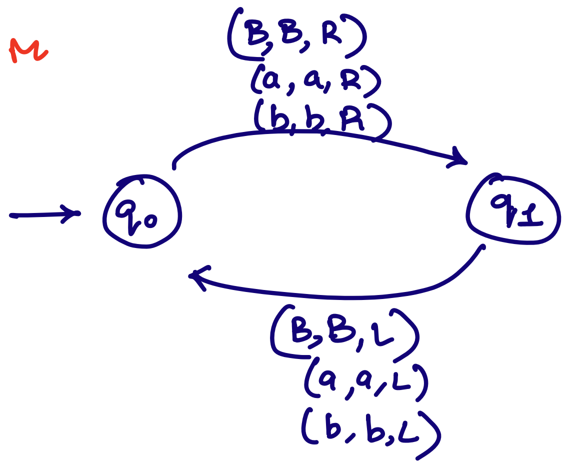
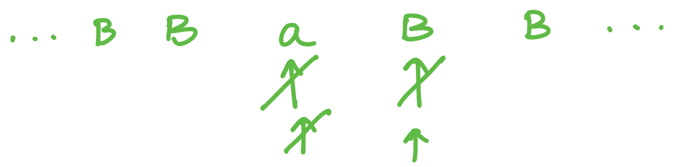
Non Halting TM

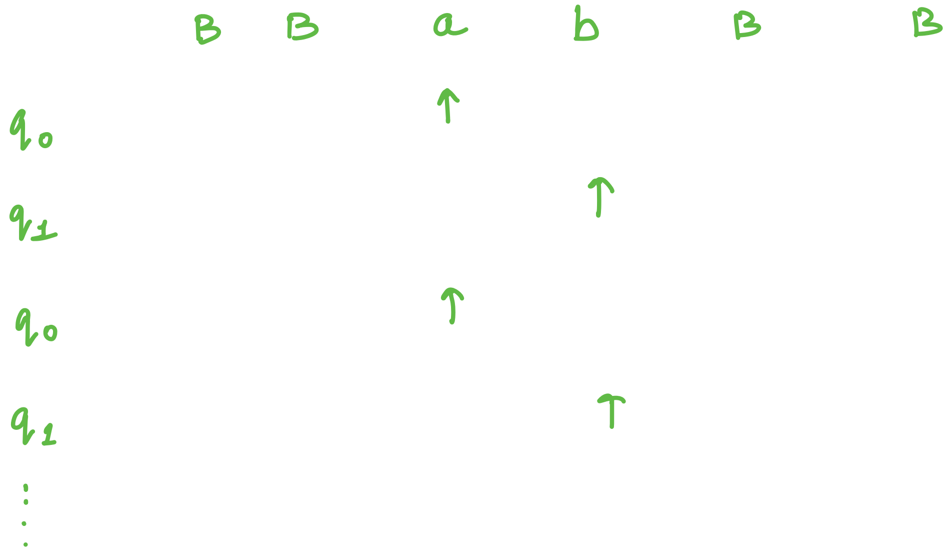


Eg:



$q_0 \xrightarrow{a} q_1 \xrightarrow{B} q_0 \xrightarrow{a} q_1 \longrightarrow \cdots$

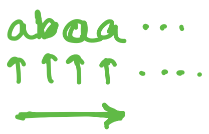
Eg:



non halting TM

In FA, PDA we didnot had non halting problem

↳ bcz in input string move only in 1 drⁿ



TM: input: left right (both $dr^{n}s$)

Input string in the language: TM will definitely halt and it will halts at a final state

IS: aabb

- a^nb^n TM
- even length strings TM

Input is not in the language.

a^nb^n: TM
abbb: IS

TM halts at a non final state

TM never halts.

Problem

Case 1:
TM is doing computations & these computations are taking some time

U feel: It is stuck in ∞ loop
Stop the TM

Case 2:
TM is stuck in ∞ loop.

U feel:
Computation

Problem with non halting TMs

you donot know how long to wait

Halting Problem of TM.

Turing Thesis:

Turing is a scientist, hypothesis 1930.

Any computation that can be carried out by any mechanical means can be performed by a TM.

TM is as powerful as a Computer.

Take every problem that can be solved by a Computer and try to have a TM for it.

Computer Solve

$P_1 \rightarrow TM_1$

$P_2 \rightarrow TM_2$

$P_3 \rightarrow TM_3$

$\vdots \quad \vdots$

P_∞

Issue: ∞ no. of problems.

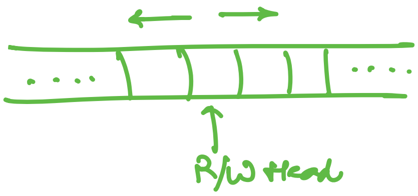
Alternatively, u come up with a problem which can be solved by a computer but can't be solved by TM.

