mood-book

UNIT- IV

Simplification of context free grammar

(1) Elimination of ε production
(2) Removal of unit production
(3) Removal of useless symbol

(1) Elimination of ε production

The productions of the form $S \to \varepsilon$ are called ε-productions, where S is any nonterminal.

Procedure:

(i) Find all nullable nonterminals (or) Variables which derives ε. (Directly (or) indirectly)

Example:

$$A \to \varepsilon \mid xA \qquad C \to AB$$
$$B \to aA \mid b \mid \varepsilon$$

A, B are nullable nonterminals.

C also nullable through A, B.

(ii) Replace nullable nonterminals by ε in the Right side of the given production and find new productions. Combine original, new productions.

problems:

1 Eliminate ε productions in the following CFG.

$$S \rightarrow ABCd$$
$$A \rightarrow BC$$
$$B \rightarrow bB$$
$$B \rightarrow \varepsilon$$
$$C \rightarrow cC$$
$$C \rightarrow \varepsilon$$

Solution:

Nullable nonterminals are B, C directly.

A is nullable through B, C.

So A, B, C are nullable variables.

Let consider first production

$$S \rightarrow ABCd \quad \text{--- (1)}$$

Replace nullable variables by ε find all possible combinations.

$A \rightarrow \varepsilon$	$B \rightarrow \varepsilon$	$C \rightarrow \varepsilon$
$S \rightarrow BCd$	$S \rightarrow ACd$	$S \rightarrow ABd$
$A \rightarrow \varepsilon, B \rightarrow \varepsilon$	$B \rightarrow \varepsilon, C \rightarrow \varepsilon$	$A \rightarrow \varepsilon, C \rightarrow \varepsilon$
$S \rightarrow Cd$	$S \rightarrow Ad$	$S \rightarrow Bd$

$$\frac{A \to \varepsilon,\ B \to \varepsilon,\ C \to \varepsilon}{S \to d}$$

So S productions are

$$\boxed{S \to ABCd \mid ABd \mid ACd \mid BCd \mid Ad \mid Bd \mid Cd \mid d}$$

Let consider second production

$$A \to BC \text{ ——— } (2)$$

$$\frac{B \to \varepsilon}{A \to C} \qquad \frac{C \to \varepsilon}{A \to B}$$

A productions are

$$\boxed{A \to BC \mid B \mid C}$$

Let consider third production

$$B \to bB \longrightarrow (3)$$

$$\frac{B \to \varepsilon}{B \to b}$$

So B productions are

$$\boxed{B \to bB \mid b}$$

The fourth production is

$$C \to cC$$

$\underline{C \rightarrow \varepsilon}$

$C \rightarrow c$

So the C productions are

$$\boxed{C \rightarrow cC \mid c}$$

The grammar after removal of ε productions are

$S \rightarrow ABCd \mid ABd \mid ACd \mid BCd \mid Ad \mid Bd \mid Cd \mid d$

$A \rightarrow BC \mid B \mid C$

$B \rightarrow bB \mid b$

$C \rightarrow cC \mid c$

(2) Remove ε productions from

$S \rightarrow ASA \mid aB \mid b$

$A \rightarrow B$

$B \rightarrow b \mid \varepsilon$

Solution:

Due to $B \rightarrow \varepsilon$, B is nullable variable.

Because of $A \to B$ productions, A also nullable.

So the nullable variables are $\{A, B\}$.

Consider

$$S \to ASA \mid aB \mid b \quad \text{——— (1)}$$

$\underline{A \to \varepsilon}$ $\qquad$ $\underline{A \to \varepsilon}$ $\qquad$ $\underline{B \to \varepsilon}$

$S \to AS$ $\qquad$ $S \to SA$ $\qquad$ $S \to a$

S productions are

$$\boxed{S \to ASA \mid AS \mid SA \mid aB \mid a \mid b}$$

Let consider

$$A \to B \quad \text{——— (2)}$$

No Replacement possible.

B production $B \to b$ — (3), here also no replacement possible. So After removal of ε the productions are

$$\boxed{\begin{array}{l} S \to ASA \mid AS \mid SA \mid aB \mid a \mid b \\ A \to B \\ B \to b \end{array}}$$

2 Removal of Unit production:

Any production rule of the form $\boxed{A \rightarrow B}$ Where A, B are nonterminals (or) variables is called unit production.

Procedure:

Step1: To remove $A \rightarrow B$, add production $A \rightarrow x$, whenever $B \rightarrow x$ occurs in the grammar, where x is terminal (or) Epsilon.

Step 2: Remove $A \rightarrow B$ from the grammar.

Step3: Repeat 1, 2 untill all Unit productions are removed.

problems:

1 Remove unit productions from the following G.

$S \rightarrow XY$

$X \rightarrow a$

$Y \rightarrow Z \mid b$

$Z \rightarrow M$

$M \rightarrow N$

$N \rightarrow a.$

Solution:

In the given grammar G Unit productions are

$Y \to Z$, $Z \to M$, $M \to N$.

Let consider $M \to N$

Replace N by a Using $N \to a$

so $\boxed{M \to a}$ delete $M \to N$.

Lest consider $Z \to M$

Replace M by a Using $M \to a$

So $\boxed{Z \to a}$ delete $Z \to M$.

Now consider production $Y \to Z$

Replace Z by a Using $Z \to a$

so $\boxed{Y \to a}$ delete $Y \to Z$

The grammar G After removal of Unit production

$S \to XY$

$X \to a$

$Y \to a \mid b$

$Z \to a$

$M \to a$

$N \to a$.

(2) Remove unit productions from the following (8)

$S \rightarrow 0A \mid 1B \mid C$

$A \rightarrow 0S \mid 00$

$B \rightarrow 1 \mid A$

$C \rightarrow 01$

Solution:

The unit productions are

$S \rightarrow C$, $B \rightarrow A$.

Let consider $S \rightarrow C$

Replace C by 01 using $C \rightarrow 01$

So $\boxed{S \rightarrow 01}$ delete $S \rightarrow C$

Let consider $B \rightarrow A$

Replace A by 0S (or) 00 using $A \rightarrow 0S \mid 00$

So $\boxed{B \rightarrow 0S \mid 00}$. The final productions are

$S \rightarrow 0A \mid 1B \mid 01$

$A \rightarrow 0S \mid 00$

$B \rightarrow 1 \mid 0S \mid 00$ $C \rightarrow 01$.

(3) Elimination of useless symbols :-

* The non terminals and terminals which are not used in the derivation are useless.
* The symbols which are not reachable from start symbol are useless.

Procedure:

(1) Identify non generating symbols and Eliminate productions which contains those symbols.

(2) Identity non reachable symbols and eliminate those productions which contains non reachable symbols.

Problems:

(1) Eliminate useless symbols from

$S \to aB \mid bX$

$A \to BAd \mid bSX \mid a$

$B \to aSB \mid bBX$

$X \to SBD \mid aBX \mid ad$

Solution:

Solution:

The non terminals are S, A, B, X.

X derives terminal ad, X is useful

A derives terminal a, A is useful.

But B does not derive terminal

$$B \rightarrow aSB \mid bBX$$

So B is useless symbol.

S is useful because $S \rightarrow bX$, where X is useful so S can also derive.

So Remove productions, which contain B.

$S \rightarrow aB \mid bX$ becomes $S \rightarrow bX$

$A \rightarrow BAd \mid bSX \mid a$ becomes $A \rightarrow bSX \mid a$

$B \rightarrow aSB \mid bBX$ becomes no production.

$X \rightarrow SBD \mid aBX \mid ad$ becomes $X \rightarrow ad$.

