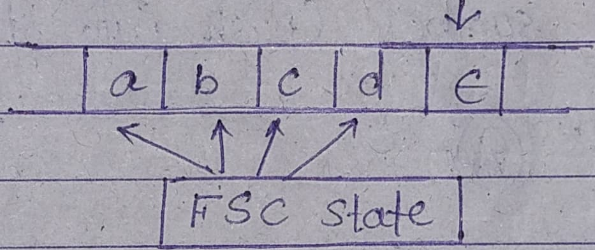
Unit - 4. Pushdown Automata

Automata

Finite State automata | Pushdown automata

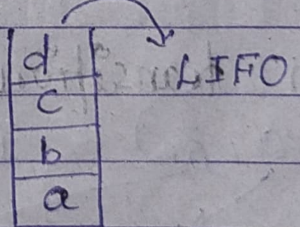
Pushdown Automata

input type



→ The basic structure of FA take but add extra data structure — Pushdown Automata.

data structure use "stack".

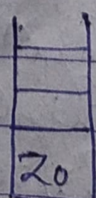


→ Push, Pop, Operation, no change

condition overflow, underflow

↓ we assume that stack is infinite

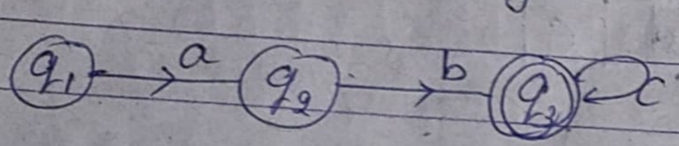
↓ we add Z_0 so that underflow cond^n not occur



if stack is empty the also we have Z_0 to pop.

pushdown has 7 tuple Automata

Q = no. of state
q_0 = initial state
F = final state
Σ = input variable
δ = Transition $Q \times (\Sigma \cup \epsilon) \times \Gamma \rightarrow Q \times \Gamma^*$
Z_0 = bottom or initial stack symbol
(tau) Γ = stack symbol (push into stack)



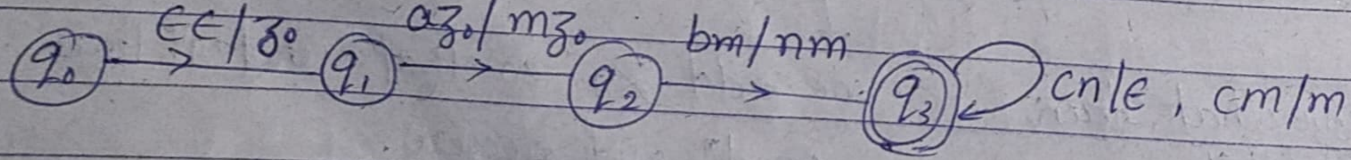
$Q = \{q_1, q_2, q_3\}$ $\quad$ $\Sigma = \{a, b, c\}$
$q_0 = q_1$ $\quad$ $Z_0 = z_0$
$F = q_3$ $\quad$ $\Gamma = \{$

pushdown transition

$\delta(q_1, a, z_0) = (q_2', mz_0)$ → output state

input state, input symbol, top of stack

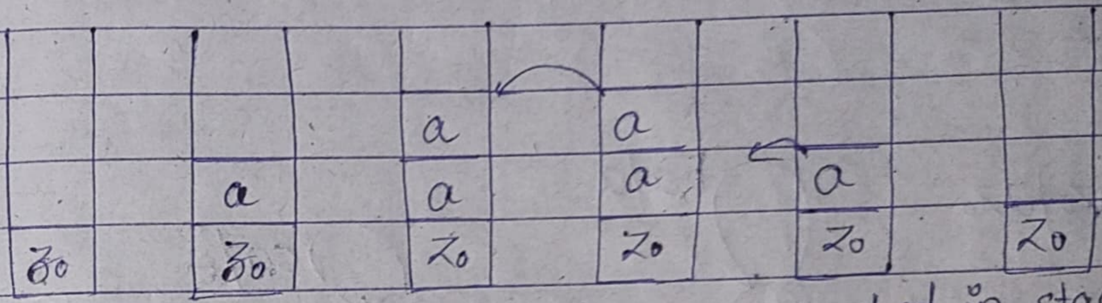
→ m is push over z_0

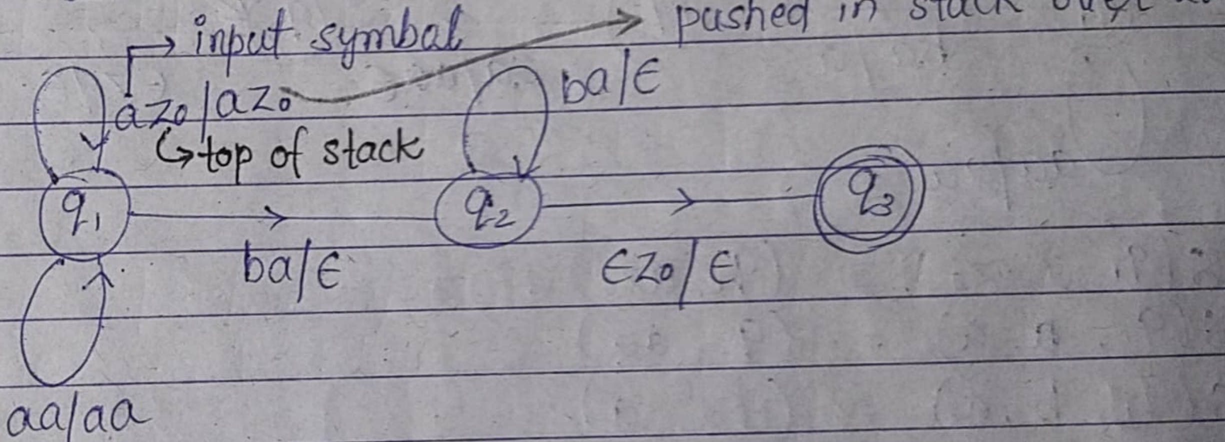


$\delta(q_2, b, m) = (q_3, nm)$ ← push
$\delta(q_3, c, m) = (q_3, \epsilon)$ → top of stack is popped
$\delta(q_3, c, m) = (q_3, m)$ ⟷ no change in stack / skip