→ Nobody was able to do this.

- TM & Computer are equally powerful
- People started believing that Turing thesis is correct

Modifications /Variants of Standard TM:

Standard TM:



Power of TM: language accepted by TM.

→ not the time complexity or space complexity

language
Power of DFA = NFA
Power of NPDA > DPDA
eg: ww^R

1 TM with stay option

Standard TM: left, right

Modified TM: will remain at the same place.

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

Power of this TM = Power of Standard TM

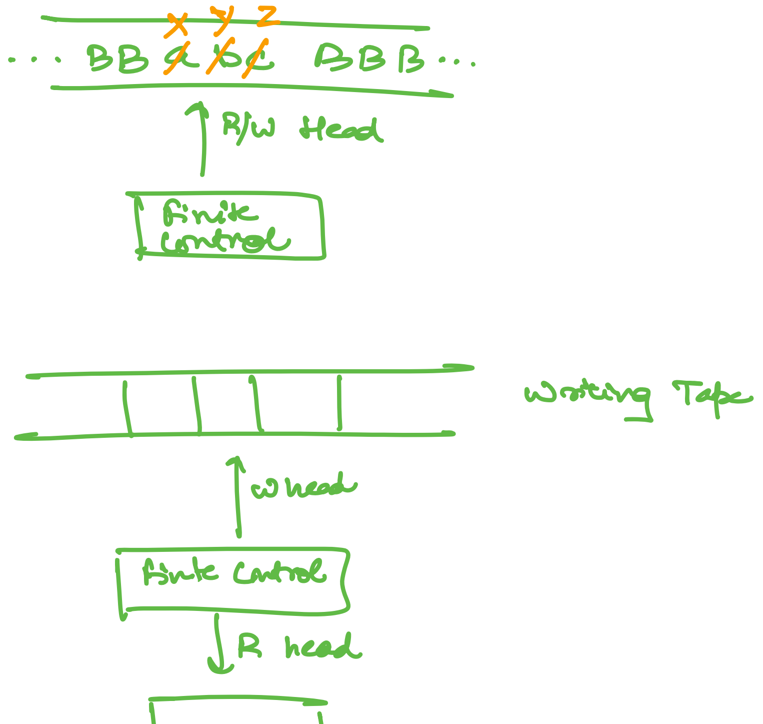
2 TM with semi infinite tape

Standard TM: ... BB abaa BBB ...

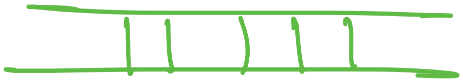
Semi-infinite TM: a b c BBB ....

3 Offline TM

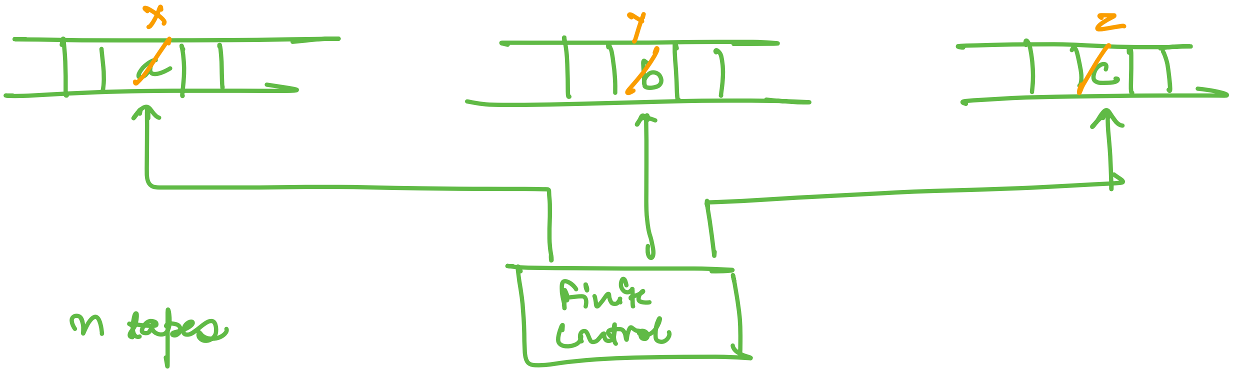
Standard TM:



4 Multitape TM

Standard TM: 

Multitape TM:



n tapes

$$\delta: Q \times \overset{abc}{\Gamma^n} \longrightarrow Q \times \overset{xyz}{\Gamma^n} \times \{L,R\}^n$$

n symbols from n tapes

new symbols in n tapes

Left, Right for n tapes.

