1. Write the grammar to generate "signed integers".

Soln: Integer : 'a number'

Defn: . A digit is a number
. number followed by digits
. Digit followed by number

Ex: +28, −35, $\epsilon\epsilon 50\epsilon$.

$S \rightarrow +$

$S \rightarrow -$

$S \rightarrow \epsilon$

$\left.\begin{matrix} SD \\ SN \end{matrix}\right\}$ ✓

$N \rightarrow D$

$D \rightarrow 0/1/2/3/4/5/6/7/8/9$

$N \rightarrow ND/DN$

$\boxed{SI \rightarrow SN}$

II Elimination of UNIT productions

G		G'	
$S \to A$	– ①	$S \to a$	– ① ✓
$A \to B$	– ②		
$B \to C$	– ③		
$C \to a$	– ④		

Unit

One Variable is simply replaced by another variable.

$S \to A$
$\to B$
$\to C$
$\to a$

$G = G'$
$L(G) = L(G')$
a   a

Unit ?

$A \to B$
LHS – single non/t
RHS – ~~n~~
Unit ✓

How to ?

Date: 4.11.2020

Elimination of Unit productions

Unit production: A production is said to be Unit production, if it has the following form

$$A \rightarrow B$$

where $A, B \in V$

i.e., LHS & RHS of each production will be having only one grammar symbol which is non-terminal.

G
$$\begin{cases} A \rightarrow B \\ B \rightarrow C \\ C \rightarrow D \\ D \rightarrow a \end{cases}$$

$L(G) = a$

G'
$$\{ A \rightarrow a$$

$L(G') = a$

Conclusion: Unit productions are undesirable, hence they must be eliminated.

all the non-unit productions of B & C are also the productions of A.

Similarly, the process is repeated for all unit productions.

Ex: Eliminate unit productions from the following grammar

$$
\begin{aligned}
S &\to AB \\
A &\to a \\
B &\to C \mid b \\
C &\to D \\
D &\to E \mid bC \\
E &\to d / Ab
\end{aligned}
$$

Step-3: List all unit productions

$$
\left.\begin{aligned}
B &\to C \\
C &\to D \\
D &\to E
\end{aligned}\right\} \text{(3)}
$$

Soln:

1. There are no ε-productions
2. List all non-unit prods.

$$
\left.\begin{aligned}
S &\to AB \\
A &\to a \\
B &\to b \\
D &\to bC \\
E &\to d / Ab
\end{aligned}\right\} \text{(6)}
$$

$$B \to C \to D \to E$$

All the non-unit productions of C, D, E are also the productions of B.

$D \Rightarrow E$: All the non-unit production of E are also the prodns of D

$$\boxed{D \to bC / d / Ab}$$

$B \Rightarrow C$ All the non-unit productions of C are also the productions of B

$$\boxed{B \rightarrow bC / d / Ab / c}$$

Now combine all non-unit productions with these new derived non-unit productions.

$S \rightarrow AB$
$A \rightarrow a$
$B \rightarrow bC / d / Ab / c$
$C \rightarrow bC / d / Ab$
$D \rightarrow bC / d / Ab$
$E \rightarrow d / Ab.$

with no Unit productions.

Q. Eliminate all unit productions from the following G.

$S \to Aa \mid B \mid Ca$

$B \to aB \mid b$

$C \to Db \mid D$

$D \to E \mid d$.

Soln:

(1) Eliminate all ϵ-productions.

No ϵ-prodns.

(2) List all non-unit productions

$$\left.\begin{array}{l} S \to Aa \mid Ca \\ B \to aB \mid b \\ C \to Db \\ D \to \underline{\underline{d}} \end{array}\right\}$$

(3) List all unit prodns

$S \to B$

$C \to D$

$D \to E$

$S \to B$ — (1)

$C \to D \to E$ — (2)

$S \to B$: All non-unit B productions are also S productions.

$\boxed{S \to aB \mid b}$

$D \Rightarrow E$

$C \Rightarrow D$

$\boxed{C \to d}$

Now merge the non-unit productions of Step-2 and derived productions of Step-3.

$$\left.\begin{aligned} &S \rightarrow Aa \mid Ca \mid aB \mid b \\ &B \rightarrow aB \mid b \\ &C \rightarrow \cancel{Db} \mid d \\ &D \rightarrow d \end{aligned}\right\} G' \text{ with no } \epsilon\text{-prods}$$

With E prodn.

$S \rightarrow Aa \mid B \mid Ca$

$B \rightarrow aB \mid b$

$C \rightarrow Db \mid D$

I

$D \rightarrow E \mid d$

$E \rightarrow ab$

$S \rightarrow B$

$C \rightarrow D$

$D \rightarrow E$

$S \rightarrow aB \mid b \mid Aa \mid Ca$

$D \rightarrow ab \mid d$

$C \rightarrow D$

$C \rightarrow ab \mid d$

II

I & II

Eliminate Unit-productions from

$$\begin{aligned} S &\to A0 \mid B \\ B &\to A \mid 11 \\ A &\to 0 \mid 12 \mid B \end{aligned}$$

$T = \{0, 1, 2\}$

$V = \{A, B\}$

$S - SS$

$$\begin{aligned} S &\to B \\ B &\to A \\ A &\to B \end{aligned}$$

$$\begin{tikzcd} S \arrow[r] & B \arrow[r, bend left] & A \arrow[l, bend left] \end{tikzcd}$$

$$\begin{aligned} S &\to 11 \mid A0 \mid 0 \mid 12 \\ A &\to 11 \mid 0 \mid 12 \\ B &\to 0 \mid 12 \mid 11 \end{aligned}$$

The resultant grammar has Unit productions, eliminate them

$S \rightarrow aA / a / B / C$
$A \rightarrow aB$
$B \rightarrow aA / a$
$C \rightarrow cCD$
$D \rightarrow abd$

To eliminate Unit productions.

1. List all non-unit productions

$S \rightarrow aA / a$

$A \rightarrow aB$

$B \rightarrow aA / a$

$C \rightarrow cCD$

$D \rightarrow abd$

2. List all Unit produ

Chomsky Hierarchy: 21. 11. 2020

Noam Chomsky - Mathematician

Classified the Grammar into 4- types.

Type- 0 - Unrestricted Grammar

Type - 1 - Context Sensitive Grammar

Type - 2 - Context Free Grammar (CFG)

Type - 3 - Regular Grammar

Type - 0 - Grammar - UNRESTRICTED GR.

Defn: $G = (V, T, P, S)$ is said to be Type-0 grammar if all the production are of the form

$$\alpha \rightarrow \beta$$

where $\alpha \in (V \cup T)^+$ $\quad (V+T)^+$

$\beta \in (V \cup T)^*$. $\quad (V+T)^*$

Ex:

$AbC \rightarrow CAa$

$CAb \rightarrow BaBc$

$BaBc \rightarrow Bbd$

* The languages / strings generated by Type-0 Grammar are called Recursively Enumerable Languages

* Usually REL's are accepted by Turing Machines.

Type-1 - Grammar - Context Sensitive Grammar (CSG)

$G = (V, T, P, S)$ is said to be type-1 Grammar if all the productions are of the form

$$\alpha \rightarrow \beta$$

where $\alpha \in (V \cup T)^+$

$\beta \in (V \cup T)^*$

$$\boxed{\& \quad |\alpha| \leq |\beta|}$$

$ABa \rightarrow BCaD$

$Bab \rightarrow ABDc$

$|\alpha| \leq |\beta|$

$aa \rightarrow BBD$

• Type-1 - CSG $\rightarrow$ Context Sensitive Languages CSL

LBA & TM

Type-2 Grammar (CFG)

Definition: $G = (V, T, P, S)$ is said to be Type-2, if all the productions are of the form

$$A \rightarrow \beta$$

where $A \in \underline{V}$
$\beta \in (V \cup T)^*$

$S \rightarrow aSb \mid ab$
$S \rightarrow a\underline{S}b$
$\rightarrow aaSbb$
$\rightarrow aaaSbbb$

$\underline{a^n b^n}$

This type-2 Grammar is also called CONTEXT FREE GRAMMAR ($\underline{CFG}$)

$CF\underline{L}$'s

In CFG:

* LHS of each production must be single non-terminal
* RHS of each production must be combination of terminals and/or non-terminals including ϵ.

$\underline{Ex}$: $G \rightarrow \left\{ \begin{array}{l} S \rightarrow aSb \\ S \rightarrow ab \end{array} \right.$ CFG

$\rightarrow L(G) = \underline{a^n b^n} \mid n \geq 1$

Type-3:

Regular Gr Regular Gramm.

Regular languages

$A \to Ba$

$\quad\;\; aB$

Right linear / left linear.

menu

Date: 23.11.2020

NORMAL FORMS

What? Why? How?	$G = (V, T, P, S)$ is said to be in Normal Form, if all the productions of the grammar

(Context Free Grammar) are of the form which is suitable for required application. i.e., CFG (G) must transformed to G' where G' is the required version. Here G & G' are equivalent such that $L(G) = L(G')$

Ex:

(CFG) Grammar G

$S \to ABC$
$A \to a$
$B \to b$
$C \to c$

CFG G'

$S \to AD$
$D \to BC$
$A \to a$
$B \to b$
$C \to c$

<<

>>

enu

<<

>>

Types of Normal Forms

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

CNF: Chomsky Normal Form.

What is CNF? How write G in CNF?

Definition: $G = (V, T, P, S)$ is said to be in CNF, if all the productions are of the form

$$\left.\begin{array}{l} A \to BC \\ \text{or} \\ A \to a \end{array}\right\} \text{ where } \begin{array}{l} A, B, C \in V \\ a \in T \end{array}$$

i.e., in CNF, ~~LHS~~ RHS of each production must contain either two non-terminals or one terminal.

Ex:

$S \to AB$
$A \to a$ ✓
$B \to b$

$S \to Aa$ ✗
$A \to a$ ✓
Is it in CNF?
Not in CNF

CFG to CNF

To express CFG in CNF:

Input: CFG which is not in CNF

Output: CFG in CNF

Method: Apply the transformation.

Ex:

CFG (G)

$S \to ABCa$

$A \to a$

$B \to b$

$C \to c$

G' in CNF

$S \to ABCA$

$A \to a$

$S \to A D_1$

$D_1 \to BCA$ ×

$S \to D_1 D_2$

$D_1 \to AB$ ✓

$D_2 \to CA$

$A \to a$

$B \to b$

$C \to c$

Given CFG

$S \to 0A / 1B$
$A \to 0AA / 1S / 1$
$B \to 1BB / 0S / 0$

Apply the transformation rules.

