- Turing Machine
- Recursively Enumerable (RE)

PS

These are my class notes.
Empty pages indicate I was absent
or highly confused/sleepy.
Read relevant topics from book for the same.

Recursively Enumerable:

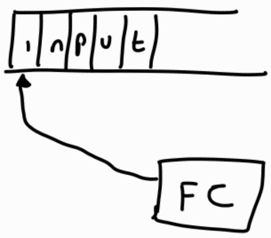
languages like $L = a^n b^n c^n \mid n \geq 1$ which are not CFG are included in RE.

Turing machine defined by 7 tuple, $M = \{Q, \Sigma, X, \delta, q_0, B, F\}$

X ↓ Tape Alphabet

B ↓ Blank.

Tape



FC- Finite Control

In case of PDA of an FA, tapehead will move only to 1 direction i.e right but turing machine can move left or right or even rewrite the cell.

Turing Machine Halts:

1] Final Halt:

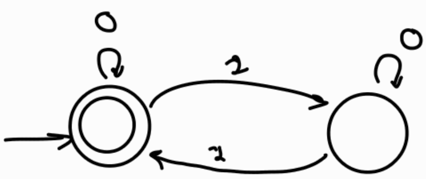
- Accepted by final state

2] Non - final Halt:

Turing machine won't be in final state. I/p won't be accepted.

3] Looping (neither accepted nor rejected.]

Q. Design a TM over binary numbers to accept all strings of even number of 1's.

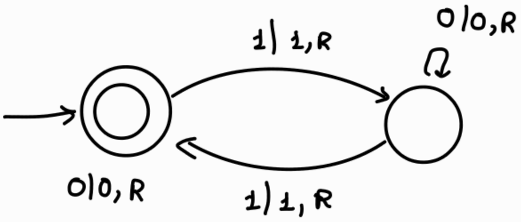


FA

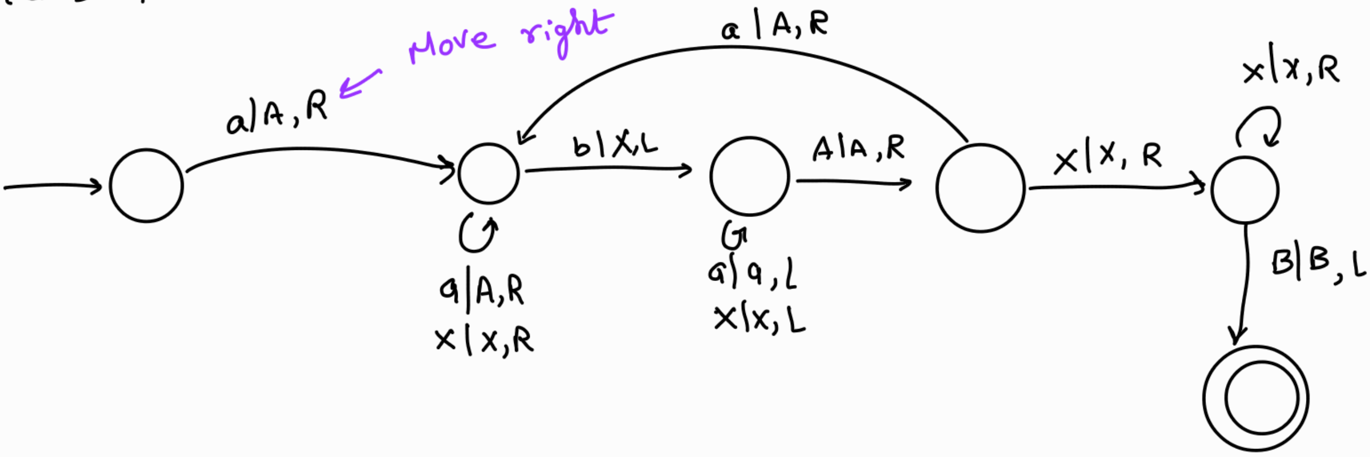
⟹

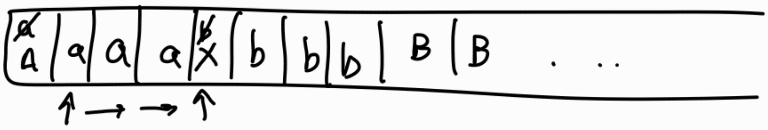
difference is in how we write.