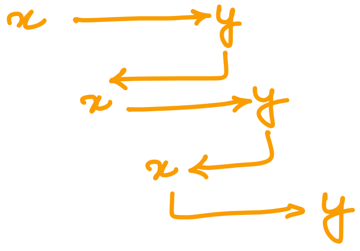
Power of Standard TM = Power of multitape TM

Benefit of Multitape TM is time is reduced.

Eg: $a^n b^n$

Standard TM

$a^3 b^3$: .... $\cancel{a}$ $\cancel{a}$ $\cancel{a}$ $\cancel{b}$ $\cancel{b}$ $\cancel{b}$ .....



Yu want to match n pairs

for every pair u have to move n steps.

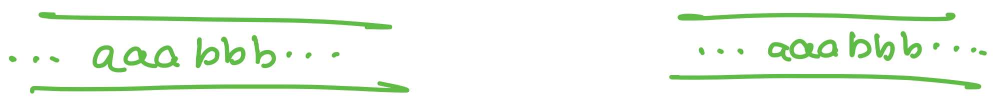
for n pairs u have to move n^2 steps.

$TC = O(n^2)$

Multitape TM:

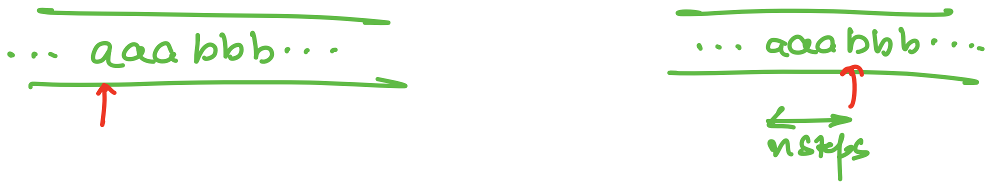
$a^3 b^3$

1. copy the entire input to other tape



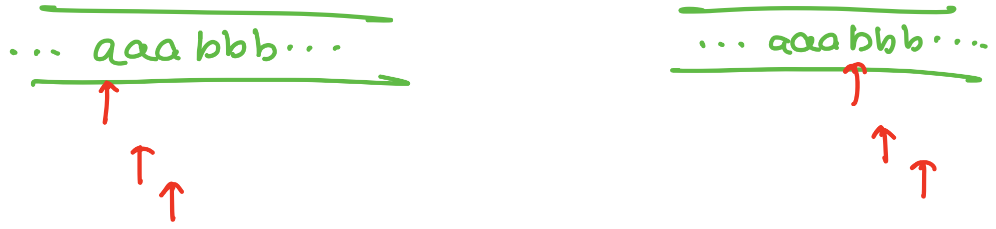
$O(n)$

2. set the Read/write head



$O(n)$

3. Scan symbols one by one in both the tapes

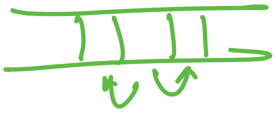


$O(n)$

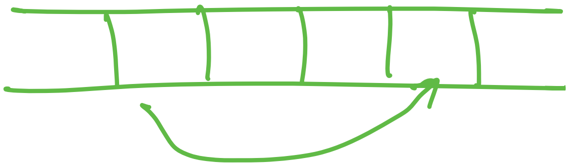
Total time: $3n = O(n)$

⑤ Jumping TM

Standard TM:



Jumping TM:

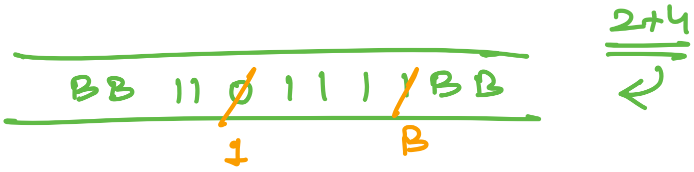


$\delta : Q \times \Gamma = Q \times \Gamma \times \{L, R\} \times \{n\}$

↓ steps in jump

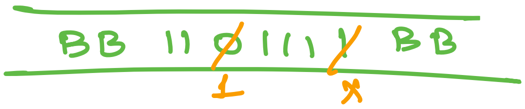
⑥ Non Erasing TM

Standard TM:



Input Symbol ⇔ blank

Non Erasing TM: Remove the option of changing input to blank.

