7. PROPERTIES OF CONTEXT FREE LANGUAGES

CHAPTER - 7
PROPERTIES OF CONTEXT FREE LANGUAGES

7.1 SIMPLIFICATION OF CONTEXT FREE GRAMMAR

In a CFG G, it may not be necessary to use all the symbols in $V_N \cup \Sigma$, or all the productions in P for deriving sentences. That is, in a language L(G), we try to eliminate symbols and productions in G, which are not useful for derivation of sentences.

(i) Variables are eliminated if it does not derive any terminal string.

(ii) Elimination of null productions ($A \to \lambda$)

(iii) Elimination of unit productions.

Example 7.1

Consider G = ({S, A, B, C, E}, {a, b, c}, P, S)

where P = {$S \to AB$

$A \to a$

$B \to b$

$B \to C$

$E \to c|\lambda$}

Eliminate the useless symbols and productions to derive a reduced grammar G_1.

Solution :

From the given productions L(G) = {ab} Let G_1=({S,A,B}, {a,b}, P^1, S) where P^1 consists of $S \to AB$, $A \to a$, $B \to b$. L(G)=L(G_1).

The symbols C, E and c and the productions $B \to C$, $E \to c/\lambda$ are eliminated in G_1, because of the following constraints :

(i) C does not derive any terminal string.

(ii) E and c do not appear in any sentential form.

(iii) $E \to \lambda$ is a null production

(iv) $B \to C$ simply replaces B by C.

7.1.1 Construction of Reduced Grammar

Theorem

For every CFG G there exists a reduced grammar G^1 which is equivalent to G.

Proof

Let $G = (N, \Sigma, P, S)$ and $G^1 = (N^1, \Sigma, P^1, S)$

Step 1 (Construction of N^1)

A grammar G^1 is constructed, equivalent to the given grammar G so that every variable in G^1 derives some terminal string.

Step 2 (Construction of P^1):

Every symbol in N^1 should derive a sentential form to reach the terminal string are considered as productions of G^1.

(i.e.) $S \overset{*}{\Rightarrow} \alpha X \beta \overset{*}{\Rightarrow} w$ for some w in T* i.e. G^1 is reduced.

Theorem

If G is a CFG such that $L(G) = \phi$, we can find an equivalent grammar G^1, such that each variable in G^1 derives some terminal string.

Proof

Let $G=(N, T, S, P)$ and $G^1 = (N^1, T^1, S, P^1)$

(a) ***Construction of N^1***

We define $W_1 \subseteq N$ by recursion. $W_1=\{A \in N |$ there exists a production $A \to w$ where $w \in T^*\}$. (If $W_1 = \phi$, some variable will remain after the application of any production, and so $L(G) = \phi$).

$W_{i+1} = W_i \cup \{A \in N |$ there exists some production $A \to \alpha$ with $\alpha \in (T \cup W_i)^*\}$

By the definition of W_i, $W_i \subseteq W_{i+1}$ for all i. As N has only finite number of variables $W_k=W_{k+1}$ for some $k \leq |N|$. Therefore $W_k=W_{k+j}$ for $j \geq 1$. We define $N^1 = W_k$.

(b) ***Construction of P^1***

$P^1 = \{A \to \alpha | A, \alpha \in (N^1 \cup T)^*\}$

We can define $G^1=(N^1, T, S, P^1)$, S is in N^1. We can prove that every variable in N^1 defines some terminal string. So $S \notin N^1$, $L(G) = \phi$. Now we prove that G^1 is the required grammar.

(i) If each $A \in N^1$ then $A \underset{G^1}{\overset{*}{\Rightarrow}} w$ for sme $w \in T^*$; conversely, if $A \underset{G^1}{\overset{*}{\Rightarrow}} w$ then $A \in N^1$.

(ii) $L(G^1) = L(G)$

To prove (i) we note that $W_{k+1} = W_1 \cup W_2 \cup \ldots\ldots \cup W_k$. We prove by induction on i that for $i = 1, 2, \ldots.. k$, $A \in W_i$ implies $A \underset{G^1}{\overset{*}{\Rightarrow}} w$ for some $w \in T^*$. If $A \in W_1$, then $A \underset{G}{\Rightarrow} w$. So the production $A \rightarrow w$ is in P^1. Therefore, $A \underset{G^1}{\overset{*}{\Rightarrow}} w$. Thus there is basis for induction. Let us assume the result for i. Let $A \in W_{i+1}$. Then either $A \in w_i$, in which case, $A \underset{G^1}{\overset{*}{\Rightarrow}} w$ for some $w \in T^*$ by induction hypothesis. Or, there exists a production $A \rightarrow \alpha$ with $\alpha \in (T \cup W_i)$. By definition of P^1, $A \rightarrow \alpha$ is in P^1. We can write $\alpha = X_1 X_2 \ldots.. X_m$, where $X_j \in T \cup W_i$. If $X_j \in W_i$ by induction hypothesis, $X_j \underset{G^1}{\overset{*}{\Rightarrow}} W_j$ for some $W_j \in T^*$. So $A \underset{G^1}{\overset{*}{\Rightarrow}} W_1 W_2 \ldots\ldots W_m \in T^*$ (when X_j is a terminal, $W_j = X_j$). By induction the result is true for $i = 1, 2, \ldots\ldots k$.

The converse part can be proved in a similar way by induction on the number of steps in the derivation $A \underset{G}{\overset{*}{\Rightarrow}} w$. We see immediately that $L(G^1) \subseteq L(G)$ as $N^1 \subseteq N$ and $P^1 \subseteq P$.

To prove $L(G) \subseteq L(G^1)$, we need an auxillary result.

$A \overset{*}{\Rightarrow} w$ if $A \overset{*}{\Rightarrow} w$ for some $w \in T^*$ $\quad \rightarrow \quad 1$

We prove the above step by induction on the number of steps in the derivation $A \underset{G}{\overset{*}{\Rightarrow}} w$ If $A \underset{G}{\Rightarrow} w$, then $A \rightarrow w$ is in P and $A \in W_1 \subseteq N^1$. As $A \in N^1$ and $w \in T^*$, $A \rightarrow w$ is in P^1. So $A \underset{G^1}{\Rightarrow} w$, and there is a basis for induction. Assume $A \overset{*}{\Rightarrow} w$ derivation in atmost k steps. Let $A \underset{G}{\overset{k+1}{\Rightarrow}} w$. We can split this as

$A \underset{G}{\Rightarrow} X_1 X_2 \ldots.. X_m \;\; A \underset{G}{\overset{K}{\Rightarrow}} w_1 w_2 \ldots\ldots w_m$ such that $X_j \underset{G}{\overset{*}{\Rightarrow}} w_j$.

If $X_j \in T$ then $w_j = x_j$. $\quad \rightarrow \quad 2$

If $X_j \in N$ then by equation 1 above, $X_j \in N^1$. As $X_j \underset{G}{\overset{*}{\Rightarrow}} w_j$ is atmost k steps, $X_j \underset{G^1}{\overset{*}{\Rightarrow}} w_j$. Also, $X_1, X_2, \ldots\ldots X_m \in (T \cup N^1)^*$ implies $A \rightarrow X_1 X_2 \ldots X_m$ is in P^1. Thus $A \underset{G^1}{\Rightarrow} X_1 X_2 \ldots\ldots X_m \underset{G^1}{\overset{*}{\Rightarrow}} w_1 w_2 \ldots\ldots w_m$. Hence by induction, equation 1 is true for all derivations. In particular, $S \underset{G}{\Rightarrow} w$ implies $S \underset{G^1}{\overset{*}{\Rightarrow}} w$. This prove that $L(G) \subseteq L(G^1)$, and equation 2 is completely proved.

Theorem

For every CFG $G = (N, T, S, P)$, we can construct an equivalent grammar $G^1 = (N^1, T^1, S, P^1)$, such that every symbol in $N^1 \cup T^1$ appears in some sentential form (i.e.) for every X in $N^1 \cup T^1$ there exists α such that $S \underset{G^1}{\overset{*}{\Rightarrow}} \alpha$ and X is a symbol in the string α.

Proof

We construct $G^1 = (N^1, T^1, S, P^1)$ as follows:

(a) ***Construction of W_i for i ≥ 1***

(i) $W_1 = \{S\}$

(ii) $W_{i+1} = W_i \cup \{X \in N \cup T \mid$ there exists a production $A \to \alpha$ with $A \in W_i$ and α containing the symbol $X\}$

We may note that $W_i \subset N \cup T$ and $W_i \subseteq W_{i+1}$. As we have only finite number of elements in $N \cup T$, $W_k = W_{k+1}$ for some k. This means that $W_k = W_{k+j}$ for all $j \geq 0$.

(b) ***Construction of N^1, T^1 and P^1***

We define $N^1 = N \cap W_k$. $T^1 = T \cap W_k$ and $P^1 = \{A \to \alpha \mid A \in W_k\}$.

To prove that G^1 is the required grammar, we have to show that

(i) every symbol in $N^1 \cup T^1$ appears in some sentential form of G^1 and

(ii) conversely, $L(G^1) = L(G)$

To prove (i), consider $X \in N^1 \cup T^1 = W_k$. By construction $W_{k+1} = W_1 \cup \ldots\ldots \cup W_k$. We prove that $X \in W_i$, $i \leq k$, appears in some sentential form by induction on i. When $i = 1$, $X = S$ and $S \overset{*}{\underset{G^1}{\Rightarrow}} S$. Thus there is a basis for induction. Assume the result for all variables in W_i. Let $x \in W_{i+1}$. Then either $X \in W_i$, in which case, X appears in some sentential form by induction hypothesis. Otherwise, there exists a production $A \to \alpha$, where $A \in W_i$ and a contains the symbol X_i, A appears in some sentential form, say $\beta A \gamma$. Therefore

$S \overset{*}{\underset{G^1}{\Rightarrow}} \beta\ A\ \gamma \underset{G^1}{\Rightarrow} \beta\alpha\gamma.$

This means that $\beta\alpha\gamma$ is some sentential form and X is a symbol in $\beta\ \alpha\ \gamma$. Thus by induction the result is true for $X \in W_i$, $i \leq k$.