⇓



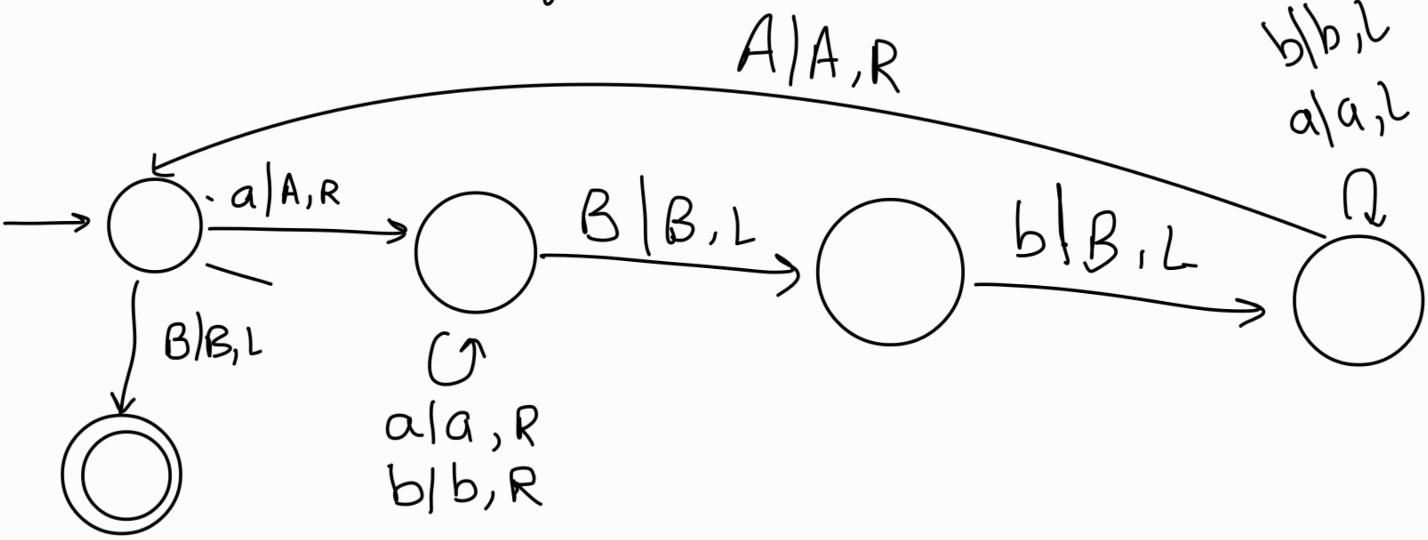
$L = \{a^n b^n \mid n > 0\}$

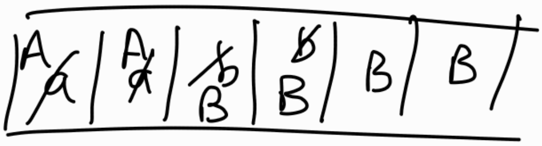


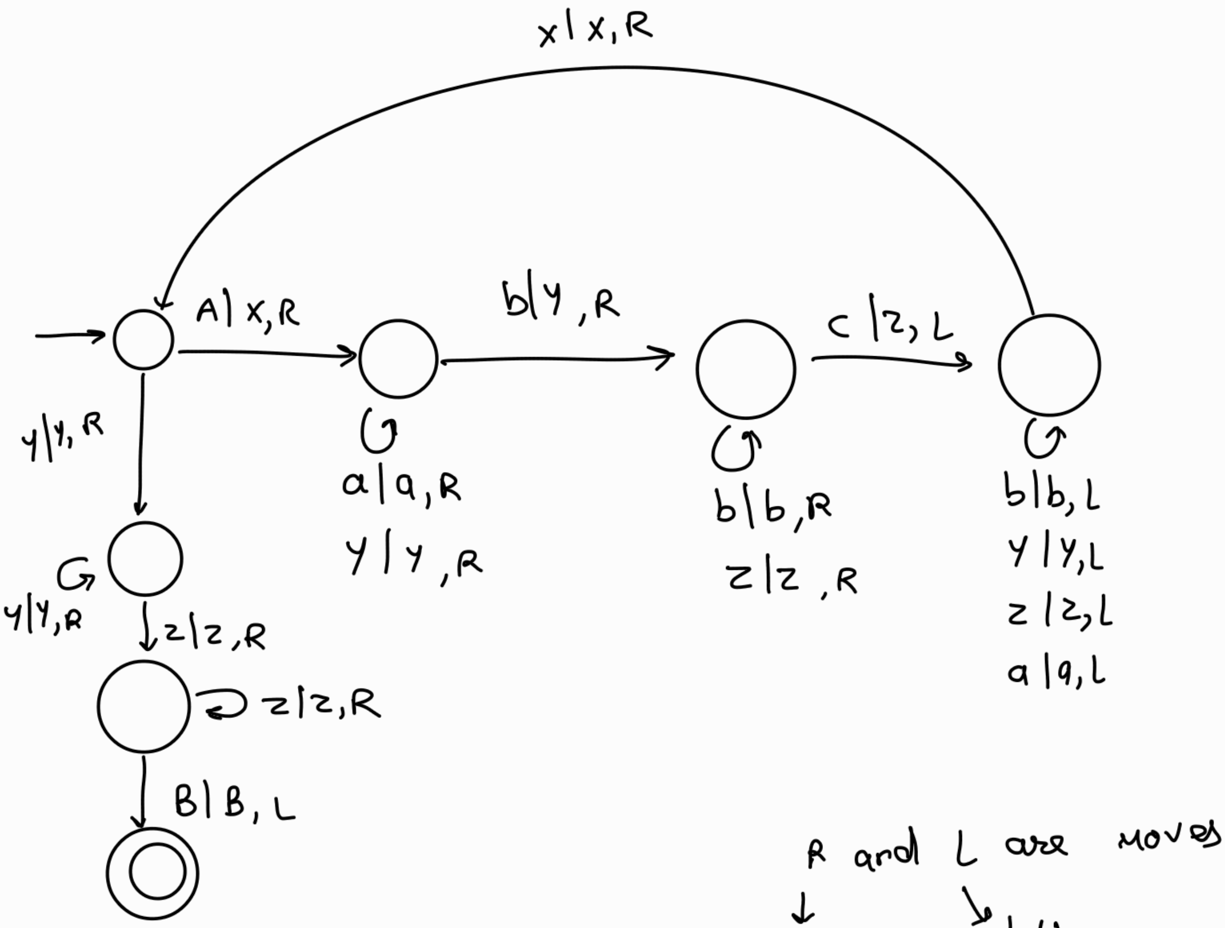


• Working:

1) a is found, replace a, Move right till B is present then replace b with X go left till you find back A.

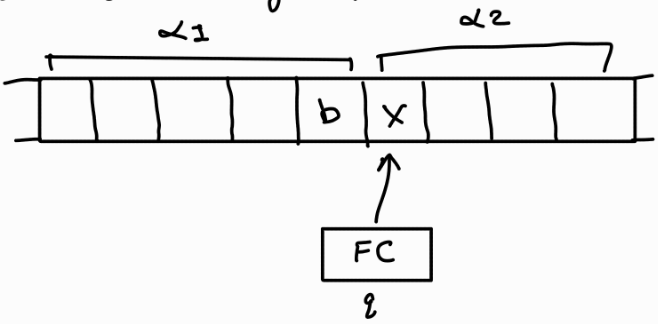






R and L are moves
Right left.