Now the productions in G are.

$$S \rightarrow bX$$
$$A \rightarrow bSX \mid a$$
$$X \rightarrow ad.$$

But A is not reachable from Start symbol S. So remove A productions.

So finally

$$\boxed{\begin{array}{l} S \to bX \\ X \to ad \end{array}}$$

2 Remove useless symbols from

$S \to aS \mid A \mid C$

$A \to a$

$B \to aa$

$C \to aCb$

Solution:

The non terminals are S, A, B, C

C is useless, because does not derive terminal. ~~So Remove~~.

A, B are usefull, because A derives a, B derives aa. S is useful, because S can derive terminal through $S \to A$.

So Eliminate productions with C

So

$S \to aS \mid A \mid \cancel{C}$ becomes $S \to aS \mid A$ (12)

$C \to aCb$ becomes no production.

After C Removal the G is

$S \to aS \mid A$

$A \to a$

$B \to aa$.

Nonterminal B can't be reachable from Start symbol S. So remove B production.

Finally G is

$$\boxed{\begin{array}{l} S \to aS \mid A \\ A \to a \end{array}}$$

Normal Forms:

A grammar G is said to be in normal form if its productions have a special structure.

Types of normal forms

(i) Chomsky Normal form (CNF)

(ii) Greibach Normal form (GNF).

1 Chomsky Normal form (CNF):

A CFG is said to be in CNF if it has productions of the form

$A \to BC$

$A \to a$ ($S \to \varepsilon$ only if S is start symbol)

where A, B, C are nonterminals, a is terminal.

Example:

1. $S \to AS$
   $S \to a$
   $A \to SA$
   $A \to b$

The above CFG is in CNF.

2. $S \to AS$
   $S \to AAS$ (3 nonterminals)
   $A \to SA$

The above CFG is not in CNF due to production number 2.

Procedure to convert into CNF:

Step1: Eliminate start symbol from the RHS. If start symbol S appears at RHS of any production, create a new production as

$\boxed{S1 \rightarrow S}$ S1 is new start symbol.

Step2: Remove null, unit and useless productions.

Step3: For every terminal a, add production $T_a \rightarrow a$, replace a by T_a in the production, where T_a is nonterminal.

Step4: Replace any production

$A \rightarrow C_1 C_2 \cdots C_n$ with

$A \rightarrow C_1 V_1$

$V_1 \rightarrow C_2 V_2$

$\vdots$

$V_{n-2} \rightarrow C_{n-1} C_n$

where $V_1, V_2 \ldots$ are nonterminals, which used as Intermediate.

Problems:

(1) Convert to CNF

$S \to ABa$

$A \to aab$

$B \to Ac$

Solution:

S1 Start Symbol S does not appear at RHS of any production. So no need to introduce new start symbol.

S2 No null, unit & useless productions.

S3 The terminals are a, b, c so introduce

$T_a \to a, \quad T_b \to b, \quad T_c \to c$

$S \to ABT_a$ —— (1)

$A \to T_a T_a T_b$ —— (2)

$B \to AT_c$ —— (3)

$T_a \to a$ —— (4)

$T_b \to b$ —— (5)

$T_c \to c$ —— (6)

3, 4, 5, 6 are in CNF.

S4

Consider production 1. from S3.

$S \rightarrow A\underbrace{B T_a}_{V_1}$ $\quad S \rightarrow A V_1$, $V_1 \rightarrow B T_a$ } Now in CNF.

Let consider production 2 from S3.

$A \rightarrow T_a \underbrace{T_a T_b}_{V_2}$ $\quad A \rightarrow T_a V_2$, $V_2 \rightarrow T_a T_b$ } Now in CNF.

The productions in CNF are

$S \rightarrow A V_1$

$V_1 \rightarrow B T_a$

$A \rightarrow T_a V_2$

$V_2 \rightarrow T_a T_b$

$B \rightarrow A T_c$

$T_a \rightarrow a$

$T_b \rightarrow b$

$T_c \rightarrow c$

(2) Convert to CNF

$S \rightarrow AbA$

$A \rightarrow Aa \mid \varepsilon$

Solution:

1. Start symbol S does not appear at RHS of any production. So no need to introduce new start symbol.
2. No unit, useless productions. But we have ε- productions. A is nullable nonterminal.

To Eliminate ε at S

$S \rightarrow AbA \mid bA \mid Ab \mid b$

To Eliminate ε at A

$A \rightarrow Aa \mid a$

3. The terminals are a, b so introduce Ta, Tb

$S \rightarrow AT_bA \mid T_bA \mid AT_b \mid b$

$A \rightarrow AT_a \mid a$

$T_a \rightarrow a$

$T_b \rightarrow b$

Except $S \rightarrow AT_bA$ all productions are in CNF.

4. So Let consider

$$S \rightarrow A\underbrace{T_b A}_{V_1}$$

$$\left.\begin{aligned} S &\rightarrow AV_1 \\ V_1 &\rightarrow T_b A \end{aligned}\right\} \text{now in CNF.}$$

The productions are

$S \rightarrow AV_1$

$V_1 \rightarrow T_b A$

$S \rightarrow T_b A$

$S \rightarrow AT_b$

$S \rightarrow b$

$A \rightarrow AT_a$

$A \rightarrow a$

$T_a \rightarrow a$

$T_b \rightarrow b$

(3) Convert to CNF

$S \rightarrow 1A$

$S \rightarrow 0B$

$A \rightarrow 1AA$

$A \rightarrow 0S$

$A \rightarrow 0$

$B \rightarrow 0BB$

$B \rightarrow 1S$

$B \rightarrow 1$

Solution:

1. Start Symbol S appears at RHS Of production. So Introduce S1.

$$S1 \rightarrow S.$$

2. No ε-productions, Unit productions and useless Symbols.

3. The terminals are 0, 1 so introduce T_0, T_1.

$$S1 \rightarrow S$$
$$S \rightarrow T_1 A$$
$$S \rightarrow T_0 B$$
$$A \rightarrow T_1 A A$$
$$A \rightarrow T_0 S$$
$$A \rightarrow 0$$
$$B \rightarrow T_0 B B$$
$$B \rightarrow T_1 S$$
$$B \rightarrow 1$$

At S1 replace with S productions.

$$S1 \rightarrow T_1 A$$
$$S1 \rightarrow T_0 B$$

Except $A \rightarrow T_1 AA$, $B \rightarrow T_0 BB$ all are in CNF.

4. Let consider

$$A \to T_1 \underbrace{A A}_{V_1}$$

$$\left.\begin{array}{l} A \to T_1 V_1 \\ V_1 \to A A \end{array}\right\}$$ now in CNF

Now consider

$$B \to T_0 \underbrace{B B}_{V_2}$$

$$\left.\begin{array}{l} B \to T_0 V_2 \\ V_2 \to B B \end{array}\right\}$$ now in CNF.

So the CFG in CNF is

$S1 \to T_1 A$

$S1 \to T_0 B$

$S \to T_1 A$

$S \to T_0 B$

$A \to T_1 V_1$

$V_1 \to A A$

$A \to T_0 S$

$A \to 0$

$B \to T_0 V_2$

$V_2 \to B B$

$B \to T_1 S$

$B \to 1.$

Greibach Normal Form (GNF)

A grammar G is said to be in GNF if it has productions of the form

$$\boxed{\begin{array}{l} A \rightarrow aB_1B_2 \ldots B_n \\ A \rightarrow a \end{array}}$$

Where $A, B_1, B_2 \ldots B_n$ are Non terminals.
a is a terminal.
Only for Start symbol ε production is allowed.

Example:

① $S \rightarrow aABC$
$A \rightarrow 0$
$B \rightarrow 1B \mid 1$
$C \rightarrow 2C \mid 2$

The above grammar is in GNF.

