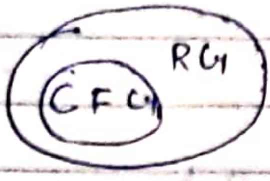


Regular Grammar: [$G = (N, \Sigma, P, S)$]

$S \rightarrow Aa$ (Left linear grammar)
$S \rightarrow bA$ (Right linear grammar)

$S \rightarrow aAb \rightarrow$ Not possible X

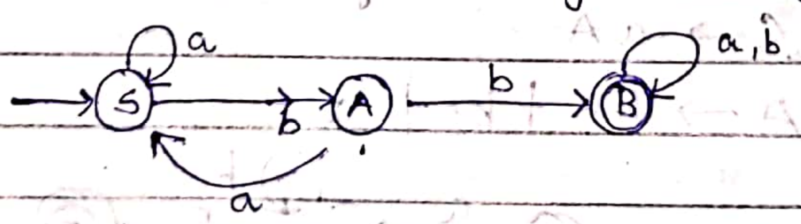
→ Regular grammar has two types:
(1) Left linear ($S \rightarrow Aabc$)
(2) Right linear ($S \rightarrow abB$)

Convert.
→ Right linear grammar to Left linear grammar:

Eg: $S \rightarrow aS \mid bA$
$A \rightarrow aS \mid bB$
$B \rightarrow aB \mid bB \mid \epsilon$

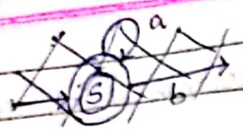
we can convert Right LG to Left LG only if all the states are in Right LG.

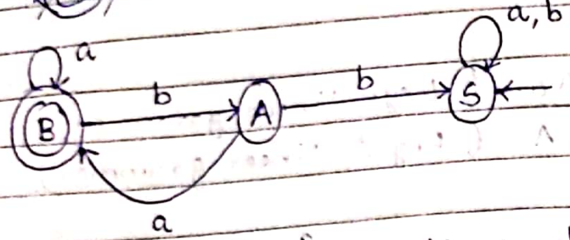
Step:1 → Design FA for given Right LG:



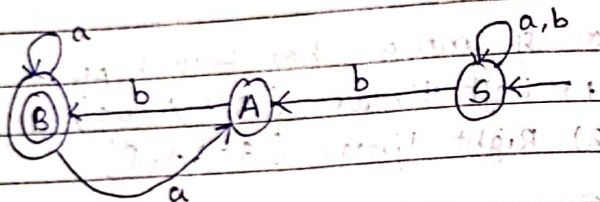
Any state which shows ϵ is the final state.

Step:2 Interchange the initial and final state.





Step: 3 Reverse the directions of transition (arrow head)



Step: 4 Left linear:

~~S → aS~~ S → Sa | Sb | Ab

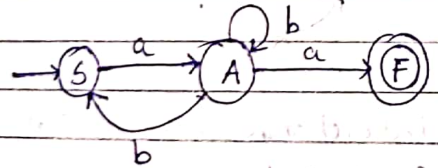
A → Bb

B → Aa | Ba | ε

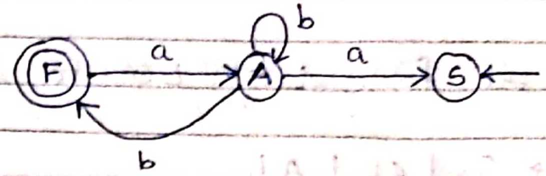
Eg: Right LG to Left LG:

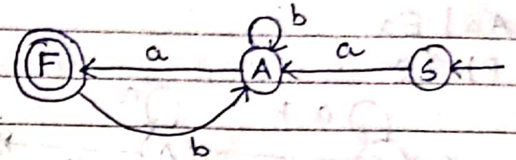
S → aA

A → bA | bS | a

Step: 1 

Note: F is the Final state.

Step: 2 

Step: 3 

Step: 4 Left linear:

S → Sa

A → Ab | Fa

F → Ab

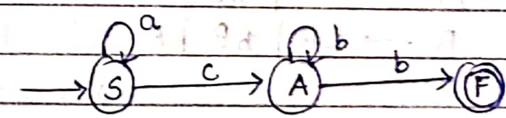
Eg: Right LG to Left LG:

