# FLAT

**Turing Machines** - to accept recursively enumerable languages.

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

$\Gamma$ → tape symbol

$B$ → blank

eg. $L = \{0^n 1^n \mid n \geq 1\}$

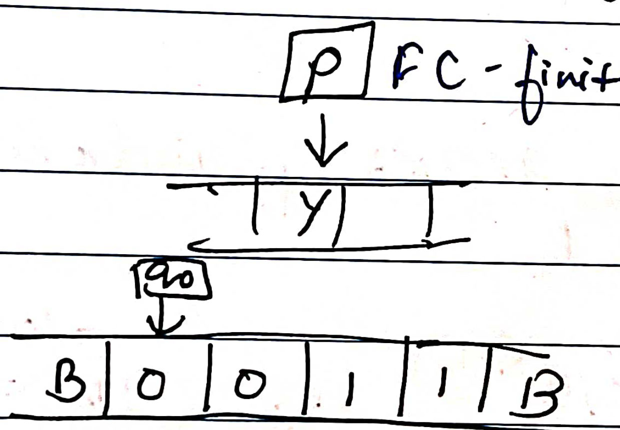
make $0 \to X$

then make $1 \to Y$

$0 \to X$

$\delta(q, X) = (P, Y, D)$

$D$ → direction




$\delta(q_0, 0) = (q_1, X, R)$

$\delta(q_1, 0) = \delta(q_1, 0, R)$

$\delta(q_1, 1) \quad (q_2, Y, L)$

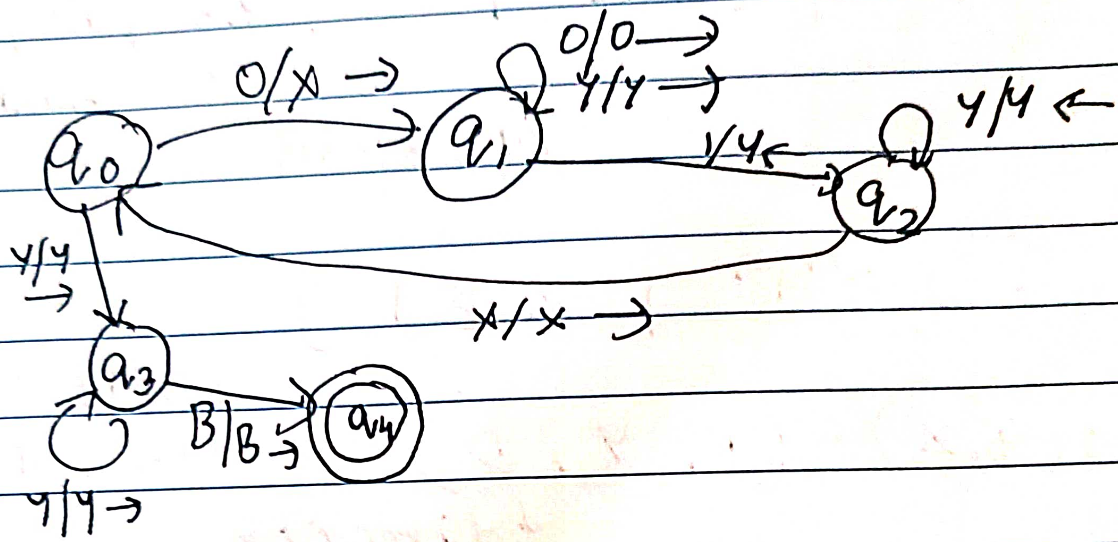
$\delta(q_2, 0) = (q_2, 0, L)$

$\delta(q_2, X) = (q_0, X, R)$

$\delta(q_0, Y) = (q_1, Y, R)$

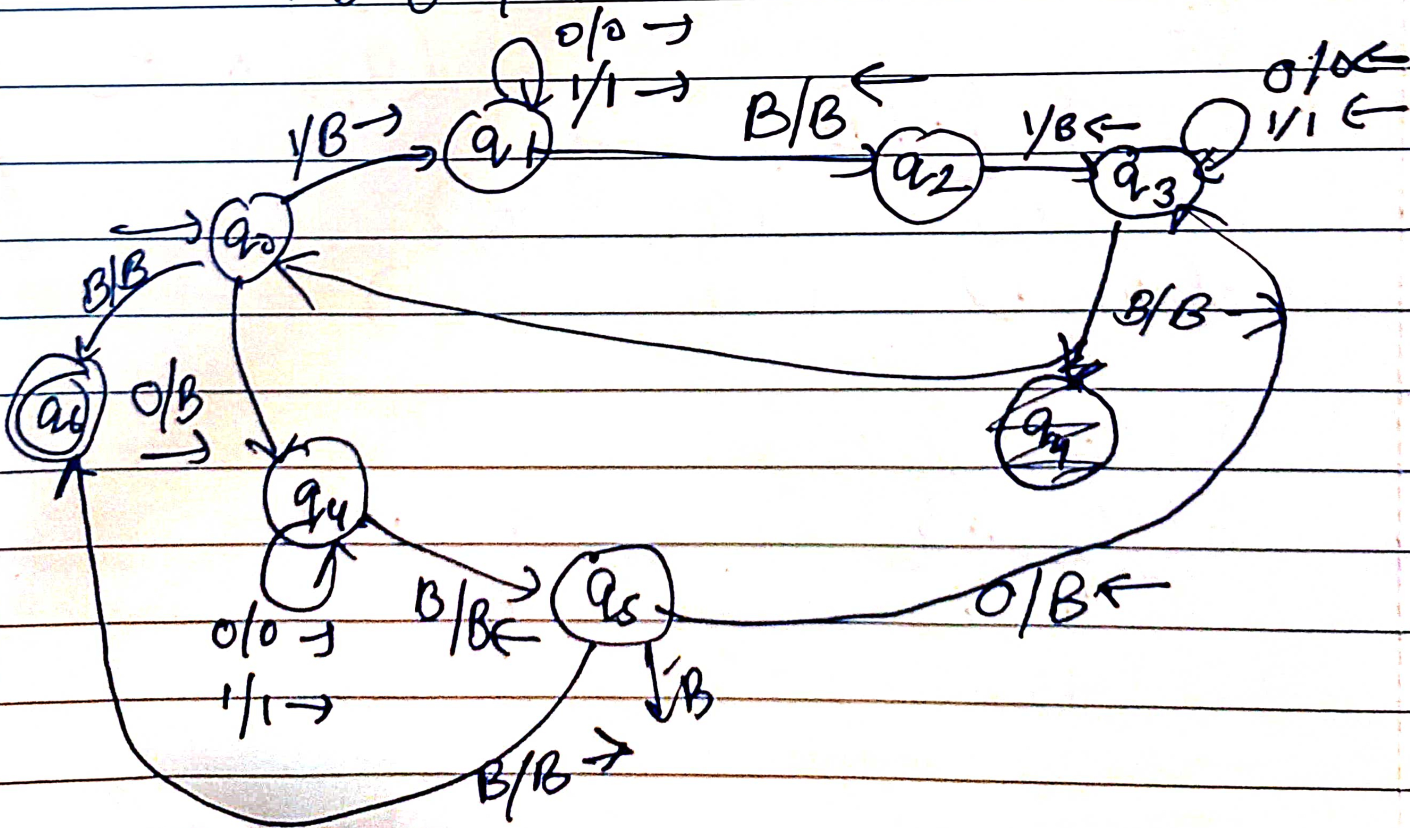
$\delta(q_2, Y) = (q_2, Y, L)$

$(q_0, Y) = (q_3, Y, R)$
$\delta(q_3, Y) = (q_3, Y, R)$
$\delta(q_3, B) = (q_4, B, R)$