Que $a^n b^n$

aabb





$\delta(q_1, a, Z_0) = (q_1, aZ_0)$

$\delta(q_1, a, a) = (q_1, aa)$

$\delta(q_1, b, a) = (q_2, \epsilon)$

$\delta(q_2, b, a) = (q_2, \epsilon)$

$\delta(q_2, \epsilon, Z_0) = (q_3, \epsilon)$

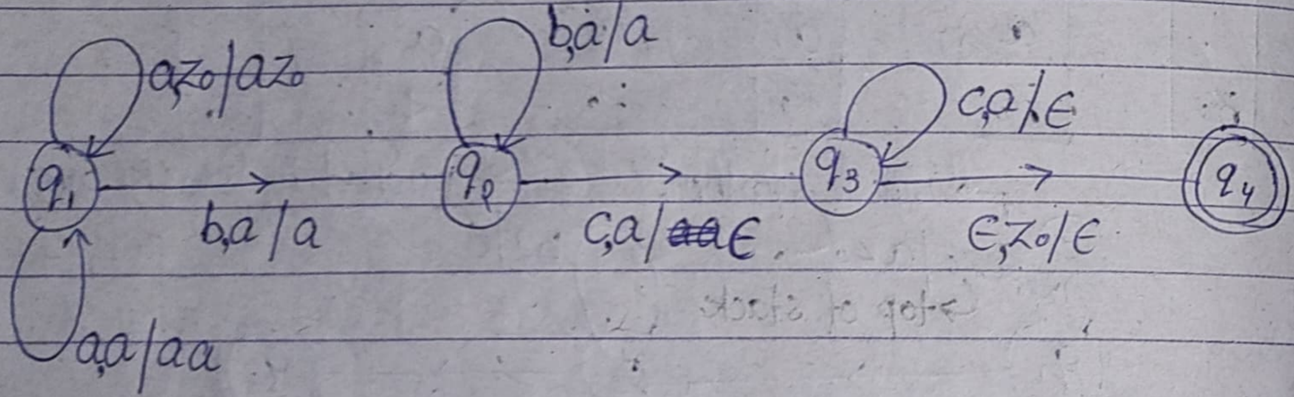
Que $a^n b^m c^n$ $(m, n \geq 0)$

Sol^n $a^n b^m c^n$

$n = 2$

$m = 3$

$a^2 b^3 c^2$

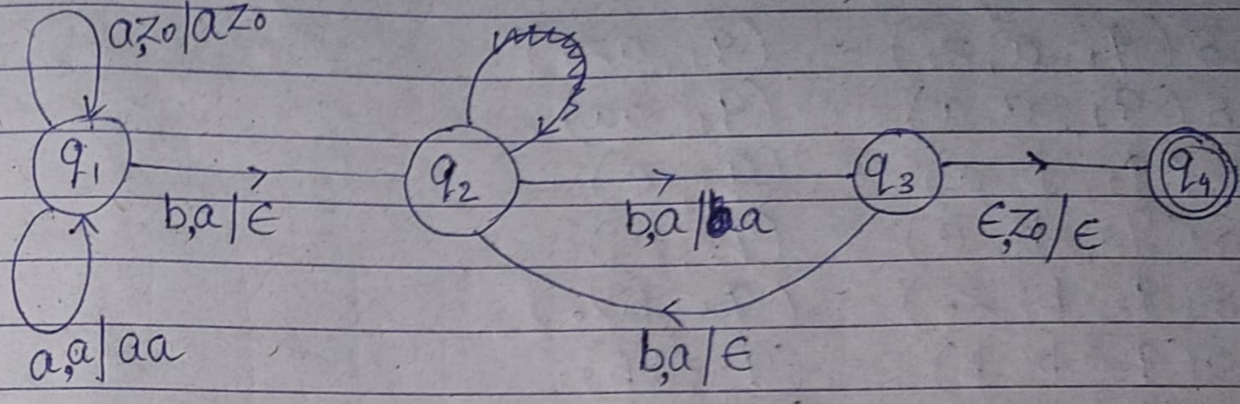


$\delta(q_1, a, z_0) = (q_1, az_0)$
$\delta(q_1, a, a) = (q_1, aa)$
$\delta(q_1, b, a) = (q_2, a)$
$\delta(q_2, b, a) = (q_2, a)$
$\delta(q_2, c, a) = (q_3, \epsilon)$
$\delta(q_3, c, a) = (q_3, \epsilon)$
$\delta(q_3, \epsilon, z_0) = (q_4, \epsilon)$

Que $a^n b^{2n}$

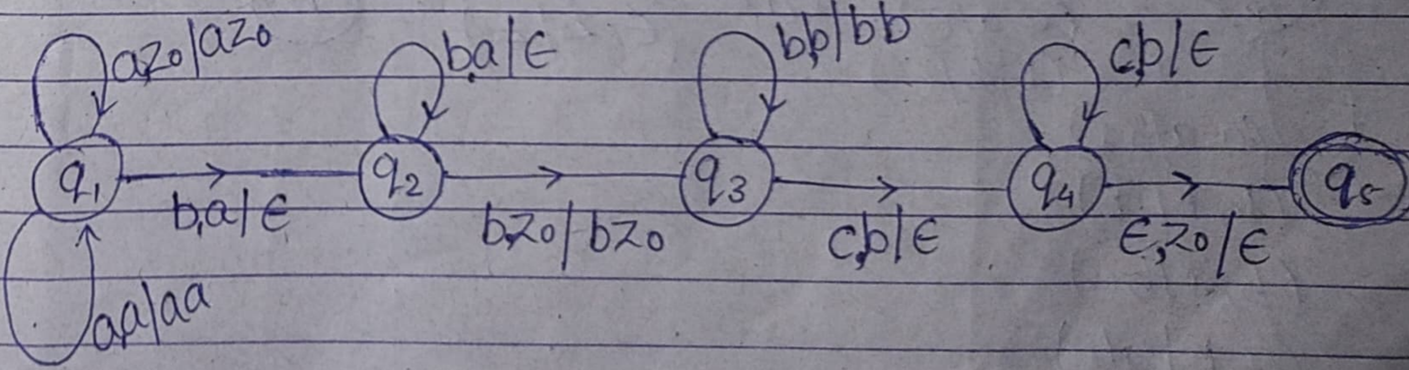
Sol^n $a = 3,\ b = 6$

$a^2 b^4 = aaabbbbbb$



$\delta(q_1, a, z_0) = (q_1, az_0)$
$\delta(q_1, a, a) = (q_1, aa)$
$\delta(q_1, b, a) = (q_2, \epsilon)$
$\delta(q_2, b, a) = (q_3, a)$
$\delta(q_3, b, a) = (q_2, \epsilon)$
$\delta(q_3, \epsilon, z_0) = (q_4, \epsilon)$

