

UNIT-II. Context Free Grammar (CFG) and Languages

- A regular language can be described by a regular expression
- A regular language can be described/represented by finite automata.
- Examples of regular languages are as follows

$L_1 = \{ \text{collection of all strings over } \{0,1\} \text{ which ends with } 0^3 \}$

$L_2 = \{ \text{collection of all strings over } \{a,b\} \text{ which starts with } a^3 \}$

$L_3 = \{ \text{A set of strings } \Sigma = \{a,b\}^* \text{ containing substring } ab^3 \}$

- Examples of non regular languages:

$L_1 = \{ a^p \mid p \text{ is a prime number} \}$

$L_2 = \{ a^n b^n \mid n \geq 0 \}$

$L_3 = \{ a^{i^2} \mid i \geq 1 \}$

$L_4 = \{ ww \mid w \notin \{a,b\}^* \}$ is not regular

$L_5 = \{ wuw \mid w \in \{a,b\}^* \}$ is not regular.

Need of grammar i.e., CFG \Rightarrow

1. Non regular languages can not be represented by FA or regular expressions. To represent regular and non regular languages we have to use context Free Grammar (CFG).
2. Grammar is a way to represent the languages.

Difference DFA & Grammar :

Sr.	DFA	Sr.	Grammar
1.	In DFA, languages are represented by states & transitions.	1.	language is represented using set of equations. Equations are recursive in nature.
2.	FA has set of states.	2.	It has set of variables/ Non terminals.
3.	It defines languages over alphabets.	3.	It defines languages over a set of terminals.
4.	FA has set of transitions.	4.	A grammar has set of equations or productions.
5.	<u>Example:</u>	6.	Example: Equivalent grammar
	<pre> graph LR S(()) --> X((X)) X -- a --> Y(((Y))) Y -- a --> Y </pre>		$X \rightarrow a$ $X \rightarrow aX$ $X \rightarrow b$ $X \rightarrow aX$ $X \rightarrow \epsilon$

Context Free Grammar :

- A context free grammar G is a collection of 4 tuples,
 $G = (V, T, P, S)$

- where,

V = finite set of ^{non} terminals / variables.

T = finite set of terminals.

P = finite set of productions.

S = start symbol of a grammar.

Language of a grammar :

- Every grammar generates a language.
- A word of a language is generated by applying productions a finite number of times.
- Derivation of a string should start from the start symbol and the final string should consists of terminals.
- If G is a grammar with start symbol ' S ' of terminals T , then the language of Grammar ' G ' is the set

$$L(G) = \{ w \mid w \in T^* \text{ and } S \xrightarrow[G]{*} w \}$$

Derivation of string from Grammar :

Two ways to derive string from grammar,

1. Sentential Form

2. Parse tree

1. Sentential Form :

- Derivation of string starts from start symbol through a finite application of productions. A string ' x ' derived so far consists of terminals and non terminals. A final string consists of terminals.

- Sentential form means derivation of string using production rules of grammar to expand the non terminals in sentence wise form. For derivation either rightmost or leftmost derivation way is followed.

Types of sentential form,

1. Left sentential form

2. Right sentential form.

1. Left sentential form :

- In left sentential form, leftmost nonterminal in sentential form is picked up for expansion.
- It follows leftmost derivation.

2. Rightmost Derivation Sentential Form :

- In right sentential form rightmost nonterminal of sentential form is picked up for expansion.
- It uses rightmost derivation.

Example 1: For the grammar given below:

$$S \rightarrow A1B$$

$$A \rightarrow 0A \mid \epsilon$$

$$B \rightarrow 0B \mid 1B \mid \epsilon$$

Give leftmost and rightmost derivation of string 1001.

⇒

1. Leftmost Derivation of string $w = 1001$.

$$S \rightarrow A1B$$

[∴ start from start symbol of appropriate production]

$$S \rightarrow \epsilon 1B$$

[∴ $A \rightarrow \epsilon$]

$$S \rightarrow 1B$$

[using production $B \rightarrow 0B$]

$$S \rightarrow 10B$$

[use production $B \rightarrow 0B$]

$$S \rightarrow 100B$$

[use production $B \rightarrow 1B$]

$$S \rightarrow 1001B$$

[∴ B \rightarrow 1B]

$$S \rightarrow 1001\epsilon$$

[∴ use B \rightarrow ϵ]

$$S \rightarrow \underline{1001}$$

ii. Right derivation:

$$S \rightarrow A1B$$

[∴ Use production S \rightarrow A1B]

$$S \rightarrow A1OB$$

[∴ Use B \rightarrow OB]

$$S \rightarrow A1OOB$$

[∴ Use B \rightarrow OB]

$$S \rightarrow A1OO1B$$

[∴ Use B \rightarrow 1B]

$$S \rightarrow A1OO1\epsilon$$

[∴ Use B \rightarrow ϵ]

$$S \rightarrow A1OO1$$

[∴ Use A \rightarrow ϵ]

$$S \rightarrow \epsilon 1001$$

$$\underline{S \rightarrow 1001}$$

2. Parse Tree :

- A set of derivations applied to generate a word can be represented using a tree. Such tree is known as a parse tree.

- In parse tree,

i. Root of the tree is represented by start symbol.

ii. Each interior node is represented by a non terminal or variable belonging to V.

iii. Each leaf node is represented by a terminal or ϵ .

iv. String generated by parse tree is seen from left to right.

Example 2: For the grammar,

$$S \rightarrow AIB$$

$$A \rightarrow OA|E$$

$$B \rightarrow OB|1B|E$$

Give parse tree for leftmost and rightmost derivation of the string 1001.

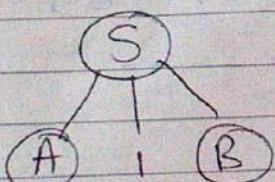
Sol: →

1. Parse Tree for leftmost derivation of 1001

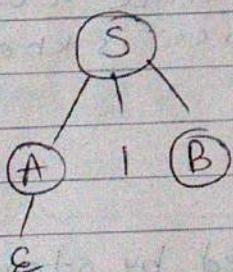
$$w = 1001$$

Step 1: (S)

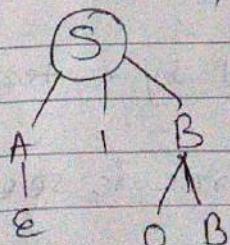
Step 2: Expand S with production rule.



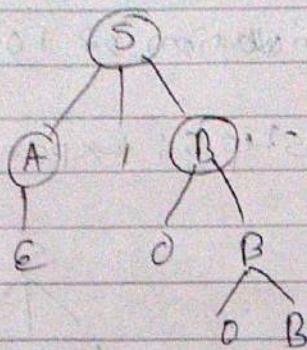
Step 3: Follow leftmost derivation & expand A by E



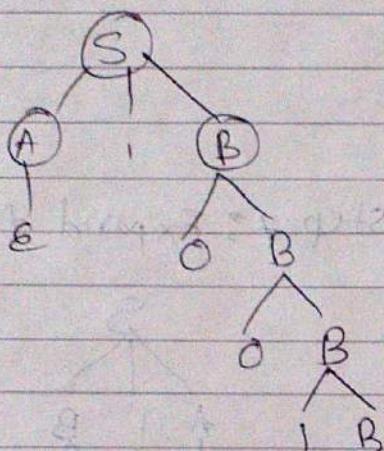
Step 4: Expand B by OB



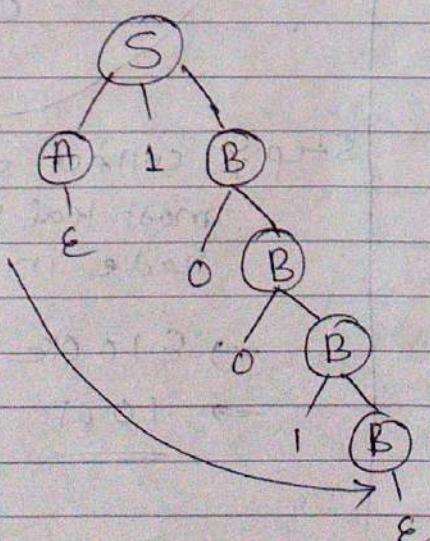
Step 5: Expand B by OB



step 5: Expand B by 1B



step 6: Expand B by 0.



Step 7: collect string from left most leaf node to right most leaf node.

