Module:4

1) $L(G) = a^i b^i c^j d^j \; ; (i, j \geq 0)$

2) $L(G) = e^5 a^i b^j c^j d^i \; ; (i, j > 0)$

3) $L(G) = \omega \# \omega^R$, ω is binary

4) $L(G) = \omega$: $n(a)$ of $\omega = 2 \cdot n(b)$ of ω

5) Write $L(G)$ to generate:

(a) atleast one a over $\{a, b\}$

(b) exactly one a over $\{a, b\}$

(c) more than one a over $\{a, b\}$

Answers

1) $S \longrightarrow AB$

$A \longrightarrow aAb \mid \epsilon$

$B \longrightarrow cBd \mid \epsilon$

2) $S \longrightarrow e^5 AB$

$A \longrightarrow$

$B \longrightarrow$

Not complete

SIMPLIFICATION/REDUCTION OF CFG

In CFG, all production symbols are not needed for the derivation of strings besides this they may be null and unit productions. Eleminations of these productions & symbols are called simplification of CFG.

There are 3 process involved in simplification of CFG.

i) Reduction of productions in CFG

ii) Removal of unit productions

iii) Removal of null productions

i) Reduction of production

Reduction of CFG is reduced in 2 phases.

Phase I: Derivation of equivalent G' from CFG G such that each variable derives some terminal strings.

Procedure:

- Include all symbol w_1, that derives some terminals and initialize $i=1$
- Include symbol w_{i+1} that derives w_i.
- Increment i and repeat step 2 until $w_{i+1} = w_i$
- Include all production rule that have w_i in it.

Phase II: Derivation of equivalent G'' from CFG G' such that each symbol appears in sentential form.

Procedure:

- Include the start symbol in Y_1 and initialize $i=1$.
- Include all symbols Y_{i+1} that can be derive from Y_i and include all ~~production~~ production rules that have been applied
- Increment i and repeat step 2 ~~unit~~ until $Y_{i+1} = Y_i$

ii) Removal of unit productions

Any production of the form $A \rightarrow B$ where A, B belongs to a non-terminal is called a unit production

Procedure:

- To remove $A \rightarrow B$ add production $A \rightarrow x$ to grammer G wherever $B \rightarrow x$ occurs. $x \in t$ and Null.
- ~~They read $A \rightarrow B$~~
- Delete $A \rightarrow B$ from G. Repeat from step 1 until all productions are removed.

NOTE:

If there are unreachable symbols follow the procedure phase 2 in production reduction to remove the same.

iii) Remove null production

In the CFG, non terminal, A is a nullable variable if there is a production $A \rightarrow \epsilon$ or there is a derivation that starts at A leads to ϵ is called null production.

Procedure:

- To remove $A \to \epsilon$ look for all productions ~~the~~ where right side contains A.
- Replace each occurance of A in each of these production with ϵ.
- Add resultant production to the grammar G.

i) Reduction of production

(1) $S \to AC \mid B$

$A \to a$

$C \to c \mid BC$

$E \to aA \mid e$

$G = V\,t\,P\,S$

$= (\{S, A, C, B, E\}\ \{a\ c\ e\}\ P\ \{S\})$

$t = \{a, c, e\} \longrightarrow$ terminal

$\longrightarrow$ terminalsne directly yield cheyuonaron terminas

$W_1 = \{A, C, E\}$ $A = ?,\ C = ?,\ E = ?$

$AC = ?,\ AE = ?,\ AS = ?$

$W_2 = \{A, C, E, S\}$ $cS = ?,\ ES = ?$

$W_3 = \{A, C, E, S\} \longrightarrow$

$AS = ?\ \ CS = ?$

$G' = (\{A, C, E, S\}, \{a, c, e\}, P', \{S\})$

Ignore B, because G' does not contain B.