Conversely, if X appears in some sentential form, say $\beta X \gamma$, then $S \overset{l}{\underset{G}{\Rightarrow}} \beta X \gamma$. This implies $X \in W_1$. If $l \leq k$, then $W_l \subseteq W_k$. If $l > k$, then $W_l = W_k$. Hence X appears in $N^1 \cup T^1$. This proves (i).

To prove (ii), we note $L(G^1) \subseteq L(G)$ as $N^1 \subseteq N$, $T^1 \subseteq T$ and $P^1 \subseteq P$. Let w be in L(G) and $S = \alpha_1 \Rightarrow \alpha_2 \Rightarrow \alpha_3 \Rightarrow \ldots\ldots \Rightarrow \alpha_{n-1} \underset{G}{\Rightarrow} w$. We prove that every symbol in α_{i+1} is in W_{i+1} and $\alpha_i \underset{G^1}{\Rightarrow} \alpha_{i+1}$ by induction on i. $\alpha_1 = S \underset{G}{\Rightarrow} \alpha_2$ implies $S \to \alpha_2$ is a production in P. By construction, every symbol in α_2 is in W_2 and $S \to \alpha_2$ is in P^1, that is $S \underset{G}{\Rightarrow} \alpha_2$. Thus there is basis for induction. Let us assume the result for i. Consider $\alpha_{i+1} \underset{G}{\Rightarrow} \alpha_{i+2}$. This one step derivation can be written in the form.

$\beta_{i+1}\ A\gamma_{i+1} \Rightarrow \beta_{i+1}\ \alpha\gamma_{i+1}$

where $A \to \alpha$ is the production we are applying. By induction hypothesis $A \in W_{i+1}$. By consturction of W_{i+2} every symbol in α is in W_{i+2}. As all the symbols in β_{i+1} and γ_{i+1} are also in W_{i+1} by induction hypothesis, every symbol in $\beta_{i+1}\,\alpha\gamma_{i+1} = \alpha_{i+2}$ is in W_{i+2}. By construction of P^1, $A \to \alpha$ is in P^1. This means that $\alpha_{i+1} \underset{G}{\Rightarrow} \alpha_{i+2}$. Thus the induction procedure is complete. So $S = \alpha_1 \underset{G^1}{\Rightarrow} \alpha_2 \underset{G^1}{\Rightarrow} \alpha_3 \;......\Rightarrow \alpha_{n-1} W$. Therefore $w \in L(G^1)$. This proves (ii).

Example 7.2

Construct a reduced grammar equivalent to the grammar G = (N, T, S, P) where N = {S, A, C, D, E}, T = {a,b},

$P = \{S \to aAa,\ A \to Sb,\ A \to bCC,\ A \to DaA,$

$C \to abb,\ C \to DD,\ E \to aC,\ D \to aDA\}$

Solution :

Step 1 :

Each variable in G^1 derives some terminal string.

$W_1 = \{C\}$ as $C \to abb$ is the only production with a terminal string on the R.H.S.

$W_2 = \{C\} \cup \{E, A\}$, because $E \to aC$ and $A \to bCC$ are production with R.H.S. in $(T \cup \{C\})^*$

$W_3 = \{C, E, A\} \cup \{S\}$ as $S \to aAa$, where aAa is in $(TUW_2)^*$

$W_4 = W_3 \cup \phi$

$N^1 = W_3 = \{S, A, C, E\}$

$P^1 = \{S \to aAa,\ A \to Sb,\ A \to bCC,\ C \to abb,\ E \to aC\}$

(i.e.) $\{A \to \alpha \mid \alpha \in (N^1 \cup T)^*\}$

$G^1 = (N^1, \{a, b\}, S, P^1\}$

Step 2 :

In G^1 every symbol in $N^1 \cup T^1$ appear in some sentential form (i.e.) for every X in $N^1 \cup T^1$ there exists α such that $S \overset{*}{\underset{G^1}{\Rightarrow}} \alpha$ and X is a symbol in the string α.

$W_1 = \{S\}$, because $S \to aAa$

$W_2 = \{S\} \cup \{A, a\}$, because $A \to Sb$, $A \to bCC$

$W_3 = \{S, A, a\} \cup \{S, b, C\}$ because $C \to abb$.

$W_4 = W_3 \cup \{a, b\} = W_3$. That is with the existing set of W_3, two terminals (a, b) has to be added. But it already exist in W_3, therefore $W_4 = W_3$.

Hence $P^1 = \{A \to \alpha \mid A \in W_3\}$

$= \{S \to aAa$

$A \to Sb$

$A \to bCC$

$C \to abb\}$

$\therefore$ $G^1 = (\{S, A, C\}, \{a, b\}, P^{11}, S\}$ is the required reduced grammar of the given G.

7.1.2 Elimination of Null Productions

In a CFG, production of the form $A \to \lambda$ (null production), can be eliminated, where A is a variable. That is for the given grammar G, the G^1 can be derived as :

$L(G^1) = L(G) - \lambda$

Definition

A variable (A) in a context free grammar is nullable if $A \overset{*}{\Rightarrow} \lambda$.

Example 7.3

Consider a CFG, G with the following productions.

$S \to b\,S \mid b\,A \mid \lambda$

$A \to \lambda$

From the given grammar, the null productions $(A \to \lambda)$ can be eliminated because it derives to $S \to b$. The resultant productions are $S \to bS|\lambda|b$.

Therefore $L(G^1) = L(G) - \lambda = \{b^n \mid n \geq 0\}$, which is equivalent to the given grammar G.

Theorem

If G = (N, T, S, P) is a context free grammar, then G_1 can be derived with no null productions such that $L(G_1) = L(G) - \lambda$.

Proof

We construct $G_1 = (N, T, P^1, S)$ as follows.

Step 1 : *Construction of the set of nullable variables*

We find the nullable variable recursively.

(i) $W_1 = \{A \in N \mid A \to \lambda \text{ is in } P\}$

(ii) $W_{i+1} = W_i \cup \{A \in N \mid \text{there exists a production } A \to \alpha \text{ with } \alpha \in W_i^*\}$

By definition of W_i, $W_i \subseteq W_{i+1}$ for all i. As N is finite $W_{k+1} = W_k$ for some $k \leq |N|$. So $W_{k+j} = W_k$ for all j. Let $W = W_k$. W is the set of all nullable variables.

Step 2 : *Construction of P¹*

(a) Any production whose R.H.S. does not have any nullable variable is included in P^1.

(b) If $A \to X_1X_2....X_k$ is in P, the productions of the form

$A \to \alpha_1\alpha_2.... \alpha_k$ are included in P^1, where $\alpha_i = X_i$ if $X_i \notin W$. $\alpha_i = X_i$ or λ if $X_i \in W$ and $\alpha_1\ \alpha_2..... \alpha_k \neq \lambda$. Actually (b) gives several production in P^1. The productions are obtained either by not erasing any nullable variable on the R.H.S. of $A \to X_1\ X_2..... X_k$ or by erasing some or all nullable variables provided some symbol appears on the R.H.S. after erasing.

Let G_1=(N, T, P^1, S). G has no null productions. Now we prove that G_1 is the required grammar.

Step 3 : $L(G_1) = L(G) - \{\lambda\}$

To prove that $L(G_1) = L(G)-\lambda$, we prove an auxillary result given by the following relation : For all $A \in N$ and $w \in T^*$.

$A \overset{*}{\underset{G^1}{\Rightarrow}} w$ if and only if $A \overset{*}{\underset{G}{\Rightarrow}} w$ and $w \neq A \quad \to$ 1

we prove "if" part first. Let $A \overset{*}{\underset{G}{\Rightarrow}} w$ and $w \neq \lambda$. We prove that $A \overset{*}{\underset{G^1}{\Rightarrow}} w$ by induction on the number of steps in the derivation $A \overset{*}{\underset{G}{\Rightarrow}} w$. If $A \overset{*}{\underset{G}{\Rightarrow}} w$ and $w \neq \lambda$. $A \to w$ is a production in P^1 and so $A \underset{G^1}{\Rightarrow} w$. Thus there is basis for induction. Assume the result for derivations in atmost i steps. Let $A \overset{i+1}{\underset{G}{\Rightarrow}} w$ and $w \neq \lambda$. We can split the derivation as $A \underset{G}{\Rightarrow} X_1\ X_2..... X_k \overset{i}{\underset{G}{\Rightarrow}} w_1\ w_2....... w_k$, where $w = w_1w_2....w_k$ and $A_j \overset{*}{\underset{G}{\Rightarrow}} w_j$. As $w \neq \lambda$, not all w_j's are λ. If $w_j \neq \lambda$, then by induction hypothesis, $x_j \overset{*}{\underset{G_1}{\Rightarrow}} w_j$. If $w_j = \lambda$ then $x_j \in w$. So using the production $A \to A_1A_2...... A_k$ in P, we construct $A \to \alpha_1$, $\alpha_2........\alpha_k$ in P^1, where $\alpha_j = X_j$ if $w_j \neq \lambda$ and $\alpha_j = \lambda$ if $w_j = \lambda$ (i.e. $X_j \in w$). Therefore, $A \underset{G_1}{\Rightarrow} \alpha_1\ \alpha_2.....\alpha_k \underset{G_1}{\Rightarrow} w_1\ \alpha_2...... \alpha_k \underset{G_1}{\Rightarrow} .... \Rightarrow w_1w_2.......w_k = w$. By the principle of induction, 'if' part of equation 1 is proved.

