1 Greibach Normal Form (GNF)

A CFG $G = (V, T, R, S)$ is said to be in GNF if every production is of the form $A \to a\alpha$, where $a \in T$ and $\alpha \in V^*$, i.e., α is a string of zero or more variables.

Definition: A production $\mathcal{U} \in R$ is said to be in the form **left recursion**, if $\mathcal{U} : A \to A\alpha$ for some $A \in V$.

Left recursion in R can be eliminated by the following scheme:
• If $A \to A\alpha_1|A\alpha_2|\ldots|A\alpha_r|\beta_1|\beta_2|\ldots|\beta_s$, then replace the above rules by
(i) $Z \to \alpha_i|\alpha_i Z, 1 \le i \le r$ and (ii) $A \to \beta_i|\beta_i Z, 1 \le i \le s$

• If $G = (V, T, R, S)$ is a CFG, then we can construct another CFG $G_1 = (V_1, T, R_1, S)$ in **Greibach Normal Form (GNF)** such that $L(G_1) = L(G) - \{\epsilon\}$.

The stepwise algorithm is as follows:

1. Eliminate null productions, unit productions and useless symbols from the grammar G and then construct a $G' = (V', T, R', S)$ in **Chomsky Normal Form (CNF)** generating the language $L(G') = L(G) - \{\epsilon\}$.

2. Rename the variables like $A_1, A_2, \ldots A_n$ starting with $S = A_1$.

3. Modify the rules in R' so that if $A_i \to A_j\gamma \in R'$ then $j > i$

4. Starting with A_1 and proceeding to A_n this is done as follows:

   (a) Assume that productions have been modified so that for $1 \le i \le k$, $A_i \to A_j\gamma \in R'$ only if $j > i$

   (b) If $A_k \to A_j\gamma$ is a production with $j < k$, generate a new set of productions substituting for the A_j the body of each A_j production.

   (c) Repeating (b) at most $k - 1$ times we obtain rules of the form $A_k \to A_p\gamma, p \ge k$

   (d) Replace rules $A_k \to A_k\gamma$ by removing left-recursion as stated above.

5. Modify the $A_i \to A_j\gamma$ for $i = n-1, n-2, ., 1$ in desired form at the same time change the Z production rules.

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 A_2

   A with A_3

   B with A_4

   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 $A_2 \to b$

   $A_4 \to bA_3A_4|A_4A_4A_4|b$

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

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

$Z \rightarrow A_4A_4|A_4A_4Z$

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

$\underline{\text{for } A_1 \rightarrow A_2A_3|A_4A_4}$

Substitute for A_2 and A_4 to convert it to GNF

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

$\underline{\text{for } Z \rightarrow A_4A_4|A_4A_4Z}$

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

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

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

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

$A_2 \rightarrow b$

$A_3 \rightarrow a$