$P' =$

$S \to AC\ \cancel{B}$

$A \to a$

$C \to c$

$E \to aA \mid e$

~~Procedure~~

Phase 2:

$Y_1 = \{S\}$ → Yield of S

$Y_2 = \{S, A, C\}$ → Yield of A

$Y_3 = \{S, A, C, a, c\}$ → Yield of C

$Y_4 = \{\underbrace{S, A, C,}_{\text{Non terminal}} a, c\}$ — terminal

P''

$G'' = (\{S, A, C\}, \{a, c\}, P'', \{S\})$

Ingore $E \to aA|c$

$S \to AC$

$A \to a$

$C \to c$

01/10/19

ii) Removal of ϵ Production

① $S \to aS | bA$

$A \to aA | \epsilon$

$S \to aS | bA$

$A \to aA | \epsilon$ Remove $A \to \epsilon$, Write aA and replace A by ϵ & remove ϵ

$A \to aA | a\epsilon$

$A \to aA | a$

$S \to aS | bA | b$; Replace A by ϵ & again write bA & replace A by ϵ & remove ϵ

$A \to aA | a$

② $S \to AaB | aaB$

$A \to \epsilon$

$B \to bbA | \epsilon$

Put $B = \epsilon$ to remove $B \to \epsilon$

$S \to AaB | aaB | Aa\epsilon | aa\epsilon$

$S \to AaB | aaB | Aa | aa$

$A \to \epsilon$

$B \to bbA$

Put A=E to remove A ⟶ E

B ⟶ bbA | bbE

B ⟶ bbA | bb

S ⟶ AaB | aaB | Aa | aa | E aB | E a

S ⟶ ~~AaB~~ | aaB | ~~Aa~~ | aa | aB | a

S ⟶ aaB | aa | aB | a

B ⟶ bb

iii) <u>Removal of unit production</u>

S ⟶ AB

A ⟶ a

Non-T B ⟶ C | b (C: Non-ter) ✗

Non-T C ⟶ D (Non T) ✗

Non D ⟶ E (Non T) ✗

Non E ⟶ a terminal

They must be change to Non-terminal ⟶ terminal

Start from last

~~E ⟶ a~~

~~D ⟶ a~~

~~C ⟶ a~~

S contains only A & B & A & B doesn't contains C, D, E.
So we can remove E, D, C

B ⟶ a | b

A ⟶ a

S ⟶ AB

∴ S ⟶ AB

A ⟶ a

B ⟶ a | b

Q2. $S \rightarrow AB \mid bX$
$A \rightarrow BAd \mid bSX \mid a$
$B \rightarrow aSB \mid bBX$
$X \rightarrow SBD \mid aBX \mid ad$

We have to check whether Grammer S will be terminate or not.

$S \rightarrow \underbrace{AB}_{\text{Nonterminal}} \mid \overset{\text{terminal}}{b}X$ ⟶ Non terminal

Then check X will be terminate or not.
So X contains ad at last. So X will be terminate.

$S \rightarrow AB \mid \underbrace{bad}_{\text{terminal}}$.

So 'S' will be terminate.

$X \rightarrow ad$, so X will be terminate.
$A \rightarrow a$, so A will be terminate.
B will not end. B is recursive, Because we can recurssively substitute B $[B \rightarrow aS\underline{B} \mid b\underline{B}X]$

* Remove B from all

$S \rightarrow \cancel{AB} \mid bX$
$A \rightarrow \cancel{BAd} \mid bSX \mid a$
$\cancel{B \rightarrow aSB \mid bBX}$.
$X \rightarrow \cancel{SBD} \mid \cancel{aBX} \mid ad$

* $S \rightarrow bX$.
$A \rightarrow bSX \mid a$
$X \rightarrow ad$

S doesnot contains A, so we can remove A

* $S \rightarrow bX$
$X \rightarrow ad$

4/10/19

Chomsky Normal Form (CNF)

$$\boxed{\begin{array}{l} NT \rightarrow NT\,NT \\ NT \rightarrow t \end{array}}$$

? $S \rightarrow bA \mid aB$
$A \rightarrow bAA \mid aS \mid a$ ✓
$B \rightarrow aBB \mid bSS \mid a$ ✓

We ~~ca~~ will change others:

$S \rightarrow bA$ not possible $S \rightarrow aB$
$Z \rightarrow b$ ~~Y→a~~ $A \rightarrow a$
$S \rightarrow ZA$ $S \rightarrow YB$

$S \rightarrow ZA \mid AB$

$A \rightarrow bAA$ $A \rightarrow aS$
$A \rightarrow ZAA$ $A \rightarrow AS$
$X \rightarrow AA$ $A \rightarrow a$
$A \rightarrow ZX$

$A \rightarrow ZX \mid AS \mid a$

$B \rightarrow aBB$ $B \rightarrow bSS$
$B \rightarrow ABB$ $B \rightarrow ZSS$
$W \rightarrow BB$ ~~B~~ $U \rightarrow SS$
$B \rightarrow AW$ $B \rightarrow ZU$

