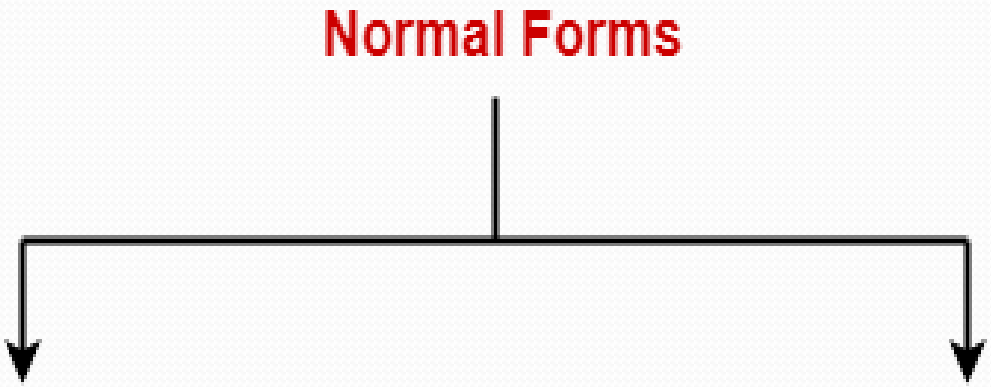
GNF (Lemma 1)

---Sakshi Surve

Normal Forms:

- By reducing the grammar, the grammar gets minimized but does not get standardized.
- This is because the RHS of productions have no specific format.
- In order to standardize the grammar, normalization is performed using normal forms.



Chomsky Normal Form (CNF)

Greibach Normal Form (GNF)

Greibach Normal Form (GNF) :

- GNF stands for **Greibach Normal Form.**

- A CFG(context free grammar) is in GNF(Greibach normal form) if all the production rules satisfy one of the following conditions:

  - A non-terminal generating a terminal.
    - $A \rightarrow a.$

  - A non-terminal generating a terminal which is followed by any number of non-terminals.
    - $S \rightarrow aA.$
    - $A \rightarrow bD_1...D_n$

Steps to convert CFG to GNF

1) Eliminate useless symbols

2) Eliminate unit productions

3) Eliminate ε-productions

4) G should be in CNF

5) Eliminate left recursion

6) Get the production in the form of

- ***i.*** ***A → aA1A2A3*** *where A1,A2,A3 are variables* or
- ***ii.*** ***A → a*** *where a is a terminal*

Lemma 1: Substitution Rule

- If G is a grammar with productions
  - $A \rightarrow Ba$
  - $B \rightarrow B_1 \mid B_2 \mid B_3 \mid \ldots B_n$

Then, the CFG can be converted to GNF as

$A \rightarrow B_1a \mid B_2a \mid \ldots\ldots B_na$

Lemma 2 : Removal of Left Recursion

- This is applicable for removing Left Recursion present in the Grammar
  - $A \rightarrow Aa_1 \mid Aa_2 \mid Aa_3 \mid \ldots\ldots\ldots \mid Aa_n \mid B_1 \mid B_2 \mid \ldots\ldots B_n$

- Equivalent GNF will be :

  - $A \rightarrow B1 \mid B2 \mid B3\ldots\ldots \mid Bn$
  - $A \rightarrow B1Z \mid B2Z \mid B3Z \mid \ldots\ldots\ldots Bn\ Z$
  - $Z \rightarrow a1 \mid a2 \mid a3 \ldots\ldots \mid an$
  - $Z \rightarrow a1\ Z \mid a2Z \mid a3Z \mid \ldots\ldots \mid anZ$

1. Find GNF of the grammar given below

S→ ABA | AB | BA | AA | A | B

A → aA | a

B → bB | b

- Simplifying the grammar and removing the Unit Productions …

  S→ ABA | AB | BA | AA | aA | a | bB | b

  A → aA | a

  B → bB | b

- A → aA | a ..... 1 in GNF

  B → bB | b ..... 2 in GNF

- S is not in GNF
- Applying Substitutions of A and B in S as per Lemma 1 ..
- S→ aABA | aBA | aAB | aB | bBA | bA | aAA | aA |
  a | bB | b ..... 3 in GNF

- Hence, the new grammar G' in GNF is :
  G' = { (S, A, B) , S , P, (a , b)}

- P→
  S→ aABA | aBA | aAB | aB | bBA | bA | aAA | aA |
  a | bB | b
  A → aA | a
  B → bB | b

