UNIT-IV (GRAMMER)

→ Regular Language
Represented
Grammer —Generator— Language —Acceptor— Automata

→ Grammer:-

The set of rules which are used in the generation of a string is called Grammer

→ Grammer is a 4-tuple variable $G=(V, T, P, S)$

where V is set of Non-terminals / variables

T is set of Terminals

P is set of productions

S is the start symbol of the grammer

Ex:- $S \longrightarrow aSB$
$S \longrightarrow aB$
$B \longrightarrow b$
} Grammer production

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

$V = \{S, B\}$, $T = \{a, b\}$ $P = \{S \to aSB, S \to aB, B - b\}$

$S = S$ (start Symbol)

Derivation :-

The process of deriving a string by using grammer rules

W = aabb

P: $S \rightarrow aSB/aB$

$B \rightarrow b$

LMD

Left Most Derivation

→ If we start replacing the left most non-terminal first is LMD

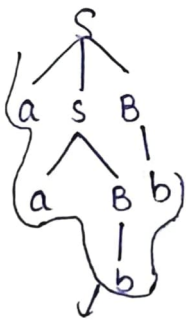
$\Rightarrow S$

$\Rightarrow a\underline{S}B$

$\Rightarrow aa\underline{B}B$

$\Rightarrow aab\underline{B}$

$\Rightarrow aabb$



yield of parts (String)

RMD

Right Most Derivation

→ If we start replacing the Right most - Non-terminal fist is RMD

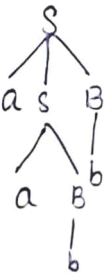
$\Rightarrow S$

$\Rightarrow as\underline{B}$

$\Rightarrow a\underline{S}b$

$\Rightarrow aa\underline{B}b$

$\Rightarrow aabb$



Derivation Tree :-

Representing derivation in a Tree structes

Ex: W = aabb

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow b$

$V = \{S, A, B\}$

$T = \{a, b\}$

$S = S$

$L(G) = \{ab\}$

② $S \rightarrow AB$

$A \rightarrow a/b$

$B \rightarrow c/d$

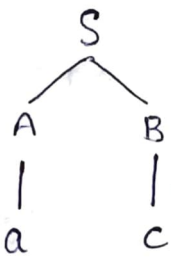
$V = \{S, A, B\}$

$T = \{a, b, c, d\}$

$S = S$

$L(G) = \{ac, bc, ad, bd\}$

$w = ac$



$w = ad$

Note:-

① Grammer G is a generating device

② If G is a Grammer then L(G) is language generated by Grammar G

③ Every Grammer represents only one language but the language can be generated by more than one language

Ex: $G_1 = ab$

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow b$

$S \rightarrow aB$

$B \rightarrow b$

$G_2 = ab$

④ Every grammer have only one start Symbol

⑤ Backus Normal form

$A \rightarrow \alpha_1$
$A \rightarrow \alpha_2$
$A \rightarrow \alpha_3$

$\Rightarrow A \rightarrow \alpha_1 | \alpha_2 | \alpha_3 | \cdots | \alpha_n$

(6) Grammer can be reccursive or non-recursive

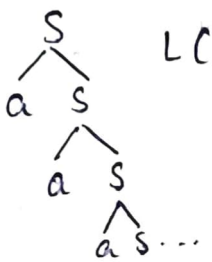
Recursive Grammar:

The Grammar G is said to be Recursive if there exist atleast one production which contain the same variable both of L.H.S and R.H.S of production

G_1 :

Ex: $S \rightarrow aS$

$S \rightarrow b$



$L(G_1) = a^*b$

G_2 : $S \rightarrow asb$

$S \rightarrow \epsilon$

$L(G_2) = \{a^n b^n / n \geq 0\}$

Non-Recursive :-

The Grammer G is said to be Non-recursive if no production contain same variable both at L.H.S and R.H.S (directly or indirectly)

Ex: $S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow b$

Ex: Find the language generated by following grammer

G1:- $S \rightarrow AaB$

$A \rightarrow a/c$

$B \rightarrow b$

$L(G1) = \{aab, cab\}$

$(a+c)ab$

G2:- $S \rightarrow AB$

$A \rightarrow 0/1/2/../9$

$B \rightarrow a/b/c/../z$

$L(G2) = \{0a, 0b, \ldots 0z, 1a, 1b, \ldots 1z, \ldots 9a, 9b, 9c, \ldots 9z\}$

$RE = (0+1+2+..9)(a+b+c+..z)$

G3:- $S \rightarrow R\ V\ O$

$R \rightarrow int/float$

$V \rightarrow x/y/z$

$O \rightarrow ;$

$RE = (int+float)(x+y+z);$

$L(G3) = \{intx; floatx;, inty; floaty;, intz; floatz;\}$

G4:- $S \rightarrow ABC$

$A \rightarrow a/d$

$B \rightarrow b$

$C \rightarrow c$

$L(G4) = \{abc, dbc\}$

$(a+d)^* bc$

G5:- $S \rightarrow XY$

$X \rightarrow aX/a \rightarrow a^+$

$Y \rightarrow Yb/b \rightarrow b^+$

$L(G5) \Longrightarrow RE \rightarrow a^+b^+$

→ Regular :- Every production in the form of

$A \rightarrow \alpha B / \beta$

(or)

$A \rightarrow B\alpha / \beta$

where $A, B \in V$

$\alpha, \beta \in T^*$

→ Context Free Grammer

$A \rightarrow \alpha$ where $\alpha \in (V+T)^*$

Ex: $A \rightarrow BCD$

$A \rightarrow abc$

$A \rightarrow BcD$ — B, D → variable; c → Terminal

① → Construct Grammer for following ~~grammer~~ Languages

① $L = \{a, ab\}$

(a) ~~$S \rightarrow aA$~~ $A \rightarrow b/\epsilon$ { (b) ~~$S \rightarrow aA/a$~~ $A \rightarrow b$