$B \rightarrow AW \mid ZU \mid a$

? $S \to abSb \mid a \mid aAb$

$A \to bS \mid aAAb$

$S \to abSb$ $\quad$ $S \to a$ $\quad$ $S \to aAb$

$S \to SAb$

$S \to SbSb$

$S \to SAZ$

$Z \to b$

$P \to SA$

$S \to SZSZ$

$S \to PZ$

$X \to SZ$

$S \to XX$

$S \to XX \mid a \mid PZ$

$A \to bS$ $\quad$ $A \to aAAb$

$A \to ZS$ $\quad$ $A \to SAAZ$

$A \to PAZ$

$Q \to AZ$

$A \to PQ$

$A \to ZS \mid PQ$

Greibach Normal Form (GNF)

√imp

$$\boxed{\begin{array}{l} NT \to t\alpha \\ \alpha \in \epsilon \\ \alpha \in NT_1 \ldots NT_\alpha \end{array}}$$

Reduction to GNF

Lemma ① [Substitutation Rule]

$A \to B\gamma$

$B \to \beta_1 \mid \beta_2 \mid \beta_3 \ldots \ldots \mid \beta_n$

replaced by, $A \to \beta_1\gamma \mid \beta_2\gamma \mid \beta_3\gamma \ldots \ldots \mid \beta_n\gamma$

Scanned by CamScanner

* Deleting some variable in 'A' production, β appearing as the 1st symbol on the RHS of some 'A' productions provided no β production has the production β as the 1st symbol

Lemma ② [Left Recurssion]

$$A \rightarrow A\alpha_1 | A\alpha_2 \cdots\cdots | \beta_1 | \beta_2 \cdots\cdots | \beta_n$$

replaced as,

$$A \rightarrow \beta_1 | \beta_2 | \beta_3 \cdots\cdots | \beta_n$$

$$A \rightarrow \beta_1 Z_1 | \beta_2 Z_1 | \beta_3 Z_1 | \cdots\cdots | \beta_n Z_1$$

$$Z_1 \rightarrow \alpha_1 | \alpha_2 \cdots\cdots | \alpha_n$$

$$Z_1 \rightarrow \alpha_1 Z_1 | \alpha_2 Z_1 | \cdots\cdots | \alpha_n Z_1$$

Step1: * Eliminate useless (symbols) production, null production and 'E' production and then construct grammar 'G' in CNF. Rename the variable of production of G as $A_1, A_2, A_3 \cdots\cdots$

Step2: $A_i \rightarrow A_j\alpha$; where $i=j, i<j, i>j$

if $i>j$, apply Lemma ①

if $i<j$, no change

if $i=j$, apply Lemma ②

Step3: Starting from the higher no. of non terminals replace the RHS of the production with its alternative unit. So that every production is of the form $A \rightarrow a\alpha$

Scanned by CamScanner

① Reduce the following G to GNF

$S \rightarrow AA \mid a$

$A \rightarrow SS \mid b$

Step 1: There is no useless symbols, null production & e production

~~Step 3~~ It is in CNF

Rename the variable of production

1st NT, 2nd NT [if there is 3rd NT, then give as A_3]

$A_1 \rightarrow A_2 A_2 \mid a$ ——— ①

$A_2 \rightarrow A_1 A_1 \mid b$ ——— ②

Step 2: $A_i \rightarrow A_j \alpha$; check the i values & j values

$A_{1(i)} \rightarrow A_{2(j)} A_2 \mid a$

$i < j$; no change

$A_2 \rightarrow A_1 A_1 \mid b$

$i > j$; apply Lemma ①

9/10/19

$\underset{A}{A_2} \longrightarrow \overset{B}{A_1} \overset{\gamma}{A_1} \mid b$ ——— ② (terminal (no change))

$\underset{B}{A_1} \longrightarrow \underbrace{A_2 A_2}_{B_1} \mid a^{B_2}$ ——— ①

Replaced by, Sub: A, B, γ, B_1 in eq ②

$A \rightarrow B_1\gamma \mid B_2\gamma \mid \ldots$ ($B_n\gamma$ + terminal)

$A_2 \rightarrow A_2 A_2 A_1 \mid a A_1 \mid b$ ——— ③ $i = j$, apply L②

$A_1 \longrightarrow A_2 A_2 \mid a$ ——— ① $i < j$, NC

In ③ ⇒ Starting with A_2 are α, others are β

③ → $A_2 \rightarrow \overset{A}{A_2} \overbrace{A_2 A_1}^{\alpha_1} \mid \overbrace{a A_1}^{B_1} \mid \overbrace{b}^{B_2}$

