Greibach Normal Form (GNF)

GNF makes the parsing or derivation linear.

-> Every step in derivation introduces exactly one terminal symbol
-> so that with GNF and LMD exactly n steps later , a string of length n is derived
Every Production in GNF looks like
A-> a α
a∈ T
a∈V*
Or
A -> λ
Leftmost symbol on RHS of every production is a terminal
There cannot be more than one terminal symbol on RHS
Note: There is specific algorithm to convert CFG to GNF

Example 1:
S -> aSb | bSb | SS | **λ**

GNF:
S -> aSB
S -> bSA
S -> λ
B -> b
A -> A

For S -> SS (Replace the first S by S -> aSB | bSA
Hence we get,
S -> aSbS | aSAS

Therefore the grammar in GNF is
S -> aSB | bSA | aSBA | bSAS | **λ**
A -> a
B -> b

Example 2:
S -> XY | Xn | p
X -> mX | m
Y -> Wn| o

Replace X in S and Y
S -> mXY | mY | mXn | mn | p
Y -> mXn | mn | o

Grammar in GNF is :
S -> mXY | mY | mXN | mN | p
X -> mXN | mN | o
N -> n