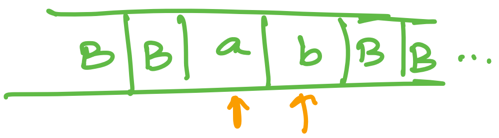


→ not a blank

→ for us, x is a blank

⑦ Always writing TM

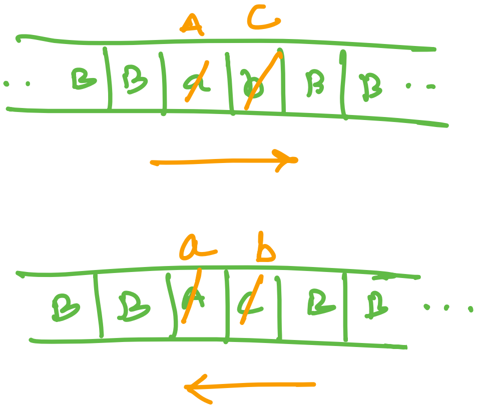
Standard TM:



You may not change the tape alphabet

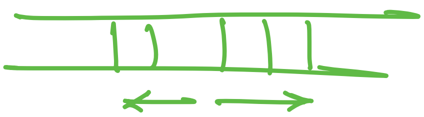
Always writing TM:

Definitely change the input alphabet



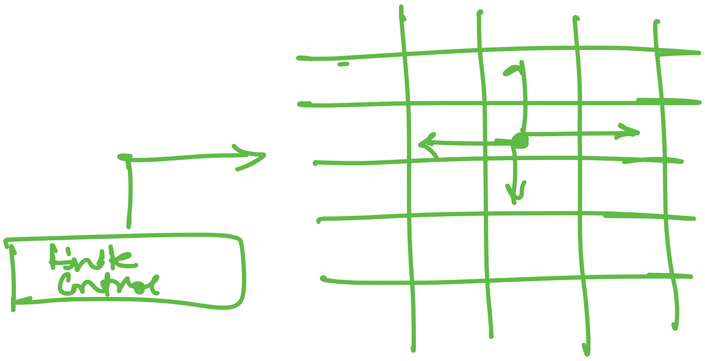
8 Multi dimensional TM

Standard TM:



Power is same

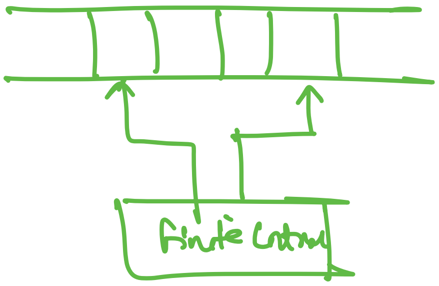
Multi dimension:



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

9 Multihead TM

Single tape, read the content from multiple places at same time



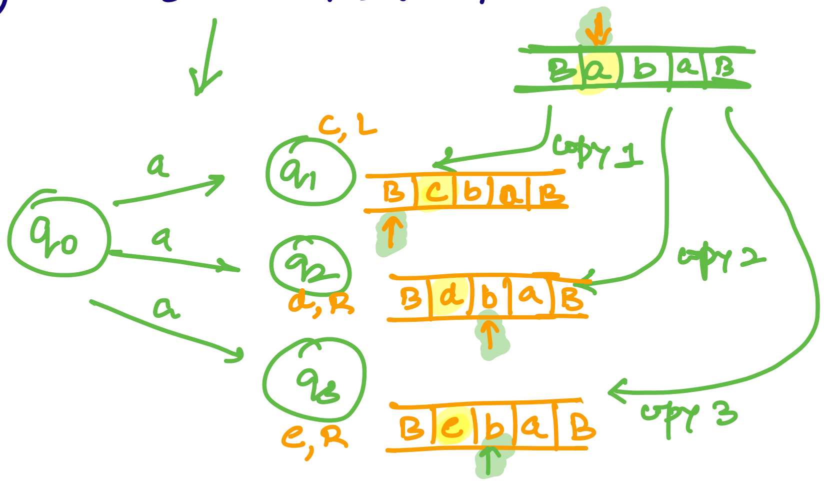
10 Automata with queue

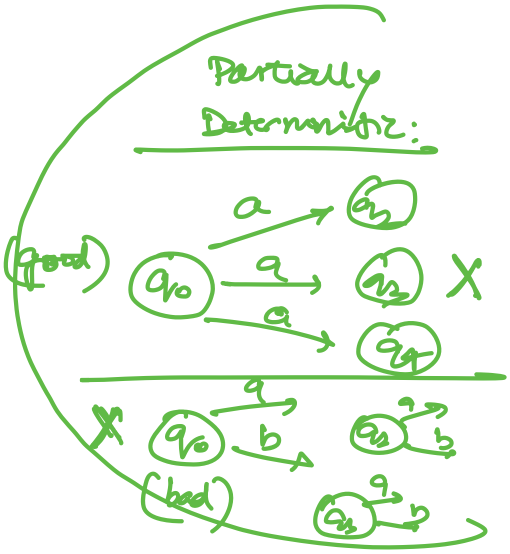
TM: Automata + Queue

11 Any TM can be minimized to a TM with only 3 states

12 Any Standard TM can be converted to a multitape TM with Stay option and atmost 2 states.

13 Non deterministic TM





On looking at one state and one symbol we can make multiple copies & can simultaneously go on many stats and can change the tape symbol

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

Non Deterministic TM & Deterministic TM have equal power.