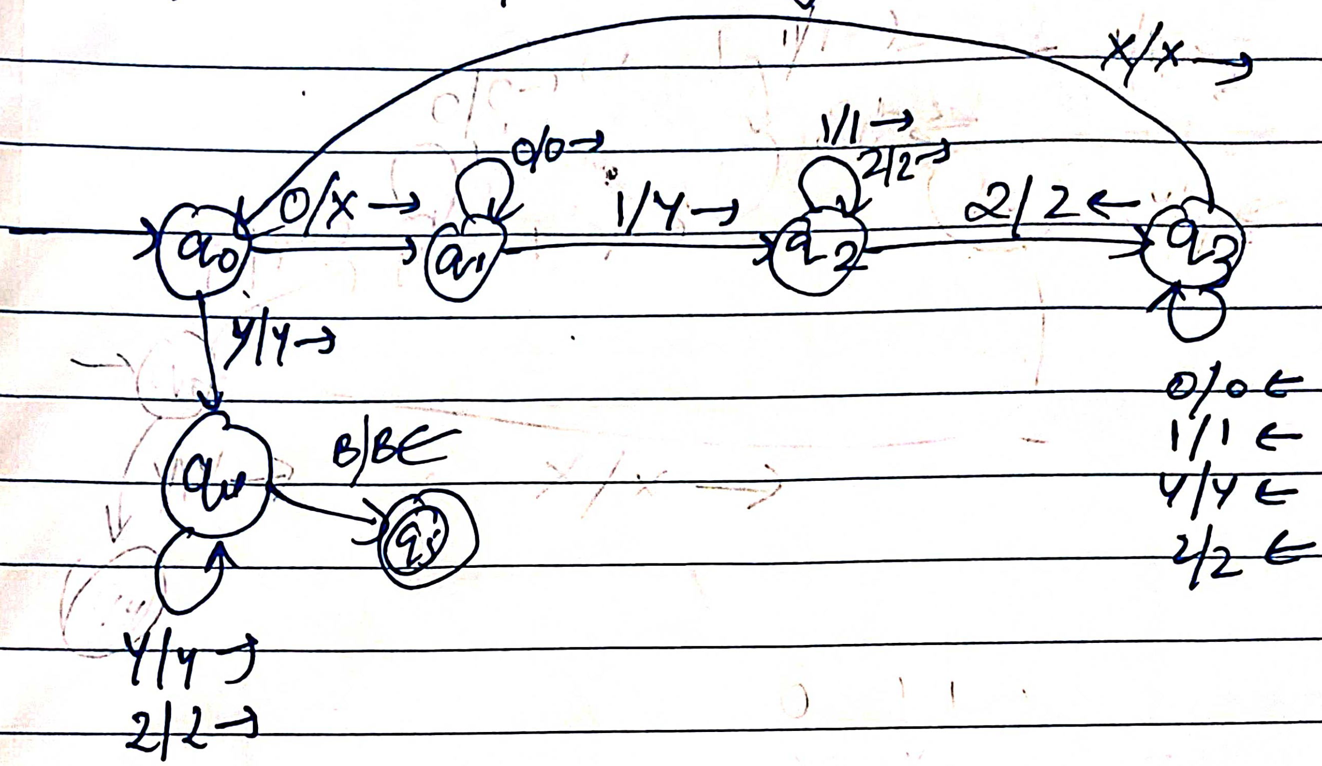

Q. Language that accepts palindromes.

1 0 1 1 0 1



0 0 1 1 2 2

$\{0^n 1^n 2^n \mid n \geq 1\}$




## Mathematical Operations

$0^x B 0^y$

add by counting

B 0 0 B 0 0 0

⇓

B 0 0 0 0 0

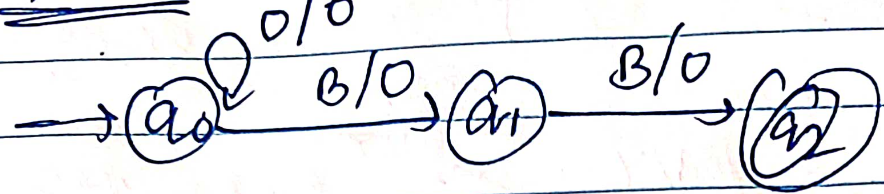
$x-y$ assuming $x > y$

I/P 0000 B 0 0.

