Grammer

$G = \{V, T, P, S\}$ —
- V - Set of (NT) variables [Capital lett)
- T - Set of Terminal symbols [lowercase, digit]
- P - Production rule
- S - Start symbol [$S \in V$]

$\longrightarrow$

Q) Language that generate equal No of a & b in form $a^n b^n$

$\longrightarrow L = \{ab, aabb \ldots a^n b^n\}$

$G = \{V, T, P, S\}$

$$G = \left\{ \left[(S, A), (a, b), \left[\begin{array}{l} (S \to aAb \\ A \to aAb / \epsilon) \end{array} \right], S \right] \right\}$$

---

1) Sentential Form — Left most, Rightmost

2) tree (Later) (Parse)

Q.) For a grammer

$S \to A1B \mid A \to 0A/\epsilon \mid B \to 0B/\ 1B/\epsilon$

For string 1001

$\to$ (Left ki value pehle substitute)

Leftmost Sentential	Rightmost sentential
$S \to \underline{A}1B$ [start]	$S \to A1B$ (start)
$\to 1B$ [$A \to \epsilon$]	$\to A10B$ [$B \to 0B$]
$\to 10B$ [$B \to 0B$]	$\to A100B$ [$B \to 0B$]
$\to 100B$ [$B \to 0B$]	$\to A1001B$ [$B \to 1B$]
$\to 1001B$ [$B \to 1B$]	$\to A1001$ [$B \to \epsilon$]
$\to 1001$ [$B \to \epsilon$]	$\to \underline{1001}$ [$A \to \epsilon$]

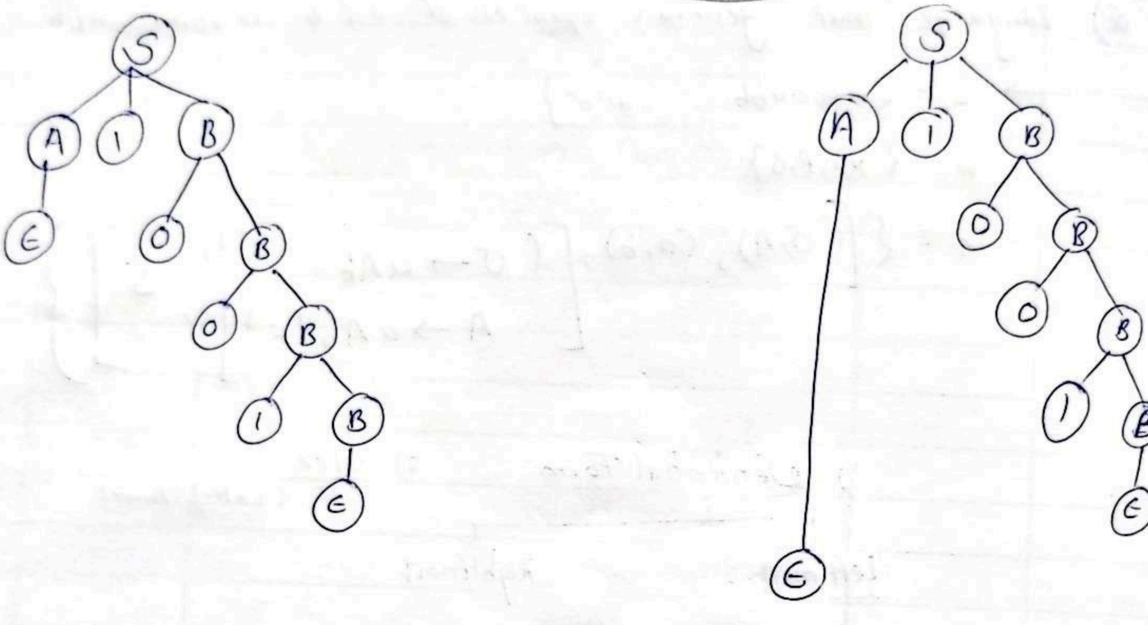
(Final string ar contain all terminal)

derivation using tree

1) Root node — S
2) Intermidiate node — V
3) leaf nodes — T or ϵ

Q. For a grammer $S \to A1B$ 1001
$A \to 0A/\epsilon$
$B \to 0B/1B/\epsilon$

Leftmost   Rightmost



grammer $\longrightarrow$ string

Q.) $E \to E + T/T$
$T \to T * F/F$
$F \to (E)\ |a|b$

Solution $\longrightarrow$

Variable $\to \{E, T, F\}$
terminals $\to \{+, *, (,), a, b,\}$
Production rule $\to \left\{ \begin{array}{c} E + T/T \\ T * F/F \\ (E)\ |a|b \end{array} \right\}$

start symbol $\to$ E
$S = E$

we need $(a+b) * a+b$.

* $E = E + T \quad [E \to E+T]$

$E = ET + T \quad [E \to T]$

$E \Rightarrow T * F + T \quad [T \to T*F]$

$E \to F * F + T \quad [T \to F]$

$E \to (E) * F + T \quad [F \to (E)]$

$E \to (E+T) * F + T \quad [E \to E+T)$

$E \to (T+T) * F + T \quad (E \to T)$

$E \to (F+F) * F + F \quad (T \to F)$

$E \to (a+b) * a + b \quad (F \to a/b)$

(part.1) language → grammer.

Q) $L = \{\epsilon, a, aa \ldots\}$

$S \to aS$

$S \to \epsilon$

string.

$S \to aS$

$S \to aaS$

$S \to aaS$

$\to aaa \; (S \to \epsilon)$

$L = \{a, aa, aaa.\}$

$S \to aS$

$S \to a$

string.

$S \to aS$

$\to aaS$

$\to aaaS$

$\to aaaaS$

$\to aaaa \; (S \to \epsilon)$

3) Language → $L[b, ab, aab, aaab \ldots]$

Grammar
$S \rightarrow aS$
$S \rightarrow b$

String
$S \rightarrow aS$
$S \rightarrow aaS$
$S \rightarrow aaaS$
$S \rightarrow aaab \quad (S \rightarrow b)$

4) $L = \{W \in \{a,b\}^*\}$

$L = \{\epsilon, a, b, aa, ab, ba, bb \ldots\}$

$S \rightarrow aS$
$S \rightarrow bS$
$S \rightarrow \epsilon$

String
$S \rightarrow aS$
$S \rightarrow abS$
$S \rightarrow abbS$
$S \rightarrow abbaS$
$S \rightarrow \underline{\underline{abba}} \quad (S \rightarrow \epsilon)$

5) $L = \{a^n b^n \mid n \geq 0\}$

$L = \{\epsilon, a, b, ab, aabb, aaabbb\}$

$S \rightarrow aSb$
$S \rightarrow ab$

String
$S \rightarrow aSb$
$S \rightarrow aabb$
$S \rightarrow aaaSbbb$
$S \rightarrow aaabbb \quad (S \rightarrow ab)$

6) $L = \{a^n b^n \mid n \geq 1\}$

$L = \{ab, aabb, aaabbb\}$

$S \to aSb$

$S \to ab$

String

$S \to aSb$

$\to aabSbb$

$\to aaaSbbb$

$\to aaaabbbb \; (S \to ab)$

[symmetric]

Left

Right

both

(Palindrome String) → babab

Language → Grammer

Right to left & left to Right same.

(in odd we get center one terminal ~~variable~~) (even we get 2)

Q.) $L = \{ w \in (a,b)^* \mid w$ is a palindrome of odd length

$S \to aSa$

$S \to bSb$

$S \to a$

$S \to b$.

→ Generation of string

$S \to aSa$

$S \to abSba$

$S \to ababa \; (S \to a)$

(String length odd)

For even. same Q. (abaaba)

$S \to aSa$

$S \to bSb$

$S \to aa$

$S \to bb$

Generating string

aSa	bSb
abSab	basba
abaaab	babbba

For even no condition $S \to aSa$ $S \to bSb$ $S \to \epsilon$

4) $L = \{w \in \{a,b\}^* \mid w \text{ is a palindrome}\}$

	even length	odd length
$S \to aSa$	$\to aSa$	$\to aSa$
$S \to bSb$	$\to abSba$	$\to abSba$
$S \to a$	$\to abaSaba$	$\to ababa\ (S \to a)$
$S \to b$	$\to abaaba\ (S \to \epsilon)$	
$S \to \epsilon$		

---

5) $L = \{w \in (a,b)^* \mid w \text{ is a palindrome } |w| \geq 0\}$

	even	odd
$S \to aSa$		
$S \to bSb$	even	odd
$S \to a$	aSa	aSa
$S \to b$	$abaSba$	$abaSba$
$S \to aa$	$abaaba$	$ababa$
$S \to bb$		

$CFL \longrightarrow CFG$

Kan-2) Language to gramer

Q) $L = \{a^n b^{n+2}, n \geq 0\}$

$L = \{abbb, aabbbb \dots a^n b^{n+2}\}$

$= S \rightarrow aSb \mid bb$

Q) $L = \{a^{2n} b^n, n \geq 0\}$ aaaaaabbb

$L = \{aab, aaaabb \dots a^{2n} b^n\}$

$\cancel{S \rightarrow aSb \mid aa}$

$S \rightarrow aaSb \mid \epsilon$

Q) $L = \{a^{2n} b^n, n \geq 1\}$

$S \rightarrow aaSb \mid aab$

Q) $L = a^{2n+3} b^n, n \geq 0$

$S \rightarrow L = \{aaa, aaaaab, aaaaaaabb, n \geq 0\}$

$S \rightarrow aaSb \mid aaa$

CFL → CFG.

(Part 3)

1) $a^m b^n$, $m > n$, $n \geq 0$

$L = \{ a, \overset{aa,aaa)}{aab}, aaab, aaabb \cdots \}$

$\boxed{S \to \cancel{a}Sb \,/\, aS /}$

genuine bnh

$S \to AS_1$

$A \to aA / a$ → (one or mon occurenc)

$S_1 \to aS_1b \,/\, \epsilon$

2) $a^m b^n$, $m \geq n$, $n \geq 0$

$L = \{ \epsilon, a, aa, aaa, ab, aab, aaab \ldots \}$

$S \to AS_1$

$A \to aA / \epsilon$

$S_1 \to aS_1b / \epsilon$

3) $a^m b^n$ | $n > m$

$L = \{ b, abb, aabbb, abbb \}$

$S \to AS_1B$

$B \to bB / b$

$S_1 \to aS_1b | \epsilon$

4) $a^n b^n$ $L = \{ab,$

~~S → aSb~~ / Any number of a & any no of b. (equal)

$L = \{ab, ba, aabb, baba, abab, abba, baab \ldots\}$

$S \rightarrow aSb \mid bSa \mid SS \mid \epsilon \rightarrow$ (empty)

(For starting & ending with same)

(Part 34)

$L = \{a^i b^j c^k \mid i+j = k \mid i,j \geq 1\}$

$L = \{abcc, aabbcccc, aaabbbcccccc, abbccc \ldots\}$

$i,j \geq 0$

$\{\epsilon,$

$a^i b^j c^{i+j}$

$a^i b^j c^j c^i$

~~aSbc~~

$S \rightarrow aSc / aXc$

$X \rightarrow bXc / bc$

$$\boxed{\begin{array}{l} S \rightarrow aSc / aXc \\ X \rightarrow bXc / \epsilon \end{array}}$$

Q.2) $L = \{ a^i b^j c^k \mid i = j + k \mid j, k \geq 1 \}$

$a^{j+k} b^j c^k$

$a^j a^k b^j c^k$

$a^k a^j b^j c^k$

$S \rightarrow aSc \mid aXc$

$X \rightarrow aXb \mid ab$

$j, k \geq 0$

$S \rightarrow aSc \mid X$

$X \rightarrow aXb \mid \epsilon$

Q.3) $L = \{ a^i b^j c^k \mid i = j \mid i, j, k \geq 1 \}$

$\rightarrow S \rightarrow XY$

$X \rightarrow aXb \mid ab$

$Y \rightarrow cY \mid c$

(for same number of a & b)

Q. $L = \{ a^i b^j c^k \mid j = k \mid i, j, k \geq 1 \}$

$\rightarrow S \rightarrow XY$

$X \rightarrow aX \mid a$

$Y \rightarrow bYc \mid bc$

(Part -)

$L = \{0^i 1^j 0^k \mid j > i+k \mid i,k \geq 1\}$

$\rightarrow \{0\,111\,0,\ 0\,111\,00,\ 00\,111\,10\}$ ...

