

Introduction to Formal Languages, Automata and Computability

Context-Free Grammars - Properties and Parsing

K. Krithivasan and R. Rama

Pumping Lemma for CFL

Theorem Let L be a context-free language. Then there exists a number k (pumping length) such that if w is a string in L of length at least ' k ', then w can be written as $w = uvxyz$ satisfying the following conditions:


1. $|vy| > 0$
2. $|vxy| \leq k$
3. For each $i \geq 0$, $uv^i xy^i z \in L$

Proof Let G be a context-free grammar in Chomsky normal form generating L . Let ' n ' be the number of nonterminals of G . Take $k = 2^n$. Let ' s ' be a string in L such that $|s| \geq k$. Any parse tree in G for s must be

contd.

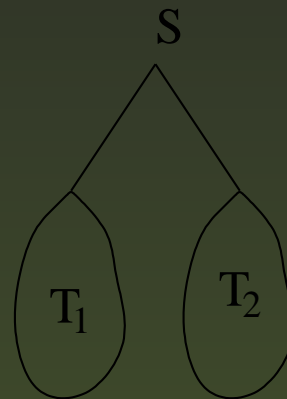
of depth at least n . This can be seen as follows:
If the parse tree has depth n , it has no path of length greater than n ; then the maximum length of the word derived is 2^{n-1} . This statement can be proved by

induction. If $n = 1$, the tree has structure $\begin{array}{c} s \\ | \\ a \end{array}$. If $n = 2$,

the tree has the structure . Assuming that the

result holds upto $i - 1$, consider a tree with depth i . No path in this tree is of length greater than i . The tree has the structure as in the above figure.

contd.



T_1 and T_2 have depth $i - 1$ and the maximum length of the word derivable in each is 2^{i-2} and so the maximum length of the string derivable in T is $2^{i-2} + 2^{i-2} = 2^{i-1}$.

Choose a parse tree for s that has the least number of nodes. Consider the longest path in this tree. This path is of length at least ' $n + 1$ '. Then there must be at least

contd.

$n + 1$ -occurrences of nonterminals along this path. Consider the nodes in this path starting from the leaf node and going up towards the root. By pigeon-hole principle some nonterminal occurring on this path should repeat. Consider the first pair of occurrences of the nonterminal A (say) which repeats while reading along the path from bottom to top. In figure 1, the repetition of A thus identified allows us to replace the subtree under the second occurrence of the nonterminal A with the subtree under the first occurrence of A . The legal parse trees are given in figure.

contd.

