CFG Normal Forms

CNF Form

Removal of Left Recursion

GNF Form

Normal Forms

- A grammar is said to be in a Normal Form when every production of a grammar has specific form
- This makes it convenient to design algorithms for working with CFGs

Normal Forms for CFG

- CNF, Chomsky Normal Form
- GNF, Greibach Normal Form

CNF, Chomsky Normal Form

- A grammar is said to be in Chomsky Normal form (i.e. CNF) if every production rule of the grammar is of the form:

  A->BC or A->a

- where A,B,C are in V and a is in T

Conversion to CNF

Step 1:Simplify the grammar G by **eliminating null productions, useless productions and unit productions**

Step 2: Add to the solution, the productions which are already in CNF

Step 3: For the productions, not in CNF:-

a) Replace the terminals by some variables

b) limit the number of variables on RHS to 2

CNF, Chomsky Normal Form-Example 1

- Consider a CFG with following productions:

S->aB|bA

A->a|aS|bAA

B->b|bS|aBB

CNF, Chomsky Normal Form-Example 1

- Consider a CFG with following productions:

S->aB|bA

A->a|aS|bAA

B->b|bS|aBB

Solution-

Introduce new non-terminals Ca and Cb and production

C_a->a and C_b->b , we get:-

S->C_aB| C_bA

A->a|C_aS| C_bAA

B->b| C_bS| C_aBB

CNF, Chomsky Normal Form-Example 1

$S \rightarrow C_aB \mid C_bA$
$A \rightarrow a \mid C_aS \mid C_bAA$
$B \rightarrow b \mid C_bS \mid C_aBB$
$C_a \rightarrow a$
$C_b \rightarrow b$

Introduce D1 and D2 as two new non-terminals and productions
$D_1 \rightarrow AA$ and $D_2 \rightarrow BB$

We get,
$S \rightarrow C_aB \mid C_bA$
$A \rightarrow a \mid C_aS \mid C_bD_1$
$B \rightarrow b \mid C_bS \mid C_aD_2$
$C_a \rightarrow a$
$C_b \rightarrow b$
$D_1 \rightarrow AA$
$D_2 \rightarrow BB$
The grammar is in CNF

CNF, Chomsky Normal Form-Example 2

Consider a CFG with the following productions:

S->aSa|bSb|ab

CNF, Chomsky Normal Form-Example 2

Consider a CFG with the following productions:

$S \to aSa | bSb | ab$

Introducing new non-terminals C_a and C_b, we get

$S \to C_aSC_a | C_bSC_b | C_aC_b$

$C_a \to a$

$C_b \to b$

Introducing D_1 and D_2, we get

$S \to C_aD_1 | C_bD_2 | C_aC_b$

$C_a \to a$

$C_b \to b$

$D_1 \to SC_a$

$D_2 \to SC_b$

The grammar is in CNF

CNF, Chomsky Normal Form-Example 3

Consider a CFG with the following productions:

S->aSb|aA|bB|a|b

A->aA|a

B->bB|b

CNF, Chomsky Normal Form-Example 3

Consider a CFG with the following productions:

S->aSb|aA|bB|a|b

A->aA|a

B->bB|b

CNF, Chomsky Normal Form-Example 3

Consider a CFG with the following productions:

S->aSb|aA|bB|a|b

A->aA|a

B->bB|b

Introducing new non-terminals C_a and C_b, we get

S->C_aSC_b| C_aA| C_bB|a|b

A->C_aA|a

B->C_bB|b

C_a->a

C_b->b

Introducing D_1, we get

S->C_aD_1 | C_aA| C_bB|a|b

A->C_aA|a

B->C_bB|b

C_a->a

C_b->b

D_1->SC_b

The grammar is in CNF

Elimination of Left Recursion

- If a grammar contains a pair of productions of the form $A \rightarrow A\alpha | \beta$
- then the grammar is left recursive grammar
- Left recursive grammar , if used for specification of the language then the top down parser may enter into the infinite loop during the parsing process on some erroneous input.