② $S \rightarrow \underline{ABC}$
$A \rightarrow a \mid \underline{BC}$
$B \rightarrow b$
$C \rightarrow c$

The above grammar is not in GNF.

Procedure to convert CFG to GNF

1. Convert given CFG into CNF first.

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

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

$$A_i \rightarrow A_j x_1 x_2 \ldots x_n \quad \boxed{j > i}$$

$$A_i \rightarrow a x_1 x_2 \ldots x_n \quad a \in T, \; x_1 \ldots x_n \in V$$

$$Z_i \rightarrow A_j x_1 x_2 \ldots x_n \quad \boxed{j \leq n}$$

If any such productions exists, Apply Substitution.

4. Identify Any productions with Left recursion. Eliminate Left recursion.

if $A \rightarrow A\alpha \mid \beta$, this production has Left recursion.

To Eliminate,

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A' \mid \varepsilon$$

Now Eliminate Epsilon productions.

5. Apply substitutions whereever required to get CFG in GNF.

Problems:

① Convert the following G into GNF.

$S \rightarrow XA \mid BB$

$B \rightarrow b \mid SB$

$X \rightarrow b$

$A \rightarrow a$

Solution:

1. The given grammar is already in CNF. S appears at RHS so introduce $\boxed{S_1 \rightarrow S.}$

2. Renaming Variables:

$S = A_1,\ X = A_2,\ A = A_3,\ B = A_4$

$A_1 \rightarrow A_2 A_3 \mid A_4 A_4$

$A_4 \rightarrow b \mid A_1 A_4$

$A_2 \rightarrow b$

$A_3 \rightarrow a.$

3. Identify productions which donot conform

$\underset{i}{A_1} \rightarrow \underset{j}{A_2} A_3 \mid \underset{j}{A_4} A_4$ $\quad 2 > 1 \quad 4 > 1$

$\underset{i}{A_4} \rightarrow b \mid \underset{j}{A_1} A_4 \quad 1 \ngtr 4.$

$A_2 \rightarrow b \quad A_3 \rightarrow a.$

The production $A_4 \to A_1A_4$ does not satisfy any of the rules.

So substitute A_1 in A_4.

$$\underset{i}{A_4} \to \underset{j}{A_2}A_3A_4 \mid \underset{j}{A_4}A_4A_4 \mid b$$

$2 \ngtr 4$ $\qquad$ $4 \ngtr 4$

So substitute b in the place of A_2.

$$A_4 \to \underbrace{bA_3A_4}_{(GNF)} \mid A_4A_4A_4 \mid b$$

$4 = 4$.

4. Identify and Eliminate left recursion.

$$\underbrace{A_4}_{A} \to \underbrace{A_4}_{A}\underbrace{A_4A_4}_{\alpha} \mid \underbrace{bA_3A_4 \mid b}_{\beta}$$

$$A_4 \to bA_3A_4Z \mid bZ$$

$$Z \to A_4A_4Z \mid \varepsilon$$

Eliminate ε productions.

$$A_4 \to bA_3A_4 \mid b \mid bA_3A_4Z \mid bZ$$

$$Z \to A_4A_4Z \mid A_4A_4.$$

Now the productions of G are.

$\boxed{A_1 \to A_2A_3 \mid A_4A_4}$ not in GNF

$A_4 \to bA_3A_4 \mid b \mid bA_3A_4Z \mid bZ$

$\boxed{Z \to A_4A_4Z \mid A_4A_4}$ not in GNF.

$A_2 \to b \quad A_3 \to a.$

5. Substitution:

(25)

$A_1 \rightarrow A_2A_3 \mid A_4A_4$.

Substitute A_2, A_4 at A_1

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

Now A_1 is in GNF.

Let consider $Z \rightarrow A_4A_4 \mid A_4A_4Z$

Substitute A_4 in Z.

$Z \rightarrow bA_3A_4A_4 \mid bA_4 \mid bA_3A_4ZA_4 \mid bZA_4 \mid$
$bA_3A_4Z \mid bA_3A_4A_4Z \mid bA_4Z \mid bZA_4Z$

Now Z also in GNF.

Finally the GNF grammar is

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

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

$Z \rightarrow bA_3A_4A_4 \mid bA_4 \mid bA_3A_4ZA_4 \mid bZA_4 \mid$
$bA_3A_4Z \mid bA_3A_4A_4Z \mid bA_4Z \mid bZA_4Z$

$A_2 \rightarrow b$

$A_3 \rightarrow a$. because $S1 \rightarrow S$, Substitute S at S1

$S1 \rightarrow bA_3 \mid bA_3A_4A_4 \mid bA_4 \mid bA_3A_4ZA_4 \mid bZA_4$

(2) Convert following Grammar into GNF.

$S \to Ax \mid a$

$A \to SY \mid b$

$X \to a$

$Y \to b$

Solution:

(1) The given grammar is already in CNF.
S appears at right side so $S1 \to S$.

(2) Rename Variables

$S = A_1,\ A = A_2,\ X = A_3,\ Y = A_4$.

$A_1 \to A_2 A_3 \mid a$

$A_2 \to A_1 A_4 \mid b$

$A_3 \to a$

$A_4 \to b$

(3) Identify productions do not conform

$\underset{i}{A_1} \to \underset{j}{A_2} A_3 \mid a \quad 2 > 1$

$\underset{i}{A_2} \to \underset{j}{A_1} A_4 \mid b \quad 1 \not> 2$

$A_3 \to a$

$A_4 \to b$

The production $A_2 \rightarrow A_1 A_4 | b$ not satisfies (27)

So substitute A_1 in A_2.

$$A_2 \rightarrow A_2 A_3 A_4 | aA_4 | b$$

(4) <u>Left recursion:</u>

$$\underline{A_2}_{A} \rightarrow \underline{A_2}_{A} \underbrace{A_3 A_4}_{\alpha} | \underbrace{aA_4 | b}_{\beta}$$

To Eliminate Left recursion

$$A_2 \rightarrow aA_4 Z | bZ$$

$$Z \rightarrow A_3 A_4 Z | \varepsilon$$

Eliminate ε-productions

$$A_2 \rightarrow aA_4 Z | bZ | aA_4 | b$$

$$Z \rightarrow A_3 A_4 Z | A_3 A_4$$

Now the grammar is

$\boxed{A_1 \rightarrow A_2 A_3 | a}$ not in GNF.

$A_2 \rightarrow aA_4 Z | bZ | aA_4 | b$

$\boxed{Z \rightarrow A_3 A_4 Z | A_3 A_4}$ not in GNF.

$A_3 \rightarrow a$

$A_4 \rightarrow b$

5. Substitution:

Let consider $A_1 \rightarrow A_2A_3 \mid a$

Replace A_2 with productions

$$\boxed{A_1 \rightarrow aA_4ZA_3 \mid bZA_3 \mid aA_4A_3 \mid bA_3 \mid a}$$ Now in GNF

$Z \rightarrow A_3A_4Z \mid A_3A_4$

Replace A_3 by a

$$\boxed{Z \rightarrow aA_4Z \mid aA_4}$$ Now in GNF.

Now GNF Grammar is

$A_1 \rightarrow aA_4ZA_3 \mid bZA_3 \mid aA_4A_3 \mid bA_3 \mid a$

$A_2 \rightarrow aA_4Z \mid bZ \mid aA_4 \mid b$

$Z \rightarrow aA_4Z \mid aA_4$

$A_3 \rightarrow a$

$A_4 \rightarrow b$

The new production $S1 \rightarrow S$, Replace S by A_1 productions.