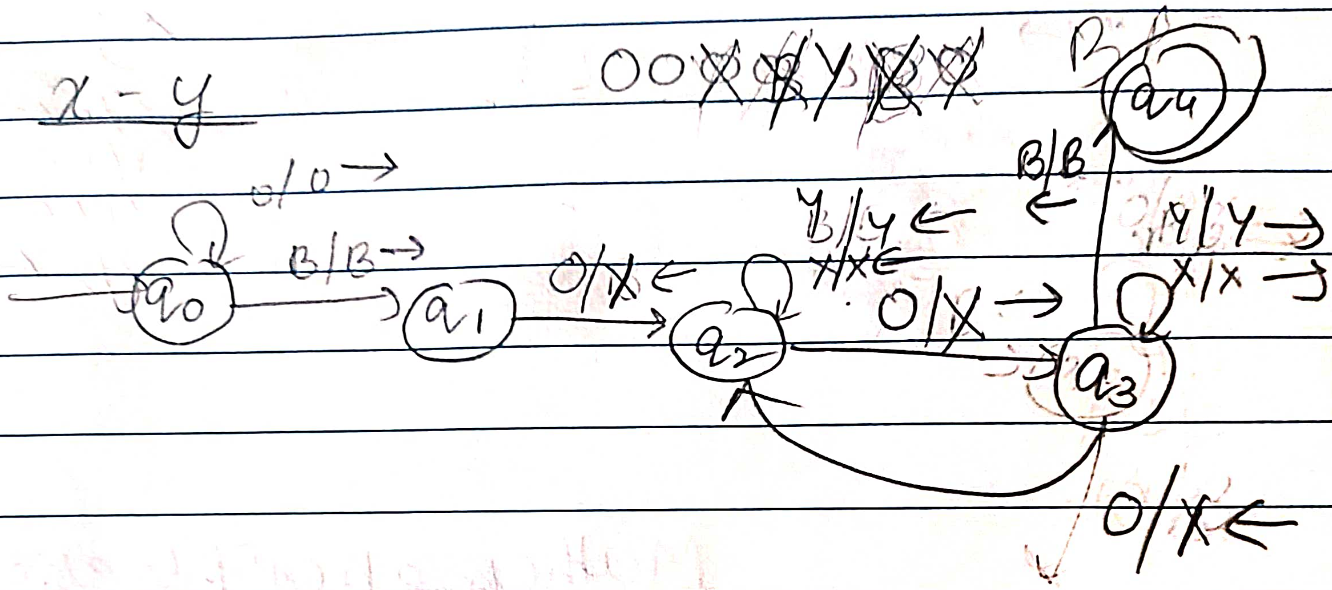
O/P 00

$0\,0\,\not{0}\,B\,B\;B\;\not{0}$

<u>$x + 2$</u>




<u>$x - y$</u>



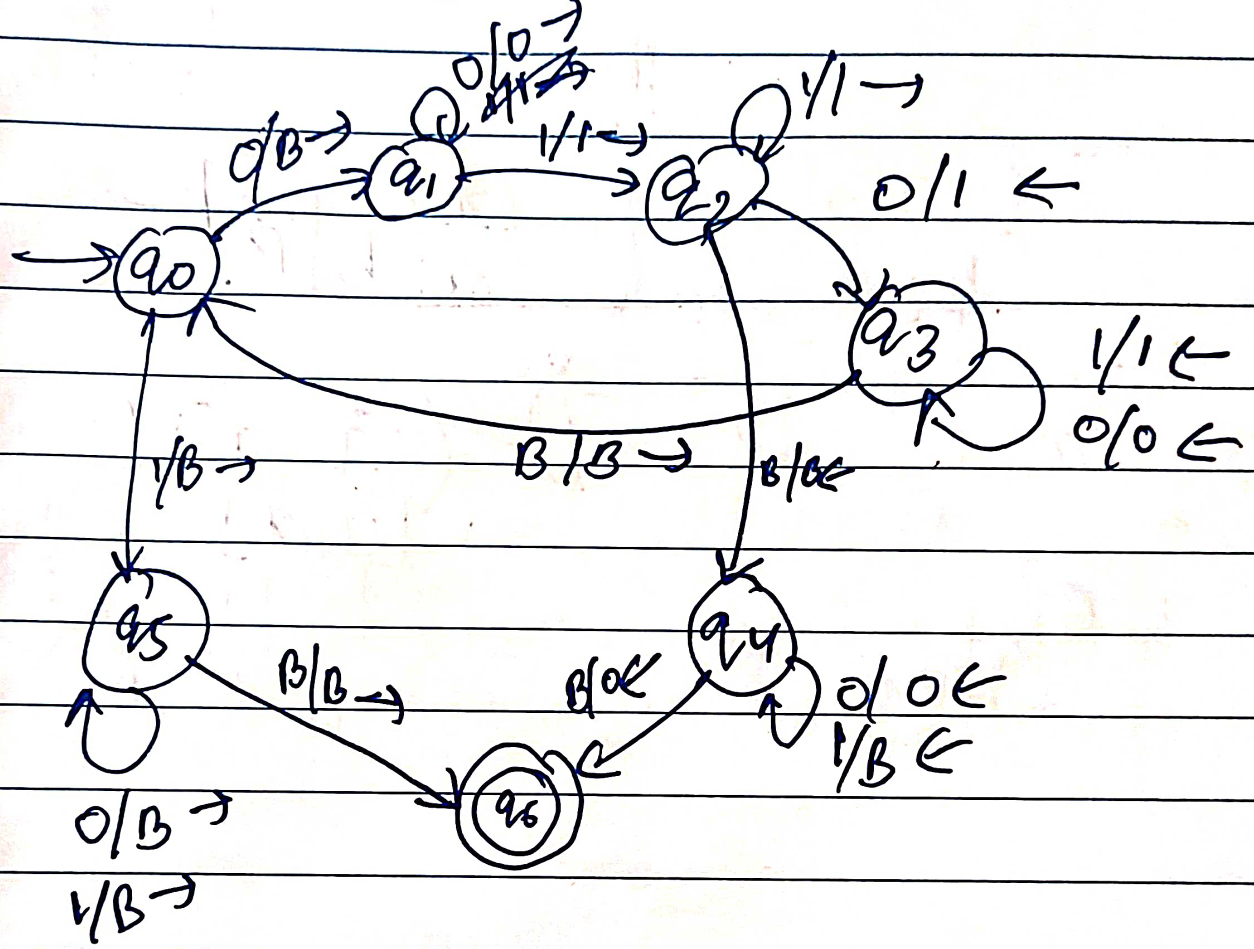

## Monus Subtraction

$$F(x) = \begin{cases} x - y & x > y \\ 0 & \text{Otherwise} \end{cases}$$

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 |
| B | 0 | 0 | 1 | 0 | 0 |