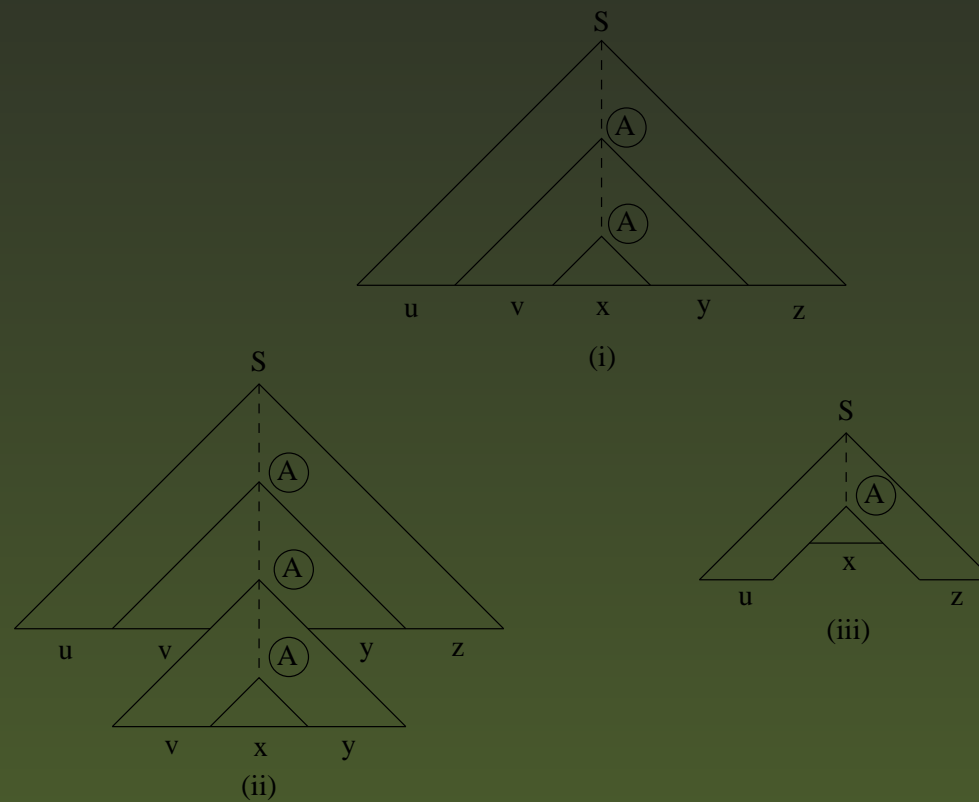


Figure 1:

contd.

We divide s as $uvxyz$ as in Figure 1(i). Each occurrence of A has a subtree under it generating a substring of s . The occurrence of A near the root of the tree generates the string ' vxy ' where the second occurrence of A produces x . Both the occurrences of A produce substrings of s . Hence one can replace the occurrence of A that produces x by a parse tree that produces vxy as shown in Figure 1(ii). Hence strings of the form $uv^i xy^i z$, for $i > 0$ are generated. One can replace the subtree rooted at A which produces ' vxy ' by a subtree which produced x as in Figure 1(iii). Hence the string ' uxz ' is generated. In essence,

$$S \xRightarrow{*} uAz \xRightarrow{*} uvAyz \xRightarrow{*} uvxyz$$

contd.

We have $A \xRightarrow{*} vAy$. Hence $A \xRightarrow{*} v^i Ay^i$.

Therefore we have $S \xRightarrow{*} uAz \xRightarrow{*} uv^i Ay^i z \xRightarrow{*} uv^i xy^i z$.

Both v and y simultaneously cannot be empty as we consider the grammar in Chomsky Normal Form. The lower A will occur in the left or right subtree. If it occurs in the left subtree, y cannot be ϵ and if it occurs in the right subtree, v cannot be ϵ .

The length of vxy is at most k , because the first occurrence of A generates vxy and the next occurrence generates x . The number of nonterminal occurrences between these two occurrences of A is less than $n + 1$.

contd.

Hence length of vxy is at most $2^n (= k)$. Hence the proof.

Example Show that $L = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free.

Suppose L is context-free. Let p be the pumping length. Choose $s = a^p b^p c^p$. Clearly $|s| > p$. Then s can be pumped and all the pumped strings must be in L . But we show that they are not. That is, we show that s can never be divided as $uvxyz$ such that $uv^i xy^i z$ as in L for all $i \geq 0$. v and y are not empty simultaneously.

contd.

If v and y can contain more than one type of symbol, then uv^2xy^2z may not be of the form $a^n b^n c^n$. If v or y contains only one type of alphabet, then uv^2xy^2z cannot contain equal number of a 's, b 's and c 's or uxz has unequal number of a 's, b 's and c 's. Thus a contradiction arises.

Hence L is not a context-free language.

Closure Properties of CFL

Theorem Let L be a context-free language over T_Σ and σ be a substitution on T such that $\sigma(a)$ is a CFL for each a in T . Then $\sigma(L)$ is a CFL.