$S1 \rightarrow aA_4ZA_3 \mid bZA_3 \mid aA_4A_3 \mid bA_3 \mid a.$

CLOSURE PROPERTIES OF CFL:

1 Union:

Statement: If L_1 and L_2 are CFLs, then $L_1 \cup L_2$ is a CFL.

Proof:

Let L_1 be generated by the CFG G_1 where

$G_1 = (V_1, T_1, P_1, S_1)$.

Let L_2 be generated by the CFG G_2 where

$G_2 = (V_2, T_2, P_2, S_2)$.

Subscript each nonterminal of G_1 with a 1, and each nonterminal of G_2 with a 2. $(V_1 \cap V_2 = \phi)$

Define CFG, G that generates $L_1 \cup L_2$

$$G = (V_1 \cup V_2 \cup S,\ T_1 \cup T_2,\ P_1 \cup P_2 \cup \{S \to S_1 | S_2\},\ S).$$

Each word (or) string generated by G is either in L_1 or in L_2.

Example:

$L_1 = \{$ palindromes with symbols $a, b \}$

$L_1 = \{\varepsilon, a, b, aa, bb, aba, bab, aabaa, bbabb, \ldots \}$

CFG for L_1 is

$S_1 \to aS_1a \mid bS_1b \mid a \mid b \mid \varepsilon$

$L_2 = \{ a^n b^n \mid n \geq 0 \}$

$L_2 = \{\varepsilon, ab, aabb, aaabbb, \ldots\}$

CFG for L_2 is

$S_2 \rightarrow aS_2b \mid \varepsilon$

Then the grammar for $L_1 \cup L_2$ is

$S \rightarrow S_1 \mid S_2$

$S_1 \rightarrow aS_1a \mid bS_1b \mid a \mid b \mid \varepsilon$

$S_2 \rightarrow aS_2b \mid \varepsilon$.

② Concatenation:

Statement: If L_1 and L_2 are CFLs, then L_1L_2 is a CFL.

Proof:
Let L_1 be generated by the CFG G_1, where $G_1 = (V_1, T_1, P_1, S_1)$.
Let L_2 be generated by the CFG G_2 where $G_2 = (V_2, T_2, P_2, S_2)$.
Subscript each nonterminal of G_1 with a 1, and each nonterminal of G_2 with a 2. $(V_1 \cap V_2 = \phi)$
Define CFG, G that generates L_1L_2

$G = (V_1 \cup V_2 \cup \{S\}, T_1 \cup T_2, P_1 \cup P_2 \cup \{S \rightarrow S_1S_2\}, S)$.

Each word (or) string generated is string in L_1 followed by string in L_2.

Example:

$L_1 = \{$ palindromes with $a, b \}$

$L_1 = \{a, b, aa, bb, abba, aaaa, bbbb, baab, \ldots\}$

The grammar for L_1 is

$S_1 \rightarrow aS_1a \mid bS_1b \mid a \mid b \mid \varepsilon$

$L_2 = \{a^n b^n \mid n \geq 1\}$

$L_2 = \{ab, aabb, \ldots\}$

The grammar for L_2 is

$S_2 \rightarrow aS_2b \mid \varepsilon$

Grammar for $L_1 L_2$ is

$S \rightarrow S1S2$

$S1 \rightarrow aS_1a \mid bS_1b \mid a \mid b \mid \varepsilon$

$S2 \rightarrow aS_2b \mid \varepsilon.$

③ KLEENE STAR (*):

Statement: If L_1 is a CFL, then L_1^* is a CFL.

Proof:

Let L_1 be the Language generated by G_1 Where $G_1 = (V_1, T_1, P_1, S_1)$. Subscript Each nonterminal of G with a 1.

Define CFG, G that generates L_1^*

$G = \{ V_1 \cup \{S\}, T_1, P_1 \cup \{S \to S_1 S | \varepsilon\}, S)$

Each string generated is either ε or some sequence of words in L_1.

Example:

$L_1 = \{ a^n b^n / n \geq 0 \}$ $L_1 = \{ \varepsilon, ab, aabb, \ldots \}$

$S_1 \to a S_1 b | \varepsilon$

Then L_1^* is generated by

$S \to S_1 S | \varepsilon$

$S_1 \to a S_1 b | \varepsilon$.

Let consider $W = abab \in L_1^*$

$$S \underset{lm}{\Longrightarrow} \underline{S_1} S$$

$\underset{lm}{\Longrightarrow} a \underline{S_1} b S$ (Replace S_1 by aS_1b)

$\underset{lm}{\Longrightarrow} ab \underline{S}$ (Replace S_1 by ε)

$\underset{lm}{\Longrightarrow} ab S_1 S$ (Replace S by $S_1 S$)

$\underset{lm}{\Longrightarrow} ab\, aS_1 b S$ (Replace S_1 by aS_1b)

$\underset{lm}{\Longrightarrow} abab S$ (Replace S_1 by ε)

$\underset{lm}{\Longrightarrow} abab$ (Replace S by ε)

$= \underline{\underline{W}}$

7 Substitution:

Statement: if L is a CFL over alphabet Σ and S is a Substitution on Σ such that $S(a)$ is a CFL for each a in Σ then $S(L)$ is a CFL.

Proof:

L is a CFL generated by G, where

$G = (V, T, P, S)$

Let $a \in \Sigma$, $S(a)$ is a CFL, there is a CFG for each $S(a)$

Let $G_a = (V_a, T_a, P_a, S_a)$

$S(L)$ is a CFL generated by G' where

$G' = (V', T', P', S)$

V' has V and all V for $a \in \Sigma$.

T' has T and all $a \in \Sigma$.

P' has P and all productions P_a for $a \in \Sigma$.

Example:

L = { Language of binary palindromes of even length }

CFG for L is

$S \rightarrow 0S0 \mid 1S1 \mid \varepsilon$

the terminals are 0, 1

The substitutions are

$S(0)$, $S(1)$

$S(0) \Rightarrow \{a^n b^n \mid n \geq 1\}$ $S(1) = \{xx, yy\}$

$S_0 \rightarrow aS_0b \mid ab$

$S_1 \rightarrow xx \mid yy$

Therefore CFG for $S(L)$ is

$S \rightarrow S_0 S S_0 \mid S_1 S S_1 \mid \varepsilon$

$S_0 \rightarrow aS_0b \mid ab$

$S_1 \rightarrow xx \mid yy$

(5) Rerersal:

Statement: if L is a CFL then L^R also CFL.

Proof:

L is a CFL generated by G, where

$G = (V, T, P, S)$.

L^R is generated by $G^R = (V, T, P^R, S)$

It is enough to reverse each production of the grammar $A \rightarrow \alpha$ by $A \rightarrow \alpha^R$. So L^R also CFL.

Example: $L = \{a^n b^n \mid n \geq 1\}$

G for L is

$S \rightarrow aSb \mid ab$

The $L^R = \{b^n a^n \mid n \geq 1\}$

G^R has productions as

$S \rightarrow bSa \mid ba$.

6 Homomorphism:

Statement: If L is a CFL, then $h(L)$ also CFL, where h is a homomorphism.

Proof:

L is generated by G where $G = (V, T, P, S)$. h is a homomorphism from T to T' $\boxed{h : T \rightarrow T'}$.

The CFL for homomorphism is $h(L)$, the Grammar is $G' = (V', T', P', S')$.

Where $V' = V$

For every $a \in T$, add X_a to V'.

For every production P of G, if a appears at RHS replace it by X_a.

For each X_a add the rule $\boxed{X_a \rightarrow h(a)}$

G' generates $h(L)$ which is CFL.