Elimination of Left Recursion

Left recursion can be eliminated from the grammar by the following rule-

G=(V,T,P,S)

For Production P:-

$A \rightarrow A\alpha_1 | A\alpha_2 | \ldots\ldots\ldots | A\alpha_m | \beta_1 | \beta_2 | \ldots\ldots\ldots | \beta_n$

β does not start with A

Replace by P1-

$A \rightarrow \beta_1 Z | \beta_2 Z | \ldots\ldots\ldots | \beta_n | \beta_1 | \beta_2 | \ldots\ldots\ldots | \beta_n$

$Z \rightarrow \alpha_1 Z | \alpha_2 Z | \ldots\ldots\ldots | \alpha_m Z | \alpha_1 | \alpha_2 | \ldots\ldots\ldots | \alpha_m$

Without Null Approach

OR

$A \rightarrow \beta_1 Z | \beta_2 Z | \ldots\ldots\ldots | \beta_n Z$

$Z \rightarrow \alpha_1 Z | \alpha_2 Z | \ldots\ldots\ldots | \alpha_m Z | \varepsilon$

With Null Approach

New Grammar G1=(V U {Z},T,P1,S)

Elimination of Left Recursion-Example 1

Consider the following grammar:

S->aBDh

B->Bb|c

D->EF

E->g|ε

F->f|ε

Elimination of Left Recursion-Example 1

Consider the following grammar:

S->aBDh

B->Bb|c

D->EF

E->g|ε

F->f|ε

The grammar is left recursive due to production

B->Bb|c

To eliminate the left recursion from the grammar,

Replace this pair of production by:-

B->cZ

Z->bZ|ε

Thus, Final grammar is :-

S->aBDh

B->cZ

Z->bZ|ε

D->EF

E->g|ε

F->f|ε

With Null Approach

Elimination of Left Recursion-Example 2

Consider the following grammar:

S->A

A->Ad|Ae|aB|aC

B->bBC|f

C->g

Elimination of Left Recursion-Example 2

Consider the following grammar:

S->A

A->Ad|Ae|aB|aC

B->bBC|f

C->g

The grammar is left recursive due to production

A->Ad|Ae|aB|aC

To eliminate the left recursion from the grammar,

Replace this pair of production by:-

A->aBZ|aCZ

Z->dZ|eZ|ε

Thus, Final grammar is :-

S->A

A->aBZ|aCZ

Z->dZ|eZ|ε

B->bBC|f

C->g

With Null Approach

Elimination of Left Recursion-Example 3

Consider the following grammar:

A->aBD|bDB|c

A->AB|AD

Remove left recursion

Elimination of Left Recursion-Example 3

Consider the following grammar:

A->aBD|bDB|c

A->AB|AD

Remove left recursion

To eliminate the left recursion from the grammar,

Replace :-

A->AB|AD

By this pair of production by:-

A->aBDZ|bDBZ|cZ

Z->BZ|DZ|ε

With Null Approach

GNF

GNF, Greibach Normal Form

- A grammar is said to be in **Greibach** Normal form (i.e. GNF) if every production rule of the grammar is of the form:

  A->aα

- where A is the non terminal, a is the terminal and α is a string of non-terminals(possibly empty)
- α belongs to V*

GNF, Greibach Normal Form

- Divide the production of Grammar G into left recursive and non-left recursive production
- Then eliminate left recursion to get GNF

Conversion to GNF

Step 1:Simplify the grammar G by **eliminating null productions, useless productions and unit productions**

Step 2: Add to the solution, the productions which are already in GNF

Step 3: For the productions, not in GNF:-

a) Use Substitution Rule

A->Bα

B->β_1| β_2|………|β_n

then we can write A as

A->$\beta_1\alpha$ | $\beta_2\alpha$ |………|$\beta_n\alpha$

b) Remove left recursion , Use without epsilon approach

GNF, Greibach Normal Form-Example 1

Consider a CFG with the following productions, Convert to GNF:

S->aSa|bSb|ab

For GNF-

A->aα

where A is the non terminal, a is the terminal and α is a string of non-terminals(possibly empty)

GNF, Greibach Normal Form-Example 1

Consider a CFG with the following productions, Convert to GNF :

S->aSa|bSb|ab

For GNF-

A->aα

where A is the non terminal, a is the terminal and α is a string of non-terminals(possibly empty)

Simple Variable Usage

Soln:

S->aSA|bSB|aB

A->a

B->b

The grammar is in GNF

GNF, Greibach Normal Form-Example 2

- Consider a CFG with the following productions, Convert to GNF :

S->XY|XW

X->YZ|a

Z->c

Y->WZ|b

W->d

For GNF-

A->aα

where A is the non terminal, a is the terminal and α is a string of non-terminals(possibly empty)

Use Substitution Rule

GNF, Greibach Normal Form-Example 2

- Consider a CFG with the following productions, Convert to GNF :

S->XY|XW

X->YZ|a

Z->c

Y->WZ|b

W->d

- Substitute W value in Y production

Y->WZ|b, rewritten as-

Y->dZ|b

- Substitute Y value in X production

X->YZ|a, rewritten as-

X->dZZ|bZ|a

- Substitute X value in S production

S->XY|XW, rewritten as-

S->dZZY|bZY|aY|dZZW|bZW|aW

- Final productions are-

S->dZZY|bZY|aY|dZZW|bZW|aW

X->dZZ|bZ|a

Y->dZ|b

W->d

Z->c

GNF, Greibach Normal Form-Example 3

Consider a CFG with the following productions, Convert to GNF :

S->AA|0

A->SS|1

GNF, Greibach Normal Form-Example 3

Consider a CFG with the following productions, Convert to GNF :

S->AA|0

A->SS|1

GNF, Greibach Normal Form-Example 3

Consider a CFG with the following productions, Convert to GNF :

S->AA|0

A->SS|1

Substitute first S in production of A

A->SS|1

A->AAS|0S|1

Substitution Rule

So left recursion is there

To eliminate Left recursion-

A->0SZ|1Z|0S|1 **GNF Form**

Z->ASZ|AS

New Productions are:

S->AA|0

A->0SZ|1Z|0S|1

Removal of Recursion

Z->ASZ|AS

Now substitute the Value of A in S production

S->0SZA|1ZA|0SA|1A|0 **GNF Form**

A->0SZ|1Z|0S|1 **GNF Form**

Z->ASZ|AS

Substitution Rule

Now substitute the Value of A in Z production

S->0SZA|1ZA|0SA|1A|0 **GNF Form**

A->0SZ|1Z|0S|1 **GNF Form**

Z-> **0SZSZ|1ZSZ|0SSZ|1SZ**

0SZS|1ZS|0SS|1S **GNF Form**

Substitution Rule

GNF, Greibach Normal Form-Example 4

Convert the following grammar into GNF

E->E+T|T

T->T*F|F

F->(E)|a

GNF, Greibach Normal Form-Example 4

Convert the following grammar into GNF

E->E+T|T

T->T*F|F

F->(E)|a

Substitution Rule-

E->E+T|T*F|F or E->E+T|T*F|(E)|a

T->T*F|(E)|a

F->(E)|a

Step 1: Remove unit productions

E->T

T->F

After removal of unit productions, we get

E->E+T|T*F|(E)|a

T->T*F|(E)|a

F->(E)|a

GNF, Greibach Normal Form-Example 4

Step 2: Replace terminals of the productions which are not in GNF form by variables

E->E+T|T*F|(E)|a

T->T*F|(E)|a

F->(E)|a

Introduce

A_1->+

A_2->*

A_3->)