$S \rightarrow XYZ$

$X \rightarrow 0X1 \mid 01 \longrightarrow (01)$

$Y \rightarrow 1Y \mid 1 \longrightarrow (1)$

$Z \rightarrow 1Z0 \mid 10 \longrightarrow (10)$

0 111 0

$\rightarrow$ " ——— " $(i,k \geq 0)$

$L = \{0^i 1^j 0^k \mid j > i+k \mid i,k \geq 0\}$

$S \rightarrow XYZ$ $\quad L = \{\cancel{010}, 1, 0\,111\,0, \cdots\}$.

$X \rightarrow 0X1 \mid \epsilon$

$Y \rightarrow 1Y \mid \cancel{\epsilon}\,1$ (important)

$Z \rightarrow 1Z0 \mid \epsilon$

(Part-5) ((Parathesis)

→ (Give CFG For matching Parenthesis)

→ $S \rightarrow (S) \mid SS \mid \epsilon$

For nesting ((( )))

$S \rightarrow (S)$

derivation of string

For $\underset{S}{( )}$ $\underset{S}{(( ))}$

$\boxed{S \rightarrow SS}$

Give CFG For set of even length string in $\{a,b\}^*$ [Length - 0, 2, 4, 6, 8..]

→ $S \rightarrow aSa \mid bSb \mid aSb \mid bSa \mid \epsilon$

$L = \{\epsilon, ab, aabb, abab, ba, baab, bbaa, \}$.

Given CFG for set of odd length string $(a,b)^*$ (odd length = 1, 3, 5, 7-}

$S \rightarrow aX \mid bX$

$X \rightarrow aS \mid bS \mid \epsilon$

{ a, b, aab, aaa, ... }

$(a,b,c)^*$

$S \rightarrow aX \mid bX \mid cX$

$X \rightarrow aS \mid bS \mid cS \mid \epsilon$

→ Same

$S \rightarrow aSb \mid bSa \mid aSa \mid bSb \mid$

$S \rightarrow a \mid b$

$L = \{a^n b^m \mid n \neq m\}$

$\{ab, b, a, \cancel{ba}, aab, aa, bbb$

$S \rightarrow aSb \mid X \mid Y$

$X \rightarrow aX \mid a$

$Y \rightarrow bY \mid b$

$S \rightarrow aSb$

$\rightarrow aaSbb$

$\rightarrow aaXbb$

$\rightarrow \boxed{aaaXbb}$

$\boxed{\text{String derivation}}$

$L = \{a^n b^m c^k \mid m \leq k\}$

$L = \{\epsilon, abc, a\underline{b}cc, aabcc, aa\underline{bb}\underline{cc},\}$

$S \rightarrow XYZ$

$X \rightarrow aX \mid \epsilon$

$Y \rightarrow bYc \mid \epsilon$ (equal of both)

$Z \rightarrow cZ \mid \epsilon$

(Simplification of CFG)

① eliminate (ϵ) Null

② eliminate Unit

③ Elimination of Useless symbol

- (Non generating)
- (Not reachable)

* after that any we can write it in simplified form:-

1) CNF (Chomsky Normal Form)
2) GNF (Greibach Normal Form)

---

① elimination of Null

Q. $\longrightarrow$ $\left.\begin{array}{l} S \rightarrow aS \mid A \\ A \rightarrow \epsilon \end{array}\right\}$ a^* $L = \{a^n \mid n \geq 0\}$

$\longrightarrow$ Nullable Variable $\{A, S\}$

$S \rightarrow aS \mid a \mid \cancel{A} \mid \epsilon$ $\longrightarrow$ (No meaning)

$\cancel{A \longrightarrow \epsilon}$

$S \rightarrow aS \mid a \mid \epsilon$

$\longrightarrow L = \{a^n \mid n \geq 0\}$

$[L(G_1) = L(G) = \{a^n \mid n \geq 0\}]$

→ Now to construct CNF & GNF we need to Remove (ϵ)

Q.) .... $L(G) - \{\epsilon\}$

$S \rightarrow aS \mid a$

(Language changes)

$L = \{a^n \mid n \geq 1\}$

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

Remove Null (ϵ) production from Grammer:-

$S \rightarrow ABAC$

$A \rightarrow aA \mid \epsilon$

$B \rightarrow bB \mid \epsilon$

$C \rightarrow c$

→ Nullable variable $\{A, B\}$

After Remove B   A remove

$S \rightarrow ABAC \mid ABAC \mid ABC \mid BC \mid AAC \mid AC \mid C$

Removing this   B

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

$C \rightarrow c$

(2) Q) $S \rightarrow ABC$

$A \rightarrow aA / \epsilon$

$B \rightarrow bB / \epsilon$

$C \rightarrow c$

$\rightarrow$ Nullable variable (A, B)

$S \rightarrow ABC \mid BC \mid AC \mid c$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

$C \rightarrow c$

Q.) $S \rightarrow AbaC$

$A \rightarrow BC$

$B \rightarrow b \mid \epsilon$

$C \rightarrow D \mid \epsilon$

$D \rightarrow d$

$\rightarrow$ Nullable variable $\{A, B, \overset{C}{\cancel{D}}\}$

$S \rightarrow AbaC \mid baC \mid Aba \mid ba$

$A \rightarrow BC \mid C \mid B$

$B \rightarrow b \mid$

$C \rightarrow D$

$D \rightarrow d$

② Remove unit production from CFG.

$(A \to B)$ — single variable derived, so its unit production

$(A \to a)$, $S \to aB$, $A \to AB$ — Not unit production

Q.) Remove Unit production from grammer

$S \to AB$
$A \to a$
$B \to C \mid db$
$C \to D$
$D \to E$
$E \to a$

unit productions: $B \to C$, $C \to D$, $D \to E$, $E \to a$

$S \to AB$
$A \to a$
$B \to a \mid b$
$C \to a$
$D \to a$
$E \to a$

Q.)

$(S \to B \to A \to ba \mid bb)$

$S \to B \to bba$

$S \to aA \mid B$
$A \to ba \mid bb$
$B \to A \mid bba$

$S \to aA \mid bba \mid bb \mid bba$
$A \to ba \mid bb$
$B \to ba \mid bb \mid bba$

$S \to ABA \mid BA \mid AA \mid A \mid B$
$A \to aA \mid a$
$B \to bB \mid b$

$S \to ABA \mid BA \mid AA \mid aA \mid a \mid bB \mid b$
$A \to aA \mid a$
$B \to bB \mid b.$

* Remove Useless Production / symbols :-

1) Remove Non-generating symbols / production.
2) Remove Non-Reachable symbols / Production.

Q.) $S \to AB / a$ terminals $\{a, b\}$

$A \to BC / b$ variable $\{A, B, C\}$

$B \to aB / C$

$C \to aC / B$

$\downarrow$

(Non generating)

Generating	Non-generating
$\{a, b, S, A,$	

So now

$S \to \cancel{A}B / a$ $\longrightarrow$ $S \to a$

$A \to \cancel{B}C / b$ $\qquad A \to b$

$\downarrow$ (Non Reachable)

$\underline{S \to a}$

Q. $S \to A\cancel{B}C / BaB$

$A \to aA / B\cancel{A}C / aaa$

$B \to bBb / a$

~~$C \to CA / AC$~~

$\{a, b\}$

$\{a, b, A, B\}$

$\{a, b, A, B, \underline{S}\}$

Now Remove C

we get

$\underline{S \to BaB}$

$A \to aA / aaa \xrightarrow{\text{Non Reachable}}$ $S \to BaB$

$\underline{B \to bBb / a}$ $\qquad B \to bBb / a$

3)

Chomsky Normal Form CNF.

$$\begin{bmatrix} A \to a \\ A \to BC \end{bmatrix} \longrightarrow \text{one terminal or 2 variable.}$$

Steps to Follow:-

1) 1) eliminate null prod $\{\epsilon, \Lambda, \lambda\}$
   2) eliminate the Unit prod
   3) eliminate useless symbols.
2) eliminate terminals on RHS having 2 or more length
   * $A \to aB \longrightarrow A \longrightarrow C_a B$ ✓
     eliminate $C \to a$ ✓
   * $A \to aBb$ $A \to C_a B C_b$ ✗
     $C_a \to a$ ✓
     $C_b \to b$ ✓
   * $A \to aAbB$ $A \to C_a A C_b B$ ✗
     $C_a \to a$ ✓
     $C_b \to b$ ✓
3) only 2 variable on RHS....

Q.) $A \to C_a \, \underline{A C_b B}$

$A \longrightarrow C_a D_1$ ✓ $(D_1 = A C_b B)$

$D_1 \longrightarrow A D_2$ ✓

$D_2 \to C_b B$ ✓

Q.) $S \rightarrow aAbB$

$A \rightarrow aA / a$

$B \rightarrow bB / b$

1) Rule Remove (λ.... unit... useless)

— None

2) terminals having 2 or more length eliminate

$S \rightarrow aAbB$

$A \rightarrow aA / a$

$B \rightarrow bB / b$

$\rightarrow S \rightarrow C_a A C_b B$

$A \rightarrow C_a A / a$

$B \rightarrow C_b B / b$

$C_a \rightarrow a$

$C_b \rightarrow b$

3)

$S \rightarrow C_a D_1$

$D_1 \rightarrow A D_2$

$D_2 \rightarrow C_b B$

$A \rightarrow C_a A / a$

$B \rightarrow C_b B / b$

$C_a \rightarrow a$

$C_b \rightarrow b$

GNF — Greibach Normal Form

$$\boxed{A \rightarrow a\alpha}$$

$A \rightarrow$ Variable
$a \rightarrow$ terminal
$\alpha \rightarrow$ It is a string of zero or more variables.

$A \rightarrow a$
$A \rightarrow a\underline{B}$
$A \rightarrow a\underline{BC}$
$A \rightarrow a\underline{BCA}\ldots$

} RHS of each production should start with single terminal followed by string of variables

---

Q.) $S \rightarrow XX \mid a$
$X \rightarrow SS \mid b$

Step ① → eliminate ϵ
" unit
" useless

Answer
$A_1 \rightarrow A_2A_2 \mid a$ $A_2 \rightarrow A_2A_2A_1 \mid aA_1 \mid b$

Step ② → modify & Rename as A_1 & A_2

$A_1 \rightarrow A_2A_2 \mid a$

$A_2 \rightarrow A_1A_1 \mid b.$

③ Rule every production should be of form.

$[A_i \rightarrow A_j\alpha]$ with $\boxed{i \leq j}$

so need to modify → only first value to be substitute

$A_2 \rightarrow \underline{A_1}A_1 \mid \cancel{b}$

$\longrightarrow A_2A_2\underline{A_1} \mid aA_1$

Step 4 :- Removing left Recursion.