We prove the only if part by induction on the number of steps in the derivation of $A \underset{G_1}{\Rightarrow} w$. If $A \underset{G_1}{\Rightarrow} w$, then $A \to w$ is in P_1. By construction of P^1, $A \to w$ is obtained from some production $A \to X_1\ X_2 ..... X_n$ in P by erasing some (or none of) nullable variables. Hence $A \underset{G}{\Rightarrow} X_1\ X_2...... X_n \overset{*}{\underset{G}{\Rightarrow}} w$. So there is a basis for induction Assume the result for derivation in atmost j steps.

Let $A \overset{j+1}{\underset{G_1}{\Rightarrow}} w$. This can be split as $A \Rightarrow X_1\ X_2 ....X_k \overset{j}{\underset{G_1}{\Rightarrow}} w_1w_2.....w_k$, where $X_i \overset{*}{\underset{G_1}{\Rightarrow}} w_i$. The first production $A \to X_1\ X_2....... X_k$ in P^1 is obtained from some production $A \to \alpha$ in P by erasing some (or none of) nullable variables in α. So $A \underset{G}{\Rightarrow} \alpha \overset{*}{\underset{G}{\Rightarrow}} X_1\ X_2..... X_k$. If $X_i \in T$ then $X_i \overset{0}{\underset{G}{\Rightarrow}} X_i =$

w_i. If $X_i \in N$ then by induction hypothesis, $X_i \overset{*}{\underset{G}{\Rightarrow}} w_i$. So we get $A \overset{*}{\underset{G}{\Rightarrow}} X_1X_2 \ldots\ldots X_k \overset{*}{\underset{G}{\Rightarrow}} w_1w_2\ldots\ldots w_k$. Hence by the principle of induction whenever $A \overset{*}{\underset{G_1}{\Rightarrow}} w$, we have $A \overset{*}{\underset{G}{\Rightarrow}} w$ and $w \neq \lambda$. Thus equation 1 is completely proved.

By applying equation 1 to S, we have $w \in L(G_1)$ if and only if $w \in L(G)$ and $w \neq \lambda$. This implies $L(G_1) = L(G) - \{\lambda\}$.

Example 7.4

Consider the grammar G whose productions are $S \rightarrow bS|AB$, $A \rightarrow \lambda$, $B \rightarrow \lambda$, $D \rightarrow a$. Construct a grammar G^1 without null productions generating $L(G) - \lambda$.

Solution :

Step 1: Construction of the set W of all nullable variables.

$W_1 = \{A \in N \mid A \rightarrow \lambda \text{ is a production in G}\}$

$= \{A, B\}$

$W_2 = \{A, B\} \cup \{S\}$, because $S \rightarrow AB$ is a production with $A, B \in W_1^* = \{S, A, B\}$

$W_3 = W_2 \cup \phi = W_2$

$\therefore W = W_2 = \{S, A, B\}$

Step 2 : Construction of P^1

(a) $D \rightarrow a$ is included in P^1

(b) $S \rightarrow bS$ gives rise to $S \rightarrow bS$ and $S \rightarrow b$

(c) $S \rightarrow AB$ gives rise to $S \rightarrow A$ and $S \rightarrow B$

We cannot erase both the nullable variables A and B in $S \rightarrow AB$, because if derives to $S \rightarrow \lambda$ in that case.

$\therefore G^1 = (\{S, A, B, D\}, \{a, b\}, S, P^1\})$, where P^1 consists of:

$D \rightarrow a$, $S \rightarrow bS$, $S \rightarrow AB$, $S \rightarrow b$, $S \rightarrow A$, $S \rightarrow B$

Corollary

There exists an algorithm to decide whether $\lambda \in L(G)$ for a given context free grammar G.

Proof

$\lambda \in L(G)$ if and only if $S \in W$ i.e. S is nullable.

(i) Contruct W.

(ii) Text whether $S \in W$.

Corollary

If G=(N, T, P, S) is a context free grammar, then the equivalent context free grammar G_1 = (N_1, T, P, S_1) without null production except $S_1 \rightarrow \lambda$, when λ is in L(G). If $S_1 \rightarrow \lambda$ is in P_1, S_1 does not appear on the R.H.S. of any production in P_1.

Proof

Case 1 :

If λ is not in L(G), the equivalent G_1can be obtained as L(G_1) = L(G)–λ

Case 2 :

If λ is in L(G)

★ Construct G^1 = (N, T, P^1, S) and prove L(G^1) = L(G)–λ

★ Define $G_1 = \{N \cup \{S\}, T, P_1, S_1\}$

where $P_1 = P^1 \cup \{S_1 \rightarrow S, S_1 \rightarrow \lambda\}$

S_1 does not appear on the R.H.S. of any production in P_1 and so G_1 is the required grammar with L(G_1) = L(G)

7.1.3 Elimination of Unit Productions

A unit production (or) a chain rule in CFG G is a production of the form A→B, where A and B are variables in G. Elimination of productions in A→B form are discussed in this section.

Example 7.5

(i) **Given :**

L(G) with the following productions :

$$P = \{S \rightarrow A,\ A \rightarrow B,\ B \rightarrow C,\ C \rightarrow a\}$$

At the end of substitution C→a as a terminal string.

$\therefore$ L(G) = {a}

(ii) If G_1 is S→a then L(G_1) = L(G)

Theorem

If G is a context free grammar, we can find a context free grammar G_1 which has no null productions such that L(G^1) = L(G). Let A be any variable in N.

Step 1 :

Construction of the set of variables derivable from A. Define $W_1(A)$ recursively as follows:

$W_0(A) = \{A\}$

$W_{i+1}(A) = W_i(A) \cup \{B \in N\} \mid C \to B$ is in P with $C \in W_i(A)\}$

By definition of $W_i(A)$, $W_i(A) \subseteq W_{i+1}(A)$. As N is finite, $W_{k+1}(A) = W_k(A)$ for some $k \leq |N|$. So $W_{k+j}(A)=W_k(A)$ for all $j \geq 0$. Let $W(A)=W_k(A)$. Then W(A) is the set of all variables derivable from A.

Step 2 :

Construction of A-productions in G_1. The A-productions in G_1 are either (a) the nonunit production in G^1 or (b) $A \to \alpha$ whenever $B \to \alpha$ is in G with $B \in W(A)$ and $\alpha \notin N$ Actually, (b) covers (a) as $A \in W(A)$. Now we define $G_1 = (N, T, S, P_1)$ where P_1 is constructed using step 2 for any $A \in N$.

Now we prove that G_1 is the required grammar.

Step 3 :

$L(G^1) = L(G)$. If $A \to \alpha$ is in P_1–P, then it is induced by $B \to \alpha$ in P with $B \in W(A)$, $\alpha \notin N$, $B \in W(A)$ implies $A \underset{G^1}{\overset{*}{\Rightarrow}} B$. Hence $A \underset{G^1}{\overset{*}{\Rightarrow}} B \underset{G^1}{\Rightarrow} \alpha$. So if $A \underset{G^1}{\Rightarrow} \alpha$, then $A \underset{G^1}{\overset{*}{\Rightarrow}} \alpha$. This proves $L(G_1) \subseteq L(G^1)$.

To prove the reverse inclusion, we start with a leftmost derivation $S \underset{G}{\Rightarrow} \alpha_1 \underset{G}{\Rightarrow} \alpha_2 \ldots\ldots \underset{G}{\Rightarrow} \alpha_n = w$ in G^1. Let i be the smallest index such that $\alpha_i \underset{G^1}{\Rightarrow} \alpha_{i+1}$ is obtained by a unit production and j be the smallest index greater than i such that $\alpha_j \underset{G^1}{\Rightarrow} \alpha_{i+1}$ is obtained by a nonunit production. So, $S \underset{G_1}{\Rightarrow} \alpha_i$, and $\alpha_i \underset{G^1}{\overset{*}{\Rightarrow}} \alpha_{j+1}$ can be written as

$\alpha_i = w_i\, A_i\, \beta_i \Rightarrow w_i A_{i+1}\, \beta_i \Rightarrow \ldots\ldots \Rightarrow w_i A_j \beta_i \Rightarrow w_i\, \gamma\, \beta_i = \alpha_{j+1}$

$A_j \in W(A_i)$ and $A_j \to \gamma$ is a nonunit production. Therefore $A_i \to \gamma$ is a production in P_1. Hence, $\alpha_i \underset{G_1}{\overset{*}{\Rightarrow}} \alpha_{j+1}$. Thus we have $S \underset{G_1}{\overset{*}{\Rightarrow}} \alpha_{j+1}$.

Repeating the argument whenever some unit production occurs in the remaing part of the derivation, we can prove that $S \underset{G_1}{\overset{*}{\Rightarrow}} \alpha_n = w$. This proves $L(G^1) \subseteq L(G)$.

Example 7.6

Let G be $S \to AB$, $A \to a$, $B \to C$, $B \to b$, $C \to D$, $D \to E$ and $E \to a$. Eliminate all unit productions and get an equivalent grammar.

Step 1 :

$w_0(S) = \{S\}$, $w_1(S) = w_0(S) \cup \phi$

$\therefore\ w(S) = \{S\}$

Step 2 :

Similarly w(A) = {A}

$w(E) = \{E\}$

Step 3 :

$w_0(B) = \{B\}$

$w_1(B) = \{B\} \cup \{C\} = \{B, C\}$

$w_2(B) = \{B, C\} \cup \{D\} = \{B, C, D\}$

$w_3(B) = \{B, C, D\} \cup \{E\} = \{B, C, D, E\}$

$w_4(B) = \{B, C, D, E\} \cup \phi = w_3(B)$

$\therefore w(B) = \{B, C, D, E\}$

Step 4 :

$w_0(C) = \{C\}, w_1(C) = \{C, D\}$

$w_2(C) = \{C, D, E\}$

$w_3(C) = \{C, D, E\} \cup \phi = w_2(C)$

$\therefore w(C) = \{C, D, E\}$

Step 5 :

$w_0(D) = \{D\}, w(D) = \{D, E\}$

$w_2(D) = \{D, E\} \cup \phi = w_1(D)$

$w(D) = \{D, E\}$

Step 6 :

The production in G_1 are $S \to AB$, $A \to a$, $E \to a$, $B \to b$,

$B \to a$, $C \to a$, $D \to a$

$\therefore$ G_1 has no unit productions.

