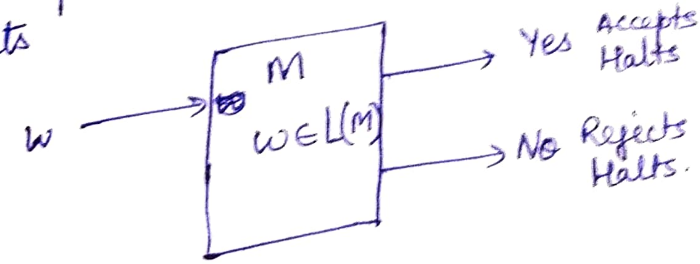
Closure Properties of Recursive languages

Recursive languages are closed under

- Union
- Complementation
- Concatenation
- Kleene star
- Intersection
- set difference
- Reversal
- Inverse Homomorphism

Recursive languages → A language is recursive if there is a turing machine that accepts the language and halts on all its inputs

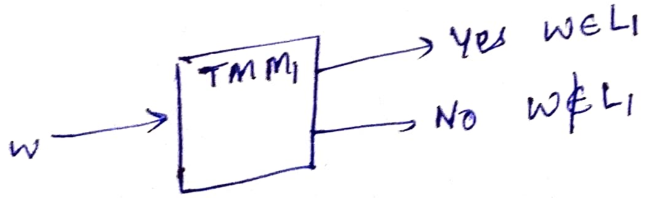


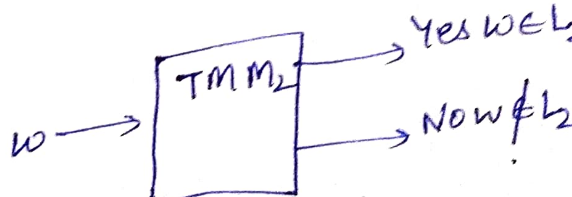
Union: The recursive languages are closed with respect to union. if L_1 and L_2 are recursive languages then $L_1 \cup L_2$ is also recursive.

M_1 and M_2 alway halt.

Proof

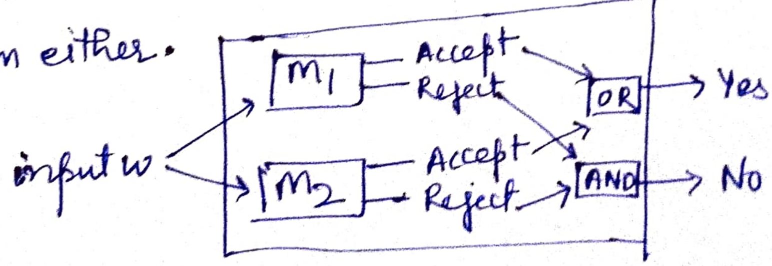
M_1 be a TM such that $L_1(M_1)$
M_2 be a TM such that $L_2(M_2)$





Now from this turing machines we can construct a TM M' such that $L_1(M_1)$ and $L_2(M_2)$

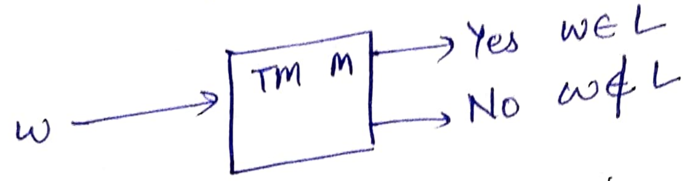
M' accepts if w is either
M' rejects if w is not in either.