2. Find GNF equivalent to given CFG

S→ AB

A→ aA | bB | b

B → b

- **Step 1:** Grammar is simplified
- **Step 2 :** Checking for each production for GNF
  a) B→ b ……1 in GNF
  b) A→ aA | bB | b ……2 in GNF
  c) S→ AB Not in GNF

  Applying Lemma 1 , substituting all productions of A in S …we get,

  S→ aAB | bBB | bB ……3 in GNF

- Hence, the new grammar G' in GNF is :

G' = { (S, A, B) , S , P, (a , b)}

P➔ S→ aAB | bBB | bB

A → aA | bB | b

B → b

3. Find GNF of the grammar given below

S→ ABAb | ab

B → ABA | a

A → a | b

- Considering each production and converting to GNF
- A → a | b ..... 1 in GNF

- B→ ABA | a

Substituting A → a and A → b in above production using Lemma 1

B → aBA | bBA | a ..... 2 is in GNF

For production S→ ABAb | ab

Introducing a non terminal C→b

S→ ABAC | aC

S→ aC …….3 in GNF

S→ ABAC Not in GNF

Substituting productions of A in above production using Lemma 1 , we get

S → aBAC | bBAC …… 4 in GNF

- Hence, the new grammar G' in GNF is :

G' = { (S, A, B, C) , S , P, (a , b)}

P➔ S → aBAC | bBAC | aC

B → aBA | bBA| a

A → a | b

C → b

4. Find GNF of the grammar given below

S→ XX1 | 0

X → 00X | Y

Y → 1X1

- Removing the Unit Production X→Y , the grammar becomes,

  S→ XX1 | 0

  X → 00X | 1X1

Introducing Non Terminals A→1 , and B→ 0

S→ XXA | 0

X → 0BX | 1XA

X→ 0BX | 1XA .....1 in GNF

A→1 ..... 2 in GNF

B→0 ..... 3 in GNF

- S→ XXA | 0 Not in GNF

- Substituting productions of X in above production using Lemma 1 , we get

  S → 0BXXA | 1XAXA | 0 ..... 3 in GNF

- Hence, the new grammar G' in GNF is :

G' = { (S, A, B, C) , S , P, (a , b)}

P➔ S → 0BXXA | 1XAXA | 0

X→ 0BX | 1XA

A→ 1

B→0

5. Find GNF equivalent to given CFG

S→ AB

A→ BSB| BB | b

B → aAb | a

- Grammar is simplified
- Add a new Non terminal C as C → b
- G becomes

S → AB

A → BSB | BB | b

B → aAC | a

C → b

- C → b ..... 1 in GNF
- B → aAC | a ..... 2 in GNF
- **A → BSB | BB ..... Not in GNF**

  Substituting productions of B in above production using Lemma 1 , we get...

  A → aACSB | aSB | aACB | aB | b ..... 3 in GNF

- **S → AB ....not in GNF**

  Substituting productions of A in above production using Lemma 1 , we get...

  S → aACSBB | aSBB | aACBB | aBB | bB ..... 4 in GNF

- Hence, the new grammar G' in GNF is :

G' = { (S, A, B, C) , S , P, (a , b)}

P➔ S → aACSBB | aSBB | aACBB | aBB | bB

A → aACSB | aSB | aACB | aB | b

B → aAC | a

C → b

6. Find GNF equivalent to given CFG

S→ AB
A→ aBa| a
B → aAb | b

- C→ a, D→b

- S→AB
  A→aBC | a
  B→ aAD | b

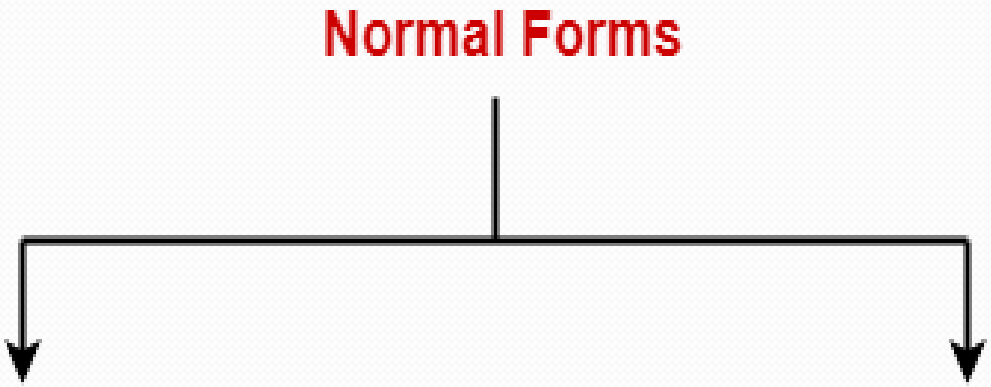
Final Productions :
S→ aBCB | aB
A→aBC | a
B→ aAD | b

