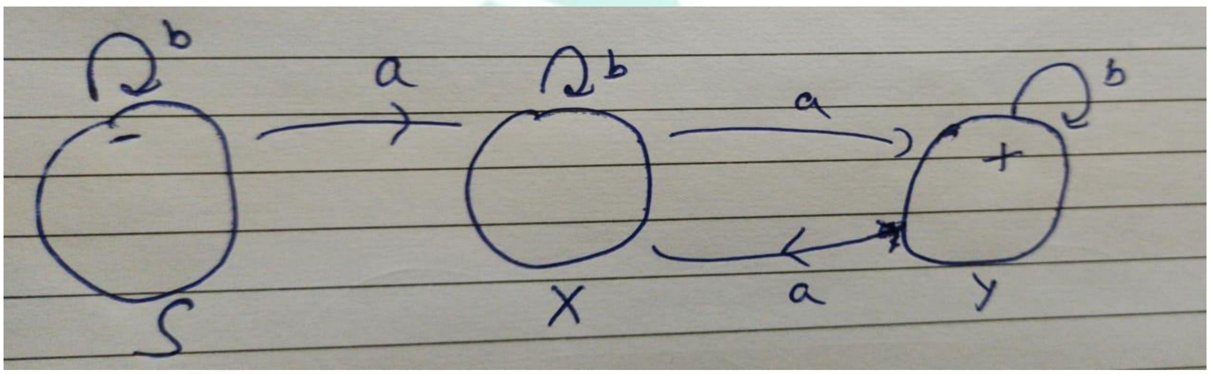
Context Free Language Part-2

Context Free Language Part-2

Content:

1. **Conversion of FA to CFG**
2. **Conversion of CFG to FA**
3. **Semi-word**
4. **Word**
5. **Null Production**
6. **Unit Production**
7. **Chomsky Normal Form(CNF)**
8. **Greibach Normal Form (GNF)**

Conversion of FA to CFG:-



S→ aX / bS

X → aY / bX

Y → aX /bY / ∧

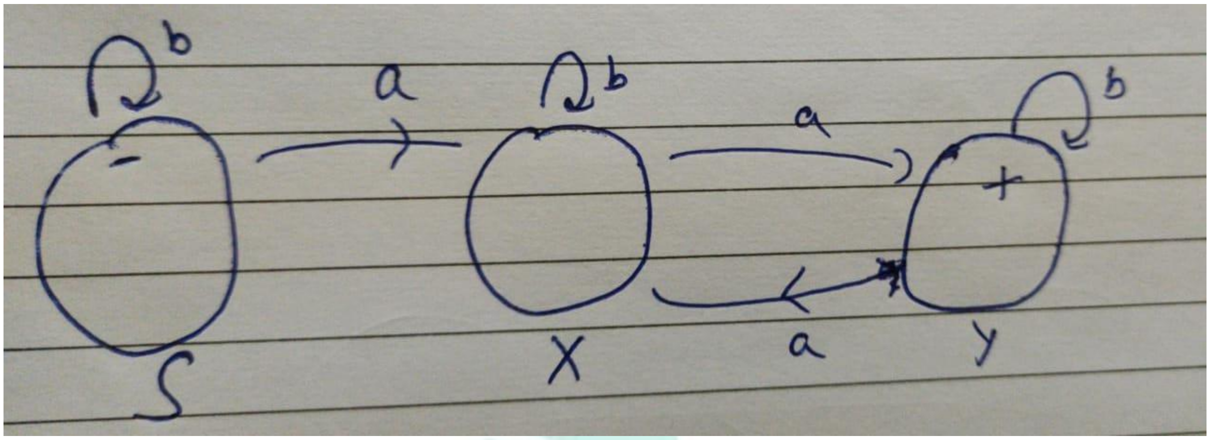
Gradeup UGC NET Super Subscription
Access to all Structured Courses & Test Series

Conversion of CFG to FA :

S → aX/ bS

X → aY / bX

Y → aX/bY/∧



Semi-word: The production rule in which ending symbol is always non-terminal & there is only one non-terminal i.e. there is one and only one terminal which is at the end

N.T. → (T)(T)(T)(T)(T)………(NT)

Word:- String of terminal

NT → (T)(T)(T)(T)…………(T)

Null Production :- Production rule to form

NT → Null

OR

NT → ∧

Gradeup UGC NET Super Subscription
Access to all Structured Courses & Test Series

Unit Production:-

A production of the form

Non-terminal → One non-terminal

(NT) →(NT)

That is a production of the form A → B (where A and B, both are non-terminals) is called unit production. Unit production increase the cost of derivation in a grammar.