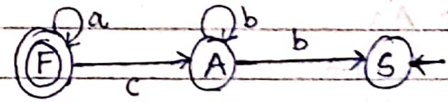
S → aS

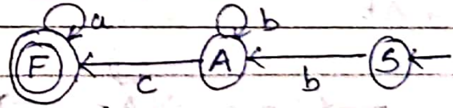
S → cA

A → bA

A → b

Step: 1 

Step: 2 

Step: 3 

Step: 4 Left linear ⇒

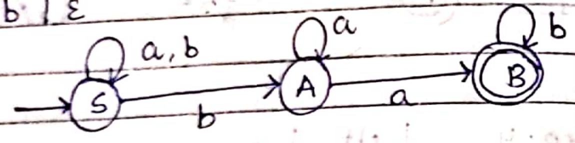
S → Ab

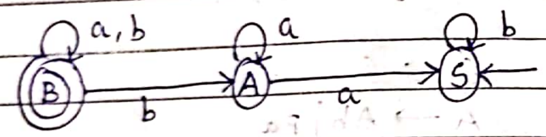
A → Ab | Fc

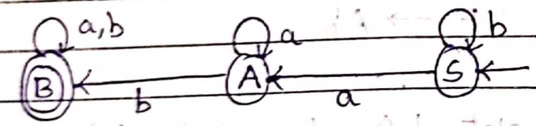
F → Fa

Q. Convert foll. Left LG to Right LG.

Eg: $S \to Sa \mid Sb \mid Ab$
$A \to Aa \mid Ba$
$B \to Bb \mid \varepsilon$

Step: 1 

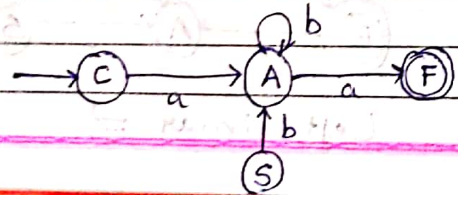
Step: 2 

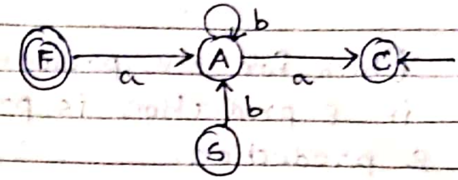
Step: 3 

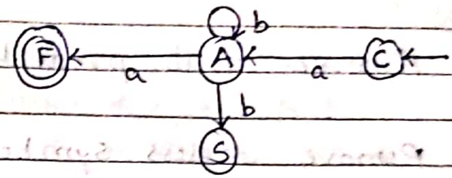
Step: 4 Right LG:

$S \to bS \mid aA$
$A \to aA \mid bB$
$B \to aB \mid bB \mid \varepsilon$

Eg: $C \to Aa$
$A \to Ab \mid a$
$S \to Ab$

Step: 1 

Step: 2 

Step: 3 

Step: 4 Right LG:

$C \to aA$
$A \to bS \mid aF \mid bA$

* Normal forms:

(1) Chomsky Normal form (CNF):
(Eg: $S \to AB$
$S \to a$
(Two non terminals or one terminal on right hand side.)

Q. Convert foll. CFG to chomsky Normal form (CNF):

Step: 1 Observe that starting symbol is appearing on RHS of any production. If starting symbol is appearing on RHS then add new production with new non-terminal, i.e. if S is starting symbol then $S' \to S$.

Step: 2 Check for the presence of ε production if ε production is present then remove ε production.

Step: 3 Remove unit production.

Step: 4 Remove useless symbols.

Step: 5 If more than 2 non terminals are present on RHS then make them 2

Eg: $S \to ASA$ (more than 2)

$\therefore$ Let $X \to AS$

$\therefore S \to ASA$

$\therefore S \to XA \quad (\because X \to AS)$

Eg: $S \to ASABD$

$S \to XABD \quad (\because X \to AS)$

$S \to X_1BD \quad (\because X_1 \to XA)$

$S \to X_2D \quad (\because X_2 \to X_1B)$

Step: 6 Observe that there ~~are~~ is single non-terminal along with single terminal symbol then convert it into 2 non-terminals.

Eg: $S \to aB$

Let $T \to a$

$\therefore S \to TB \quad (\because T \to a)$

Q. $S \to ASA \mid aB$

$A \to B \mid S$

$B \to b \mid \varepsilon$

Step: 1 $S' \to S$

$S \to ASA \mid aB$

$B \to b \mid \varepsilon$

$A \to B \mid S$

Step: 2 Nullable variables $\to A, B$

$B \to \varepsilon$ and as $A \to B,\ A \to \varepsilon$

$\therefore S' \to S$

$S \to ASA \mid AS \mid SA \mid S \mid aB \mid a$

$A \to B \mid S$

$B \to b$

Step: 3 $S' \to ASA \mid AS \mid SA \mid S \mid aB \mid a$

$S \to ASA \mid AS \mid SA \mid S \mid aB \mid a$

1st iteration $A \to b \mid ASA \mid AS \mid SA \mid S \mid aB \mid a$

$B \to b$

$\Rightarrow S' \to ASA \mid AS \mid SA \mid aB \mid a$

$S \to ASA \mid AS \mid SA \mid aB \mid a$

2nd iteration $A \to b \mid ASA \mid AS \mid SA \mid aB \mid a$

$B \to b$

Step: 4 No useless symbols.