Configuration of a Turing Machine

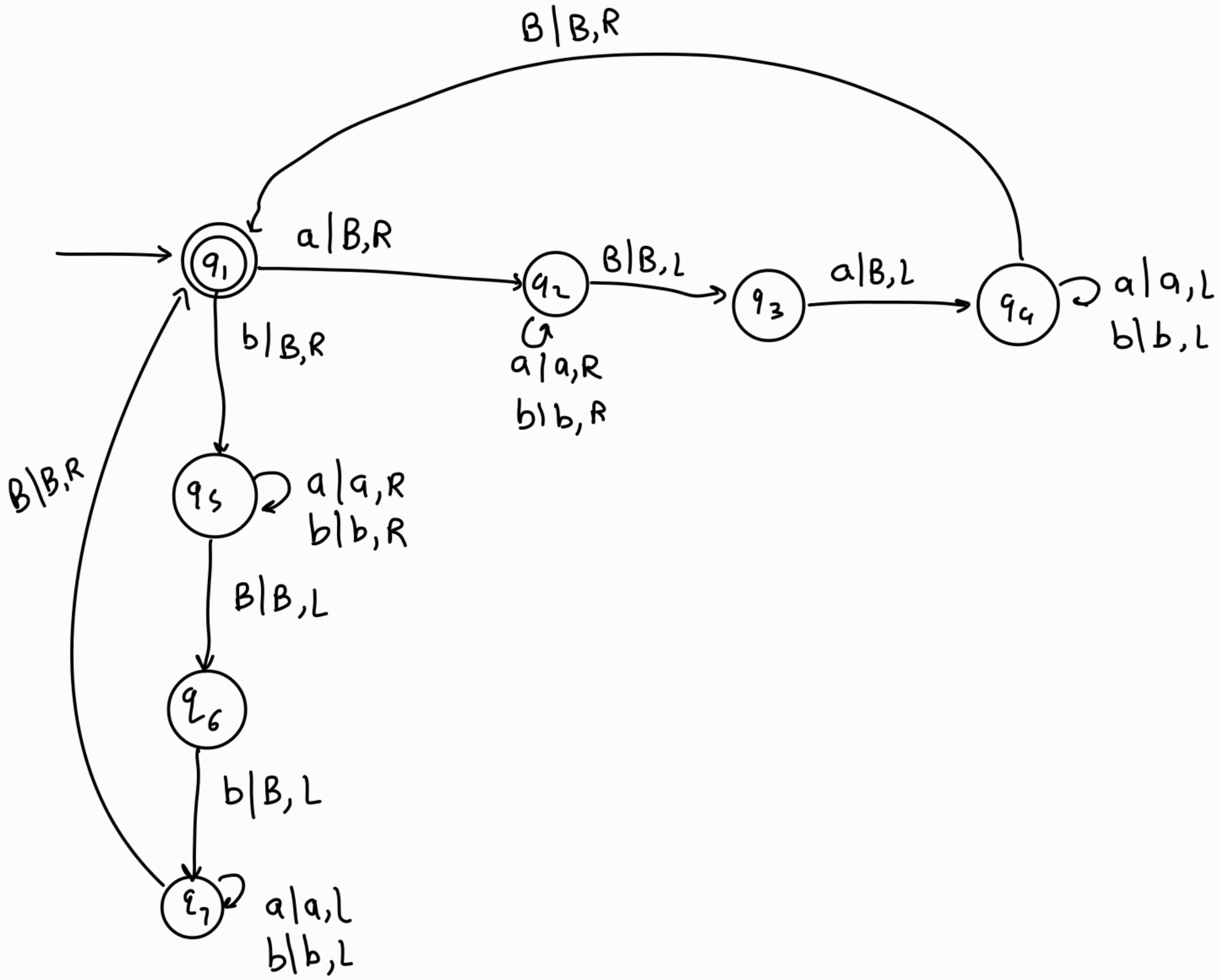


q is current state, tapehead is placed on first cell/character of α_2 & α_1 is cell info to the left of α_2.

$$\delta(q, b) \vdash (q', b', R)$$

$$(q, \alpha_1, b\alpha_2) \vdash (q', \alpha_1, b', \alpha_2)$$

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



$(q_1, B, abba) \vdash (q_2, B, bbaB) \vdash (q_2, b, baB)$

$\vdash (q_2, bb, aB)$

$\vdash (q_2, bba, B)$

$\vdash (q_3, bb, aB)$

$\vdash (q_4, b, b)$

$\top$

(q_4, B, bb)

$\top$

(q_4, B, Bbb)

$\top$

(q_1, B, bb)

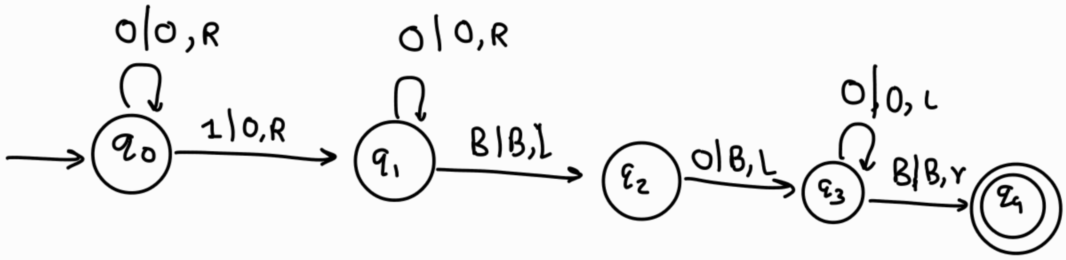
$\vdash ($

The language $L(M)$ accepted by turing machine is a set of string S.T. $(q_0, \in, w)$ by M moves of the Turing Machine will reach some $(P_1, \alpha_1, \alpha_2)$ where $P \in F$, 'α_1, α_2' $\in X^*$.