| B | 0 | 0 | 1 | 0 | $\not{0}$1 |
| B | B | 0 | 1 | 0 | $\not{0}$1 |
| B | B | 0 | 1 | 1 | 1 |
| B | B | B | 1 | 1 | 1 |

B B B B B B

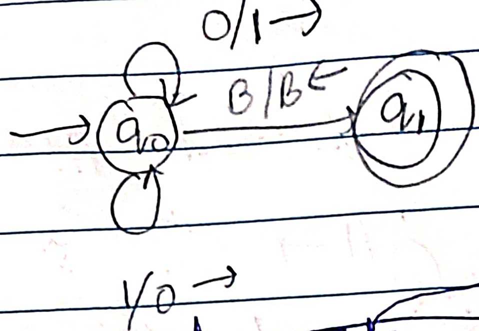
0




Programming techniques in TM to ~~find 1S complement~~ -

1. Storage in finite control
2. Multi-track tape.
3. Subroutines

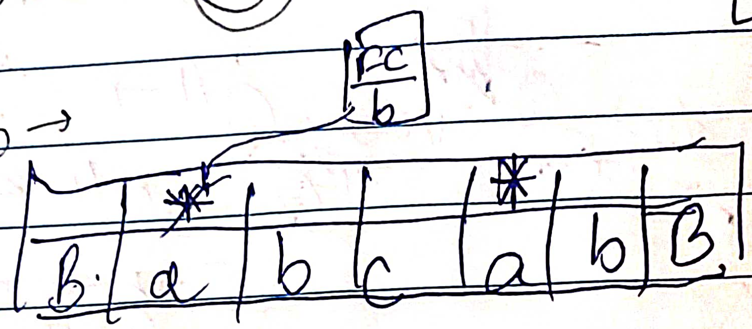
Q/ Find is complement.