→ E 1 0 0 1 E

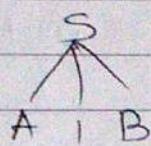
→ 1 0 0 1

ii. Parse Tree for rightmost derivation of 1001

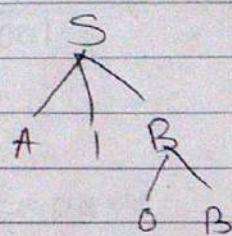
Step 1:

S

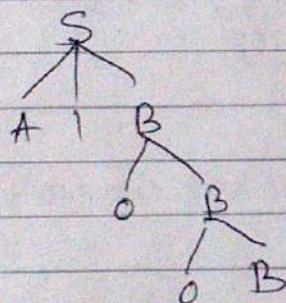
Step 2:



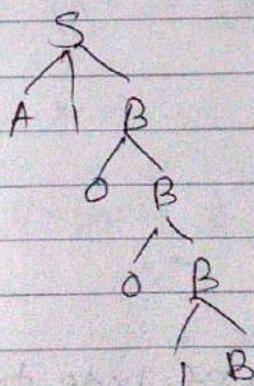
Step 3:



Step 4: Expand B by OB

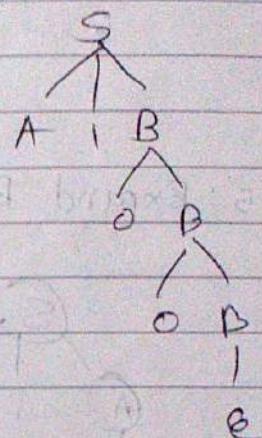


Step 5: Expand B by 1B

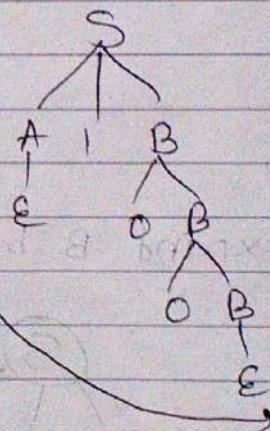


Step 6:

Step 6: Expand B by E



Step 7: Expand A by E



Step 7: collect string from left most leaf node to rightmost node in a sequence.

$\rightarrow E1001E$

$\rightarrow \underline{1001}$

Example 3: for the grammar given below,

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid a \mid b$$

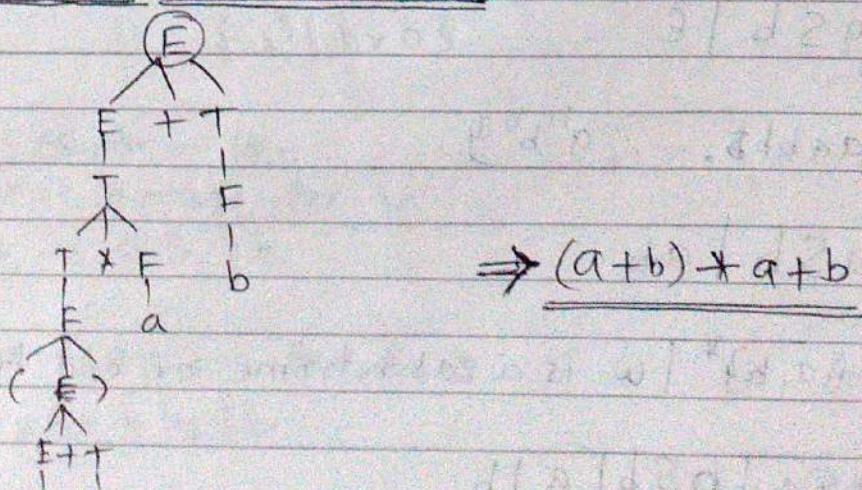
Give derivation of $(a+b)*a+b$ [Dec-2013]

Sol: →

(i) Derivation in sentential form:

$$\begin{aligned}
 &\Rightarrow E && [\because \text{use } E] \\
 &\Rightarrow E + T && [\because \text{use } E \rightarrow E + T] \\
 &\Rightarrow T + T && [\because \text{use } E \rightarrow T \cancel{\mid T}] \\
 &\Rightarrow T * F + T && [\because \text{use } T \rightarrow T * F] \\
 &\Rightarrow F * F + T && [\because \text{use } T \rightarrow F] \\
 &\Rightarrow (E) * F + T && [\because \text{use } F \rightarrow (E)] \\
 &\Rightarrow (E + T) * F + T && [\because \text{use } E \rightarrow E + T] \\
 &\Rightarrow (T + T) * F + T && [\because \text{use } E \rightarrow T] \\
 &\Rightarrow (F + T) * F + T && [\because \text{use } T \rightarrow F] \\
 &\Rightarrow (a + T) * F + T && [\because \text{use } F \rightarrow a] \\
 &\Rightarrow (a + T) * F + b && [\because \text{use } T \rightarrow F] \\
 &\Rightarrow (a + F) * F + T && [\because \text{use } F \rightarrow b] \\
 &\Rightarrow (a + b) * F + T && [\because \text{use } F \rightarrow a] \\
 &\Rightarrow (a + b) * a + T && [\because \text{use } T \rightarrow F] \\
 &\Rightarrow (a + b) * a + F && [\because \text{use } T \rightarrow b] \\
 &\Rightarrow (a + b) * a + b
 \end{aligned}$$

ii. Derivation of Parse Tree



writting Grammar for languages :

1. $L = \{ \epsilon, a, aa, aaa, aaaa, \dots \}$

Sol: \Rightarrow

$$S \rightarrow aS | \epsilon$$

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

$$\therefore G = (S, \{a\}, \{S \rightarrow aS, S \rightarrow \epsilon\}, S)$$

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

Sol: \Rightarrow

$$S \rightarrow aS | a$$

3. $L = \{b, ab, aab, aabb, \dots\}$

Sol: \Rightarrow

$$S \rightarrow aS$$

$$S \rightarrow b$$

4. $L = \{w \in \{a, b\}^*\}$

Sol: \Rightarrow

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

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

5. $L = \{\epsilon, ab, aabb, \dots, a^nb^n\}$

Sol: \Rightarrow

$$S \rightarrow aSb | \epsilon$$

6. $L = \{ab, aabb, \dots, a^nb^n\}$

Sol: \Rightarrow

$$S \rightarrow aSb | ab$$

7. $L = \{w \in \{a, b\}^* \mid w \text{ is a palindrome of odd length}\}$

Sol: \Rightarrow

$$S \rightarrow aSa | bSb | a | b$$

8. $L = \{w \in \{a, b\}^* \mid w \text{ is an even length palindrome with } |w| > 0\}$

Ans: \Rightarrow

[May-2012]

$$S \rightarrow aSa \mid bSb \mid \epsilon$$

9. Language $L = \{w \in \{a, b\}^* \mid w \text{ is a palindrome of either even length or odd length with } |w| > 0\}$

Ans: \Rightarrow

$$S \rightarrow aSa \mid bSb \mid aa \mid bb \mid a \mid b$$

10. $L = \{w \in \{a, b\}^* \mid w \text{ is an even length palindrome} \Rightarrow\}$

Ans: \Rightarrow

$$S \rightarrow aSa \mid bSb \mid \epsilon$$

Union Rule for grammar \Rightarrow

If a language L_1 is generated by a grammar with start symbol S_1 , and L_2 is generated by a grammar with start symbol S_2 then the union of the language $L_1 \cup L_2$ can be generated with start symbol S ,

$$S \rightarrow S_1 \mid S_2$$

Example 2: Let the language L_1 and L_2 given as below:

$$L_1 = \{a^n \mid n > 0\}$$

$$L_2 = \{b^n \mid n > 0\}$$

Ans: \Rightarrow

start symbol for L_1 :

$$S_1 \rightarrow aS_1 \mid a$$

start symbol for L_2 :

$$S_2 \rightarrow bS_2 \mid b$$

$$S \rightarrow S_1 \mid S_2$$

$$S_1 \rightarrow qS_1 \mid a$$

$$S_2 \rightarrow bS_2 \mid b$$

$$G = (\{S, S_1\}, \{q, b\}, \{S \rightarrow S_1 \mid S_2, S_1 \rightarrow qS_1 \mid a, S_2 \rightarrow bS_2 \mid b\}, S)$$

2. Concatenation Rule for grammar :-

If a language L_1 is generated by a grammar with start symbol S_1 , and L_2 is generated by a grammar with start symbol S_2 then concatenation (product) of the languages $L_1 \cdot L_2$ can be generated with start symbol S ,

$$S \rightarrow S_1 S_2$$

Example : Let the languages L_1 and L_2 are given as below:

$$L_1 = \{q^n \mid n > 0\}$$

$$L_2 = \{b^n \mid n > 0\}$$

Sol :-

Production for L_1 :

$$S_1 \rightarrow qS_1 \mid a$$

Production for L_2 :

$$S_2 \rightarrow bS_2 \mid b$$

Then, production for $L = L_1 \cdot L_2$,

$$S \rightarrow S_1 S_2$$

$$S_1 \rightarrow qS_1 \mid a$$

$$S_2 \rightarrow bS_2 \mid b$$

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

$$G = (\{S, S_1, S_2\}, \{q, b\}, \{S \rightarrow S_1 S_2, S_1 \rightarrow qS_1 \mid a, S_2 \rightarrow bS_2 \mid b\}, S)$$

Example 1 : Give context free grammar for the following language

$$0(0+1)^* 01(0+1)^* 1$$

Sol :- Here, for $(0+1)^*$ can be generated by the following production,

$S \rightarrow 0S_1 | 1S_1 | \epsilon$

using product rule,

$S \rightarrow 0S_1 0S_1 |$

$S_1 \rightarrow 0S_2 1S_2 | \epsilon$

Example 2: Construct CFG for the regular expression.