<u>Que</u> $a^n b^{m+n} c^m$
<u>Sol</u> $a^n b^{m+n} c^m$
$n = 2, \; m = 3$
aabbbbbccc



$\delta(q_1, a, z_0) = (q_1, az_0)$
$\delta(q_1, a, a) = (q_1, aa)$
$\delta(q_1, b, a) = (q_2, \epsilon)$
$\delta(q_2, b, a) = (q_2, \epsilon)$
$\delta(q_2, b, z_0) = (q_3, bz_0)$
$\delta(q_3, b, b) = (q_3, bb)$
$\delta(q_3, c, b) = (q_4, \epsilon)$
$\delta(q_4, c, b) = (q_4, \epsilon)$
$\delta(q_4, \epsilon, z_0) = (q_5, \epsilon)$

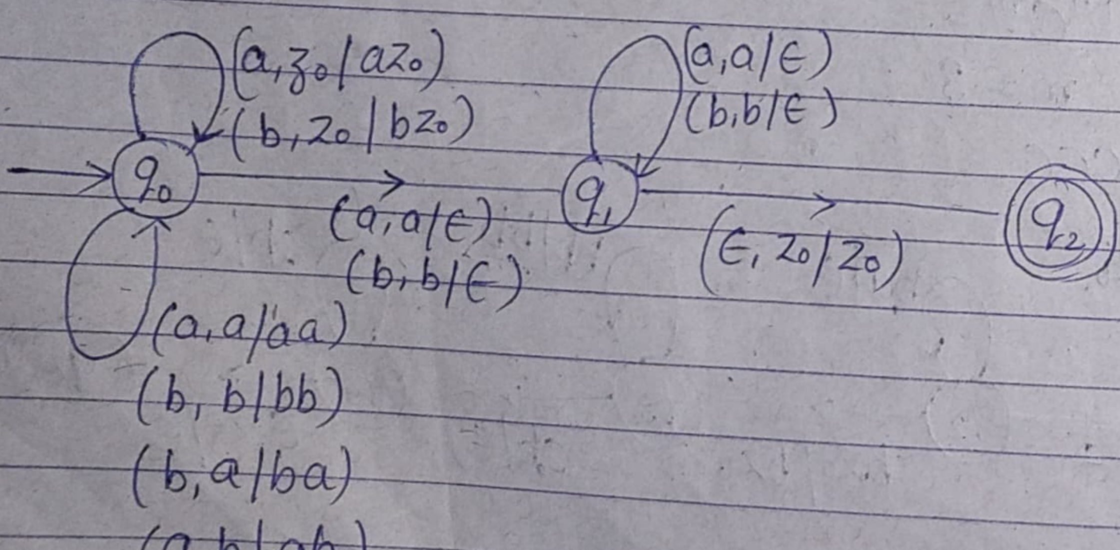
Que (ww^r), $w \in (0,1)^*$

Sol^n^ ww^r

$w = 0011$, $w^r = 1100$

$ww^r = 00111100$

$L = \{\epsilon, 00, 11, 0110, 1001 \ldots\}$



Que $a^n b^n c^n$

Sol^n

aaabbbccc

$b \leftarrow$

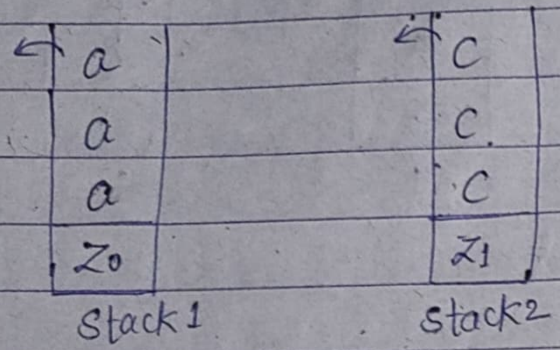
S1
a
a
a
Z_0

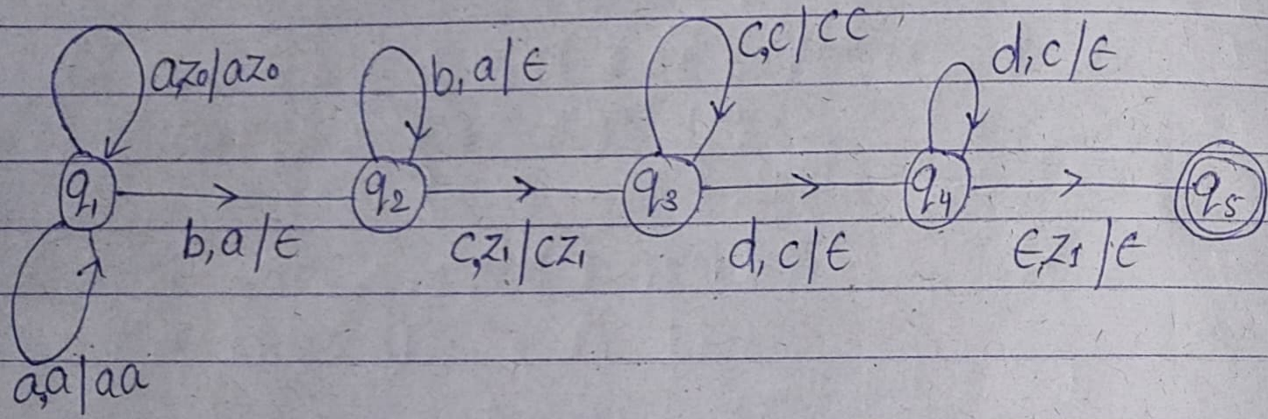
$c \leftarrow$

S2
b
b
b
Z_1

Que $a^n b^n c^n d^n$

Sol^n. aaabbbcccddd





Pushdown Automata and Context Free Language.

→ If L is a CFL then we can construct a PDA A accepting L by empty store.