② $L = \{ab, ba, aa, bb\}$

G1: $S \rightarrow AA$

$A \rightarrow a/b$

G2: $S \rightarrow aA/bA$

$A \rightarrow a/b$

③ $L = \{abb, aba, bab\}$

$S \rightarrow Xb/Xa/bX$

$X \rightarrow AB$

$A \rightarrow a$

$B \rightarrow b$

④ Strings of a's and b's including ϵ

$S \rightarrow aS/bS/\epsilon$ // $S \rightarrow Sa/Sb/\epsilon$

Left Right Recursive

⑤ L = string starts with a $\Sigma = \{a, b\}$

$RE = a(a+b)^*$

$S \rightarrow aA$

$A \rightarrow aA/bA/\epsilon$

⑥ L = starts with a and ends with b

$S \rightarrow aAb$ $RE = a(a+b)^*b$

$A \rightarrow aA/bA/\epsilon$

⑦ starts and ends with different symbol

$S \rightarrow aAb/bAa$

$A \rightarrow aA/bA/\epsilon$

(8) Starts and ends with same symbol $\Sigma = \{a, b\}$

Sol:- G: $S \rightarrow aAa / bAb / a / b / \epsilon$ (if ask, can we include ϵ)

$A \rightarrow aA / bA / \epsilon$

(9) $L = \{a^n / n \geq 0\}$

G:- a^*

$S \rightarrow aS / \epsilon$

(10) $L = \{$ set of all strings of length atleast '2' $\}$

Sol:- $\Sigma = \{a, b\}$

$S \rightarrow AAB$

$A \rightarrow a / b$

$B \rightarrow aB / bB / \epsilon$

$RE = (a+b)^2 (a+b)^*$

(11) $L = (a+b)^*$

G: $S \rightarrow as / bs / \epsilon$

$L = (a+b)^+$

G: $S \rightarrow as / bs / a / b$

For the termination

(12) $L =$ atmost 2

$RE = (a+b+\epsilon)(a+b+\epsilon)$

G:- $S \rightarrow AA$

$A \rightarrow a / b / \epsilon$

(13) $L = \{$ set of all palindrome number $\}$

Can't construct FA because its not regular

WaW^R / WBW^R

$S \rightarrow aSa/bSb/\epsilon/a/b$

(14) $L = \{$ strings of even length $\}$ $\Sigma = \{a, b\}$

$S \rightarrow AS/\epsilon$

$A \rightarrow BB$

$B \rightarrow a/b$

$RE = [(a+b)(a+b)]^*$

(15) $L = \{a^n b^m / n, m \geq 1\}$

~~$S \rightarrow aAbA$~~

~~$A \rightarrow a/b/\epsilon$~~

~~$B \rightarrow$~~

G1:-

$S \rightarrow AB$

$A \rightarrow aA/a$

$B \rightarrow bB/b$

G2:-

$S \rightarrow aAbB$

$A \rightarrow aA/\epsilon$

$B \rightarrow bB/\epsilon$

(16) $L = \{a^n b^n c^m / n, m \geq 1\}$

$S \rightarrow AB$

$A \rightarrow aAb/ab$

$B \rightarrow cB/c$

(17) $L = \{a^n c^m b^n / n, m \geq 1\}$

$S \rightarrow aSb/aAb$

$A \rightarrow cA/c$

(18) $L = \{a^n b^n c^m d^n / n, m \geq 1\}$

$S \rightarrow AB$

$A \rightarrow aAb/ab$

$B \rightarrow cBd/cd$

(19) $L = \{ a^n b^n c^n / n \geq 1 \}$

(20) $L = \{ a^n b^{2n} / n \geq 1 \}$

$S \rightarrow aSbb / abb$

(21) $L = \{ a^n b^m c^m d^n / n, m \geq 1 \}$

$S \rightarrow aAd / asd$

$A \rightarrow bAc / bc$

(22) $L = \{ a^{m+n} b^m c^n / m, n \geq 1 \}$ $a^n . a^m . b^m c^n$

$S \rightarrow asc / aAc$

$A \rightarrow aAb / ab$

(23) $L = \{ a^n b^{n+m} c^m / m, n \geq 1 \} \Rightarrow a^n b^n . b^m c^m$

$S \rightarrow AB$

$A \rightarrow aAb / ab$

$B \rightarrow bBc / bc$

(24) $L = \{ a^n b^m c^{n+m} / m, n \geq 1 \}$ $a^n b^m c^m . c^n$

$S \rightarrow aAc / aSc$

$A \rightarrow bAc / bc$

(25) $L = \{$ each string contains exactly 2 a's $\}$

$S \rightarrow AaAaA$

$A \rightarrow bA/\epsilon$

$r = b^*ab^*ab^*$

(26) $L = \{$ each string atleast 2 a's $\}$

$S \rightarrow AaAa$

$A \rightarrow aA/bA/\epsilon$

$r = (a+b)^*a(a+b)^*a(a+b)^*$

(27) $L = \{$ Atmost 2a's $\}$

$S \rightarrow BABAB$

$B \rightarrow bB/\epsilon$

$A \rightarrow a/\epsilon$

$r = \epsilon + b^*ab^* + b^*ab^*ab^*$

$r = b^*(a+\epsilon)b^*(a+\epsilon)b^*$

(28) $L = \{$ NO of a's Congurent to 0 mod 3 $\}$ $\Sigma = \{a, b\}$

$S \rightarrow AS/X/\epsilon$

$A \rightarrow XaXaX$

$X \rightarrow bX/\epsilon$

$r = b^* + (b^*ab^*ab^*)^*$

(29) 4th symbol from left end end is b

$S \rightarrow AbB$

$A \rightarrow XXX$

$B \rightarrow aB/bB/\epsilon$

$X \rightarrow a/b$

$r = XXXbX$

$(a+b)(a+b)(a+b)b(a+b)^*$

(30) 5th symbol from Right end is a. $\Sigma = \{a, b\}$

$S \rightarrow XaAAAA$

$X \rightarrow aX/bX/\epsilon$

$A \rightarrow a/b$

$(a+b)^* a\,(a+b)(a+b)(a+b)(a+b)$

$R = XaAAAA$

(31) L - length of sting exactly 3 $\Sigma = \{a, b\}$

$S \rightarrow XXX$

$X \rightarrow a/b$

(32) L = length of string is divisible by 3

$S \rightarrow AS/\epsilon$

$A \rightarrow XXX$

$X \rightarrow a/b$

⇒ Chamskey Hierarchy

→ Based on the form of production, the gramme can be classified into 4 types

① Type 3 or Regular Grammer

② Type 2 or Context Free Grammer

③ Type 1 or Context Sensitive Grammer

④ Type 0 or Recursive Enumerable Grammer

① Type 3 - Grammar (RG)