Consider

$S \to 0A$: Replace $S \to 0A$ by $\boxed{S \to B_0 A}$ where $\boxed{B_0 \to 0}$ ✓

$S \to 1B$: Replace $S \to 1B$ by $\boxed{S \to B_1 B}$ where $\boxed{B_1 \to 1}$

$S \to 0AA$: Replace $S \to 0AA$ by $\boxed{S \to B_0 D_1}$ where $D_1 \to AA$

$B_0 \to 0$

$S \to B_0 D_1$
$D_1 \to AA$ ✓
$B_0 \to 0$ ✓

$S \to 1S$: $\boxed{S \to B_1 S}$

$\boxed{B_1 \to 1}$

$B \to 1BB$

$B \to B_1 BB$
$B \to B_1 D_2$
$D_2 \to BB$

$B_1 \to B_1 D_2$
$D_2 \to BB$

enu

$S \rightarrow 0A \mid 1B$
$A \rightarrow 0AA \mid 1S \mid 1$
$B \rightarrow 1BB \mid 0S \mid 0$ } Express in CNF

Soln:

1. Apply all ε-productions – (No ε-prodn)
2. Apply all unit productions – (No unit prodn)
3. Consider each production.

*) $S \rightarrow 0A \Rightarrow \boxed{S \rightarrow B_0 A}$ $\boxed{B_0 \rightarrow 0}$ ✓

*) $S \rightarrow 1B \Rightarrow \boxed{S \rightarrow B_1 B}$ $\boxed{B_1 \rightarrow 1}$

*) $A \rightarrow 0AA$:

$A \rightarrow B_0 AA$? $\boxed{B_0 \rightarrow 0}$

Further consider

$A \rightarrow B_0 AA$ & replace

$\boxed{A \rightarrow B_0 D_1}$ where $\boxed{B_0 \rightarrow 0}$

*) $A \rightarrow 1S$ $\boxed{A \rightarrow B_1 S}$ $\boxed{B_1 \rightarrow 1}$

*) $B \rightarrow 1BB$

$B \rightarrow B_1 BB$ where $\boxed{B_1 \rightarrow 1}$

$\boxed{B \rightarrow B_1 D_2}$ $\boxed{D_2 \rightarrow BB}$

<< >>

$B \to 0S$ $\boxed{B \to B_0 S}$ $\boxed{B_0 \to 0}$

Final G in CNF

$S \to B_0 A / B_1 B$

$B_0 \to 0$

$B_1 \to 1$

$A \to B_0 D_1$

$D_1 \to AA$

$A \to B_1 S$

$B \to B_1 D_2$

$D_2 \to BB$

$B \to B_0 S$

Derived — 1

G' in CNF

$A \to 1$

$B \to 0$

Original CNF — 2

1+2

Normal Forms

Express the following CFG in CNF.

$S \rightarrow ASB \mid \epsilon$
$A \rightarrow aAS \mid a$
$B \rightarrow SbS \mid A \mid bb$

Soln: (1) Eliminate ϵ-productions

Nullable Variable $NV = \{S\}$

$$G' \begin{cases} S \rightarrow ASB \mid AB \\ A \rightarrow aAS \mid aA \mid a \\ B \rightarrow SbS \mid bS \mid Sb \mid b \mid A \mid bb \end{cases}$$

G' after elimination of ϵ-productions.

(2) Eliminate Unit productions

$B \rightarrow A$

$$G'' \begin{cases} S \rightarrow ASB \mid AB \\ A \rightarrow aAS \mid aA \mid a \\ B \rightarrow SbS \mid bS \mid Sb \mid b \mid aAS \mid a \mid bb \end{cases}$$

with no UNIT prods

- $B \to Sb$ $\quad \boxed{B \to SC_b}$ $\quad \boxed{C_b \to b}$
- $B \to bS$ $\quad \boxed{B \to C_b S}$ $\quad \boxed{C_b \to b}$
- $\boxed{B \to b}$
- $B \to bb$ $\quad \boxed{B \to C_b C_b}$ $\quad \boxed{C_b \to b}$
- $B \to aAS$ $\quad B \to C_a AS$ $\quad \boxed{B \to C_a D_4}$

  $\boxed{D_4 \to AS}$
- $B \to aA$ $\quad \boxed{B \to C_a A}$ $\quad \boxed{C_a \to a}$
- $\boxed{B \to a}$

CFG in CNF

$S \to AD_1 \mid AB$

$D_1 \to SB$

$A \to C_a D_2 \mid C_a A \mid a$

$D_2 \to AS$

$C_a \to a, \quad C_b \to b$

$B \to SD_3 \mid SC_b \mid C_b S \mid C_a C_a$

$D_3 \to C_b S$ $\quad \boxed{D_4 \to AS}$

$B \to C_b C_b \mid C_a D_4 \mid C_a A$

Q Express the following CFG in CNF

$S \rightarrow AaB \mid aaB$

$A \rightarrow \epsilon$

$B \rightarrow bbA \mid \epsilon$

Solution: (1) Eliminate ϵ-prodns

$S \rightarrow AaB \mid aB \mid Aa \mid aaB \mid aa \mid a$

$B \rightarrow bbA \mid bb$

(2) No unit prodns.

(3) • $S \rightarrow AaB$ · $S \rightarrow AC_aB$

$\boxed{S \rightarrow AD_1}$ $\boxed{D_1 \rightarrow C_aB}$

$\boxed{C_a \rightarrow a}$.

• $S \rightarrow aB$ $\boxed{S \rightarrow C_aB}$ $\boxed{C_a \rightarrow a}$

• $S \rightarrow Aa$ $\boxed{S \rightarrow AC_a}$ $\boxed{C_a \rightarrow a}$

• $S \rightarrow aaB$ $S \rightarrow C_aC_aB$. $\boxed{S \rightarrow C_aD_2}$

$\boxed{D_2 \rightarrow C_aB}$

enu

$S \to aa$

$\boxed{S \to C_a C_a}$ $\boxed{C_a \to a}$

$\boxed{S \to a}$ ✓

$B \to bbA$

$B \to C_b C_b A$

$\boxed{B \to C_b D_3}$ $\boxed{D_3 \to C_b A}$

$B \to bb$ | $\boxed{B \to C_b C_b}$ $\boxed{C_b \to b}$

final G_1 in CNF

$S \to AD_1 / C_a B / A C_a / C_a D_2 / C_a C_a / a$

$D_1 \to C_a B$

$C_a \to a$ in CNF

$D_2 \to C_a B$

$B \to C_b D_3 / C_b C_b$

$D_3 \to C_b A$

<<

>>

Date: 25.11.2020

Recap: Normal Forms

- CNF — What, Why, How
- GNF — What?, Why — PDA, Defn?

Elimination of Useless Symbols

G — CFG

$S \rightarrow AB$
$A \rightarrow a$
$C \rightarrow d$
$B \rightarrow b$

$\Longrightarrow$

G′

$S \rightarrow AB$
$A \rightarrow a$
$B \rightarrow b$

$L(G) = L(G')$

$C \rightarrow d$ useless prodn / symbols

$S \rightarrow AB$
$\rightarrow aB$
$\rightarrow ab$

$S \rightarrow AB$
$\rightarrow aB$
$\rightarrow ab$

Conclusion: Sometimes the given Grammar may contain unwanted productions

Q. Simplify the following Grammar

$S \to aA \mid a \mid Bb \mid cC$
$A \to aB$
$B \to a \mid Aa$
$C \to aCD$
$D \to ddd$

} G

Soln:

1. Eliminate all ϵ-prodns.
   No ϵ-productions.
2. Eliminate Unit prodns
   No unit productions
3. Eliminate Useless Symbols

Part-A: To obtain G' with the non-terminals generating terminals.

OV	NV	Productions
$\emptyset$	$\{S, B, D\}$	$S \to a$, $B \to a$, $D \to ddd$
$\{S, B, D\}$	$\{S, A, B, D\}$	$S \to Bb$, $A \to aB$
$\{S, A, B, D\}$	$\{S, A, B, D\}$	$S \to aA$, $B \to Aa$

PART-B Variables
List out the productions which are reachable from S.

P'	T'	V'
		S
$S \to a \mid Bb$, $S \to aA$	$\{a, b\}$	$\{S, A, B\}$
$A \to aB$, $B \to a$, $B \to Aa$	$\{a, b\}$	$\{S, A, B\}$

III Iteration $\quad S \to cC$

$S \to aA$

$(OV \cup T)^*$

$[\{S, A, B, D\} \{a, b, c, d\}]^*$

NV - 2nd iteration.

$NV = (OV \cup T)^*$
$= [\{S, B, D\} \cup \{a, b, c, d\}]^*$

$S \to aA$

$S \to Bb$

$S \to cC$

$A \to aB$

Resultant Simplified Grammar (CFG) with no useless symbols.

I $S \to a / Bb / aA$
$A \to aB$
$B \to a / Aa$ (G)

Most simplified Grammar.

(G)

$S \to a$ ✓

$S \to Bb$
$\to ab$

aaaab

$S \to aA$
$\to aaB$
$\to aaaB$
$\to aaaAa$
$\to aaaab$
$\to aaaab$

$S \to cC$
$\to cCCD$
$\to ccCCD$
⋮
Stop

$L(G) = L(G')$

Verification: (G')

II Original Grammar

$S \to aA / a / Bb / cC$
$A \to aB$
$B \to a / Aa$
$C \to cCD$
$D \to dda$ (G')

$S \to a$ ✓

$S \to Bb$
$\to ab$

$S \to aA$
$\to aaB$
$\to aaAa$
$\to aaaBa$
$\to aaaaa$

Procedure for elimination of Useless Symbols from G.