A language is Recursively enumerable (RE) if it is accepted by a turing machine

Construct a TM to add two unary numbers.

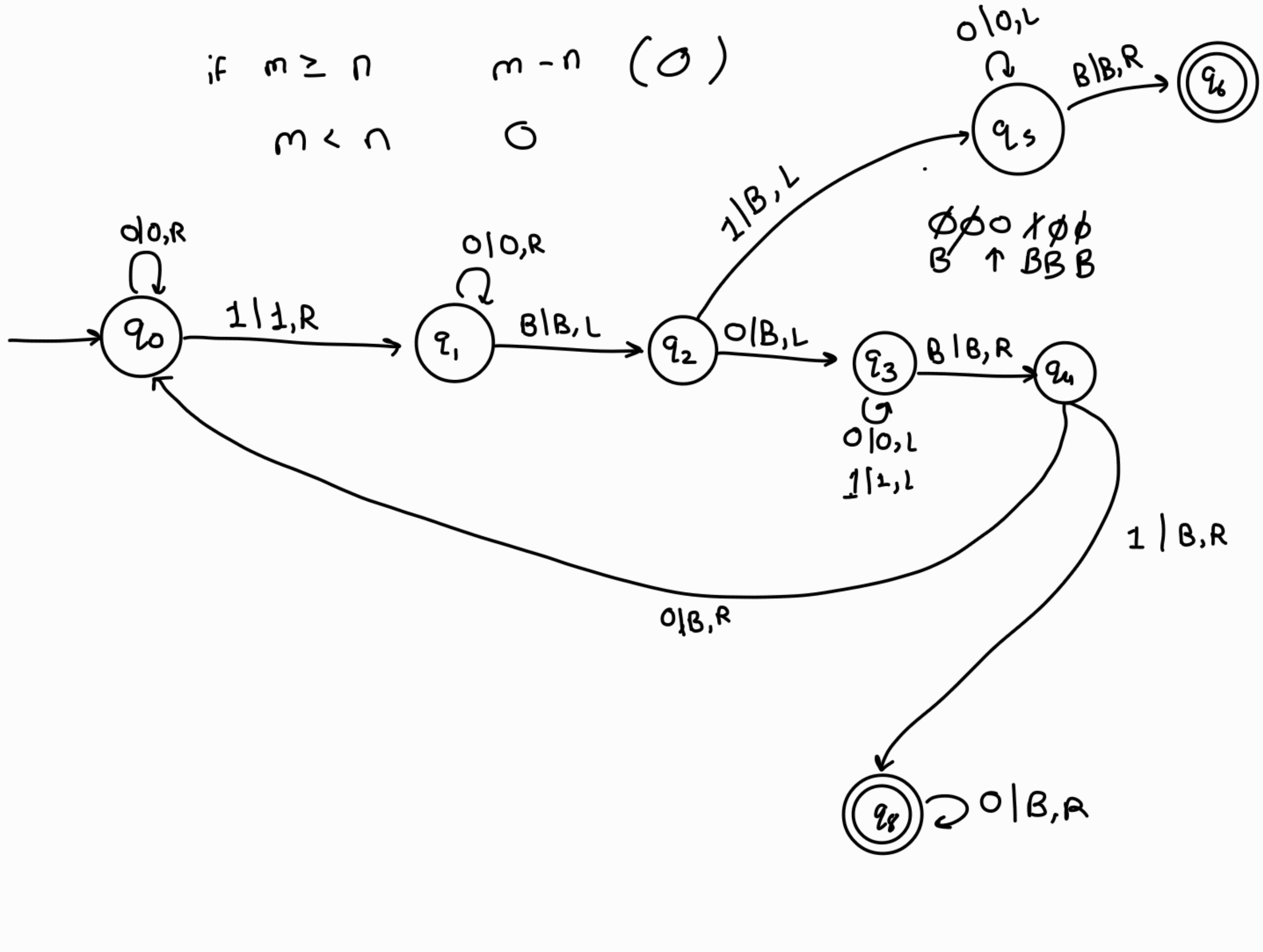
$0^m 1 0^n$

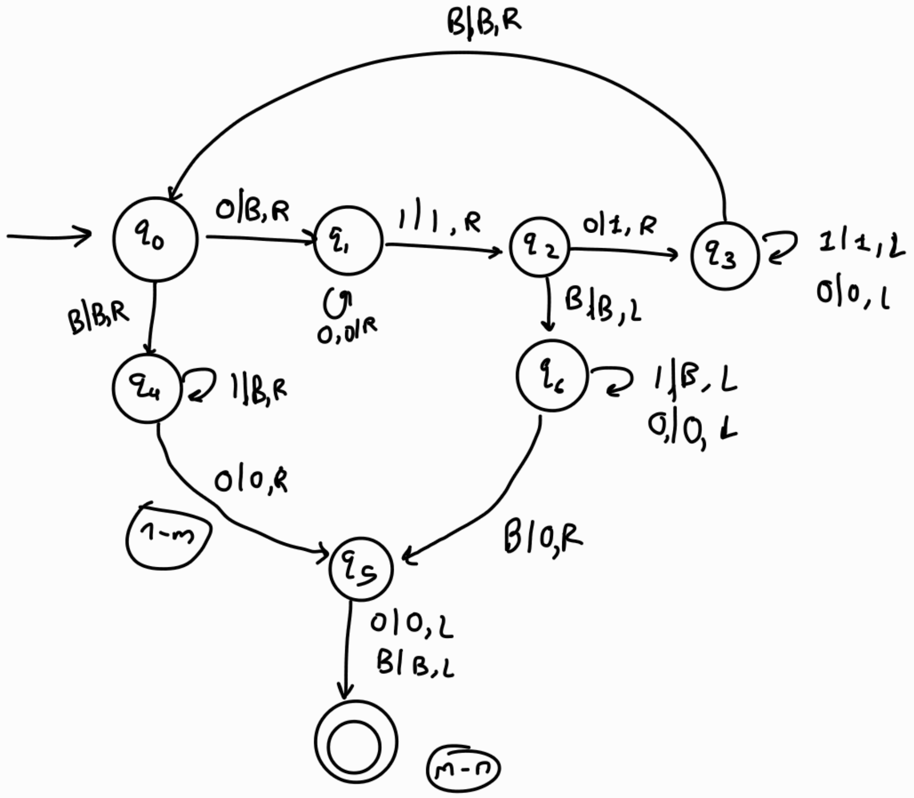


$0^m 1 0^n$

if $m \geq n$ $\quad m - n \;(0)$

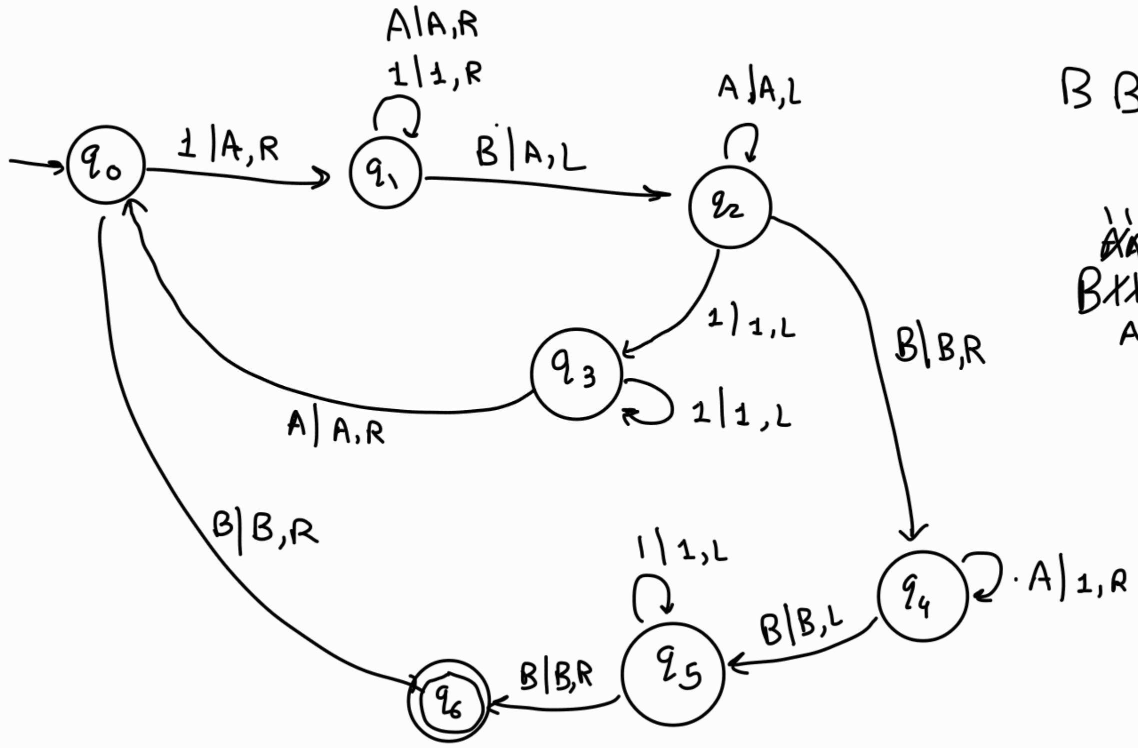
$m < n \quad 0$





$q_0\, \omega \vdash q_f\, \omega\omega \mid \omega \in \{1^*\}$

if ω=111



ww^R

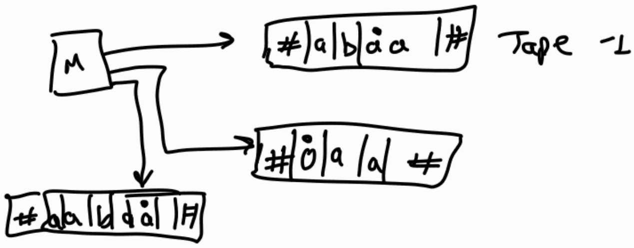
000100

Every multi tape TM has an equivalent one tape TM. IMP Proof