The Grammer G is said to be in the Type-3 if every production is of the form

Right Restricted

$A \to \alpha B / \beta$ → (Right Linear Grammer) where $\alpha, \beta \in T^*$, $A, B \in V$

(or)

$A \to B\alpha / \beta$ where $\alpha, \beta \in T^*$, $A, B \in V$

↳ (Left linear Grammer)

Ex: ① $S \to as/\epsilon$ $L(G) = a^*$ (RLG)

② $S \to as/bs/\epsilon$ $L(G) = (a+b)^*$ (RLG)

③ $S \to Sa/sb/\epsilon$ $L(G) = (a+b)^*$ (LLG)

② Type 2 - Grammar (CFG)

The Grammer 'G' is said to be CFG if every production of the form

Left Restricted

$A \to \alpha$ wher $\alpha \in (V+T)^*$

Ex: $S \to asb/\epsilon$ $L(G) = a^n b^n / n \geq 0$

$S \to as/bs/\epsilon$ $L(G) = (a+b)^*$

↓

Both RG and CFG

RG ⊂ CFG

$S \rightarrow asb/bsa/\epsilon$ $L(G)$: ~~Equal number of a's and b's~~ $(a^n b^n + b^n a^n)$

③ Type-1 Grammer (CSG):

The Grammer G is said to be in Type-1 if every production is of the form

No restrictions on both left and right

$\alpha \rightarrow \beta$ where $\alpha, \beta \in (V+T)^*$

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

$\beta \neq \epsilon$

Ex: $S \rightarrow aSAC/abc$

$cA \rightarrow Ac$

$bA \rightarrow bb$

Note: → Type-1 Grammer is called as length increasing grammer

→ The Language generated by Type-1 free from ϵ

④ Type-0 Grammer (Unrestricted Grammer):

The Grammer G is said to be unrestricted Grammer if every production is of the form

$\alpha \rightarrow \beta$ where $\alpha, \beta \in (V+T)^*$

Note: Atleast one variable must be there at L.H.S

of any production in every grammer.

Ex:

$S \longrightarrow aSBc/\epsilon$

$aB \longrightarrow ba$

$bB \longrightarrow c$

Type3 C Type2 C Type1 C Type0

→ Regular Grammer (Type-3)

The Grammer which generates the RL is Regular Grammer

$$A \longrightarrow \alpha B/B\alpha/B$$

Types of RG

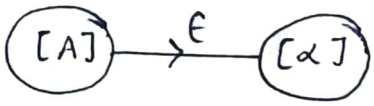
Right linear Grammer	Left linear Grammer
① $A \longrightarrow \alpha B/\beta$ where $\alpha, \beta \in T^*$ $A, B \in V$	① $A \longrightarrow B\alpha/\beta$ where $\alpha, \beta \in T^*$ $A, B \in V$
Ex: $S \longrightarrow as/bs/\epsilon$	Ex: $S \longrightarrow Sa/Sb/\epsilon$

⇒ Conversion from RLG to ϵ-NFA

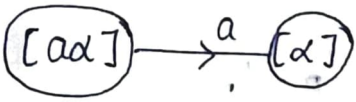
Note: ① Start symbol is the initial state

② If $A \rightarrow \alpha$ is a production

$$\delta(A, \epsilon) = [\alpha]$$



③ $\delta(a\alpha, a) = [\alpha]$

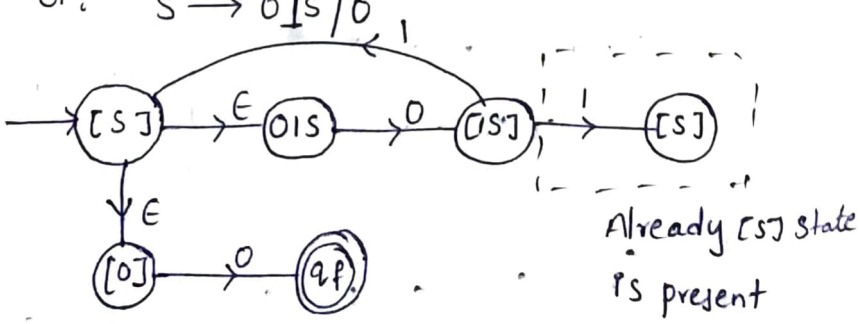


④ For every terminal a

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

Ex: RLG to ϵ-NFA

G: $S \rightarrow 01S/0$



Work

Ex2: $S \rightarrow 010S/0/1$ (Construct ϵ-NFA)