Soln: PART-A: List all the variables which are generating terminals

$OV = \emptyset$

$NV = OV \cup \{A \mid A \to \gamma \text{ and } \gamma \in (OV \cup T)^*\}$

While $(OV \mathrel{!=} NV)$

{

$OV = NV;$

$NV = OV \cup \{A \mid A \to \gamma \text{ and } \gamma \in (OV \cup T)^*\}$

}

$\underline{V_1 = OV}$

PART-B: List all the productions connected to Start symbol S

$V' = \{S\}$

For each A in V'

If $A \to \alpha$ then

Add variables of α to V'

Add Terminals of α to T'

endif

Q. Eliminate Useless symbols from the following G:

Soln:

$$S \to aA \mid bB$$
$$A \to aA \mid a$$
$$B \to bB$$
$$D \to ab \mid Ea$$
$$E \to aC \mid d$$

G

Soln:

1) Eliminate ϵ-productions
NIL

2) Eliminate Unit prodns
NIL

3) Eliminate Useless Symbols

PART: A

OV	NV	Prodns.
$\emptyset$	$\{A, D, E\}$	$A \to a$, $D \to ab$, $E \to d$
$\{A, D, E\}$	$\{S, A, D, E\}$	$S \to aA$, $A \to aA$, $D \to Ea$
$\{S, A, D, E\}$	$\{S, A, D, E\}$	

G'

$(OV \cup T)^*$

$[\{A, D, E\} \cup \{a, b, d\}]^*$

$\Downarrow$

$[\{S, A, D, E\} \cup \{a, b, d\}]^*$

PART-B: Reachable from S

P'	T'	V'
		$\{S\}$
$S \to aA$	$\{a\}$	$\{S, A\}$
$A \to a$, $A \to aA$	$\{a\}$	$\{S, A\}$

Most Simplified Grammar with no Useless Symbols

$$S \to aA$$
$$A \to a \mid aA$$

G''

1. Eliminate ϵ-productions:

Set of nullable variables $= \{B, D\}$

Using these nullable variables write the grammar which has no ϵ-productions.

$$
\begin{aligned}
&S \rightarrow A\underline{B}C \,/\, AC \\
&S \rightarrow \underline{B}a\underline{B} \,/\, aB \,/\, Ba \,/\, a \\[6pt]
&A \rightarrow aA \\
&A \rightarrow \underline{B}aC \,/\, aC \\
&A \rightarrow aaa \\[6pt]
&B \rightarrow b\underline{B}b \,/\, bb \\
&B \rightarrow a \\
&B \rightarrow D \\[6pt]
&C \rightarrow CA \\
&C \rightarrow AC
\end{aligned}
$$

Resultant Grammar with no ϵ-productions.

2. Eliminate Unit productions from the resultant grammar.

Only Unit production is $B \rightarrow D$

$$B \longrightarrow D$$

All the non-unit productions of D are also the productions of B.

Resultant Grammar with no unit-productions

$S \rightarrow ABC \mid AC \mid BaB \mid aB \mid Ba \mid a$

$A \rightarrow aA \mid BaC \mid aC \mid aaa$

$B \rightarrow bBb \mid bb \mid a$

$C \rightarrow CA$

$C \rightarrow AC$

G with no unit productions

Step-3

Part-B: List all the productions which are reachable form S.

Prod	T'	V'
		S
$S \rightarrow BaB / aB$ $S \rightarrow Ba / a$	{a}	{S, B}
$B \rightarrow bBb / bb / a$	{a, b}	{S, B}

Final Grammar with no Useless Symbols

$S \rightarrow BaB / aB / Ba / a$

$B \rightarrow bBb / bb / a$

G'''

Original G.

$L(G) = L(G''')$

$S \rightarrow ABC$
$\rightarrow aaaBC$
$\rightarrow aaaaC$
$\rightarrow aaaa$ ⋮

(4) Express the resultant Grammar in CNF.

$S \rightarrow BaB$

$S \rightarrow B\underline{C_aB}$ $\boxed{C_a \rightarrow a}$

$\boxed{S \rightarrow BD_1}$ $\boxed{D_1 \rightarrow C_aB}$

$S \rightarrow aB$ $\boxed{S \rightarrow C_aB}$

$S \rightarrow Ba$ $\boxed{S \rightarrow BC_a}$

$\boxed{S \rightarrow a}$

$B \rightarrow bBb$ $B \rightarrow C_bBC_b$ $\boxed{C_b \rightarrow b}$

$\boxed{B \rightarrow C_bD_2}$ $\boxed{D_2 \rightarrow BC_b}$

$B \rightarrow bb$ $\boxed{B \rightarrow C_bC_b}$

$\boxed{B \rightarrow a}$

Resultant Grammar which is CNF.

$S \rightarrow BD_1 / C_aB / BC_a / a$

$D_1 \rightarrow C_aB$

$C_a \rightarrow a$

$B \rightarrow C_bD_2 / C_bC_b / a.$

$D_2 \rightarrow BC_b$

CNF

All the prodns are of the from

$A \rightarrow BC$

or

$A \rightarrow a$

Date : 28/11/2020. ATC

Simplification of CFG & Normal Forms

Q. Begin with the grammar (CFG).

$S \rightarrow ABC \mid BaB$

$A \rightarrow aA \mid BaC \mid aaa$

$B \rightarrow bBb \mid a \mid D$

$C \rightarrow CA \mid AC$

$D \rightarrow \epsilon$

a) Eliminate ϵ-productions

b) Eliminate any unit productions in the resulting grammar

c) Eliminate useless symbols from the resulting grammar

d) Express the resultant grammar in CNF.

Date: 30/11/2020.

1. What is simplification of CFG?
2. Why simplification
3. How to do simplification
   - ε - pro
   - Unit
   - Elimination of Useless Symbols
   - Normal Forms
     - ✓ CNF
     - GNF

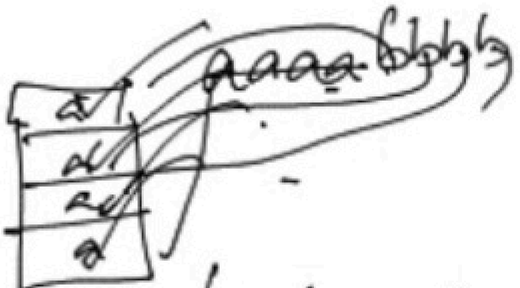
Also, Chomsky hierarchy:

Classification of Grammars

- Type - 0
- Type - 1
- $\boxed{\text{Type - 2} \quad \text{CFG - Compiler design.}}$
- Type - 3

Need for Stack:

Ex: CFL $L = \{a^n b^n : n \geq 0\}$.



Type of / nature of Problems to be solved in PDA

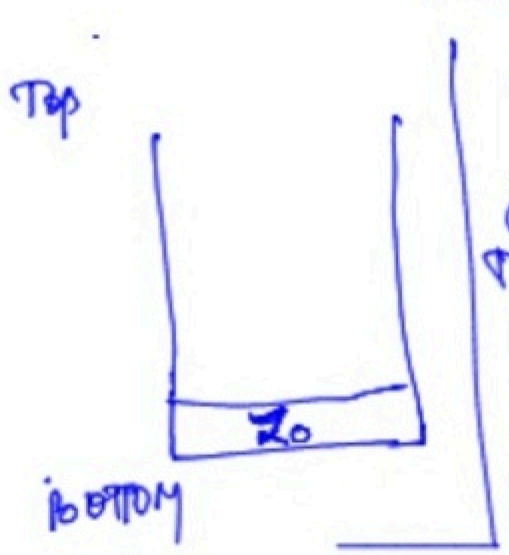
Q. Design PDA for $L = \{a^n b^n : n \geq 0\}$

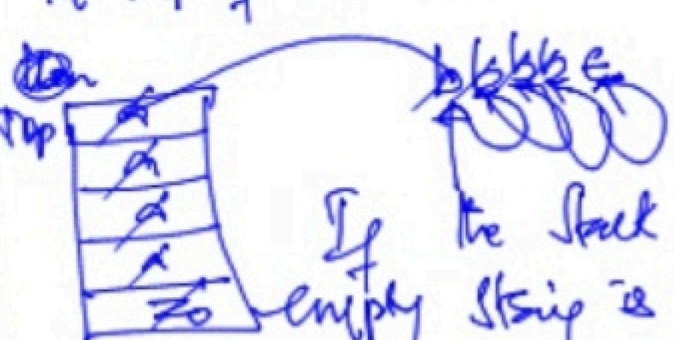
$L = \{\epsilon, aabb, ab, a^3b^3 \ldots\}$

Acceptable String: aaaabbbbϵ

ϵ - end of the string.

LOGIC: Push all 4 a's onto the top of the stack.





If the stack is empty string is valid.

Pushdown Automata

STACK

?

$+ \; (Q, \Sigma, \delta, q_0, F)$

Push

Data Structures – II

Operation (STACK)

STACK :- linear data structure – Physical

One active end

Top

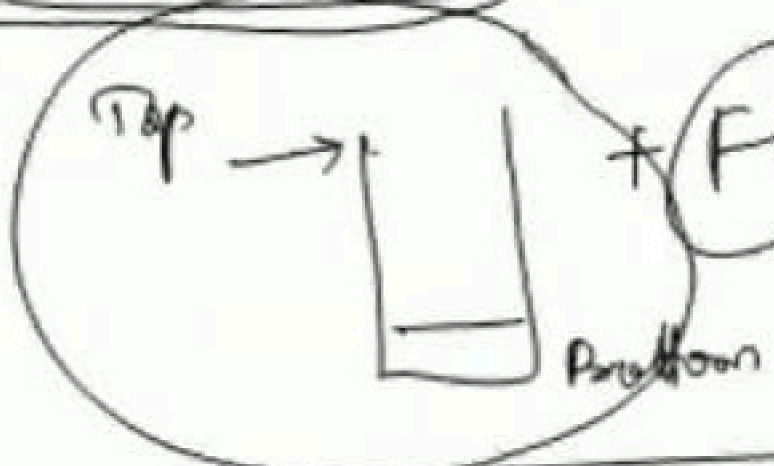
Bottom

Passive

push – insert

pop – delete

Auxiliary Memory → Microprocessor / CO



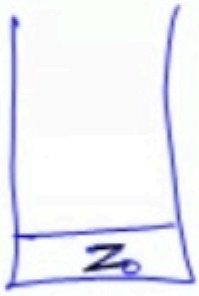
+ FA = PDA

$\boxed{\text{STACK + FA = PDA}}$

Transitions of PDA:

aaaabbbb ε

q_0

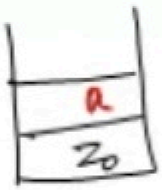


Transitions

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$ -①

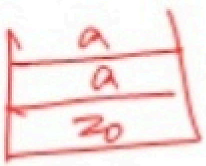
aaabbbbε

q_0



Top of the

$\delta(q_0, a, a) = (q_0, aa)$ -②



After applying ② for sequence no. of times

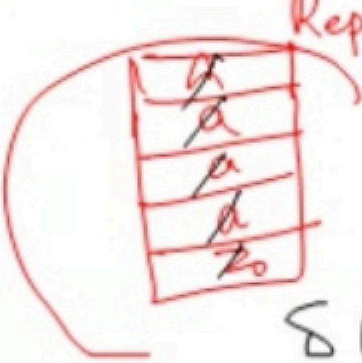
bbbbε

For each b one a is to be popped out ③

aabbbbε

q_0

Repeat the transition ②



abbbbε

q_0

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

$\delta(q_0, \epsilon, Z_0) = (q_1, \epsilon)$

Formal definition of PDA.

PDA is 7- tuple

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

✓ Q. Set of States

✓ Σ. input Symbols

δ. is transition function.

✓ ~~q_0 - is Start State~~

✓ Z_0 - is initial Symbol on Stack.