$A_1 \rightarrow A_2A_2 \mid a$

$A_2 \rightarrow A_2 A_2 A_1 \mid aA_1 \mid b.$

Left Recursion

$A_2 \rightarrow A_2\ A_2A_1 \mid aA_1 \mid b$

$A \quad A \quad \alpha \mid \beta \mid \beta$

$\boxed{A \rightarrow A\alpha \mid \beta}$

Left Recursive Grammer

$A_2 \rightarrow \begin{cases} A_2 \rightarrow aA_1B_2 \mid bB_2 \mid aA_1 \mid b \\ B_2 \rightarrow A_2A_1B_2 \mid A_2A_1 \end{cases}$

$\boxed{\begin{array}{l} A \rightarrow \beta B \mid \beta \\ B \rightarrow \alpha B \mid \alpha \end{array}}$

Right Recursive grammar

Final Anwer

$A_1 \rightarrow A_2 A_2 \mid a$ —— ①

$A_2 \rightarrow aA_1B_2 \mid bB_2 \mid aA_1 \mid b$ —— ② → GNF

$B_2 \rightarrow A_2A_1B_2 \mid A_2A_1$ —— ③

Now to make ① & ③ in GNF we can.

$A_1 \rightarrow aA_1B_2A_2 \mid bB_2A_2 \mid aA_1A_2 \mid bA_2 \mid a$ (by 2)

$A_2 \rightarrow$

$B_2 \rightarrow aA_1B_2A_1B_2 \mid bB_2A_1B_2 \mid aA_1BA_1B_2 \mid bA_1B_2 \mid$
$aA_1B_2A_1 \mid bB_2A_1 \mid aA_1A_1 \mid bA_1$

variables → (A_1, A_2, B_2)

T → (a, b)

Start symbol → $\underline{\underline{A}}$

Pushdown Automata

$RG \rightarrow RL \rightarrow FA$ → NFA, → DFA

1) Finite Automata cannot recognise more powerful Language than RL.

Ex → $a^n b^n \mid n \geq 1$

cannot be recognised by FA

memory not sufficient.

$CFG \rightarrow CFL \rightarrow PDA$ → DPDA (one move in every situation), → NDPDA (powerful) (multiple moves in every situation)

All (cannot) be — DPDA

(can be) All — NDPDA

$\boxed{\dfrac{PDA > FA}{\text{powerful}}}$ → ∴ $\boxed{FA + \text{stack} \rightarrow PDA}$

★ (Stack.

→ TOP

Z_0 → infinite size stack so no overflow. (Push / POP) operation performed

↓ (initial stack symbol so No underflow)

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

Q = set of states

$\Sigma \rightarrow$ Input alphabet

(uppercase gama) (toe) $\Gamma \rightarrow$ Stack symbol

$\delta = Q \times (\Sigma \cup \epsilon) \times \Gamma$ to $Q \times \Gamma^*$

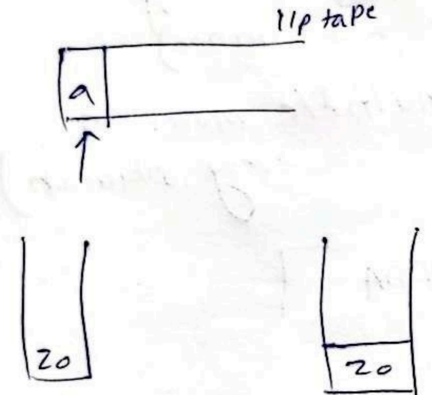
transition.

$$\delta = Q \times (\Sigma \cup \epsilon) \times \Gamma \text{ to } Q \times \Gamma^*$$

- Q → current state
- $(\Sigma \cup \epsilon)$ → i/p
- Γ → Top most stack symbol
- Q → next state
- Γ^* → operation on stack
  - * Push
  - * Pop
  - * No-operation

(Operations on PDA)

Push



→ $q_0 \xrightarrow{a, z_0 / a z_0} q_1$

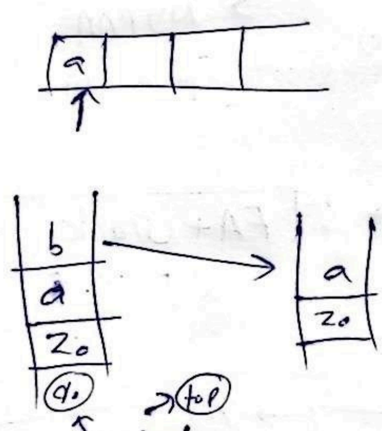
→ $\delta(q_0, a, z_0)$

- q_0 → (current state)
- a → (input)
- z_0 → (stack top)

$\delta(q_0, a, z_0) = (q_1, \underline{a z_0})$

$a z_0$ → (stack top now)

Pop

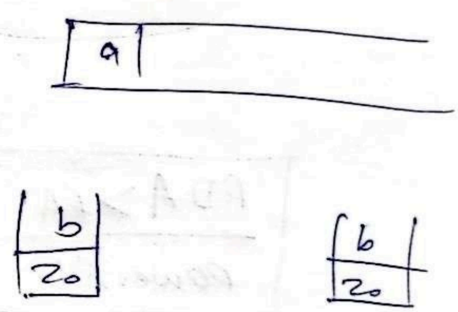


(q0) → (top)

$q_0 \xrightarrow{a, b / \epsilon} q_1$

$\delta(q_0, a, b) = (q_1, \epsilon)$

Skip / No operation

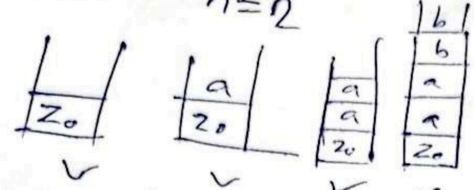


$q_0 \xrightarrow{a, b / b} q_1$

$\delta(q_0, a, b) = (q_1, b)$

* Q.) Push down automata $a^nb^n, n \geq 1$ PDA

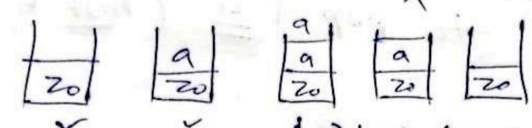
Consider $n=2$



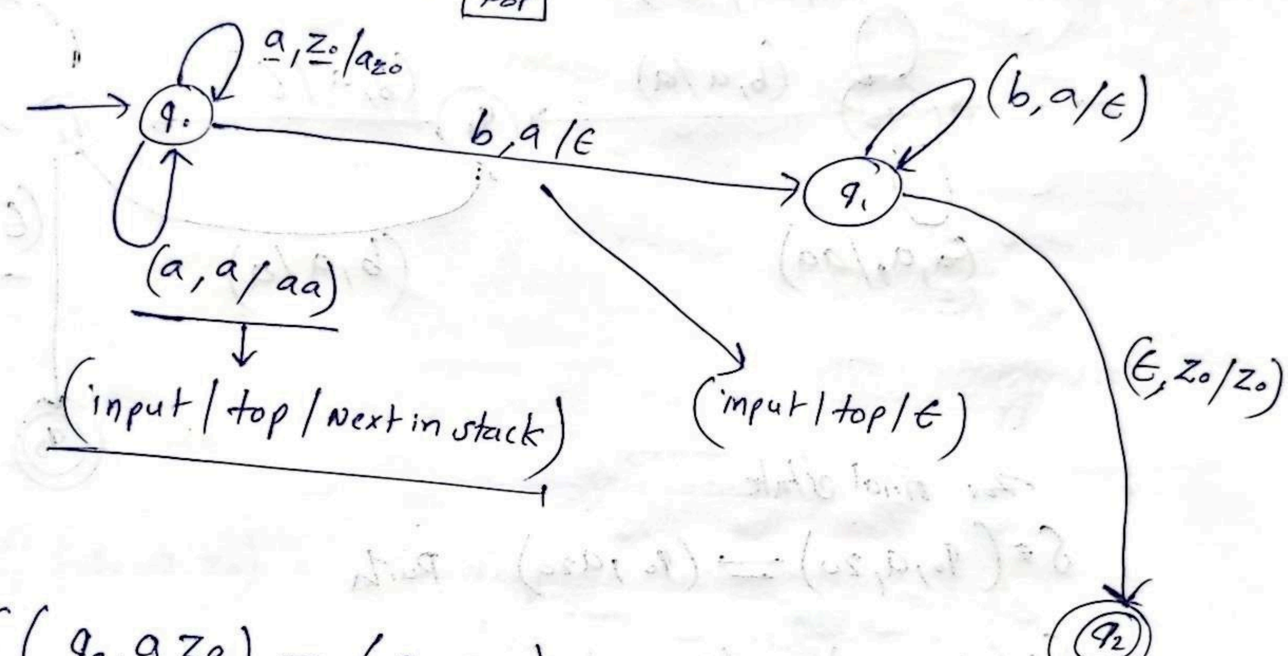
Not accepted

Q.

$\boxed{aabb\epsilon}$



Accepted



$\delta(q_0, a, Z_0) = (q_0, aZ_0)$ Push

$\delta(q_0, a, a) = (q_0, aa)$ Push

$\delta(q_0, b, a) = (q_1, \epsilon)$ Pop

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

$\delta(q_1, \epsilon, Z_0) = \delta(q_2, Z_0)$ Skip / No

or

$\delta(q_1, \epsilon, Z_0) = (q_1, \epsilon)$

(empty)

DPDA → Final state
→ empty stack

For empty stak.

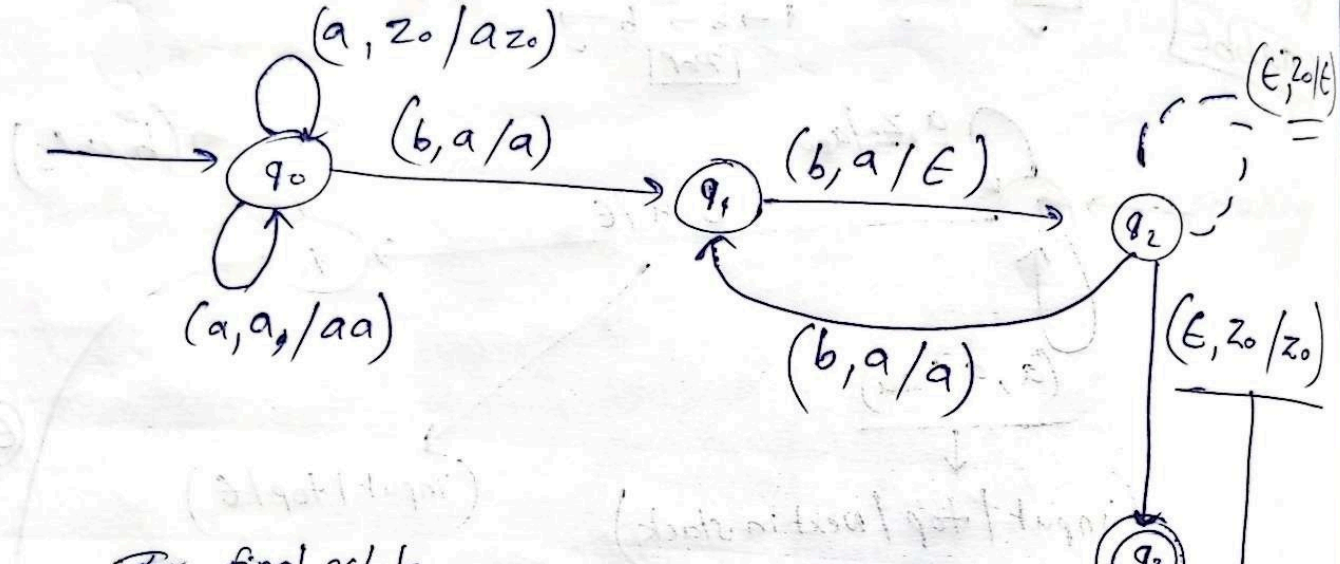
apply selfloop of $\epsilon, Z_0/\epsilon$ on q_1

Q.) $a^n b^{2n}, \quad n \geq 1$

lets take $n=2$

aa bbbb ϵ $\quad$ Z_0

logic → skip/push 2 b to pop 1 a (first b skip)



For final state

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$ Push

$\delta(q_0, a, a) = (q_0, aa)$ Push

$\delta(q_0, b, a) = (q_1, a)$ skip

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

$\delta(q_2, b, a) = (q_1, a)$ skip.