L = wcw

$\Sigma = \{a, b\}$

Q/




$\delta([q_1, B], [B, a]) = ([q_1, a] [X, a], R)$

$\delta([q_1, x], [B, y]) = ([q_1, x], [B, y], R)$

$\delta([q_1, x], [B, C]) = ([q_2, x], [B, C] R)$

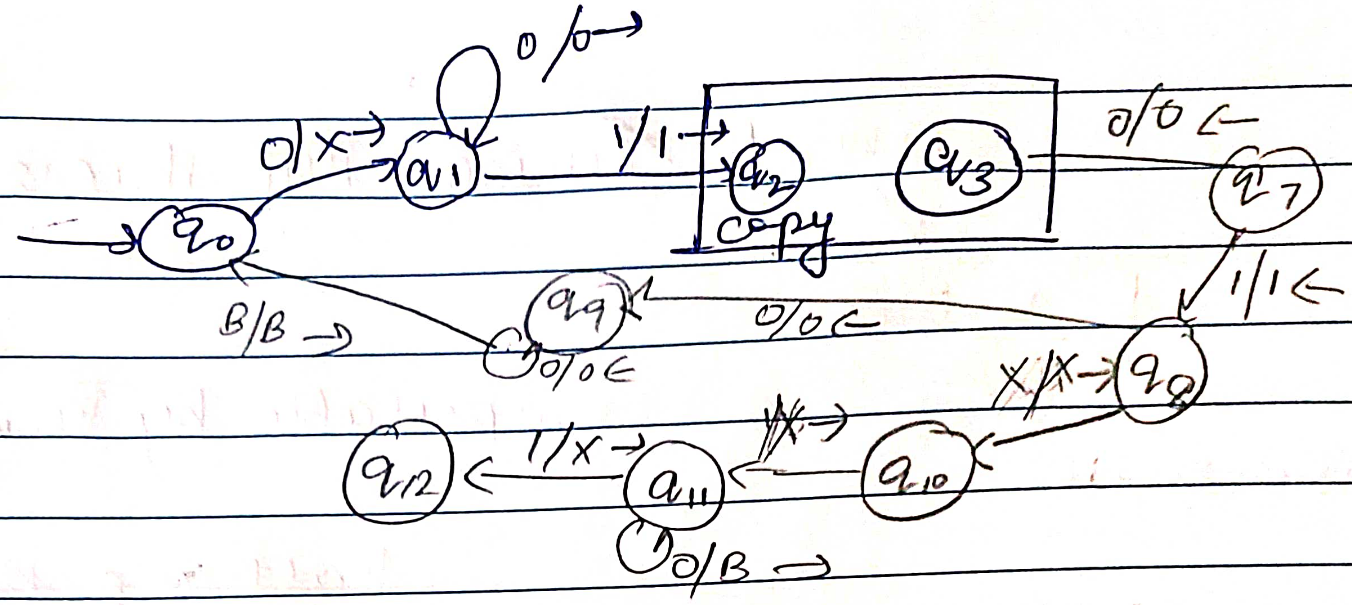
$\delta([q_2, x], [*, y]) = ([q_2, x], [*, y] R)$