Γ - Set of Stack Symbol

$$\delta(q_0, a, Z_0) = (q_0, \underline{aZ_0})$$

PDA $\boxed{Q \times \Sigma \times \Gamma = (Q \times \Gamma^*)}$

Date: 1.12.2020

$L = \{a^n b^n : n \geq 0\}$ is Context-Free language. (cfl)

$\Downarrow$ $S \to aSb \mid \epsilon$ - CFG.

Formal Definition of PUSHDOWN AUTOMATA:

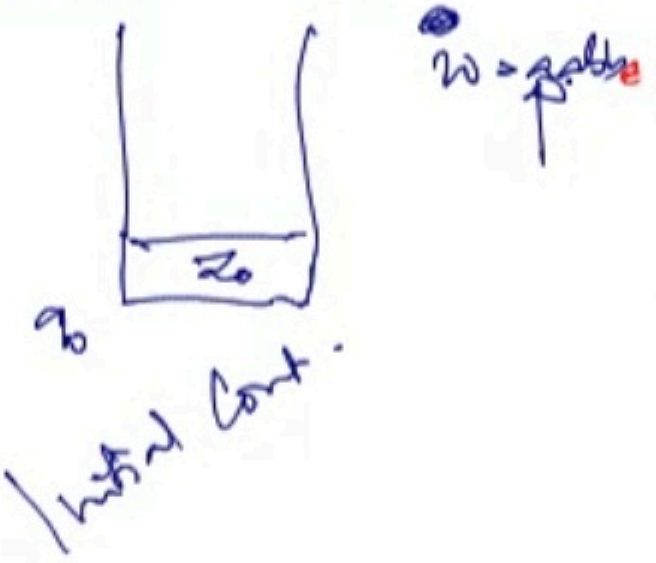
$M = (Q, \Sigma, \delta, q_0, F, Z_0, \Gamma) \longrightarrow \Gamma = \{Z_0, a\}$

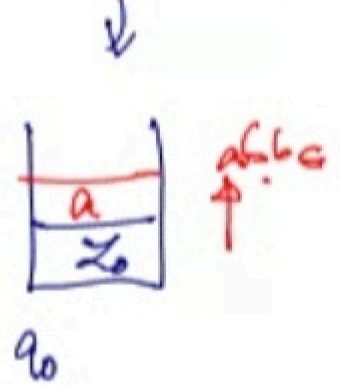
F ↓ Set of final states.

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

In the state q_0, when the input symbol encountered is a and the top of the stack is Z_0.

Push the input symbol a onto the top of the stack and be there in same state q_0

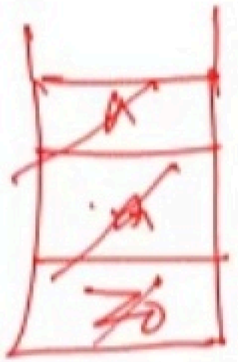




Pop the element from top of the Stack:

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

Signal/indicate pop operation



q_0

$b\ \epsilon$

q_0

$b\ \epsilon$

Replace Transition.

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

Q. Design PDA for $L = \{a^n b^n : n \geq 0\}$

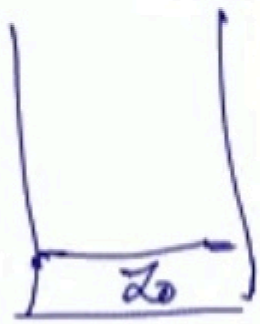
Solution:

1. Set of acceptable strings

$L = \{\epsilon, ab, aabb \ldots\}$

2. LOGIC:

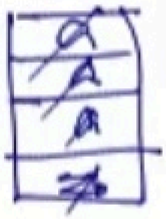
Consider the string $w = aaabbb\epsilon$



(1) Push all a's on to the top of the stack in the state q_0.

(2) Next, from left most b, for each b in string w there will be corresponding 'a' on the top of the stack

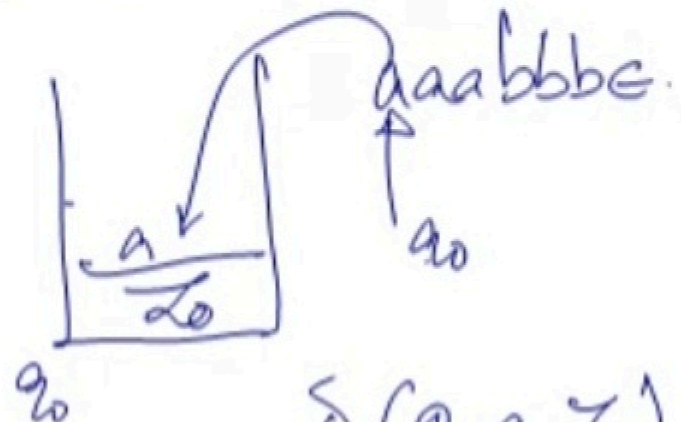
$aaabbb\epsilon$



~~bbbε~~ } ✓

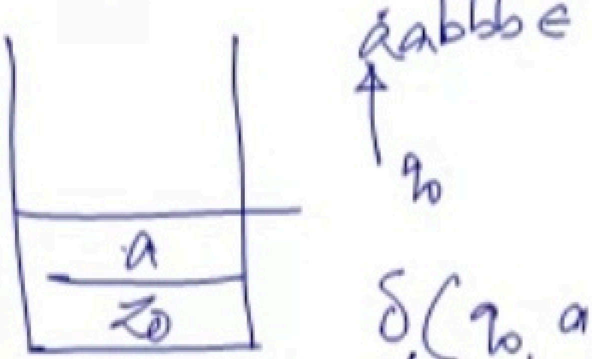
q_0 string will be accepted as the stack becomes empty

③ To write the transitions

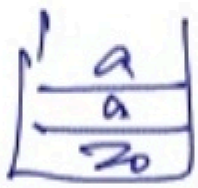


$$\delta(q_0, a, Z_0) = (\underline{q_0}, aZ_0) \quad \text{— ①}$$

After applying the transition ①, the string aaabbbε becomes



$$\delta(q_0, a, a) = (q_0, \underline{aa}) \quad \text{— ②}$$



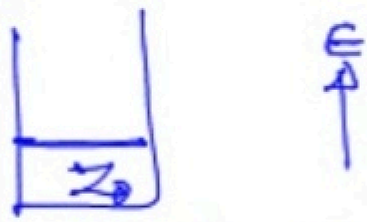
this becomes

abbbε

Apply ② for required no of times

bbbε

After applying (3) for required no. of times



$$\delta(q_0, \epsilon, Z_0) = (q_0, \epsilon)$$

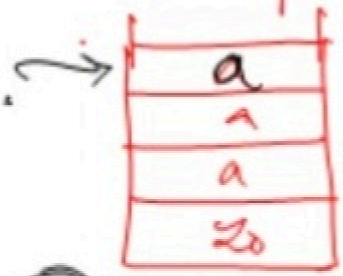
hence the

Last out step is accepted (stack is empty)

List out the transitions

1. $\delta(q_0, a, Z_0) = (q_0, aZ_0)$
2. $\delta(q_0, a, a) = (q_0, aa)$
3. $\delta(q_0, b, a) = (q_0, \epsilon)$
4. $\delta(q_0, \epsilon, Z_0) = (q_0, \epsilon)$

After pushing all a's on to the stack



w

$\cancel{b}bb\epsilon$

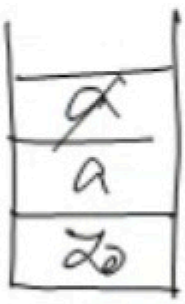
q_0.

Pop one a for each b.

$$\delta(q_0, b, a) = (q_0, \epsilon) \quad ③$$

pop the top of the stack.

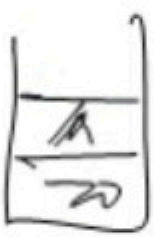
after applying ③



$\cancel{b}b\epsilon$

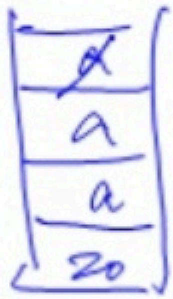
q_0

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



$\cancel{b}\epsilon$

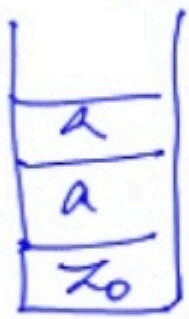
After applying ③ for required no. of times the top of the stack is Z_0



$\cancel{b}bb\epsilon$ ↑ q_0

$$\delta(q_0, b, a) = (q_1, \underbrace{\epsilon}_{\text{pop}}) \quad - \text{③}$$

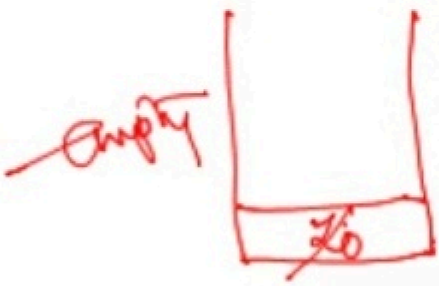
after applying transition ③ Stack contents are as follows



$bb\epsilon$ ↑ q_1

$$\delta(q_1, b, a) = (q_1, \epsilon) \quad - \text{④}$$

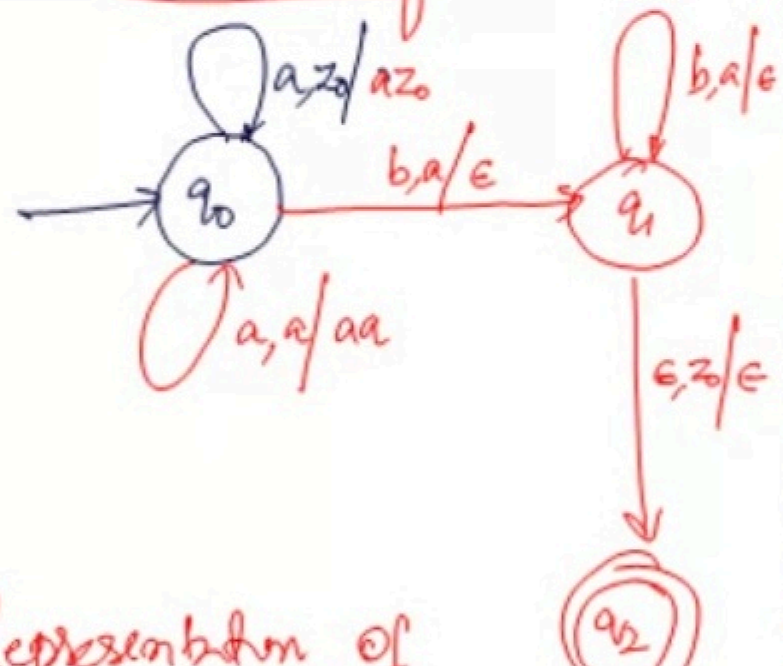
after applying ④ for required no. of times



$\cancel{\epsilon}$ ↑ q_1

$\delta(q_1, \epsilon, Z_0) = (q_2, \epsilon)$ → empty

Transition diagram



δ

Transition

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

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

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

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

$\delta(q_1, \epsilon, Z_0) = (q_2, \epsilon)$

Representation of transition using digraph. (Transition diagram)

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

$Q = \{q_0, q_1, q_2\}$

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

δ:

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

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

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

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

$\delta(q_1, \epsilon, Z_0) = (q_2, \epsilon)$

$q_0 \rightarrow$ Start State

$\Gamma = \{a, Z_0\}$

Z_0 is initial Symbol on the top of the Stack.