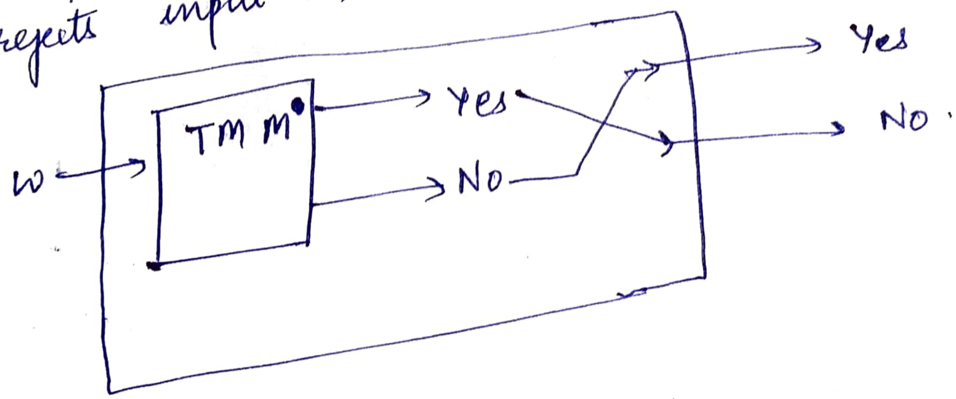
Complementation The recursive languages are closed with complementation
if L be a recursive language then $\bar{L}$ is also recursive

$$\bar{L} = \Sigma^* - L$$

Proof Let M be a turing machine such that $L = L(M)$ and M always halts.

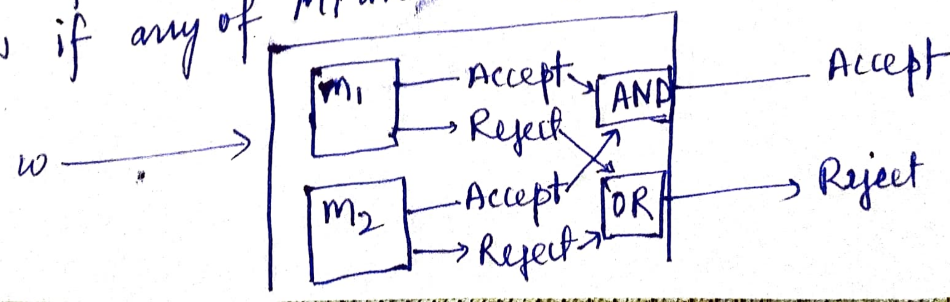


Now from this we can create a turing machine M' such that
when M accepts input $\Rightarrow$ M' rejects
M rejects input $\Rightarrow$ M' accepts.



Intersection The recursive languages are closed with respect to intersection.
L_1 and L_2 are recursive then $L_1 \cap L_2$ is also recursive.

Proof Let M_1 be a TM such that $L_1 = L(M_1)$
Let M_2 be a TM such that $L_2 = L(M_2)$
Now from M_1 and M_2 we can create a TM such that
M' accepts w if both M_1 and M_2 accept w and halt.
M' rejects w if any of M_1 and M_2 rejects w.



Concatenation The recursive languages are closed with respect to concatenation.

if L_1 and L_2 are recursive then $L_1 \cdot L_2$ is also recursive.

Proof Let M_1 be a TM such that $L_1 = L(M_1)$

Let M_2 be a TM such that $L_2 = L(M_2)$

~~For each~~ input w for each of the $|w|+1$ ways to divide w as x

run M_1 on x and

run M_2 on y

and accept if both accept

else reject

Kleene star: The recursive language is closed with respect to Kleene star.

Proof Let L be a recursive language

Let M be a turing machine such that $L = L(M)$

on input w, if $w = \epsilon$ accept

else for each of $2^{|w|-1}$ ways to divide w as

$w_1, w_2 \text{ --- } w_K$ $(w_i \neq \epsilon)$

run M on each w_i

Accept if M accepts all.

else reject.