Ex3: Find out the number of states in ϵ-NFA, NFA and DFA for the Grammer G

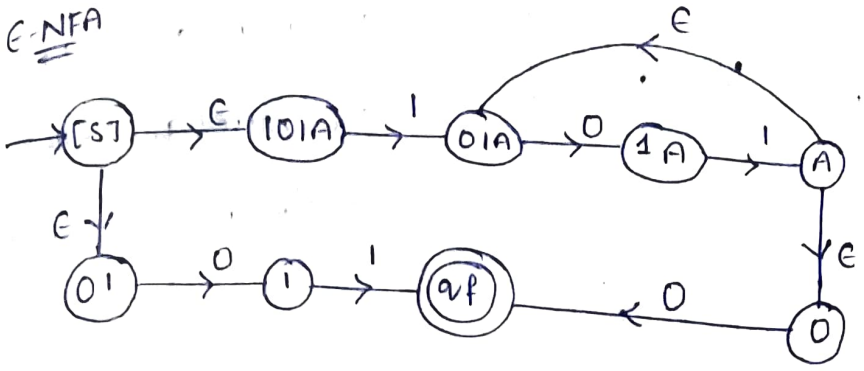
G: $S \rightarrow 11S/0$

For Two productions

(4) $S \rightarrow 101A/01$

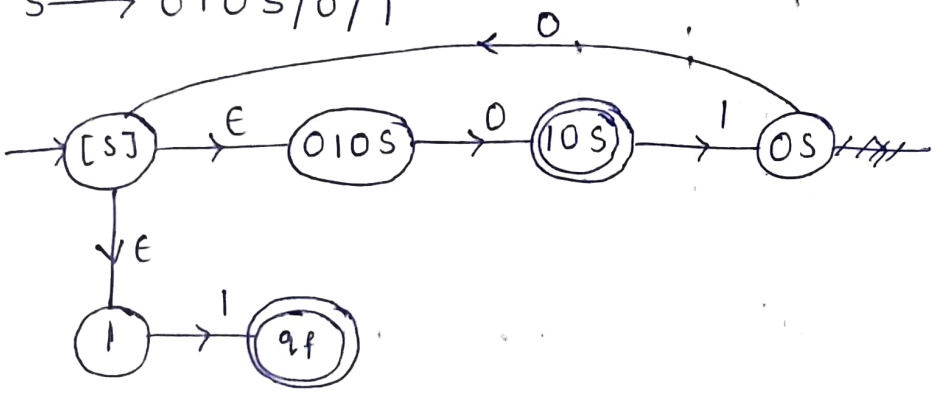
$A \rightarrow 01A/0$

ϵ-NFA



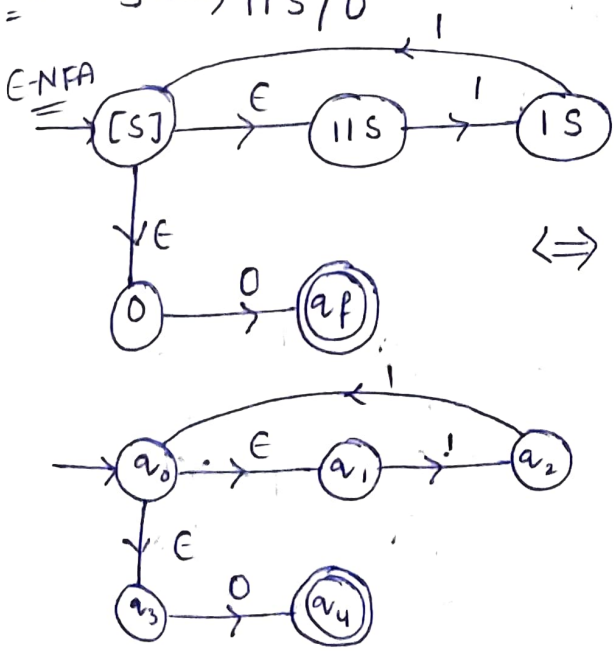
Assignment

Ex1:- $S \rightarrow 010S/0/1$



Ex3:- $S \rightarrow 11S/0$

ϵ-NFA



$\Leftrightarrow$

$\epsilon(q_0) = \{q_0, q_1, q_3\}$

ϵ-closure$(q_1) = \{q_1\}$

ϵ-closure$(q_2) = \{q_2\}$

ϵ-closure$(q_3) = \{q_3\}$

ϵ-closure$(q_4) = \{q_4\}$

ε-NFA

δ	0	1	ε
→ q_0	∅	∅	$\{q_1, q_3\}$
q_1	∅	q_2	∅
q_2	∅	q_0	∅
q_3	q_4	∅	∅
* q_4	∅	∅	∅

ϵ-closure$(q_0) = \{q_0, q_1, q_3\}$

ϵ-closure$(q_1) = \{q_1\}$

ϵ-closure$(q_2) = \{q_2\}$

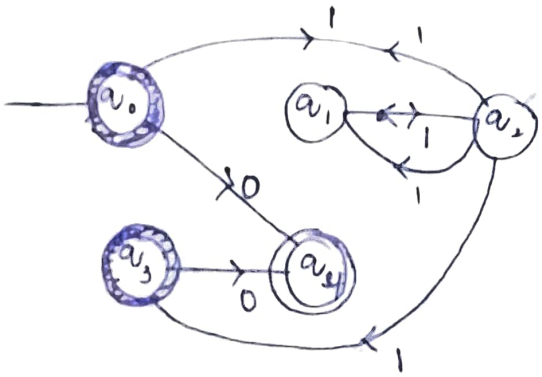
ϵ-closure$(q_3) = \{q_3\}$

ϵ-closure$(q_4) = \{q_4\}$

NFA

δ'	0	1
→ q_0	$\{q_4\}$	$\{q_2\}$
q_1	∅	$\{q_2\}$
q_2	∅	$\{q_0, q_1, q_2\}$
* q_3	$\{q_4\}$	∅
* q_4	∅	∅

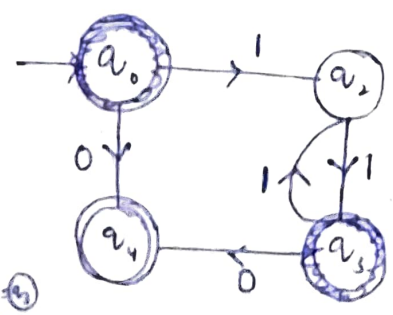
NFA



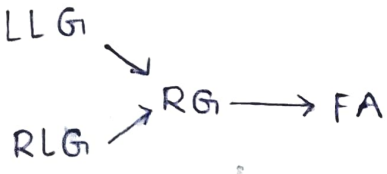
NFA - DFA

DFA

	0	1
→ q_0	$q_0 q_4$	q_2
* q_4	∅	∅
* q_2	∅	$q_0 q_1 q_3$
* $q_0 q_1 q_3$	q_4	q_2



Conversion between Grammer and Finite Automata



conversion from LLG (Left Linear Grammer) to FA

$A \longrightarrow \alpha B / \beta$ (RLG)

$A \rightarrow B\alpha / \beta$ (Reverse of RLG)

Step1:

① Convert LLG into RLG by Reversing RHS of every production

$A \longrightarrow B\alpha / \beta$ (LLG)

$A \rightarrow \alpha B / \beta$ (RLG)

② Obtain ϵ-NFA

③ Interchanging the initial and final states and change the direction of edges (Reverse of FA)

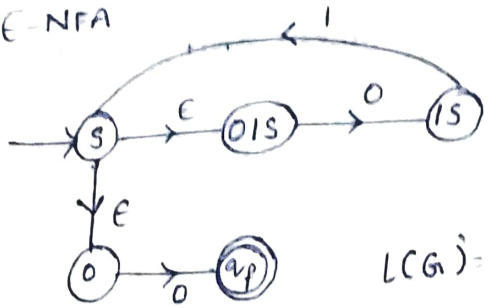
Ex:-1 Construct FA by using following Grammer

$S \longrightarrow S10/0$ (LLG)

Step1: Reverse

$S \longrightarrow 01S/0$ (RLG)

step1: ∈-NFA

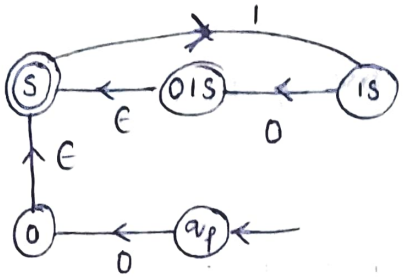


$L(G) = \{0, 010, 01010, \ldots\}$

$= (01)^*0$

step3: a) Interchange the final and initial states

b) change the directions



Language accepted by the FA

$L(G) = \{0, 010, 01010, \ldots\}$

$= 0(10)^*$.