F is set of final States

To show that the string aaabbb is accepted.

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

$$\vdash (q_0, aabbb\epsilon, aZ_0)$$

$$\vdash (q_0, abbb\epsilon, aaZ_0)$$

$$\vdash (q_0, bbb\epsilon, aaaZ_0)$$

$$\vdash (q_1, bb\epsilon, aaZ_0)$$

$$\vdash (q_1, b\epsilon, aZ_0)$$

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

$$\vdash (q_2, \epsilon, \epsilon)$$

Final Condition.

Date: 15.12.2020

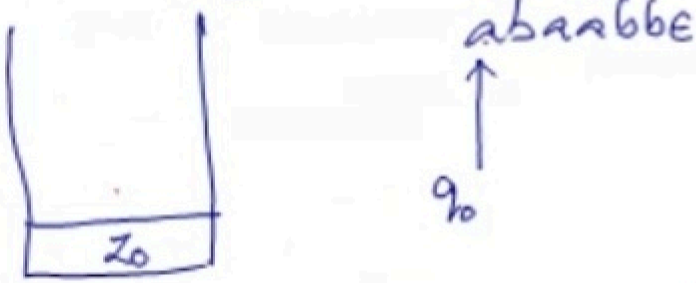
Q. Design PDA for $L = \{ w : w \in \{a,b\}^*$ with $n_a(w) = n_b(w) \}$

Solution:

1. $L = \{ \epsilon, ab, ba, abaabb, aabb, \ldots \}$

2. To write logic:

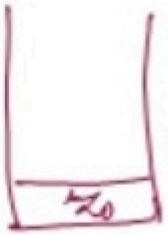
Initial configuration



1. Push whether it is 'a' or 'b' on to the Stack Z_0.
2. Without changing the State, apply the transitions:

   i) if the top of the Stack and the input symbol encountered are same then 'PUSH' otherwise 'Pop'.

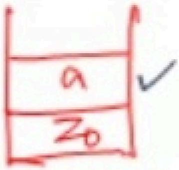
   ii) Repeat Step (1). →



abaabbϵ

q_0

$$\boxed{\delta(q_0, a, Z_0) = (q_0, aZ_0) \quad \text{— (1)}}$$

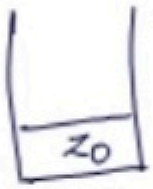


q_0

baabbϵ

q_0

$$\boxed{\delta(q_0, \underline{a}, \underline{b}) = (\underline{q_0}, \epsilon) \quad \text{— (2)}}$$



q_0

aabbϵ

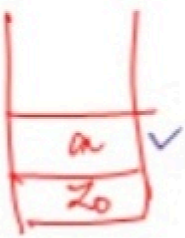
q_0

~~$\delta(q_0, a, Z_0$~~

(3)

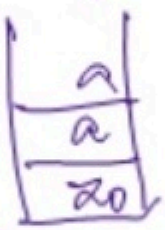
$$\boxed{\delta(q_0, a, a) = (q_0, \underline{a}a) \quad \text{— (3)}}$$

abbϵ



bbϵ

q_0



a

$$\boxed{\delta(q_0, \underline{b}, \underline{a}) = (q_0, \epsilon) \quad \text{— (4)}}$$

$$\delta(q_0, a, b) = (q_0, \epsilon) \text{ --- } 5$$
$$\delta(q_0, b, a) = (q_0, \epsilon) \text{ --- } 6$$

When the top of the stack and the input symbol encountered are different.

After applying the above transitions for required no. of times

$$\boxed{\begin{array}{l} \delta(q_0, \epsilon, Z_0) \text{ --- } 7 \\ \delta(q_0, \epsilon, Z_0) = (q_1, \underline{\epsilon, \epsilon}) \end{array}}$$

empty stack.

$\boxed{w = abab}$

Instanta

$$(q_0, ababe, Z_0) \vdash (q_0, babe, aZ_0) \text{ --- }$$
$$\vdash (q_0, ab\epsilon, Z_0)$$
$$\vdash (q_0, b\epsilon, aZ_0)$$
$$\vdash (q_0, \epsilon, Z_0)$$
$$\vdash \underline{\underline{(q_1, \epsilon, \epsilon)}}$$

To Show that the string w is rejected :

$w = aba$ (must be rejected means, it should not reach $\delta(q_0, \epsilon, Z_0)$).

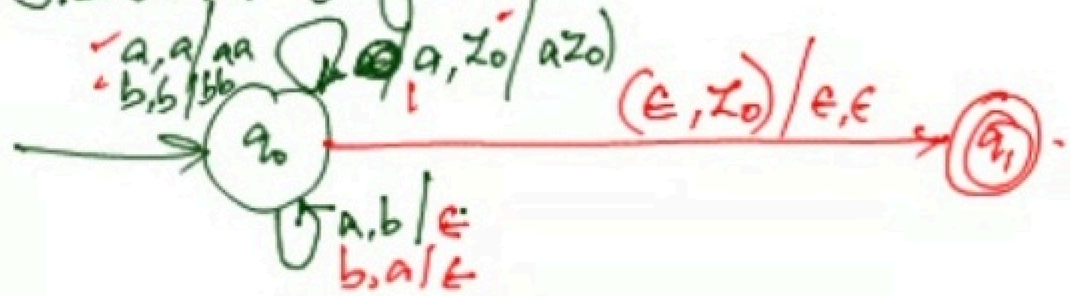
$$(q_0, aba\epsilon, Z_0) \vdash (q_0, ba\epsilon, aZ_0)$$

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

$$\vdash (q_0, \epsilon, aZ_0)$$

$\delta(q_0, \epsilon, a)$, the transition is not in the list. hence the string aba is rejected.

Transition Diagram



Finally

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

$Q = \{q_0, q_1\}$

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

δ :

$$\delta(q_0, a, Z_0) = (q_0, aZ_0)$$
$$\delta(q_0, b, Z_0) = (q_0, bZ_0)$$
$$\delta(q_0, a, a) = (q_0, aa)$$
$$\delta(q_0, b, b) = (q_0, bb)$$
$$\delta(q_0, a, b) = (q_0, \epsilon)$$
$$\delta(q_0, b, a) = (q_0, \epsilon)$$
$$\delta(q_0, \epsilon, Z_0) = \underline{(q_1, \epsilon, \epsilon)}$$

$$Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma^*$$

$\Gamma = \{\underline{a, b, Z_0}\}$

q_0 is the Start State

$F = \{q_1\}$

Z_0 is initial symbol on the top of the stack.

Show that ϵ is accepted.

$$(q_0, \epsilon, Z_0) \vdash (q_1, \epsilon, \epsilon)$$

Q. Design PDA for

Soln:

$$L = \{ a^n b^n : n \geq 1 \}$$

$$L = \{ ab, a^2b^2, a^3b^3 \ldots \}$$

$$\delta(q_0, a, Z_0) = (q_0, aZ_0) \quad \text{— (1)}$$

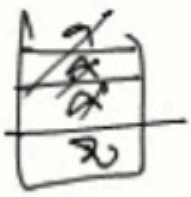
$$\delta(q_0, a, a) = (q_0, aa) \quad \text{— (2)}$$

$$\delta(q_0, b, a) = (q_1, \epsilon) \quad \text{— (3)}$$

$$\delta(q_1, b, a) = (q_1, \epsilon)^* \quad \text{— (4)}$$

$\cancel{b}\cancel{b}bb \cdot \epsilon$

$\uparrow q_1$



$$\delta(q_1, \epsilon, Z_0) = (q_2, \epsilon, \epsilon)$$

Menu

Q: Design PDA for $L = \{w : w \in \{a,b\}^*$ with $n_a(w) \geq n_b(w)$.

Solution:

① $L = \{a, aba, baa, abaa \ldots\}$

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, b, Z_0) = (q_0, bZ_0)$
} To push a/b onto the top of the Stack.

② $\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, b) = (q_0, bb)$
} Similar push.

$\delta(q_0, a, b) = (q_0, \epsilon)$ pop.
$\delta(q_0, b, a) = (q_0, \epsilon)$
} Dissimilar pop.

$\delta(q_0, \epsilon, \underline{a}) = (q_1, \underline{a})$ top.

(ϵ, ϵ) (q_0, ϵ)

String is Rejected, as there is no acceptance Config.

<< >>

Menu

o Show that aab is accepted
(Instantaneous Description)

$(q_0, aab\epsilon, Z_0)$ $\delta(q_0, a, Z_0)$

$\vdash (q_0, ab\epsilon, aZ_0)$ $\delta(q_0, a, a) = (q_0, aa)$

$\vdash (q_0, b\epsilon, aaZ_0)$ $\delta(q_0, b, a) = (q_1, \epsilon)$

$\vdash (q_0, \epsilon, aZ_0)$ $\delta(q_0, \epsilon, a) = (q_1, a)$

To show that the string bab is rejected.

$(q_0, bab\epsilon, Z_0)$ $\delta(q_0, b, Z_0) = (q_0, bZ_0)$

$\vdash (q_0, ab\epsilon, bZ_0)$ $\delta(q_0, a, b) = ($

$\vdash (q_0, b\epsilon, Z_0)$ $\delta(q_0, b, Z_0) = (q_0, bZ_0)$

$\vdash (q_0, \epsilon, bZ_0)$ $\delta(q_0, \epsilon, b)$

<<

>>

Q. Design PDA for

$L = \{ \omega : \omega \in \{a, b\}^* \text{ with } n_b(\omega) > n_a(\omega) \}$

Solution: $\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, b, Z_0) = (q_0, bZ_0)$ Push on to Z_0

$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, b) = (q_0, bb)$ Push a onto a & b onto b. (Similar)

$\delta(q_0, b, a) = (q_0, \epsilon)$
$\delta(q_0, a, b) = (q_0, \epsilon)$ Dissimilar (pop)

$b > a$ $\delta(q_0, b, Z_0) = (q_1, b)$

$(q_0, abbb\epsilon, Z_0)$

$\vdash (q_0, bbb\epsilon, aZ_0)$

$(q_0, bb\epsilon, Z_0)$

no. of b's is more than a's

Date: 21.12.2020; Pushdown Automata:

Q. Design PDA for $L = \{wcw^R : w \in \{a,b\}^*\}$ and w^R is reverse of w.

Solution: (1). $L = \{c, aca, abcba, bacab \ldots\}$

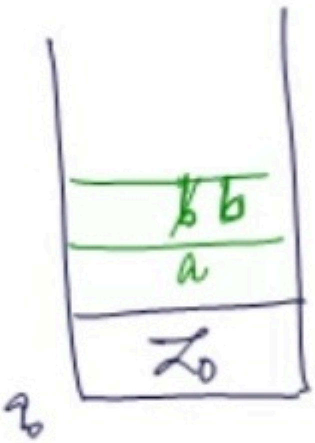
$\boxed{\epsilon c \epsilon = c}$

(2) To write the logic.