Substitute A, α_1, B_1, B_2 in eq ③

~~$A_2 \rightarrow A\alpha \mid B_1 \mid B_2$~~

$A \to \beta_1 | \beta_2 | \cdots | \beta_n$

$A_2 \to a A_1 | b$

$A_2 \to a A_1 Z_2 | b Z_2$; $A \to \beta_1 z_1 | \beta_2 z_1 \cdots | \beta_n z_1$

$z_1 \to \alpha_1 | \alpha_2 \cdots \alpha_n$

$Z_2 \to A_2 A_1$ (Z_2 = Any letter)

$Z_2 \to A_2 A_1 Z_2$

$z_1 \to \alpha_1 z_1 | \alpha_2 z_1 \cdots \alpha_n z_n$

10/10/19 combine the eqns of A_2 & Z_2

$A_2 \to a A_1 | b | a A_1 Z_2 | b Z_2$ —— (4)

$Z_2 \to \underline{A_2} A_1 | \underline{A_2} A_1 Z_2$ —— (5)

$A_1 \to \underline{A_2} A_2 | a$ —— (1)

Because (A_2 is NT)

A_2 is already in GNF form So NC

Replace A_2 in Z_2 → ($NT \to t\alpha$ to convert to GNF form $\alpha \in NT$, $\alpha \in \epsilon$)

~~$A_2 \to a A_1$~~

$Z_2 \to a A_1 A_1 | b A_1 | a A_1 Z_2 A_1 | b Z_2 A_1 |$
$a A_1 A_1 Z_2 | b A_1 Z_2 | a A_1 Z_2 A_1 Z_2 | b Z_2 A_1 Z_2$ —— (6)

Replace A_2 in A_1 to convert to $NT \to t\alpha$ form

$A_1 \to a A_1 A_2 | b A_2 | a A_1 Z_2 A_2 | b Z_2 A_2 | a$ —— (7)

So eqn (4), (6) & (7) are in $NT \to t\alpha$ form

(2) $S \to AB$

$A \to aA | bB | b$

$B \to b$

A & B are in CNF form so we only change S into CNF form.

Replace A in S.

$S \to aAB | bBB | bB$.

Scanned by CamScanner

③ $S \to AB$
$A \to BS \mid b$
$B \to SA \mid a$

step1:

$A_1 \to A_2 A_3$ —— ①

$A_2 \to A_3 A_1 \mid b$ —— ②

$A_3 \to A_1 A_2 \mid a$ —— ③

Step2:

① ⇒ $A_1 \to A_2 A_3$; $i<j$, No change

② ⇒ $A_2 \to A_3 A_1 \mid b$; $i<j$, No change

③ ⇒ $\underset{A}{A_3} \to \underset{\beta}{A_1}\underset{\gamma}{A_2} \mid a$; $i>j$, apply L① → ③

$\underset{\beta}{A_1} \to \underbrace{A_2 A_3}_{\beta_1}$ ——→ ①

Sub A, β, γ eqn ③

$A \to \beta_1\gamma \mid \beta_2\gamma \mid \dots \beta_n\gamma \mid$ terminal

$\underset{A}{A_3} \to \underset{\beta}{A_2}\underset{\gamma}{A_3} \underset{\alpha}{A} \mid a$ —— ④

$i>j$; apply L①

$\underset{\beta}{A_2} \to \underbrace{A_3 A_1}_{\beta_1} \mid \underset{\beta_2}{b}$ —— ②

Sub $A, \beta\, \gamma$ eqn ④

$A \to \beta_1\gamma \mid \beta_2\gamma \mid \dots \beta_n\gamma \mid$ terminal

$\underset{A}{A_3} \to A_3 \overbrace{A_1 A_3 A_2}^{\alpha_1} \mid \overbrace{b\, \cancel{A_3 A_2}\, A_3 A_2}^{\beta_1} \mid \overbrace{a}^{\beta_2}$ —— ⑤

$i=j$; apply L②

Sub $A, \alpha_1, \beta_1, \beta_2$ in eqn ⑤

$A \to \beta_1 \mid \beta_2 \mid \dots \beta_n$

$A_3 \to b A_3 A_2 \mid a$

$A_3 \rightarrow bA_3A_2Z_3 \mid aZ_3$

$Z_3 \rightarrow A_1A_3A_2$

$Z_3 \rightarrow A_1A_3A_2Z_3$

Combining :-

$A_1 \rightarrow A_2A_3$

$A_2 \rightarrow A_3A_1 \mid b$

$A_3 \rightarrow bA_3A_2 \mid a \mid bA_3A_2Z_3 \mid aZ_3$ ——— (6)

$Z_3 \rightarrow A_1A_3A_2 \mid A_1A_3A_2Z_3$ ——— (7)

Converting to GNF :-

A_3 in GNF form.