The resulting Productions are-

E->EA_1T|TA_2F|(EA_3|a

T->TA_2F|(EA_3|a

F->(EA_3|a

Now all F productions are in GNF form

GNF, Greibach Normal Form-Example 4

Step 3:Remove Left Recursion for T

$E \to EA_1T|TA_2F|(EA_3|a$

$T \to TA_2F|(EA_3|a$

$F \to (EA_3|a$

Left Recursion Removal-for T

Replace $T \to TA_2F|(EA_3|a$ by

$T \to (EA_3Z_1|aZ_1|(EA_3|a$

$Z_1 \to A_2FZ_1|A_2F$

Substituting, we get

$A_2 \to *$

$T \to (EA_3Z_1|aZ_1|(EA_3|a$

$Z_1 \to *FZ_1|*F$

So Productions are

$E \to EA_1T|TA_2F|(EA_3|a$

$\mathbf{T \to (EA_3Z_1|aZ_1|(EA_3|a}$

$\mathbf{Z_1 \to *FZ_1|*F}$

$\mathbf{F \to (EA_3|a}$

GNF, Greibach Normal Form-Example 4

Productions are

$E \to EA_1T|TA_2F|(EA_3|a$

$T \to (EA_3Z_1|aZ_1|(EA_3|a$

$Z_1 \to *FZ_1|*F$

$F \to (EA_3|a$

For Production $E \to TA_2F$, Substitute the value of T

Replace $E \to TA_2F$ by

$E \to (EA_3Z_1A_2F|aZ_1A_2F|(EA_3A_2F|aA_2F$

so Replace $E \to EA_1T|TA_2F|(EA_3|a$ by

$E \to EA_1T$**$|(EA_3Z_1A_2F|aZ_1A_2F|(EA_3A_2F|aA_2F|(EA_3|a$**

Productions are

$E \to EA_1T$**$|(EA_3Z_1A_2F|aZ_1A_2F|(EA_3A_2F|aA_2F|(EA_3|a$**

$T \to (EA_3Z_1|aZ_1|(EA_3|a$

$Z_1 \to *FZ_1|*F$

$F \to (EA_3|a$

All productions of E are in GNF except $E \to EA_1T$

GNF, Greibach Normal Form-Example 4

Productions are

E->EA$_1$T|(EA$_3$Z$_1$A$_2$F|aZ$_1$A$_2$F|(EA$_3$A$_2$F|aA$_2$F|(EA$_3$|a

T->(EA$_3$Z$_1$|aZ$_1$|(EA$_3$|a

Z$_1$->*FZ$_1$|*F

F->(EA$_3$|a

All productions of E are in GNF except E->EA$_1$T

Removing Left recursion for E->EA$_1$T

Replace E->EA$_1$T**|(EA$_3$Z$_1$A$_2$F|aZ$_1$A$_2$F|(EA$_3$A$_2$F|aA$_2$F|(EA$_3$|a by**

E->**(EA$_3$Z$_1$A$_2$FZ$_2$|aZ$_1$A$_2$FZ$_2$|(EA$_3$A$_2$FZ$_2$|aA$_2$FZ$_2$|(EA$_3$Z$_2$|aZ$_2$|**
(EA$_3$Z$_1$A$_2$F|aZ$_1$A$_2$F|(EA$_3$A$_2$F|aA$_2$F|(EA$_3$|a|

Z$_2$->A$_1$T**Z$_2$**| A$_1$T

Substituting + for A$_1$

Z$_2$->+T**Z$_2$**| +T

Resulting grammar is-

E->(EA$_3$Z$_1$A$_2$FZ$_2$|aZ$_1$A$_2$FZ$_2$|(EA$_3$A$_2$FZ$_2$|aA$_2$FZ$_2$|(EA$_3$Z$_2$|aZ$_2$|
(EA$_3$Z$_1$A$_2$F|aZ$_1$A$_2$F|(EA$_3$A$_2$F|aA$_2$F|(EA$_3$|a|

Z$_2$->+TZ$_2$| +T

T->(EA$_3$Z$_1$|aZ$_1$|(EA$_3$|a

Z$_1$->*FZ$_1$|*F

F->(EA$_3$|a