$\delta(q_2, \epsilon, Z_0) = (q_3, Z_0)$ skip. → $\left[\delta(q_2, \epsilon, Z_0) = \underline{(q_2, \epsilon)}\right]$

For empty

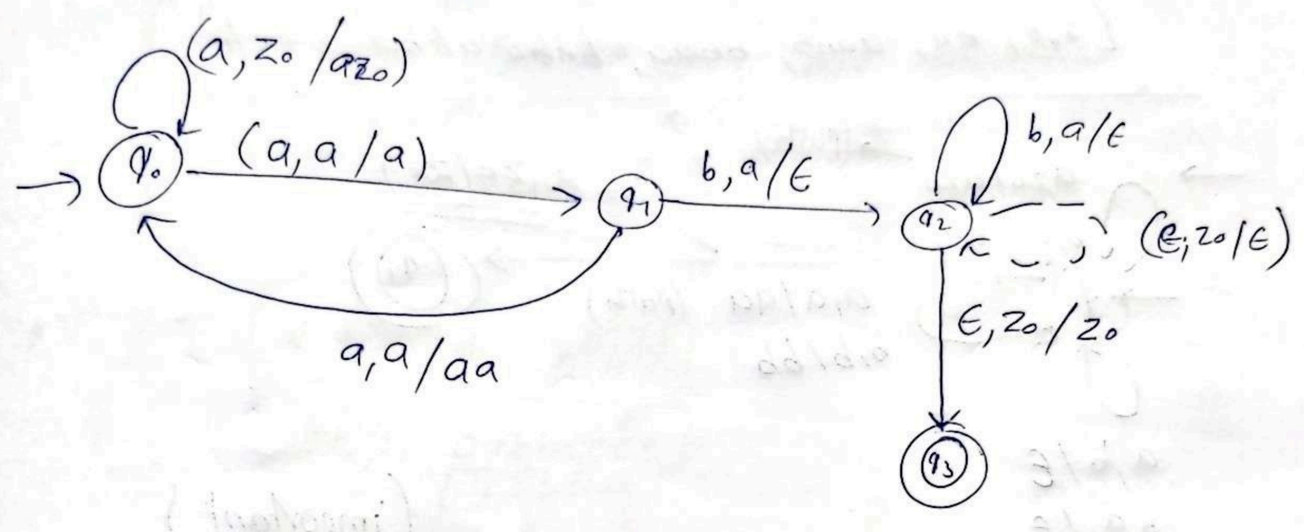
Replace final with ϵ & apply self loop

Q) $a^{2n} b^n$, $n \geq 1$

Push skip

$\{ aab,\ aaaabb,\ aaaaaabbb \ldots \}$

Push skip p p



Final state

For empty stack

~~δ(a, z0/a~~

$$\delta(q_0, a, z_0) = (q_0, a z_0)$$

$$\delta(q_0, a, a) = (q_1, a)$$

$$\delta(q, a, a) = (q_0, aa)$$

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

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

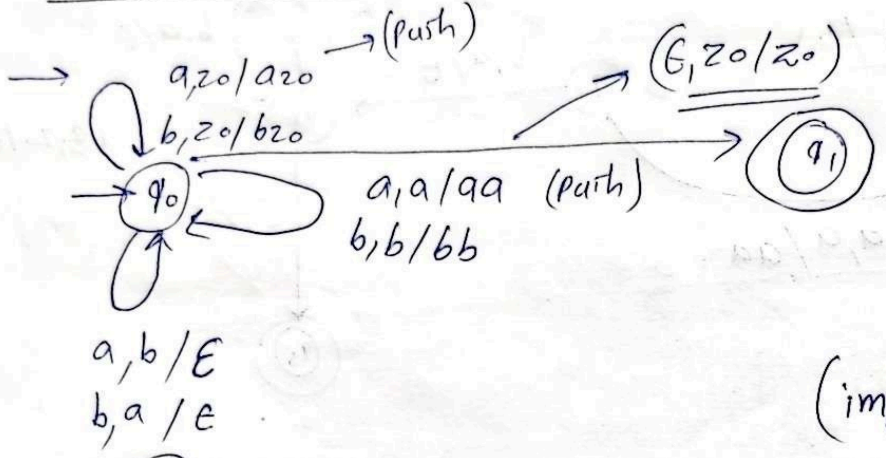
$$\delta(q_2, \epsilon, z_0) = (q_3, z_0) \longleftrightarrow \delta(q_2, \epsilon, z_0) = (q_2, \epsilon)$$

Q.) desing PDA For CFL. (Number of a's) = (number of b's)

$L = \{ w \in (a,b)^* \mid n.a(w) = n.b(w)$

equal no of a's & b's

{ab, ba, aabb, bbaa, baba, abba, baab}



(important)