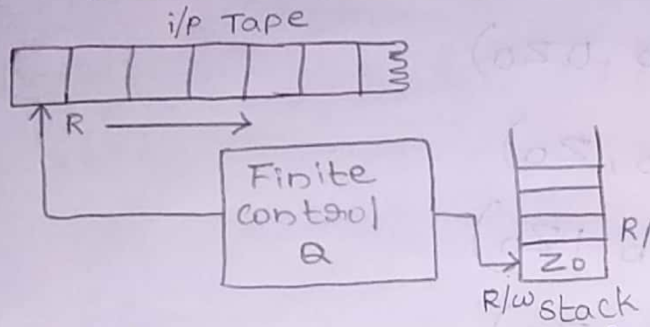
$\therefore A_2 \rightarrow bA_3A_2A_1 \mid aA_1 \mid bA_3A_2Z_3A_1 \mid aZ_3A_1 \mid b$

Now A_2 in GNF form.

$A_1 \rightarrow bA_3A_2A_1A_3 \mid aA_1A_3 \mid bA_3A_2Z_3A_1A_3 \mid aZ_3A_1A_3 \mid bA_3$

$Z_3 \rightarrow bA_3A_2A_1A_3A_3A_2 \mid aA_1A_3A_3A_2 \mid bA_3A_2Z_3A_1A_3A_3A_2$
$\mid aZ_3A_1A_3A_3A_2 \mid bA_3A_3A_2 \mid bA_3A_2A_1A_3A_3A_2Z_3$
$\mid aA_1A_3A_3A_2A_3A_2A_3 \mid bA_3A_2Z_3A_1A_3A_3A_2A_3A_2A_3$
$\mid aZ_3A_1A_3A_3A_2A_3 \mid bA_3A_3A_2A_3$

Push Down Automata (PDA)



7 tuple

$$(Q, q_0, F, \delta, \Sigma, \underset{\text{top of stack}}{Z_0}, \underset{\text{stack symbol}}{\Gamma})$$

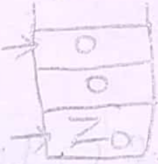
$\delta : Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma$

PDA = FA + stack.

PDA is the mathematical model for CFG

→ Design a PDA for recognizing $L = \{n_a(\omega) = n_b(\omega)\}$ over $\{a, b\}$