Example:

$S \rightarrow 0S0 \mid 1S1 \mid \varepsilon$

$h : \{0,1\} \rightarrow \{a,b\}^*$ given by

$h(0) = aba$
$h(1) = bb$

$S \to X_0 S X_0 \mid X_1 S X_1 \mid \varepsilon$

$X_0 \to aba$

$X_1 \to bb$.

CFL

Not closed under (does not satisfy)

① Intersection:-

Statement If L_1 and L_2 are CFLs then $L_1 \cap L_2$ may not be a CFL.

Proof:
proof by example.

$L_1 = \{a^n b^n c^m \mid n, m \geq 0\}$ is CFL

$S \to XA$
$X \to aXb \mid \varepsilon$
$A \to Cc \mid \varepsilon$

$L_2 = \{a^n b^m c^m \mid n, m \geq 0\}$ is CFL

$S \to AX$
$A \to Aa \mid \varepsilon$

$X \rightarrow bXc \mid \varepsilon$

The intersection of L_1, L_2 is

$L_1 \cap L_2 = \{a^n b^n c^n, n \geq 0\}$ is not a CFL.

No G, PDA exists for $a^n b^n c^n$.

② Complement:

Statement: If L_1 is a CFL then $\overline{L_1}$ may not be a CFL

Proof:

Assume that complement of CFL is CFL.

Let L_1 and L_2 be any two CFLs.

we assume complement also CFL.

$L_1 \cup L_2$ is CFL,

So $\overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2$ is a CFL.

But $L_1 \cap L_2$ is may not be CFL.

So our assumption $\overline{L_1}$ is CFL is false.

Hence Complement of CFL may not be a CFL.

Example:

$L = \{a^n b^n \mid n \geq 1\}$

$L = \{ab, aabb, aaabbb, \ldots\}$

$\overline{L} = \{$number of a's not equal to number of b's$\}$

$\bar{L}$ = { number of a's morethan number of b's $\cup$
number of a's Less than number of b's }

$$\bar{L} = \{\{a^i b^j \mid i > j\} \cup \{a^i b^j \mid i < j\}\}$$

Decision properties

① **Finiteness:**

If L is a CFL, then L is finite or infinite.

procedure:

1. Reduce given grammar by Eliminating ε-productions, unit productions & useless productions.
2. Draw Directed graph, nodes are nonterminals, edge exists if there is a production between nonterminals.
3. If directed graph contains cycle, then the Language is infinite otherwise finite.

Example:

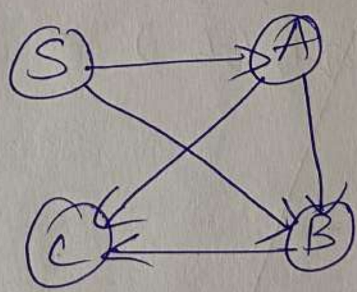
1. $S \to AB \mid a$

   $A \to BC \mid b$

   $B \to CC \mid c$ $\quad C \to x$

Step1: The given grammar does not Contain ε, unit & useless symbols.

step2: The nonterminals are S, A, B, C.



Step3: Graph does not contain cycle. So Language of the given grammar is finite.

2. $S \to XS \mid b$

   $X \to YZ$

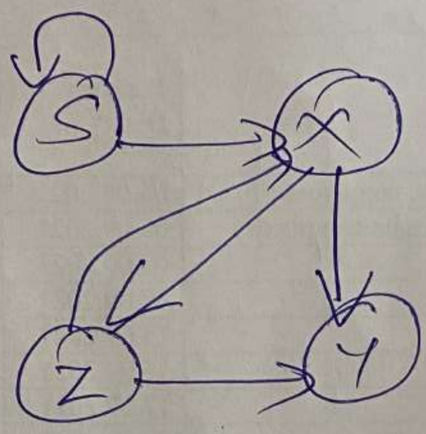
   $Y \to ab$

   $Z \to XY \mid c$

Step1: The given grammar does not Contain ε, Unit & useless productions.

Step 2:

The nonterminals are S, X, Y, Z



Step 3: Graph contains cycles.

∴ The language is infinite.

(2) Emptiness:

If L is a CFL, then L is empty or not.

Procedure:

Step 1: Remove all useless symbols from the grammar. If start symbol is useless then language is empty otherwise not.

Example:

$S \rightarrow XY$

$X \rightarrow AX \mid AA$

$A \rightarrow a$

$Y \rightarrow B \mid BB \quad B \rightarrow b$

Start symbol S is generating so it is useful.

let $S \underset{lm}{\Longrightarrow} \underline{X}Y$

$\underset{lm}{\Longrightarrow} \underline{A}AY$

$\underset{lm}{\Longrightarrow} a\underline{A}Y$

$\underset{lm}{\Longrightarrow} aa\underline{Y}$

$\underset{lm}{\Longrightarrow} aa\underline{B}B$

$\underset{lm}{\Longrightarrow} aab\underline{B}$

$S \underset{lm}{\Longrightarrow} aabb$

one of the string is aabb. So L(G) is not empty.

(2) $S \rightarrow XY$

$X \rightarrow 1X \mid 1$

$Y \rightarrow 2Y$

X is generating but Y is not generating. Y is useless. because of Y, S also useless.

$L(G) = \{ \} = \phi$. so empty.

3 Membership:

If L is a CFL, whether String w belongs to L or not.

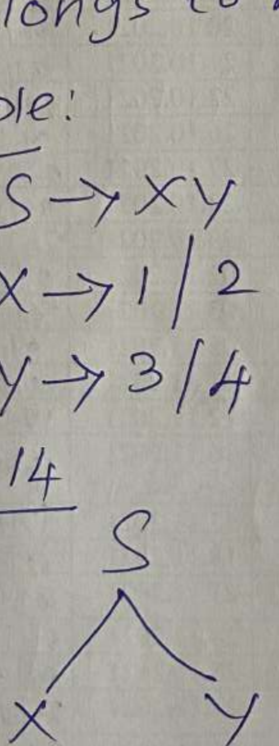
Procedure:

1. Convert grammar G for L into Chomsky Normal Form (CNF).
2. If length of w is n, then 2n-1 steps are required in derivations to determine w can be derived or not.
3. Try all 2n-1 derivations, still w can't be obtained from G then w does not belongs to L. Otherwise w belongs to L.

Example:

1. $S \to XY$
   $X \to 1/2$
   $Y \to 3/4$

w = 14



w can be derivable from G
so w belongs to L.

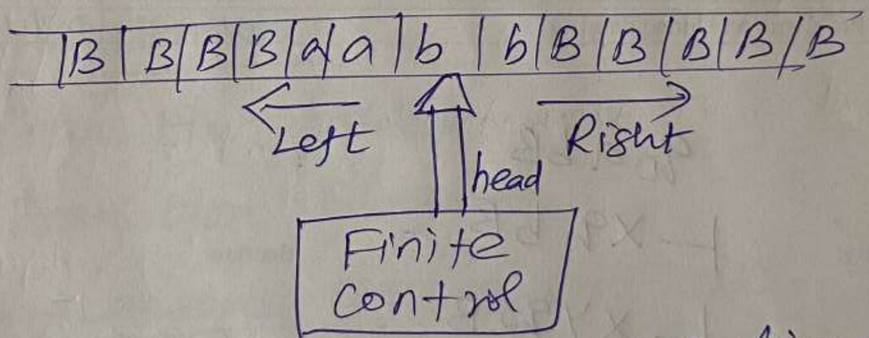
Turing Machine (TM):

* TM used to accept Recursively Enumerable Language (REL)
* REL is generated by unrestricted (or) phase structured grammar.

Turing Machine Components

① tape → divided into cells, Each cell can hold one symbol

② Finite control. → It can be in any one of a finite states.