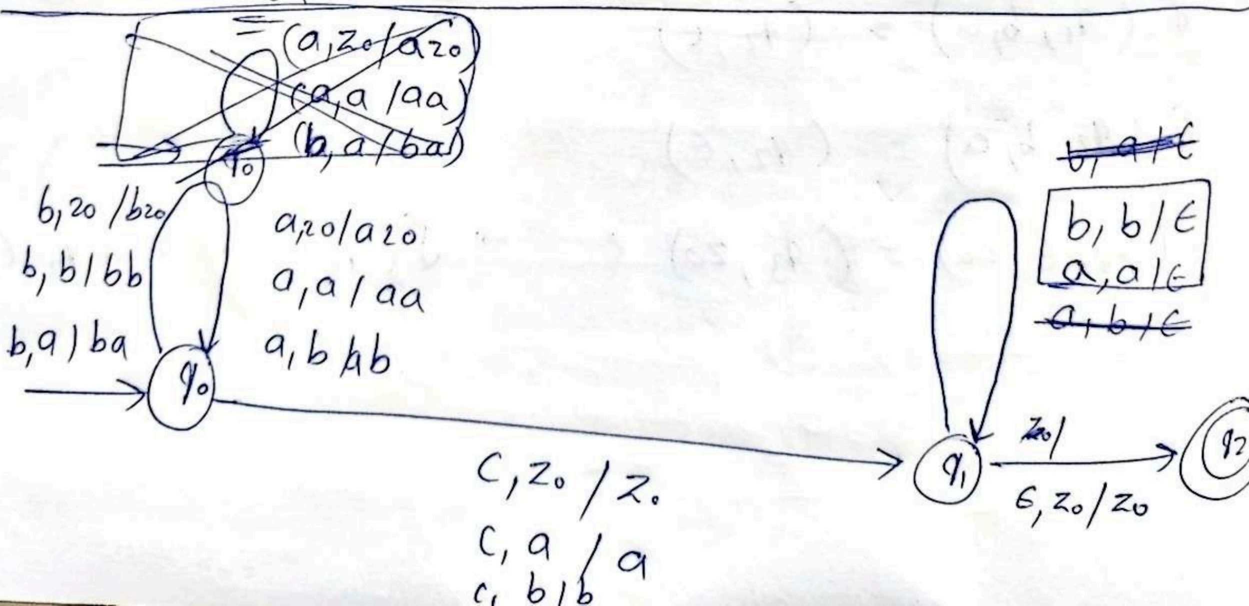
design PDA For CFL

$L = \{ wcw^R \mid w \in (a,b)^* \}$

w   w Reverse

w = aab
wR = baa

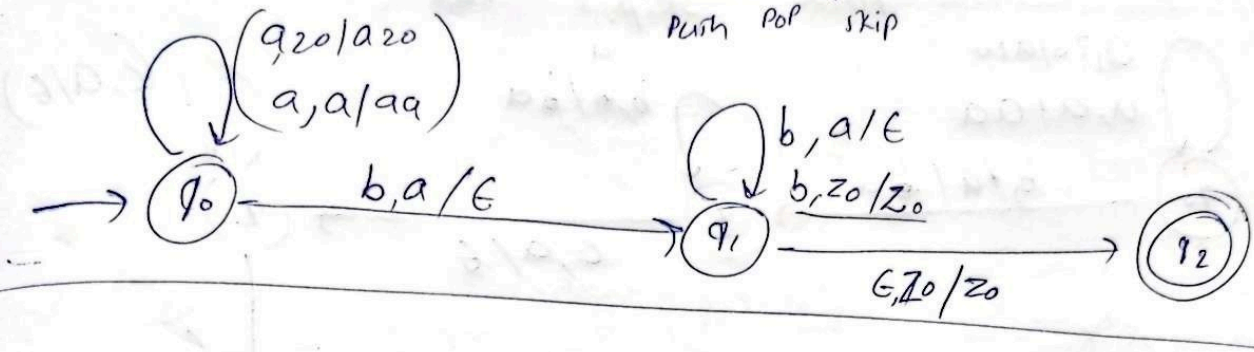
push skip pop
aabcbaa



* $L = \{a^n b^m \mid n \leq m\}$

$\{ab, abb, aabb, aabbb \ldots\ aabbbb \ldots\}$

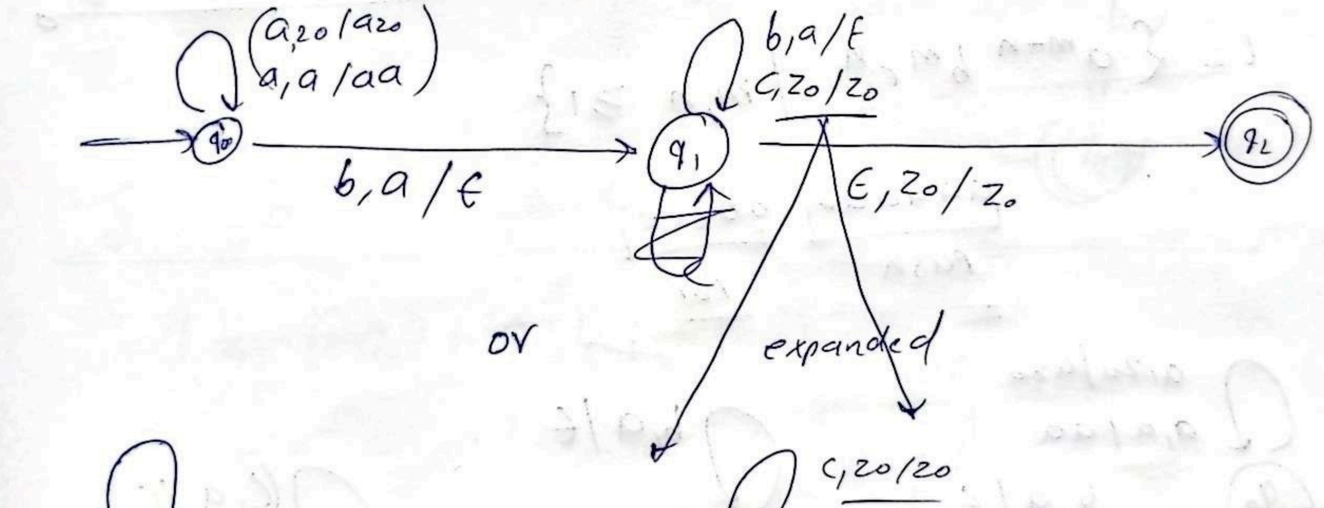
push pop skip



* $L = \{a^n b^n c^m \mid n, m \geq 1\}$

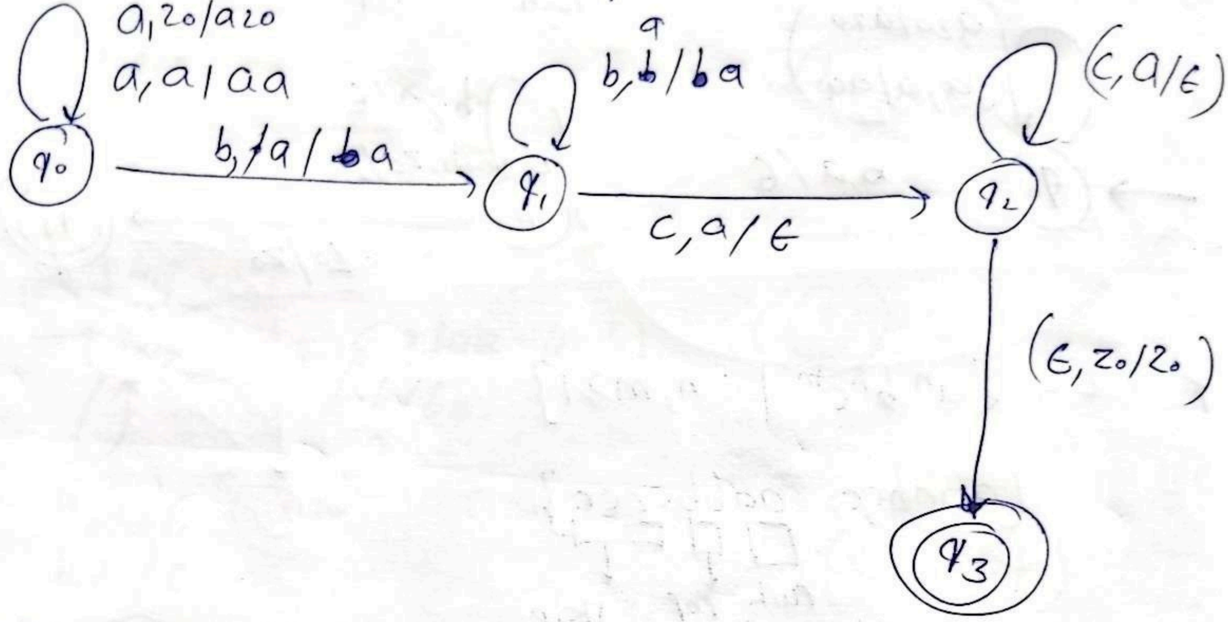
$\{aabbcc, aabbcccc\}$

push pop skip



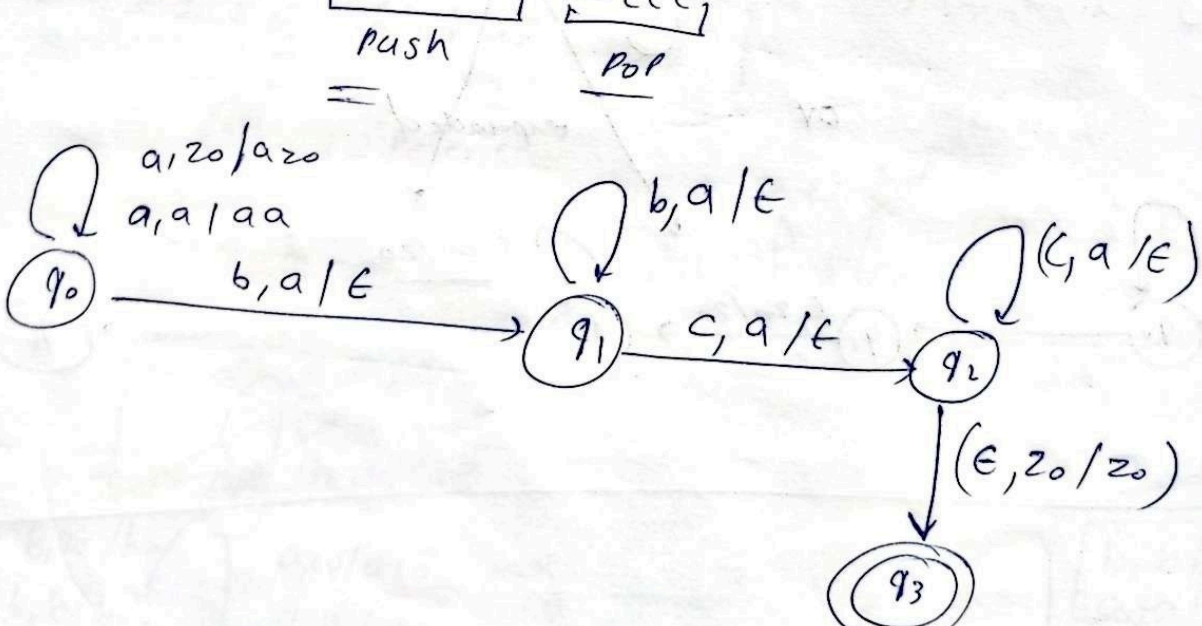
$L = \{a^{n} b^{m} c^{n} \mid n, m \geq 1\}$

$\underbrace{aa}_{\text{push}} \ \underbrace{bbb}_{\text{skip}} \ \underbrace{cc}_{\text{pop}}$



$L = \{a^{m+n} b^{m} c^{n} \mid m, n \geq 1\}$

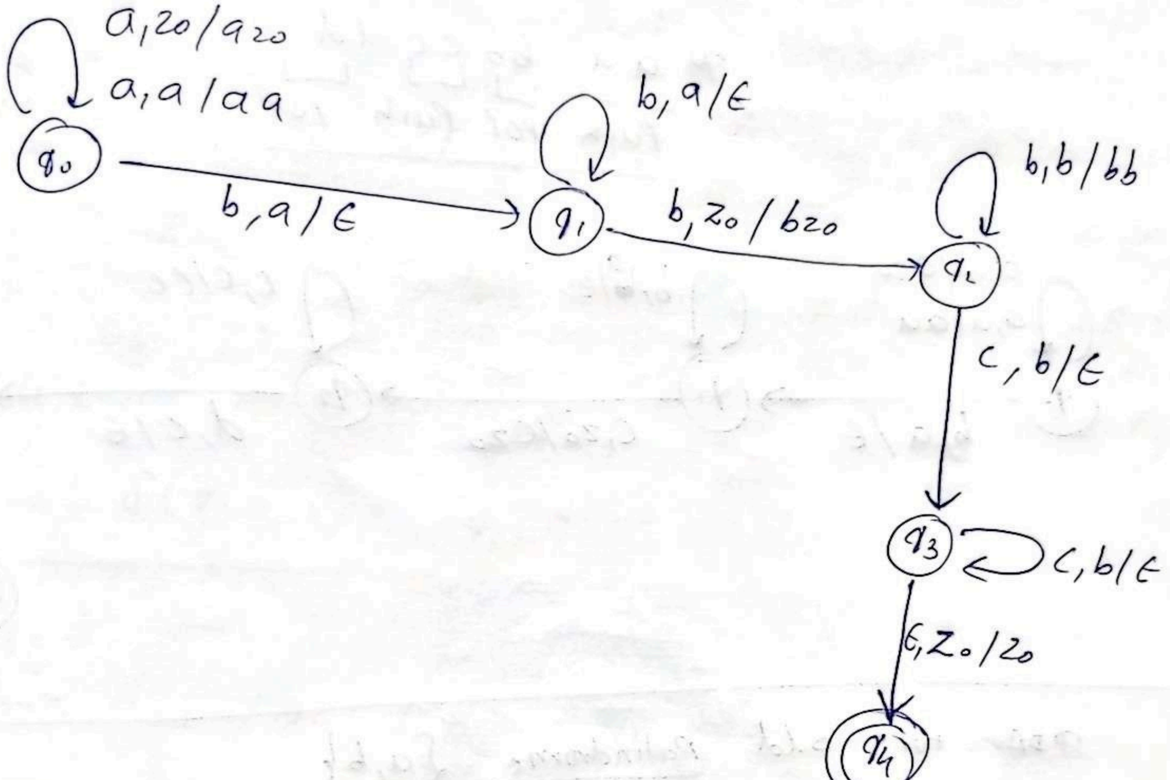
$\underbrace{aaaaaa,}_{\text{push}} \ \underbrace{bbccc,}_{\text{pop}}$



$L = \{ a^m b^{m+n} c^n \mid m,n \geq 1 \}$

aa bb bbb ccc

Push pop Push Pop



$L = (a^n b^{2n+1} \mid n \geq 1 \}$

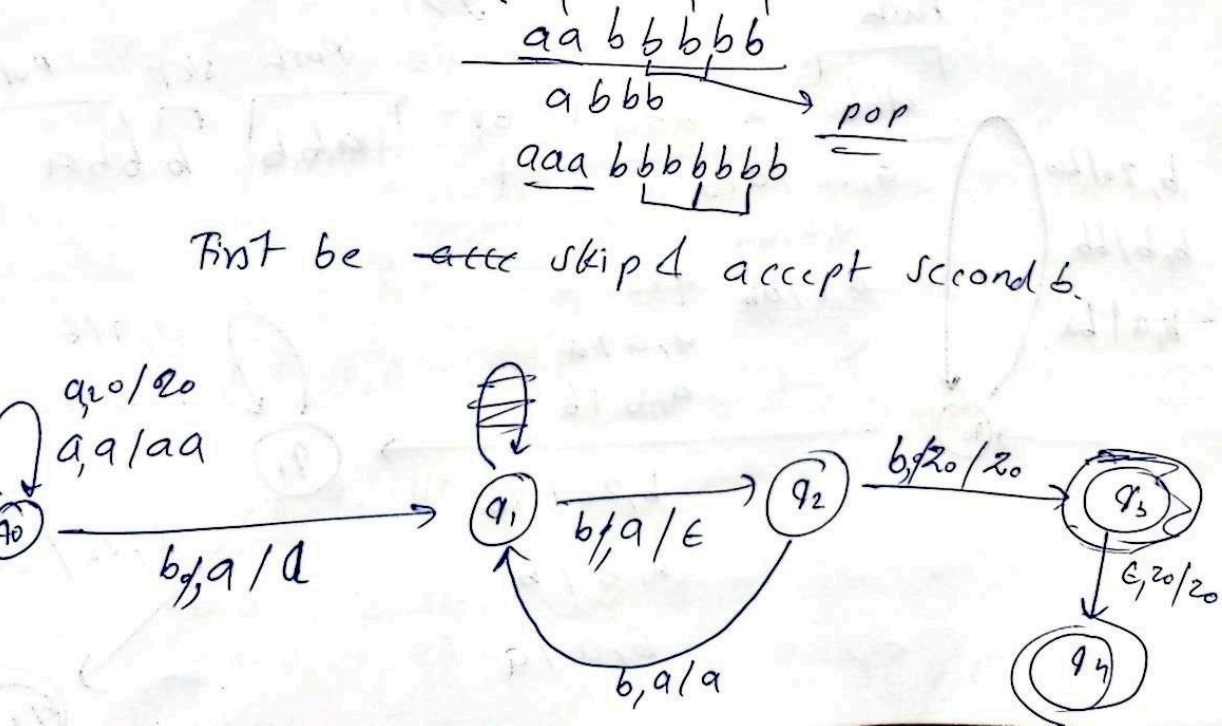
Push Skip

aa bbbbb

abbb pop

aaa bbbbbbb

First be ~~acce~~ skip & accept second b.