b	a	b	a	b	a	
a	a	b	a	b	a	



$(q_0, a, Z_0) \rightarrow (q_0, 0Z_0)$

$(q_0, b, Z_0) \rightarrow (q_0, 1Z_0)$

$(q_0, a, 0) \rightarrow (q_0, 00)\ (q_0, \Lambda)(q_0, \epsilon, Z_0)$

$(q_0, b, 0) \rightarrow (q_0, \Lambda)$

$(q_0, a, 1) \rightarrow (q_0, \Lambda)$

$(q_0, b, 1) \rightarrow (q_0, 11)$

$(q_0, \epsilon, Z_0) \rightarrow (q_f, Z_0)$

$(q_0, abbaab, Z_0)$

$\vdash (q_0, bbaab, 0Z_0)$

$\vdash (q_0, baab, Z_0)$

$\vdash (q_0, aab, 1Z_0)$

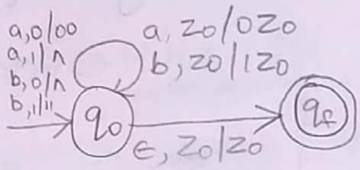
$\vdash (q_0, ab, Z_0)$

$\vdash (q_0, b, 0Z_0)$

$\vdash (q_0, \epsilon, Z_0)$

$\vdash (q_f, Z_0)$

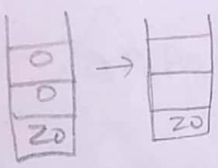
Non acceptance :-



$M = (\{q_0, q_f\}, q_0, \{q_f\}, \delta, \{a, b\}, \{Z_0\}, \{Z_0, 0, 1\})$

→ Design a PDA for $L = \{a^n b^n / n \geq 0\}$ over $\{a, b\}$

→ Design a PDA for $L = \{a^n b^n / n > 0\}$, over $\{a, b\}$



$(q_0, a, Z_0) \rightarrow (q_0, 0Z_0)$

$(q_0, a, 0) \rightarrow (q_0, 00)$

$(q_0, b, 0) \rightarrow (q_1, \Lambda)$

$(q_1, b, 0) \rightarrow (q_1, \Lambda)$

Scanned by CamScanner

$(q_1, \epsilon, Z_0) \longrightarrow (q_f, Z_0)$

→ Design a PDA for $L = \{ wcw^R / w \in \{a, b\} \}$

abbabcbabba

$(q_0, a, Z_0) \longrightarrow (q_0, aZ_0)$

$(q_0, b, Z_0) \longrightarrow (q_0, bZ_0)$

$(q_0, a, a) \longrightarrow (q_0, aa)$

$(q_0, b, a) \longrightarrow (q_0, ba)$

$(q_0, a, b) \longrightarrow (q_0, ab)$

$(q_0, b, b) \longrightarrow (q_0, bb)$

$(q_0, c, a) \longrightarrow (q_1, \text{NOP})$

$(q_0, c, b) \longrightarrow (q_1, \text{NOP})$

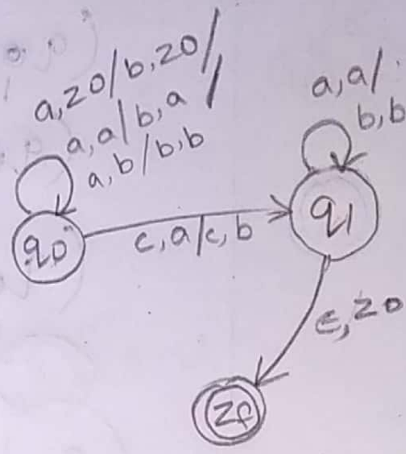
$(q_1, a, a) \longrightarrow (q_1, \Lambda)$

$\cancel{(q_1, b, a) \longrightarrow (q_1, \Lambda)}$

$\cancel{(q_1, a, b) \longrightarrow (q_1, \Lambda)}$

$(q_1, b, b) \longrightarrow (q_1, \Lambda)$

$(q_1, \epsilon, Z_0) \longrightarrow (q_f, Z_0)$



→ Design a PDA for $L = \{ ww^R / w \in \{a, b\} \}$.