→ Let $L = L(G)$ where

$$G = (V_N, \Sigma, P, S)$$

is a CFG we construct a PDA A as

$$A = (\{q\}, \Sigma, \underline{V_N \cup \Sigma}, \delta, q, S, \phi)$$

Γ

where δ is defined by the following rules –

use in place of ϵ

$$\delta(q, \Lambda, A) = \{(q, \alpha) \mid A \to \alpha \text{ is in } P\}$$

$$\delta(q, a, a) = \{(q, \epsilon)\} \text{ for every } a \text{ in } \Sigma\}$$

example $A \longrightarrow 1B \mid a$

$$\delta(q, \epsilon, A) = (q, 1B)$$
$$\delta(q, a, a) = (q, \epsilon)$$

Que $S \longrightarrow aSb$

$S \longrightarrow a \mid b \mid \epsilon$

Solⁿ Transitions

$$\delta(q, \epsilon, S) = (q, aSb)$$
$$\delta(q, \epsilon, S) = (q, a)$$
$$\delta(q, \epsilon, S) = (q, b)$$
$$\delta(q, \epsilon, S) = (q, \epsilon).$$

$\delta(q, a, a) = (q, \epsilon)$
$\delta(q, b, b) = (q, \epsilon)$
$\delta(q, \epsilon, Z_0) = (q, \epsilon)$

Is string aaabbb accept?

$(q, \cancel{\epsilon}\, aaabbb, S) = (q, aSb)$
$\longrightarrow (q, \epsilon, S) = (q, aSb)$

a
ϵ
b
b
b
Z_0

$(q, aaabbb, S) = (q, aSb)$
$(q, aabbb, S) = (q, \underbrace{Sb}_{\text{put } S = aSb})$

$(q, \cancel{a}abbb, S) = (q, \cancel{a}Sbb)$
$(q, abbb, S) = (q, \underbrace{Sbb}_{\text{again put } S = aSb})$

$(q, \cancel{a}bbb, S) = (q, \cancel{a}Sbbb)$
$(q, bbb, S) = (q, \underbrace{Sbbb}_{\text{put } S = \epsilon})$

$(q, bbb, S) = (q, bbb)$

S
Z_0

a
S
b
Z_0

a
S
b
b
Z_0

a
S
b
b
b
Z_0

$\cancel{a}$
ϵ
b
b
b
Z_0

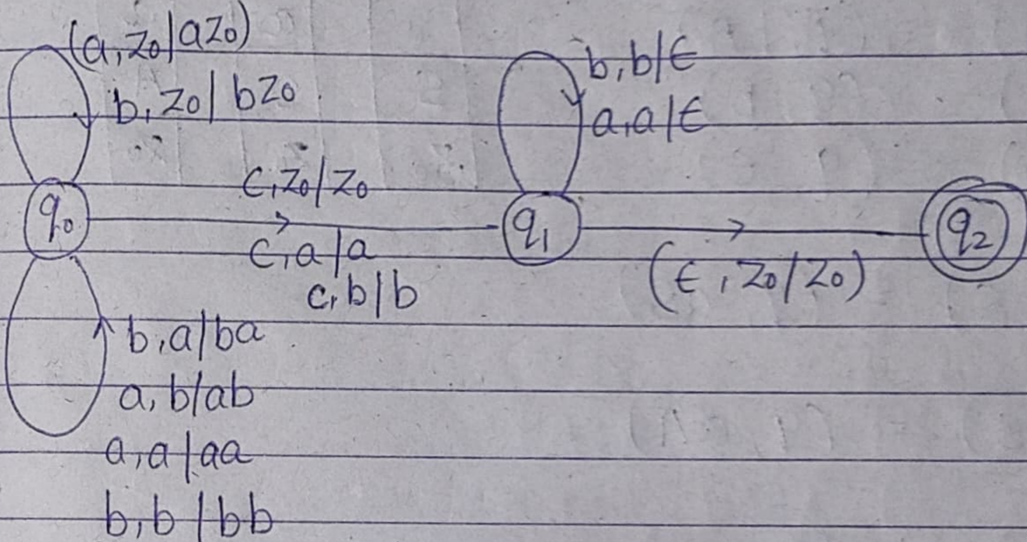
Que Construct pushdown Automata for wcw^r

Sol^n wcw^r

$L = \{c, aca, bcb, abcba, \text{------} \}$

$wcw^r = aabacabaa$

$\hookrightarrow w = aaba$, $w^r = abaa$



Que ① $S \rightarrow aS \mid aA$

$A \rightarrow bA \mid b$

String = aabb

Solⁿ Transition -

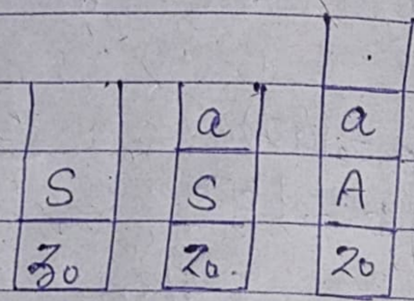
$\delta(q, \epsilon, S) = (q, aS)$

$\delta(q, \epsilon, S) = (q, aA)$

$\delta(q, \epsilon, A) = (q, bA)$

$\delta(q, \epsilon, A) = (q, b)$

$\delta(q, b, b) = (q, \epsilon)$



$(q, \epsilon aabb, S) = (q, aS)$

$(q, abb, S) = (q, aA)$

$(q, bb, S) = (q, bA)$

$(q, b, S) = (q, b)$

String is accepted.

Que ② $S \rightarrow aSb$

$S \rightarrow ab$

String - aaabbb

Date			
Page No.			

$S \rightarrow aSb$ $\quad\quad S \rightarrow aSZ$

$S \rightarrow ab$ $\quad \Longrightarrow \quad S \rightarrow aZ$

$Z \rightarrow b$ $\quad\quad Z \rightarrow b$

$\delta(q, \epsilon, S) = (q, aSZ)$

$\delta(q, \epsilon, S) = (q, aZ)$

$\delta(q, \epsilon, S) = (q, b)$

$\delta(q, b, b) = (q, \epsilon)$

$(q, \epsilon, S) = (q, aabb, S)$

$= (q, \cancel{a}abb, \cancel{a}SZ)$

$= (q, abb, SZ)$

$= (q, \cancel{a}bb, \cancel{a}ZZ)$

$= (q, bb, ZZ)$

$= (q, \cancel{b}b, \cancel{b}Z)$

$= (q, b, Z)$

$= (q, \cancel{b}, \cancel{b})$

$= (q, \epsilon, \epsilon)$

<u>Que</u> 3 $\quad S \rightarrow aSa$

$S \rightarrow bSb$

$S \rightarrow c$

string = abbcbba

Date | Page No.

