Greibach Normal Form (GNF)

A CFG is in Greibach Normal Form (GNF) if all production rules satisfy one of the following conditions:

- ✓ A non-terminal generating a terminal
  e.g., X→x
- ✓ A non-terminal generates a terminal followed by any number of non-terminals
  e.g., X→xX1X2…XN

Consider the following grammar:

G1={ S→aA|bB, B→bB|b,A→aA|a }

G2={S→aA|bB, B→bB|ϵ, A→aA|ϵ}

- ✦ The grammar G1 is in CNF as production rules satisfy the rules specified for CNF, so it can be directly used to convert to GNF. According to the rules, G1is also in GNF form.
- ✦ However, the grammar G2 is not in CNF as the production rules B→ϵ and A→ϵ do not satisfy the rules specified for CNF (only the start symbol can generate ϵϵ), so first remove the unit and null production and convert it into GNF.

How to Convert CFG to GNF

Step 1. If the given grammar is not in CNF, convert it to CNF.

Step 2. Change the names of non terminal symbols to A1 till AN in same sequence.

Step 3. Check for every production rule if RHS has first symbol as non-terminal say Aj for the production of Ai, *it is mandatory that i should be less than j. Not great and not even equal*.

- If **i> j** then *replace the production rule of Aj at its place in Ai*.
- If **i=j,** it is the ***left recursion.***
  (Create a new state Z which has the symbols of the left recursive production, once followed by Z and once without Z, and change that production rule by removing that particular production and adding all other production once followed by Z).

Step 4. Replace very first non-terminal symbol in any production rule with its production until production rule satisfies the above conditions.

- ✦ For converting a CNF to GNF ***always move left to right*** for **renaming the variables**.

Example:

S → XA|BB
B → b|SB
X → b
A → a

For converting a CNF to GNF, first rename the non-terminal symbols to A1, A2 till AN in same sequence as they are used.

A1 = S
A2 = X
A3 = A
A4 = B

Therefore, now the new production rule is,

A1 → A2A3 | A4A4
A2 → b
A3 → a
A4 → b | A1A4

Now, check for every production Ai → Aj X .

- If i<j in the production then it is good to go to the next step
- but if i>j then change the production by replacing it with that terminal symbol's production.
- if i=j then it is a left recursion and you need to remove left recursion.

Here for A4, **4 !< 1**, so now replace it with A1's production rule.

1.
A1 → A2A3 | A4A4
A2 → b
A3 → a
A4 → b | A2A3A4 | A4A4A4

2.
A1 → A2A3 | A4A4
A2 → b
A3 → a
A4 → b | bA3A4 | A4A4A4

Here **A4A4A4** in production rule of **A4** is the example of **left recursion**.

To replace the left most recursion take a new Non terminal symbol Z, which has the X part or the trailing part of the left most recursive production once followed by Z and once without Z. Here in A4A4A4, the part after the first A4 is A4A4, therefore

Z → A4A4 | A4A4Z

Now change the above production rule by putting Z after every previous production of that Ai, and remove the left recursive production.