$\overset{w}{ab}c\overset{w^R}{ba}\epsilon$

$\uparrow$

q_0



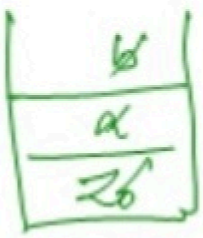
q_0

$cba\epsilon$

$\uparrow$

q_0

$\boxed{\delta(q_0, c, b) = (q_0, b) \quad \text{Replace transition}}$



$ba\epsilon$

$\uparrow$

empty

Transitions: abaabCbaaba $\frac{w}{a/b}$

abcbaϵ

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, b, Z_0) = (q_0, bZ_0)$

Whether it is a or b push it onto the top of the stack Z_0.

$\delta(q_0, a, b) = (q_0, ab)$
$\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, b) = (q_0, bb)$

All possibilities to be considered.

cbaϵ

q_0

$\delta(q_0, c, b) = (q_1, b)$
$\delta(q_0, c, a) = (q_1, a)$
$\delta(q_0, c, Z_0) = (q_1, Z_0)$

Replace transition.

baϵ

q_1 For each b there will be same symbol on top of the stack.

Transitions: abaabCbaaba $\quad w, a/b$

abcbaϵ

push z_0

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$ ①
$\delta(q_0, b, Z_0) = (q_0, bZ_0)$ ②

Whether it is a or b push it onto the top of the stack Z_0.

Push

$\delta(q_0, a, b) = (q_0, ab)$ ③
$\delta(q_0, b, a) = (q_0, ba)$ ④
$\delta(q_0, a, a) = (q_0, aa)$ ⑤
$\delta(q_0, b, b) = (q_0, bb)$ ⑥

All possibilities to be considered.

cbaϵ

q_0

$\delta(q_0, c, b) = (q_1, b)$ ⑦
$\delta(q_0, c, a) = (q_1, a)$ ⑧
$\delta(q_0, c, Z_0) = (q_1, Z_0)$ ⑨

Replace transition.

baϵ

q_1 For each b there will be same symbol on top of the stack.

Pop

$\delta(q_1, b, b) = (q_1, \epsilon)$ ⑩
$\delta(q_1, a, a) = (q_1, \epsilon)$ ⑪

$\delta(q_1, \epsilon, Z_0) = (q_2, \epsilon, \epsilon)$ ⑫

Show that the String $\underbrace{abaab}_{w}\underbrace{c}_{c}\underbrace{baaba}_{w^R}$

Instantaneous Description (ID)

$(q_0, abaabcbaaba\epsilon, Z_0)$ — ①

$\vdash (q_0, baabcbaaba\epsilon, aZ_0)$ — ②

$\vdash (q_0, aabcbaaba\epsilon, baZ_0)$ — ③

$\vdash (q_0, abcbaaba\epsilon, abaZ_0)$ — ⑤

$\vdash (q_0, bcbaaba\epsilon, aabaZ_0)$ — ④

Push

$\vdash (q_0, cbaaba\epsilon, baabaZ_0)$ ⑥ — ⑦

$\vdash (q_1, baaba\epsilon, baabaZ_0)$ —

enu

Show that the String $\underbrace{abaab}_{w}\, \underset{c}{c}\, \underbrace{baaba}_{w^R}$

Instantaneous Description (ID)

$(q_0, abaabcbaaba\epsilon, Z_0)$ — ①

$\vdash (q_0, baabcbaaba\epsilon, aZ_0)$ — ②

$\vdash (q_0, aabcbaaba\epsilon, baZ_0)$ — ③

$\vdash (q_0, abcbaaba\epsilon, abaZ_0)$ — ⑤

$\vdash (q_0, bcbaaba\epsilon, aabaZ_0)$ — ④

Push

$\vdash (q_0, cbaaba\epsilon, baabaZ_0)$ R — ⑦

$\vdash (q_1, baaba\epsilon, baabaZ_0)$ — ⑩

$\vdash (q_1, aaba\epsilon, aabaZ_0)$ — ⑪

$\vdash (q_1, aba\epsilon, abaZ_0)$ — ⑪

$\vdash (q_1, ba\epsilon, baZ_0)$ — ⑩

$\vdash (q_1, a\epsilon, aZ_0)$ — ⑪

$\vdash (q_1, \epsilon, Z_0)$ —

$\vdash (q_2, \epsilon, \epsilon)$ — empty ⑫

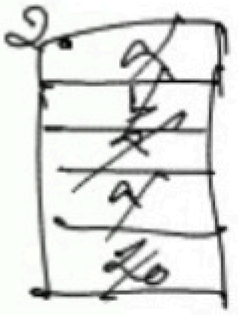
<< >>

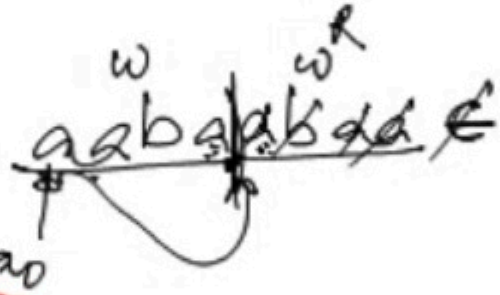
Menu

Q. ① Design PDA for $L = \{ww^R : w \in \{a,b\}^*\}$ & w^R is reverse of w.

Soln:

1. $L = \{\epsilon, \underset{w\,w^R}{aa}, \underset{w\,w^R}{bb}, \underset{w}{(ab)}\underset{w^R}{(ba)}, \underset{w}{(aba)}\underset{w^R}{(aba)} \ldots\}$





①

$\left.\begin{array}{l}\delta(q_0, a, z_0) = (q_0, az_0) \\ \delta(q_0, b, z_0) = (q_0, bz_0)\end{array}\right\}$ On to z_0 ②

NPDA

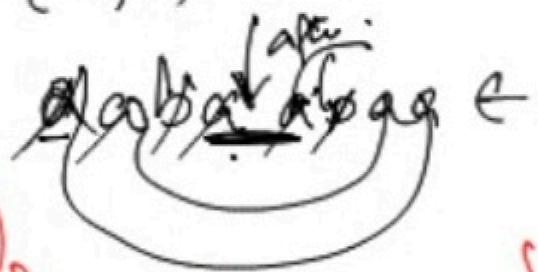
③ before mid / after

$\delta(q_0, \underline{a}, \underline{a}) = (q_0, \underline{aa}) / (q_0, \epsilon)$ } on to a or b.

$\delta(q_0, b, b) = (q_0, bb) / (q_0, \epsilon)$

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

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



$\delta(q_0, \epsilon, z_0) \rightarrow (q_1, \epsilon, \epsilon)$

<< >>

To show that the string

$a\underbrace{aba}_{w}\underbrace{aba a}_{w^R}\epsilon$ is accepted

$(q_0, aabaabaa\epsilon, Z_0)$ — ①

$\vdash (q_0, abaabaa\epsilon, aZ_0)$ — ③

$\vdash (q_0, baabaa\epsilon, aaZ_0)$ —

$\vdash (q_0, aabaa\epsilon, baaZ_0)$ — ⑤

$\vdash (q_0, abaa\epsilon, abaaZ_0)$ — ⑦

$\vdash (q_0, baa\epsilon, baaZ_0)$ — ⑧

$\vdash (q_0, aa\epsilon, aaZ_0)$ — ⑦

$\vdash (q_0, a\epsilon, aZ_0)$ — ⑦

$\vdash (q_0, \epsilon, Z_0)$ — q_1, ϵ, ϵ

Date: 22.12.2020

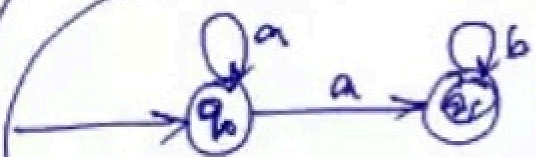
Q. Design PDA for

$L = \{ ww^R : w \in \{a,b\}^* \text{ where } w^R \text{ is reverse of } w \}$

$\delta(q_0, a, a) = (q_0, aa)$ → before mid of the string

OR

$\delta(q_0, a, a) = (q_0, \epsilon)$ → after mid



non-deterministic

$(q_0, a) = q_0$ ✓

$\quad\quad\quad\; q_1$ ✓

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

$\delta(q_0, b, Z_0) = (q_0, bZ_0)$

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

$\delta(q_0, b, b) = (q_0, bb)$

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

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

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

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

$w = \overset{w}{aa}\overset{w^R}{aa}$

$(q_0, aaaa\epsilon, Z_0)$

Conclusion:

Since $\delta(q_0, a, a)$ has two options

$$\delta(q_0, a, a) = (q_0, aa) \quad \text{I}$$

$$\delta(q_0, a, a) = (q_0, \epsilon) \quad \text{II}$$

the PDA is non-Deterministic

Verify whether the following PDA is deterministic.

$L = \{ w : w \in \{a, b\}^* \text{ with } n_a(w) = n_b(w) \}$

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

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

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

$$\delta(q_0, b, b) = (q_0, bb)$$

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

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

$$\delta(q_0, \epsilon, z_0) = (q_1, \epsilon)$$

Deterministic PDA:

A PDA $M = (Q, \Sigma, \delta, q_0, F, \Gamma, Z_0)$ is said to be deterministic, if it satisfies the following conditions:

1. $\delta(q, a, Z)$ has only one option.
2. If $\delta(q, \epsilon, Z)$ is present, then $\delta(q, a, Z)$ must not.

<<

>>

Ex: Whether the PDA designed for

$L = \{ ww^R : w \in (a+b)^*$ and w^R is rev of $w \}$

$Z)$ has only one option.