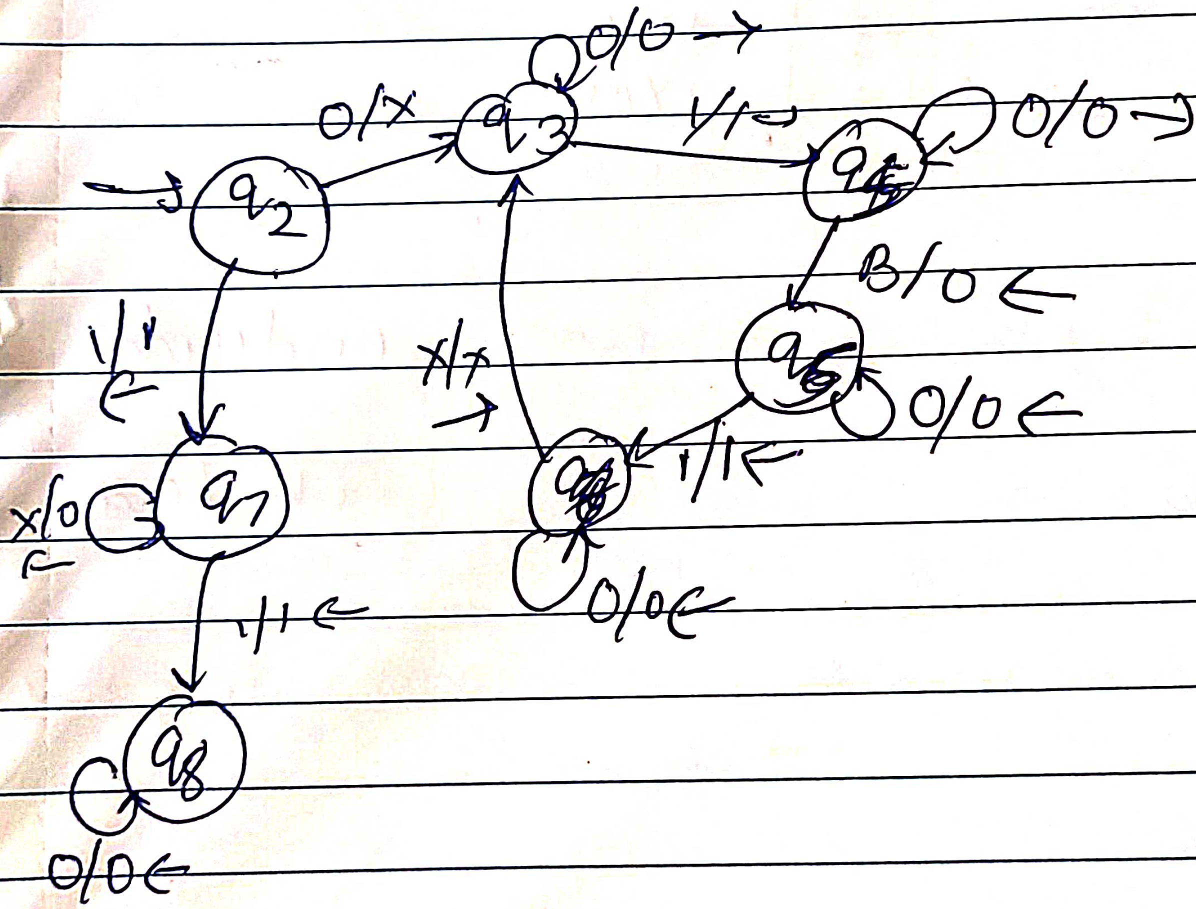
Q/ $0^m 1 0^n 1$ multiply $m \times n$

0001001

X001001 00

XX~~X~~1001000000




Extensions to basic turing machines

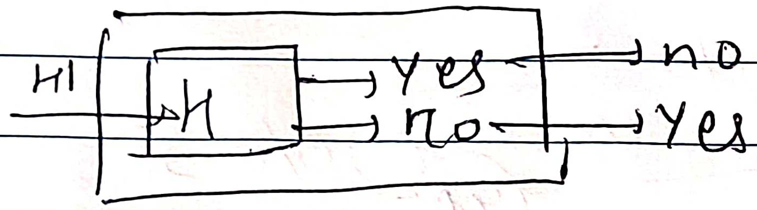
multi tape — non deterministic

## Church Turing Hypothesis

values

every function is computable by Turing machine.

**Halting problem**




Fermat $a^n + b^n = c^n$ $n > 2$ undecidable

$(a+b)(b+c)(c+a)$ Cook Levin

NP complete

will terminate at some point - long time -

intractable

Diagonalisation language

$M = (Q, \{0,1\}, \Gamma, \delta, q_0, B, F)$

$\delta(q_i, x_j) = (q_k, x_l, D_m)$

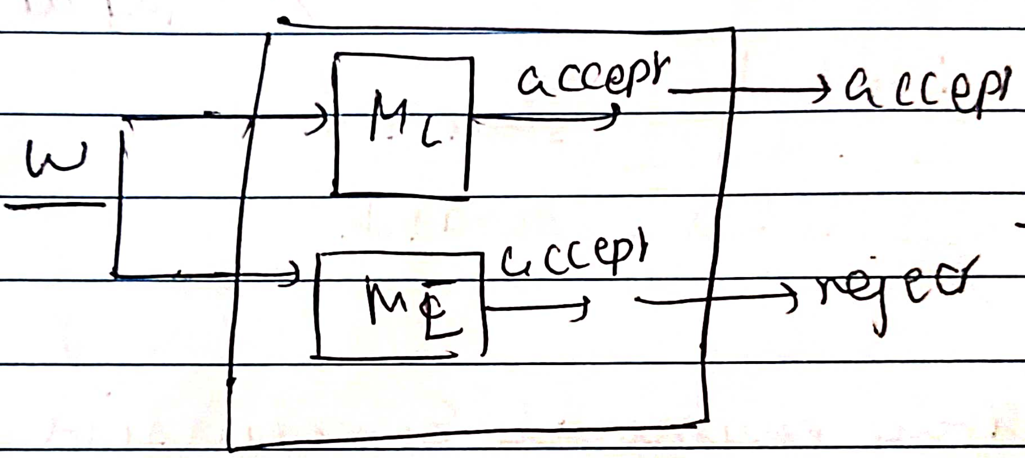
$0^i 1 0^j 1 0^k 1 0^l 1 0^m 11$

$((q_r, B) = q_0, B, R)$

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 |
| 2 | 1 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 | 1 |
| 4 | 1 | 0 | 0 | 1 |

Diagonalisation Language – no TM.