Following algorithm can be used to eliminate the unit production.

Algorithm: Removal of unit production →

While (there exist a unit production, A → B)

{

Select a unit production A → B, such that there exist a production B → α, where α is a terminal.

For (every non-unit production, B → α)

Add production A → α to the grammar

Elimination A → B from the grammar.

.**Example:** Consider the context free grammar G.

S → AB

A → a

B → C/b

C → D

D → E

Gradeup UGC NET Super Subscription
Access to all Structured Courses & Test Series

Remove the unit production.

Solution: Given CFG

S → AB

A → a

B → C/b

C → D

D → E

E → a

Contain three unit productions

B → C

C → D

D → E

Now to remove unit production B → C, we see if there exists a production whose left side has C and right side contains a terminal (i.e. C → a), but there is no such productions in G. similar things holds for production C → D. now we try to remove unit production D → E, before there is a production E → a. therefore, eliminate D → E and introduction D → a, grammar becomes

S → AB

A → a

B → C/b

C → D

D → a

gradeup

$E \rightarrow a$

Now we can remove $C \rightarrow D$ by using $D \rightarrow a$, we get

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow C/b$

$C \rightarrow a$

$D \rightarrow a$

$E \rightarrow a$

Similarly, we can remove $B \rightarrow C$ by using $C \rightarrow a$, we obtain

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow a/b$

$C \rightarrow a$

$D \rightarrow a$

$E \rightarrow a$

Now it can be easily seen that production $C \rightarrow a$, $D \rightarrow a$ $E \rightarrow a$ are useless because if we start deriving from S, these productions will never be used. Hence eliminating them gives,

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow a/b$

Which is completely reduced grammar.


Gradeup UGC NET Super Subscription
Access to all Structured Courses & Test Series

CHOMSKY NORMAL FORM

If CFG has only production of the form

Non-terminal → string of exactly two non-terminal or of the form

i.e. (NT) →(NT)(NT)

Non-terminal → one terminal

i.e.

(NT) →(T)

Is said to be Chomsky normal form or CNF.

Example :

S → XY

A→ a

Q. Change the following grammar in to CNF.

S → abSb/a/aAb

A → bS/aAAb.

Q**. Convert CFG which is given below in to CNF form.**

S → bA/aB

A → bAA/aS/a

B → aBB/bS/b.


Gradeup UGC NET **Super Subscription**
Access to all Structured Courses & Test Series

GREIBACH NORMAL FORM(GNF)

Tips

For every context free language L without $\in$, there exist a grammar in which every production is of the form $A \to aV$, where 'A' is a variable, 'a' is exactly one terminal and 'V' is the string of none or more variables, clearly $V \in V^*n$.

"In other words if every production of the context free grammar is of the form $A \to aV/a$, then it is in Greibach Normal Form".

Greibach normal form will be used to construct a push down automata that recognize the language generated by a context free grammar.

To convert a grammar to GNF we start with a production in which the left side has a higher numbered variable than first variable in the right side and make replacements in right side.

Production Rules :

1. (NT) → a α

Single Terminal String of NT

2. NT → one terminal

Ex: S→ aXYZ

A→b

Q.

$S \rightarrow S_1S_2$ $S_1 \rightarrow aS_1c/S_2/\lambda$

$S_2 \rightarrow aS_2b/\lambda$ $S_3 \rightarrow aS_3b\ /\ S_4/\ \lambda$

Gradeup UGC NET Super Subscription
Access to all Structured Courses & Test Series

gradeup SUPER

Gradeup UGC NET
Super Superscription

Features:

1. 7+ Structured Courses for UGC NET Exam
2. 200+ Mock Tests for UGC NET & MHSET Exams
3. Separate Batches in Hindi & English
4. Mock Tests are available in Hindi & English
5. Available on Mobile & Desktop

Gradeup Super Subscription, Enroll Now

gradeup.co