The head can move both directions from the current positions either left or right. The tape filled with blank symbols B.

Formal Definition:

TM is a 7-tuple denoted by

$(Q, \Sigma, q_0, F, \delta, \Gamma, B)$

$Q \rightarrow$ finite set of states

$\Sigma \rightarrow$ Input symbols.

$q_0 \rightarrow$ Initial state

$F \rightarrow$ Set of all Final states

$\Gamma \rightarrow$ tape symbols

$B \rightarrow$ Blank symbol

$\delta \rightarrow$ Transition Function

$$\boxed{Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}}$$

$L \rightarrow$ Left

$R \rightarrow$ Right

Example:

$$(q_0, 1) \rightarrow (q_1, X, L)$$

Change State q_0 to q_1, Change 1 by X, Move Left.

TM representation:

1. Transition tables
2. Transition Diagrams.

Instantaneous Description (ID):

$\Rightarrow$ To verify whether given string is accepted or not.

1. Construct TM for the Language $L = \{a^n b^n \mid n \geq 1\}$

Solution:

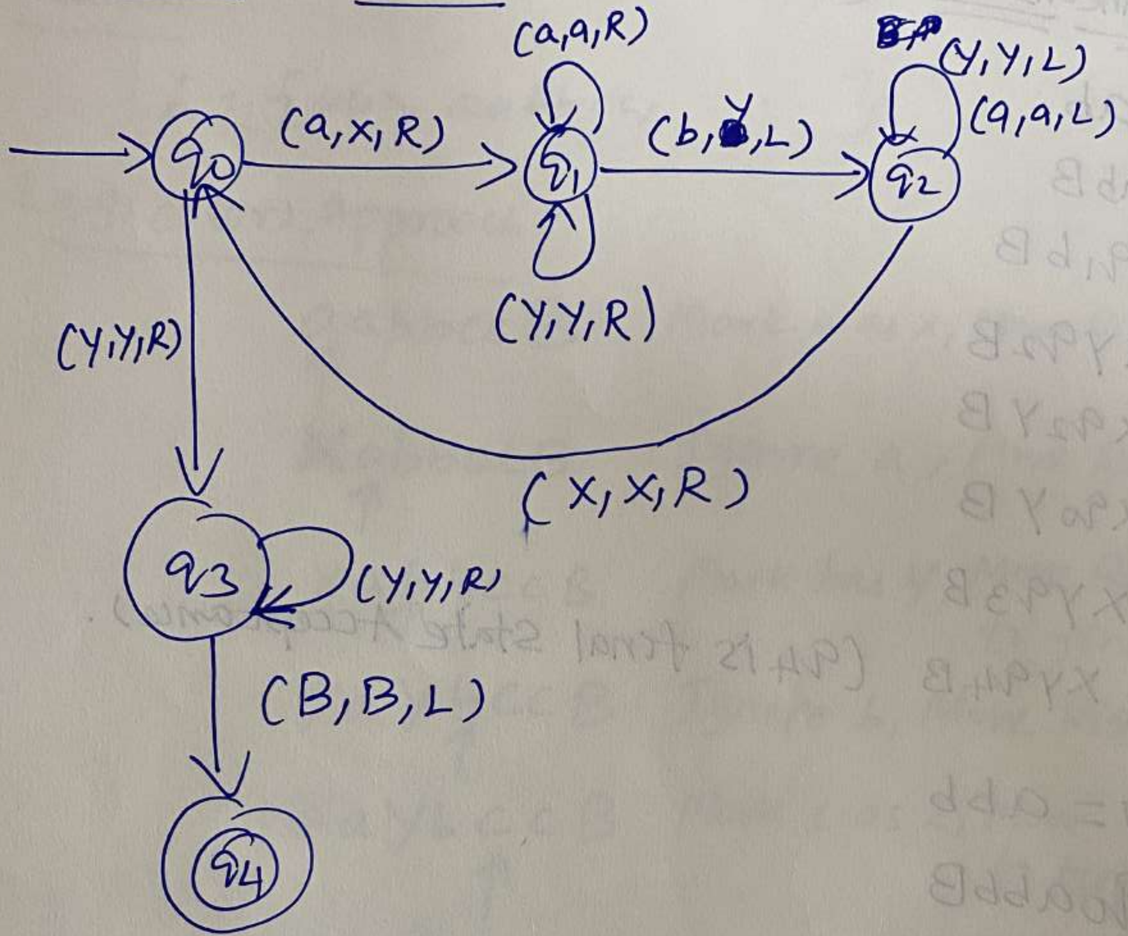
$L = \{$ Equal number of a's and b's $\}$

$L = \{ab, aabb, \ldots\}$

Logic $w = aabb$.

aabbB	Convert a to x Move right in search of b.
XabbB	Skip it, move ahead
XabbB	Convert b to y and move left till X.
XaybB	Move left
XaybB	Move right
XaybB	Convert a to x Move right in search of b.
XxybB	Move right
XxybB	Convert b to y move left till X.
XxyyB	Move left
XxyyB	Now Immediately before y, x so all a's are Converted into x. Move right
XxyyB	Move right
XxyyB	Move right
XxyyB	Stop and Accept.

Transition Diagram:



Transition table:

	a	b	X	Y	B
q0	(q1, X, R)	—	—	(q3, Y, R)	—
q1	(q1, a, R)	(q2, Y, L)	—	(q1, Y, R)	—
q2	(q2, a, L)	—	(q0, X, R)	(q2, Y, L)	—
q3	—	—	—	(q3, Y, R)	(q4, B, L)
q4	—	—	—	—	—

Instantaneous Description (ID):

$W = ab$

q_0abB

$\vdash Xq_1bB$

$\vdash XYq_2B$

$\vdash Xq_2YB$

$\vdash Xq_0YB$

$\vdash XYq_3B$

$\vdash XYq_4B$ (q_4 is final state Acceptance).

$W = abb$

q_0abbB

$\vdash Xq_1bbB$

$\vdash XYq_2bB$

$\vdash Xq_2YbB$

$\vdash Xq_0YbB$

$\vdash XYq_3bB$ (No transition on b at q_3)

So abb is not accepted. because we are unable to reach final state q_4.

② Construct TM for $L = \{a^n b^n c^n \mid n \geq 1\}$

Solution:

$L = \{abc, aabbcc, \ldots\}$

Logic (or) Approach:

aabbccB   Mark a as x, Move Right, to find b

xabbccB   Ignore a, Move Right

xabbccB   Mark b as y, Move Right, to find c

XaYbccB   Ignore b, Move Right

XaYbccB   Mark c as Z, Move Left, to find x.