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

$\delta(q_0, b, Z_0) = (q_0, bZ_0)$

after mid

mid

$\delta(q_0, a, a) = (q_0, aa) \,/\, (q_0, \epsilon)$ $\delta(q, \epsilon, Z)$ is P

$\delta(q_0, b, b) = (q_0, bb) \,/\, (q_0, \epsilon)$ $\delta(q, a, Z)$ must not

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

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

menu

Conclusion:

Since $\delta(q_0, a, a)$ has two options

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

$\delta(q_0, a, a) = (q_0, \epsilon)$ II

the PDA is non-Deterministic

Verify whether the following PDA is deterministic.

$L = \{ w : w \in \{a, b\}^* \text{ with } n_a(w) = n_b(w) \}$

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

$\delta(q_0, b, Z_0) = (q_0, bZ_0)$

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

$\delta(q_0, b, b) = (q_0, bb)$

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

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

PDA to be deterministic

1\.
$\delta(q, a,$

Satisfied

2\. If $\delta($:

$\delta(q_0, \epsilon, Z_0)$ is P

a

<< >>

Conclusion:

Since $\delta(q_0, a, a)$ has two options

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

$\delta(q_0, a, a) = (q_0, \epsilon)$ $a, z)$ has only one option

the PDA is non-deterministic

Verify whether the following PDA is deterministic.

$\delta(q, \ldots)$ is P

$L = \{ w : w \in \{a,b\}^* \text{ with } n_a(w) = n_b(w) \}$

$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, b, z_0) = (q_0, bz_0)$

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

$\delta(q_0, b, b) = (q_0, bb)$

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

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

PDA to be deterministic

1. $\delta(q, a,$

Satisfied

2. If $\delta($

$\delta(q_0, \epsilon, z_0)$ is P

q_0, a, z_0

menu

Q. Verify whether the following PDA is deterministic.

$L = \{a^n b^n : n \geq 0\}$

NDPA

PDA to be deterministic

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

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

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

$\delta(q_0, \epsilon, Z_0) = (q_1, \epsilon, \epsilon)$

(1) $\delta(q, a, Z)$ has only one option

(2) If $\delta(q, \epsilon, Z)$ is P
$\delta(q, a, Z)$ must A

(1) No. 1 is TRUE / satisfied

(2) If $\delta(q_0, \epsilon, Z_0)$ is P
then $\delta(q_0, a, Z_0)$ must A
✓ not satisfied

<<

>>

~~RE~~ FA → RL → RG.

PDA → CFL → CFG

Generated by CFG → PDA.

$$\begin{tikzcd} \text{CFG} \arrow[r, "\text{generates}"] \arrow[rr, bend right, "\text{relationship}"'] & \text{CFLs} \arrow[r, "\text{accepted}"] & \text{PDA.} \end{tikzcd}$$

For every CFG there exist an equivalent PDA.

Production ⟷ Transitions

For a given CFG we can write its equivalent PDA.

$$\begin{tikzcd} \text{CFG} \arrow[r] \arrow[rr, bend right] & \text{CFL} \arrow[r] & \text{PDA} \end{tikzcd}$$

CFG to PDA:

Procedure:

Input: Given CFG

Output: PDA

GNF:

$A \to a\alpha$

$A \in V$

$\alpha \in V^*$

$a \in T$

$S \to aA$

$A \to b$

Method:

① Express the given CFG in GNF

② In the state q_0, without consuming any input, push the start symbol S on to the top of the stack Z_0. Change state q_1

$$\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0) \quad — ①$$

③ For each prodn

$$A \to a\alpha$$

$$\delta(q_1, a, A) = (q_1, \alpha)$$

④ Finally, in the state q_1, without consuming any input, change the state to q_f, which is an accepting state

enu

Consider the following CFG

$S \to aABC$

$A \to aB \mid a$

$B \to bA \mid b$

$C \to a$.

Obtain corresponding PDA.

δ

$B \to b$ ⑥

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

$C \to a$

$\delta(q_1, a, C) = (q_1, \epsilon)$ ⑦

Sol: ① Express the CFG in GNF.

The CFG is in GNF.

② For each production of the form $A \to a\alpha$ write $\delta($

α — take q_0, without consuming any input symbol, push S of CFG to Z_0. and change the state to q_1

$\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$ ①

③ $S \to aABC$

$A \to aB$

$A \to a$

$B \to bA$ ⑤

$\delta(q_1, a, S) = (q_1, ABC)$ ②

$\delta(q_1, a, A) = (q_1, B)$ ③

$\delta(q_1, a, A) = (q_1, \epsilon)$ ④

$\delta(q_1, b, B) = (q_1, A)$ ⑤

<<

>>

CFG to PDA:

Procedure:

Input: Given CFG

Output: PDA

GNF:

$A \to a\alpha$

$A \in V$

$\alpha \in V^*$

$a \in T$

$S \to aA$

$A \to b$

Method:

(1) Express the given CFG in

$\delta(q_f, \epsilon, Z_0)$

~~consuming~~ (q_f, Z_0)

(2) In the state q_0, without consuming input, push the start symbol S ~~to the top of the stack Z_0. change state q_1~~

$\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$ — (1)

(2) In the S

(3) For each prodn

$A \to a\alpha$

$\delta(q_1, a, A) = (q_1, \alpha)$

(4) Finally, in the state q_1, without (q_1, ϵ) any input, change its state to q_f.

which is an accepting state

Q. $S \to aABC$

$A \to aB \mid a$

$B \to bA \mid b$

$C \to a$

CFG to PDA

① CFG is in GNF ✓

② $(q_0, \epsilon, Z_0) = (q_1, SZ_0)$ ①

$A \to a\alpha$	$\delta(q_1, a, A) = (q_1, \alpha)$
$S \to aABC$	$\delta(q_1, a, S) = (q_1, ABC)$
$A \to aB$	$\delta(q_1, a, A) = (q_1, B)$
$A \to a\epsilon$	$\delta(q_1, a, A) = (q_1, \epsilon)$
$B \to bA$	$\delta(q_1, b, B) = (q_1, A)$
$B \to b$	$\delta(q_1, b, B) = (q_1, \epsilon)$
$C \to a$	$\delta(q_1, a, C) = (q_1, \epsilon)$
$\delta(q_1, \epsilon, Z_0) =$	(q_f, Z_0)

Q. To write TM for

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

Soln:

$L = \{ \cancel{aab}\ ab,\ aabb,\ aaabbb \ldots \}$

$W = aaabbbB$

… B	B	a	a	a	b	b	b	B	B	…

q_0

Ex aaaaabbbbbB

$q_2 \xleftarrow{L}$ bb

a a

… X X a a a Y Y b b b B .

$q_1 \xrightarrow{R}$

XXXXXYYYYYB

X-a

Y-b

$Q \times \Sigma \xrightarrow{} Q \times \Gamma \times (L/R)$

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

Transition of TM depends on

1) Current State :

2) Next input Symbol.

After apply-

Changing the State

Replacement i/p by other

Move towards L/R.

Q. Design TM for $L = \{0^n 1^n : n \geq 1\}$.

<u>Solution:</u> I $L = \{01, 0011, 0^3 1^3 \dots\}$

II To write the logic $w = 000111B$

$X\,0\,0\,Y\,1\,1\,B$

$\uparrow q_0$

$X - 0$
$Y - 1$

Replace 0 by X and 1 by Y till it reaches B. If no. of X are matching with no. of Y's on the tape the string is accepted

$\delta(q_0, 0) = (q_1, X, R)$ ——— (1)

<<

>>

$X\,X\,0\,Y\,1\,1\,B$

$\uparrow q_1 \ \xrightarrow{R}$

$\delta(q_1, 0) = (q_1, 0, R)$ ——— (2)

$X\,X\,0\,Y\,1\,1\,B$

$\uparrow q_1$

$\delta(q_1, Y) = (q_1, Y, R)$ ——— (3)

$X\,X\,0\,Y\,1\,1\,B$

$\uparrow q_1$

$\cancel{0}$ X X 0 Y 1 1 B

q_1

$\delta(q_1, 1) = (q_2, Y, L)$ — 4

$\xleftarrow{L}$ q_2

X X 0 Y Y 1 B

XXX YYY B

$\delta(q_2, Y) = (q_2, Y, L)$ — 5

$\delta(q_2, 0) = (q_2, 0, L)$ — 6

X X 0 Y Y 1 B

q_2

$\delta(q_2, X) = (q_0, X, R)$ — 7

X X 0 Y Y 1 B

q_0

(apply these seven branch ga.

XXX YY$\cancel{Y}$ B -

X X 0 Y Y 1 B

$$\boxed{\delta(q_0, 0) = (q_1, X, R) \quad - \quad 1}$$

all Y. transitions

X X X Y Y Y B

q_0

$$\delta(q_0, Y) = (q_3, Y, R) \quad - \quad 8$$

X X X Y Y Y B

q_3

$$\delta(q_3, Y) = (q_3, Y, R) \quad - \quad 9$$

X X X Y Y Y B 1

q_3

$$\boxed{\delta(q_3, B) = (q_4, B, R) \quad - \quad 10}$$

Q. Design TM for $L = \{0^n 1^n : n \geq 1\}$.

Solution:

I $L = \{01, 0011, 0^3 1^3 \ldots\}$

II To write the logic $w = 000111B$

$X\,0\,0\,Y\,1\,1\,B$

$\uparrow q_0$

$X - 0$
$Y - 1$

Replace 0 by X and 1 by Y till it reaches B. If no. of X are matching with no. of Y's on the tape the string is accepted

$\delta(q_0, 0) = (q_1, X, R)$ —— ①

$X\,X\,0\,Y\,1\,1\,B$

$\uparrow q_1 \; \xrightarrow{R}$

$\delta(q_1, 0) = (q_1, 0, R)$ —— ②

$X\,X\,0\,Y\,1\,1\,B$

$\uparrow q_1$

$\delta(q_1, Y) = (q_1, Y, R)$ —— ③

$X\,X\,0\,Y\,1\,1\,B$

$\uparrow q_1$

XX0YY1B

$$\boxed{\delta(q_0, 0) = (q_1, X, R)} \quad \text{— } (1)$$

all Y transitions

XXXYYYB

q_0

$$\delta(q_0, Y) = (q_3, Y, R) \quad \text{— } 8$$

XXXYYYB

q_3

$$\delta(q_3, Y) = (q_3, Y, R) \quad \text{— } (9)$$

XXXYYYB1

q_3

$$\boxed{\delta(q_3, B) = (q_4, B, R)} \quad \text{— } (10)$$

$\delta(q_0, a) = (\overset{\text{next}}{q_1}, \overset{\text{Repl.}}{x}, R)$ d.

Turing Machine:

Q. Design TM for $L = \{0^n 1^n : n \geq 1\}$

Soln: 1. $L = \{01, 0011, 0^3 1^3 \ldots\}$ 0's.