Sol^n $S \to aSa$ String = abbcbba

we can consider

$A \to a$

$S \to aSA$

productions –

$S \to aSA$

$S \to bSB$

$S \to c$

$A \to a$

$B \to b$

Transitions –

$\delta(q, \epsilon, S) = (q, aSA)$

$\delta(q, \epsilon, S) = (q, bSB)$

$\delta(q, \epsilon, S) = (q, c)$

$\delta(q, \epsilon, A) = (q, a)$

$\delta(q, \epsilon, B) = (q, b)$

$\delta(q, c, c) = (q, \epsilon)$

$\delta(q, a, a) = (q, \epsilon)$

$\delta(q, b, b) = (q, \epsilon)$

$(q, \epsilon, S) = (q, abbcbba, S)$

$= (q, \cancel{a}bbcbba, \cancel{a}SA)$

$= (q, bbcbba, SA)$

$= (q, \cancel{b}bcbba, \cancel{b}SBA)$

$= (q, bcbba, SBA)$

$= (q, \cancel{b}cbba, \cancel{b}SBBA)$

$= (q, cbba, SBBA)$

$= (q, \cancel{c}bba, \cancel{c}BBA)$

$= (q, bba, BBA)$

$= (q, \cancel{b}ba, \cancel{b}BA)$

$= (q, ba, BA)$

$= (q, \cancel{b}a, \cancel{b}A)$

$= (q, a, A)$

$= (q, \cancel{a}, \cancel{a})$

$= (q, \epsilon, \epsilon)$

<u>Que</u> (4) $S \rightarrow aSbb / a$

$S \rightarrow aSA / a$

$A \rightarrow bB$

$B \rightarrow b$

String = aaabbbb

<u>Sol^n</u>

Que $S \longrightarrow 0B/1A$
$A \longrightarrow 0/0S/1AA$
$B \longrightarrow 1/1S/0BB$

String $= 00110$

Sol^n Transition –

$\delta(q, \epsilon, S) = (q, 0B)$
$\delta(q, \epsilon, S) = (q, 1A)$
$\delta(q, \epsilon, A) = (q, 0)$
$\delta(q, \epsilon, A) = (q, 0S)$
$\delta(q, \epsilon, A) = (q, 1AA)$
$\delta(q, \epsilon, B) = (q, 1)$
$\delta(q, \epsilon, B) = (q, 1S)$
$\delta(q, \epsilon, B) = (q, 0BB)$

String 00110

$\delta(q, \epsilon, S) = (q, 00110, S)$
$= (q, \cancel{0}0110, \cancel{0}B)$
$= (q, 0110, B)$
$= (q, 0110,$