$$R = (0+1) 1^* (1+(01)^*)$$

Sol: \Rightarrow Grammar for the following terms,

$$(0+1) \Rightarrow S_1 \rightarrow 0 | 1$$

$$1^* \Rightarrow S_2 \rightarrow 1S_2 | \epsilon$$

$$(1+(01)^*) \Rightarrow$$

$$S_3 \rightarrow S_4 | S_5$$

$$S_4 \rightarrow 1$$

$$S_5 \rightarrow 01S_5 | \epsilon$$

\Rightarrow per union rule

$$\therefore S \rightarrow S_1 S_2 S_3$$

$$S_1 \rightarrow 0 | 1$$

$$S_2 \rightarrow 1S_2 | \epsilon$$

$$S_3 \rightarrow S_4 | S_5$$

$$S_4 \rightarrow 1$$

$$S_5 \rightarrow 01S_5 | \epsilon$$

Example 3: Give the CFG for $L = \{a^i b^j, i \leq j \leq 2i, i \geq 1\}$

Sol: \Rightarrow

- A finite number of a's followed by a finite number of b's.
- Number of b's should be at least as many as number of a's.
- Number of b's should not be more than twice the number of a's.

$$S \rightarrow aSb \quad [\because \text{number of } b's = \text{Number of a's}]$$

$$S \rightarrow aSbb \quad [\because \text{Number of } b's = 2 \times \text{Number of a's}]$$

$$S \rightarrow ablabbb$$

Example 4: Give CFG for

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

Sol:

$$i + j = q$$

$$= a^i b^j c^q$$

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

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

equal number of b's and c's

Equal number of
a's and c's.

$$S \rightarrow aSc | X$$

$$X \rightarrow bXc | bc$$

Example 5: Give CFG

$$a. (011+1)^*$$

$$b. 0^i 1^{i+k} 0^k \text{ where, } i, k \geq 0 \quad [\text{May-2009, May-2014}]$$

Sol:

$$a. \underline{(011+1)^* (01)^*}$$

$$(011+1)^* \Rightarrow A \rightarrow 011A | 1A | \epsilon$$

$$(01)^* \Rightarrow B \rightarrow 01B | \epsilon$$

$$X \rightarrow AB$$

$$A \rightarrow 011A | 1A | \epsilon$$

$$B \rightarrow 01B | \epsilon$$

$$b. 0^i 1^{i+k} 0^k \text{ where, } i, k \geq 0$$

Sol:

$$\Rightarrow 0^i 1^{i+k} 0^k$$

$$\Rightarrow 0^i 1^i 1^k 0^k$$

$$X \rightarrow 0X1 | \epsilon$$

$$Y \rightarrow 1Y0 | \epsilon$$

$$S \rightarrow XY$$

$$X \rightarrow 0X1 | \epsilon$$

$$Y \rightarrow 1Y0 | \epsilon$$

Simplified grammar,

$$S \rightarrow CA$$

$$A \rightarrow a$$

Example 2: Eliminate non generating symbols.

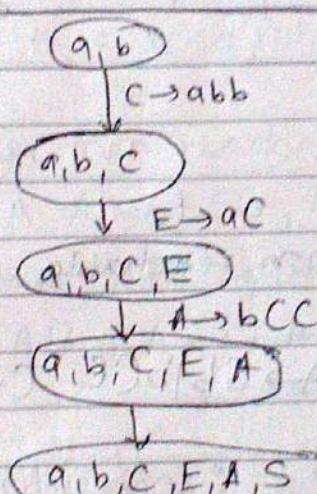
$$S \rightarrow aAA$$

$$A \rightarrow Sb \mid bCC$$

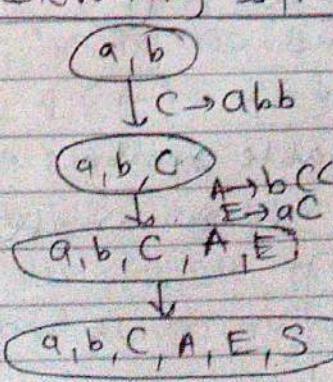
$$C \rightarrow abb$$

$$E \rightarrow aC$$

η
Soln \Rightarrow

Grammar	Generating symbols	Non generating symbols	Simplified grammar
$S \rightarrow AA$ $A \rightarrow Sb \mid bCC$ $C \rightarrow abb$ $E \rightarrow aC$		Every symbol is generating symbol	$S \rightarrow aAA$ $A \rightarrow Sb \mid bCC$ $C \rightarrow abb$ $E \rightarrow aC$

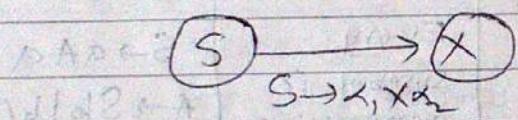
Example 3: Eliminate non generating symbols.

Grammar	Generating symbols	Non generating symbols	Simplified grammar
$S \rightarrow AAA$ $A \rightarrow Sb \mid bCC \mid DaA$ $C \rightarrow abb \mid DD$ $E \rightarrow aC$ $D \rightarrow aDa$		symbol D is non generating	$S \rightarrow AAA$ $A \rightarrow Sb \mid bCC$ $C \rightarrow abb$ $E \rightarrow aC$

2. Non reachable symbol :

- A symbol X is reachable if it can be reached from the start symbol S .
 - i.e. if $S \xrightarrow[G]{\alpha} X$ and α contains a variable X then X is reachable.
- A grammar containing a non reachable symbol V_i should be simplified by deleting every production containing the non reachable symbol V_i .
- Finding non reachable symbol.

$S \rightarrow \alpha, X\alpha_2$:- variable X is dependent on S .
 It can be represented by dependency graph as follows:

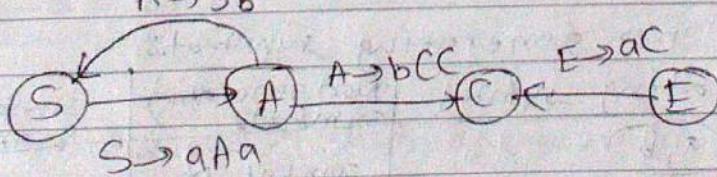


Example 1: Eliminate non reachable symbols from given grammar,

$$P = \{ S \rightarrow aAa, A \rightarrow Sb | bCC, C \rightarrow abb, E \rightarrow aC \}$$

Sol: →

Dependency graph :



Here, from S reachable symbols are A, C only.
 E is not reachable so eliminate production of E .

Simplified grammar,

$$\begin{aligned} S &\rightarrow aAa \\ A &\rightarrow Sb | bCC \\ C &\rightarrow abb \end{aligned}$$

Example 2: Eliminate non reachable symbols,

$$S \rightarrow aBa \mid BC, A \rightarrow aC \mid BCC, C \rightarrow a, B \rightarrow bCC, D \rightarrow E, E \rightarrow d$$

Sol:

Grammar	Dependency graph	Non reachable symbol	Simplified Grammar
$S \rightarrow aBa \mid BC$ $A \rightarrow aC \mid BCC$ $C \rightarrow a$ $B \rightarrow bCC$ $D \rightarrow E$ $E \rightarrow d$	<pre> graph LR S((S)) -- "S → aBa" --> B((B)) S -- "S → BC" --> C((C)) B -- "B → BCC" --> C A((A)) -- "A → aC" --> C A -- "A → BCC" --> A D((D)) -- "D → E" --> E((E)) E -- "E → d" --> d((d)) </pre>	A, D, E	$S \rightarrow aBa \mid BC$ $A \rightarrow aC \mid BCC$ $C \rightarrow a$ $B \rightarrow bCC$

Example 3: Eliminate non reachable symbols,

$$S \rightarrow aAA, A \rightarrow bBB, B \rightarrow ab, C \rightarrow aB$$

Example 4: Eliminate non reachable symbols,

$$S \rightarrow aS \mid AB, A \rightarrow bA, B \rightarrow AA$$

2. Eliminate ϵ -productions:

A production of the form $A \rightarrow \epsilon$ is called a null production or ϵ -production. For every CFG G with ϵ -productions, we can find a CFG G_1 having no ϵ -productions such that,

$$L(G_1) = L(G) - \{\epsilon\}$$

Step 1: Find nullable variable / Non terminal

Step 2: Add new all possible productions / production related to each production rule after eliminating nullable variable.

Step 3: Remove ϵ -production.

Step 1: Find Nullable variable:

A symbol X is nullable if,

$$i. X \rightarrow \epsilon$$

ii. $X \rightarrow \alpha$ and all symbols of α are nullable.

Example:

$$S \rightarrow ABA, A \rightarrow aA|\epsilon, B \rightarrow bB|\epsilon$$

A is nullable as there is production $A \rightarrow \epsilon$

B is nullable as there is a production $B \rightarrow \epsilon$

S is nullable as there is a production $S \rightarrow ABA$ with both A & B nullable.

The set of nullable symbols = {S, A, B}

Step 2: Addition of productions with nullable variables removed.

for each production of the form $A \rightarrow \alpha$, create all possible productions of the form $A \rightarrow \alpha_1$ where α_1 is obtained from α , by removing one or more occurrences of nullable variables.