abbaabba

$(q_0, a, Z_0) \longrightarrow (q_0, aZ_0)$

$(q_0, b, Z_0) \longrightarrow (q_0, bZ_0)$

$(q_0, a, b) \longrightarrow (q_0, ab)$

$(q_0, b, a) \longrightarrow (q_0, ba)$

$(q_0, a, a) \longrightarrow (q_0, aa)$

Non deterministic

Non determ

$(q_0, a, a) \longrightarrow (q_1, \wedge)$

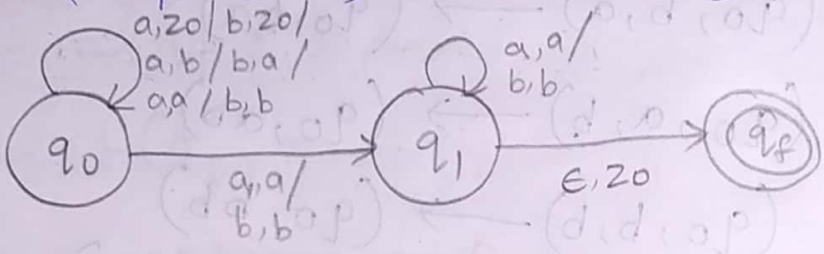
$(q_0, b, b) \longrightarrow (q_0, bb)$

$(q_0, bb) \longrightarrow (q_1, \wedge)$

$(q_1, aa) \longrightarrow (q_1, \wedge)$

$(q_1, bb) \longrightarrow (q_1, \wedge)$

$(q_1, \epsilon, Z_0) \longrightarrow (q_f, Z_0)$



$(q_0, abbbba, Z_0)$

$\vdash (q_0, bbbba, aZ_0)$

$\vdash (q_0, bbba, ba)$

$\vdash (q_0, bba, bba) \cup (q_1, bba, aZ_0)$

$\vdash (q_0, ba, bbba) \cup (q_1, ba, baZ_0)$

$\vdash (q_1, a, aZ_0)$

$\vdash (q_1, \epsilon, Z_0)$

$\vdash (q_f, Z_0)$

→ Design DPDA for i) $L = \{a^n b^{2n} / n \geq 0\}$

ii) $L = \{a^n b^{n+1} / n \geq 0\}$

iii) $L = \{a^{2n} b^n / n \geq 0\}$

i) $(q_0, a, Z_0) \longrightarrow (q_0, aZ_0)$

$(q_0, a, a) \longrightarrow (q_0, aa)$

$(q_0, b, a) \longrightarrow (q_1, NOP)$

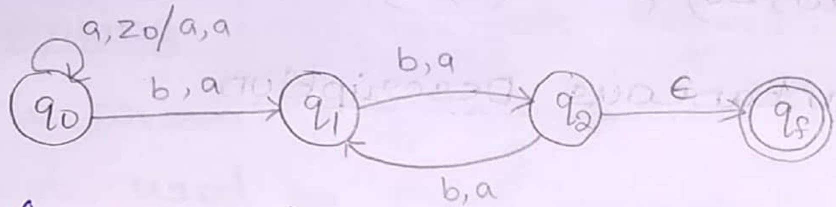
$(q_1, b, a) \rightarrow (q_2, \Lambda)$

~~$(q_2, b, b) \rightarrow (q_f, NOP)$~~

~~$(q_0, \epsilon$~~

$(q_2, b, a) \rightarrow (q_1, NOP)$

$(q_2, \epsilon, Z_0) \rightarrow (q_f, Z_0)$



ii) $(q_0, a, Z_0) \rightarrow (q_0, aZ_0)$

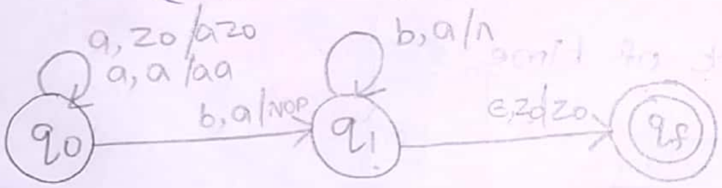
$(q_0, b, Z_0) \rightarrow (q_1, NOP)$

$(q_0, a, a) \rightarrow (q_0, aa)$

$(q_0, b, a) \rightarrow (q_1, NOP)$

$(q_1, b, a) \rightarrow (q_1, \Lambda)$

$(q_1, \epsilon, Z_0) \rightarrow (q_f, Z_0)$



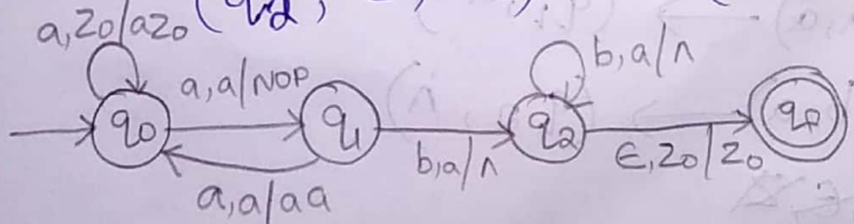
iii) $(q_0, a, Z_0) \rightarrow (q_0, aZ_0)$

$(q_0, a, a) \rightarrow (q_1, NOP)$

$(q_1, a, a) \rightarrow (q_0, aa)$

$(q_1, b, a) \rightarrow (q_2, \Lambda)$

$(q_2, \epsilon, Z_0) \rightarrow (q_f, Z_0)$



Scanned by CamScanner

Acceptance of PDA

1. Acceptance by final state.

$$(q_0, w, Z_0) \overset{*}{\vdash} (q_f, Z_0)$$

after multiple steps $\; \overset{n-1}{\vdash} (q', \epsilon, Z_0)$

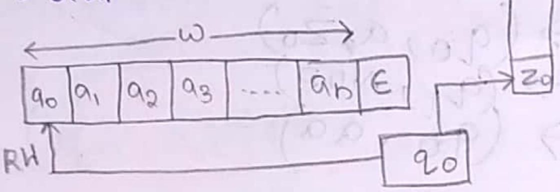
context free grammar
finite grammar
regular expression