Given production

$S \rightarrow S10/0$

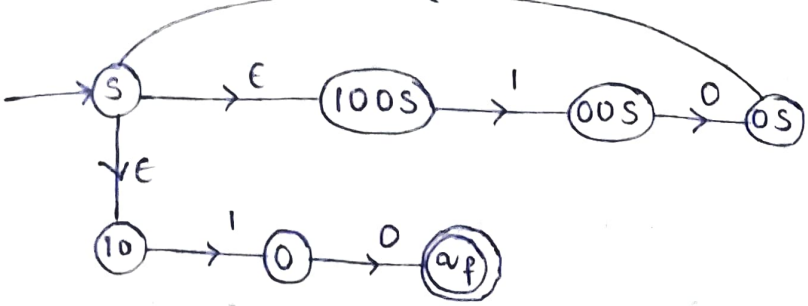
$L(G) = 0(10)^*$

(2) Construct ∈-NFA by using following Grammer

$S \rightarrow S001/01$

step1: $S \rightarrow 100S/10$

step2: Obtain ∈-NFA



$L = \{ 10, 10010, 10010010, \ldots \}$

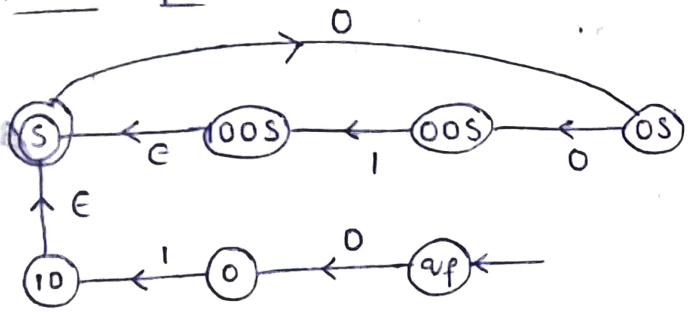
$= (100)^* 10$

steps:

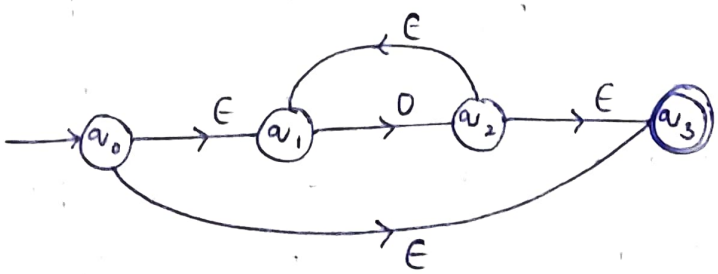
$S \rightarrow S001/01$

$L(G) = 01[001)^*$

Reverse FA



→ ϵ-NFA for 0^*



⇒ Conversion from FA to RG (Equivalence between FA and RG)

Note: ① No. of variables = No of states

② Initial state: start symbol of RG (variable)

③ If $\delta(A,a) = B$ then

$A \longrightarrow aB$ is the production

④ If B is final state then add $B \to \epsilon$ or $A \to a$

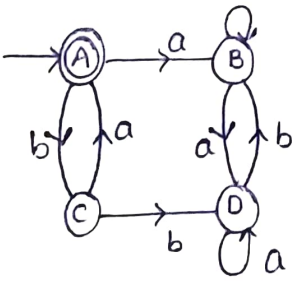
Ex: $\delta(A,a) = B$

$A \to aB$ if B is not a final state

$A \to aB$ (or) $A \to aB$, $B \to \epsilon$ | $A \to a$

If B is a final state

Ex: 1



Grammer

$A \longrightarrow aB / bc / \epsilon$

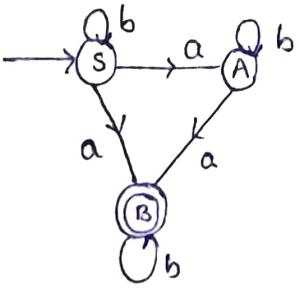
$B \longrightarrow aD / bB$

$C \longrightarrow aA / bD$

$D \longrightarrow aD / bB$

$C \to aA / bD / a$ → If ϵ is not present in A

Ex2: Construct RLG for the following FA



Grammer

~~S → S b / a A / a B~~

~~A →~~

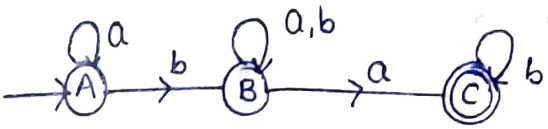
$S \rightarrow bS/aA/aB$

$A \rightarrow bA/aB$

$B \rightarrow bB/\epsilon$

Conversion from FA to LLG

FA



Grammer

$A \rightarrow aA/bB$

$B \rightarrow aB/bB/aC$

$C \rightarrow bC/\epsilon$

} RLG

LLG

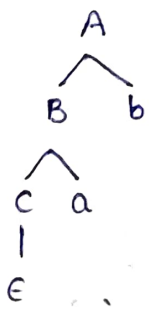
$A \rightarrow Aa/Bb$

$B \rightarrow Ba/Bb/Ca$

$C \rightarrow Cb/\epsilon$

$\Rightarrow$ But we can't consider this is LLG for the given FA

It will accept other strings also



ab is not accepting

ba is accepting

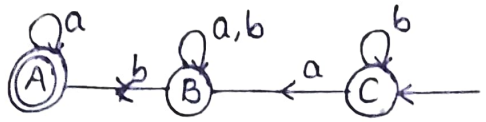
Steps to convert FA into LLG

step1: Reverse the FA

step2: Write RLG for FA

step3: Write reverse of RLG. It means it is LLG

step1:



step2:

$C \longrightarrow bC/aB$

$B \longrightarrow aB/bB/bA$

$A \longrightarrow aA/\epsilon$ . . .

