**Chomsky's Normal Form (CNF)**

* CNF stands for Chomsky normal form.
* A CFG(context free grammar) is in CNF(Chomsky normal form) if all production rules satisfy one of the following conditions:
* Start symbol generating ε. For example, A → ε.
* A non-terminal generating two non-terminals. For example, S → AB.
* A non-terminal generating a terminal. For example, S → a.

**For example:**

1. G1 = {S → AB, S → c, A → a, B → b}
2. G2 = {S → aA, A → a, B → c}
* The production rules of Grammar G1 satisfy the rules specified for CNF, so the grammar G1 is in CNF.
* However, the production rule of Grammar G2 does not satisfy the rules specified for CNF as S → aZ contains terminal followed by non-terminal. So the grammar G2 is not in CNF.

## Steps for converting CFG into CNF

**Step 1:** Eliminate start symbol from the RHS. If the start symbol T is at the right-hand side of any production, create a new production as:

1. S1 → S

Where S1 is the new start symbol.

**Step 2:** In the grammar, remove the null, unit and useless productions.

**Step 3:** Eliminate terminals from the RHS of the production if they exist with other non-terminals or terminals. For example, production S → aA can be decomposed as

1. S → RA
2. R → a

**Step 4:** Eliminate RHS with more than two non-terminals. For example, S → ABC can be decomposed as:

1. S → RC
2. R → AB

Example:

Convert the given CFG to CNF. Consider the given grammar G1:

1. S → a | aA | B
2. A → aBB | ε
3. B → Aa | b

**Solution:**

**Step 1:** We will create a new production S1 → S, as the start symbol S appears on the RHS. The grammar will be:

1. S1 → S
2. S → a | aA | B
3. A → aBB | ε
4. B → Aa | b

**Step 2:** As grammar G1 contains A → ε null production, its removal from the grammar yields:

1. S1 → S
2. S → a | aA | B
3. A → aBB
4. B → Aa | b | a

Now, as grammar G1 contains Unit production S → B, its removal yield:

1. S1 → S
2. S → a | aA | Aa | b
3. A → aBB
4. B → Aa | b | a

Also remove the unit production S1 → S, its removal from the grammar yields:

1. S0 → a | aA | Aa | b
2. S → a | aA | Aa | b
3. A → aBB
4. B → Aa | b | a

**Step 3:** In the production rule S0 → aA | Aa, S → aA | Aa, A → aBB and B → Aa, terminal a exists on RHS with non-terminals. So we will replace terminal a with X:

1. S0 → a | XA | AX | b
2. S → a | XA | AX | b
3. A → XBB
4. B → AX | b | a
5. X → a

**Step 4:** In the production rule A → XBB, RHS has more than two symbols, removing it from grammar yield:

1. S0 → a | XA | AX | b
2. S → a | XA | AX | b
3. A → RB
4. B → AX | b | a
5. X → a
6. R → XB

Hence, for the given grammar, this is the required CNF.