2. Acceptance by empty stack

$$(q_0, w, Z_0) \overset{*}{\vdash} (p, \epsilon)$$

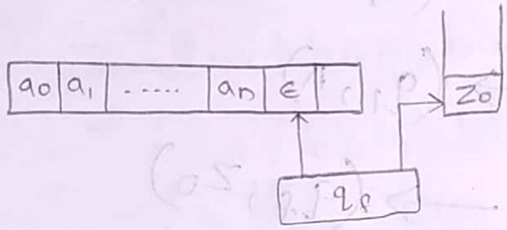
Instantaneous Description

Initial

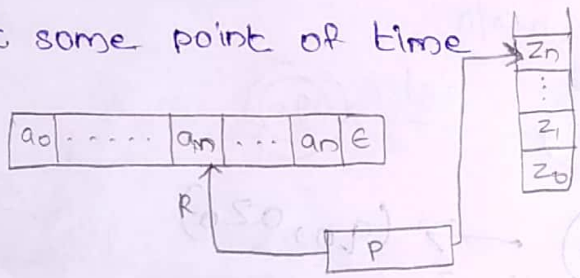


RH - read head

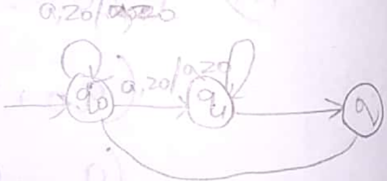
Final (by FS)



at some point of time



$\rightarrow L = a^{n+1} b^n \mid n \geq 0$



$(q_0, a, Z_0) \longrightarrow (q_0, aZ_0)$

$(q_0, a, a) \longrightarrow (q_1, \text{NOP})$

$(q_1, a, a) \longrightarrow (q_1, aa)$

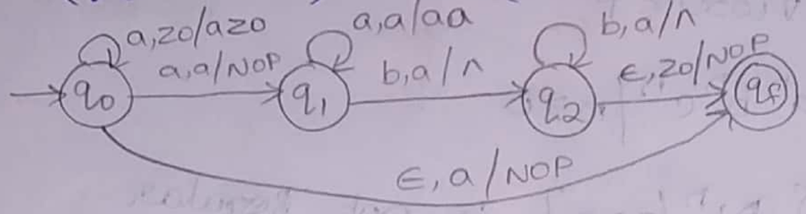
$(q_1, b, a) \longrightarrow (q_2, \wedge)$

~~(q_2, ϵ, Z)~~

$(q_0, \epsilon, a) \rightarrow (q_f, NOP)$

$(q_2, b, a) \rightarrow (q_2, \wedge)$

$(q_2, \epsilon, Z_0) \rightarrow (q_f, Z_0)$



Pumping Lemma for Regular Language

* To prove language is not Regular
* Not used to prove language is Regular.
* Proof of contradiction.

Theorem :-

If 'A' is Regular language and A has pumping length 'p' will generate string 's' $\forall |s| \geq P$ can be divided into $s = xyz$ has to satisfy.

i) $s = xy^iz \in A$ for $i = 0, 1, 2, 3, \ldots n$

ii) $|y| > 0$

iii) $|xy| \leq P$

eg :- $a^n b^n / n \geq 0$

Pumping Lemma for CFL

* To prove language is not CF
* Cannot prove language is CF.
* Proof of contradiction.

Theorm :-

$A \rightarrow CFL$

$A \rightarrow PL \rightarrow P$

$|s| \geq P$

Scanned by CamScanner

$s = uvwxy$

i) $s = uv^i wx^i y \in A$ for $i = 0, 1, 2, \cdots n$

ii) $|vx| > 0$

iii) $|vwx| \le P$

PL - pumbing length

eg :- $a^n b^n c^n / n \ge 0$

→ PT $L = \{a^n b^n / n \ge 0\}$ is not Regular

↕ theorm write

Assume $L \Rightarrow A = \{a^n b^n / n \ge 0\}$ is regular

(P) PL of $A = 3$

$s = aaabbb$

$\underset{x}{\underline{aa}}\,\underset{y^i}{(ab)}\,\underset{z}{\underline{bb}}$ $\underset{a}{\underline{aaa}}\,\underset{}{(b)_{y_i}}\,\underset{z}{\underline{bb}}$ $\underset{x}{\underline{aaa\ b}}\,\underset{y^i}{(bb)}$

$i=0$, aabb	$i=0$, aaabb	$i=0$, aaab
$i=1$, aaabbb	$i=1$, aaabbb	$i=1$, aaabbb
$i=2$, aaababbb	$i=2$, aaabbbb	$i=2$, aaabbbbb

Apart from 4 cases all the other are not regular.

So the L is a non regular.