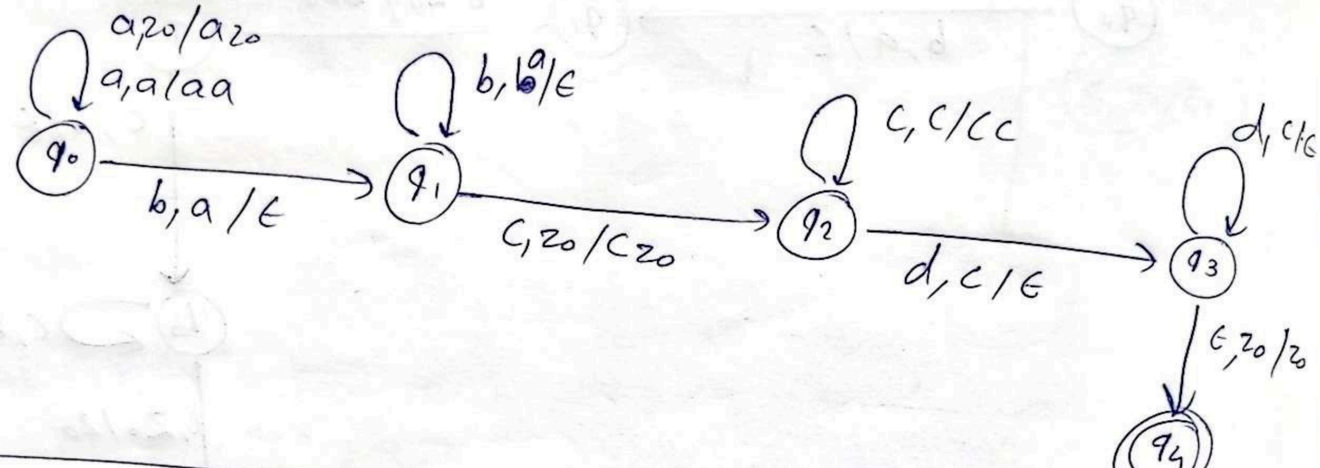


PDA For CFL.

$L = \{a^n b^n c^m d^m \mid n, m \geq 1\}$

~~P~~ aa bb cc dd

Push Pop Push Pop



PDA for odd Palindrome {a,b}

waw^R → ~~aa~~ bbabb

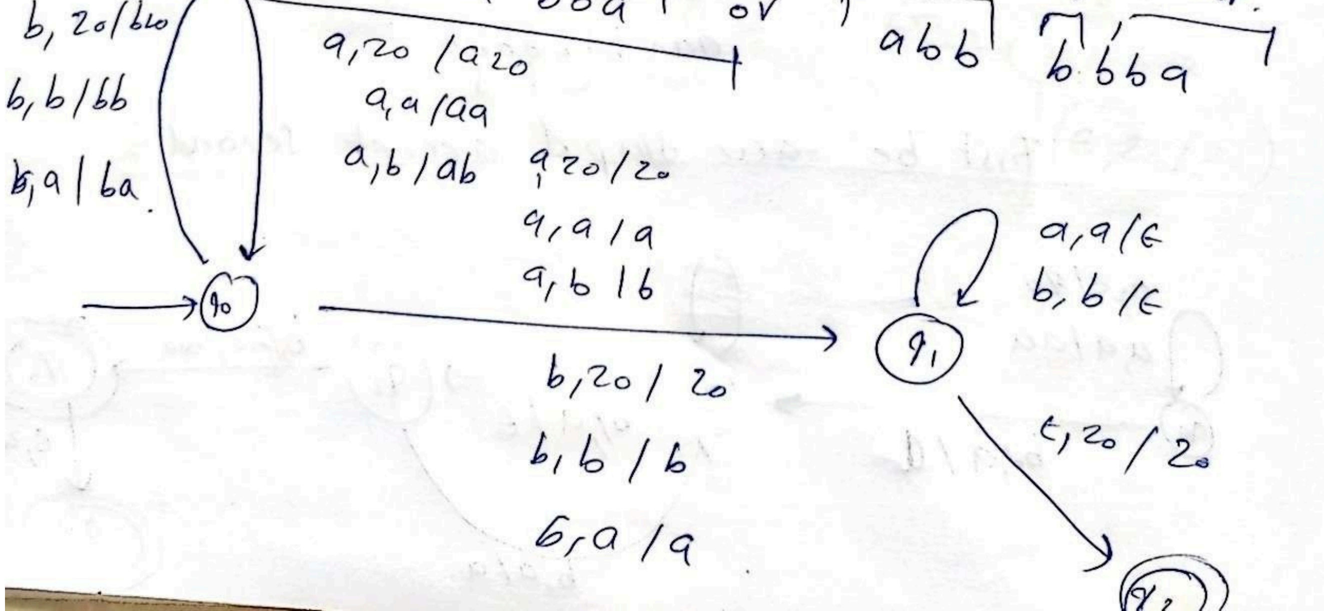
wbw^R → aabaa

(odd hai isiliye)

Push Pop

abb a bba or abb b bba

Push skip Pop.



* CFG to PDA

$$\begin{array}{ccc} & \text{CFL} & \\ \text{difined by} \swarrow & & \searrow \text{Accepted by} \\ \text{CFG} & \rightleftarrows & \text{PDA} \end{array}$$

<u>Rule 1</u> :- For each variable <u>A</u>

$\delta(q, \epsilon, A) = (q, \beta)$ (where $A \rightarrow \beta$ is a production of gramer)

Rule 2 :- For each terminal 'a'

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

Q.)

$S \rightarrow 0S1 \,/\, 00 \,/\, 11$

$\delta(q, \epsilon, S) = (q, 0S1), (q, 00), (q, 11)$ — ①

$\delta(q, 0, 0) = (q, \epsilon)$ —— ②

$\delta(q, 1, 1) = (q, \epsilon)$ — ③.

Now generate string, 0111

$\delta(q, 0111, S)$ using ①

$\delta(q, 0111, 0S1) \longrightarrow$ 0 read → pop hogaya —— ②

$\delta(q, 111, S1) \longrightarrow$ ①

$\delta(q, 111, \cancel{1}11) \longrightarrow$ ③

$\delta(q, 11, 11) \longrightarrow$ ③ $\longrightarrow \delta(q, \epsilon, \epsilon)$

$\delta(q, 1, 1)$ —— ③

Q.) $S \to 0BB$ $\to$ 010000

$B \to 0S \mid 1S \mid 0$

$$\delta(q, \epsilon, S) = (q, 0BB) \quad \text{— (1)}$$

$$\delta(q, \epsilon, B) = (q, 0S), (q, 1S), (q, 0) \quad \text{— (2)}$$

$$\delta(q, 0, 0) = (q, \epsilon) \quad \text{— (3)}$$

$$\delta(q, 1, 1) = (q, \epsilon) \quad \text{— (4)}$$

$\to \delta(q, 010000, S) \to$ ~~(q, 0~~

$\to \delta(q, \cancel{0}10000, \cancel{0}BB)$ $\to$ so use $\underline{3}$

input TOP

$\delta(q, 10000, BB) \to$ use 2 select 1 value.

$\delta(q, 10000, 1SB) \to$ use 4

$\delta(q, 0000, SB) \to$ (1)

$\delta(q, 0000, 0BBB) \to$ (3)

$\delta(q, 000, BBB) \to$ use (2) to get $B \to 0$

$\delta(q, \cancel{0}00, \cancel{0}BB) \to$ (3)

$\delta(q, 00, BB) \to$ (2)

$\delta(q, 00, 0B) \to$ (3)

$\delta(q, 0, B) \to$ (2)

$\delta(q, 0, 0) \to$ (3)

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

<u>Accept</u>

PDA to CFG

$M = \underbrace{\{p, q\}}_{\text{states}}, \underbrace{\{0, 1\}}_{\Sigma}, \underbrace{\{x, z\}}_{\text{stack}}, \underbrace{\delta}_{\text{transitions}}, \overset{\text{i.s}}{q}, z \rightarrow$ (initial stack)

① $\delta(q, 1, z) \rightarrow (q, x z) \rightarrow$ Push

② $\delta(q, 1, x) \rightarrow (q, x x)$ — skip

③ $\delta(q, \epsilon, x) \rightarrow (q, \epsilon) \rightarrow$ Pop

④ $\delta(q, 0, x) \rightarrow (p, x) \rightarrow$ Nop

⑤ $\delta(p, 1, x) \rightarrow (p, \epsilon) \rightarrow$ POP

⑥ $\delta(p, 0, z) \rightarrow (q, z) \rightarrow$ NOP

Solution :— $S \rightarrow [\overset{\text{initial}}{q}, z, q]$

$S \rightarrow [q, z, p]$

Now

1) $\delta(q, 1, z) \rightarrow (q, x z)$ Push.

$[q, z, q] \rightarrow 1 [q, x, q] [q, z, q]$

$[q, z, q] \rightarrow 1 [q, x, p] [p, z, q]$

$[q, z, p] \rightarrow 1 [q, x, q] [q, z, p]$

$[q, z, p] \rightarrow 1 [q, x, p] [p, z, p]$

② $\delta(q, 1, x) \rightarrow (q, xx)$ Push

$[q, x, q] \rightarrow 1\,[q, x, q]\,[q, x, q]$

$[q, x, q] \rightarrow 1\,[q, x, p]\,[p, x, q]$

$[q, x, p] \rightarrow 1\,[q, x, q]\,[q, x, p]$

$[q, x, p] \rightarrow 1\,[q, x, p]\,[p, x, p]$

③ $\delta(q, \epsilon, x) \rightarrow (q, \epsilon)$ POP.

$[q, x, q] \rightarrow \epsilon$ ✓

④ $\delta(q, 0, x) \rightarrow (p, x)$ no operation.

$[q, x, q] \rightarrow 0\,[p, x, q]$

$[q, x, p] \rightarrow 0\,[p, x, p]$

⑤ $\delta(p, 1, x) \rightarrow (p, \epsilon)$ — POP

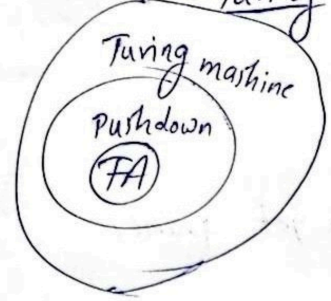
$[p, x, p] \rightarrow 1$

⑥ $\delta(p, 0, z) \rightarrow (q, z)$ NoP

$[p, z, q] \rightarrow 0\,[q, z, q]$

$[p, z, p] \rightarrow 0\,[q, z, p]$

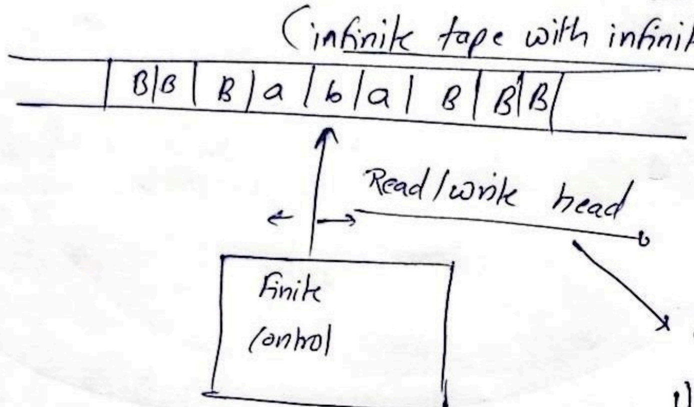
*Turing mashine *



$\longrightarrow$

$a^n b^n$	(PDA)
$a^n b^n c^n$	(TM)