GNF (Lemma 2)

---Sakshi Surve

Normal Forms:

- By reducing the grammar, the grammar gets minimized but does not get standardized.
- This is because the RHS of productions have no specific format.
- In order to standardize the grammar, normalization is performed using normal forms.



Chomsky Normal Form (CNF)

Greibach Normal Form (GNF)

Greibach Normal Form (GNF) :

- GNF stands for **Greibach Normal Form.**

- A CFG(context free grammar) is in GNF(Greibach normal form) if all the production rules satisfy one of the following conditions:

  - A non-terminal generating a terminal.
    - $A \rightarrow a.$

  - A non-terminal generating a terminal which is followed by any number of non-terminals.
    - $S \rightarrow aA.$
    - $A \rightarrow bD_1...D_n$

Steps to convert CFG to GNF

1) Eliminate useless symbols

2) Eliminate unit productions

3) Eliminate ε-productions

4) G should be in CNF

5) Eliminate left recursion

6) Get the production in the form of

- ***i.*** ***A → aA1A2A3*** *where A1,A2,A3 are variables* or
- ***ii.*** ***A → a*** *where a is a terminal*

Lemma 1: Substitution Rule

- If G is a grammar with productions
  - $A \rightarrow Ba$
  - $B \rightarrow B_1 \mid B_2 \mid B_3 \mid \ldots B_n$

Then, the CFG can be converted to GNF as

$A \rightarrow B_1a \mid B_2a \mid \ldots\ldots B_na$

Lemma 2 : Removal of Left Recursion

- This is applicable for removing Left Recursion present in the Grammar
  - $A \rightarrow Aa_1 \mid Aa_2 \mid Aa_3 \mid \ldots\ldots\ldots \mid Aa_n \mid B_1 \mid B_2 \mid \ldots\ldots B_n$

- Equivalent GNF will be :
  - A → B1 | B2 | B3…..| Bn
  - A → B1Z | B2Z | B3Z | …….Bn Z
  - Z → a1 | a2 | a3 ….. | an
  - Z → a1 Z| a2Z |a3Z |….|anZ
- **Remove the rule with direct Left Recursion and, create a new one with recursion on right**
- E.g. Before A → Aa | b

  After A → bZ | b

  Z → aZ | a
- E.g. Before A → Aa | Ab | b | c

  After A → bZ | cZ | b | c

  Z → aZ | bZ | a | b

6. Find GNF equivalent to given CFG

S→ SS | aSB | aB

B → b

- **Step 1:** Grammar is simplified
- **Step 2 :** Checking for each production for GNF
  a) B→ b ..... **1** in GNF
  b) S→ SS | aSB | aB
  c) Now, S is in Left Recursion
  d) S→ aSB | aB | aSBZ | aBZ **2**
     Z → S | SZ.....Adding a new Variable by applying Lemma 2
  e) Now, S is in GNF

- Z is not in GNF
- To make Z in GNF apply Lemma 1 , substitute production of S
- Z → aSB | aSBZ | aB | aBZ | aSBZ | aSBZZ | aBZ | aBZZ

..... **3**

- New Grammar G' is :
- G' = { (A,S,B) , S , P, (a,b) }
- P →

S→ aSB | aB | aSBZ | aBZ

Z → aSB | aSBZ| aB | aBZ | aSBZ | aSBZZ | aBZ | aBZZ

B → b

7

Find GNF:

$S \rightarrow BA \mid ab$

$B \rightarrow AB \mid a$

$A \rightarrow Bb \mid BB$.

i) Grammar is simplified.

ii) To get $A \rightarrow a\alpha$ where, α is non-terminal substituting D as $D \rightarrow b$ in production of $A \rightarrow Bb$ and $S \rightarrow ab$.

$\therefore$

$S \rightarrow BA \mid aD$

$B \rightarrow AB \mid a$

$A \rightarrow BD \mid BB$

$D \rightarrow b$.

iii) a) $D \rightarrow b$ ---- (1) in GNF.

b) $A \rightarrow BD \mid BB$. Not in GNF.

$B \rightarrow AB \mid a$ Not completely in GNF.

Substituting productions of 'B' in 'A' as per Lemma

$\therefore A \rightarrow ABD \mid ABB \mid aD \mid aB$.

Now, productions of A contain left recursion.

i.e. $A \rightarrow ABD \mid ABB$.

Applying Lemma 2 to remove Left Recursion

Add new production as Z.

$A \rightarrow aD \mid aB \mid aDZ \mid aBZ$

In A keep productions without L. Rec

$\therefore Z \rightarrow BDZ \mid BBZ \mid BD \mid BB$.

In Z consider productions starting with A but while without A.

Now, $A \rightarrow aD \mid aB \mid aDz \mid aBz$ ... (2) in GNF.

$B \rightarrow AB \mid a$ Not in GNF.

Applying Lemma 1 & substituting productions of A in 'B'

$B \rightarrow aDB \mid aBB \mid aDZB \mid aBZB \mid a$ ... (3) in GNF.

c) $S \rightarrow BA \mid aD$ Not in GNF completely.

Applying Lemma 1, substituting production of B in 'S'.

$S \rightarrow aDBA \mid aBBA \mid aDZBA \mid aBZBA \mid aA \mid aD$

... (4) in GNF

d) $Z \rightarrow BDZ \mid BBZ \mid BD \mid BB$ ... Not in GNF.

Applying Lemma 1 by substituting productions of B.

$Z \rightarrow BD \Rightarrow Z \rightarrow aDBD \mid aBBD \mid aDZBD \mid aBZBD \mid aD$

$Z \rightarrow BDZ \Rightarrow Z \rightarrow aDBDZ \mid aBBDZ \mid aDZBDZ \mid aBZBDZ \mid aDZ$

$Z \rightarrow BD \Rightarrow aDBD \mid aBBD \mid aDZBD \mid aBZBD \mid aD$

$Z \rightarrow BB \Rightarrow aDBB \mid aBBB \mid aDZBB \mid aBZBB \mid aB.$

Merging all productions of Z. .... (5)

$Z \rightarrow \ldots$

∴ New Grammar -

$S \rightarrow$

$B \rightarrow$

$A \rightarrow$

$D \rightarrow$

8. Convert the CFG to GNF

S->aSb|aX

X->Xa|Sa|a

- Introducing two new Non Terminals A → a , B→ b
- The grammar becomes

S→ aSB | aX

X → XA | SA | a

B → b

D → a

- B→ b ……..1 in GNF
- A → a ……. 2 in GNF
- X → a …..3 in GNF
- X → SA by Lemma 1 becomes…

  X → aSBA | aXA

  Overall… X → XA | aSBA | aXA | a
- X → XA Not in GNF and this is showing Left Recursion . So, introducing a new Variable Z

- X → aSBA | aXA | aSBAZ |aXAZ | a | aZ …..4 in GNF
- Z → a | aZ …….. 5 in GNF
- S→ aSB | aX …… 6 in GNF

- New Grammar G' is :
- G' = { (A,S,B) , S , P, (a,b) }
- P ➔

S → aSB | aX

X → aSBA | aXA | aSBAZ |aXAZ | a | aZ

Z → a | aZ

A → a

B → b

9. Convert CFG to GNF

S → AA | 0

A → SS | 1

- A → SS | 1 ..... Not in GNF
- A → AAS | 0S | 1 This is showing Left Recursion
- A → 0S | 0SZ | 1 | 1Z .... 1 in GNF
  Z → AS | ASZ Not in GNF
- Substituting A in Z
- Z → 0SS | 0SSZ | 0SZS | 0SZSZ | 1S | 1SZ | 1ZS | 1ZSZ ..... 2 in GNF

- Substituting A in S using Lemma 1

- S → 0SA | 0SZA | 1A | 1ZA | 0 ..... 3 in GNF
- New Grammar G' is :
- G' = { (A,S,B) , S , P, (a,b) }
- P →

S → 0SA | 0SZA | 1A | 1ZA | 0

A → 0S | 0SZ | 1 | 1Z

Z → 0SS | 0SSZ | 0SZS | 0SZSZ | 1S | 1SZ | 1ZS | 1ZSZ