In a CFG, it may happen that all the production rules and symbols are not needed for the derivation of strings. Besides, there may be some null productions and unit productions. Elimination of these productions and symbols is called **simplification of CFGs**. Simplification essentially comprises of the following steps −

* Reduction of CFG
* Removal of Unit Productions
* Removal of Null Productions

Reduction of CFG

CFGs are reduced in two phases −

**Phase 1** − Derivation of an equivalent grammar, **G’**, from the CFG, **G**, such that each variable derives some terminal string.

**Derivation Procedure** −

Step 1 − Include all symbols, **W1**, that derive some terminal and initialize **i=1**.

Step 2 − Include all symbols, **Wi+1**, that derive **Wi**.

Step 3 − Increment **i** and repeat Step 2, until **Wi+1 = Wi**.

Step 4 − Include all production rules that have **Wi** in it.

**Phase 2** − Derivation of an equivalent grammar, **G”**, from the CFG, **G’**, such that each symbol appears in a sentential form.

**Derivation Procedure** −

Step 1 − Include the start symbol in **Y1** and initialize **i = 1**.

Step 2 − Include all symbols, **Yi+1**, that can be derived from **Yi** and include all production rules that have been applied.

Step 3 − Increment **i** and repeat Step 2, until **Yi+1 = Yi**.

Problem

Find a reduced grammar equivalent to the grammar G, having production rules, P: S → AC | B, A → a, C → c | BC, E → aA | e

Solution

**Phase 1** −

T = { a, c, e }

W1 = { A, C, E } from rules A → a, C → c and E → aA

W2 = { A, C, E } U { S } from rule S → AC

W3 = { A, C, E, S } U ∅

Since W2 = W3, we can derive G’ as −

G’ = { { A, C, E, S }, { a, c, e }, P, {S}}

where P: S → AC, A → a, C → c , E → aA | e

**Phase 2** −

Y1 = { S }

Y2 = { S, A, C } from rule S → AC

Y3 = { S, A, C, a, c } from rules A → a and C → c

Y4 = { S, A, C, a, c }

Since Y3 = Y4, we can derive G” as −

G” = { { A, C, S }, { a, c }, P, {S}}

where P: S → AC, A → a, C → c

Removal of Unit Productions

Any production rule in the form A → B where A, B ∈ Non-terminal is called **unit production.**.

Removal Procedure −

**Step 1** − To remove **A → B**, add production **A → x** to the grammar rule whenever **B → x** occurs in the grammar. [x ∈ Terminal, x can be Null]

**Step 2** − Delete **A → B** from the grammar.

**Step 3** − Repeat from step 1 until all unit productions are removed.

**Problem**

Remove unit production from the following −

S → XY, X → a, Y → Z | b, Z → M, M → N, N → a

**Solution** −

There are 3 unit productions in the grammar −

Y → Z, Z → M, and M → N

**At first, we will remove M → N.**

As N → a, we add M → a, and M → N is removed.

The production set becomes

S → XY, X → a, Y → Z | b, Z → M, M → a, N → a

**Now we will remove Z → M.**

As M → a, we add Z→ a, and Z → M is removed.

The production set becomes

S → XY, X → a, Y → Z | b, Z → a, M → a, N → a

**Now we will remove Y → Z.**

As Z → a, we add Y→ a, and Y → Z is removed.

The production set becomes

S → XY, X → a, Y → a | b, Z → a, M → a, N → a

Now Z, M, and N are unreachable, hence we can remove those.

The final CFG is unit production free −

S → XY, X → a, Y → a | b

Removal of Null Productions

In a CFG, a non-terminal symbol **‘A’** is a nullable variable if there is a production **A → ε** or there is a derivation that starts at **A** and finally ends up with

ε: A → .......… → ε

Removal Procedure

**Step 1** − Find out nullable non-terminal variables which derive ε.

**Step 2** − For each production **A → a**, construct all productions **A → x** where **x** is obtained from **‘a’** by removing one or multiple non-terminals from Step 1.

**Step 3** − Combine the original productions with the result of step 2 and remove **ε - productions**.

**Problem**

Remove null production from the following −

S → ASA | aB | b, A → B, B → b | ∈

**Solution** −

There are two nullable variables − **A** and **B**

**At first, we will remove B → ε.**

After removing **B → ε**, the production set becomes −

S→ASA | aB | b | a, A ε B| b | &epsilon, B → b

**Now we will remove A → ε.**

After removing **A → ε**, the production set becomes −

S→ASA | aB | b | a | SA | AS | S, A → B| b, B → b

This is the final production set without null transition.