$\rightarrow$ (infinite tape with infinite cells)



operations by head

1) Read/scan
2) write/update
3) Move/Head

---

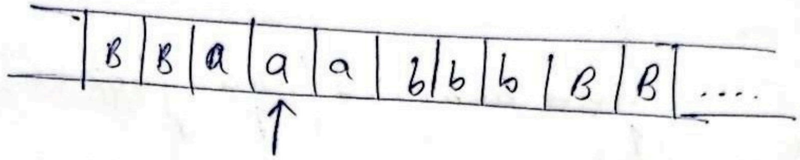
7 tupless

$$M = [Q, \Sigma, \Gamma, B, q_0, F, \delta]$$

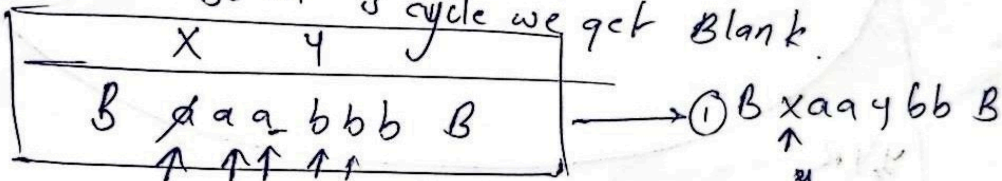
- Q → states
- Σ → input not containing B (Blank)
- Γ (Tau) → tape symbols including B
- B → Blank cell
- q_0 → initial state.
- F → Final
- δ → $(Q \times \Gamma \to Q \times \Gamma \times [L, R])$

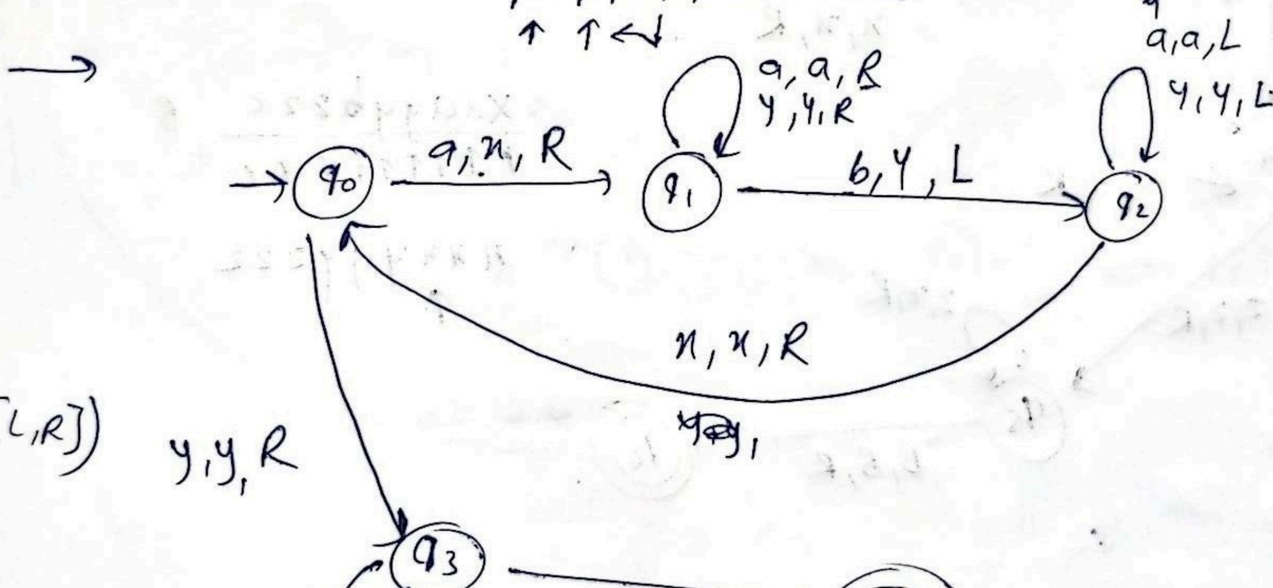
Q.) $a^n b^n \mid n \geq 1$

$\{ab, aabb, aaabbb, \ldots\}$



① Read First a then note it as X
then first b as Y
So in 3 cycle we get Blank.





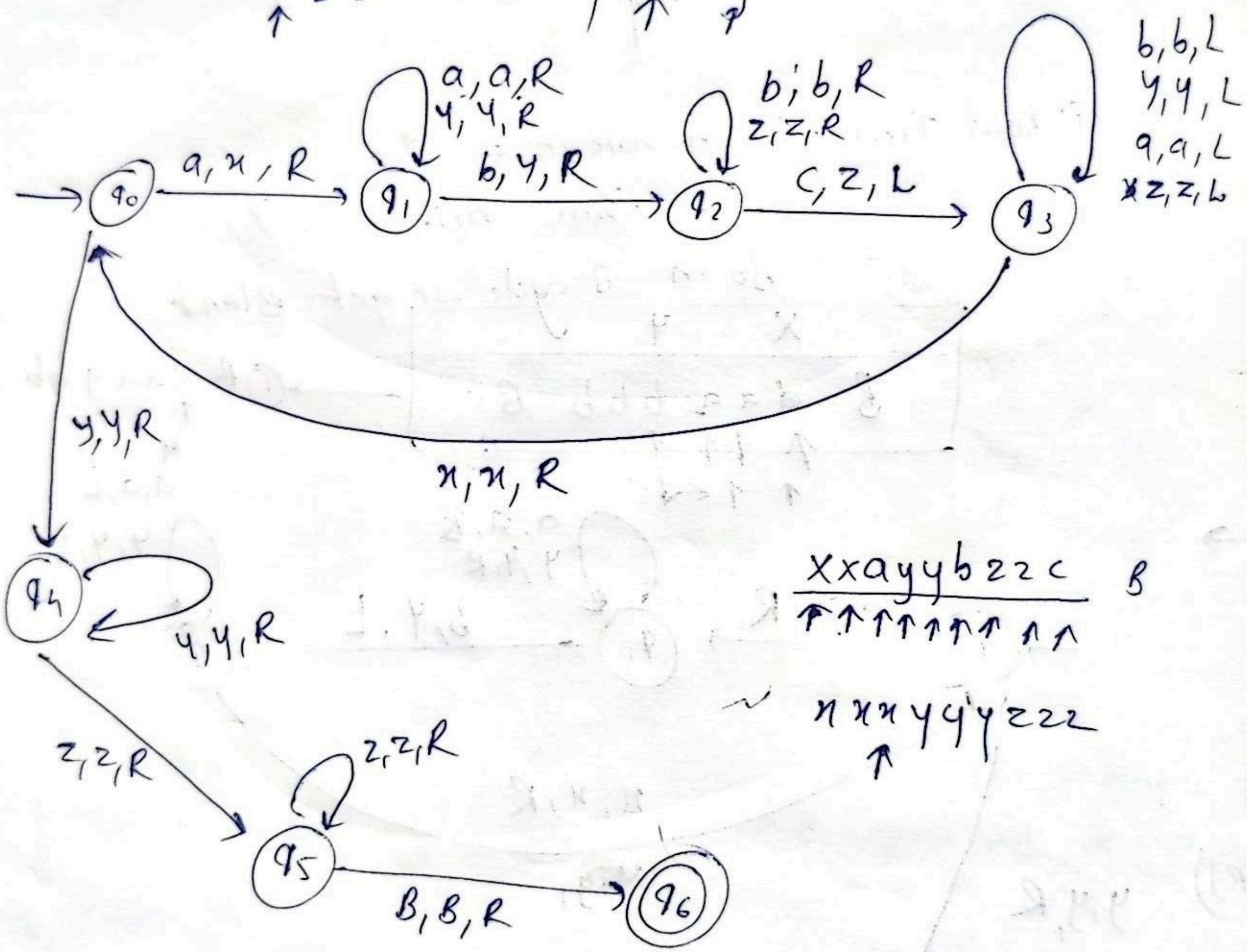
x[L,R])