7.2 NORMAL FORMS FOR CONTEXT FREE GRAMMARS (CFG)

In a CFG, the R.H.S. of a production in G satisfy certain conditions, then G is said to be in a "normal form". In this topic we discuss about two different types of normal forms. They are:

(i) Chomsky Normal Form (CNF)

(ii) Greibach Normal Form (GNF)

7.2.1 Chomsky Normal Form (CNF)

Definition

A context free grammar G is in CNF if every production is of the form $A \to a$ or $A \to BC$ and $S \to \lambda$ is in G if $\lambda \in L(G)$. When λ is in L(G) we assume that S does not appear on the R.H.S. of any production.

Example 7.7

Consider G with the productions of

$S \to AB$

$S \to \lambda$

$A \to a$

$B \to b$

The above example satisfies the conditions of CNF ($A \to BC$, $A \to a$, $S \to \lambda$). Therefore the given G is in CNF.

Note :

For a grammar in CNF, the derivation tree has the following property. Every node has atmost two descendants either two internal vertices ($A \to BC$) or a single leaf ($A \to a$)

Theorem

For every context free grammar, there is an equivalent grammar in Chomsky Normal Form (CNF)

Proof

Step 1 : ***Elimination of null productions.*** We then apply theorem to eliminate chain productions. Let the grammar thus obtained be G = (N, T, S, P).

Step 2 : ***Elimination of terminals on R.H.S***. We define $G_1 = (N^1, T, S, P_1)$ where P_1 and N^1 are constructed as follows:

(i) All the productions in P of the form $A \to a$ or $A \to BC$ are included in P_1. All the variables in N are included in N^1.

(ii) Consider $A \to X_1 X_2 \,.....\, X_n$ with some terminal on R.H.S. If X_i is a terminal, say a_i, add a new variable C_{a_i} to N^1 and $C_{a_i} \to a_i$ to P_1. In production $A \to X_1 X_2 \,......\, X_n$, every terminal on R.H.S. is repaced by the corresponding new variable and the variables on the R.H.S. are retained. The resulting production is added to P_1. Thus we get $G_1 = (N^1, T, P_1, S)$.

Step 3 : *Restricting the number of variables on R.H.S.* For any production in P_1, the R.H.S. consists of either a single terminal (or λ in $S \to \lambda$) or two or more variables. We define $G_2 = (N'', T, P_2, S)$ as follows:

(i) All productions in P_1 are added to P_2 if they are in the required form. All the variables in N^1 are added to N''.

(ii) Consider $A \to A_1 A_2 ..... A_m$, where $m \geq 3$. We introduce new productions

$A \to A_1 C_1,\ C_1 \to A_2 C_2,\ ......\ C_{m-2} \to A_{m-1} A_m,$

and new variables $C_1, C_2, ... C_{m-2}$. These are added to P'' and N'' respectively.

Thus we get G_2 in Chomsky Normal Form.

Step 4 :

To complete the proof we have to show that $L(G) = L(G_1) = L(G_2)$.

To show that $L(G) \subseteq L(G_1)$, we start with $w \in L(G)$. If $A \to X_1 X_2 ..... X_n$ is used in the derivation of w, the same effect can be achieved by using the corresponding production in P_1 and the productions involving the new variables. Hence

$A \overset{*}{\Rightarrow} X_1\ X_2 ..... X_n$. Thus $L(G) \subseteq L(G_1)$.

Let $w \in L(G_1)$. To show that $w \in L(G)$, it is enough to prove the following

$A \overset{*}{\underset{G}{\Rightarrow}} w$ if $A \in N$, $A \overset{*}{\underset{G}{\Rightarrow}} w$ 1

We prove 1 by induction on the number of steps in $A \overset{*}{\underset{G_1}{\Rightarrow}} w$.

If $A \overset{*}{\underset{G_1}{\Rightarrow}} w$, then $A \to w$ is a production in P_1. By construction of P_1, w is a single terminal. So $A \to w$ is in P i.e., $A \overset{*}{\underset{G}{\Rightarrow}} w$. This is basis for induction.

Let us assume 1 for derivations in atmost k steps. Let $A \overset{k+1}{\underset{G_1}{\Rightarrow}} w$. We can split this derivation as $A \underset{G_1}{\Rightarrow} A_1 A_2 ..... A_m \overset{k}{\underset{G_1}{\Rightarrow}} w_i . w_m = w$ such that $A_i \overset{*}{\underset{G_1}{\Rightarrow}} w_i$. Each A_i is either in N or a new variable, say Ca_i. When $A_i \in N$, $A_i \overset{*}{\underset{G_1}{\Rightarrow}} w_i$ is a derivation in atmost k steps, and so by induction hypothesis, $A_i \overset{*}{\underset{G_1}{\Rightarrow}} w_i$. Thus 1 is true for all derivations. Therefore $L(G) = L(G_1)$.

The effect of applying $A \to A_1 A_2 ..... A_m$ in a derivation for $w \in L(G_1)$ can be achieved by applying the production $A \to A_1 C_1, C_1 \to A_2 C_2, ... C_{m-2} \to A_{m-1} A_m$ in P_2. Hence it is easy to see that $L(G_1) \subseteq L(G_2)$.

To prove $L(G_2) \subseteq L(G_1)$, we can prove an auxillary result.

$A \overset{*}{\underset{G_1}{\Rightarrow}} w$ if $A \in N^1$, $A \overset{*}{\underset{G_2}{\Rightarrow}} w$ 2

Condition 2 can be proved by induction on the number of steps $A \overset{*}{\underset{G_2}{\Rightarrow}} w$. Applying 1 to S, we get $L(G_2) \subseteq L(G_1)$.

Thus $L(G) = L(G_1) = L(G_2)$

Example 7.8

Find a grammar in CNF equivalent to

$S \to aAbB$, $A \to aA|a$, $B \to bB|b$

Solution :

Step 1 :

Since there is no unit productions or null productions in given G, proceed to step 2.

Step 2 :

Let $G_1 = (N^1, \{a, b\}, S, P_1\}$ where P_1 and N^1 are constructed as follows :

(i) $A \to a$, $B \to b$ are added to P_1.

(ii) $S \to aAbB$, $A \to aA$, $B \to bB$ yield

$S \to C_1AC_2B$, $A \to C_1A$, $B \to C_2B$

where $C_1 \to a$, $C_2 \to b$

$\therefore N^1 = \{S, A, B, C_1, C_2\}$

Step 3 :

P_1 consists of

$S \to C_1AC_2B$

$A \to C_1A$

$B \to C_2B$

$C_1 \to a$

$C_2 \to b$

$A \to a$

$B \to b$

The first production $S \to C_1AC_2B$ is replaced as :

$S \to C_1D_1$

$D_1 \to AD_2$

$D_2 \to C_2B$

The remaining productions in P_1 are added to P_2 without any modification.

Therefore $G_2 = (\{S, A, B. C_1, C_2, D_1, D_2\}, \{a, b\}, P_2\ S)$ where P_2 consists of :

$S \rightarrow C_1D_1$

$D_1 \rightarrow AD_2$

$D_2 \rightarrow C_2B$

$A \rightarrow C_1A$

$B \rightarrow C_2B$

$C_1 \rightarrow a$

$C_2 \rightarrow b$

$A \rightarrow a$

$B \rightarrow b$

Hence G_2 is in CNF for the given grammar

7.2.2 Greibach Normal Form (GNF)

Definition

A context free grammar G is in GNF if every production is of the form $A \rightarrow a\alpha$ where $\alpha \in N^*$ and $a \in T$ (α may be λ) and $S \rightarrow \lambda$ is in G if $\lambda \in L(G)$, where S does not appear on the RHS of any production.

Example 7.9

Consider G with the productions of

$S \rightarrow aAB \mid \lambda$

$A \rightarrow bC$

$B \rightarrow b$

$C \rightarrow c$

The above example satisfies the condition of GNF ($A \rightarrow a\alpha$, $S \rightarrow \lambda$). Therefore the given G is in GNF.

Theorem

Every context free language L can be generated by a context free grammar G in GNF.

Proof

We prove the theorem when $\lambda \notin L$ and then extend the construction to L having λ.

Case 1: Construction of G when $\lambda \notin L$

Step 1 :

We eliminate null productions and then construct a grammar in G in CNF generating L. We rename the variables as $A_1, A_2, \ldots\ldots A_n$ with $S = A_1$. We write G as $(\{A_1, A_2, \ldots\ldots A_n\}, T, P, A_1)$.

Step 2 :

To get the productions in the form $A_i \to A_i\gamma$ or $A_i \to A_j\gamma$, where $j > i$, convert the A_i-productions (i=1, 2,..... n–1) to the form $A_i \to A_j\gamma$ such that $j > i$. Prove that such modifications is possible by induction on i.

Consider A_1-production. If we have some A_1-productions of the form $A_1 \to A_1\gamma$, then we can introduce a new variable to get rid of such productions. We get a new variable, say z_1, and A_1-production of the form $A_1 \to a$ or $A_1 \to A_j\gamma'$, where $j > 1$. Thus there is a basis for induction.