DTM ≅ NTM

Power:

DFA : NFA
DTM = NTM

NPDA > DPDA
Eg: ww^R

UNIVERSAL TURING MACHINE

Turing Thesis: TM is as powerful as a computer.

Computer: can run any program

Addition · * · Division

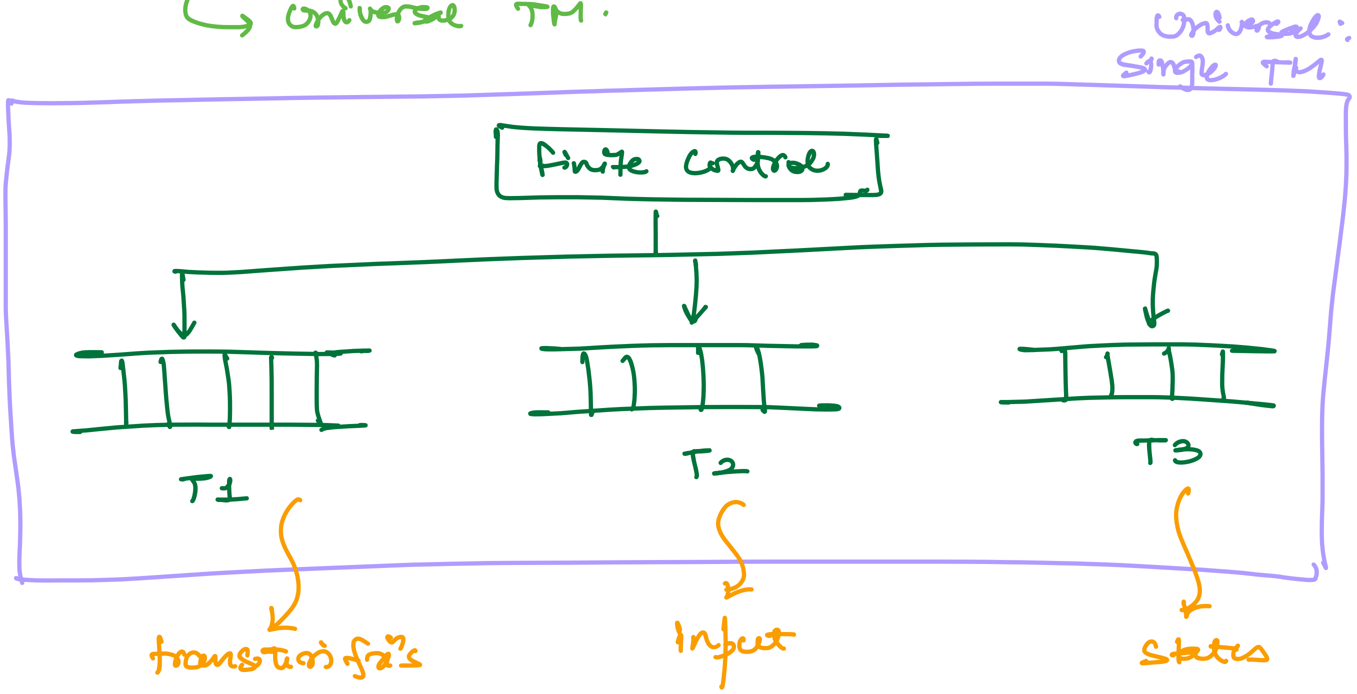
can solve multiple problems.

TM works for a single problem

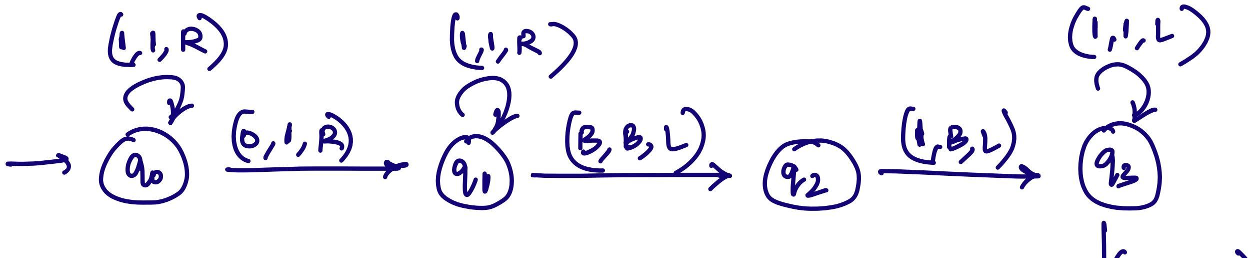
a^nb^n → TM1

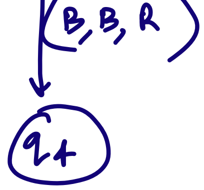
a+b → TM2

Aim: 1 TM which can solve every problem.
↳ universal TM.