XaYbZcB   Ignore b, Move Left

XaYbZcB   Ignore Y, move Left

XaYbZCB   Ignore a, Move Left

XaYbZCB   Ignore x, Move Right

XaYbZCB   Mark a as x, Move Right, to find b

XXYbZCB   Ignore Y, Move Right

(31) (15)

XXYbZCB Mark b as Y, Move Right, to find C

(49)

XXYYZCB Ignore Z, Move Right

XXYYZCB Mark C as Z, Move Left, to find X

XXYYZZB Ignore Z, Move Left

XXYYZZB Ignore Y, Move Left

XXYYZZB Ignore Y, Move Left

XXYYZZB Ignore X, Move Right

XXYYZZB Ignore Y, Move Right

XXYYZZB Ignore Y, Move Right

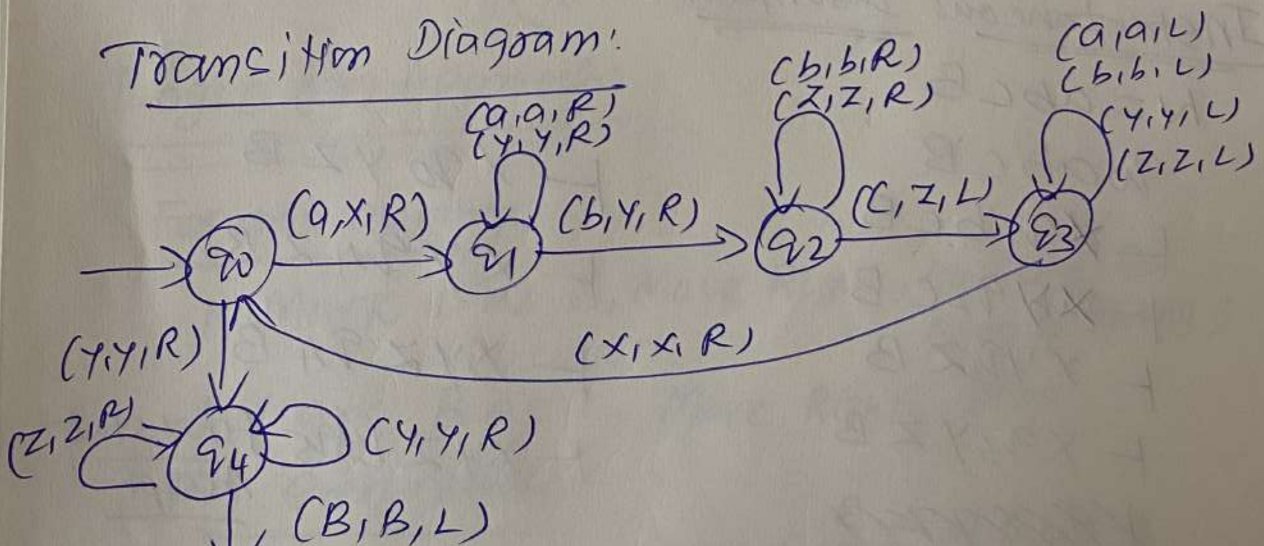
XXYYZZB Ignore Z, Move Right

XXYYZZB Ignore Z, Move Right

XXYYZZB Blank Symbol, Accept, Move Left (or) Right

1. Mark a as x, Move right, to find b
   (Ignore all a's, y's)
2. Mark b as y, Move right, to find c
   (Ignore all b's, z's)
3. Mark c as z, Move Left, to find x
   (Ignore z's, b's, y's, a's)
4. Repeat ①, ②, ③ untill all a, b, c are marked with x, y, z respectively. Then goto step 5.
5. Ignore all y's, z's Move right, to find B
6. Accept the String, Move Left (or) Right.

Transition Diagram:



Transition table:

	a	b	c	X	Y	Z	B
$\rightarrow q_0$	(q_1, X, R)	—	—	—	(q_4, Y, R)	—	—
q_1	(q_1, a, R)	(q_2, Y, R)	—	—	(q_2, Y, R)	—	—
q_2	—	(q_2, b, R)	(q_3, Z, L)	—	—	(q_2, Z, R)	—
q_3	(q_3, a, L)	(q_3, b, L)	—	(q_0, X, R)	(q_3, Y, L)	(q_3, Z, L)	—
q_4	—	—	—	—	(q_4, Y, R)	(q_4, Z, R)	(q_5, B, L)
q_5	—	—	—	—	—	—	—

Instantaneous Description:

$W = abcB$

$q_0 abcB$

$\vdash X q_1 bcB$

$\vdash XY q_2 cB$

$\vdash XY q_3 ZB$

$\vdash X q_3 YZB$

~~$\vdash q_0 XYZB$~~

~~$\vdash q_0 XYZB$~~

$\vdash X q_0 YZB$

$\vdash XY q_4 ZB$

$\vdash XYZ q_4 B$

$\vdash XYZB q_5$ final state Accept

③ Construct TM for $f(n) = n \bmod 2$.

Solution:

The given integer n is represented using unary string (number of 1's)

For example

$n = 3 \quad n = 111$

$n = 4 \quad n = 1111$

mod is used to find remainder after division.

if n is even number then n mod 2 is equal to $\boxed{0}$

if n is odd number then n mod 2 is equal to $\boxed{1}$

$$f(n) = \begin{cases} 0 & \text{if } n \text{ is even} \\ 1 & \text{if } n \text{ is odd.} \end{cases}$$

Logic (or) Approach:

For odd number

1. Mark 1 as B, Move Right, to find B (end of string)
2. Mark B as 1, Move Right, Stop.

For even number

1. Mark 1 as B, Move Right, to find B (end of string)
2. Mark B as 0, Move Right, Stop.

n=3 (3 mod 2 = 1)

111B Mark 1 as B, Move Right, to find B
↑

B11B Mark 1 as B, Move Right, to find B
↑

BB1B Mark 1 as B, Move Right, to find B
↑

BBBB Mark B as 1, Move Right, Stop
↑

BBB1 Accept
↑

n= 2 (2 mod 2 = 0)

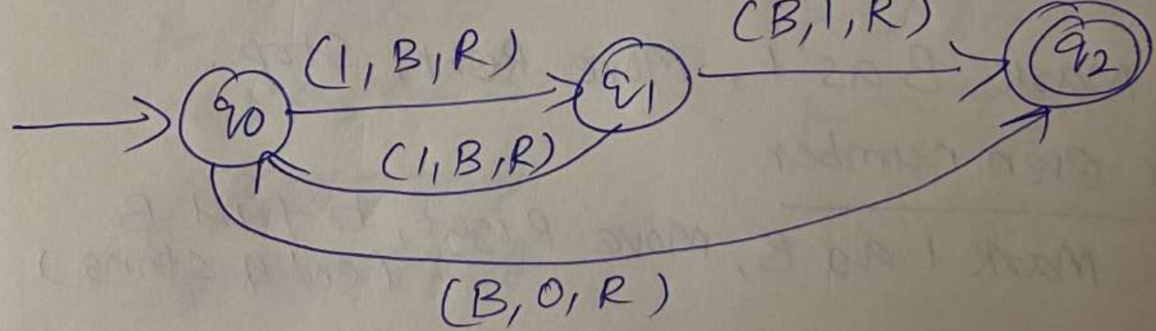
11B Mark 1 as B, Move right, to find B
↑

B1B Mark 1 as B, Move Right
↑

BBB Mark B as 0, Move Right, Stop
↑

BB0 Accept.
↑

Transition Diagram:



Transition table:

	1	B
→ q_0	(q_1, B, R)	$(q_2, 0, R)$
q_1	(q_0, B, R)	$(q_2, 1, R)$
q_2	—	—

Instantaneous Description:

$n = 1111 = 4$

$q_0 1111B$

$\vdash Bq_1 111B$

$\vdash BBq_0 11B$

$\vdash BBBq_1 1B$

$\vdash BBBBq_0 B$

$\vdash BBBB\underline{0}q_2$

$n = 111 = 3$

$q_0 111B$

$\vdash Bq_1 11B$

$\vdash BBq_0 1B$

$\vdash BBBq_1 B$

$\vdash BBB\underline{1}q_2$

(4) construct TM to work as copier.
To copy given information once on tape.

Solution:

Let consider the given data is Unary String format. (number of 1's)

Example: n=4
n=1111

Logic (or) Approach:

1. Mark 1 as X, Move right, to find B
2. Ignore B, Move Left
3. Mark X as 1, Move right, to find B
   (Ignore all 1's on the way)
4. Mark B as 1, Move left, to find X
   (Ignore All 1's on the way)

W = B11BBB

B11BBB Mark 1 as X, Move right, to find B
↑
BX1BBB Mark 1 as X, Move Right
↑
BXXBBB Ignore B, Move left
↑
BXXBBB Mark X as 1, Move Right to find B
↑

BX1BBB Mark B as 1, Move left, to find X

↑

BX11BB Ignore 1, Move left

↑

BX11BB Mark X as 1, Move right, to find B

↑

B111BB Ignore 1, Move Right

↑

B111BB Ignore 1, Move Right

↑

B111BB Mark B as 1, Move left, to find X.