step3: Reverse of RLG

LLG

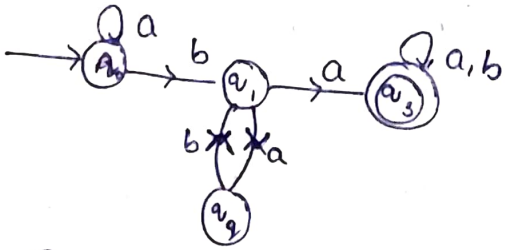
$C \longrightarrow Cb/Ba$

$B \longrightarrow Ba/Bb/Ab$

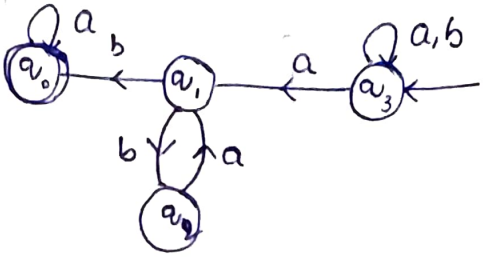
$A \longrightarrow Aa/\epsilon$

Construct LLG for the following FA

a)



step1: Reverse of FA



step2: RLG

$q_3 \rightarrow aq_3 / bq_3 / aq_1$

$q_1 \rightarrow bq_0 / bq_2$

$q_2 \rightarrow aq_1$

$q_0 \rightarrow aq_0 / \epsilon$

step3: LLG

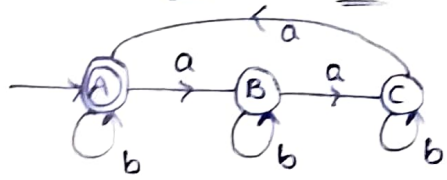
$q_3 \rightarrow q_3a / q_3b / q_1a$

$q_1 \rightarrow q_0b / q_2b$

$q_2 \rightarrow q_1a$

$q_0 \rightarrow q_0a / \epsilon$

→ Construct LLG and RLG for the following FA

① 

RLG

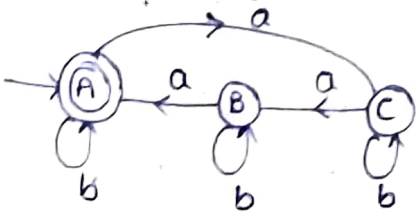
$A \rightarrow bA / aB / \epsilon$

$B \rightarrow bB / aC$

$C \rightarrow b\epsilon / aA$

For LLG

step1: Reverse of FA



step2: RLG

$A \rightarrow bA / aC / \epsilon$

$B \rightarrow bB / aA$

$C \rightarrow cB / bC$

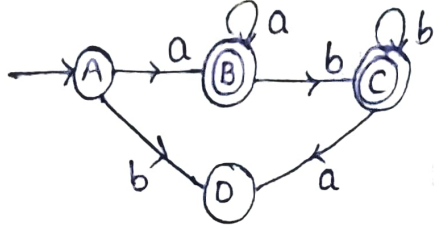
step3: LLG

$A \rightarrow Ab / Ca / \epsilon$

$B \rightarrow Bb / Aa$

$C \rightarrow B\epsilon / cb$

(2)



RLG

$A \rightarrow aB/bD$

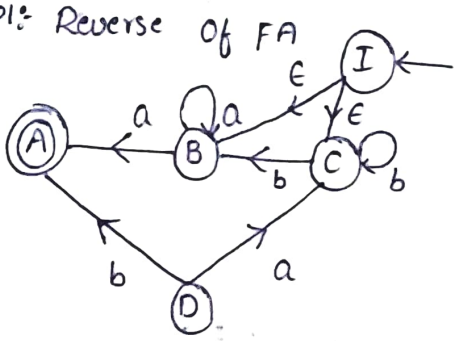
$B \rightarrow aB/bC/\epsilon$

$C \rightarrow bC/aD/\epsilon$

D is a dead state

FOR LLG

Step1:- Reverse of FA



step2:-

RLG

$I \rightarrow B/C$

$B \rightarrow aB/aA/\epsilon$

$C \rightarrow bC/bB$

$A \rightarrow \epsilon$

$D \rightarrow bA/aC$

Step3:- Reverse RLG

LLG

$I \rightarrow B/C$

$B \rightarrow Ba/Aa$

$C \rightarrow Cb/Bb$

$A \rightarrow \epsilon$

$D \rightarrow Ab/Ca$

Context Free Grammer

CFG:- The grammer G is said to be CFG if every production is of the form

$A \longrightarrow \alpha$ where $\alpha \in (V+T)^*$

$A \in V$

Ex:- ① $S \longrightarrow aSb/\epsilon$

② $S \longrightarrow aSb/bSa/\epsilon$

CFL (Context Free Language)

The language which is generated by CFG is called as CFL

Ex:- $S \longrightarrow aSb/\epsilon \longrightarrow$ CFG

$L = \{a^n b^n / n \geq 0\} \longrightarrow$ CFL

↳ Non-Regular but accepted by pull-Down Automate (PDA)

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

↳ Not accepted by PDA, but it is Non-regular.

Note:-

① Every Regular Language is CFL

② Every Non-Regular Language need not be CFL i.e some of the NRL are CFL

→ simplification of CFG

The process of detection and elimination of useless symbol, unit production and empty or null production is known as simplification of CFG

Useless Symbol:

The variable which are not involved in the derivation of any string is called as useless symbol.

Ex:1 $S \to aA$

$A \to b$

$B \to a$ → useless symbol, so eliminate it

⇓

$S \to aA \mid ab$

$A \to b$

Ex2: $S \to aA$

$A \to bB/c/d$

↳ B is undefined, so B is useless.

⇓ so, eliminate it.

$S \to aA$

$A \to c/d$

Ex3: $S \to aAb/aD$

$A \to aB/bB$ ⇒

$B \to bB/b$

$S \to aAb$

$A \to aB$

$B \to bB/b$

Note:

① select the variables which are not reachable from the start symbol of the grammer and remove them along with the production.