Assume we have modified A_1-productions, A_2-productions, ... A_i productions. Consider A_{i+1}-productions. Productions of the form $A_{i+1} \to \alpha\gamma$ required no modification. Consider the first symbol (this will be a variable) on the R.H.S. of the remaining A_{i+1}-productions. Let t be the smallest index among the indices of such symbols (variables). If $t > i + 1$, there is nothing to prove. Otherwise, apply induction hypothesis, to A_i-productions for $t \leq i$. So any A_t production is of the form $A_t \to A_j\gamma$, where $j > t$ or $A_t \to a\gamma^1$. Now we can apply lemma to delete a variable, A_{i+1}-production whose R.H.S. starts with A_t. The resulting A_{i+1}-productions are of the form $A_{i+1} \to A_j\gamma$, where $j > t$ (or $A_{i+1} \to \alpha\gamma^1$).

We repeat the above constructions by finding t for the new set of A_{i+1} productions. Ultimately, the A_{i+1}-productions are converted to the form $A_{i+1} \to A_i\gamma$, where $j \geq i + 1$ or $A_{i+1} \to a\gamma^1$. Productions of the form $A_{i+1} \to A_{i+1} \to A_{i+1}\gamma$ can be modified by inserting a new variable. Thus we have converted A_{i+1}-productions to the required form. By the principle of induction, the constructions can be carried out for i = 1, 2,.....n. Thus for i = 1, 2, ... n–1, any A_i-production is of the form $A_i \to A_j\gamma$, where $j > i$ or $A_i \to a\gamma^1$. Any A_n-production is of the form $A_n \to A_n\gamma$ or $A_n \to a\gamma^1$.

Step 3 :

Convert A_n-productions to the form $A_n \to \alpha\gamma$. Here productions of the form $A_n \to A_n\gamma$ are eliminated by inserting a new variable. The resulting A_n-productions are of the form $A_n \to \alpha\gamma$.

Step 4 :

Modify A_i-productions to the form $A_i \to a\gamma$ for i=1, 2,..... n–1. At the end of step 3, the A_n-productions are of the form $A_n \to \alpha\gamma$. The A_{n-1}-productions are of the form $A_{n-1} \to \alpha\gamma'$ or $A_{n-1} \to A_n\gamma$. We eliminate productions of the form $A_{n-1} \to A_n\gamma$.

The resulting A_{n-1}-productions are in the required form. We repeat the construction by considering A_{n-2}, A_{n-3}, ... A_1.

Step 5 :

Modify Z_i-productions. Every time we apply lemma to get new variable. (We take it as Z_i when we apply the lemma for A_i-productions). The Z_i productions are of the form $Z_i \rightarrow \alpha z_i$ or $Z \rightarrow \alpha$ (where α is obtained from $A_i \rightarrow A_i\alpha$) and hence of the form $Z_i \rightarrow \alpha\gamma$ or $Z_i \rightarrow A_k\gamma$ for some k. At the end of step 4, the R.H.S of any A_k-productions starts with a terminal. So we can apply lemma to eliminate a variable $Z_i \rightarrow A_k\gamma$. Thus at the end of step 5, we get an equivalent grammar G_1 in GNF.

It is easy to see that G_1 is in GNF. We start with G in CNF. In G any production is of the form $A \rightarrow a$ or $A \rightarrow AB$ or $A \rightarrow CD$. When we apply lemma to eliminate a variable and lemma to introduce a new variable in step 2, we get new productions of the form $A \rightarrow a$ or $A \rightarrow \beta$, where $\alpha \in N^*$ and $\beta \in N+$ and $a \in T$. In step 3 to 5, the productions are modified to the form $A \rightarrow a\alpha$ or $Z \rightarrow a^1\alpha^1$ where $a, a^1 \in T$ and $a, a^1 \in N^*$.

Case 2 : Construction of G when $\alpha \in L$

By the previous construction we get $G^1 = (N^1, \Sigma, P_1, S)$ in GNF such that $L(G^1) = L - \{\lambda\}$. Define a new grammar G_1 as

$G_1 = (N^1 \cup \{S\}, T, P_1 \cup \{S^1 \rightarrow S, S^1 \rightarrow \lambda\}, S^1)$

$S^1 \rightarrow S$ can be eliminated by using elimination of null productions. As S-production are in the required form. S^1-production are also in the required form so $L(G) = L(G_1)$ and G_1 is GNF.

Example 7.10

Construct a grammar in GNF equivalent to $P = \{S\rightarrow aSa, S\rightarrow bSb, S\rightarrow aa, S\rightarrow bb\}$

Solution :

We know that G=(N, T, S, P), where

$N = \{S\}, T = \{a, b\}$

$P = \{S\rightarrow aSa, S\rightarrow bSb, S\rightarrow aa, S\rightarrow bb\}$

Step 1 :

Let $G_1 = (N_1, T, S, P_1)$, where

$N_1 = \{S, A, B\}, T = \{a, b\}$ and

$P_1 = \{S\rightarrow ASA, S\rightarrow BSB, S\rightarrow AA, S\rightarrow BB, A\rightarrow a, B\rightarrow b\}$

Then $L(G_1) = L(G)$

Step 2 :

As per the rule of GNF ($A \to a\alpha$) the first symbol on R.H.S. to be a terminal. Therefore after renaming the non-terminal, G_1 is rewritten as :

$G_2 = (N_2, T, A_1, P_2)$ where

$N_2 = \{A_1, A_2, A_3\}$, $T=\{a, b\}$ and

$P_2 = \{A_1 \to A_2A_1A_2$

$A_1 \to A_3A_1A_3$

$A_1 \to A_2A_2$

$A_1 \to A_3A_3$

$A_2 \to a$

$A_3 \to b\}$

Let $G=(N, T, S, P)$ be a CFG. Let $A \to B\gamma$, Let B production be $B \to \beta_1|\beta_2.....|\beta_S$ then P' is defined as

$P' = (P-\{A \to B\gamma\} \cup \{A \to \beta_i\gamma\} \mid 1 \le i \le s)$

Then $G_1 =$ (N, T, S, P′) is a CFG in GNF form. This is called as Lemma 1.

Step 4 :

Replace the first symbol as terminal symbol.

$G_3 = (N_3, T, A_1, P_3)$, where

$N_3 = \{A_1, A_2, A_3\}$, $T=\{a, b\}$ and

$P_3 = \{A_1 \to aA_1A_2$

$A_1 \to aA_2$

$A_1 \to bA_1A_3$

$A_1 \to bA_3$

$A_2 \to a$

$A_3 \to b\ \}$

Hence G_3 is in GNF

7.3 CLOSURE PROPERTIES OF CONTEXT-FREE LANGUAGE

1. Regular Vs Context-free language

Theorem

Every regular language is context-free.

Proof

(i) Let L be regular.

(ii) Given a DFA (Finite Automata) for L, add a stack, but do not use the stack.

(iii) That is, change each DFA transition (p,a,q) to a DPA transition

$\delta(p,a,z) = \{(q,\lambda)\}$

(iv) The result is DPA whose language is L.

(v) Therefore, L is context-free

2. *Closure under Union*

Theorem

Let L_1 and L_2 be CFLs. Then $L_1 \cup L_2$ is also a CFL.

Proof

(i) Let L_1 have grammar (V_1, T_1, P_1, S_1) and let L_2 have grammar (V_2, T_2, P_2, S_2)

(ii) Then $L_1 \cup L_2$ has grammar (V_3, T_3, P_3, S_3)

where

- $V_3 = V_1 \cup V_2 \cup S_3$
- $T_3 = T_1 \cup T_2$
- S_3 = new start symbol
- $P_3 = P_1 \cup P_2 \cup \{S_3 \rightarrow S_1 \mid S_2\}$

(iii) Therefore $L_1 \cup L_2$ is CFL.

3. *Closure under Concatenation*

Theorem

Let L_1 and L_2 be CFLs. Then L_1L_2 is also CFL.

Proof

(i) Let L_1 have grammar (V_1, T_1, P_1, S_1) and L_2 have grammar (V_2, T_2, P_2, S_2)

(ii) Then L_1L_2 has grammar (V, T, P, S), where

$V = V_1 \cup V_1 \cup \{S\}$

$T = T_1 \cup T_2$

$P = P_1 \cup P_2 \cup \{S \rightarrow S_1S_2\}$

S = start symbol

(iii) Therefore L_1L_2 is a CFL.

4. Closure under Kleene star

Theorem

Let L be a CFL. Then L* is also a CFL.

Proof

(i) Let L have grammar (V_1, T_1, P_1, S_1)

(ii) Then L* has a grammar (V, T, P, S)

where

$V = V_1 \cup \{S\}$

$T = T_1$

$P = P_1 \cup \{S \to e, S \to SS_1\}$

S = start symbol

(iii) Therefore, L is a CFL.

5. Intersection of a CFL and RE.

Theorem

Intersection of a CFL and a Regular Language is a CFL.

Proof

(i) *Given :* Let $L_1 = L(M_1)$ for some PDA,

$M_1 = (Q_1, \Sigma_1, \Gamma_1, \delta_1, S_1, F_1)$

and $L_2 = L(M_2)$ for some DFA

$M_2 = (Q_2, \Sigma_2, \delta_2, S_2, F_2)$

(ii) *Need to show :*

$L_1 \cap L_2 = L(M)$ for some PDA, M

where $M = (Q, \Sigma, \Gamma, \delta, S, F)$

(iii) *Idea :*

Construct a PDA M that operates in the same way as M_1 except that it also keeps track of the change in state in M_2 caused by reading the same input.