a+b:





2+4:

11 0 1111 1 Tape 2 : input

Tape 3: states

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$\Gamma = \{a_1, a_2, a_3 \cdots \}$

Encode every symbol

$q_0 \to 1$	$a_1 \to 1$
$q_1 \to 11$	$a_2 \to 11$
$q_2 \to 111$	$a_3 \to 111$
$q_3 \to 1111$	$\vdots$
$\vdots$	

Tape 1: Transition fx^{n}s

$\delta(q_1, a_1) = (q_2, a_2, R)$

$\underbrace{11}_{q_1}\ 0\ \underbrace{1}_{a_1}\ 0\ \underbrace{111}_{q_2}\ 0\ \underbrace{11}_{a_2}\ 0\ \underbrace{11}_{R}$

L → 1
R → 11

Entire transition fx^{n} can be written as a string of 0's & 1's.

Entire TM can be represented as string of 0's & 1's.

TM is one of the string of Σ^* $\Sigma = \{0,1\}$

Not every string of 0's & 1's is a TM.

Machine → language

FA → Regular Lang.

PDA → CFL (Context free language)

TM → RE or Recursive

TM is powerful than PDA.

CFL is a subset of RE language.

Exception

TM doesnot have power to accept ε, but ε can be accepted by PDA.

CFL are a subset of RE language (not considering ε)

$\underbrace{\text{Recursively Enumerable}}_{\text{RE}}$ and Recursive language:

Language accepted by TM is called as RE language.

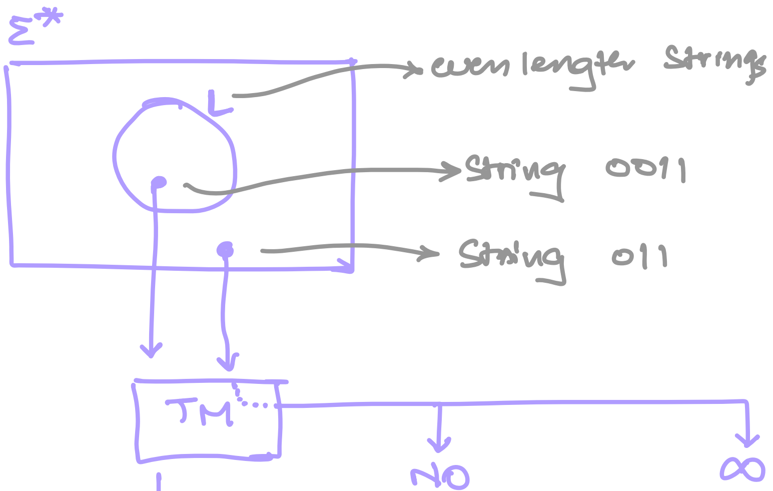
RE V/S Recursive:

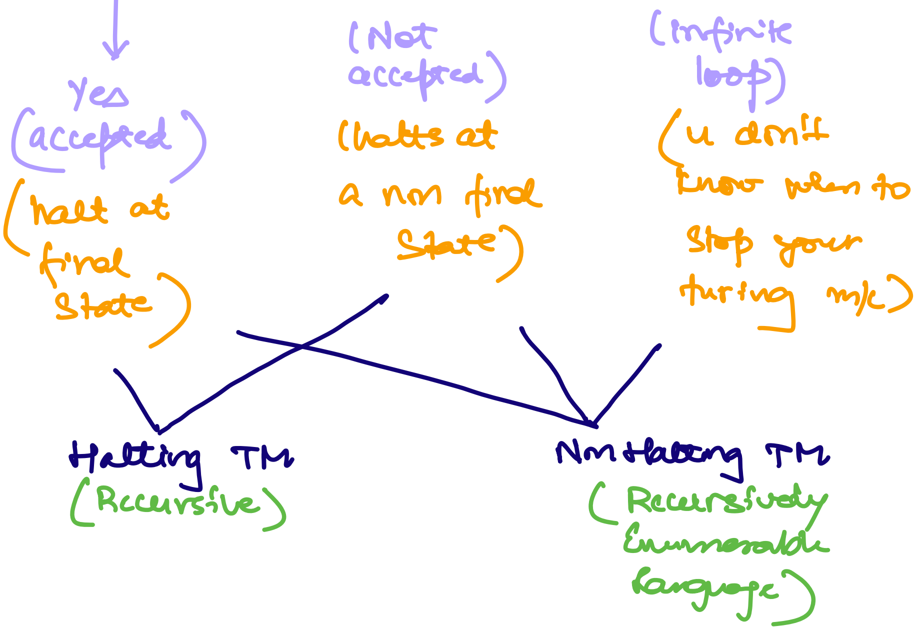
{0,1}

Σ^* = set of all strings possible → 0, 1, 00, 01, 10, 11, 100 .....