Example:

$$S \rightarrow aS$$

$$S \rightarrow \epsilon$$

Nullable symbol

$$\Rightarrow \{S\}$$

Grammar after addition of effect of nullable symbols

$$S \rightarrow aS$$

$$S \rightarrow a$$

$$S \rightarrow \epsilon$$

Step 3: Remove ϵ -productions,

Final grammar,

$$S \rightarrow aS$$

$$S \rightarrow a$$

Example 1: Remove ϵ -productions,

$$S \rightarrow qS$$

$$S \rightarrow \epsilon$$

sol: \Rightarrow

Step 1: Find nullable symbols

(S)

Step 2: Add all possible productions after removing nullable symbol either one or more occurrences.

$$S \rightarrow qS$$

$$S \rightarrow q$$

$$S \rightarrow \epsilon$$

Step 3: Remove ϵ production,

$$S \rightarrow qS$$

$$S \rightarrow q$$

Example 2: Remove ϵ productions from

$$S \rightarrow ABA$$

$$A \rightarrow \epsilon$$

$$B \rightarrow \epsilon$$

sol: \Rightarrow

Step 1: Find nullable symbols,

(A, B)

\downarrow
S → ABA

(S, A, B)
nullable symbols

Step 2: Add production after removing nullable symbols.

$$S \rightarrow ABA \quad \dots \text{(i)}$$

$$A \rightarrow \epsilon \quad \dots \text{(ii)}$$

$$B \rightarrow \epsilon \quad \dots \text{(iii)}$$

$$\underline{S \rightarrow ABA} \quad \dots \text{(i)} \quad \Rightarrow$$

$$S \rightarrow ABA$$

$$S \rightarrow AB$$

$$S \rightarrow BA$$

$$S \rightarrow AA$$

$$\begin{array}{l|l} S \rightarrow A & \\ S \rightarrow B & \\ S \rightarrow \epsilon & \\ A \rightarrow \epsilon & \\ B \rightarrow \epsilon & \end{array}$$

Step 1: Remove ϵ productions,

$$S \rightarrow ABA$$

$$S \rightarrow AB$$

$$S \rightarrow BA$$

$$S \rightarrow AA$$

$$S \rightarrow A$$

$$S \rightarrow B$$

Example 3: Eliminate ϵ -productions,

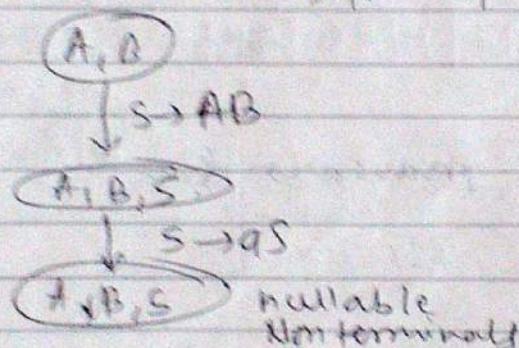
$$S \rightarrow qS \mid AB$$

$$A \rightarrow \epsilon$$

$$B \rightarrow \epsilon$$

sol:

Step 1: Find nullable symbols/non terminals.



Step 2: Add productions:

a) $S \rightarrow qS$ \Rightarrow

$$S \rightarrow qS$$

$$S \rightarrow a$$

b) $S \rightarrow AB$ \Rightarrow

$$S \rightarrow AB$$

$$S \rightarrow A$$

$$S \rightarrow B$$

c) $A \rightarrow \epsilon$
 $B \rightarrow \epsilon$
 $S \rightarrow \epsilon$

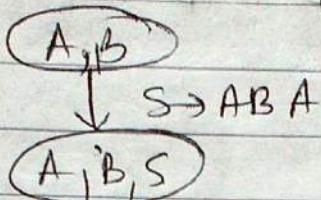
Step 3: Remove ϵ .

$$S \rightarrow qS \mid a \mid AB \mid A \mid B$$

Example 4: Remove ϵ -production,

$$\text{sgn} \Rightarrow S \rightarrow ABA, A \rightarrow aA|\epsilon, B \rightarrow bB|\epsilon$$

Step 1: Find nullable symbols



Step 2: Add production.

$$a) \underline{S \rightarrow ABA}$$

$$\begin{aligned} S &\rightarrow ABA \\ S &\rightarrow AB \\ S &\rightarrow BA \\ S &\rightarrow AA \\ S &\rightarrow A \\ S &\rightarrow B \\ S &\rightarrow \epsilon \end{aligned}$$

$$b) \underline{A \rightarrow aA}$$

$$\begin{aligned} A &\rightarrow aA \\ A &\rightarrow a \\ A &\rightarrow \epsilon \\ c) \underline{B \rightarrow bB} \\ B &\rightarrow bB \\ B &\rightarrow b \\ B &\rightarrow \epsilon \end{aligned}$$

Step 3: Remove ϵ .

$$S \rightarrow ABA | AB | BA | AA | A | B$$

$$A \rightarrow aA | a$$

$$B \rightarrow bB | b$$

Example 5: Remove ϵ -production,

$$S \rightarrow AB$$

$$A \rightarrow aAA | \epsilon$$

$$B \rightarrow bBB | \epsilon$$

sgn

$S \rightarrow AB A B$
$A \rightarrow aAA aA a$
$B \rightarrow bBB bB b$

3) Elimination of Unit production:

Example 1:

$$S \rightarrow ABA | BA | AA | AB | A | B$$

$$A \rightarrow aA | a$$

$$B \rightarrow bB | b$$

STⁿ

Here, Unit production,

$$S \rightarrow A$$

$$S \rightarrow B$$

It can be eliminated by reducing to

$$A \rightarrow aA | a \quad \& \quad B \rightarrow bB | b$$

$$\boxed{S \rightarrow ABA | BA | AA | AB | A | B}$$

$$A \rightarrow aA | a$$

$$B \rightarrow bB | b$$

Normal Forms:

- Productions in G, satisfying certain restrictions are said to be in normal form.
- complex grammar can be reduced to simple form.

There are two normal forms for CFG

1. Chomsky Normal Form (CNF)

2. Greibach Normal Form (GNF)

1. Chomsky Normal Form (CNF):

- A grammar is said to be in chomsky normal form (CNF) iff its productions are of following form.

Rules:

a) Non-terminal $\xrightarrow{\text{derives}}$ exactly two non terminals.

b) Non terminal $\xrightarrow{\text{derives}}$ one terminal

Steps to convert CFG to CNF :-

Step 1: Simplify given grammar (if required) i.e. remove E-productions / Unit productions / useless symbols.

Step 2: Restrict number of variables / Non terminals on RHS of the productions to two.

Step 3: If RHS contains combinations of variable and terminals then replace terminals by new variables.

Example:

1. $A \rightarrow V_1 V_2 a V_3 b V_4$ terminals a & b can be removed by adding $R_1 \rightarrow a$ & $R_2 \rightarrow b$
After, $A \rightarrow V_1 V_2 R_1 V_3 R_2 V_4$

2. $A \rightarrow x_1 x_2 x_3 \dots x_n$ should be broken down as given below:

$A \rightarrow x_1 C_1 \quad | \quad C_2 \rightarrow x_2 C_3 \quad | \quad C_{n-2} \rightarrow x_{n-1} x_n$ Each with the variable on right side.
 $C_1 \rightarrow x_2 C_2$

Example 1: convert CFG to CNF

$$S \rightarrow bA \mid aB$$

$$A \rightarrow bAA \mid aS \mid a$$

$$B \rightarrow aBB \mid bS \mid b$$

Sol: →

Step 1: Given productions are,

(i) $S \rightarrow bA$

(ii) $S \rightarrow aB$

(iii) $A \rightarrow bAA$

(iv) $A \rightarrow aS$

(v) $A \rightarrow a$

(vi) $B \rightarrow aBB$

(vii) $B \rightarrow bS$

(viii) $B \rightarrow b$

Step 2: Grammar is already simplified so no need to simplify the grammar.

Step 3: Production (v) & (viii) are in CNF

$$A \rightarrow a$$

$$B \rightarrow b$$

Step 4: Introduce, $R_1 \rightarrow a$, $R_2 \rightarrow b$

(i) Production, $S \rightarrow bA$:

$$S \rightarrow R_2 A$$

(ii) Production $S \rightarrow aB$:

$$S \rightarrow R_1 B$$

(iii) Production $A \rightarrow bAA$:

$$\begin{array}{c|c} A \rightarrow R_2 AA & R_4 \rightarrow R_2 A \\ R_2 \rightarrow b & A \rightarrow R_4 A \end{array}$$

(iv) Production $B \rightarrow aBB$:

$$B \rightarrow R_1 BB$$

$$R_3 \rightarrow R_1 B$$

$$B \rightarrow R_3 B$$

(v) $A \rightarrow aS$:

$$A \rightarrow R_1 S$$

(vi) $B \rightarrow bS$;

$$B \rightarrow R_2 S$$

Step 5 : Final CFG,

$$A \rightarrow a$$

$$B \rightarrow b$$

$$R_1 \rightarrow a$$

$$R_2 \rightarrow b$$

$$S \rightarrow R_2 A$$

$$S \rightarrow R_1 B$$

$$A \rightarrow R_4 A$$

$$R_4 \rightarrow R_2 A$$

$$B \rightarrow R_3 B$$

$$R_3 \rightarrow R_1 B$$

$$A \rightarrow R_1 S$$

$$B \rightarrow R_2 S$$

i.e.

$$S \rightarrow R_2 A \mid R_1 B$$

$$A \rightarrow R_4 A \mid R_1 S \mid a$$

$$B \rightarrow R_3 B \mid R_2 S \mid b$$

$$R_1 \rightarrow a$$

$$R_2 \rightarrow b$$

$$R_3 \rightarrow R_1 B$$

$$R_4 \rightarrow R_2 A$$

Example 2 : Find CNF equivalent to

$$S \rightarrow aAbB$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow bB \mid b$$

do : \Rightarrow^n

Step 1 : Grammar is simplified. So no need to

i. Eliminate ϵ productions.

ii. Elimination of unit production.

iii. Elimination of useless symbols.