Turing Machine

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

where

Q = Set of states

Σ = set of input symbol

δ = Transition function

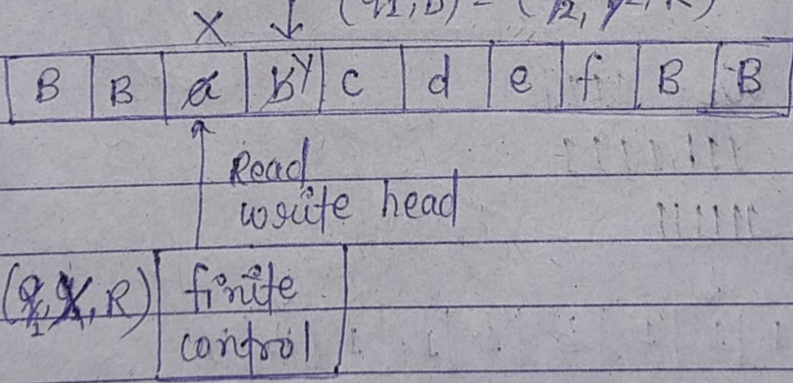
q_0 = Start state

F = Final state

Γ = Set of all symbols read or written on input type

B = Blank Symbol

$(q_1, b) = (q_2, Y, R)$

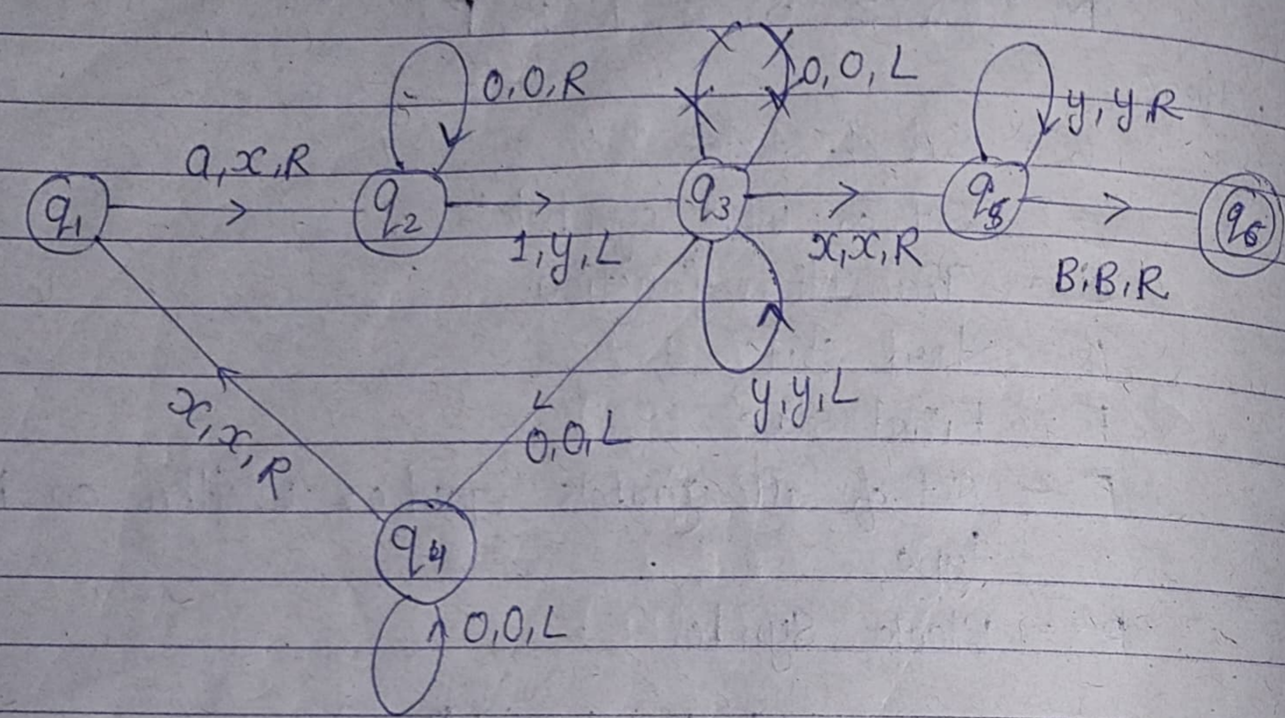


$(q, a) = (q_1, X, R)$

Transition function –

$$\boxed{\delta(Q, \Sigma) = (Q, \Gamma, (L, R))}$$

Que 0011



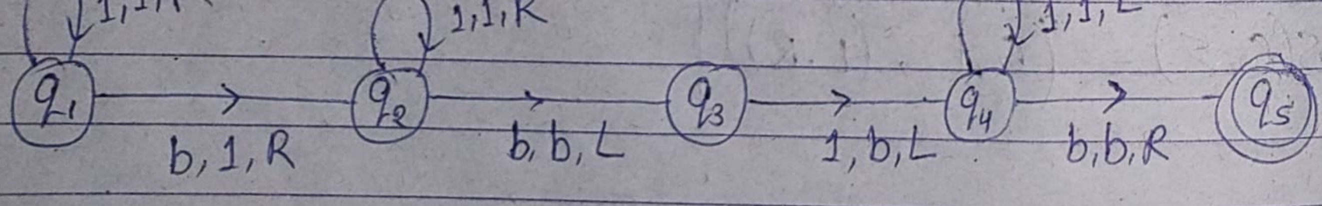
Que $w_1 = 11$, $w_2 = 1111$

$w_1 b w_2 = 11b1111$

find $w_1 w_2 = 111111$

Ans

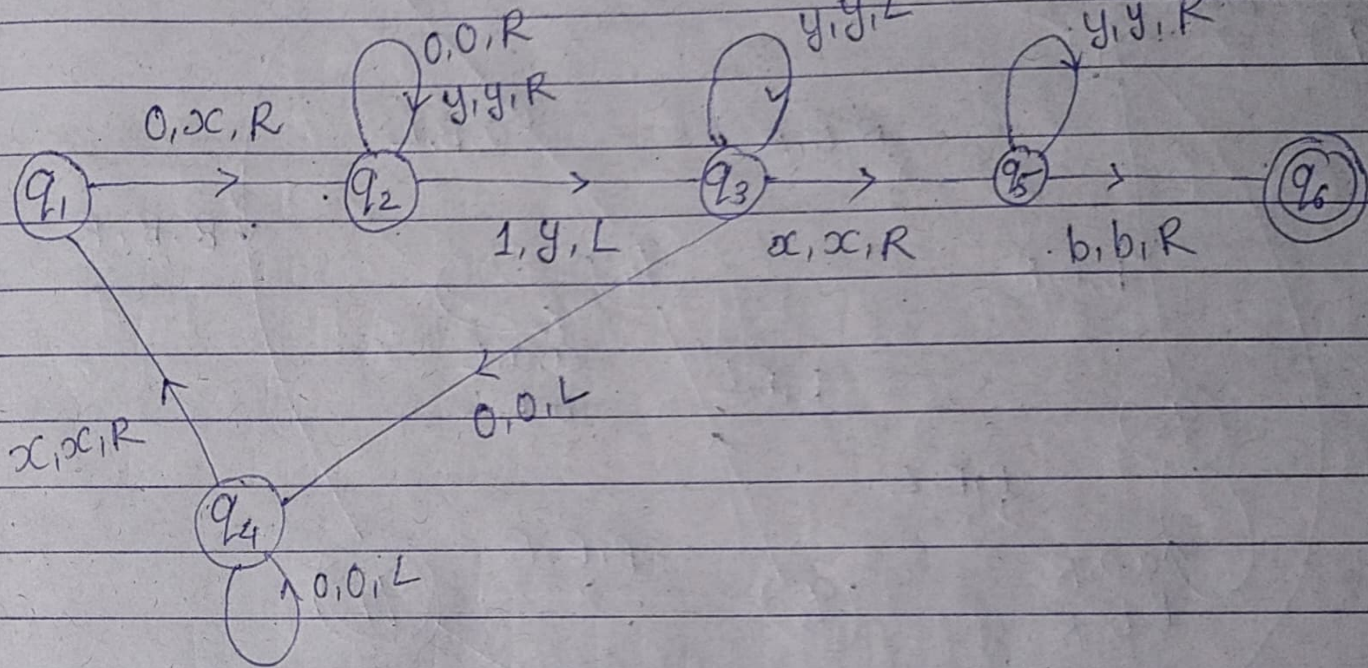
b	1	1	b	1	1	1	b	b



Que $L = 0^n 1^n$ $n=4$

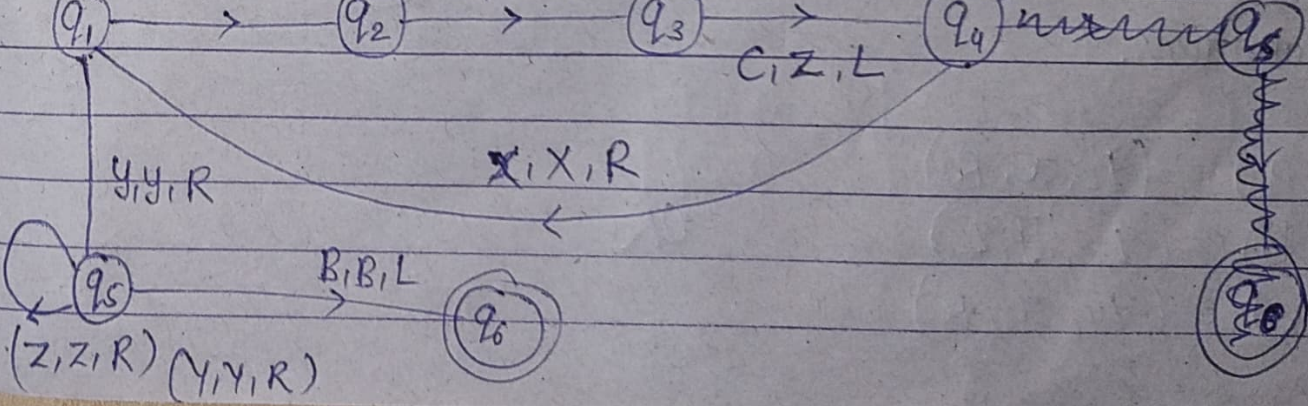
Sol^n

$L = 00001111$



Que $a^n b^n c^n$

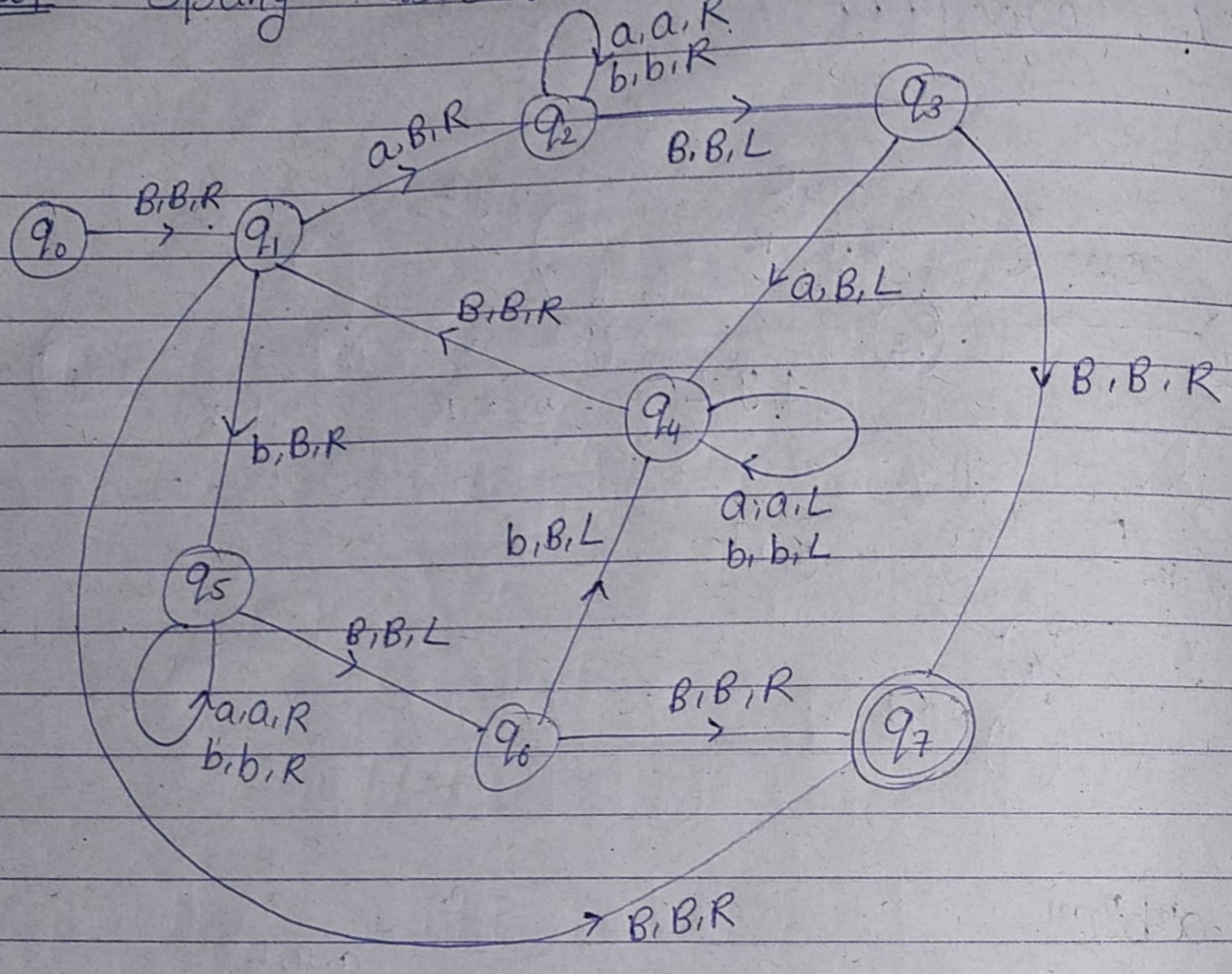
Sol^n



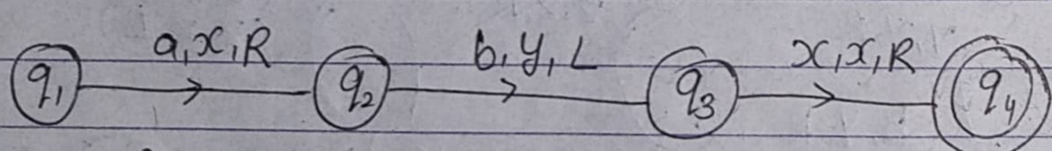
Date
Page No.

Que Turing machine of palindrome

Sol^n String abbbba



Que



write transitions.

$\delta(q_1, a) = (q_2, x, R)$