(iv) *Construction :*

$Q = Q_1 \text{ x } Q_2$

$\Sigma = \Sigma_1 \cup \Sigma_2$

$\Gamma = \Gamma_1$

$S = \{S_1, S_2\}$

$F = F_1 \times F_2$

- for each transition $\{(q_1, a, \beta), (p_1, \gamma)\} \in \delta_1$ and for each state $q_2 \in Q_2$ add to δ the transition

$(((q_1,q_2), a, \beta), ((p_1,\delta(q_2,a)),\gamma))$

- for each transition $\{(q_1, \lambda, \beta), (p_1,\gamma)\} \in \delta_1$ and for each state $q_2 \in Q_2$ add to δ the transition

$(((q_1,q_2), \lambda, \beta), ((p_1, q_2), \gamma))$

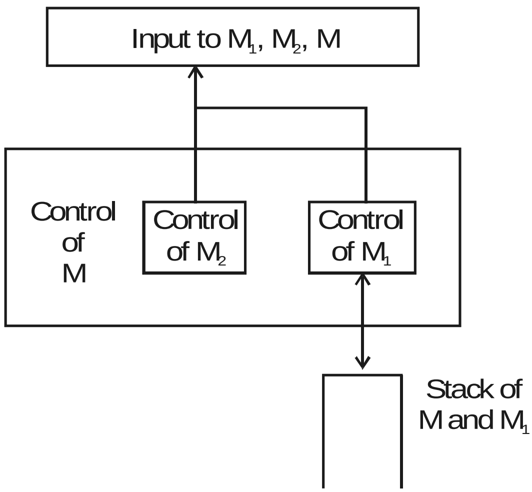


Fig. Running of DFA and PDA in parallel

6. *Complementation and Intersection*

(i) The complement of a context-free language is not necessarily context-free.

(ii) The intersection of two context-free language is not necessarily context-free.

7. *Property of CFL (Fanout & Height)*

Let G = (V, T, P, S) be a CFG.

- The fanout of G, ϕ (G) is the largest number of symbols on the RHS of any rule in R.
- The height of a parse tree is defined as the length of the longest path from the root to some leaf.

Example 7.11

$G = S \rightarrow e, S \rightarrow SS, S \rightarrow \{S\}$

Height = 4

ϕ (G) = 3

Example 7.12

L = the set of all strings of *a*'s and *b*'s with e*q*ual number of *a*'s and *b*'s but containing no substring *abaa* (or) *babb*. Is *L* context free?

Solution :

(i) Let $L_1 = \{w \in \{a,b\}^* : w$ has e*q*ual number of *a*'s and *b*'s$\}$

Let $L_2 = \{w \in \{a,b\}^* : w$ contains no string *abaa* or *babb*$\}$

(ii) Then $L = L_1 \cap L_2$

(iii) Since $L_2 = L(a \cup b)^* - L((a \cup b)^* (abaa \cup babb) (a \cup b)^*)$. Thus L_2 is regular

(iv) Since L_1 is context free and L_2 is regular, thus L is context-free

7.4 PUMPING LEMMA FOR CONTEXT FREE LANGUAGES

Concept

(i) The pumping lemma for CFLs will allow us to show that some languages are not context-free.

(ii) If a CFL, contains a word *w* with a sufficiently long derivation $S \overset{*}{\Rightarrow} w$, then some nonterminal must appear more than once.

(iii) That is, we have

$S \Rightarrow^* uAz \Rightarrow^* uvAyz \Rightarrow^* uvxyz$

(iv) Thus $A \Rightarrow^* vAy$ and $A \Rightarrow^* x$

(v) We may repeat the derivation $A \Rightarrow^* vAy$ as many times as we like (including zero times), producing $uv^n xy^n z$, for any $n \geq 0$.

Theorem

Let L be an infinite context-free Language. Then there exists some positive integer m such that any $w \in L$ with $|w| \geq m$ can be decomposed as

$$w = uvxyz \qquad .....(1)$$

with

$$|vxy| \leq m \qquad .....(2)$$

and

$$|vy| \geq 1 \qquad .....(3)$$

such that

$$uv^ixy^iz \in L \qquad .....(4)$$

for all $i = 0, 1, ...$ This is known as pumping lemma for context free languages.

Proof

Consider the context free grammar G without unit productions (or) λ - productions. L $-\{\lambda\}$ is the language which is generated by G. The length of the string on the right hand side of any production is bounded. Since L is infinite, there exists arbitrarily long derivations and corresponding derivation trees of arbitrary height.

Consider a high derivation tree from root to leaf. Since the number of variables in G is finite, there must be some variables that repeats on this path.

Consider the derivation

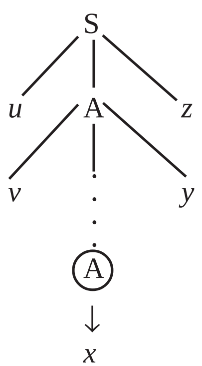
$S \Rightarrow^* uAz \Rightarrow^* uvAyz \Rightarrow^* uvxyz$

where u, v, x, y, z are all strings of terminals.

$A \Rightarrow^* uAy$ and $A \Rightarrow^* x$.

If this derivations are repeated, we can generate all strings uv^ixy^iz, $i=0,1.....$

Derivation tree



We can assume that no variables repeats (repeating variable-A). The length of the strings v, x, and y depends only on productions of the grammar and can be bounded independently of w (condition (2) holds). Since there is no λ-productions, v and y cannot be empty string (condition (3) holds).

Example 7.13

Show that the language L = $\{a^n b^n c^n : n \geq 0\}$ is not context-free. **Apr/May 2004**

Solution :

(i) Assume it is context-free.

(ii) Let m be the positive integer.

(iii) Let $w = a^n b^n c^n \in L$ & $w \geq m$.

(iv) Then $w = uvxyz$ where

$|v|>0$ (or) $|y| > 0$

and $|vxy| \leq m$

(v) The vxy contains at most two different symbols. Suppose it contains at most a's and b's (no c's). Then either v contain atleast one a (or) y contains at least one b.

(vi) Say v contains i a's and y contains j b's with $i, j>0$. Then uv^2xy^2z contain atleast $n+i$ a's and atleast $n+j$ b's, which is greater than m. But uv^2xy^2z will contain n c's

(vii) Thus $uv^2xy^2z \notin$ L, which is a contradiction. Hence L is not context free.

7.5 SOLVED PROBLEMS

1. Consider the Grammar G with the following productions:

$S \rightarrow Aa$

$S \rightarrow B$

$B \rightarrow A$

$B \rightarrow bb$

$A \rightarrow a$

$A \rightarrow bc$

$A \rightarrow B$

Eliminate all unit productions and get an equivalent grammar G_1.

Solution :

Step 1 :

The unit productions of the form $S \to B$, $B \to A$, $A \to B$, $B \to A$ derives $S \overset{*}{\Rightarrow} A$.

Step 2 :

The productions in G_1 are :

$S \to Aa$, $S \to a|bc|bb$

$B \to bb$, $B \to a|bc$

$A \to a|bc$, $A \to bb$

$\therefore$ G_1 has no unit production.

2. Consider the grammar G with the following productions :

$S \to A$

$S \to B$

$A \to B|C|aB|b$

$B \to C$

$C \to B|Aa$

Eliminate all unit productions and find an equivalent grammar G_1.

Solution :

Step 1 :

The unit production of the form $S \to A$, $A \to B$, $B \to C$, $C \to B$, $B \to C$ (i.e.) $S \overset{*}{\Rightarrow} C$, and $S \to B$, $B \to C$, $C \to B$, $B \to C$ (i.e.) $S \overset{*}{\Rightarrow} C$ can be derived.

Step 2 :

The productions in G_1 are :

$S \to aB|b|Aa$

$A \to Aa|aB|b$

$B \to Aa$

$C \to Aa$

$\therefore$ G_1 has no unit productions.

3. Consider the grammar $S \to aS|aSbS|\varepsilon$. Find an unambiguous grammar to generate same

Solution :

The idea is to introduce another nonterminal T that cannot generate an unbalanced *a*.

That strategy corresponds to the usual rule in programming languages that an "else" is associated with the closest previous, unmatched "then".

Here, we force *a b* to match the previous unmatched *a*.

The grammar that is unambiguous

$S \rightarrow aS|aTbS|\varepsilon$

$T \rightarrow aTbT|\varepsilon$

generates same language

4. Find a grammar without useless symbols equivalent to

$S \rightarrow AB \mid CA$

$A \rightarrow a$

$B \rightarrow BC \mid AB$

$C \rightarrow aB \mid b$

Solution :

A and C are clearly generating, since they have productions with terminal strings.

S is generating since $S \rightarrow CA$, whose body consists of only symbols that are generating.

B is not generating any terminal string. Eliminating B,

$S \rightarrow CA$

$A \rightarrow a$

$C \rightarrow b$

Since S, A, and C are each reachable from S, all the remaining symbols are useful.

5. Consider with the grammar

$S \rightarrow ASB \mid \varepsilon$

$A \rightarrow aAS \mid a$

$B \rightarrow SbS \mid A \mid bb$

a) Are there any useless symbols? Eliminate if so.

Solution :

Observe that A and B each derive terminal strings, and therefore so does S. Thus, there are no useless symbols.

b) Eliminate ε-productions

Solution :

Only S is nullable, so we must choose, at each point where S occurs in a body, to eliminate it or not. Since there is no body that consists only of S's, we do not have to invoke the rule about not eliminating an entire body.

The resulting grammar has no ε-productions

$S \rightarrow ASB \mid AB$

$A \rightarrow aAS \mid aA \mid a$

$B \rightarrow SbS \mid bS \mid Sb \mid b \mid A \mid bb$

c) Eliminate unit productions

Solution :

The only unit production is $B \rightarrow A$. It suffices to replace this body A by the bodies of all the A-productions.