Step: 5 $S' \rightarrow ASA$

Let, $X \rightarrow AS$

$\therefore S' \rightarrow XA$

So, rewrite following productions,

$S \rightarrow ASA$, $A \rightarrow ASA$

$\therefore S \rightarrow XA$, $A \rightarrow XA$

Now,

$S' \rightarrow XA \mid AS \mid SA \mid \underline{a}B \mid a$

$S \rightarrow XA \mid AS \mid SA \mid \underline{a}B \mid a$

$A \rightarrow b \mid XA \mid \cancel{AB}\ AS \mid SA \mid \underline{a}B \mid a$

$B \rightarrow b$

$X \rightarrow AS$

Step: 6 $S' \rightarrow aB$

Let, $X_1 \rightarrow a$

$\therefore S' \rightarrow X_1B$ $(\because X_1 \rightarrow a)$

Similarly,

$S \rightarrow aB$, $A \rightarrow aB$

$\therefore S \rightarrow X_1B$ $(\because X_1 \rightarrow a)$ $\therefore A \rightarrow X_1B$ $(\because X_1 \rightarrow a)$

Now,

$S' \rightarrow XA \mid AS \mid SA \mid X_1B \mid a$

$S \rightarrow XA \mid AS \mid SA \mid X_1B \mid a$

$A \rightarrow b \mid XA \mid AS \mid SA \mid X_1B \mid a$

$B \rightarrow b$, $X \rightarrow AS$, $X_1 \rightarrow a$

Hence, this is in CNF

Q. $A \rightarrow BAB$

$A \rightarrow B \mid \varepsilon$

$B \rightarrow 11 \mid \varepsilon$

Step: 1 $A' \rightarrow A$

$A \rightarrow BAB \mid B \mid \varepsilon$

$B \rightarrow 11 \mid \varepsilon$

Step: 2 Nullable variables $\rightarrow A', A, B$

$A' \rightarrow A$

$A \rightarrow BAB \mid BA \mid AB \mid BB$

(2) Greibach Normal Form (GNF) :

$S \rightarrow aABC$

~~non~~ terminals followed by non terminal symbols.

(Single ~~non~~ terminal and zero or more non terminal symbols)

Q. Convert foll? CFG to GNF :

Eg: $S \rightarrow AA \mid a$

$A \rightarrow SS \mid b$

Step : 1 S is replace with A_1

A is replace with A_2

$A_1 \rightarrow A_2A_2 \mid a$

$A_2 \rightarrow A_1A_1 \mid b$

Step : 2 $A_2 \rightarrow A_1A_1$

$A_2 \rightarrow A_2A_2A_1$ $(\because A_1 \rightarrow A_2A_2)$

aA_1 $(\because A_1 \rightarrow a)$

This is in GNF

If a non terminal symbol produces a production which starts with itself its called $\boxed{\text{left recursion}}$

(Eg: $T \rightarrow T*F$, $E \rightarrow E+T$)

$A_2 \rightarrow A_2 A_2 A_1$

Let $Z_2 \rightarrow A_2 A_1 \mid A_2 A_1 Z_2$

$\therefore \boxed{A_2 \rightarrow b \mid aA_1}$ are in GNF

We have, $A_2 \rightarrow A_2 A_2 A_1$

$\therefore A_2 \rightarrow bA_2A_1 \mid aA_1A_2A_1$

$\therefore \boxed{A_2 \rightarrow bZ_2 \mid aA_1Z_2}$ $(\because Z_2 \rightarrow A_2A_1)$

$\rightarrow \therefore A_2 \rightarrow b \mid aA_1 \mid bZ_2 \mid aA_1Z_2$

Now,

For $A_1 \rightarrow A_2A_2 \mid a$ → Its already in GNF.

$A_1 \rightarrow A_2A_2$

$\rightarrow \therefore A_1 \rightarrow bA_2 \mid aA_1A_2 \mid bZ_2A_2 \mid aA_1Z_2A_2 \mid a$

Final production list,

Grammar in GNF,

$A_1 \rightarrow bA_2 \mid aA_1A_2 \mid bZ_2A_2 \mid aA_1Z_2A_2 \mid a$

$A_2 \rightarrow b \mid aA_1 \mid bZ_2 \mid aA_1Z_2$

$Z_2 \rightarrow bA_1 \mid aA_1A_1 \mid bZ_2A_1 \mid aA_1Z_2A_1 \mid$
$bA_1Z_2 \mid aA_1A_1Z_2 \mid bZ_2A_1Z_2 \mid aA_1Z_2A_1Z_2$

Eg: Convert foll. CFG to GNF.