② select the variables which are reachable from start symbol of the grammer but they are not deriving

Unit production

The production is of the form $A \to B$ where

$A, B \in V$

Note: Identify the unit production and remove them by replacing the equivalent derivations

Ex:1 $S \to A$

$A \to bA/B$ $(A \to B) \Rightarrow$

$B \to d$

$S \to A$

$A \to bA/d$

Ex:2 $S \to Aa/Bb$ (Eliminate unit productions)

$A \to aA/B/a$ ④

$B \to bB/\epsilon/b$ ③

$C \to cC/D/e$ ②

$D \to dD/E/d$ ①

$E \to e$

Bottom-top

① $D \rightarrow dD/e/d$

② $C \rightarrow cC/dD/e/d$

③ $B \rightarrow bB/cC/dD/e/d/b$

④ $A \rightarrow aA/bB/cC/dD/e/d/b/a$

Equivalent production

$S \rightarrow Aa/Bb$

$A \rightarrow aA/bB/cC/dD/e/d/b/a$

$B \rightarrow bB/cC/dD/b/d/e$

$C \rightarrow cC/dD/e/d$

$D \rightarrow dD/e/d$

Ex3: Eliminate unit production from the following CFG

(a) $S \rightarrow Aa/B$

$A \rightarrow a/bc/B$

$B \rightarrow A/bb$

eliminate $B \rightarrow A$

$\Downarrow$

$S \rightarrow Aa/(B)$

$A \rightarrow a/bc/A/bb$

$\Downarrow$

$S \rightarrow Aa/A/bb$

$A \rightarrow a/bc/A/bb$

(b) $S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow C/b$

$C \rightarrow D$

$D \rightarrow E$

$E \rightarrow a$

$B \rightarrow C$, $C \rightarrow D$, $D \rightarrow E$ — Unit prod

$\Downarrow$

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow a/b$

Empty / Null / Epsilon production :-

The production is of the form $A \to \epsilon$

Note : Identify Null / Empty productions and remove them by replacing the equivalent derivations

Ex :- $S \to AaB$

$A \to aA/\epsilon$

$B \to bB/\epsilon$

step 1 :- Eliminate $A \to \epsilon$

$S \to AaB/aB$ (substitute ϵ in place of A)

$A \to aA/a$

$B \to bB/\epsilon$

step 2 :- Eliminate $B \to \epsilon$

$S \to AaB/aB/Aa/a$ (substitute ϵ in place of B)

$A \to aA/a$

$B \to bB/b$

Ex :-2 $S \to AaB$

$A \to \epsilon$ ($A \to \epsilon$ is alone, so directly substitute inplace of A)

$B \to bB/\epsilon$

⇓

step :- $S \to aB$

$B \to bB/\epsilon$ ($B \to \epsilon$ is not alone)

step2:- Eliminate $B \to \epsilon$

$S \to aB/a$

$B \to bB/b$

Ex3:- $S \to AaB$

$A \to aA/\epsilon$ (Already done)

$B \to bB/\epsilon$

Ste

Eliminate ϵ-production from the following

① $S \to AB$

$A \to aAA/\epsilon$

$B \to bB/\epsilon$

② $S \to AbaC$

$A \to BC$

$B \to b/\epsilon$

$C \to D/\epsilon$

$D \to d$

③ $S \to ABC$

$A \to Bc/a$

$B \to bAc/\epsilon$

$C \to CAB/\epsilon$

① step Eliminate $A \to \epsilon$

Sol:-

$S \to AB/B$

$A \to aAA/a$

$B \to bB/\epsilon$

step2: Eliminate $B \rightarrow \epsilon$

$S \rightarrow AB/B/A$

$A \rightarrow aAA/a$

$B \rightarrow bB/b$

(2A) step1: Eliminate $B \rightarrow \epsilon$

$S \rightarrow AbaC$

$A \rightarrow BC/C$

$B \rightarrow b$

$C \rightarrow D/\epsilon$

$D \rightarrow d$

step2: Eliminate $C \rightarrow \epsilon$

$S \rightarrow AbaC/Aba$

$A \rightarrow BC/C/B$

$B \rightarrow b$

$C \rightarrow D$

$D \rightarrow d$

(3A) step1: Eliminate $B \rightarrow \epsilon$

$S \rightarrow ABC/AC$

$A \rightarrow BC/a/C$

$B \rightarrow bAC$

$C \rightarrow \epsilon AB/\epsilon A/\epsilon$

Step:- Eliminate $C \rightarrow \epsilon$

$S \rightarrow ABC/AC/AB/A$

$A \rightarrow BC/a/C/B$

$B \rightarrow bAC/bA$

$C \rightarrow \epsilon AB/\epsilon A/AB/A$

Ex:- $S \rightarrow AB$

$A \rightarrow aA/B/\epsilon$

$B \rightarrow b$

Unit production

$S \rightarrow AB$

$A \rightarrow aA/b/\epsilon$

ϵ-production

$S \rightarrow AB$

$A \rightarrow aA/b/a$

Order of simplification process of CFG

1. Removal of ϵ-production
2. Removal of Unit-production
3. Removal of Useless symbols

Note:- The grammer which is obtained after elimination of useless sylombles (ϵ-production, Unit-production) is called as Reduced CFG/ Simplified CFG.

(1) simplify the following CFG

$S \rightarrow ABC / BaB$

$A \rightarrow aA / BaC / aaa$

$B \rightarrow bBb / a / D$

$C \rightarrow CA / AC$

$D \rightarrow \epsilon$

sol: ① elimination of ϵ-production

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

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

$B \rightarrow bBb / a / bb$

$[B \rightarrow D, D \rightarrow \epsilon]$

$C \rightarrow CA / AC$ → useless symbol (NO terminals)

② Elimination of Unit production (NO - Unit production)

③ Elimination of useless symbols

Simplified CFG

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

$A \rightarrow aA / aaa$ → useless symbol

$B \rightarrow bBb / a / bb$

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

$B \rightarrow bBb / a / bb$

Normal Form

- CNF (Cchomskey Normal Form)
- GNF

⇒ Chomskey Normal Form:

The CFG is said to be in CNF if every production of CFG is in the form of

$A \longrightarrow BC / a$ where $A, B, C \in V$ and $a \in T$

(1) (2)