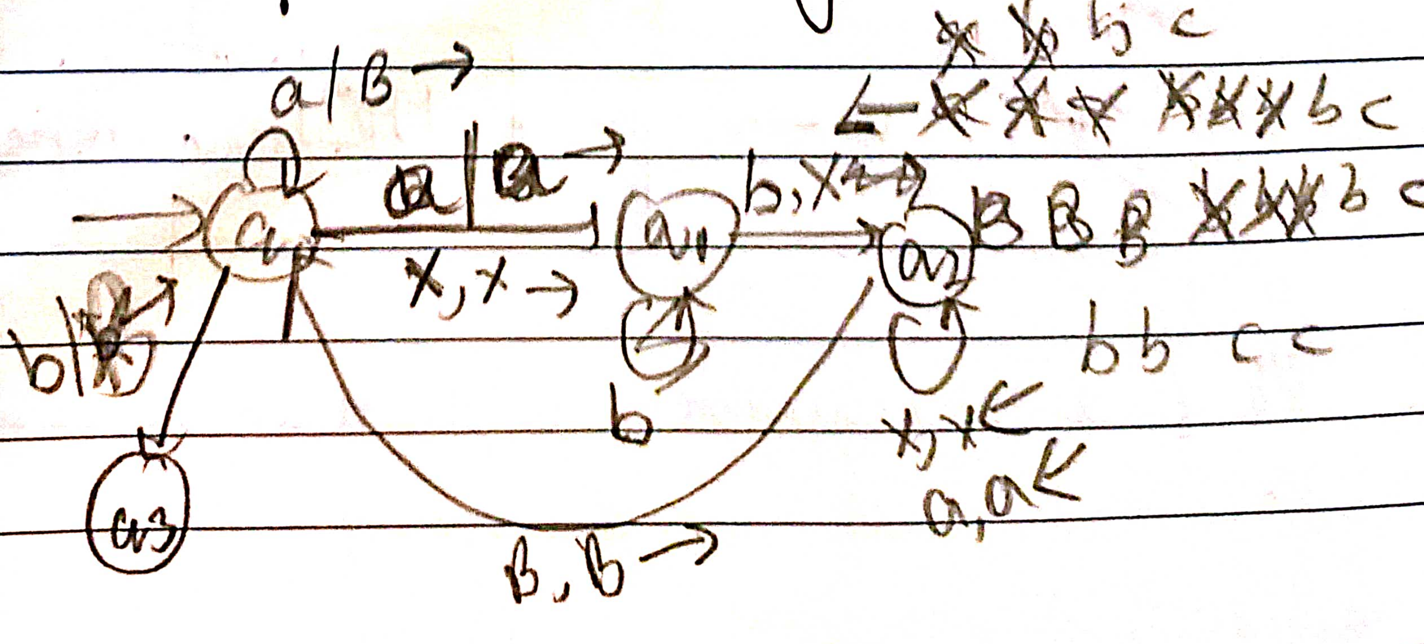
0 1 1 1 ←



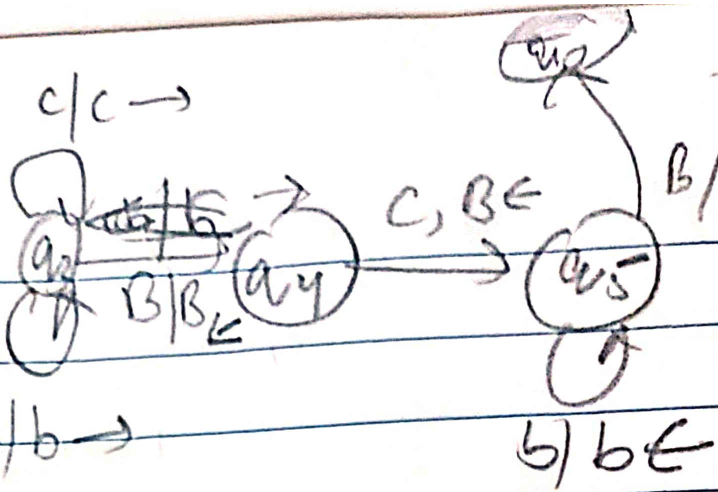

$\underbrace{L \text{ and } \bar{L}}$
RE

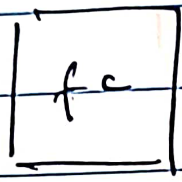
11) $L = \{a^i b^j c^k \mid i + k = j\}$
$i, j, k \geq 0$

Design a Turing Machine



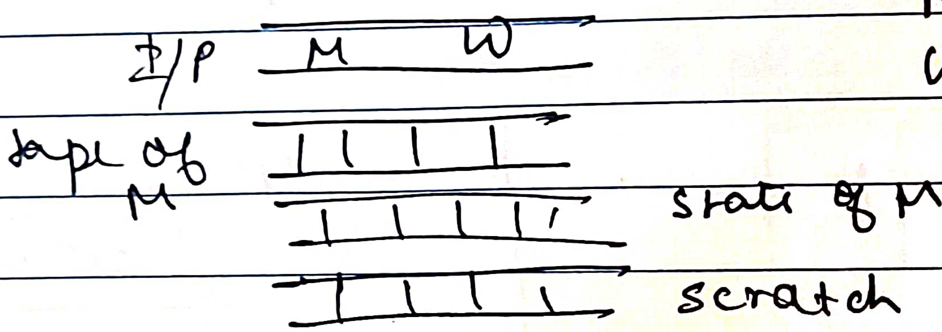
bb c B B




M : turing machine

W : input

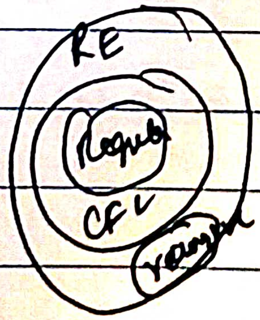


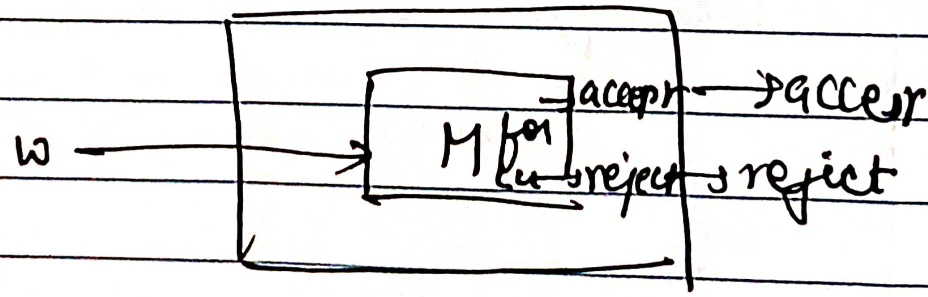

Recursively enumeral & recursive lang

accepted by TM.

may/may not halt

will halt




If $L$ is recursive, $\bar{L}$ is also recursive

W11 W  If same → rejects

| | A | B |
|---|---|---|
| 1 | 110 | 110110 |
| 2 | 0011 | 00 |
| 3 | 0110 | 110 |

2 3 1

A : 0011 0110 110
B : 00 110 110 110 } equal.