$S \rightarrow AB$

$A \rightarrow BS \mid b$

$B \rightarrow SA \mid a$

Step 1: S is replace with A_1

A is replace with A_2

B is replace with A_3

$A_1 \rightarrow A_2A_3$

$A_2 \rightarrow A_3A_1 \mid b$

$A_3 \rightarrow A_1A_2 \mid a$

Step 2: $A_3 \rightarrow A_1A_2$

$A_3 \rightarrow A_2A_3A_2$ $(\because A_1 \rightarrow A_2A_3)$

$A_3 \rightarrow A_3A_1A_3A_2$ $(\because A_2 \rightarrow A_3A_1)$

$\therefore A_3 \rightarrow A_3A_1A_3A_2 \mid bA_3A_2$ $(\because A_2 \rightarrow b)$

Now, $A_3 \rightarrow A_3A_1A_3A_2$

Let $Z_3 \rightarrow A_1A_3A_2 \mid A_1A_3A_2Z_3$

$\therefore A_3 \rightarrow \cancel{a} \; A_3Z_3$

$\boxed{A_3 \rightarrow a \mid bA_3A_2}$ are in GNF

$\therefore A_3 \rightarrow aZ_3 \mid bA_3A_2Z_3$

$\rightarrow \therefore A_3 \rightarrow a \mid bA_3A_2 \mid aZ_3 \mid bA_3A_2Z_3$

Now,

For $A_2 \rightarrow A_3A_1 \mid b$ → Its in GNF

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

Now, For $A_1 \rightarrow A_2A_3$

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

Final production list,

Grammar in GNF:

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

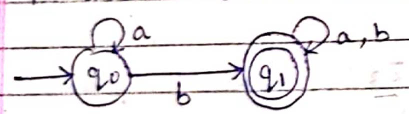
$A_2 \rightarrow aA_1 \mid bA_3A_2A_1 \mid aZ_3A_1 \mid bA_3A_2Z_3A_1 \mid b$

$A_3 \rightarrow a \mid bA_3A_2 \mid aZ_3 \mid bA_3A_2Z_3$

$Z_3 \rightarrow aA_1A_3A_3A_2 \mid bA_3A_2A_1A_3A_3A_2 \mid aZ_3A_1A_3A_3A_2 \mid bA_3A_2Z_3A_1A_3A_3A_2 \mid bA_3A_3A_2 \mid aA_1A_3A_3A_2Z_3 \mid bA_3A_2A_1A_3A_3A_2Z_3 \mid aZ_3A_1A_3A_3A_2Z_3 \mid bA_3A_2Z_3A_1A_3A_3A_2Z_3 \mid bA_3A_3A_2Z_3$

* Finite automata to Regular grammar:

Eg.



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

(RG) $G = (N, E, P, S)$

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

$Q = \{q_0, q_1\}$

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

$\delta(q_0, a) = q_0$

$\delta(q_0, b) = q_1$

$\delta(q_1, a) = q_1$

$\delta(q_1, b) = q_1$

$q_0 = q_0$

$F = \{q_1\}$

$\delta(q_0, a) = q_0$

$q_0 \rightarrow aq_0$

$q_0 \rightarrow bq_1, \; q_0 \rightarrow b$

$q_1 \rightarrow aq_1, \; q_1 \rightarrow a$

$q_1 \rightarrow bq_1, \; q_1 \rightarrow b$

q_0 will replace with A

q_1 will replace with B

$\therefore A \rightarrow aA$

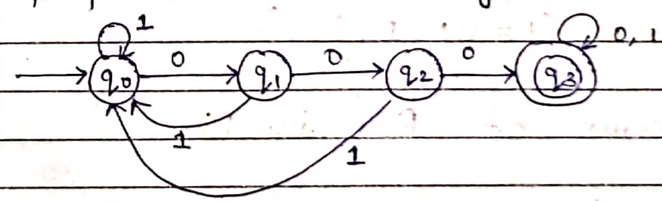
$A \rightarrow bB \mid b$

$B \rightarrow aB$

$B \rightarrow a$

$B \rightarrow bB \mid b$

Eg: Prepare a RG for given FA.