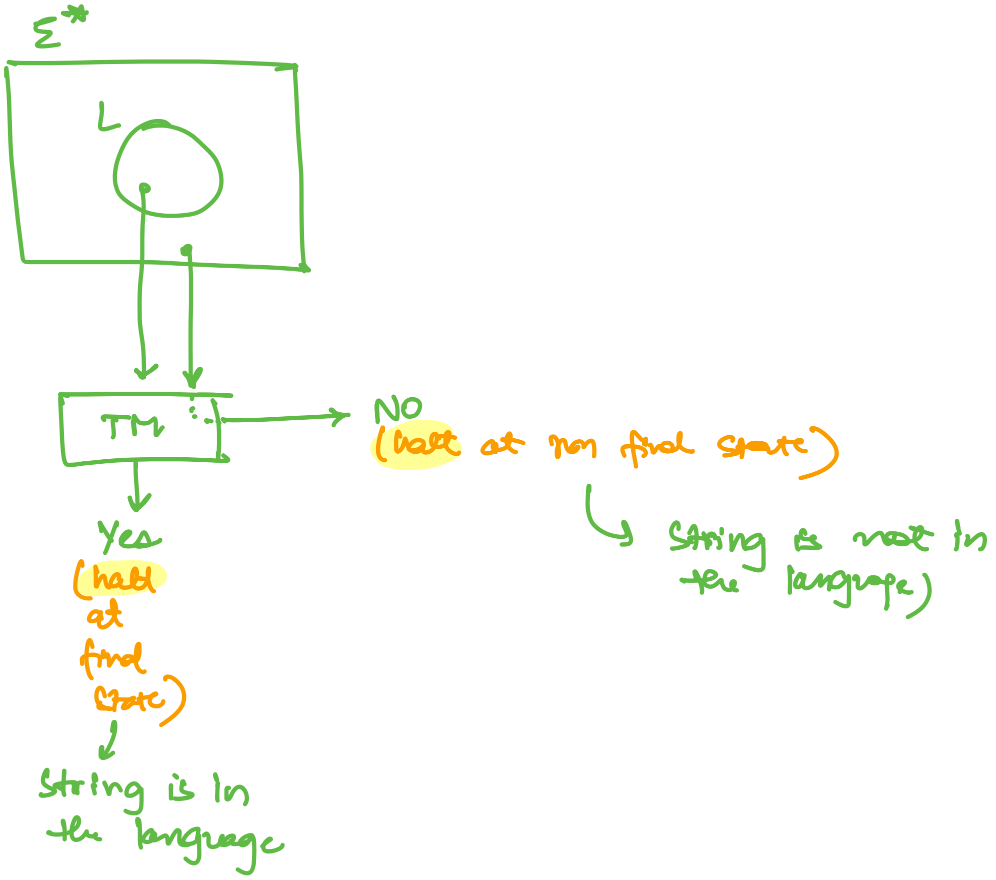
L = subset of Σ^* → L: even length

00, 11, 0000, 1111 ......

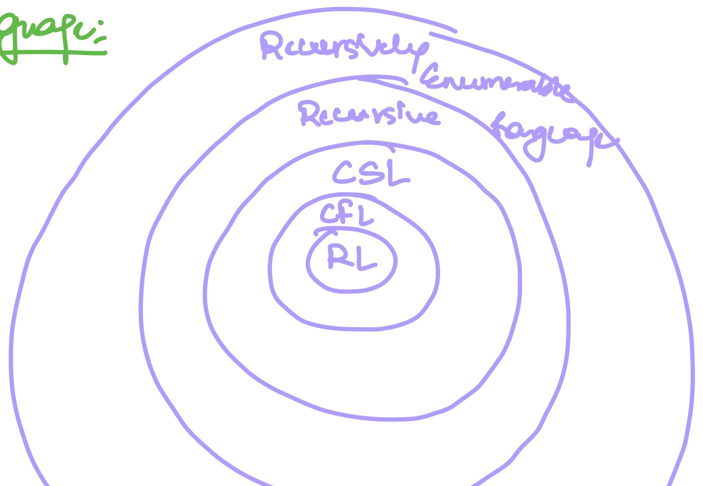



Yes
(accepted)
(halt at final State)
(Not accepted)
(halts at a non final State)
(infinite loop)
(u don't know when to stop your turing m/c)
Halting TM
(Recursive)
Non Halting TM
(Recursively Enumerable language)