Slide pg. 6

Proof

→ delimiter → #



We put a • 'dot' here to indicate tape head.

M' → Makes two passes → Pass 1 → finds position of tape head.

↓ traversals through TM.

Pass 2 → Updates tape cell at tape head (i.e. performs action) as defined by trans. func.

? what is M and M'?

Non Deterministic TM:

Transitions → $\delta : Q \times \Gamma$ power set of $Q \times \Gamma (L \times R)$

as Non deterministic, represented as tree.

- Each branch is a branch of non determinism.
- You must do 'Breadth First Search'.

Theorem: Every non deterministic TM can be simulated using TM.

Imp.

PPT. pg 13

Tape 1 — input →

M'

input tape (1) | Simulation Tape (2) | Address Tape (3)

Enumerator → TM with a printer (jaisa)

Undecidability.

↳ Decision problems

Decidability problems:

- Answer is "Yes" or "No"

Categories.

(*)

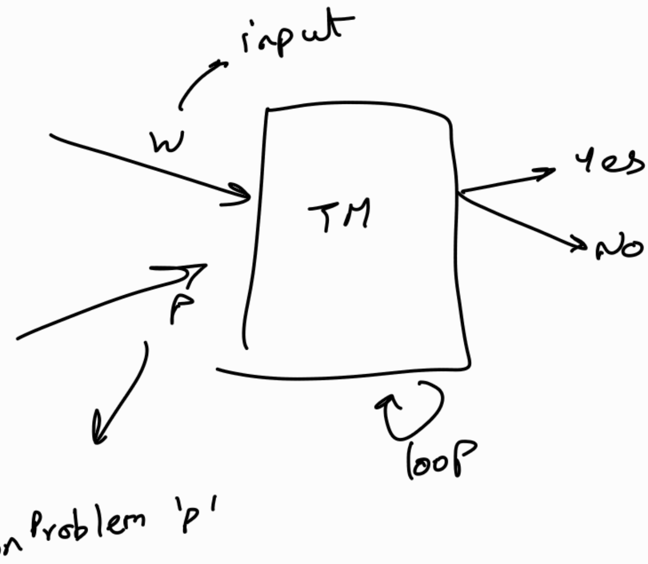
TM recognisable
OR
Recursively Enumerable

↑
TM design possible.

Non recursively enumerable
OR
Non TM recognisable.