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

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

$\Sigma = \{0, 1\}$

$\delta(q_0, 0) = q_1$

$\delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_2$

$\delta(q_1, 1) = q_0$

$\delta(q_2, 0) = q_3$

$\delta(q_2, 1) = q_0$

$\delta(q_3, 0) = q_3$

$\delta(q_3, 1) = q_3$

$q_0 = q_0$

$F = \{q_3\}$

$q_0 \rightarrow 0q_1$ $\qquad$ $q_2 \rightarrow 0q_3$, $q_2 \rightarrow 0$

$q_0 \rightarrow 1q_0$ $\qquad$ $q_2 \rightarrow 1q_0$

$q_1 \rightarrow 0q_2$ $\qquad$ $q_3 \rightarrow 0q_2$, $q_3 \rightarrow 0$

$q_1 \rightarrow 1q_0$ $\qquad$ $q_3 \rightarrow 1q_3$, $q_3 \rightarrow 1$

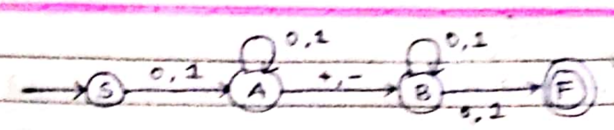
q_0 replace with A
q_1 replace with B
q_2 replace with C
q_3 replace with D

$\therefore$ $A \rightarrow 0B$ $\qquad$ $C \rightarrow 0D \mid 0$
$A \rightarrow 1A$ $\qquad$ $C \rightarrow 1A$
$B \rightarrow 0C$ $\qquad$ $D \rightarrow 0D \mid 0$
$B \rightarrow 1A$ $\qquad$ $D \rightarrow 1D \mid 1$

$\therefore$ 11 productions.
$G = (N, \Sigma, P, S)$

* Conversion of RG to FA.

$S \rightarrow 0A \mid 1A$ $\qquad$ (operators are considered as terminals)
$A \rightarrow 0A \mid 1A \mid +B \mid -B$
$B \rightarrow 0B \mid 1B \mid 0 \mid 1$



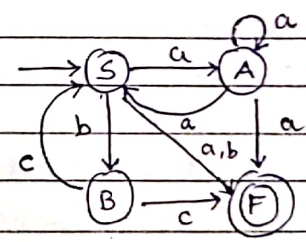
Assume F as a final state.

Eg: Construct FA for given grammar:

$S \rightarrow a \mid aA \mid bB \mid \varepsilon$
$A \rightarrow aA \mid aS$
$B \rightarrow cS \mid \varepsilon$

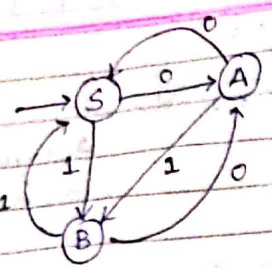
→ Here S and B are nullable variables.

$S \rightarrow a \mid aA \mid bB \mid b$
$A \rightarrow aA \mid aS \mid a$
$B \rightarrow cS \mid c$



We have added new state F as final state.

Eg: Convert foll. grammar to FA.
$S \rightarrow 0A \mid 1B$
$A \rightarrow 0S \mid 1B$
$B \rightarrow 0A \mid 1S$



→ If there is no new final state is formed (i.e. no need to add final state) then total no. of states is equal to the productions otherwise no. of states will be 1 more then productions.

- Left recursion:

$\boxed{A \to A\alpha \mid \beta}$ (General form) → Rule to identify left recursion.

Eg: $S \to SA \mid a$
$A \to b$
Here, $A = S$
$\alpha = A$
$\beta = a$

$\left.\boxed{\begin{array}{l} A \to \beta A' \\ A' \to \alpha A' \mid \varepsilon \end{array}}\right\}$ Rules to eliminate left recursion.

$S \to aS' \mid a$
$S' \to AS' \mid \varepsilon$
$A \to b$

Q. Eliminate the left recursion from the given grammar.

(1) $\underbrace{S}_{A} \to \underbrace{S}_{A}\underbrace{0S1S}_{\alpha} \mid \underbrace{01}_{\beta}$ Here, S having two production but only one having left recursion.

Here, $A = S$
$\alpha = 0S1S$
$\beta = 01$

Eliminating left recursion.
$A \to \beta A'$
$A' \to \alpha A' \mid \varepsilon$

$S \to \overbrace{01}^{\beta}S'$
$S' \to \underbrace{0S1S}_{\alpha}S' \mid \varepsilon$

After elimination, final production list.
$S \to 01S'$
$S' \to 0S1SS' \mid \varepsilon$

Note: Here, we don't need to add 01 in S production because at the time of eliminating ε we will get $S \to 01$, so now no need to write.