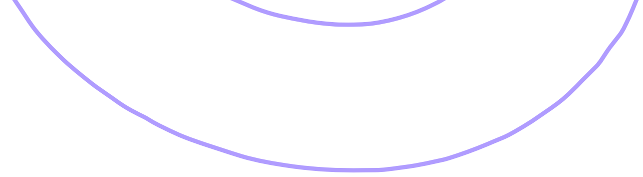
Halting TM


Σ*
L
TM
NO
(halt at non final state)
String is not in the language)
Yes
(halt at final State)
String is in the language

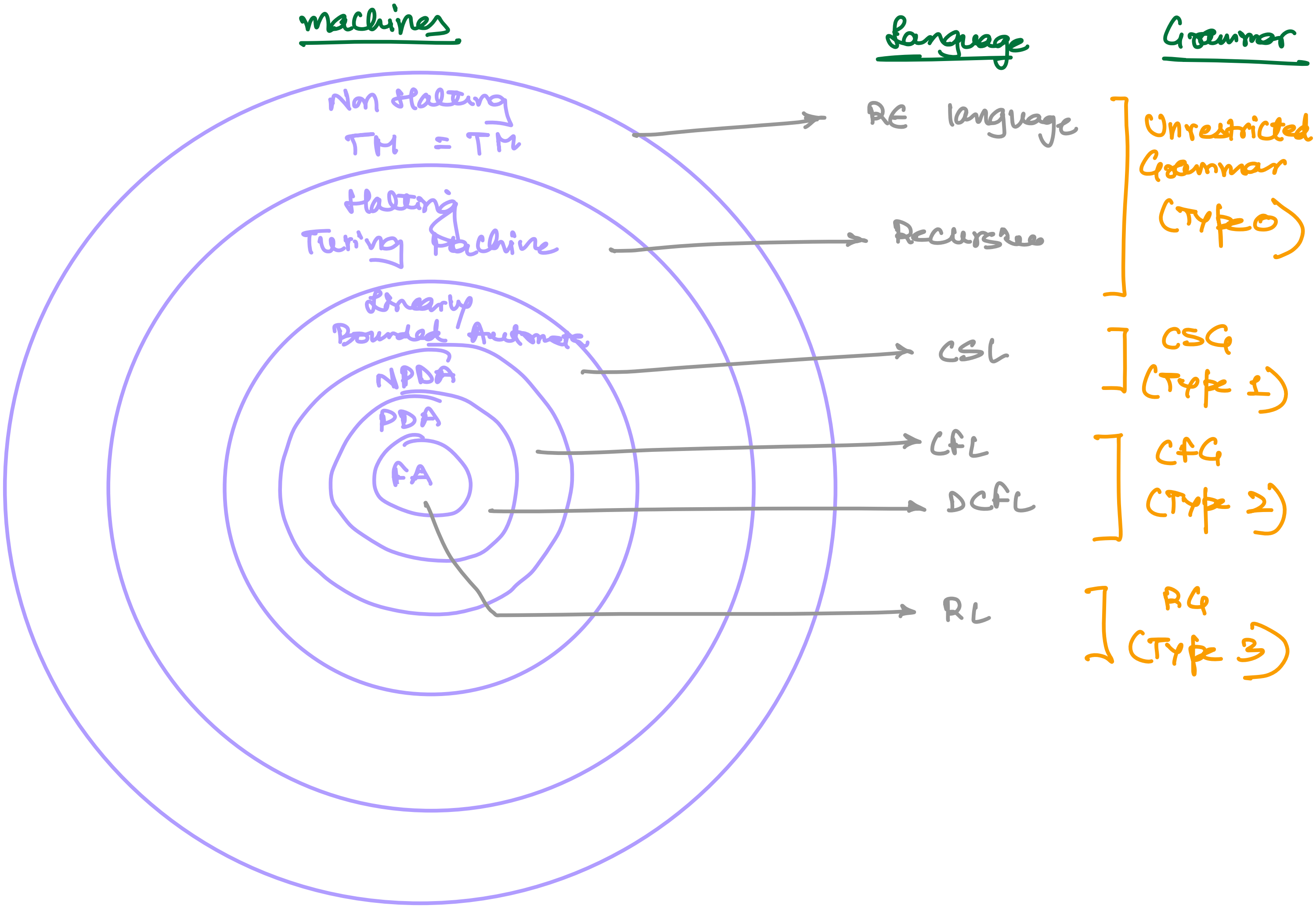
Language:



Every Recursive language is Recursively Enumerable.



Big Picture



Every Recursive language is RE
Every CSL " is Recursive
" CFL " is CSL
" RL " is CFL

Unrestricted Grammar:

A grammar is called unrestricted grammar if all the productions are of the form $u \rightarrow v$

$u \in (V \cup T)^+$

$v \in (V \cup T)^*$

V = variables
T = Terminals

→ LHS of productions can't be empty.

LHS can't be ε.