$\delta(q_2, b) = (q_3, y, L)$ OR

$\delta(q_3, x) = (q_4, x, R)$

	a	b	x	y
q_1	$x R q_2$			
q_2		$y L q_3$		
q_3			x, R, q_4	
q_4				

Conversion of PDA to CFG

$G = (V, \Sigma, P, S)$

(I) Variables

1. Special Symbol S
2. $[pxq] \rightarrow p, q \rightarrow$ no. of states in Q and x is in Γ

   e.g. p, q - states & $x, y - \Gamma$

   $pxq,\ pxp,\ qxp,\ qxq$

   $pyq,\ pyp,\ qyp,\ qyq$

(II) Productions

1. For all states P

   $S \rightarrow [q_0 Z_0 P]$

2. Let $\delta(q, a, x) = (r, y_1 y_2 \cdots y_k)$

$$[q, x, r_k] = a[r, y_1, r_1][r_1, y_2 r_2] \cdots [r_{k-1}, y_k r_k]$$

current state, top of stack, final state

Date
Page No.

states

3) if $\delta(q, x, A) = (p, \epsilon)$

$[q, A, p] \rightarrow x$

) $\delta(q, \epsilon, x) = (r, \epsilon)$

$[q, x, r) \rightarrow \epsilon$

$P = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$

Q → $\{q_0, q_1\}$, Σ → $\{a, b\}$, Γ → $\{x, y\}$, q_0 → initial state, Z_0 → top of stack, δ → transition, F → Final state

$G = (V, T, P, S)$

V → variable, T → terminal (a, b), S → start

Step (I)

variable = Q and Γ combination

qxp , qyp , S
qxq , qyq
pxq , pyq
pxp , pyp

$S \rightarrow [q_0 Z_0 P] \rightarrow [q_0, a, p]$
$\rightarrow [q_0, a, q]$

Transition Step II

$\rightarrow \quad \delta(q_1, a, b) = (q_2, b)$

$[q_1, b, q_1] \rightarrow a[q_2, b, q_1]$

$[q_1, b, q_2] \rightarrow a[q_2, b, q_2]$

↳ for all states

$\rightarrow \quad \delta(q_1, a, b) = (q_2, ab)$

same

$[q_1, b, q_1] \rightarrow a[q_2, a, q_1][q_1, b, q_1]$

$[q_1, b, q_1] \rightarrow a[q_2, a, q_2][q_2, b, q_1]$

$[q_1, b, q_2] \rightarrow a[q_2, a, q_1][q_1, b, q_2]$

$[q_1, b, q_2] \rightarrow a[q_2, a, q_2][q_2, b, q_2]$

same

Step III

$\delta(q_1, \epsilon, z) = (q_1, \epsilon)$

$[q_1, z, q_1] \rightarrow \epsilon$

$\delta(q_1, b, z) = (q_1, \epsilon)$

$[q_1, z, q_1] \longrightarrow b$

Que $P = (\{q, p\}, \{0, 1\}, \{x, z\}, \delta, z, q\}$

$\delta(q, 1, z) = (q, xz)$
$\delta(q, 1, x) = (q, xx)$
$\delta(q, \epsilon, x) = (q, \epsilon)$
$\delta(q, 0, x) = (p, x)$
$\delta(P, 1, x) = (P, \epsilon)$
$\delta(P, 0, z) = (q, z)$

Sol^n

Given $Q = \{P, q\}$, $\Gamma = \{x, z\}$, $z_0 = z$
$\Sigma = \{0, 1\}$, $q_0 = q$

Step 1 variable
① Special symbol S
② qxp, qxq, pxq, pxp, qzp, qzq, pzq, pzp

Step 2 productions
for all state P
$S \longrightarrow [q_0 z_0 P]$

(1) $S \longrightarrow [q, z, p]$
(2) $S \longrightarrow [q, z, q]$

(I) $\delta(q, 1, z) = (q, xz)$

(3) $[q, z, p] \rightarrow 1\,[q, x, p]\,[p, z, p]$
(4) $[q, z, p] \rightarrow 1\,[q, x, q]\,[q, z, p]$
(5) $[q, z, q] \rightarrow 1\,[q, x, p]\,[p, z, q]$
(6) $[q, z, q] \rightarrow 1\,[q, x, q]\,[q, z, q]$

(II) $\delta(q, 1, x) = (q, xx)$

(7) $[q, x, p] \rightarrow 1\,[q, x, p]\,[p, x, p]$
(8) $[q, x, p] \rightarrow 1\,[q, x, q]\,[q, x, p]$
(9) $[q, x, q] \rightarrow 1\,[q, x, p]\,[p, x, q]$
(10) $[q, x, q] \rightarrow 1\,[q, x, q]\,[q, x, q]$

(III) $\delta(q, \epsilon, x) = (q, \epsilon)$

(11) $[q, x, q] \rightarrow \epsilon$

(IV) $\delta(q, 0, x) = (p, x)$

(12) $[q, x, p] \rightarrow 0\,[p, x, p]$
(13) $[q, x, q] \rightarrow 0\,[p, x, q]$

(V) $\delta(p, 1, x) = (p, \epsilon)$

(14) $[p, x, p] \rightarrow 1$

(VI) $\delta(P, 0, Z) = (q, Z)$

(15) $[P, Z, P] \rightarrow 0\,[q, Z, P]$
(16) $[P, Z, q] \rightarrow 0\,[q, Z, q]$

Definations :-

Recursive language — A language L is said to be recursive if there exist a turing machine which will accept all the strings in L and reject all the strings not in L.

→ The turing machine will half every time and give an answer (accepted or rejected) for each and every string input.

Recursively enumerable language — A language L is said to be recursively enumerable language if there exist a turing machine which will accept (and therefore half) for all the input string which are in L.
But may or may not half for all input strings which are not in L.

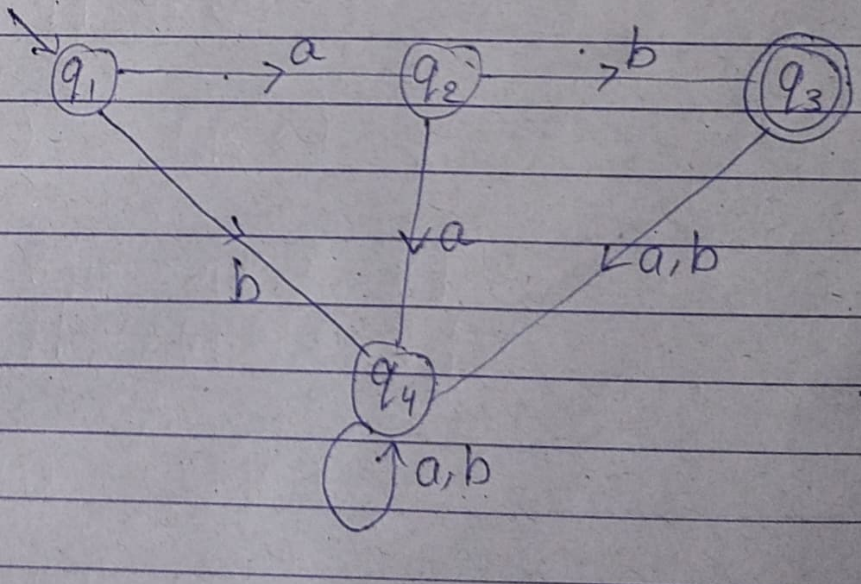
Decidable language – A language L is decidable if it is recursive language. All decidable languages are recursive languages & vice-versa.

Partially decidable language – A language L is partially decidable if L is a recursively enumerable language

Undecidable language – A language is undecidable if it is not decidable.

→ An undecidable language may sometimes be partially decidable but not decidable.

→ If a language is not even partially decidable then there exist no turing machine for that language.



string = ab
accep

not accept – ba, aa, abab, abba