(2) $S \rightarrow A$

$A \rightarrow Ad \mid Ae \mid aB \mid ac$

$B \rightarrow bBc \mid f$

Here from 7 productions, 2 productions having left recursion i.e. $A \rightarrow Ad \mid Ae$

$A \rightarrow A\alpha \mid \beta$

$A \rightarrow Ad$

Here, $A = A$

$\alpha = d$

$\beta_1 = aB$

$\beta_2 = ac$

Eliminating left recursion,

$A \rightarrow \beta_1 A' \mid \beta_2 A'$

$A' \rightarrow \alpha A' \mid \varepsilon$

$\therefore A \rightarrow aBA' \mid acA'$

$A' \rightarrow dA' \mid \varepsilon$

$A \rightarrow A\alpha \mid \beta$

$A \rightarrow Ae$

Here, $A = A$

$\alpha = e$

$\beta_1 = aB$

$\beta_2 = ac$

Eliminating left recursion

$A \rightarrow \beta_1 A' \mid \beta_2 A'$

$A' \rightarrow \alpha A' \mid \varepsilon$

$\therefore A \rightarrow aBA' \mid acA'$

$A' \rightarrow eA' \mid \varepsilon$

$\therefore$ Final productions,

$S \rightarrow A$

$A \rightarrow aBA' \mid acA'$

$A' \rightarrow dA' \mid eA' \mid \varepsilon$

$B \rightarrow bBc \mid f$

(3) $A \rightarrow Ba \mid Aa \mid c$

$B \rightarrow Bb \mid Ab \mid d$

2 productions are having left recursion,

$A \rightarrow A\alpha \mid \beta$

$A \rightarrow Aa$

Here, $A = A$

$\alpha = a$

$\beta_1 = Ba$

$\beta_2 = c$

Eliminating left recursion,

$A \rightarrow \beta_1 A' \mid \beta_2 A'$

$A' \rightarrow \alpha A' \mid \varepsilon$

$\therefore A \rightarrow BaA' \mid cA'$

$A' \rightarrow aA' \mid \varepsilon$

$A \rightarrow A\alpha \mid \beta$

$B \rightarrow Bb$

Here, $B\ A = B$

$\alpha = b$.

$\beta_1 = Ab$

$\beta_2 = d$

Eliminating left recursion

$A \rightarrow \beta_1 A' \mid \beta_2 A'$

$A' \rightarrow \alpha A' \mid \varepsilon$

$\therefore B \rightarrow AbB' \mid dB'$

$B' \rightarrow bB' \mid \varepsilon$

$\therefore$ Final productions.

$A \rightarrow BaA' \mid cA'$

$A' \rightarrow aA' \mid \varepsilon$

$B \rightarrow AbB' \mid dB'$

$B' \rightarrow bB' \mid \varepsilon$

- **Left factoring:** (common factor which is appearing maximum no. of production)

Eg: $S \rightarrow \underline{a}SSbS \mid \underline{a}SaSb \mid \underline{a}Sb \mid b$

After removing left factoring.

$S \rightarrow aSS' \mid b$
$S' \rightarrow SbS \mid aSb \mid b$

Q. Eliminate left factor from the foll. CFG:

Eg: $S \rightarrow \underline{b}SSaaS \mid \underline{b}SSaSb \mid \underline{b}Sb \mid a$

$S \rightarrow bSS' \mid a$
$S' \rightarrow SaaS \mid SaSb \mid b$
$S' \rightarrow SaS'' \mid b$
$S'' \rightarrow aS \mid Sb$

$\therefore$ G:

$S \rightarrow bSS' \mid a$
$S' \rightarrow SaS'' \mid b$
$S'' \rightarrow aS \mid Sb$

Eg: $S \rightarrow \underline{iEtS} \mid \underline{iEtS}eS \mid a$ (Try to face the substring)
$E \rightarrow b$

$S \rightarrow iEtS' \mid a$
$S' \rightarrow \underline{S} \mid \underline{S}eS$
$E \rightarrow b$

This ans. is also correct.