TM decidable → It will surely halt in a finite amount of time.

TM non decidable



Recursive languages are decidable.

Church's Thesis.

X ← read in ppt.

Decision problem:

Do theorems: read them out at least one.

Undecider TM:

for some i/p loops forever in case of rejection. e.g. slide 13

e.g. undecider → Reg expr. $1(0+1)^*$

RE but not recognisable.

Two popular Undecidable Problems

Halting Problems

PCP

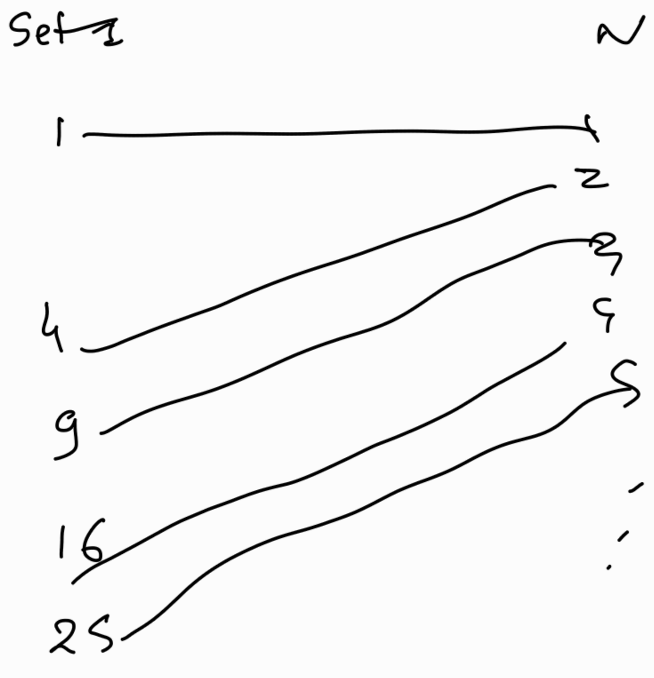
Encoding TM.

Diagonalization

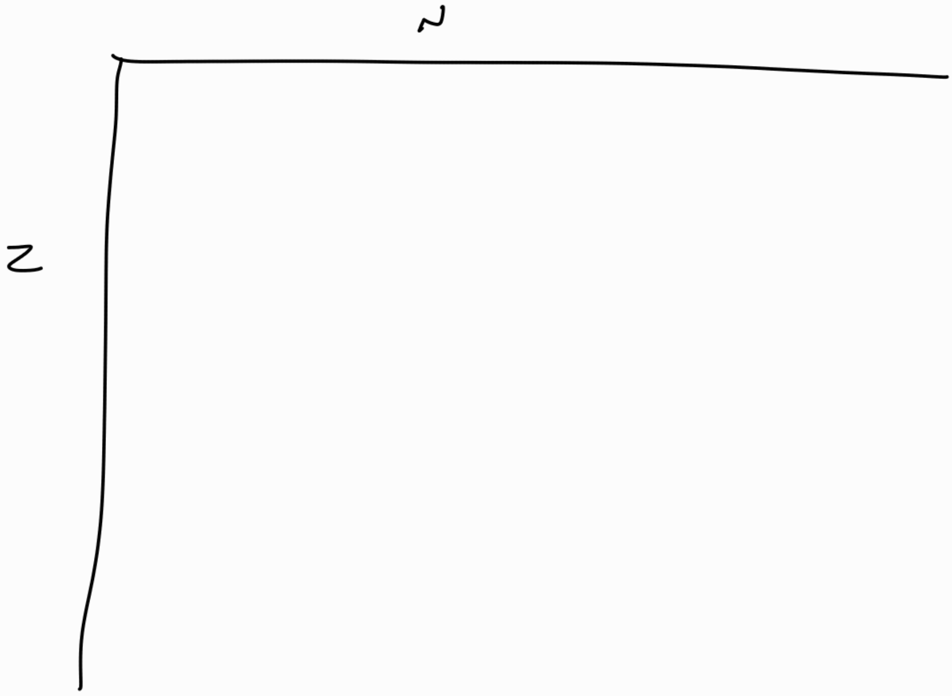
↳ bijunction (onetoone & onto!)

Countable infinite set:

A set is countable if there exist a bijection between it & set of naturals.



Set of squares, countable infinite set



i^{th} row, j^{th} column

Look inSipsee