Proof Let $G = (N, T, P, S)$ be a context-free grammar generating L . Since $\sigma(a)$ is a CFL, let $G_a = (N_a, T_a, P_a, S_a)$ be a CFG generating $\sigma(a)$ for each $a \in T$. Without loss of generality, $N_a \cap N_b = \phi$ and $N_a \cap N = \phi$ for $a \neq b, a, b \in T$. We now construct a CFG $G' = (N', T', P', S)$ which generates $\sigma(L)$ as follows :

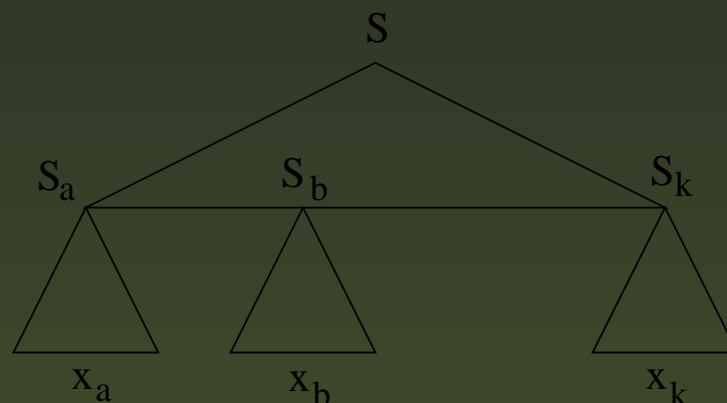
- N' is the union of N_a 's, $a \in T$ and N
- $T' = \bigcup_{a \in T} T_a$

contd.

- P' consists of :
 - all productions in P_a for $a \in T$
 - all productions in P , but for each terminal a occurring in any rule of P , is to be replaced by S_a . i.e., in $A \rightarrow \alpha$, every occurrence of a ($\in T$) in α is replaced by S_a .

Any derivation tree of G' will typically look as in the following figure.

contd.



Here $ab \dots k$ is a string of L and $x_a x_b \dots x_k$ is a string of $\sigma(L)$. To understand the working of G' producing $\sigma(L)$, we have the following discussion:

A string w is in $L(G')$ if and only if w is in $\sigma(L)$.

Suppose w is in $\sigma(L)$. Then there is some string $x = a_1 \dots a_k$ in L and strings

contd.

x_i in $\sigma(a_i)$, $1 \leq i \leq k$, such that $w = x_1 \dots x_k$.

Clearly from the construction of G' , $S_{a_1} \dots S_{a_k}$ is generated (for $a_1 \dots a_k \in L$). From each S_{a_i} , x_i 's are generated where $x_i \in \sigma(a_i)$. This becomes clear from the above picture of derivation tree. Since G' includes productions of G_{a_i} , $x_1 \dots x_k$ belongs to $\sigma(L)$.

Conversely for $w \in \sigma(L)$, we have to understand the proof with the help of the parse tree constructed above.

That is, the start symbol of G and G' are S . All the nonterminals of G , G_a 's are disjoint. Starting from S , one can use the productions of G' and G and reach

contd.

$w = S_{a_1} \dots S_{a_k}$ and $w' = a_1 \dots a_k$ respectively.

Hence whenever w has a parse tree T , one can identify a string $a_1 a_2 \dots a_k$ in $L(G)$ and string x_i in $\sigma(a_i)$ such that $x_1 \dots x_k \in \sigma(L)$. Since $x_1 \dots x_k$ is a string formed by substitution of strings x_i 's for a_i 's, we conclude $w \in \sigma(L)$.

Theorem Context-free languages are closed under union, catenation, catenation closure (*), catenation + and homomorphism.

Proof

- **Union :** Let L_1 and L_2 be two CFLs. If $L = \{1, 2\}$ and $\sigma(1) = L_1$ and $\sigma(2) = L_2$. Clearly $\sigma(L) = \sigma(L_1) \cup \sigma(L_2) = L_1 \cup L_2$ is CFL by the above theorem.

contd.

- **Catenation** : Let L_1 and L_2 be two CFLs. Let $L = \{12\}$. $\sigma(1) = L_1$ and $\sigma(2) = L_2$. Clearly $\sigma(L) = \sigma(1).\sigma(2) = L_1L_2$ is CFL as in the above case.
- **Catenation Closure (*)** : Let L_1 be a CFL. Let $L = \{1\}^*$ and $\sigma(1) = L_1$. Clearly $L_1^* = \sigma(L)$ is a CFL.
- **Catenation +** : Let L_1 be a CFL. Let $L = \{1\}^+$ and $\sigma(1) = L_1$. Clearly $L_1^+ = \sigma(L)$ is a CFL.
- **Homomorphism** : This follows as homomorphism is a particular case of substitution.

contd.