3 1

A : 1101011
B : 110101

| | A | B |
|---|---|---|
| 1 | 011 | 101 |
| 2 | 11 | 011 |
| 3 | 1101 | 110 |

Q

| | A | B |
|---|---|---|
| 1 | 1 | 111 |
| 2 | 10111 | ~~111~~ 10 |
| 3 | 10 | 0 |

~~1011110~~ ~~111~~ ~~10110~~

1011111 1011111

0 0

A $\boxed{2113}$

$L_u \rightarrow MPCP \longrightarrow PCP$

| | A | B | |
|---|---|---|---|
| 0 | *1* $y_0$ | *1*1 | $z_0$ |
| 1 | 1* $y_1$ | *1*1 | $z_1$ |
| 2 | 1*0*1*1*1* $y_2$ | *1*0 | . |
| 3 | 1*0* | *0 | . |
| 4 | $ | *$ | |

Let $i_1, i_2 \ldots i_m$ be solution of MPCP

$w_1 w_2 w_3 \ldots w_m : x_1 x_2 \ldots x_m$

$*y_1 y_2 \ldots y_m = z_1 \ldots z_n *$

$y_{k+1} = \$$. $z_{k+1} = *\$$

0 2 1 1 3 4

$Q_3$ $L_u \rightarrow PCP$

| | $\delta(q_1, 0)$ | $\delta(q_1, 1)$ | $\delta(q_1, B)$ |
|---|---|---|---|
| $q_1$ | $(q_2, 1, R)$ | $(q_2, 0, L)$ | $(q_2, 1, L)$ |
| $q_2$ | $(q_3, 0, L)$ | $(q_1, 0, R)$ | $(q_2, 0, R)$ |
| $q_3$ | — | — | — |

input string 01

$\underbrace{q_1 0}_{\text{see table}} 1 \vdash 1 \underbrace{q_2}_{\text{move Right}} 1 \vdash 1 0 q_1$

next symbol

$\vdash 1 q_2 01 \vdash$ ~~$10 q_1 \vdash$~~

$q_3 101 \nvdash$

accepting

PCP → MPCP

~~list A~~ ~~list B~~

1. # ~~# q w #~~

1st pair is decided

$L_u \xrightarrow{I} MPCP \rightarrow PCP$

1. First pair — list A: # — list B: # $q_0$ w #

2. Append — list A: X, # — list B: X, # — for each X in Γ

3. For all non-accepting states

| list A | list B | |
|---|---|---|
| $qX$ | $Yp$ | if $\delta(q,X) = (p,Y,R)$ |
| $Zq X$ | $pZY$ | if $\delta(q,X) = (p,Y,L)$ |
| $q\#$ | $Yp\#$ | if $\delta(q,B) = (p,Y,R)$ |
| $Zq\#$ | $pZY\#$ | if $\delta(q,B) = (p,Y,L)$ |

4. If q is accepting state

| list A | list B | |
|---|---|---|
| $XqY$ | $q$ | for all tape symbols $X$ and $Y$ |
| $Xq$ | $q$ | |
| $qY$ | $q$ | |

5. Find pair

| list A | list B |
|---|---|
| $q\#\#$ | $\#$ |

Example

Q/

| | $\delta(q_i, 0)$ | $\delta(q_i, 1)$ | $\delta(q_i, B)$ |
|---|---|---|---|
| $q_1$ | $(q_2, 1, R)$ | $(q_2, 0, L)$ | $(q_2, 1, L)$ |
| $q_2$ | $(q_3, 0, L)$ | $(q_1, 0, R)$ | $(q_2, 0, R)$ |
| $q_3$ | | | |

I/P = 01

Ans:

| | list A | list B |
|---|---|---|
| 1) | $\#$ | $\#q_1 01\#$ |
| 2) | $0$ | $0$ |
| | $1$ | $1$ |
| | $\#$ | $\#$ |
| 3) | $q_1 0$ | $1 q_2$ |
| | $0 q_1 1$ | $q_2 0 0$ |
| | $1 q_1 1$ | $q_2 1 0$ |

$0q_1\#$ $\quad$ $q_2 01\#$

$1q_1\#$ $\quad$ $q_2 11\#$

4) $0q_3 1$ $\quad$ $q_3$

$1q_3 0$ $\quad$ $q_3$

$1q_3$ $\quad$ $q_3$

$0q_3$ $\quad$ $q_3$

$q_3 0$ $\quad$ $q_3$

$q_3 1$ $\quad$ $q_3$

$1q_3 1$ $\quad$ $q_3$

$0q_3 0$ $\quad$ $q_3$

5) $q_3\#\#$ $\quad$ $\#$

Final pair

list A $\quad$ list B

$q_1\#\#$ $\quad$ $\#$

$\#q_1 01\#$

$\#q_1 01\#1q_2 1\#10q_1\#1q_2 0\#q_3 10\#$

$q_3 01\# q_3 1 \# q_3 \#\#$

$q_1 01 \vdash 1q_2 1 \vdash 10q_1 \vdash q_2 01 \vdash$

$q_3 101$

CFG /

An automata and a given string - decidable.

Empty string - decidable

Ambiguity in grammar - undecidable.

Given two grammars - do they accept same language → not decidable

Is intersection of two CFL empty or not →

Calculating ECLOSE → $O(n^3)$

Subset construction → $O(n^3 s)$ s: states

DFA → NFA → $O(n)$

DFA/NFA → RE → $O(n^3 4^{n^3 2^n})$

RE → automata → $O(n)$

PDA → CFG → $O(n^3)$

Grammar → CNF → $O(n^2)$