* design turing mashine

for $\{a^n b^n c^n \mid n \geq 1\}$

$\{abc, aabbcc, aaa, bbb\,ccc, \ldots\}$

x
aaa bbbcccc / xaaybbzcc



xxayybzzc B

xxxyyyzzz

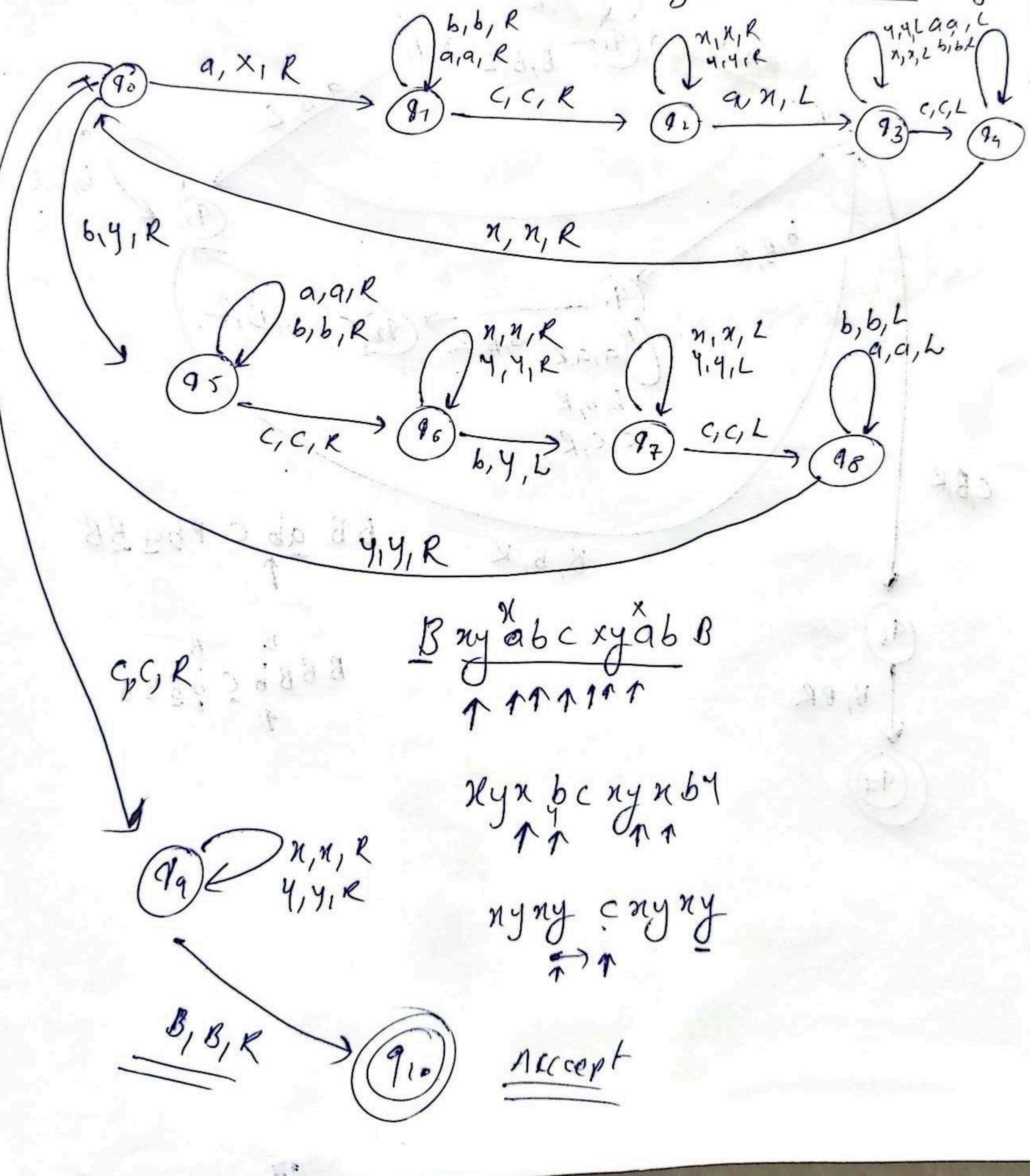
* desing

$WcW \mid W \in (a,b)^*$

ababcabab

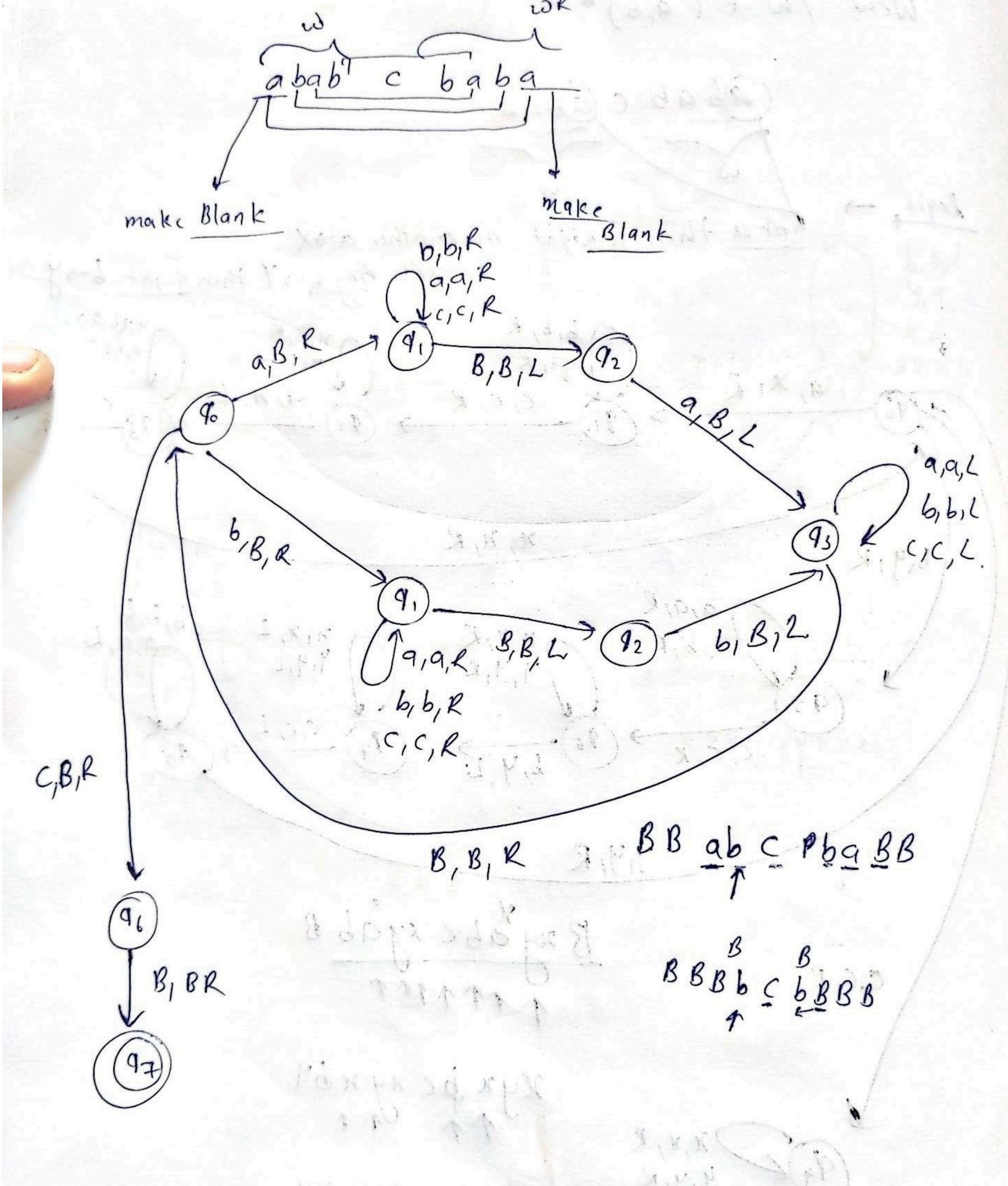
$\underbrace{abab}_{w}\ c\ \underbrace{abab}_{w}$

Logic → first a then go Right make other a → X
go left fing b → Y then Right b → y



Palindrome

$W c W^R \mid W \in (a,b)^*$



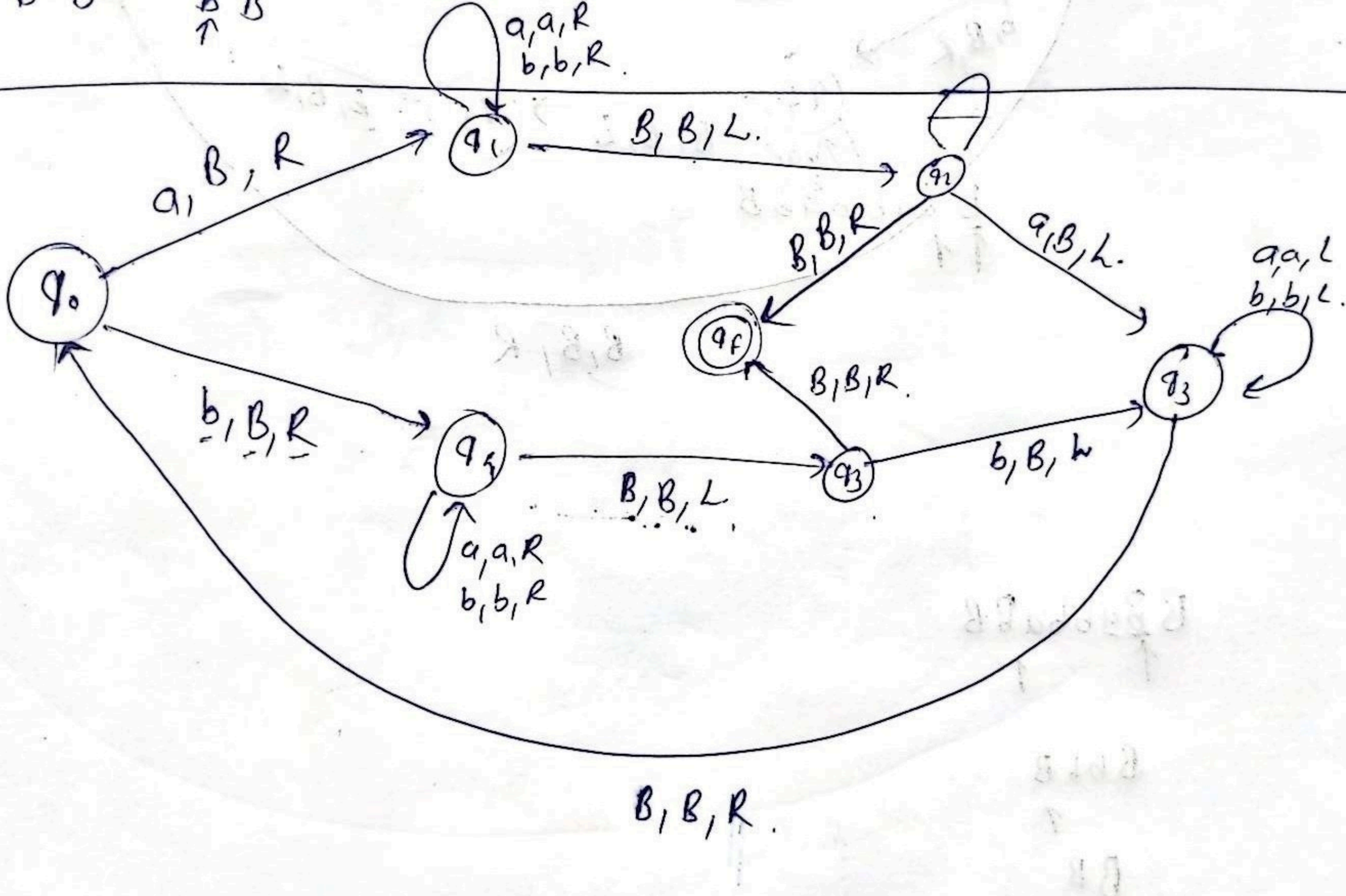
odd palindrome over $\{a,b\}$.

$\omega a \omega^R$
$\omega b \omega^R$ $\mid \omega \in (a,b)^*$

B B a B B

a b a b a
B B B B

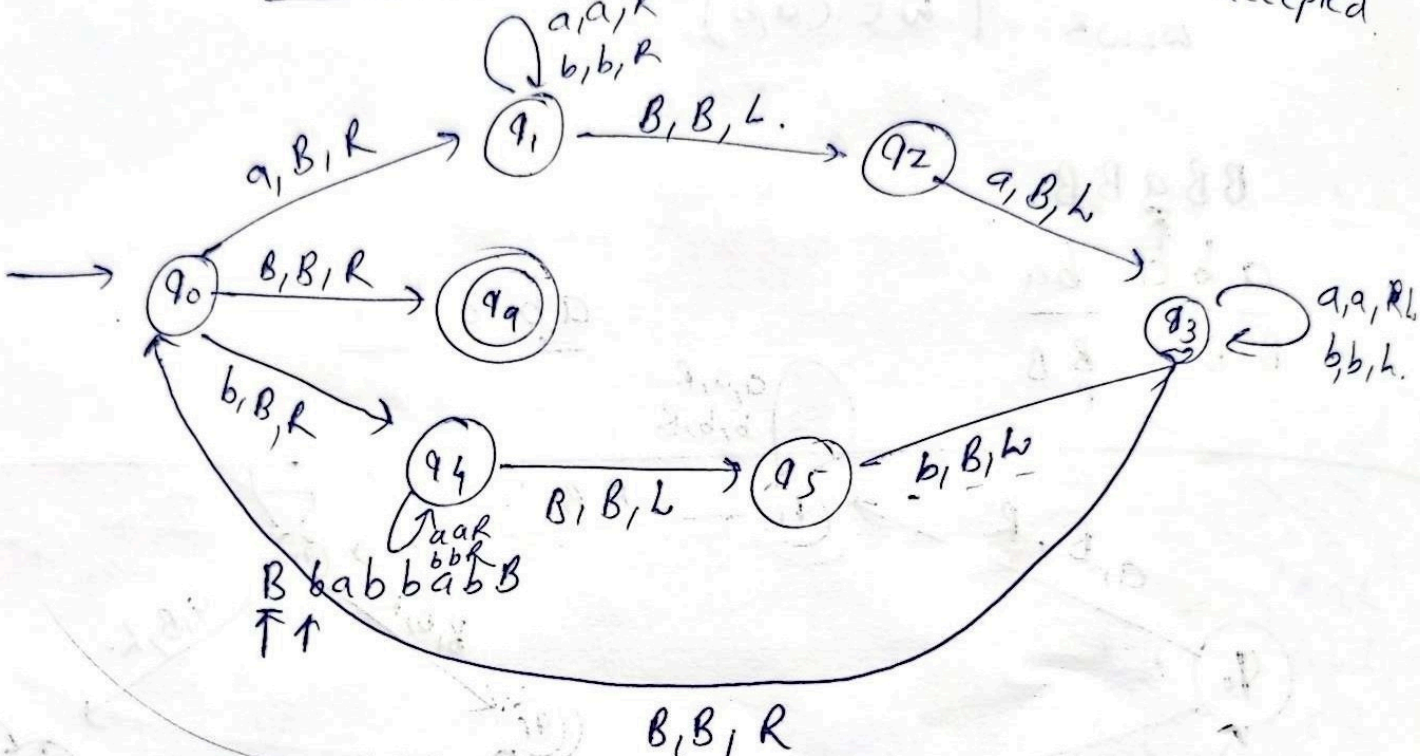
ab b ba



even palindrome over $(a,b)^*$ = ϵ so Blank also accepted

$\omega \omega^R$

a b a b b a b a



B babbabB

BBabbaBB

BbbB

BB

Palindrome — both odd
even combined turing

ww^R → even

waw^R → odd
wbw^R →

Combination to draw

(even + odd)