Step 2 :

(i) $S \rightarrow aAbB$:

$$\begin{array}{l} R_1 \rightarrow a \\ R_2 \rightarrow b \end{array}$$

$$S \rightarrow R_1 R_2 B$$

$$S \rightarrow R_1 A R_2 B$$

$$S \rightarrow R_1 R_3$$

$$R_3 \rightarrow A R_4$$

$$R_4 \rightarrow R_2 B$$

$$(i) \quad \underline{A \rightarrow aA}^{\circ}$$

$$A \rightarrow R_1 A$$

$$(ii) \quad \underline{B \rightarrow bB}^{\circ}$$

$$B \rightarrow R_2 B$$

$$(iii) \quad \underline{B \rightarrow b}^{\circ}$$

$$B \rightarrow b$$

Step 3: Grammar in CNF,

$$S \rightarrow R_1 R_3$$

$$A \rightarrow R_1 A | a$$

$$B \rightarrow R_2 B$$

$$B \rightarrow b$$

$$R_1 \rightarrow a$$

$$R_2 \rightarrow b$$

$$R_3 \rightarrow A R_4$$

$$R_4 \rightarrow R_2 B$$

Example 3: convert the grammar in CNF form

$$S \rightarrow P Q P$$

$$P \rightarrow O P | \epsilon$$

$$Q \rightarrow I Q | \epsilon$$

[May-2013, May-2014]

Sol: \Rightarrow

Step 1: Simplify the grammar

(i) Eliminate ϵ -productions

Nullable symbols are P, Q & S

$$\text{for } S \rightarrow P Q P :$$

$$S \rightarrow P Q P$$

$$S \rightarrow P Q$$

$$S \rightarrow Q P$$

$$S \rightarrow P P$$

$$S \rightarrow P$$

$$S \rightarrow Q$$

for production $P \rightarrow O P$:

$$P \rightarrow O P$$

$$P \rightarrow O$$

for production $Q \rightarrow I Q$:

$$Q \rightarrow I Q$$

$$Q \rightarrow I$$

$$S \rightarrow \epsilon$$

$$P \rightarrow \epsilon$$

$$Q \rightarrow \epsilon$$

After eliminating ϵ production,

$$S \rightarrow P Q P | P Q | Q P | P P | P | Q$$

$$P \rightarrow O P | O$$

$$Q \rightarrow I Q | I$$

(ii) Eliminate Unit production

$S \rightarrow P$ & $S \rightarrow Q$ are unit production so after removing unit production,

$$S \rightarrow O P | O$$

$$S \rightarrow I Q | I$$

Grammar after eliminating unit production,

$$S \rightarrow P\&P \mid P\& \mid QP \mid PP \mid OP \mid O \mid I\&I$$

$$P \rightarrow OP \mid O'$$

$$Q \rightarrow I\&I$$

There is no any useless symbol.
so grammar is simplified.

Step 2: Arrange the RHS of production if it's length is ≥ 2 then it should contain all nonterminals.

Introduce,

$$R_1 \rightarrow O$$

$$R_2 \rightarrow I$$

$$S \rightarrow P\&P \mid P\& \mid QP \mid PP \mid R_1P \mid O \mid R_2Q \mid I\&I$$

$$P \rightarrow R_1P \mid O$$

$$Q \rightarrow R_2Q \mid I$$

Step 3:

SS Grammar production	Equivalent CNF
1. $S \rightarrow P\&P$	$S \rightarrow PR_3$ $R_3 \rightarrow \&P$
2. $S \rightarrow P\&$	$S \rightarrow P\&$
3. $S \rightarrow \&P$	$S \rightarrow \&P$
4. $S \rightarrow PP$	$S \rightarrow PP$
5. $S \rightarrow R_1P$	$S \rightarrow R_1P$
6. $S \rightarrow O$	$S \rightarrow O$
7. $S \rightarrow R_2Q$	$S \rightarrow R_2Q$
8. $S \rightarrow I$	$S \rightarrow I$
9. $P \rightarrow R_1P$	$P \rightarrow R_1P$
10. $P \rightarrow O$	$P \rightarrow O$
11. $Q \rightarrow R_2Q$	$Q \rightarrow R_2Q$ $Q \rightarrow I$ $R_1 \rightarrow O$ $R_2 \rightarrow I$

Step 4: Final CNF,

$$S \rightarrow PR_3 | P\bar{Q} | \bar{Q}P | PP | R_1P | O | R_2\bar{Q} | I$$

$$R_3 \rightarrow QP$$

$$P \rightarrow R_1 P | D$$

$$\bar{Q} \rightarrow R_2 Q | I$$

$$R_1 \rightarrow O$$

$$R_2 \rightarrow I$$

Example 4: check whether given grammar is in CNF or not

$$S \rightarrow bA | aB$$

$$A \rightarrow bAA | aS | a$$

$$B \rightarrow aBB | bS | b$$

If it is not in CNF and it's equivalent CNF.

[Dec-2005, May-2011, Dec-2012, Dec-2013]

Step 1: \Rightarrow

Step 1: Grammar is not in CNF because productions are not as per CNF.

Step 2: Grammar is simplified already.

Step 3: Introduce $R_1 \rightarrow b$ $R_2 \rightarrow a$

Step	Grammar	Equivalent CNF
1.	$S \rightarrow bA aB$	$S \rightarrow R_1 A R_2 B$
2.	$A \rightarrow bAA$ rewrite it as <u>$A \rightarrow R_1 AA$</u>	$A \rightarrow R_1 R_3$ $R_3 \rightarrow AA$
3.	$A \rightarrow aS a$	$A \rightarrow R_2 S a$
4.	$B \rightarrow aBB$ rewrite the grammar $B \rightarrow R_2 BB$	$B \rightarrow R_2 R_4$ $R_4 \rightarrow BB$
5.	$B \rightarrow bS b$	$B \rightarrow R_1 S b$

Step 4: Final grammar in CNF is,

$$S \rightarrow R_1 A | R_2 B$$

$$A \rightarrow R_1 R_3$$

$$R_3 \rightarrow AA$$

$$A \rightarrow R_2 S | a$$

$$B \rightarrow R_2 R_4 | R_1 S | b$$

$$R_4 \rightarrow BB$$

$$R_1 \rightarrow b$$

$$R_2 \rightarrow a$$

Example 5: Find CNF^{grammar} for the set of strings of balanced parentheses,

\Rightarrow

Step 1: $V = \{ \{S\}, \{(,)\}, \{S \rightarrow (S), S \rightarrow SS, S \rightarrow \epsilon\}, S \}$

~~$S \rightarrow (S)$~~

$S \rightarrow SS$

$S \rightarrow \epsilon$

Step 2: Simplify the grammar,
 Remove ϵ -production,

(i) $S \rightarrow (S)$:

$S \rightarrow (S)$

$S \rightarrow ()$

(ii) $S \rightarrow SS$

$$\boxed{S \rightarrow SS \\ S \rightarrow S}$$

$S \rightarrow \epsilon$

After removing ϵ production we get grammar,

$S \rightarrow (S) | () | SS | S$

$S \rightarrow S$ is unit production remove it by,

$S \rightarrow (S) | () | SS \}$

Simplified grammar,

$$\boxed{S \rightarrow (S) | () | SS}$$

Step 3: $R_1 \rightarrow C$ $R_2 \rightarrow)$

(i) $S \rightarrow (S)$

$$\boxed{S \rightarrow R_1 SR_2}$$

$$\boxed{S \rightarrow R_1 R_3}$$

$$\boxed{R_3 \rightarrow SR_2}$$

(ii) $S \rightarrow ()$

$$\boxed{S \rightarrow R_1 R_2}$$

(iii) $S \rightarrow SS$

Step 4: Final CNF

$S \rightarrow R_1 R_3 | R_1 R_2 | SS$

$R_3 \rightarrow SR_2$

$R_1 \rightarrow C$

$R_2 \rightarrow)$

Example 6: convert the following grammar to CNF

$$S \rightarrow A B a \quad \boxed{2} \quad \boxed{3}$$

$$S \rightarrow a A b$$

$$B \rightarrow A C$$

[May-2006, May-2012]

Sol:

Step 1: simplify grammar;

B is not reachable. (Remove)

$$S \rightarrow A B a$$

$$S \rightarrow a A b$$

Step 3: $R_1 \rightarrow b$ & $R_2 \rightarrow a$

$$S \rightarrow A R_1 R_2 \mid R_2 R_1 R_1$$

Step 4: CNF

$$\begin{array}{l} S \rightarrow A R_3 \\ R_3 \rightarrow R_1 R_2 \end{array}$$

$$\begin{array}{l} S \rightarrow R_2 R_4 \\ R_4 \rightarrow R_2 R_1 \end{array}$$

$$\boxed{\begin{array}{l} S \rightarrow A R_3 \mid R_2 R_4 \\ R_3 \rightarrow R_1 R_2 \\ R_4 \rightarrow R_2 R_1 \\ R_1 \rightarrow b \\ R_2 \rightarrow a \end{array}}$$

2. Grreibach Normal Form

- A grammar is said to be in Grreibach Normal form iff its productions are of following form.