Variable → Variable Variable
Variable → Terminal

Ex: $S \xrightarrow{(1)} aA \xrightarrow{(2)} b / bB$

$A \xrightarrow{(3)} bB / b^{(4)}$

$B \xrightarrow{(5)} b$

④, ⑤ are in CNF

In place b, can substitute B

$S \xrightarrow{(1)} aAB / BB^{(2)}$

$A \xrightarrow{(3)} BB / b^{(4)}$

$B \xrightarrow{(5)} b$

② ③, ④, ⑤ are in CNF

Let's take C = aA

$S \longrightarrow CB / BB$

$A \longrightarrow BB / b$

$B \longrightarrow b$

$C \longrightarrow aA$

Let's D = a

⇒

S → CB / BB
A → BB / b
B → b
C → DA
D → a

→ Griebach Normal Form (GNF):-

The Grammer (CGF) is said to be in GNF if every production is in the form of

$A \rightarrow a\alpha$ where

$A \in V$

$a \in T$

$\alpha \in V^*$

Variable → Terminal var

var → Term variable

var → Term var, var... var

Ex: $S \rightarrow a\overset{①}{A}b / b\overset{②}{B}$ ✗

$A \rightarrow b\overset{③}{B} / \overset{④}{a}$

$B \rightarrow \overset{⑤}{b}$

②, ③, ④, ⑤ are in GNF

⇓

$S \rightarrow a\overset{①}{A}B / b\overset{②}{B}$ ✓

$A \rightarrow b\overset{③}{B} / \overset{④}{a}$

$B \rightarrow \overset{⑤}{b}$

Ex: $S \rightarrow \overset{①}{A}aB$

$A \rightarrow \underset{②}{a} / \underset{③}{b}$

$B \rightarrow \underset{④}{B}a / \underset{⑤}{b}$

②, ③, ⑤ are in GNF

⇓

$S \rightarrow aaB / baB$

$A \rightarrow a/b$ → useless

$B \rightarrow bC/b$

$C \rightarrow ac/a$

$B \rightarrow Ba/b$

$B \rightarrow bC/b$

$C \rightarrow aC/a$

$\Rightarrow$

$S \rightarrow aaB / baB$

$B \rightarrow bC / b$

$C \rightarrow aC / a$

$\Downarrow$

$S \rightarrow aDB / bDB$

$D \rightarrow a$

$B \rightarrow bC / b$

$C \rightarrow aC / a$

$\Rightarrow$ Convert the following CFG into CNF

① $S \rightarrow aAb / bB$

$A \rightarrow bB / a$

$B \rightarrow b$

② $S \rightarrow bA / aB$

$A \rightarrow bAA / aS / a$

$B \rightarrow aBB / bS / b$

③ $S \rightarrow aAbB$

$A \rightarrow Ab / b$

$B \rightarrow Ba / a$

④ $S \rightarrow aA / bB$

$A \rightarrow bAA / a$

$B \rightarrow BBa / b$

⑤ $S \rightarrow Absb / ab$

⇒ Represent the following CFG into GNF

① $S \rightarrow AaB$

$A \rightarrow a/b$

$B \rightarrow Ba/b$

② $S \rightarrow YY/0$

$Y \rightarrow SS/1$

③ $S \rightarrow XY$

$X \rightarrow YS/1$

$Y \rightarrow SX/0$

④ $S \rightarrow XY$

$X \rightarrow 0X/1X/1$

$Y \rightarrow 1$

⇒

CFG = CNF

②sol:-

$S \rightarrow b\overset{①}{A}/a\overset{②}{B}$

$A \rightarrow b\overset{③}{A}A/a\overset{④}{S}/a\overset{⑤}{}$

$B \rightarrow a\overset{⑥}{B}B/b\overset{⑦}{S}/b\overset{⑧}{}$

⑤ ⑧ are in CNF

⇓

$S \rightarrow C\check{A}/D\check{B}$

$C \rightarrow b\checkmark$

$D \rightarrow a\check{}$

$A \rightarrow \overset{\times}{b\check{A}}A/D\check{S}/a\check{}$

$B \rightarrow a\check{B}B/C\check{S}/b\check{}$

$\Downarrow$

$S \to CA/DB$

$C \to b$ ✓

$D \to a$ ✓

$A \to XA/DS/a$

$B \to YB/CS/b$

$X \to bA$

$Y \to aB$

$\Rightarrow$

$S \to CA/DB$

$C \to b$

$D \to a$

$A \to XA/DS/a$

$B \to YB/CS/b$

$X \to CA$

$Y \to DB$

③ Sol:- $S \to aA/bB$

$A \to Ab/b$ ✓

$B \to Ba/a$ ✓

$\Downarrow$

$S \to CA/DB$

$C \to a$ ✓

$D \to b$ ✓

$A \to AD/b$

$B \to BC/a$

④ Sol:- $S \to aA/bB$

$A \to bAA/a$ ✓

$B \to BBa/b$ ✓

$\Downarrow$

$S \to CA/DB$

$C \to a$

$D \to b$

$A \to \underline{bA}A/a$

$B \to BB\underline{a}/b$

$\Rightarrow$

$S \to CA/DB$

$C \to a$

$D \to b$

$A \to XA/a$ $\Rightarrow$

$B \to BY/b$

$X \to bA$

$Y \to Ba$

$S \to CA/DB$

$C \to a$

$D \to b$

$A \to XA/a$

$B \to BY/b$

$X \to DA$

$Y \to BC$

⑤ $S \to absb/ab$

$\Downarrow$

$S \to \underset{X}{ab}\underset{Y}{Sb}/AB$

$A \to a$

$B \to b$ $\Rightarrow$

$S \to XY/AB$

$X \to ab$

$Y \to Sb$

$A \to a$

$B \to b$

$\Downarrow$

$S \to XY/AB$

$X \to AB$

$Y \to SB$

$A \to a$

$B \to b$

CGF - GNF

① $S \to AaB$

$A \to a/b$ $\Rightarrow$

$B \to Ba/b$

$S \rightarrow AaB$

$A \rightarrow \check{a}/\check{b}$

$B \rightarrow b\check{c}/\check{b}$

$C \rightarrow a\check{C}/\check{a}$

$\Rightarrow$

$S \rightarrow aaB / baB$

$\boxed{A \rightarrow a/b}$ → useless

$B \rightarrow bc/b$

$C \rightarrow aC/a$

$\Downarrow$

$S \rightarrow aDB/bDB$

$D \rightarrow a$

$B \rightarrow bc/b$

$C \rightarrow aC/a$

$\Leftarrow$

$S \rightarrow aaB/baB$

$B \rightarrow bC/b$

~~$B \Rightarrow bc/b$~~

$C \rightarrow aC/a$