A1 → A2A3 | A4A4
A2 → b
A3 → a
A4 → b | bA3A4 | bZ | bA3A4Z
Z → A4A4 | A4A4Z

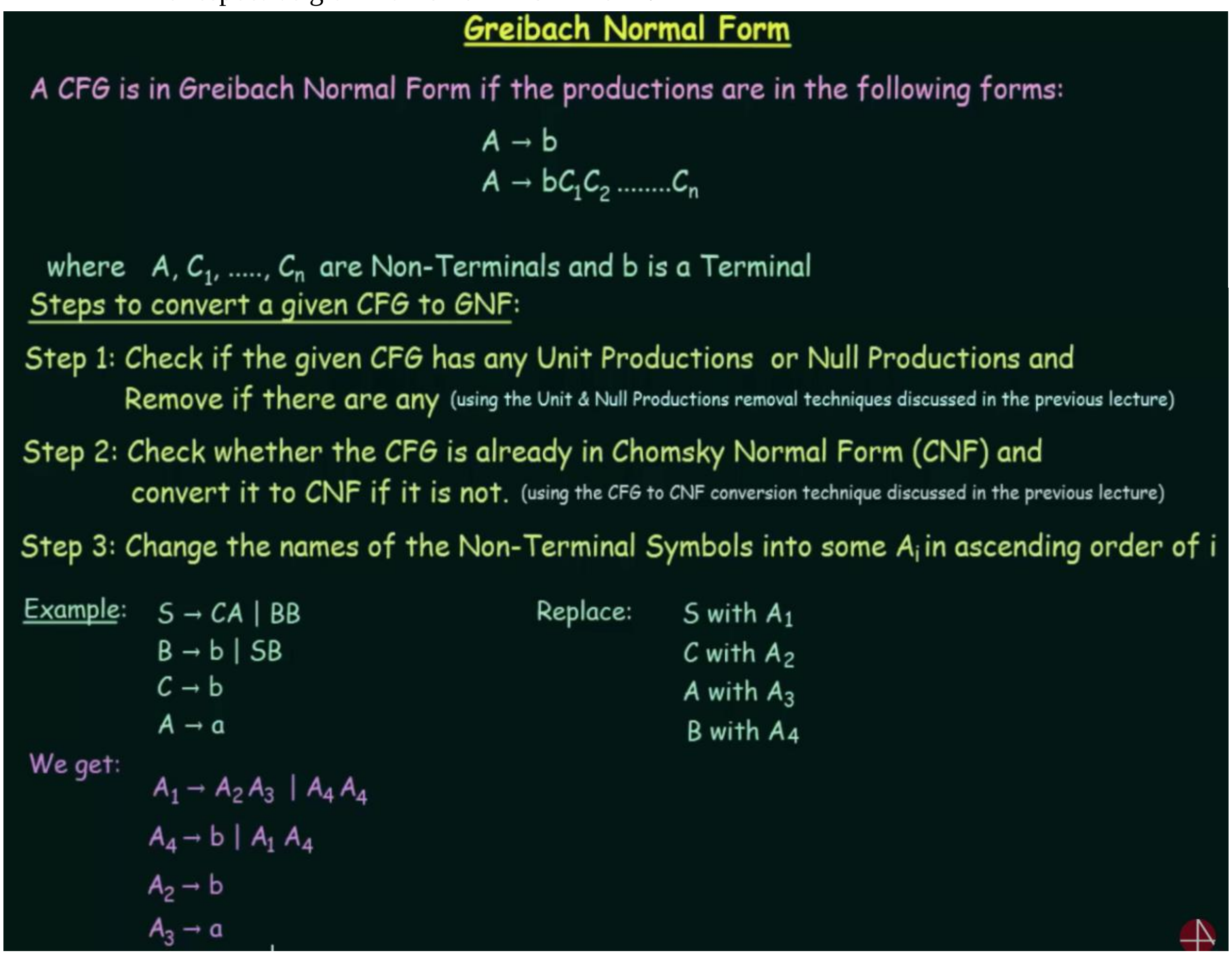
The Last step is to replace the production to the form of either

Ai → x (any single terminal symbol)
OR
Ai → xX (any single terminal followed by any number of non-terminals)

So here we need to replace A2 in production rule of A1 and so on.

A1 → bA3 | bA4 | bA3A4A4 | bZA4 | bA3A4ZA4
A2 → b
A3 → a
A4 → b | bA3A4 | bZ | bA3A4Z
Z → bA4 | bA3A4A4 | bZA4 | bA3A4ZA4 | bA4Z | bA3A4A4Z | bZA4Z | bA3A4ZA4Z

The respective grammar is non in GNF form.



Step 4: Alter the rules so that the Non -Terminals are in ascending order, such that,

If the Production is of the form $A_i \to A_j\, x$, then,

$i < j$ and should never be $i \geqslant j$

$$A_4 \to b \mid \underline{A_1} A_4$$

$$A_4 \to b \mid \underline{A_2} A_3 A_4 \mid A_4 A_4 A_4$$

$$A_4 \to b \mid b A_3 A_4 \mid A_4 A_4 A_4$$

$A_1 \to A_2 A_3 \mid A_4 A_4$

$\boxed{A_4 \to b \mid A_1 A_4} \longrightarrow A_4 \to b \mid b A_3 A_4 \mid A_4 A_4 A_4$

$A_2 \to b$

$A_3 \to a$

Left Recursion

Step 5: Remove Left Recursion

Introduce a New Variable to remove the Left Recursion

$$A_4 \to b \mid b A_3 A_4 \mid A_4 A_4 A_4$$

$$Z \to A_4 A_4 Z \mid A_4 A_4$$

$$A_4 \to b \mid b A_3 A_4 \mid bZ \mid b A_3 A_4 Z$$

Now the grammar is:

$A_1 \to A_2 A_3 \mid A_4 A_4$

$A_4 \to b \mid b A_3 A_4 \mid bZ \mid b A_3 A_4 Z$

$Z \to A_4 A_4 \mid A_4 A_4 Z$

$A_2 \to b$

$A_3 \to a$

$$A_1 \to b A_3 \mid b A_4 \mid b A_3 A_4 A_4 \mid bZ A_4 \mid b A_3 A_4 Z A_4$$

$$A_4 \to b \mid b A_3 A_4 \mid bZ \mid b A_3 A_4 Z$$

$$Z \to b A_4 \mid b A_3 A_4 A_4 \mid bZ A_4 \mid b A_3 A_4 Z A_4 \mid$$
$$b A_4 Z \mid b A_3 A_4 A_4 Z \mid bZ A_4 Z \mid b A_3 A_4 Z A_4 Z$$

$$A_2 \to b$$

$$A_3 \to a$$

This is the required GNF grammar.

Example: Convert the following grammar G into Greibach Normal Form (GNF).

$$
\boxed{\begin{array}{l} S \to XA|BB \\ B \to b|SB \\ X \to b \\ A \to a \end{array}}
$$

To write the above grammar G into GNF, we shall follow the following steps:

1. Rewrite G in Chomsky Normal Form (CNF) : It is already in CNF.
2. Re-label the variables

   S with A_1

   X *with* $A2$

   A *with* $A3$

   B *with* $A4$

   After re-labeling the grammar looks like:

   $A_1 \to A_2A_3|A_4A_4$

   $A_4 \to b|A_1A_4$

   $A_2 \to b$

   $A_3 \to a$

3. Identify all productions which do not conform to any of the types listed below:

   $A_i \to A_jx_k$ such that $j > i$

   $Z_i \to A_jx_k$ such that $j \leq n$

   $A_i \to ax_k$ such that $x_k \in V*$ and $a \in T$

4. $A_4 \to A_1A_4$..............................identified
5. $A_4 \to A_1A_4|b$.

   To eliminate A_1 we will use the substitution rule $A_1 \to A_2A_3|A_4A_4$.

   Therefore, we have $A_4 \to A_2A_3A_4|A_4A_4A_4|b$

   The above two productions still do not conform to any of the types in step 3.

Substituting for $A_2 \rightarrow b$

$A_4 \rightarrow bA_3A_4|A_4A_4A_4|b$

Now we have to remove left recursive production $A_4 \rightarrow A_4A_4A_4$

$A_4 \rightarrow bA_3A_4|b|bA_3A_4Z|bZ$

$Z \rightarrow A_4A_4|A_4A_4Z$

6. At this stage our grammar now looks like

   $A_1 \rightarrow A_2A_3|A_4A_4$

   $A_4 \rightarrow bA_3A_4|b|bA_3A_4Z|bZ$

   $Z \rightarrow A_4A_4|A_4A_4Z$

   $A_2 \rightarrow b$

   $A_3 \rightarrow a$

   All rules now conform to one of the types in step 3. But the grammar is still not in Greibach Normal Form.

7. All productions for A_2, A_3 and A_4 are in GNF for

   $A_1 \rightarrow A_2A_3|A_4A_4$

   Substitute for A_2 and A_4 to convert it to GNF

   $A_1 \rightarrow bA_3|bA_3A_4A_4|bA_4|bA_3A_4ZA_4|bZA_4$

   for $Z \rightarrow A_4A_4|A_4A_4Z$

   Substitute for A_4 to convert it to GNF

   $Z \rightarrow bA_3A_4A_4|bA_4|bA_3A_4ZA_4|bZA_4|bA_3A_4A_4Z|bA_4Z|bA_3A_4ZA_4Z|bZA_4Z$

8. Finally the grammar in GNF is

   $A_1 \rightarrow bA_3|bA_3A_4A_4|bA_4|bA_3A_4ZA_4|bZA_4$

   $A_4 \rightarrow bA_3A_4|b|bA_3A_4Z|bZ$

   $Z \rightarrow bA_3A_4A_4|bA_4|bA_3A_4ZA_4|bZA_4|bA_3A_4A_4Z|bA_4Z|bA_3A_4ZA_4Z|bZA_4Z$

   $A_2 \rightarrow b$

   $A_3 \rightarrow a$