- format,

Non terminal \longrightarrow one terminal followed by string of non terminals.

Steps: convert CFG to GNF

Step 1: Simplify the grammar if required.

Step 2: Strictly check first letter on RHS should be terminal followed by string of non terminals.

Step 3: If first letter is non terminal then replace by its production that are in GNF.

Example: Convert the following grammar to GNF

$$S \rightarrow ABA | AB | BA | AA | A | B$$

$$A \rightarrow aA | a$$

$$B \rightarrow bB | b$$

Sol:

Step 1: Given grammar need to simplify.

$$S \rightarrow A \quad \& \quad S \rightarrow B \text{ are unit productions,}$$

$$S \rightarrow aA | a$$

$$S \rightarrow bB | b$$

After simplification,

$$\boxed{S \rightarrow ABA | AB | BA | AA | a + a | bB | b}$$

Step 2: As per GNF,

(i) $S \rightarrow ABA$:

Replace first 'A' in RHS. by all productions of A which are in GNF

$$S \rightarrow aABA | aBA$$

(ii) $S \rightarrow AB$

$$S \rightarrow aAB | AB$$

$$(Vii) A \rightarrow aA | a$$

$$(Viii) B \rightarrow bB | b$$

Step 3: Find GNF grammar,

$$S \rightarrow aABA | aBA | aAB | aB | bBA | bA | aAA | aA | bB | b$$

(iii) $S \rightarrow BA$:

$$S \rightarrow bBA | BA$$

$$A \rightarrow aA | a$$

$$B \rightarrow bB | b$$

(iv) $S \rightarrow AA$

$$S \rightarrow aAA | AA$$

(v) $S \rightarrow aA$

(vi) $S \rightarrow bB$

Chomsky classification for grammar:

A grammar can be classified on the basis of production rules. Chomsky classified grammars into the following types:

1. Type 3 : Regular grammar
2. Type 2 : Context Free Grammar.
3. Type 1 : Context sensitive grammar.
4. Type 0 : Unrestricted grammar.

1. Type 3 or Regular Grammar :

A grammar is called type 3 or regular grammar if all its productions are of the following form:

$$A \rightarrow E$$

$$A \rightarrow a$$

$$A \rightarrow aB$$

$$A \rightarrow Ba$$

where, $a \in \Sigma$ and $A, B \in V$

A language generated by type 3 / regular grammar is called regular language.

2. Type 2 or context Free Grammars :

- A grammar is called type 2 or CFG if all its productions are of the following form $A \rightarrow \alpha$ where $A \in V$ and $\alpha \in (V \cup T)^*$.

V is a set of variables and T is a set of terminals.

- The languages generated by type 2 grammar is called a CFG.

Example:

$$A \rightarrow aAbB$$

$$B \rightarrow b \mid bBb$$

3. Type 1 or context sensitive Grammar :

- A grammar is called as type 1 or context sensitive grammar if all it's productions are of the following form.

$$\alpha \rightarrow \beta$$

where, β is atleast as long as α .

Example: The set of productions for string of the form

$$a^n b^n c^n.$$

↓ ↓ : \Rightarrow

$$S \rightarrow aSBC \mid aBC$$

$$CB \rightarrow BC$$

$$aB \rightarrow ab$$

$$bB \rightarrow bb$$

$$bC \rightarrow bc$$

$$cC \rightarrow cc$$

4. Type 0 or unrestricted Grammar :⇒

- productions can be written without any restriction in a unrestricted grammar.

- If there is a production of the form $\alpha \rightarrow \beta$ then length of α could be more than length of β .

also

- Every grammar is ~~not~~ a type 0 grammar.

- Type 2 grammar is a type 1 grammar.

- Type 3 grammar is a type 2 grammar.

Regular Grammar :

- A language accepted by FA can be described using a set of productions known as regular grammar.
- Productions of regular grammar as follows:

$$A \rightarrow a$$

$$A \rightarrow aB$$

$$A \rightarrow Ba$$

$$A \rightarrow \epsilon$$

where, $a \in T$ & $A, B \in V$

- A language generated by a regular grammar is known as regular language.
- A regular grammar could be written in two forms:

1. Right linear grammar

2. Left linear grammar

1. Right Linear Grammar :

- A right linear regular grammar will have production of the given form.

$$- A \rightarrow a, A \rightarrow aB, A \rightarrow \epsilon$$

2. Left linear grammar :

- A left linear regular grammar will have productions of the following form.

$$- A \rightarrow a, A \rightarrow Ba, A \rightarrow \epsilon$$

DFA to Right Linear Regular Grammar :-

$$1. \text{ DFA} \Rightarrow M = (Q, \Sigma, \delta, q_0, F)$$

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

Rename $q_0 \in Q$ as $S \in V$, relating start state of M with starting symbol of G .

Steps:

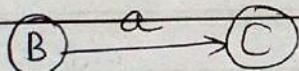
Step 1: start state of DFA becomes start symbol of grammar,

Step 2: Rename states of Q as A, B, C, D, \dots
where, $A, B, C, D, \dots \in V$

Step 3: creating set of productions P ,

1. If $\$ \in F$ then add a production where $\$ = q_0 = \text{start state}$.
 $S \rightarrow \$$ to the production set.

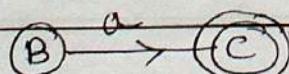
2. for every transition of the form,



then,

add a production $B \rightarrow aC$, where 'C' is a non-accepting state.

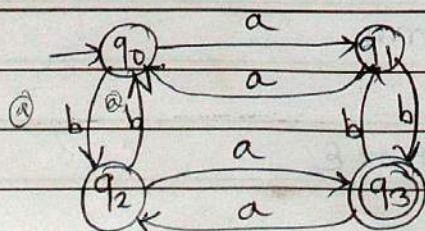
3. For every transition of the form,



then,

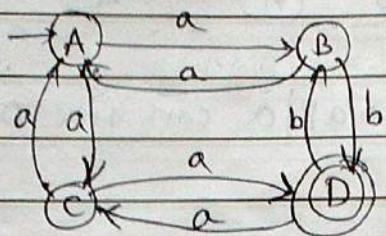
add two productions $B \rightarrow aC$, $B \rightarrow a$ where C is accepting state / final state.

Example 2: Give a right linear grammar for the following DFA.



n
Sol:→

Step 1: Rename states .



Step 2:

$$A \rightarrow aB \mid aC$$

$$B \rightarrow bD \mid aA \mid b$$

$$C \rightarrow aD \mid aA \mid a$$

$$D \rightarrow bB \mid aC$$

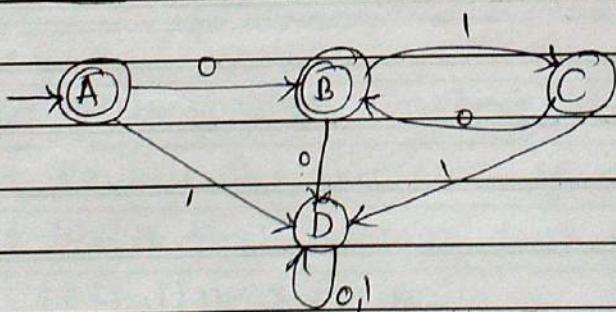
where,

set of variable , $V = \{D, A, B, C\}$

set of terminals , $T = \{a, b\}$

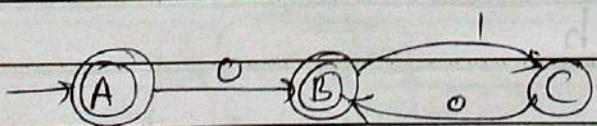
start symbol $s = A$

Example 2: Give a right linear grammar for the following DFA



Sol:→

Step 1: Here 'D' is dead state , it can be removed ,



Step 2:

$$A \rightarrow 0B \mid \epsilon \mid 0$$

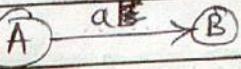
$$B \rightarrow 1C$$

$$C \rightarrow 0B \mid 0$$

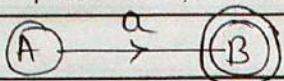
RIGHT LINEAR GRAMMAR to DFA :

1. A production of the form

$A \rightarrow aB$ can give DFA as



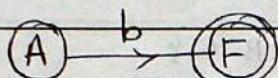
2. A production of the form $A \rightarrow aB | a$ can give DFA as,



3. A production of the form $A \rightarrow \epsilon$ can make A as a final state



4. An independent production of the form $A \rightarrow b$, will generate a transition



here, F is a new state and it should be a final state

Example 1: convert the following right linear grammar to an equivalent DFA.

$$S \rightarrow bB$$

$$B \rightarrow bC$$

$$B \rightarrow aB$$

$$C \rightarrow a$$

$$B \rightarrow b$$

Sol:

Step 1: Rewrite the productions,

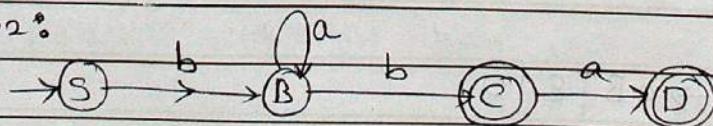
$$S \rightarrow bB$$

$$B \rightarrow bC | b$$

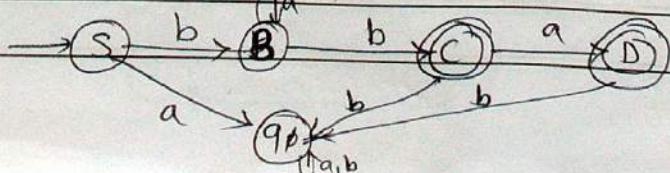
$$B \rightarrow aB$$

$$C \rightarrow a$$

Step 2:



Step 3: Add state q_ϕ to handle ϕ -transitions, we get final DFA.



(4)

Example 2 : Convert following Regular Grammar to DFA.

$$S \rightarrow 0A \mid 1B$$

$$A \rightarrow 0C \mid 1A \mid 0$$

$$B \rightarrow 1B \mid 1A \mid 1$$

$$C \rightarrow 0 \mid OA$$

Sol : \Rightarrow

Step 1:

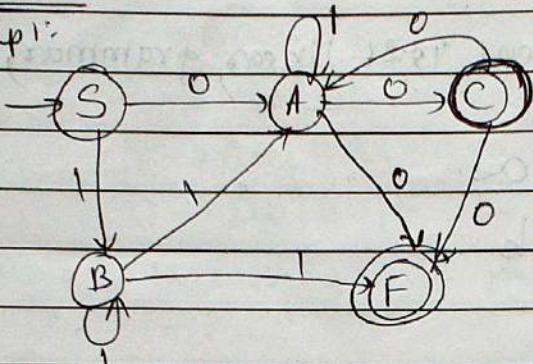
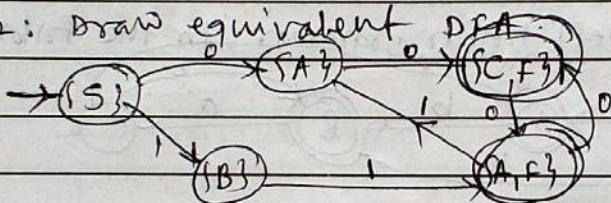


Fig-NFA.

Step 2: Draw equivalent DFA



DFA to left linear grammar : \Rightarrow

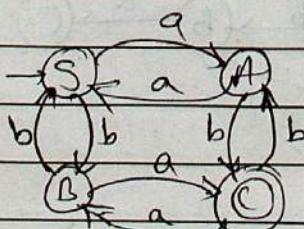
steps :

Step 1 : Interchange starting state & final state.

Step 2 : Reverse the direction of all the productions.

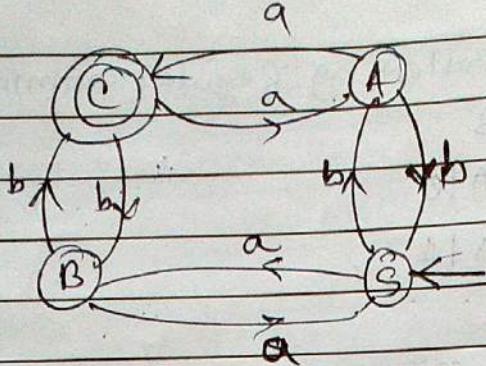
Step 3 : Write the grammar from the transition graph in left linear form.

Example 1 : Give a left linear grammar for the following DFA



Sol : \Rightarrow

Step 1 : Interchanging the starting state & the final state & reversing the direction of transitions, we get a transition graph as,



Step 2: write an equivalent left linear grammar,

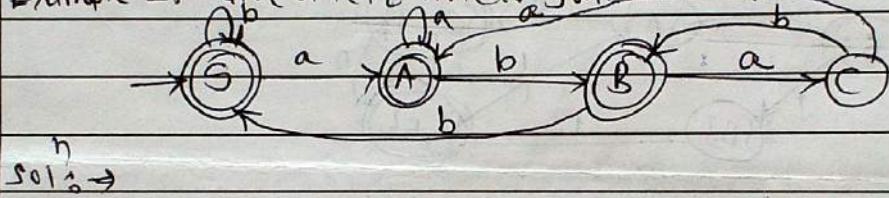
$$S \rightarrow Ba \mid Ab$$

$$A \rightarrow Sb \mid Ca \mid a$$

$$B \rightarrow Sa \mid Cb \mid b$$

$$C \rightarrow Bb \mid Aa$$

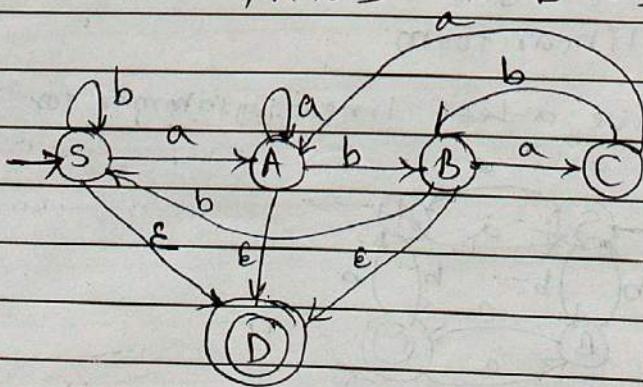
Example 2: Give a left linear grammar for the following DFA



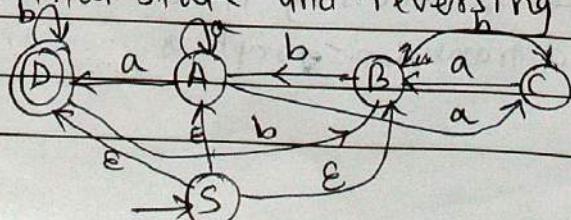
Step 1: The given DFA contains three final states

~~S, A, B.~~

We can draw an equivalent DFA with a single final state D by adding ϵ -transitions from S to D, A to D and B to D.



Step 2: Interchanging the starting state and the final state and reversing direction of transitions



Step 3 : writing an equivalent left linear grammar
We get,

$$S \rightarrow D$$

$$S \rightarrow A$$

$$S \rightarrow B$$

$$C \rightarrow Ba$$

$$B \rightarrow Cb \mid Ab$$

$$A \rightarrow Aa \mid D \mid Ca \mid a$$

$$D \rightarrow Db \mid Bb \mid b$$

Step 4 : Removing unit productions, the resulting productions
are -

$$S \rightarrow Db \mid Bb \mid b \mid Aa \mid D \mid Ca \mid a \mid cb \mid Ab$$

$$C \rightarrow Ba$$

$$B \rightarrow Cb \mid Ab$$

$$A \rightarrow Aa \mid D \mid Ca \mid a$$

$$D \rightarrow Db \mid Bb \mid b$$

Left Linear Grammar to DFA : \Rightarrow

Every left linear grammar can be represented using an equivalent DFA.

1. Draw a transition graph from the given left linear grammar.
2. Reverse the direction of all the transitions.
3. Interchange starting state & the final state.
4. carry out conversions from FA to DFA.

Example 1: construct DFA accepting the regular language generated by the left linear grammar given below:

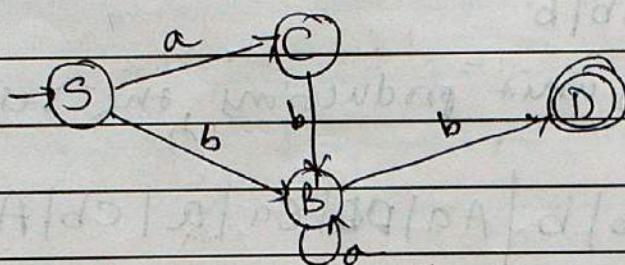
$$S \rightarrow Ca \mid Bb$$

$$C \rightarrow Bb$$

$$B \rightarrow Ba \mid b$$

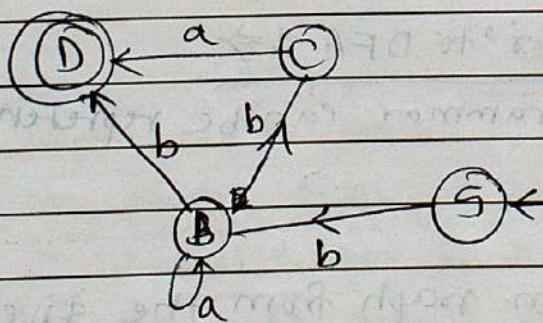
Sol: \Rightarrow

Step 1: Draw a transition graph from the given left linear grammar,

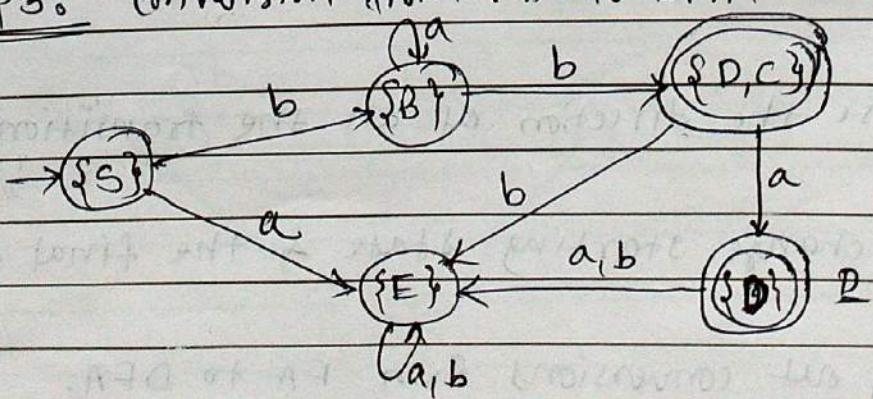


D is an accepting state. It is added for the production $B \to b$.

Step 2: Reverse the direction of transitions and interchange start & final state.



Step 3: conversion from FA to DFA.



Dead state is added to handle ϕ -transitions.

Right Linear Grammar to left linear grammar:

Steps to convert Right Linear Grammar to left linear grammar:

1. Represent the right linear grammar using a transition graph.

Mark the final state as ϵ

2. Interchange start & final state.

3. Reverse the direction of all transitions.

4. Write left linear grammar from the transition graph.

Example 4: Convert the following right linear grammar to an equivalent left linear grammar.

$$S \rightarrow bB|b$$

$$B \rightarrow bC$$

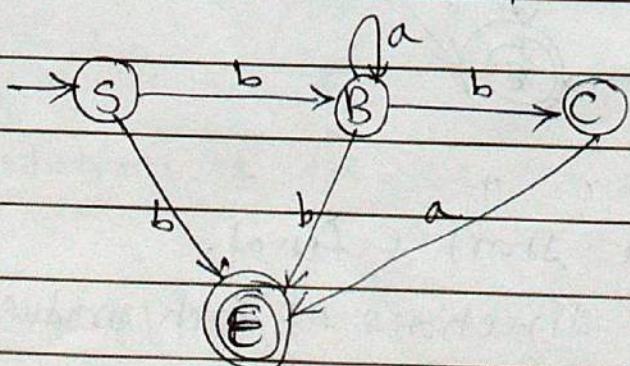
$$B \rightarrow aB$$

$$C \rightarrow a$$

$$B \rightarrow b$$

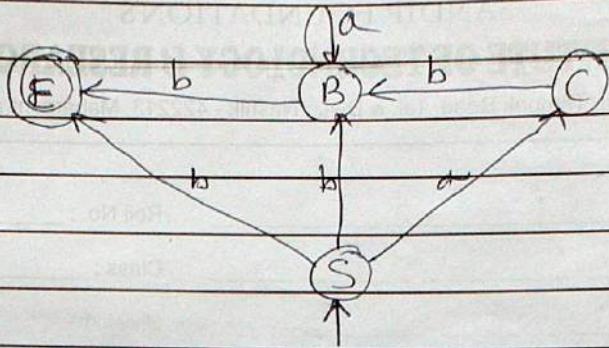
Sol: →

1. construct transition graph.



2. Interchange start and final states.

Reverse direction of every production transition.



Step 3: write left linear grammar.

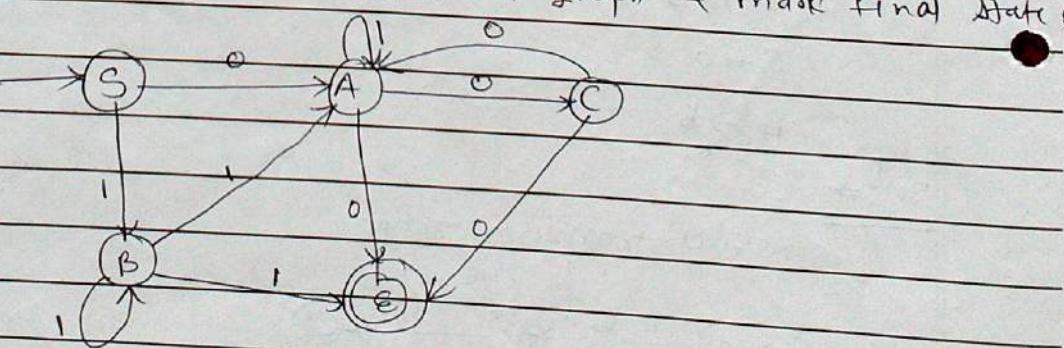
$S \rightarrow Bb Ca \cancel{Bb} \cancel{Ca}$	$S \rightarrow Bb Ca b$
$B \rightarrow Ba \cancel{Bb} \cancel{Ba}$	$B \rightarrow Ba b$
$\cancel{C} \rightarrow Bb$	$C \rightarrow Bb$

Example 2: write an equivalent left linear grammar from the given right linear grammar.

$$\begin{aligned} S &\rightarrow oA | +B \\ A &\rightarrow oC | 1A | 0 \\ B &\rightarrow 1B | 1A | 1 \\ C &\rightarrow o | oA \end{aligned}$$

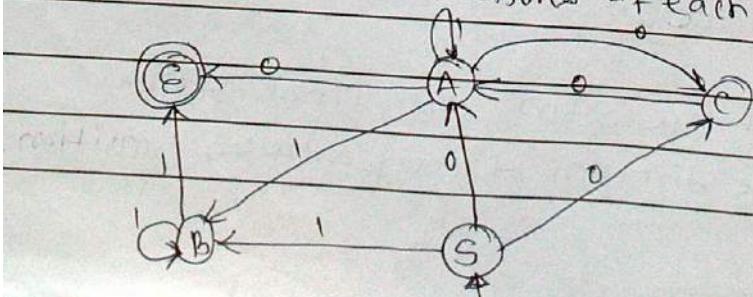
Soln: →

Step 1: construct the transition graph & mark final state ϵ



Step 2: Interchange start & final.

Reverse the directions of each production.



Step 3: write productions of left linear grammar,

$$S \rightarrow C0 \mid A0 \mid B1$$

$$A \rightarrow A1 \mid C0 \mid 0 \mid B1$$

$$B \rightarrow B1 \mid 1$$

$$C \rightarrow A0 \mid 1$$

Example 3: for right linear grammar given below, obtain an equivalent left linear grammar.

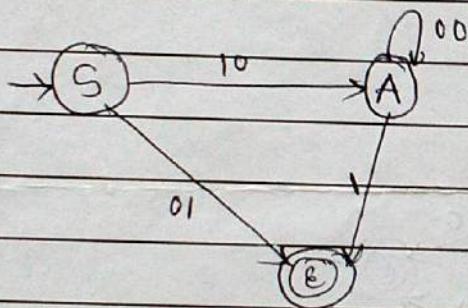
[Dec-2008, May-2009]

$$S \rightarrow 10A \mid 01$$

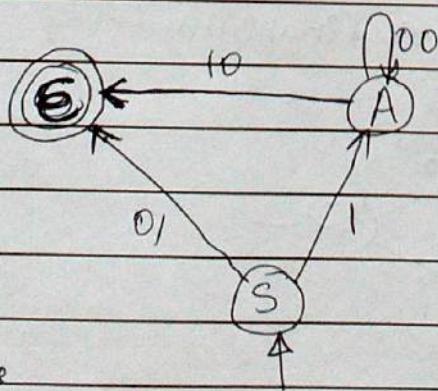
$$A \rightarrow 00A \mid 1$$

Sol: \Rightarrow

1. Transition graph & add ϵ .



2. Exchange the start & final and reverse the direction of each productions.



3. productions of left linear grammar.

$$S \rightarrow A1 \mid 01$$

$$A \rightarrow A00 \mid 10$$

Left linear to right linear grammar \Rightarrow

1. Represent the left linear grammar using a transition graph. Mark the final state as ϵ
2. Interchange the start & final state.
3. Reverse the direction of all transitions.
4. write right-linear grammar from the transition graph.

Example 2: convert right linear grammar from left linear grammar.

$$S \rightarrow C0 | A0 | B1$$

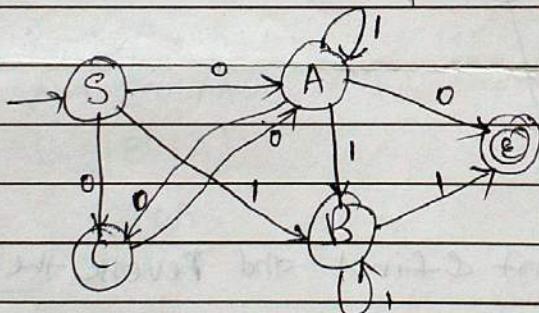
$$A \rightarrow A1 | C0 | B1 | 0$$

$$B \rightarrow B1 | 1$$

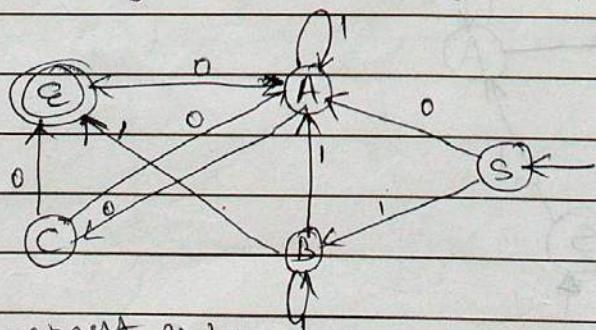
$$C \rightarrow A0$$

$\Delta G \Rightarrow$

1. construct transition graph.



2. Interchange start state & final state.



3. construct right linear grammar

$$S \rightarrow A0 | B1$$

$$A \rightarrow A1 | C0 | 0$$

$$B \rightarrow B1 | A1 | 1$$

$$C \rightarrow A0 | 0$$

$$S \rightarrow 0A | 1B$$

$$A \rightarrow 1A | 0C | 0$$

$$B \rightarrow 1B | 1A | 1$$

$$C \rightarrow 0A | 0$$