$S \rightarrow iEtS' \mid a$
$S' \rightarrow SS''$
$S'' \rightarrow \varepsilon \mid eS$
$E \rightarrow b$

Eg: $S \rightarrow \underline{a}Ad \mid \underline{a}B$
$A \rightarrow \underline{a} \mid \underline{a}b$
$B \rightarrow ccd \mid ddc$

G: $S \rightarrow aS'$
$S' \rightarrow Ad \mid B$
$A \rightarrow aA'$
$A' \rightarrow \varepsilon \mid b$
$B \rightarrow ccd \mid ddc$

Eg: $E \to T + E \mid T$
$T \to int \mid int * T \mid (E)$

$G: E \to TE'$
$E' \to +E \mid \varepsilon$
$T \to int T' \mid (E)$
$T' \to \varepsilon \mid *T$

Eg: $S \to a \mid ab \mid abc \mid abcd$

$\to S \to aS'$
$S' \to \varepsilon \mid b \mid bc \mid bcd$
$\to S' \to bS'' \mid \varepsilon$
$S'' \to \varepsilon \mid c \mid cd$
$\to S'' \to cS''' \mid \varepsilon$
$\to S''' \to \varepsilon \mid d$

$\therefore G: S \to aS'$
$S' \to bS'' \mid \varepsilon$
$S'' \to cS''' \mid \varepsilon$
$S''' \to \varepsilon \mid d$

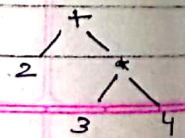
Eg: $S \to aAB$
$A \to aAD$
$B \to b$
$D \to e$

Here, aA are not common factors because they are produce from diff. Non-terminals

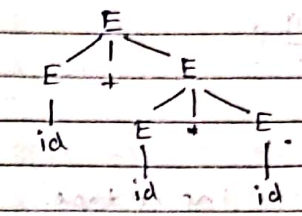
Q. Show that foll. grammar is ambiguous.

$E \to E+E \mid E*E \mid id$

$\to$ id + id * id
2 + 3 * 4

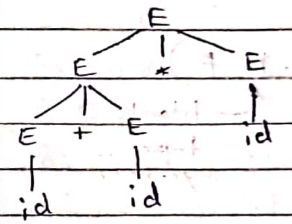

Abstract index tree

This is correct


Parse tree

$E \to E+E$
$\to id + E$
$\to id + E * E$
$\to id + id * E$
$\to id + id + id$

This is incorrect


Parse tree (Here + operation performs first) which is wrong. $(\underline{2+3} * 4)$

$E \to E * E$
$\to \cancel{id + E}\ (E+E) * E$
$\to id + E * E$
$\to id + id * E$
$\to id + id * id$

$\to$ We can derivate this string (id + id * id) using two left most derivations, from that one is correct and second is incorrect hence the given grammar is called ambiguous gramm

UNIT-5

TURING MACHINE:

... | B | B | 0 | 0 | 1 | 1 | B | ...

L ← Head → R ← input tape

$0^n 1^n$, $n > 0$

'B' → ~~Black~~ Blank tape symbol. ($\not b$)

$TM = (Q, \Sigma, \Gamma, q_0, b, F)$

~~Finite~~ ; tape symbols ; blank tape symbol.

Eg: $L = \{0^n 1^n, n > 0\}$

$= \{01, 0011, 000111, \ldots\}$

... | B | B | 0 | 0 | 1 | 1 | B | B | ...

Head

State/task	input	transition.
q_0	0	XRq_1
q_1	1	YLq_2
q_2	X	XRq_0

$\delta(q_0, 0) = XRq_1$
$\delta(q_1, 0) = 0Rq_1$
$\delta(q_1, 1) = YLq_2$
$\delta(q_2, 0) = 0Lq_2$
$\delta(q_2, X) = XRq_0$
$\delta(q_0, 0) = XRq_1$
$\delta(q_1, Y) = YRq_1$
$\delta(q_1, 1) = YLq_2$
$\delta(q_2, Y) = YLq_2$
$\delta(q_2, X) = XRq_0$
$\delta(q_0, Y) = YRq_1$
$\delta(q_1, Y) = YRq_1$
$\delta(q_1, B) = BNq_3$ (Halt)

State\input	0	1	X	Y	B
q_0	XRq_1			YRq_1	
q_1	$0Rq_1$	YLq_2		YRq_1	BNq_3
q_2	$0Lq_2$		XRq_0	YLq_2	
q_3					

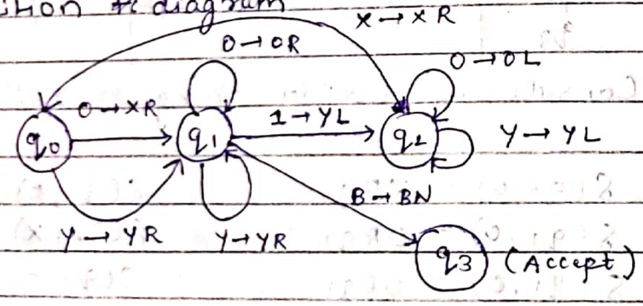
→ Considering invalid string : 00011

$\delta(q_0, 0) = XRq_1$
$\delta(q_1, 0) = 0Rq_1$
$\delta(q_1, 0) = 0Rq_1$
$\delta(q_1, 1) = YLq_2$
$\delta(q_2, 0) = 0Lq_2$
$\delta(q_2, 0) = 0Lq_2$
$\delta(q_2, X) = XRq_0$
$\delta(q_0, 0) = XRq_1$
$\delta(q_1, 0) = 0Rq_1$
$\delta(q_1, Y) = YRq_1$