The result :

$S \rightarrow ASB \mid AB$

$A \rightarrow aAS \mid aA \mid a$

$B \rightarrow SbS \mid bS \mid Sb \mid b \mid aAS \mid aA \mid a \mid bb$

6. Construct a grammar in Greibach Normal Form (GNF) equivalent to the grammar. $S \rightarrow AA|a$, $A \rightarrow SS|b$ **(Nov/Dec 2003)**

Solution :

Step 1 :

The given grammar has no null productions and is on CNF. Therefore the variables S and A are renamed as A_1, A_2. Hence the productions are:

$A_1 \rightarrow A_2A_2 \mid a$, $A_2 \rightarrow A_1A_1 \mid b$

Step 2 :

Derive the productions of the form $A_i \rightarrow a\gamma$ or $A_i \rightarrow A_{jv}$ where $j>i$ in P. Apply $A_1 \rightarrow A_2A_2|a$ in $A_2 \rightarrow A_1A_1$, derives $A_2 \rightarrow A_2A_2A_1$, $A_2 \rightarrow aA_1$

Therefore A_2 productions are:

$A_2 \rightarrow A_2A_2A_1$, $A_2 \rightarrow aA_1$, $A_2 \rightarrow b$

Step 3 :

To derive the productions of the form $A_n \rightarrow a\gamma$, from $A_n \rightarrow A_n\gamma$. Let z_2 be the new variable to apply in $A_2 \rightarrow A_2A_2A_1$ as per lemma 2 of GNF.

Therefore the resulting productions are :

$A_2 \to aA_1$, $A_2 \to b$

$A_2 \to aA_1z_2$, $A_2 \to bz_2$

$z_2 \to A_2A_1$, $z_2 \to A_2A_1z_2$.

Lemma 2:

Let G=(N, T, S, P) be a CFG. Let the set of A productions be $A \to A\alpha_1|A\alpha_2|\ldots\ldots|A\alpha_r|\beta_1|\beta_2|\ldots\ldots|\beta_S$. Let B be a new variable then $G_1 = (N \cup \{B\}, T, P', S)$ where P′ is defined as

(a) the set of A productions in P′ are

$A \to \beta_1|\beta_2|\ldots\ldots \beta_S$

$A \to \beta_1B|\beta_2B|\ldots\ldots|\beta_SB$

(b) the set of B productions in P′ are

$B \to \alpha_1|\alpha_2\ldots\ldots|\alpha_r$

$B \to \alpha_1B|\alpha_2B|\ldots\ldots|\alpha_rB$

(c) The productions for other variables are as in P, then G_1 is a CFG in GNF equivalent to G.

Step 4 :

Apply the same steps (2, 3) for A_1 also.

(a) A_2 productions are :

$A_2 \to aA_1|b|aA_1z_2|bz_2$

(b) A_1 productions are :

$A_1 \to a$, retained as it is.

$A_1 \to A_2A_2$ is modified as :

$A_1 \to aA_1A_2 \mid bA_2$

$A_1 \to aA_1z_2A_2 \mid bz_2A_2$

∴ The total A_1 productions are :

$A_1 \to a \mid aA_1A_2 \mid bA_2 \mid aA_1z_2A_2 \mid bz_2A_2$

Step 5 :

Modification of new variable production to the form of $z_i \to a\gamma$. Therefore the z_2 productions ($z_2 \to A_2A_1$, $z_2 \to A_2A_1z_2$) are modified as :

$z_2 \to aA_1A_1 \mid bA_1 \mid aA_1z_2A_1 \mid bz_2A_1$

$z_2 \to aA_1A_1z_2 \mid bA_1z_2 \mid aA_1z_2A_1z_2 \mid bz_2A_1z_2$

Hence the equivalent grammar is :

$G^1 = (\{A_1,A_2,z_2\}, \{a,b\},P_1,A_1)$

where P_1 consists of :

$A_1 \to a|aA_1A_2|bA_2|aA_1z_2A_2|bz_2A_2$

$A_2 \to aA_1|b|aA_1z_2|bz_2$

$z_2 \to aA_1A_1 \mid bA_1 \mid aA_1z_2A_1 \mid bz_2A_1$

$z_2 \to aA_1A_1z_2 \mid bA_1z_2 \mid aA_1z_2A_1z_2 \mid bz_2A_1z_2$

7. Reduce the following grammar G to CNF. G is $S \to aAD$, $A \to aB|bAB$, $B \to b$, $D \to d$

(Apr/May 2004)

Solution :

Step 1 :

As there are no null productions or unit productions, proceed step 2.

Step 2 :

Let $G_1 = (V^1, \{a, b, d\}, P_1, S)$, where P_1, V^1 are constructed as follows:

(i) $B \to b$, $D \to d$ are included in P_1

(ii) $S \to aAD$ gives rise to $S \to C_aAD$ and $C_a \to a$.

$A \to aB$ gives rise to $A \to C_aB$

$A \to bAB$ gives rise to $A \to C_bAB$ and $C_b \to b$

$\therefore V^1 = \{S,A,B,D,C_a,C_b\}$

$\therefore P_1 = \{S \to C_aAD,$

$A \to C_aB \mid C_bAB,$

$B \to b,$

$D \to d,$

$C_a \to a,$

$C_b \to b\ \}$

Step 3 :

Let $G_2 = (V^1\ \{a, b, d\}, P_2, S)$, where P_2, V^1 are constructed as follows :

$A \to C_aB$, $B \to b$, $D \to d$, $C_a \to a$, $C_b \to b$ are added to P_2

$S \to C_aAD$ is replaced by $S \to C_aC_1$ and $C_1 \to AD$

$A \to C_bAB$ is replaced by $A \to C_bC_2$ and $C_2 \to AB$

$\therefore V^1 = \{S, A, B, D, C_a, C_b, C_1, C_2\}$

$\therefore P_2 = \{S \to C_aC_1$

$A \to C_aB \mid C_bC_2$

$C_1 \to AD$

$C_2 \to AB$

$B \to b,$

$D \to d,$

$C_a \to a,$

$C_b \to b \}$

Hence G_2 is in CNF and it is equivalent to G.

8. Convert the given grammar to its equivalent GNF.

$G = (\{A_1, A_2, A_3\}, \{a, b\}, P, A_1)$

where P consists of the following :

$A_1 \to A_2A_3$

$A_2 \to A_3A_1 \mid b$

$A_3 \to A_1A_2 \mid a$ **(Apr/May 2004)**

Solution :

Step 1 :

The given grammar has no null productions or unit production. Therefore proceed to step 2 with the same.

Step 2 :

Since the righthand side of the productions for A_1, A_2 start with terminals or higher-numbered variables, perform the substitution from A_3 production (i.e.) $A_3 \to A_2A_3A_2$ (Because $A_1 \to A_2A_3$)

The resulting productions are:

$A_1 \to A_2A_3$

$A_2 \to A_3A_1 \mid b$

$A_3 \to A_2A_3A_2 \mid b$

Step 3 :

Since the right side of the production $A_3 \to A_2A_3A_2$ begins with a lower numbered variable, substitute A_2 in A_3. The resulting productions are :

$A_1 \rightarrow A_2A_3$

$A_2 \rightarrow A_3A_1 \mid b$

$A_3 \rightarrow A_3A_1A_3A_2 \mid bA_3A_2 \mid a$

Step 4 :

A new symbol B_3 is introduced in A_3 to derive the first symbol as a terminal string as per lemma 2 of GNF. The resulting productions are :

$A_1 \rightarrow A_2A_3$

$A_2 \rightarrow A_3A_1 \mid b$

$A_3 \rightarrow bA_3A_2B_3 \mid aB_3 \mid bA_3A_2 \mid a$

$B_3 \rightarrow A_1A_3A_2 \mid A_1A_3A_2B_3$

Step 5 :

Now all the productions with A_3 on the left starts with terminals on the right. Therefore apply the A_3 production in the A_2 and A_1 to derive the same.

$A_3 \rightarrow bA_3A_2A_3$	$A_3 \rightarrow bA_3A_2$
$A_3 \rightarrow aB_3$	$A_3 \rightarrow a$
$A_2 \rightarrow bA_3A_2B_3A_1$	$A_2 \rightarrow bA_3A_2A_1$
$A_2 \rightarrow aB_3A_1$	$A_2 \rightarrow aA_1$
$A_2 \rightarrow b$	
$A_1 \rightarrow bA_3A_2B_3A_1A_3$	$A_1 \rightarrow bA_3A_2A_1A_3$
$A_1 \rightarrow aB_3A_1A_3$	$A_1 \rightarrow aA_1A_3$
$A_1 \rightarrow bA_3$	
$B_3 \rightarrow A_1A_3A_2$	$B_3 \rightarrow A_1A_3A_2B_3$

Step 6 :

Now the two B_3 productions are converted into the required GNF form. (i.e.) $B_3 \rightarrow A_1A_3A_2$, in which substitution of A_1 is done using five productions of A_1.

$B_3 \rightarrow bA_3A_2B_3A_1A_3A_3A_2$

$B_3 \rightarrow aB_3A_1A_3A_3A_2$

$B_3 \rightarrow bA_3A_3A_2$

$B_3 \rightarrow bA_3A_2A_1A_3A_3A_2$

$B_3 \rightarrow aA_1A_3A_3A_2$

Similarly $B_3 \rightarrow A_1A_3A_2B_3$ is also done.

Step 7 :

The final set of productions are :

$A_3 \to bA_3A_2B_3$ $\quad$ $A_3 \to bA_3A_2$

$A_3 \to aB_3$ $\quad$ $A_3 \to a$

$A_2 \to bA_3A_2B_3A_1$ $\quad$ $A_2 \to bA_3A_2A_1$

$A_2 \to aB_3A_1$ $\quad$ $A_2 \to aA_1$

$A_2 \to b$

$A_1 \to bA_3A_2B_3A_1A_3$ $\quad$ $A_1 \to bA_3A_2A_1A_3$

$A_1 \to aB_3A_1A_3$ $\quad$ $A_1 \to aA_1A_3$

$A_1 \to bA_3$

$B_3 \to bA_3A_2B_3A_1A_3A_3A_2B_3$

$B_3 \to bA_3A_2B_3A_1A_3A_3A_2$

$B_3 \to aB_3A_1A_3A_3A_2B_3$

$B_3 \to aB_3A_1A_3A_3A_2$

$B_3 \to bA_3A_3A_2B_3$

$B_3 \to bA_3A_3A_2$

$B_3 \to bA_3A_2A_1A_3A_3A_2B_3$

$B_3 \to bA_3A_2A_1A_3A_3A_2$

$B_3 \to aA_1A_3A_3A_2B_3$

$B_3 \to aA_1A_3A_3A_2$

9. Show that the language $\{a^i b^j c^k \mid i<j<k\}$ is not context-free.

Solution :

Let n be the pumping-lemma constant and consider string

$z = a^n b^{(n+1)} c^{(n+2)}$

Write $z = uvwxy$, where v and x, may be "pumped," and $|vwx| \leq n$.