Theorem Context-free languages are not closed under intersection and complementation.

Proof Let $L_1 = \{a^n b^n c^m \mid n, m \geq 1\}$ and $L_2 = \{a^m b^n c^n \mid n, m \geq 1\}$.

Clearly L_1 and L_2 are context-free languages. (Exercise : Give CFG's for L_1 and L_2).

$L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 1\}$ which has been shown to be noncontext-free. Hence CFLs are not closed under \cap .

For nonclosure under complementation, if CFL's are closed under complementation, then for any two CFLs L_1 and L_2 , $L_1 \cap L_2 = (L_1^c \cup L_2^c)^c$ which is a CFL. Hence we get CFLs are closed under intersection, which is a contradiction.

contd.

Theorem If L is a CFL and R is a regular language, then $L \cap R$ is a CFL.

Proof Let $M = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA such that $T(M) = L$ and let $A = (\bar{K}, \Sigma, \bar{\delta}, \bar{q}_0, \bar{F})$ be a DFA such that $T(A) = R$. A new PDA M' is constructed by combining M and A such that the new automaton simulates the action of M and A on an input parallelly.

Hence the new PDA M' will be as follows:

$M' = (K \times \bar{K}, \Sigma, \Gamma, \delta', [q_0, \bar{q}_0], Z_0, F \times \bar{F})$ where $\delta'([p, q], a, X)$ is defined as follows: $\delta'([p, q], a, X)$ contains $([r, s], \gamma)$ where $\bar{\delta}(q, a) = s$ and $\delta(p, a, X)$ contains (r, γ) .

contd.

Clearly for each move of the PDA M' , there exists a move by the PDA M and a move by A . The input a may be in Σ or $a = \epsilon$. When a is in Σ , $\bar{\delta}(q, a) = s$ and when $a = \epsilon$, $\bar{\delta}(q, a) = q$ i.e., A does not change its state while M' makes a transition on ϵ .

To prove $L(M') = L \cap R$. We can show that $(q_0, w, Z_0) \stackrel{*}{\vdash}_M (q_f, \epsilon, \gamma)$ if and only if $([q_0, \bar{q}_0], w, Z_0) \stackrel{*}{\vdash}_M ([q_f, p], \epsilon, \gamma)$ where $\bar{\delta}(\bar{q}_0, w) = p$.

contd.

The proof is by induction on the number of derivation steps and is similar to that of closure of regular languages with respect to intersection. If $q_f \in F$ and $p \in \overline{F}$, then w belongs to both L and R . Therefore M' accepts $L \cap R$.

Theorem Family of context-free languages is closed under inverse homomorphism.

Decidability Results for CFL

Theorem Given a CFL L , there exists an algorithm to test whether L is empty, finite or infinite.

Proof To test whether L is empty, one can see whether the start symbol S of the CFG $G = (N, T, S, P)$ which generates L is useful or not. If S is a useful symbol, then $L \neq \phi$.

To see whether the given CFL L is infinite, we have the following discussion. By pumping lemma for CFL, if L contains a word of length t , with $|t| > k$ for a constant k (pumping length), then clearly L is infinite.

Conversely if L is infinite it satisfies the conditions of the pumping lemma, otherwise L is finite. Hence we have to test whether L contains a word of length greater than k .

CYK Algorithm

We fill a triangular table where the horizontal axis corresponds to the positions of an input string $w = a_1a_2 \dots a_n$. An entry X_{ij} which is an i th row j th column entry will be filled by a set of variables A such that $A \Rightarrow^* a_i a_{i+1} \dots a_j$. The triangular table will be filled row wise in upward fashion. For example if $w = a_1a_2a_3a_4a_5$, the table will look like,

| | | | | |
|----------|----------|----------|----------|----------|
| X_{15} | | | | |
| X_{14} | X_{25} | | | |
| X_{13} | X_{24} | X_{35} | | |
| X_{12} | X_{23} | X_{34} | X_{45} | |
| X_{11} | X_{22} | X_{33} | X_{44} | X_{55} |
| a_1 | a_2 | a_3 | a_4 | a_5 |

contd.

Note by the definition of X_{ij} , bottom row corresponds to a string of length one and top row corresponds to a string of length n , if $|w| = n$. The computation of the table is as below.

First Row (from bottom) : Since the strings beginning and ending position is i , they are simply those variable for which we have $A \rightarrow a_i$, and listed in X_{ii} . We assume that the given CFG in CNF generates L .

To compute X_{ij} which will be in $(j - i + 1)^{th}$ row we fill all the entries in the rows below. Hence we know all the variables which give strings $a_i a_{i+1} \dots a_j$. Clearly we take $j - i > 0$. Any derivation of the form $A \Rightarrow^* a_i a_{i+1} \dots a_j$ will have a derivation step like

contd.

$A \Rightarrow BC \Rightarrow^* a_i a_{i+1} \dots a_j$. B derives a prefix of $a_i a_{i+1} \dots a_j$ and C derives a suffix of $a_i a_{i+1} \dots a_j$. i.e., $B \Rightarrow^* a_i a_{i+1} \dots a_k$, $k < j$ and

$C \xRightarrow{*} a_{k+1} a_{k+2} \dots a_j$. Hence we place A in X_{ij} if, for a k , $i \leq k < j$, there is a production $A \rightarrow BC$ with $B \in X_{ik}$ and $C \in X_{k+1j}$. Since X_{ik} and X_{k+1j} entries are already known for all k , $1 \leq k \leq j$, X_{ij} can be computed.

The algorithm terminates once an entry X_{1n} is filled where n is the length of the input. Hence we have the following theorem.

Theorem The algorithm described above correctly computes X_{ij} for all i and j . Hence $w \in L(G)$, for a CFL L if and only if $S \in X_{1n}$.

contd.

Example Consider the CFG G with the following productions:

$$S_0 \rightarrow AB|SA$$

$$S \rightarrow AB|SA|a$$

$$A \rightarrow AB|SA|a|b$$

$$B \rightarrow SA$$

We shall test the membership of aba in $L(G)$ using CYK algorithm.

The table thus produced on application of CYK algorithm is as below:

| | | |
|----------------|--------|--------|
| S_0, S, A, B | | |
| S_0, S, A, B | ϕ | |
| S, A | A | S, A |
| a | b | a |

Since X_{13} has S_0 , aba is in $L(G)$.

Sub Families of CFL

Definition A CFG $G = (N, T, P, S)$ is said to be linear if all rules are of the form $A \rightarrow xBy$ or $A \rightarrow x$, $x, y \in T^*$, $A, B \in N$.
i.e., the right-hand side consists of at most one nonterminal.

Example $G = (\{S\}, \{a, b\}, P, S)$ where
 $P = \{S \rightarrow aSb, S \rightarrow ab\}$ is a linear CFG generating
 $L = \{a^n b^n \mid n \geq 1\}$.

Definition For an integer $k \geq 2$, a CFG, $G = (N, T, P, S)$ is termed k -linear if and only if each production in P is one of the three forms, $A \rightarrow xBy$, $A \rightarrow x$, or $S \rightarrow \alpha$, where α contains at most k nonterminals and S does not appear on right hand side of any production, $x, y \in T^*$.

contd.

A context-free language is k -linear if and only if it is generated by a k -linear grammar.

Example $G = (\{S, X, Y\}, \{a, b, c, d, e\}, P, S)$ where $P = \{S \rightarrow XcY, X \rightarrow aXb, X \rightarrow ab, Y \rightarrow dYe, Y \rightarrow de\}$ generates $\{a^n b^n c d^m e^m \mid n, m \geq 1\}$. This is a 2-linear grammar.

Definition A grammar G is metalinear if and only if there is an integer k such that G is k -linear. A language is metalinear if and only if it is generated by a metalinear grammar.

Definition A minimal linear grammar is a linear grammar with the initial letter S as the only nonterminal and with $S \rightarrow a$, for some terminal symbol a , as the only

contd.

production with no nonterminal on the right side. Furthermore it is assumed that a does not occur in any other production.

Example $G = (\{S\}, \{a, b\}, \{S \rightarrow aSa, S \rightarrow b\})$ is a minimal linear grammar generating $\{a^n ba^n \mid n \geq 0\}$.

Definition An even linear grammar is a linear grammar where all productions with a nonterminal Y on the right-hand side are of the form $X \rightarrow uYv$ where $|u| = |v|$.

Definition A linear grammar $G = (N, T, P, S)$ is deterministic linear if and only if all production in P are of the two forms.

$$X \rightarrow aYv \quad X \rightarrow a, \quad a \in T, v \in T^*$$

contd.

and furthermore for any $X \in N$ and $a \in T$, there is at most one production having a as the first symbol on the right-hand side.

Definition A context-free grammar $G = (N, T, P, S)$ is sequential if and only if an ordering on symbols of N can be imposed $\{X_1, \dots, X_n\}$ where $S = X_1$ and for all rules $X_i \rightarrow \alpha$ in P , we have $\alpha \in (V_T \cup \{X_j | 1 \leq j \leq n\})^*$.

Example $G = (\{X_1, X_2\}, \{a, b\}, P, X_1)$ where $P = \{X_1 \rightarrow X_2X_1, X_1 \rightarrow \epsilon, X_2 \rightarrow aX_2b, X_2 \rightarrow ab\}$ is sequential generating L^* where $L = \{a^n b^n | n \geq 1\}$.

Definition The family of languages accepted by deterministic PDA are called deterministic CFL.

contd.

Definition A PDA $M = (K, \Sigma, \Gamma, \delta, q_r, Z_0, F)$ is called a k -turn PDA, if and only if the stack increases and decreases (makes a turn) at most k times. If it makes just one turn, it is called a one turn PDA. When k is finite it is called finite line PDA. It should be noted that for some CFL number of turns of the PDA cannot be bounded.

We state some results without giving proofs.

Theorem The family of languages accepted by one turn PDA is the same as the family of linear languages.

Theorem The class of regular sets forms a subclass of even linear languages.

Definition A context-free grammar $G = (N, T, P, S)$ is said to be

contd.

ultralinear (sometimes called nonterminal bounded) if and only if there exists an integer k such that any sentential form α such that $S \xRightarrow{*} \alpha$, contains at most k nonterminals (whether leftmost, rightmost or any derivation is considered). A language is ultralinear (nonterminal bounded) if and only if it is generated by an ultralinear grammar.

Theorem The family of ultralinear languages is the same as the family of languages accepted by finite turn PDA.

For example, consider the CFL

$$L = \{w \mid w \in \{a, b\}^+, w \text{ has equal number of } a's \text{ and } b's\}.$$

For accepting arbitrarily long strings, the number of turns of the PDA cannot be bounded by some k .

contd.

Definition Let $G = (N, T, P, S)$ be a CFG. For a sentential form α , let $\#_N(\alpha)$ denote the number of nonterminals in α . Let D be a derivation of a word w in G .

$$D : S = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \alpha_1 \cdots \Rightarrow \alpha_r = w$$

The index of D is defined as

$$ind(D) = \max_{0 \leq j \leq r} \#_N(\alpha_j)$$

For a word w in $L(G)$, there may be several derivations, leftmost, rightmost, etc. Also if G is ambiguous, w may have more than one leftmost derivation.

contd.

For a word $w \in L(G)$, we define

$$\text{ind}(w, G) = \min_D \text{ind}(D)$$

where D ranges over all derivations of w in G . The index of G , $\text{ind}(G)$, is the smallest natural number u such that for all $w \in L(G)$, $\text{ind}(w, G) \leq u$. If no such u exists, G is said to be of infinite index. Finally, the index of a CFL L is defined as $\text{ind}(L) = \min_G \text{ind}(G)$ where G ranges over all the context-free grammars generating L .

We say that a CFL is of finite index then the index of L is finite.

The family of CFL with finite index is denoted as \mathcal{FI} . Sometimes, this family is also called the family

contd.

of derivation bounded languages.

Example Let $G = (\{X_1, X_2\}, \{a, b\}, P, X_1)$ where

$P = \{X_1 \rightarrow X_2X_1, X_1 \rightarrow \epsilon, X_2 \rightarrow aX_2b, X_2 \rightarrow ab\}$

is of index 2. The language consists of strings of the form

$a^{n_1}b^{n_1}a^{n_2}b^{n_2} \dots a^{n_r}b^{n_r}$. In a leftmost derivation, the maximum

number of nonterminals that can occur is 2 whereas in a

rightmost derivation it is r and keeps increasing with r . This

grammar is not a nonterminal bounded grammar but it is of finite index.

Example $L = \text{Dyck set} = \text{well formed strings of parentheses}$ is

generated by $\{S \rightarrow SS, S \rightarrow aSb, S \rightarrow ab\}$ ($a = (, b =)$). Here

we find that as the length of the string increases, and the level of

contd.

nesting increases the number of nonterminals in a sentential form keeps increasing and cannot be bounded. This CFG is not of finite index. L is not of finite index.

Definition A context-free grammar $G = (N, T, P, S)$ is termed nonexpansive if there is no nonterminal

$A \in N$ such that $A \xRightarrow{*} \alpha$ and α contains two occurrences of A . Otherwise G is expansive. The family of languages generated by nonexpansive grammars is denoted by \mathcal{NE} .

Theorem $\mathcal{NE} = \mathcal{FI}$.

Self-embedding Property

In this section we consider the self-embedding property which makes *CFL* more powerful than regular sets. Pumping lemma for CFL makes use of this property. By this property it is possible to pump equally on both sides of a substring which is lacking in regular sets.

Definition Let $G = (N, T, P, S)$ be a CFG. A nonterminal $A \in N$ is said to be self-embedding if $A \xRightarrow{*} xAy$ where $x, y \in (N \cup T)^+$. A grammar G is self-embedding if it has a self-embedding nonterminal.

A context-free grammar is nonself-embedding if none of its nonterminals are self-embedding.

contd.

Any right linear grammar is nonself-embedding as the nonterminal occurs as the rightmost symbol in any sentential form. Hence a regular set is generated by a nonself-embedding grammar. We have the following result.

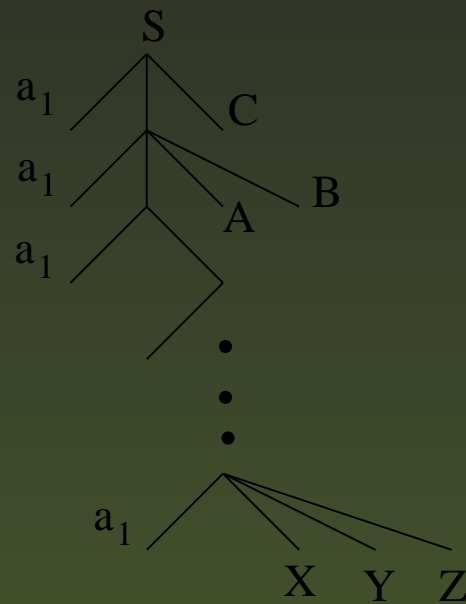
Theorem If a CFG G is nonself-embedding, then $L(G)$ is regular.

Proof Let $G = (N, T, P, S)$ be a nonself-embedding CFG. Without loss of generality we can assume that $\epsilon \notin L(G)$ and G is in GNF. [While converting a CFG to GNF, the self-embedding or nonself-embedding property does not get affected].

contd.

Let k be the number of nonterminals in G and l be the maximum length of the right-hand side of any production in G . Let $w \in L(G)$ and consider a leftmost derivation of w in G . Every sentential form is of the form $x\alpha$ where $x \in T^*$ and $\alpha \in N^*$. The length of α can be at most kl . This can be seen as follows. Suppose there is a sentential form $x\alpha$ where $|\alpha| > kl$. Consider the corresponding derivation tree which is of the form given in figure.

contd.



Consider the path from S to X , the leftmost nonterminal in α . Consider the nodes in this path where nonterminals are introduced to the right of the nodes. Since the maximum number of nodes introduced on the right

contd.

is $l - 1$, there must be more than k such nodes as $|\alpha| > kl$. So two of such nodes will have the same label say A and we get $A \xRightarrow{*} x' A \beta$, $x' \in T^+$, $\beta \in N^+$. Hence A is self-embedding and G is not nonself-embedding as supposed. Hence the maximum number of nonterminals which can occur in any sentential form in a leftmost derivation in G is kl . Construct a right linear grammar $G' = (N', T, P', S')$ such that $L(G') = L(G)$.

$$N' = \{[\alpha] \mid \alpha \in N^+, |\alpha| \leq kl\}.$$

$$S' = [S]$$

contd.

P' consists of rules of the following form.

If $A \rightarrow aB_1 \dots B_m$ is in P , then

$[A\beta] \rightarrow a[B_1 \dots B_m\beta]$ is in P' for all possible $\beta \in N^*$ such that $|A\beta| \leq kl$, $|B_1 \dots B_m\beta| \leq kl$. So if there is a derivation in G .

$$S \Rightarrow a_1\alpha_1 \Rightarrow a_1a_2\alpha_2 \Rightarrow \dots \Rightarrow a_1 \dots a_{n-1}\alpha_{n-1} \Rightarrow a_1 \dots a_n$$

there is a derivation in G' of the form

$$[S] \Rightarrow a[\alpha_1] \Rightarrow a_1a_2[\alpha_2] \Rightarrow \dots \Rightarrow a_1 \dots a_{n-1}[\alpha_{n-1}] \Rightarrow a_1 \dots a_n$$

and vice versa. Hence $L(G) = L(G')$ and $L(G)$ is regular.

contd.

Theorem Every context-free language over a one letter alphabet is regular. Thus a set $\{a^i \mid i \in A\}$ is a CFL if and only if A is ultimately periodic.

Proof Let $L \subseteq a^*$ be a context-free language. By pumping lemma for CFL, there exists an integer k such that for each word w in L such that $|w| > p$, w can be written as $uvxyz$ such that $|vxy| \leq k$, $|vy| > 0$ and $uv^i xy^i z \in L$ for all $i \geq 0$, w is in a^* . Hence u, v, x, y, z all are in a^* . So $uxz(vy)^i$ is in L for all $i \geq 0$. Let $vy = a^j$. So $uxz(a^j)^i$ is in L for all $i \geq 0$. Let $n = k(k - 1) \dots 1 = k!$. Then $w(a^n)^m$ is in L , because $w(a^n)^m$ can be written as $w(a^j)^i$

contd.

for $i = m \times \frac{k!}{j}$, $1 \leq j \leq k$. $w(a^n)^* \subseteq L \subseteq a^*$ for each word w in L such that $|w| > k$.

For each i , $1 \leq i \leq n$, let $A_i = a^{k+i}(a^n)^* \cap L$. If $A_i \neq \phi$, let w_i be the word in A_i of minimum length. If $A_i = \phi$, let w_i be undefined.

Then w is in $\bigcup_i w_i(a^n)^*$ for each w in L with $|w| > k$. Let B be

the set of strings in L of length $\leq k$. Then $L = B \cup \bigcup_{i=1}^n w_i(a^n)^*$.

B is a finite set represented by $u_1 + \dots + u_r$ (say). Then L is represented by $(u_1 + \dots + u_r) + (w_1 + \dots + w_n)(a^n)^*$. Therefore

L is regular.

contd.

Example As seen earlier, it immediately follows that $\{a^{n^2} \mid n \geq 1\}$, $\{a^{2^n} \mid n \geq 0\}$, $\{a^p \mid p \text{ is a prime}\}$ are not regular and hence they are not context-free.