↑

B1111B Ignore 1, Move Left

↑

B1111B Ignore 1, Move Left

↑

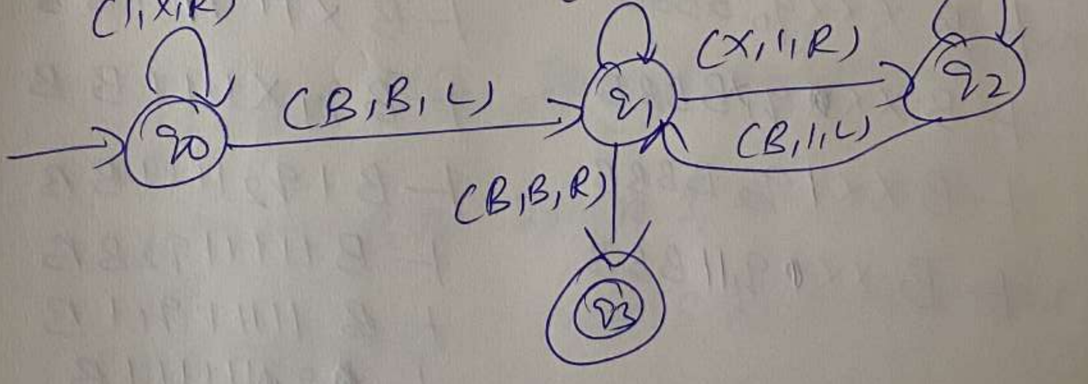
B1111B Ignore 1, Move Left

↑

B1111B Ignore B, Move Right, Accept.

↑

Transition Diagram:



Transition table:

	1	B	X
$\rightarrow q_0$	(q_0, X, R)	(q_1, B, L)	—
q_1	$(q_1, 1, L)$	(q_3, B, R)	$(q_2, 1, R)$
q_2	$(q_2, 1, R)$	$(q_1, 1, L)$	—
q_3	—	—	—

Instantaneous Description (ID)

$w = 3 = 111$

$B q_0 111 BBBB$

$\vdash BX q_0 11 BBBB$

$\vdash BXX q_0 1 BBBB$

$\vdash BXXX q_0 BBBB$

$\vdash BXX q_1 X BBBB$

$\vdash BXX1 q_2 BBBB$

$\vdash BXX q_1 11 BBB$

$\vdash BX q_1 X11 BBB$

$\vdash BX1 q_2 11 BBB$

$\vdash BX111 q_2 BBB$

$\vdash BX11 q_1 11 BB$

$\vdash B q_1 X1111 BB$

$\vdash B1 q_2 1111 BB$

$\vdash B11111 q_2 BB$

$\vdash B1111 q_1 1 B$

$\vdash q_1 B111111 B$

$\vdash q_3 111111 B$

5 Construct TM for addition of Unary numbers.

Solution:

$a = 2 = 11 \qquad b = 4 = 1111$

B	1	1	S	1	1	1	1	B	B

$\underbrace{1\ 1}_{a}\ \underbrace{S}_{\text{Separator.}}\ \underbrace{1\ 1\ 1\ 1}_{b}$

$a+b = 2+4 = 6 = 111111$

Logic (or) Approach:

1. Ignore all 1's, move right, to find S.
2. Mark S as 1, move right, to find B, (Ignore all 1's on the way,)
3. ~~Reach B,~~ Ignore B, move left
4. Mark 1 as B, Move right
5. Ignore B, Move Right, Accept.

11S1111BB Ignore 1, Move right, to find S
↑

11S1111BB Ignore 1, Move right
 ↑

11S1111BB Mark S as 1, Move right
  ↑

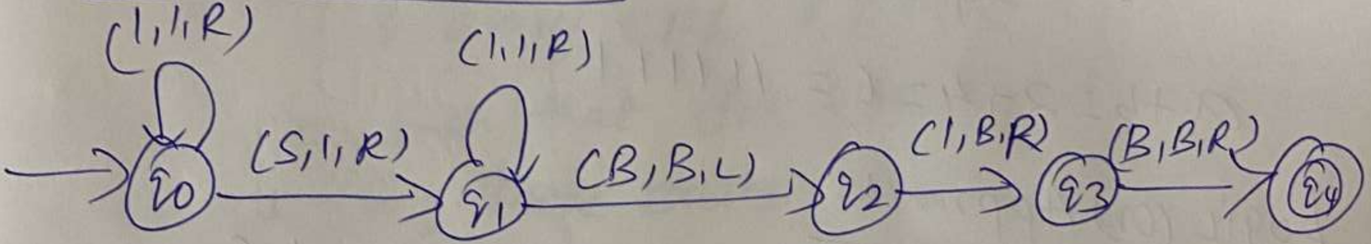
111 1111 B B Ignore all 1's untill B

111 1111 B B Ignore B, Move Left

1111111 B B Mark 1 as B, Move right

111111 B B B Ignore B, Accept, Move right.

Transition Diagram:



Transition table:

	1	S	B
q0	(q0,1,R)	(q1,1,R)	—
q1	(q1,1,R)	—	(q2,B,L)
q2	(q3,B,R)	—	—
q3	—	—	(q4,B,R)
q4	—	—	—

Pumping Lemma for CFL:

For any CFL L, it is possible to find two Substrings that can be pumped any number of times and still be in the same Language.

Statement:

For every CFL L, There is an integer n, such that for every String z in L of length $\geq n$, There exists $z = uvwxy$ such that

1. $|vwx| \leq n$
2. $|vx| > 0$
3. For all $i \geq 0$, uv^iwx^iy is in L.

Applications:

1. pumping lemma is used to prove the given Language is not CFL.

Example:

① Show that $L = \{a^n b^n c^n \mid n \geq 1\}$ is not CFL.

Solution:

Let us assume that L is CFL. So L satisfies pumping Lemma.

Let ~~z~~ = aaaabbbbcccc $= a^4b^4c^4$ (61)

<u>case 1:</u>

Let divide ~~z~~ into uvwxy. v, x contains two kinds of symbols.

$$\underbrace{aa}_{u}\ \underbrace{aabb}_{v}\ \underbrace{b}_{w}\ \underbrace{bc}_{x}\ \underbrace{ccc}_{y}$$

$v = aabb$, $x = bc$ both have two different types of symbols.

Let pump 2 into v, x. So

$$z = uv^2wx^2y$$

$$z = \underbrace{aa}_{u}\ \underbrace{aabbaabb}_{v^2}\ b\ \underbrace{bcbc}_{x}\ \underbrace{ccc}_{y}$$

but pattern of z is wrong. So $z \notin L$

<u>Case 2:</u>

v and x both contain one kind of symbol.

$$z = \underbrace{a}_{u}\underbrace{aa}_{v}\underbrace{ab}_{w}\underbrace{bb}_{x}\underbrace{bcccc}_{y}$$

Let pump 2 into v, x

$$z = uv^2wx^2y$$

$$= \underbrace{a}_{u}\ \underbrace{aaaaag}_{v^2}\ \underbrace{bb}_{w}\ \underbrace{bbbb}_{x^2}\ cccc$$

$$z = a^7b^6c^4 \notin L$$

So $L = \{a^nb^nc^n \mid n \geq 1\}$ is not a CFL.