2. Replace 0 by X and 1 by Y & move (Same time later)

$X X 0 0 0 Y Y 1 1 1 B$

$\uparrow q_0$

$\delta(q_0, 0) = (q_1, X, R)$ — (1)

Now the read write head is moving towards R to reach right most 1. In this process the read-write head will encounter 0's & Y's.

Replace 0 by 0 & 1 by 1 be there in same state and move towards R

$\delta(q_1, 0) = (q_1, 0, R)$ — (2)

$\delta(q_1, Y) = (q_1, Y, R)$ — (3)

Now machine is in q_1 R/W head is pointing to

$X X X 0 0 Y Y 1 1 1 B$

$\uparrow q_1$

Now left most 1 is reached. Replace 1 by Y and move towards L. Change state.

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

$X X X 0 0 \overleftarrow{Y Y Y} 1 1 B$

Now the R/W head is moving towards left to reach left most 0.

Left most 0 is one which is to the right of right most X.. hence it has to reach X first.

In this process, the machine, in state q_2 it will encounter with Y's & 0's

Replace Y by Y and 0 by 0 & move towards

$$\xleftarrow{B \quad q_2}$$

$$X\,X\,0\,0\,Y\,Y\,Y\,1\,1\,B$$

$$\left.\begin{array}{l}\delta(q_2, Y) = (q_2, Y, L) \\ \delta(q_2, 0) = (q_2, 0, L)\end{array}\right\} \begin{array}{l}— 5 \\ — 6\end{array}$$

If we apply these two transitions repeatedly R/w will reach right most X.

$$X\,X\,\overset{q_2}{X}\,\underline{0\,0}\,Y\,\underline{Y}\,Y\,1\,1\,B$$

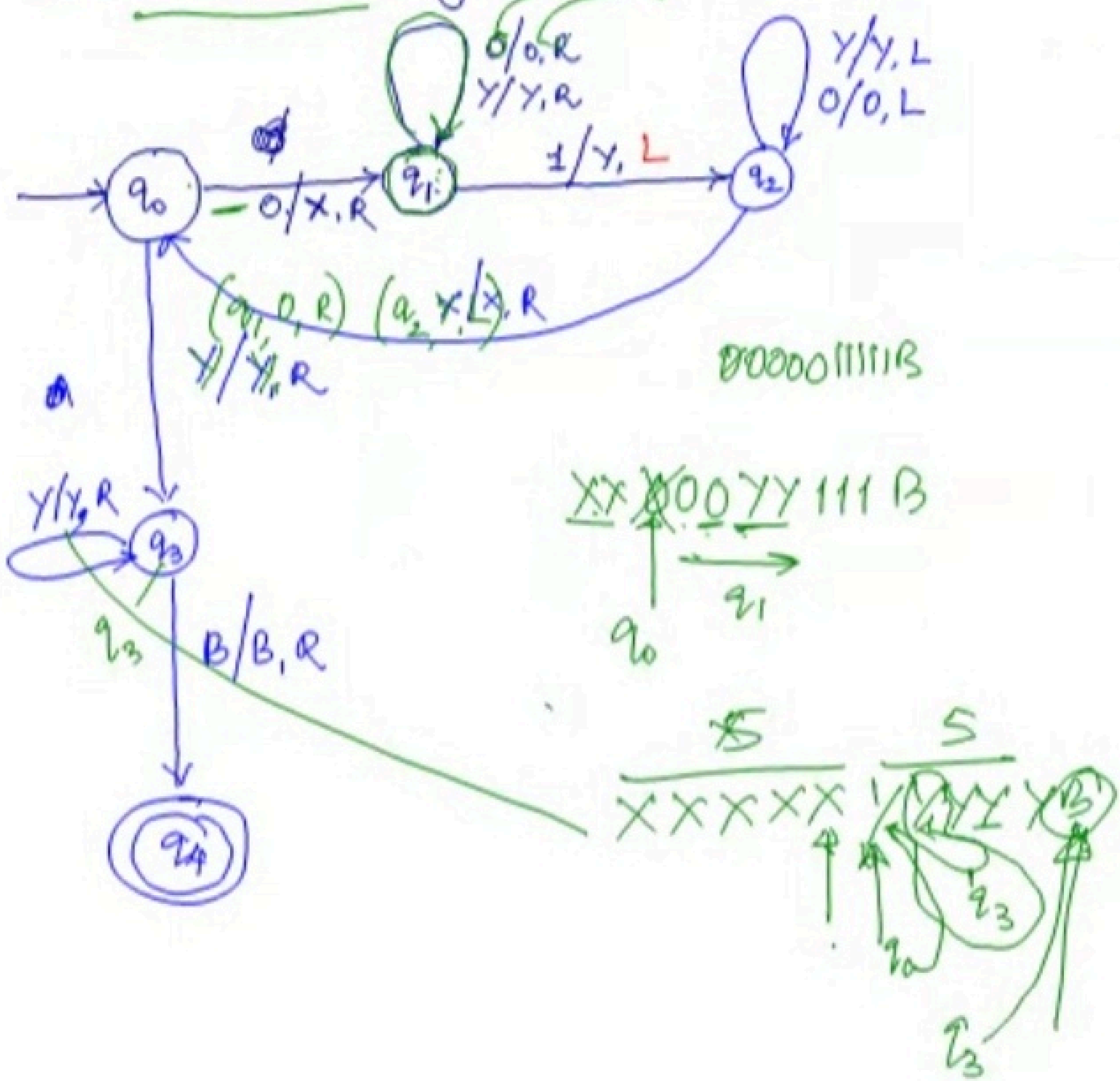
$$\delta(q_2, X) = (q_0, X, R) \; - 7$$

$$X\,X\,X\,\underset{\uparrow q_0}{0}\,0\,Y\,Y\,Y\,1\,1\,B$$

After apply the above 7 transitions for two iterations

$$X\,X\,X\,X\,X\,Y\,Y\,Y\,Y\,Y\,B$$

Transition diagram



Menu

Transition Table

	0	1	X	Y	B
q_0	(q_1, X, R)	–	–	(q_3, Y, R)	–
q_1	$(q_1, 0, R$	(q_2, Y, L)	–	(q_1, Y, R)	–
q_2					
q_3	–	–	–	$(q_3 Y, R$	(q_4, B, R)

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

$Q = \{q_0, q_1, q_2, q_3\}$

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

δ : TF —

$F = \{q_4\}$ final state

$\Gamma = \{0, 1, X, Y)$

B Special symbol

<<

>>

3. To write the transitions required.

X aaa Y bbb Z ccc B . .

$\delta(q_0, a) = (q_1, X, R)$ — ①

X X aa Y bbb Z ccc B .

$\uparrow q_1$

$\delta(q_1, a) = (q_1, a, R)$ — ②

$\delta(q_1, Y) = (q_1, Y, R)$ — ③

$\delta(q_1, b) = (q_1, b, R)$ — ④

If we apply the transitions ②, ③ & ④ for required number of times the read/write head will reach the left most c.

X X aa Y bbb Z ccc B

$\uparrow q_1$

Now the machine is in the state q_1 and the required position (left most c) is reached, hence replace c by Z and change the state to q_2, move leftwards L.

$\delta(q_1, c) = (q_2, Z, L)$ — ⑤

enu

<<

>>

Present State:

XXXaaYYYbb ZZccc B

$\uparrow q_2$

Presently the read/write is pointing b and moving towards RIGHT to replace the leftmost c by Z.

During this process of reaching leftmost c it will encounter b's & Z's.

Replace b by b & Z by Z without changing the state & move towards RIGHT.

Transitions $\delta(q_2, b) = (q_2, b, R)$ — ⑤

$\delta(q_2, Z) = (q_2, Z, R)$ — ⑥

After applying transition 5 & 6, for required no. of times, the read/write head will reach leftmost c.

XXXaaYYYbbZZccc B

$\uparrow$

Now replace the left most c by Z

and change the State to q_3.

Transition

$$\delta(q_2, c) = (q_3, Z, L) - (7)$$

Status of Strip/tape

$$\xleftarrow{\hspace{6em} q_3}$$

X X X a a Y Y Y b b Z Z Z c c B

Now, the read/write head is moving towards ~~RIGHT~~ LEFT to reach the ~~left~~ right most X. The required input symbol 'a' is to the right of right most X.

In the state q_3 the read/write head will encounter Z's, b's, Y's & a's before reaching right most X. Replace all these by themselves & move towards left without changing the state.

Transitions:

$$\delta(q_3, Z) = (q_3, Z, L) - (8)$$
$$\delta(q_3, b) = (q_3, b, L) - (9)$$
$$\delta(q_3, Y) = (q_3, Y, L) - (10)$$
$$\delta(q_3, a) = (q_3, a, L) - (11)$$

The transition for the same will be

$$\delta(q_0, a) = (q_1, X, R) \quad — ①$$

Status

$$XXXaaYYbbbZZcccB.$$
$$\uparrow q_1$$

Now in the state q_1, the r/w head is moving towards RIGHT to reach the left most b. In this process the r/w head will encounter a's & Y's. Replace a by a and Y by Y be there in q_1 and move towards R.

Transitions
$$\delta(q_1, a) = (q_1, a, R) \quad — ②$$
$$\delta(q_1, Y) = (q_1, Y, R) \quad — ③$$

After applying these two transitions for required number of times, the r/w head will reach the leftmost b.

Replace b by Y and change the state to q_2

$$\delta(q_1, b) = (q_2, Y, R) \quad — ④.$$

Status of String / tape:

XXX aa YYY bb ZZZ cc B

$\uparrow$

q_3

Now to reach leftmost a, apply

$$\delta(q_3, X) = (q_0, X, R) \quad - (12)$$

after applying (12) the status

XXX aa YYY bb ZZZ cc B

$\uparrow$

q_0

After applying the (1-12) twelve transitions for required no. of times

XXXXX YYYYY ZZZZZ B

$\uparrow$

In the state q_0, if a's is encountered repeat the previous twelve transitions & if the input symbol encountered is Y

$$\delta(q_0, Y) = (q_4, Y, R) \quad - (13)$$

Current State:

X X X X X $\overset{a/a,R}{\underset{Y/Y,R}{Y}}$ Y Y Y Z Z Z Z Z B B B...

$\uparrow$

q_4

Now the read/write is moving towards right to search the leftmost B.

$$\delta(q_4, Y) = (q_4, Y, R) \quad - (14)$$

If we apply (14) for required no. of times, the r/w head will search left B, which is accepting State (q_5)

$$\boxed{\delta(q_4, B) = (\underline{q_5,} B, R) \quad - (15)}$$

Accepting transition.

After (15)th transition

X X X X X Y Y Y Y Y Z Z Z Z Z B B

$\uparrow$

q_5

Accept

Transition Diagram: