UNIT-III 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 = { collection of all strings over {0,1} which ends with 0 }

L_2 = { collection of all strings over {a,b} which starts with a }

L_3 = { A set of strings Σ = {a,b} containing substring aba }

- Examples of non regular languages:

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

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

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

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

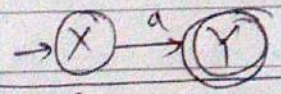
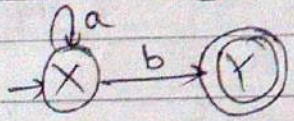
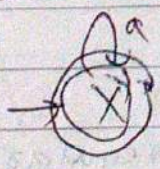
$L_5 = \{ ww^R \mid w \in \{a,b\}^* \}$ is not regular.

Need of grammar i.e. CFG ⇒

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.	Example:   	6.	Example: Equivalent gramm $X \rightarrow a$ $X \rightarrow aX$ $X \rightarrow b$ $X \rightarrow aX$ $X \rightarrow \varepsilon$

Context free Grammar :

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

- where,
  V = finite set of 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

$$L(G) = \{ w \mid w \in T^* \text{ and } S \overset{*}{\underset{G}{\Rightarrow}} w \}$$

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 'α' 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 terminal in sentence wise form. For derivation either rightmost or leftmost derivation way is followed.

Page No.
Date :

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.

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

Give leftmost and rightmost derivation of string 1001.

n

Sol :⇒

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

$S \rightarrow A|B$ [∵ start from start symbol of appropriate production]

$S \rightarrow \epsilon|B$ [∵ $A \rightarrow \epsilon$]

$S \rightarrow 1B$ [using production $B \rightarrow 0B$]

$S \rightarrow 10B$ [use production $B \rightarrow 0B$]

$S \rightarrow 100B$ [use production $B \rightarrow 1B$]

$S \rightarrow 1001B$ [∵ $B \rightarrow 1B$]
$S \rightarrow 1001\varepsilon$ [∵ use $B \rightarrow \varepsilon$]

$S \rightarrow 1001$

ii. Right derivation:

$S \rightarrow A1B$ [∵ Use production $S \rightarrow A1B$]
$S \rightarrow A10B$ [∵ Use $B \rightarrow 0B$]
$S \rightarrow A100B$ [∵ Use $B \rightarrow 0B$]
$S \rightarrow A1001B$ [∵ Use $B \rightarrow 1B$]
$S \rightarrow A1001\varepsilon$ [∵ Use $B \rightarrow \varepsilon$]
$S \rightarrow A1001$ [∵ Use $A \rightarrow \varepsilon$]
$S \rightarrow \varepsilon 1001$
$S \rightarrow 1001$

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;

$S \rightarrow A1B$

$A \rightarrow 0A \mid \varepsilon$

$B \rightarrow 0B \mid 1B \mid \varepsilon$

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

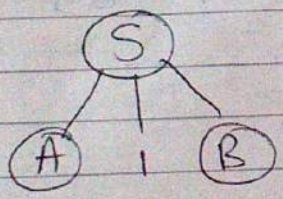
Sol^{n} :→

1. Parse Tree for leftmost derivation of 1001

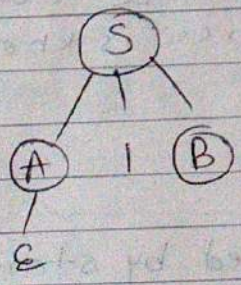
$w = 1001$

step 1 :

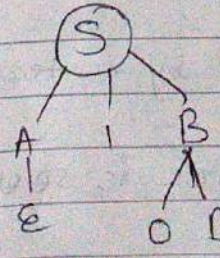
step 2: Expand S with suitable production rule.



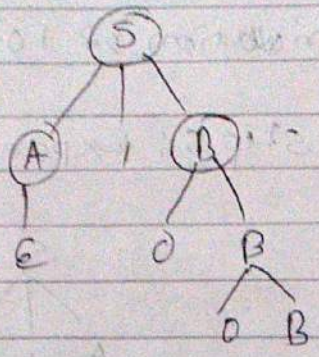
step 3: Follow leftmost derivation & expand A by ε



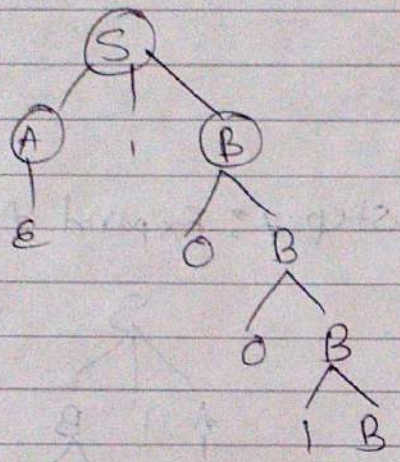
step 4: Expand B by 0B



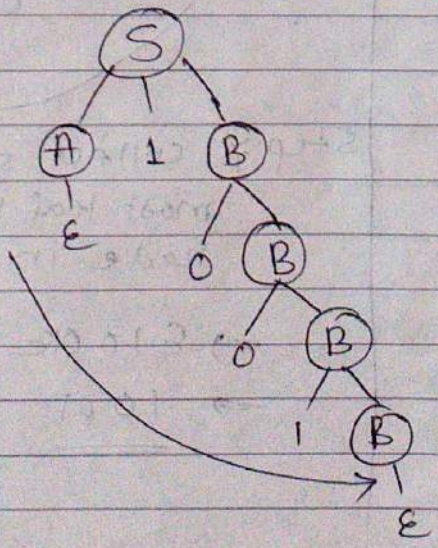
step 5: Expand B by 0B



step 5 : Expand B by 1B



step 6 : Expand B by ε.



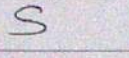
Step 7 : collect string from left most leaf node to rightmost leaf node.

⇒ ε 1 0 0 1 ε

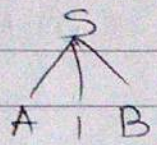
⇒ 1 0 0 1

ii. Parse Tree for rightmost derivation of 1001

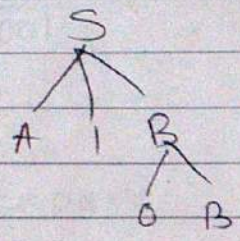
step1:



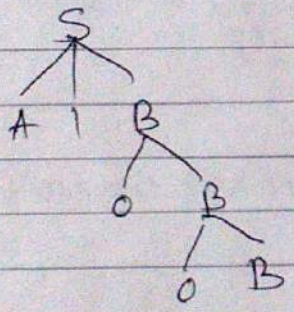
step2:



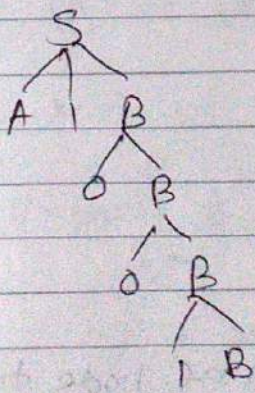
step3:



step4: Expand B by 0B

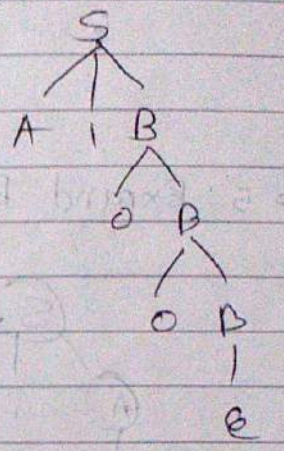


step5: Expand B by 1B

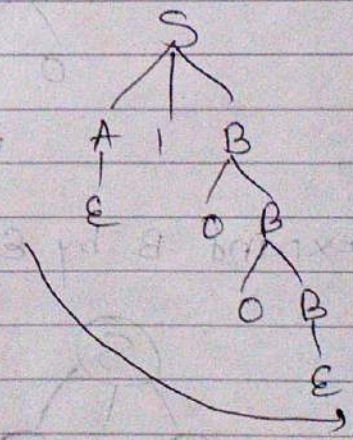


step6:

step6: Expand B by ε



step7: Expand A by ε



step7: collect string from left most leaf node to rightmost node in a sequence.

⇒ ε10 0ε

⇒ 1001

Example 3: for the grammar given below,

$E \to E+T \mid T$
$T \to T*F \mid F$
$F \to (E) \mid a \mid b$

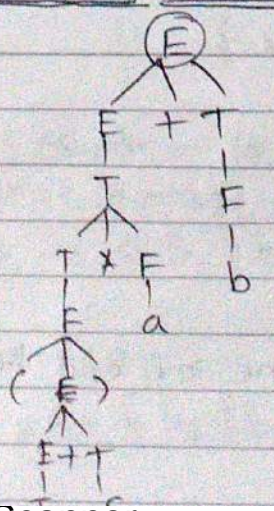
Give derivation of $(a+b)*a+b$ [Dec-2013]

Sol":→

(i) **Derivation in sentential form:**

$\Rightarrow E$ [∵ use E]
$\Rightarrow E+T$ [∵ use $E \to E+T$]
$\Rightarrow T+T$ [∵ use $E \to T$]
$\Rightarrow T*F+T$ [∵ use $T \to T*F$]
$\Rightarrow F*F+T$ [∵ use $T \to F$]
$\Rightarrow (E)*F+T$ [∵ use $F \to (E)$]
$\Rightarrow (E+T)*F+T$ [∵ use $E \to E+T$]
$\Rightarrow (T+T)*F+T$ [∵ use $E \to T$]
$\Rightarrow (F+T)*F+T$ [∵ use $T \to F$]
$\Rightarrow (a+T)*F+T$ [∵ use $F \to a$]
$\Rightarrow (a+F)*F+T$ [∵ use $T \to F$]
$\Rightarrow (a+b)*F+T$ [∵ use $F \to b$]
$\Rightarrow (a+b)*a+T$ [∵ use $F \to a$]
$\Rightarrow (a+b)*a+F$ [∵ use $T \to F$]
$\Rightarrow (a+b)*a+b$ [∵ use $F \to b$]

ii. **Derivation of Parse Tree:**



$\Rightarrow (a+b)*a+b$

Page No.
Date :

Writting Grammar for languages :

1. $L = \{\varepsilon, a, aa, aaa, aaaa, \ldots\}$

Sol^n :→

$S \to aS \mid \varepsilon$

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

$\therefore G = (S, \{a\}, \{S \to aS, S \to \varepsilon\}, S)$

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

Sol^n :→

$S \to aS \mid a$

3. $L = \{b, ab, aab, aaab, \ldots\}$

Sol^n :→

$S \to aS$

$S \to b$

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

Sol^n :→

$L = \{\varepsilon, a, b, aa, ab, bb, ba, \ldots\}$

$S \to aS \mid bS \mid \varepsilon$

5. $L = \{\varepsilon, ab, aabb, \ldots, a^n b^n\}$

Sol^n :→

$S \to aSb \mid \varepsilon$

6. $L = \{ab, aabb, \ldots, a^n b^n\}$

Sol^n :→

$S \to aSb \mid ab$

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

Sol^n :→

$S \to aSa \mid bSb \mid a \mid b$

8. $L = \{ w \in \{a,b\}^* \mid w$ is an even length palindrome with $|w| > 0 \}$ [May-2012]

Sol:⇒

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

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

Sol:⇒

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

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

Sol:⇒

$$S \rightarrow aSa \mid bSb \mid \varepsilon$$

Union 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 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 1: Let the language L_1 and L_2 given as below:

$$L_1 = \{ a^n \mid n > 0 \}$$
$$L_2 = \{ b^n \mid n > 0 \}$$

Sol:⇒

start symbol for L_1:

$$S_1 \rightarrow aS_1 \mid a$$

start symbol for L_2:

$$S_2 \rightarrow bS_2 \mid b$$

$S \to S_1 \mid S_2$

$S_1 \to aS_1 \mid a$

$S_2 \to bS_2 \mid b$

$G = (\{S, S_1\}, \{a, b\}, \{S \to S_1 \mid S_2, S_1 \to aS_1 \mid a, S_2 \to bS_2 \mid b\}, S)$

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,

$S \to S_1 S_2$

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

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

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

Sol :⇒

Production for L_1 :

$S_1 \to aS_1 \mid a$

Production for L_2 :

$S_2 \to bS_2 \mid b$

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

$S \to S_1 \cdot S_2$

$S_1 \to aS_1 \mid a$

$S_2 \to bS_2 \mid b$

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

$G = (\{S, S_1, S_2\}, \{a, b\}, \{S \to S_1 S_2, S_1 \to aS_1 \mid a, S_2 \to bS_2 \mid b\}, S)$

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 productions,

$S_1 \rightarrow 0S_1 | 1S_1 | \varepsilon$

Using product rule,

$S \rightarrow 0S_1 0 | S_1 1$

$S_1 \rightarrow 0S_1 | 1S_1 | \varepsilon$

Example 2: Construct CFG for the regular expression.

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

Sol:⇒ Grammar for the following terms,

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

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

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

$S_3 \rightarrow S_4 | S_5$

$S_4 \rightarrow 1$

$S_5 \rightarrow 01S_5 | \varepsilon$

As per union rule

$\therefore S \rightarrow S_1 S_2 S_3$

$S_1 \rightarrow 0 | 1$

$S_2 \rightarrow 1S_2 | \varepsilon$

$S_3 \rightarrow S_4 | S_5$

$S_4 \rightarrow 1$

$S_5 \rightarrow 01S_5 | \varepsilon$

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

Sol:⇒

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

$S \rightarrow aSb$ [∵ number of b's = Number of a's]

$S \rightarrow aSbb$ [∵ Number of b's = 2 × Number of a's]

$S \rightarrow ab | abb$

Example 4: Give CFG for

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

Sol:⇒

$\because i+j=q$

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

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

$\Rightarrow a^i b^j c^j c^i$

Equal number of b's and c's

Equal number of a's and c's.

$S \rightarrow aSc \mid X$

$X \rightarrow bXc \mid bc$

Example 5: Give CFG

a. $(011+1)^* (01)^*$

b. $0^i 1^{i+k} 0^k$ where, $i, k >= 0$ [May-2009, May-2014]

Sol:⇒

a. $(011+1)^* (01)^*$

$(011+1)^* \Rightarrow A \rightarrow 011A \mid 1A \mid \varepsilon$

$(01)^* \Rightarrow B \rightarrow 01B \mid \varepsilon$

$X \rightarrow AB$

$A \rightarrow 011A \mid 1A \mid \varepsilon$

$B \rightarrow 01B \mid \varepsilon$

b. $0^i 1^{i+k} 0^k$ where, $i, k >= 0$

Sol:⇒

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

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

$X \rightarrow 0X1 \mid \varepsilon$

$Y \rightarrow 1Y \mid 0 \mid \varepsilon$

$S \rightarrow XY$

$X \rightarrow 0X1 \mid \varepsilon$

$Y \rightarrow 1Y \mid 0 \mid \varepsilon$

Simplified grammar,

$S \to CA$

$A \to a$

$C \to b$

Example 2: Eliminate non generating symbols.

$S \to aAa$

$A \to Sb \mid bCC$

$C \to abb$

$E \to aC$

Sol:⇒

Grammar	Generating symbols	Non generating symbols	simplified grammar
$S \to AAa$ $A \to Sb \mid bCC$ $C \to abb$ $E \to aC$	a, b ↓ C → abb a, b, C ↓ E → aC a, b, C, E ↓ A → bCC a, b, C, E, A ↓ a, b, C, E, A, S	Every symbol is generating symbol	$S \to aAa$ $A \to Sb \mid bCC$ $C \to abb$ $E \to aC$

Example 3: Eliminate non generating symbols.

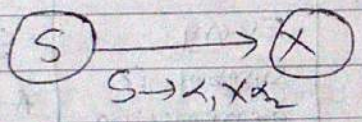
Grammar	Generating symbol	Non generating symbols	simplified grammar
$S \to aAa$ $A \to Sb \mid bCC \mid DaA$ $C \to abb \mid DD$ $E \to aC$ $D \to aDa$	a, b ↓ C → abb a, b, C ↓ A → bCC, E → aC a, b, C, A, E ↓ a, b, C, A, E, S	Symbol D is non generating	$S \to aAa$ $A \to Sb \mid bCC$ $C \to abb$ $E \to aC$

2. Non reachable symbol :

- A symbol X is reachable if it can be reached from the start symbol S.
  i.e. if $S \overset{*}{\underset{G}{\Rightarrow}} \alpha$ 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.
  $S \rightarrow \alpha_1 X \alpha_2$ ∴ variable X is dependent on S.
  It can be represented by dependency graph as follows:

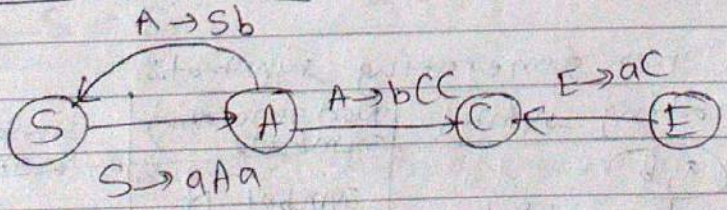


Example 1: Eliminate non reachable symbols from given grammar,

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

Sol:→

Dependency graph :



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

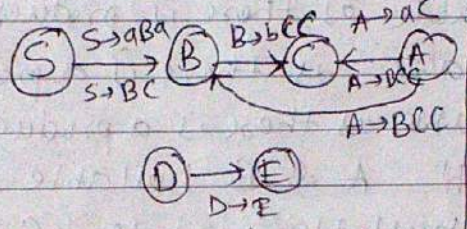
Simplified grammar,

$S \rightarrow aAa$
$A \rightarrow Sb | bCC$
$C \rightarrow abb$

Example 2: Eliminate non reachable symbols,

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

Sol:

Grammar	Dependency graph	Non reachable symbol	Simplified Grammar
$S \to aBa \mid BC$ $A \to aC \mid BCC$ $C \to a$ $B \to bCC$ $D \to E$ $E \to d$		A, D, E	$S \to aBa \mid BC$ ~~$A \to aC \mid BCC$~~ $C \to a$ $B \to bCC$

Example 3: Eliminate non reachable symbols,

$S \to aAa$, $A \to bBB$, $B \to ab$, $C \to aB$

Example 4: Eliminate non reachable symbols,

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

Eliminate ε-productions:

A production of the form $A \to \varepsilon$ 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-production such that,

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

Step1: Find nullable variable/Non terminal

Step2: Add new all possible productions ~~productions~~ related to each production rule after eliminating nullable variable.

Step3: Remove ε-production.

Step 1: Find Nullable variable:

A symbol X is nullable if,

i. $X \to \varepsilon$

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

Example:

$S \to ABA$, $A \to aA | \varepsilon$, $B \to bB | \varepsilon$

A is nullable as there is production $A \to \varepsilon$

B is nullable, as there is a production $B \to \varepsilon$

S is nullable as there is a production $S \to ABA$ 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 $A \to \alpha_1$, create all possible productions of the form $A \to \alpha_2$ where α_2 is obtained from α_1 by removing one or more occurrences of nullable variables.

Example:

	Nullable symbol	Grammar after addition & effect of nullable symbols
$S \to aS$	$\to \{S\}$	$S \to aS$
$S \to \varepsilon$		$S \to a$
		$S \to \varepsilon$

Step 3: Remove ε-productions,

final grammar,

$S \to aS$

$S \to a$

Example 1: Remove ε-productions,

$S \to aS$

$S \to \varepsilon$

Sol^n :⇒

Step 1: Find nullable symbols

(S)

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

$S \to aS$

$S \to a$

$S \to \varepsilon$

Step 3: Remove ε productions,

$S \to aS$

$S \to a$

Example 2 :⇒ Remove ε productions from

$S \to ABA$

$A \to \varepsilon$

$B \to \varepsilon$

Sol^n :⇒

Step 1: Find nullable symbols,

(A, B)

$S \to ABA$

(S, A, B) nullable symbols

Step 2: Add productions after removing nullable symbols.

$S \to ABA$ ---(i)

$A \to \varepsilon$ ---(ii)

$B \to \varepsilon$ ---(iii)

$S \to ABA$ ---(i) :⇒

$S \to ABA$	ε	$S \to A$
$S \to AB$		$S \to B$
$S \to BA$		$S \to \varepsilon$
$S \to AA$		$A \to \varepsilon$
		$B \to \varepsilon$

step3: Remove ε productions,

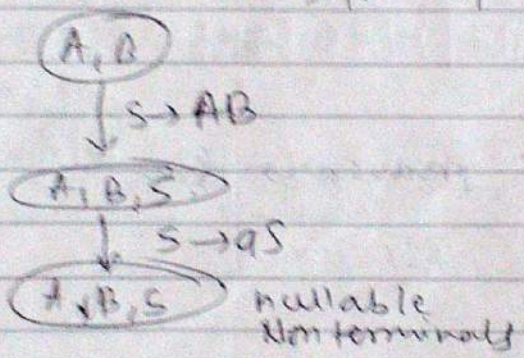
S → ABA
S → AB
S → BA
S → AA
S → A
S → B

Example 3: Eliminate ε-productions,

S → aS | AB
A → ε
B → ε

Sol :→

step1: Find nullable symbols/non terminals.



step2: Add productions:

a) S → aS :→

S → aS
S → a

b) S → AB :→

S → AB
S → A
S → B

c) A → ε
B → ε
S → ε

step3: Remove ε

S → aS | a | AB | A | B

Example 4: Remove ε-production,

$S \to ABA$, $A \to aA \mid \varepsilon$, $B \to bB \mid \varepsilon$

Sol^n :⇒

Step1 : Find nullable symbol

A, B

↓ $S \to ABA$

A, B, S

Step2 : Add production.

a) $S \to ABA$:

$S \to ABA$
$S \to AB$
$S \to BA$
$S \to AA$
$S \to A$
$S \to B$
$S \to \varepsilon$

b) $A \to aA$:

$A \to aA$
$A \to a$
$A \to \varepsilon$

c) $B \to bB$:

$B \to bB$
$B \to b$
$B \to \varepsilon$

Step3 : Remove ε.

$S \to ABA \mid AB \mid BA \mid AA \mid A \mid B$
$A \to aA \mid a$
$B \to bB \mid b$

Example 5: Remove ε-production,

$S \to AB$
$A \to aAA \mid \varepsilon$
$B \to bBB \mid \varepsilon$

Sol^n :⇒

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

3) Elimination of Unit production:

Example 1:

$S \to ABA \mid BA \mid AA \mid AB \mid A \mid B$

$A \to aA \mid a$

$B \to bB \mid b$

Here, Unit production,

$S \to A$
$S \to B$ } It can be eliminated by reducing t

$A \to aA \mid a$ & $B \to bB \mid b$

$S \to ABA \mid BA \mid AA \mid aA \mid a \mid bB \mid b$

$A \to aA \mid a$

$B \to bB \mid b$

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. Griebach Normal form (GNF)

1. Chomsky Normal Form (CNF) :

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

- Rules:

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

OR

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

Steps to convert CFG to CNF :⇒

step1 : Simplify given grammar (if required) i.e. remove ε-productions / Unit productions / Useless symbols.

step2 : Restrict number of variable / Non terminals on RHS of the productions to two.

step3 : If RHS contains combinations of variable and terminals then replace terminals by new variables.

Example :

1. $A \to V_1 V_2 a V_3 b V_4$ terminals a & b can be removed by adding $R_1 \to a$ & $R_2 \to b$
   After, $A \to V_1 V_2 R_1 V_3 R_2 V_4$
2. $A \to X_1 X_2 X_3 \cdots X_n$ should be broken down as given below:
   $A \to X_1 C_1 \mid C_2 \to X_3 C_3 \cdots C_{n-2} \to X_{n-1} X_n$ Each with two variables on right side.

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 $S \rightarrow bA$:

$S \rightarrow R_2A$

(ii) Production $S \rightarrow aB$:

$S \rightarrow R_1B$

(iii) Production $A \rightarrow bAA$:

$A \rightarrow R_2AA$

$R_2 \rightarrow b$

$R_4 \rightarrow R_2A$

$A \rightarrow R_4A$

(iv) Production $B \rightarrow aBB$:

$B \rightarrow R_1BB$

$R_3 \to R_1 B$

$B \to R_3 B$

(v) $A \to aS$:

$A \to R_1 S$

(vi) $B \to bS$:

$B \to R_2 S$

Step 5 : Final CFG,

$A \to a$
$B \to b$
$R_1 \to a$
$R_2 \to b$
$S \to R_2 A$
$S \to R_1 B$
$A \to R_4 A$
$R_4 \to R_2 A$
$B \to R_3 B$
$R_3 \to R_1 B$
$A \to R_1 S$
$B \to R_2 S$

i.e.

$S \to R_2 A \mid R_1 B$
$A \to R_4 A \mid R_1 S \mid a$
$B \to R_3 B \mid R_2 S \mid b$
$R_1 \to a$
$R_2 \to b$
$R_3 \to R_1 B$
$R_4 \to R_2 A$

Example 2 : Find CNF equivalent to

$S \to aAbB$
$A \to aA \mid a$
$B \to bB \mid b$

n
do :⇒

Step 1 : Grammar is simplified. So no need to
i. Eliminate ε productions.
ii. Elimination of unit productions.
iii. Elimination of useless symbols.

step 2 :

(i) $S \to aAbB$:

$R_1 \to a$
$R_2 \to b$

~~$S \to R_1 A R_2 B$~~

$S \to R_1 A R_2 B$

$S \to R_1 R_3$
$R_3 \to A R_4$
$R_4 \to R_2 B$

(i) $A \to aA$:
$A \to R_1 A$

(ii) $B \to bB$:
$B \to R_2 B$

(iii) $B \to b$
$B \to b$

Step3: Grammar in CNF,

$S \to R_1 R_3$
$A \to R_1 A \mid a$
$B \to R_2 B$
$B \to b$
$R_1 \to a$
$R_2 \to b$
$R_3 \to A R_4$
$R_4 \to R_2 B$

Example 3: convert the grammar in CNF form

$S \to PQP$
$P \to 0P \mid \varepsilon$
$Q \to 1Q \mid \varepsilon$ [May-2013, May-2014]

Sol: $\Rightarrow$

Step 1: simplify the grammar

(i) Eliminate ε-productions

Nullable symbols are P, Q & S

for $S \to PQP$:
$S \to PQP$
$S \to PQ$
$S \to QP$
$S \to PP$
$S \to P$
$S \to Q$

for production $P \to 0P$:
$P \to 0P$
$P \to 0$

for production $Q \to 1Q$:
$Q \to 1Q$
$Q \to 1$

$S \to \varepsilon$
$P \to \varepsilon$
$Q \to \varepsilon$

After eliminating ε production,

$S \to PQP \mid PQ \mid QP \mid PP \mid P \mid Q$
$P \to 0P \mid 0$
$Q \to 1Q \mid 1$

(ii) Eliminate unit production

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

$S \to 0P \mid 0$
$S \to 1Q \mid 1$

Grammar after eliminating unit productions,

$S \to PQP \mid PQ \mid QP \mid PP \mid OP \mid 0 \mid 1Q \mid 1$

$P \to OP \mid 0$

$Q \to 1Q \mid 1$

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,

$R_1 \to 0$

$R_2 \to 1$

$S \to PQP \mid PQ \mid QP \mid PP \mid R_1P \mid 0 \mid R_2Q \mid 1$

$P \to R_1P \mid 0$

$Q \to R_2Q \mid 1$

step 3:

Sr	Grammar production	Equivalent CNF
1.	$S \to PQP$	$S \to PR_3$ $R_3 \to QP$
2.	$S \to PQ$	$S \to PQ$
3.	$S \to QP$	$S \to QP$
4.	$S \to PP$	$S \to PP$
5.	$S \to R_1P$	$S \to R_1P$
6.	$S \to 0$	$S \to 0$
7.	$S \to R_2Q$	$S \to R_2Q$
8.	$S \to 1$	$S \to 1$
9.	$P \to R_1P$	$P \to R_1P$
10.	$P \to 0$	$P \to 0$
11.	$Q \to R_2Q$	$Q \to R_2Q$ $Q \to 1$ $R_1 \to 0$ $R_2 \to 1$

step4: Final CNF,

$S \to PR_3 | PQ | QP | PP | R_1P | 0 | R_2Q | 1$

$R_3 \to QP$

$P \to R_1P | 0$

$Q \to R_2Q | 1$

$R_1 \to 0$

$R_2 \to 1$

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

$S \to bA | aB$

$A \to bAA | aS | a$

$B \to aBB | bS | b$

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

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

Sol:⇒

step1: Grammar is not in CNF because productions are not as per CNF.

step2: Grammar is simplified already.

step3: Introduce $R_1 \to b$ $R_2 \to a$

Sr.	Grammar	Equivalent CNF
1.	$S \to bA \mid aB$	$S \to R_1A \mid R_2B$
2.	$A \to bAA$ rewrite it as $A \to R_1AA$	$A \to R_1R_3$ $R_3 \to AA$
3.	$A \to aS \mid a$	$A \to R_2S \mid a$
4.	$B \to aBB$ rewrite the grammar $B \to R_2BB$	$B \to R_2R_4$ $R_4 \to BB$
5.	$B \to bS \mid b$	$B \to R_1S \mid b$

step4: Final grammar in CNF is,

$S \to R_1A \mid R_2B$

$A \to R_1R_3$

$R_3 \to AA$

$A \to R_2S \mid a$

$B \to R_2R_4 \mid R_1S \mid b$

$R_4 \to BB$

$R_1 \to b$

$R_2 \to a$

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

Sol: ⇒

Step1: $V = \{\{S\}, \{(,)\}, \{S \to (S),\ S \to SS,\ S \to \varepsilon\}, S\}$

$S \to (S)$

$S \to SS$

$S \to \varepsilon$

Step2: Simplify the grammar,

Remove ε-production,

(i) $S \to (S)$:

$S \to (S)$

$S \to ()$

(ii) $S \to SS$

$S \to SS$

$S \to S$

$S \to \varepsilon$

After removing ε production we get grammar,

$S \to (S) \mid () \mid SS \mid S$

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

$S \to (S) \mid () \mid SS$

Simplified grammar,

$S \to (S) \mid () \mid SS$

Step3: $R_1 \to ($ $R_2 \to)$

(i) $S \to (S)$

$S \to R_1 S R_2$

$S \to R_1 R_3$

$R_3 \to S R_2$

(ii) $S \to ()$

$S \to R_1 R_2$

(iii) $S \to SS$

Step4: Final CNF

$S \to R_1 R_3 \mid R_1 R_2 \mid SS$

$R_3 \to S R_2$

$R_1 \to ($

$R_2 \to)$

Example 6: convert the following grammar to CNF

$S \to Aba$
$S \to aab$
$B \to Ac$ [May-2006, May-2012]

Sol:-

step1: simplify grammar;
B is not reachable. (Remove)

$S \to Aba$
$S \to aab$

step3: $R_1 \to b$ & $R_2 \to a$

$S \to AR_1R_2 \mid R_2R_2R_1$

step4: CNF

$S \to AR_3$ | $S \to R_2R_4$
$R_3 \to R_1R_2$ | $R_4 \to R_2R_1$

$S \to AR_3 \mid R_2R_4$
$R_3 \to R_1R_2$
$R_4 \to R_2R_1$
$R_1 \to b$
$R_2 \to a$

2. Greibach Normal Form:

- A grammar is said to be in Greibach Normal form iff it's productions are of following form.
- format,

Non terminal ⟶ one terminal followed by string of non terminals.

Steps: Convert CFG to GNF

step1: simplify the grammar if required

step2: Strictly check first letter on RHS should be terminal followed by string of non terminals.

step3: If first letter is non terminal then replace by it's production that are in GNF.

Example: convert the following grammar to GNF

$S \to ABA \mid AB \mid BA \mid AA \mid A \mid B$

$A \to aA \mid a$

$B \to bB \mid b$

Sol:→

step1: Given grammar need to simplify.

$S \to A$ & $S \to B$ are unit productions,

$S \to aA \mid a$

$S \to bB \mid b$

After simplification,

$S \to ABA \mid AB \mid BA \mid AA \mid aA \mid a \mid bB \mid b$

$A \to aA \mid a$

$B \to bB \mid b$

step2: As per GNF,

(i) $S \to ABA$:

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

$S \to aABA \mid aBA$

(ii) $S \to AB$

$S \to aAB \mid aB$

(iii) $S \to BA$:

$S \to bBA \mid bA$

(iv) $S \to AA$

$S \to aAA \mid aA$

(v) $S \to aA \mid a$

(vi) $S \to bB \mid b$

(vii) $A \to aA \mid a$

(viii) $B \to bB \mid b$

step3: Final GNF grammar,

$S \to aABA \mid aBA \mid aAB \mid aB \mid bBA \mid bA \mid aAA \mid aA \mid bB \mid b$

$A \to aA \mid a$

$B \to bB \mid b$

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 it's productions are of the following forms:

$A \to \varepsilon$

$A \to a$

$A \to aB$

$A \to Ba$

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 Grammar:

- A grammar is called type 2 or CFG if all it's productions are of the following form $A \to \alpha$ 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:

$A \to aAbB$

$B \to b \mid bBb$

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

Sol :⇒

$$S \rightarrow aSBC \mid aBC$$
$$CB \rightarrow BC$$
$$aB \rightarrow ab$$
$$bB \rightarrow bb$$
$$bC \rightarrow bc$$
$$cC \rightarrow cc$$

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.

- Every grammar also is 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 :

$A \to a$

$A \to aB$

$A \to Ba$

$A \to \varepsilon$

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.

- $A \to a$, $A \to aB$, $A \to \varepsilon$

2. Left linear grammar :⇒

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

- $A \to a$, $A \to Ba$, $A \to \varepsilon$

DFA to Right Linear Regular Grammar :⇒

1. DFA ⇒ $M = (Q, \Sigma, \delta, q_0, F)$

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

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

Steps:

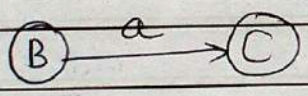
step1: start state of DFA becomes start symbol of grammar,

step2: Rename states of Q as $A, B, C, D, \ldots$
where, $A, B, C, D, \ldots \in V$

step3: creating set of productions P,

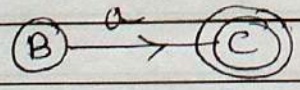
1. If ~~q_0~~ $S \in F$ then add a production. where ~~q_0~~ S = start state.
$S \to \varepsilon$ to the production ~~set~~.

2. ~~for every transition of the form,~~



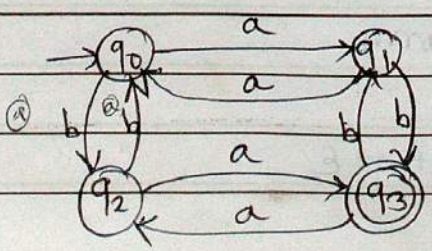
then,
add a production $B \to aC$, where 'C' is a non-accepting ~~state~~.

3. For every transition of the form,



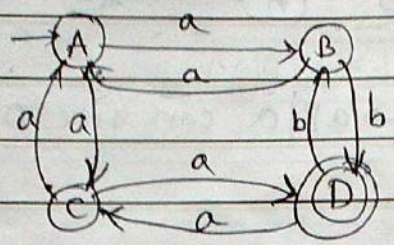
then,
add two productions $B \to aC$, $B \to a$ where c is accepting state / final state.

Example 1: Give a right linear grammar for the following DFA.



Sol:→

step1 : Rename states.



step2 :

$A \rightarrow aB \mid aC$

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

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

$D \rightarrow bB \mid aC$

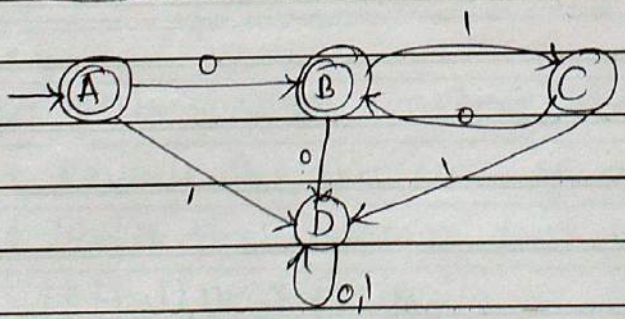
where,

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

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

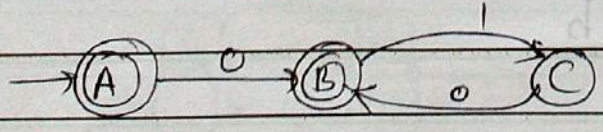
start symbol, $S = A$

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



Sol:⇒

step1 : Here 'D' is dead state, it can be removed,



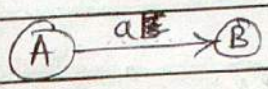
step2 :

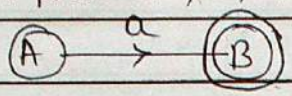
$A \rightarrow 0B \mid \varepsilon \mid 0$

$B \rightarrow 1C$

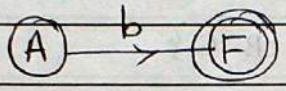
$C \rightarrow 0B \mid 0$

RIGHT LINEAR GRAMMAR to DFA :→

1. A production of the form
$A \to aB$ can give DFA as 

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


3. A production of the form $A \to \varepsilon$ can make A as a final ~~state~~ Start ~~Start~~


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


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

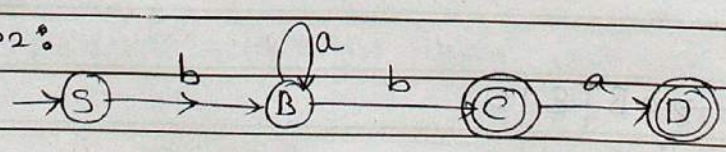
Example 1: convert the following right linear grammar to an equivalent DFA.

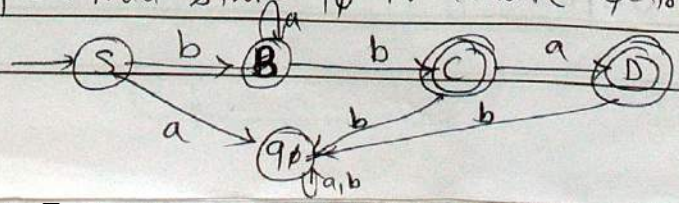
$S \to bB$
$B \to bC$
$B \to aB$
$C \to a$
$B \to b$

Sol:→

Step1: Rewrite the productions,

$S \to bB$
$B \to bC \mid b$
$B \to aB$
$C \to a$

Step2:


Step3: Add state q_ϕ to handle ϕ-transitions, we get final DFA.


Example 2: convert following Regular Grammar to DFA.

$S \rightarrow 0A \mid 1B$

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

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

$C \rightarrow 0 \mid 0A$

Soln:⇒

Step 1:

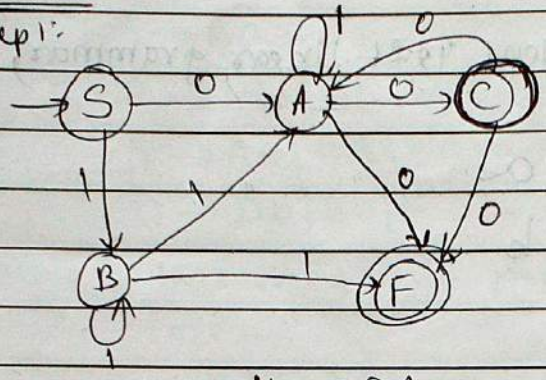
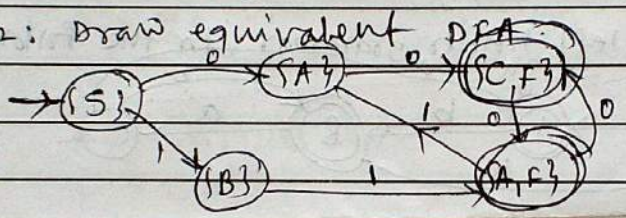


fig. NFA.

Step 2: Draw equivalent DFA.



DFA to left linear grammar:⇒

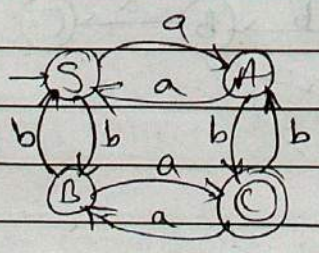
steps:

step 1: Interchange starting state & final state.

step 2: Reverse the direction of all the productions.

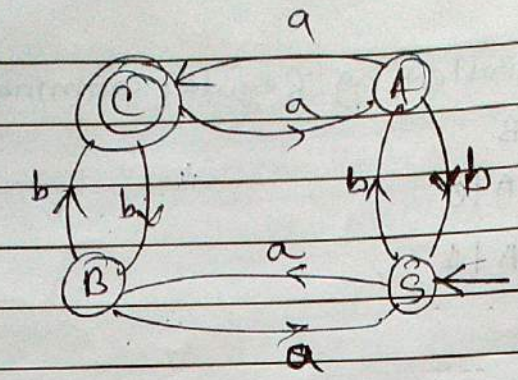
step 3: Write the grammar from the transition graph in left linear form.

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



Soln:⇒

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



step2: write an equivalent left linear grammar,

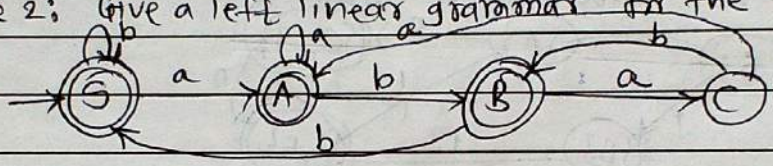
$S \rightarrow Ba \mid Ab$

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

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

$C \rightarrow Bb \mid Aa$

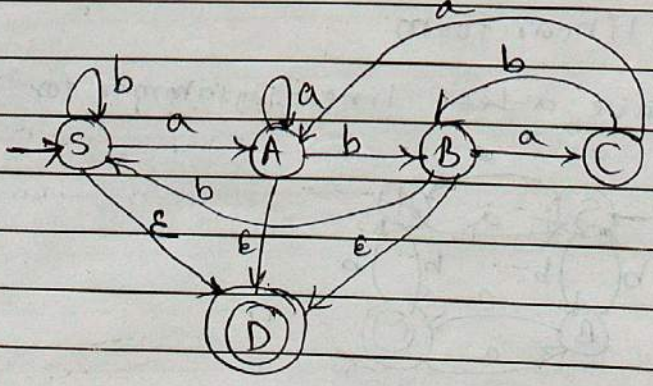
Example 2: Give a left linear grammar for the following DFA.



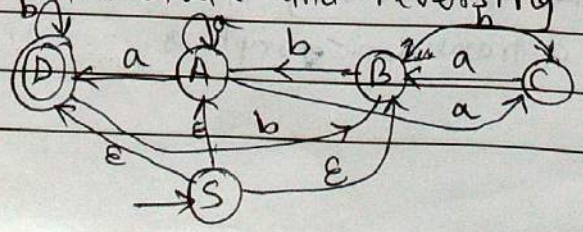
Sol:→

step1: 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.



step2: Interchanging the starting state and the final state and reversing direction of transitions



Step3 : writing an equivalent left linear grammar we get,

$S \rightarrow D$

$S \rightarrow A$

$S \rightarrow B$

$C \rightarrow Ba$

$B \rightarrow Cb \mid Ab$

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

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

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

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

$C \rightarrow Ba$

$B \rightarrow Cb \mid Ab$

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

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

Left Linear Grammar to DFA :⇒

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:

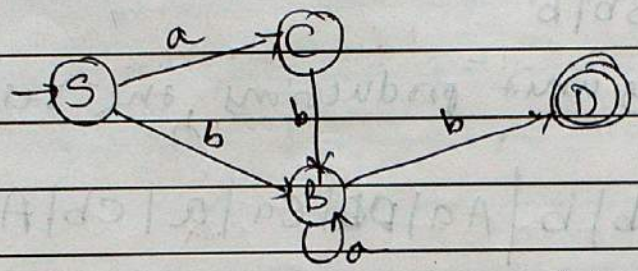
$S \rightarrow Ca \mid Bb$

$C \rightarrow Bb$

$B \rightarrow Ba \mid b$

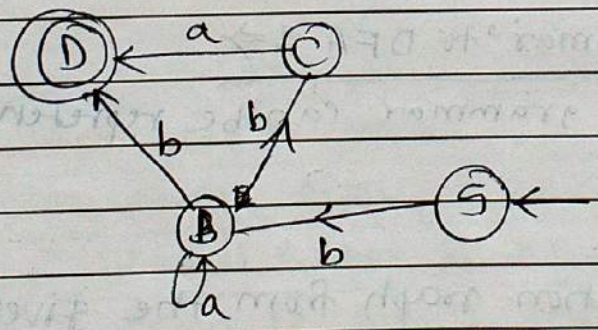
Sol:⇒

step1: Draw a transition graph from the given left linear grammar,

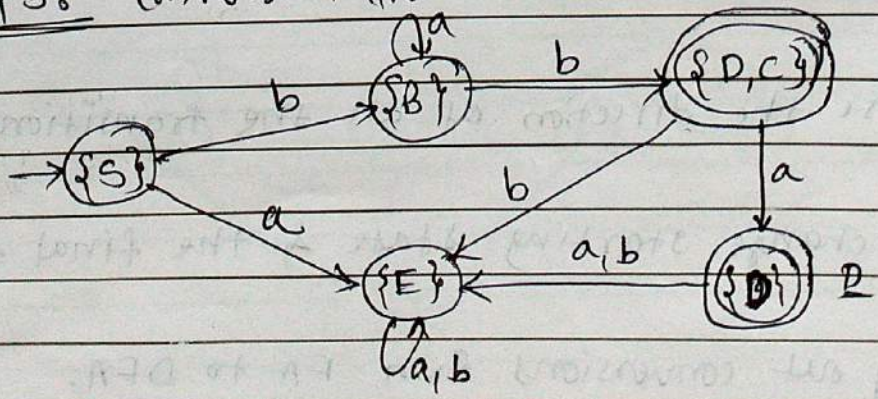


D is an accepting state. It is added for the production B→b

step2: Reverse the direction of transitions and interchange start & final state.



step3: 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 (E)
2. Interchange start & final state.
3. Reverse the direction of all transitions.
4. Write left linear grammar from the transition graph.

Example 1: Convert the following right linear grammar to an equivalent left linear grammar.

$S \rightarrow bB \mid b$

$B \rightarrow bC$

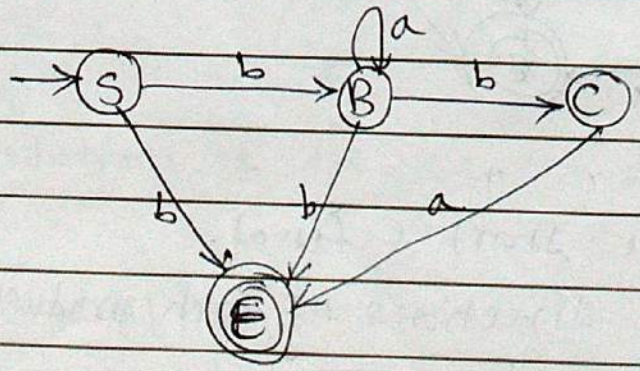
$B \rightarrow aB$

$C \rightarrow a$

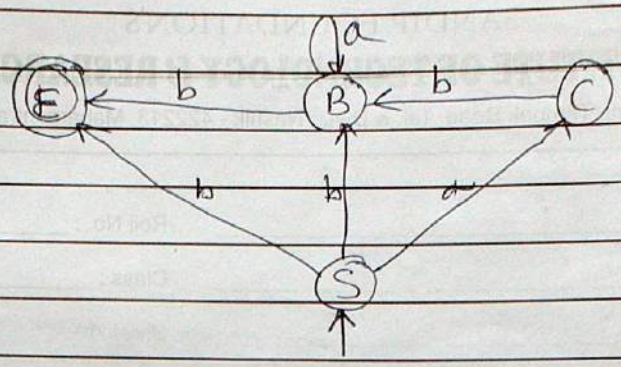
$B \rightarrow b$

Sol :→

1. Construct transition graph.



2. Interchange start and final states.
   Reverse direction of every ~~production~~ transition.



Step3 : write left linear grammar.

~~S → Bb | Ca | Eb | b~~
~~B → Ba | Eb | b~~
~~C → Bb~~

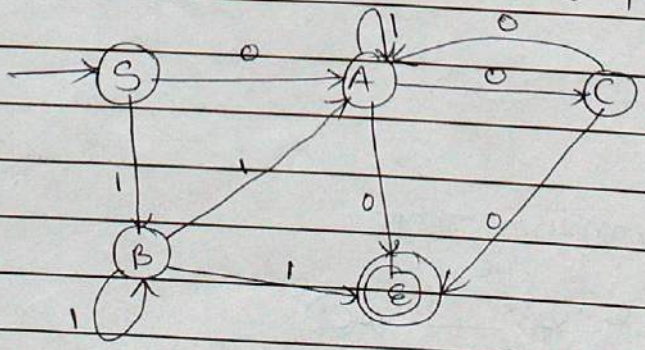
$S \rightarrow Bb \mid Ca \mid b$
$B \rightarrow Ba \mid b$
$C \rightarrow Bb$

Example2 : write an equivalent left linear grammar from the given right linear grammar.

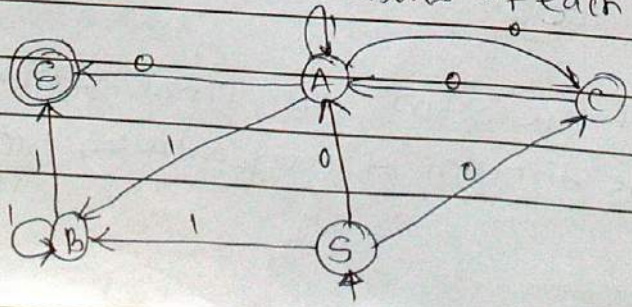
$S \rightarrow 0A \mid 1B$
$A \rightarrow 0C \mid 1A \mid 0$
$B \rightarrow 1B \mid 1A \mid 1$
$C \rightarrow 0 \mid 0A$

Soln :→

step1 : construct the transition graph & mark Final State



step2 : Interchange start & final.
Reverse the directions of each production.



Step3: write productions of left linear grammar,

$S \rightarrow Ca \mid A0 \mid B1$

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

$B \rightarrow B1 \mid 1$

$C \rightarrow A0$

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

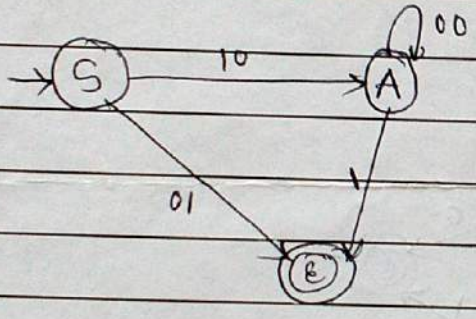
[Dec-2008, May-2009]

$S \rightarrow 10A \mid 01$

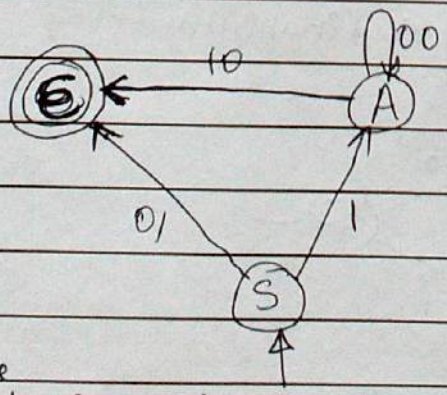
$A \rightarrow 00A \mid 1$

Sol:⇒

1. Transition graph & add ε.



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



3. write productions of left linear grammar.

$S \rightarrow A1 \mid 01$

$A \rightarrow A00 \mid 10$

Left linear to right linear grammar :⇒

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.

$S \to C0 \mid A0 \mid B1$

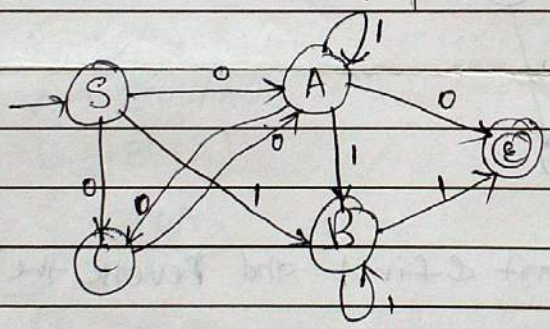
$A \to A1 \mid C0 \mid B1 \mid 0$

$B \to B1 \mid 1$

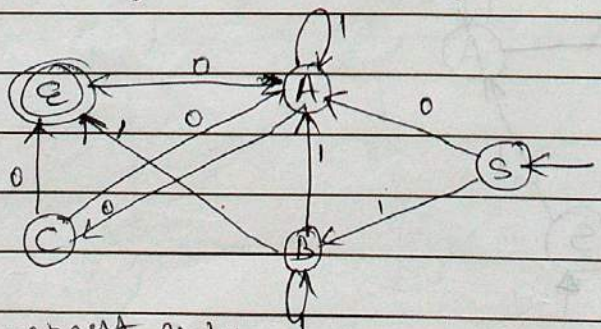
$C \to A0$

Soln :⇒

1. construct transition graph.



2. Interchange start state & final state.



3. construct Right linear grammar

~~$S \to A0 \mid B1$~~

~~$A \to A1 \mid C0 \mid 0$~~

~~$B \to B1 \mid A1 \mid 1$~~

~~$C \to A0 \mid 0$~~

$S \to 0A \mid 1B$

$A \to 1A \mid 0C \mid 0$

$B \to 1B \mid 1A \mid 1$

$C \to 0A \mid 0$