If vwx does not have c's, then uv^3wx^3y has at least $n+2$ a's or b's, and thus could not be in the language.

If vwx has a c, then it could not have an a, because its length is limited to n.

Thus, uwy has n a's, but no more than $2n+2$ b's and c's in total.

Thus, it is not possible that uwy has more b's than a's and also has more c's than b's.

We conclude that uwy is not in the language, and now have a contradiction no matter how z is broken into $uvwxy$.

10. Show that the language is not context free $\{0^i 1^j \mid j=i^2\}$

Solution :

Let n be the pumping-lemma constant and consider $z=0^n 1^{n^2}$.

We break $z = uvwxy$ according to the pumping lemma.

If vwx consists only of 0's, then uwy has $n^2$1's and fewer than n 0's; it is not in the language.

If vwx has only 1's, then we derive a contradiction.

Similarly, if either v or x has both 0's and 1's, then uv^2wx^2y is not in 0*1*, and thus could not be in the language.

Finally, consider the case where v consists of 0's only, say k 0's, and x consists of m 1's only, where k and m are both positive.

Then for all 1, $uv^{(i+1)}wx^{(i+1)}y$ consists of $(n+ik)^2 = n^2 + 2ink + i^2k^2$ 0's and n^2+1m 1's.

If the number of 1's is always to be the square of the number of 0's, we must have, for some positive k and m: $2ink + i^2k+=im$.

But the left side grows quadratically in i, while the right side grows linearly, and so this equality for all i is impossible. We conclude that for at least some i, $uv^{(i+1)} wx^{(i+1)}y$ is not in the language and have thus derived a contradiction in all cases.

11. If L = $\{c^i b^j c^i d^j \mid i \geq 1$ and $j \geq 1)$. Is it a CFL? **(Nov/Dec 2003)**

Solution :

Let L is a CFL, and n be the constant. Consider the string $z = a^n b^n c^n d^n$. Let z = $uvwxy$ satisfy the conditions of the pumping lemma. Then as $|vwx| \leq n$, vx can contain at most two different symbols. Furthermore, if vx contains two different symbols, they must be consecutive, for example, a and b. If vx has only a's, then uwy has fewer a's then c's and is not in L, a contradiction. We proceed similarly if vx consists of only b's, only c's, or only d's. Now suppose vx has a's and b's. Then vwy still has fewer a's than c's. A similar contradiction occurs if vx consists of b's and c's or c's and d's. Since these are the only possibilities, we conclude that L is not context free.

12. Show that $a^n\, b^n\, c^n$ is not context free language i.e., show that the set of strings of a's b's and c's with an equal number of each is not context free.

Solution :

The given language $L = \{a^n\, b^n\, c^n\}$

Let z be any string that belongs to L

Let z = $a^P\, b^P\, c^P \in L$

According to pumping lemma, if z is in L and $|z| > n$, z can be written as

$$z = uvwxy$$

$$z = a^P\ b^P\ c^P \text{ as}$$

u, vwx and y respectively, we get

$$u = a^P$$

$$vwx = b^P \qquad \text{where } |vwx| \le n$$

$$vx = b^{p-m} \qquad \text{where } |vx| \ge 1$$

$$y = c^P$$

Substituting these values in uv^iwx^iy

$= uv^{i-1}\ vwx\ x^{i-1}\ y$ (uv^iwx^iy is expressed in this form)

$= uvwx\ (vx)^{i-1}\ y$

$= a^P\ b^P\ (b^{P-m})^{i-1}\ c^P$

$= a^P\ b^P\ b^{Pi-mi-P+m}\ c^P \notin L$ for all values of i

Let $i = 0$.

$$uv^{i-1}\ vwx\ x^{i-1}\ y = a^P\ b^P\ b^{P(0)-m(0)-P+m}\ c^P$$

$$= a^P\ b^P\ b^{m-P}\ c^P$$

$$= a^P\ b^m\ c^P \notin L$$

Hence L is not a context free grammar.

13. Show that $L = \{a^K\ b^j\ c^K\ d^j \mid K \ge 1 \text{ and } j \ge 1\}$ is not context free grammar.

Solution :

Given : $L = \{a^K\ b^j\ c^K\ d^j \mid K \ge 1 \text{ and } j \ge 1\}$

Let $z = a^n\ b^P\ c^n\ d^P \in L$ where $|z| \ge n$

We split z into u, vwx and y such that it satisfies pumping lemma

Let $u = a^n$

$$vwx = b^P\ c^n \qquad \text{where } |vwx| \le n$$

$$vx = b^{P-m}\ c^{n-m} \qquad \text{where } |vx| \ge 1$$

$$y = d^P$$

uv^iwx^iy is expressed as

$uv^{i-1}vwx\,x^{i-1}\,y$ and substituting value of u, vwx, vx and y, we get

$$\begin{aligned} uv^iwx^iy &= uv^{i-1}\,vwx\,x^{i-1}\,y \\ &= u\,vwx\,(vx)^{i-1}\,y \\ &= a^n\,b^P\,c^n\,(b^{P-m}\,c^{n-m})^{i-1}\,d^P \\ &= a^n\,b^P\,c^n\,b^{(P-m)\ (i-1)}\,c^{(n-m)\ (i-1)}\,d^P \notin L \end{aligned}$$

for all values of i

$\therefore$ the given languages L = $\{a^K\,b^j\,c^K\,d^j \mid K \geq 1 \text{ and } j \geq 1\}$ is not a context free grammar.

14. Show that the language given by $L = \{0^{2^i} : i \geq 1\}$ is not context free grammar.

Solution :

Given $L = \{0^{2^i} : i \geq 1\}$

Let $z = 0^{2^P} \in L$ where $|z| \geq n$

$= 0^{2^P} = 0^m$ where $m = 2^P$

$z = 0^m$ can be represented in the form $uvwxy$ as

$u = 0^q$ where $q < m$

$vwx = 0^r$ where $r < (m\text{-}q)$ and $|vwx| \geq n$

$vx = 0^S$ where $|vx| \geq 1$

$y = 0^{m-(q+r)}$

Substituting the values of u, vwx and y in uv^iwx^iy we get

$$\begin{aligned} uv^{i-1}\,vwx\,x^{i-1}\,y &= uvwx\,(vx)^{i-1}\,y \\ &= 0^q\,0^r\,(0^S)^{i-1}\,0^{m-(q+r)} \\ &= 0^{q+r+S(i-1)+m-q-r} \\ &= 0^{m+S(i-1)} \\ &= 0^{2^P+S(i-1)} \notin \text{L for all values of } i \end{aligned}$$

Therefore L = $\{0^{2^i} : i \geq 1\}$ is not context free language.

15. Show that the language L = $\{b^{n^2} : n \geq 1\}$ is not context free.

Solution :

Given $L = \{b^{n^2} : n \geq 1\}$

Let $z = b^{m^2} = b^P$ where $P = m^2$, $|z| \geq n$

$z = b^P$ can be represented in the form $uvwxy$ which can be decomposed as

$u = b^q$ where $q < P$

$vwx = b^r$ where $r < (m\text{-}q) < P$

and $|vwx| \geq n$

$vx = b^s$ where $S < r$ and $|vx| \geq 1$

$y = b^{P-(q+r)}$

Substituting these values in uv^iwx^iy we get

$$uv^iwx^iy = uv^{i-1}wx^{i-1}y$$

$$= uvwx\ (vx)^{i-1}\ y$$

$$= b^q\, b^r\, b^{S(i-1)} b^{P-(q+r)}$$

$$= b^{q+r+S(i-1)+P-q-r}$$

$$= b^{p+S(i-1)} \notin L \text{ for all values of } i.$$

For example :

Let $m = 5$

Then $p = m^2 = 5^2 = 25$

Then $z = b^{25}$

Let $u = b^q = b^{19}$ where $q < P$

$vwx = b^r = b^3$ where $r < (m\text{-}q)$

$vx = b^S = b^1$ where $S < r$ & $|v| \geq 1$

$y = b^{P-(q+r)}$ $= b^{25-(19+3)}$

$= b^3$

Substituting these values in uv^iwx^iy we get

$b^{P+S(i-1)}$ when $i = 0$

$b^{25+1(0-1)}$ $= b^{25-1} = b^{24} \notin$ L

(Because 24 cannot be expressed as m^2 (perfect square) for any $m > 1$)

Hence language $L = \{b^{n^2} : n \geq 1\}$ is not CFL.

16. Show that the language $L = \{a^n\, b^{n+1}\, c^{n+2} : n \geq 1\}$ is not CFL.

Solution :

Given $L = \{a^n\, b^{n+1}\, c^{n+2}\}$

Let $z = a^m\, b^{m+1}\, c^{m+2} \in L$

by pumping lemma, z can be written as $uvwxy$ such that

$$u = a^m$$

$$vwx = b^{m+1} \qquad \text{where } |vwx| \leq n$$

$$vx = b^1 \qquad \text{where } |vx| \geq 1$$

$$y = c^{m+2}$$

Then $uv^iwx^iy = uvwx\,(vx)^{i-1}\,y$

$= a^m\, b^{m+1}\, b^{1(i-1)}\, c^{m+2}$

$= a^m\, b^{m+1+i-1}\, c^{m+2}$

$= a^n\, b^{m+i}\, c^{m+2} \notin L$ for all values of i.

Hence language L is not CFL.