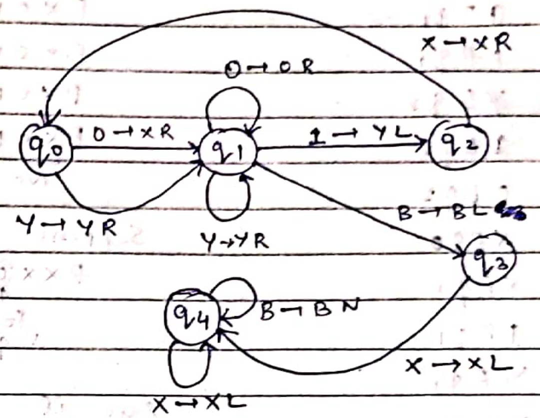
$\delta(q_1, 1) = YLq_2$
$\delta(q_2, Y) = YLq_2$
$\delta(q_2, 0) = 0Lq_2$
$\delta(q_2, X) = XRq_0$
$\delta(q_0, 0) = XRq_1$
$\delta(q_1, Y) = YRq_1$
$\delta(q_1, Y) = YRq_1$
$\delta(q_1, B) = \boxed{BNq_3 \text{ (Halt)}} \rightarrow BLq_3$

State\input	0	1	X	Y	B
q_0	XRq_1			YRq_1	
q_1	$0Rq_1$	YLq_2		YRq_1	BLq_3
q_2	$0Lq_2$		XRq_0	YLq_2	
q_3			XLq_4	YLq_3	
q_4			XLq_4		BNq_4

→ Transition ~~fe~~ diagram



→ Transition diagram for ~~invalid~~ generalized string.



Q. Design a turing machine for given language:

→ $L = \{ 0^i 1^j 2^k,\ i, j, k > 0 \ \& \ i = k \}$.
$\hookrightarrow (0^n 1^i 2^n)$

$= \{ 012,\ 001122,\ \ldots \}$

...	B	B	0	0	1	2	2	B	B	...
			↑ Head.							

State\input	input	transition.
q_0	0	XRq_1
q_1	2	YLq_2
q_2	X	XRq_0

$\delta(q_0, 0) = XRq_1$
$\delta(q_1, 0) = 0Rq_1$
$\delta(q_1, 1) = 1Rq_1$
$\delta(q_1, 2) = YLq_2$
$\delta(q_2, 1) = 1Lq_2$
$\delta(q_2, 0) = 0Lq_2$
$\delta(q_2, X) = XRq_0$
$\delta(q_0, 0) = XRq_1$
$\delta(q_1, 1) = 1Rq_1$
$\delta(q_1, Y) = YRq_1$
$\delta(q_1, 2) = YLq_2$
$\delta(q_2, Y) = YLq_2$
$\delta(q_2, 1) = 1Lq_2$
$\delta(q_2, X) = XRq_0$
$\delta(q_0, 1) = 1Rq_3$
$\delta(q_3, Y) = YRq_3$
$\delta(q_3, Y) = YRq_3$
$\delta(q_3, B) = \cancel{BNq_4}$ BLq_4
$\delta(q_4, Y) = YLq_4$
$\delta(q_4, Y) = YLq_4$
$\delta(q_4, 1) = 1Lq_4$
$\delta(q_4, X) = XLq_4$
$\delta(q_4, X) = XLq_4$
$\delta(q_4, B) = BNq_f$ (Accept)

B 0 0 1 2 2 B
B X 0 1 2 2 B
B X 0 1 Y 2 B
B X X 1 Y 2 B
$B \underbrace{XX}\, 1 \underbrace{YY}\, B$
No zeroes → q_4 will accept.

Suppose,
B X X 0 1 Y Y B
found 1 zero
∴ q_4 will reject.

Also, if
B X X 1 Y Y 2 B
found one 2
∴ q_3 will reject.

Transition Table.

State \ input	0	1	2	X	Y	B
q_0	XRq_1	$1Rq_3$				
q_1	$0Rq_1$	$1Rq_1$	YLq_2		YRq_1	
q_2	$0Lq_2$	$1Lq_2$	$\cancel{XRq_0}$	XRq_0	YLq_2	
q_3			Reject		YRq_3	BLq_4
q_4	Reject	$1Lq_4$		XLq_4	YLq_4	BNq_f