Closure Properties of Recursive Enumerable Languages

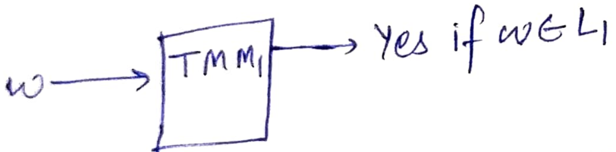
- Union, Intersection, concatenation, star closure, Reversal, Homo-morphism inverse Homomorphism

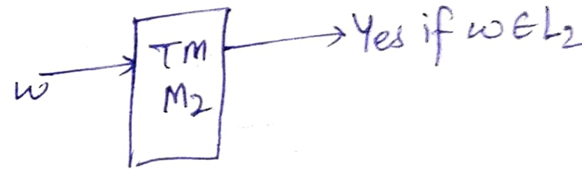
Recursive Enumerable Language :- TM ~~halt~~ accepts and halts if $w \in L$
TM may or may not halt if $w \notin L$

Union: The Recursive enumerable languages are closed under union operation.
if L_1 and L_2 are recursive enumerable languages then $L_1 \cup L_2$ is also recursive enumerable.

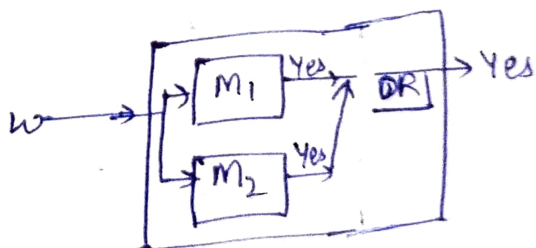
Proof Let M_1 be a TM such that $L_1 = L(M_1)$
Let M_2 be a TM such that $L_2 = L(M_2)$

M_1 and M_2 always halt for accept.

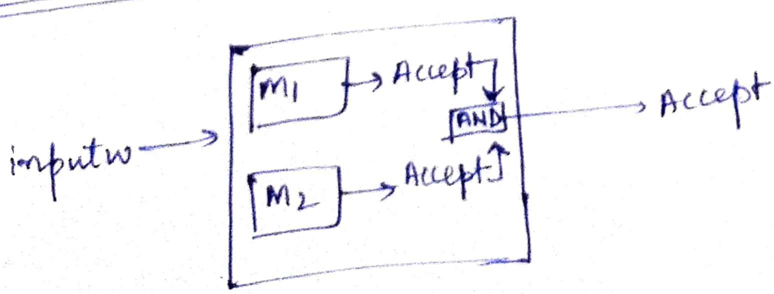




Now we can create a turing machine M' such that when any of M_1 and M_2 accepts $\Rightarrow$ M' accepts.



Intersection RE Languages are closed with respect to intersection.



Concatenation of RE Languages

Let $L_1 = L(M_1)$ and $L_2 = L(M_2)$

Assume M_1 and M_2 are single semi-infinite TM's

construct 2-Tape Non-deterministic TM M:

- guess a break in input $w = xy$
- Move y to second tape
- simulate M_1 on x, M_2 on y.
- Accept if both accepts.

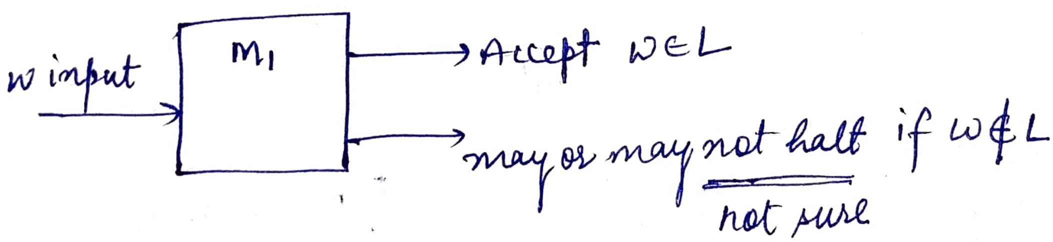
Complementation

RE languages are not closed under complementation.

Let L is recursive enumerable then $\bar{L}$ is not recursive enumerable.

Let $L = L(M_1)$.

Let M_1 be a TM such that



So we can not create a TM such that $\bar{L} = L(M')$

because we will not be sure that M' accept and halt if $w \in \bar{L}$ and $w \notin L$