

Chap. 7 Properties of Context-free Languages

7.1 Normal Forms for Context-free Grammars

Context-free grammars

$A \rightarrow \alpha$ where $A \in N$, $\alpha \in (N \cup \Sigma)^* = (N \cup \Sigma)^{\geq 0}$.

Chomsky Normal Form

$A \rightarrow BC$ or $A \rightarrow a$ except $S \rightarrow \varepsilon$ where $A, B, C \in N$, $a \in \Sigma$.

1. Eliminating *useless symbols*
2. Eliminating ε -productions (no $A \rightarrow \varepsilon$ except $S \rightarrow \varepsilon$)
3. Eliminating *unit productions* (no $A \rightarrow B$)

$A \rightarrow \alpha$ or $A \rightarrow a$ where $A \in N$, $\alpha \in (N \cup \Sigma)^{\geq 2}$, $a \in \Sigma$.

4. Introducing variables for each terminals ($A_a \rightarrow a$, $\forall a \in \Sigma$)

$A \rightarrow \alpha$ or $A \rightarrow a$ where $A \in N$, $\alpha \in N^{\geq 2}$, $a \in \Sigma$.

5. Reducing length of RHS to two

$A \rightarrow BC$ or $A \rightarrow a$ except $S \rightarrow \varepsilon$ where $A, B, C \in N$, $a \in \Sigma$.

7.1.1 Eliminating Useless Symbols

We say $X \in N \cup \Sigma$ is **useful**, if $S \Rightarrow^* \alpha X \beta \Rightarrow^* w$, $w \in \Sigma^*$,
and useless otherwise.

1. We say X is **generating**, if $X \Rightarrow^* w$, $w \in \Sigma^*$.

2. We say X is **reachable**, if $S \Rightarrow^* \alpha X \beta$.

1. Eliminate non generating symbols and productions
2. Eliminate non reachable symbols and productions

Theorem 7.2 Let $G = (N, \Sigma, P, S)$ be a CGF where $L(G) \neq \emptyset$.

1. Eliminate **non generating** symbols, and productions in G ,

$$G_2 = (N_2, \Sigma_2, P_2, S).$$

2. Eliminate **non reachable** symbols and productions from G_2 ,

$$G_1 = (N_1, \Sigma_1, P_1, S).$$

Then G_1 has no useless symbols, and $L(G_1) = L(G)$.

Theorem 7.4 Finding generating symbols

basis $\forall a \in \Sigma, a$ is generating.

induction $\forall A \rightarrow X_1 \dots X_n \in P$, if $(1 \leq \forall i \leq n: X_i$ is generating) or $(n = 0)$,
then X is generating.

Proof $\forall X \Rightarrow^* w, w \in \Sigma^*$. (induction on number of steps in algorithm)

basis zero step. $a \in \Sigma$.

induction Consider $X \Rightarrow \alpha \Rightarrow^{n-1} w, w \in \Sigma^*$.

$X \rightarrow \alpha = X_1 \dots X_n \in P. \quad 1 \leq \forall i \leq n, X_i \Rightarrow^* w_i, w_i \in \Sigma^*$ by IH.

$\therefore X \Rightarrow X_1 \dots X_n \Rightarrow^* w_1 \dots w_n = w, w \in \Sigma^*$.

Theorem 7.6 Finding reachable symbols.

basis S is reachable.

induction If $A \in N$ is reachable, $\forall A \rightarrow X_1 \dots X_n, 1 \leq \forall i \leq n, X_i$ is reachable.

Proof $\forall X, S \Rightarrow^* \alpha X \beta$. Similar to Thm 7.4.

7.1.3 Eliminating ε -Productions

$A \rightarrow \varepsilon$ is called ε -production. G is ε -free, if P has **no** ε -production.

A is **nullable**, iff $A \Rightarrow^* \varepsilon$.

Theorem 7.7 Finding nullable symbols

basis If $A \rightarrow \varepsilon \in P$, A is **nullable**.

induction $\forall A \rightarrow B_1 \dots B_n \in P$, if $(1 \leq \forall i \leq n) B_i \in N$, is **nullable**, A is **nullable**.

Proof $A \Rightarrow^* \varepsilon$, if and only if, A is in **nullable**.

(If) trivial.

(Only if) **induction** on number the shortest derivation to ε .

basis One step. $A \rightarrow \varepsilon \in P$, A is discovered in the **basis** of the algorithm.

induction Suppose $A \Rightarrow^* \varepsilon$ in $n > 1$ derivations. Then

$$A \Rightarrow B_1 \dots B_n \Rightarrow^* \varepsilon.$$

$\therefore 1 \leq \forall i \leq n, B_i \Rightarrow^{<n} \varepsilon$, Then B_i is in **nullable** by IH.

$\therefore A$ is **nullable** by induction of Thm. 7.7.

Theorem 7.9 Let $G = (N, \Sigma, P, S)$ be a cfg. Then

$\exists G_1 = (N, \Sigma, P_1, S)$ is ε -free and $L(G_1) = L(G) - \{\varepsilon\}$.

$P_1: \forall A \rightarrow X_1 \dots X_n \in P$, add $A \rightarrow Z_1 \dots Z_n$ to P_1 .

i) If X_i is **not nullable**, $Z_i = X_i$.

ii) If X_i is **nullable**, $Z_i = (X_i \mid \varepsilon)$. (if m -nullable symbols, 2^m rules)

iii) **remove** $A \rightarrow \varepsilon$. (if $m=n$, upper bound 2^{n-1} , rule)

Proof $A \Rightarrow_{G_1}^* w$ if and only if $A \Rightarrow_G^* w$ and $w \neq \varepsilon$ ($w \in \Sigma^+$).

(If) If $A \Rightarrow_G^k w$ and $w \neq \varepsilon$, $A \Rightarrow_{G_1}^* w$.

basis If $A \Rightarrow_G w$ and $w \neq \varepsilon$, and $w \in \Sigma^+$, then $A \rightarrow w \in P_1$. $\therefore A \Rightarrow_{G_1}^* w$.

induction $A \Rightarrow_G X_1 \dots X_n \Rightarrow_G^{k-1} w = w_1 \dots w_n$, $\forall X_i \Rightarrow_G^* w_i$ and $w \neq \varepsilon$.

If $w_i \neq \varepsilon$, $X_i \Rightarrow_{G_1}^* w_i$ by IH ($X_i \Rightarrow_G^{<k} x_i$).

If $w_i = \varepsilon$, X_i is nullable.

$$\therefore A \Rightarrow_G X_1 \dots X_{i-1} X_i X_{i+1} \dots X_n \Rightarrow_G^* w_1 \dots w_{i-1} \varepsilon w_{i+1} \dots w_n = w.$$

$$\therefore \exists A \Rightarrow_{G_1} X_1 \dots X_{i-1} X_{i+1} \dots X_n \Rightarrow_{G_1}^* w_1 \dots w_{i-1} w_{i+1} \dots w_n = w.$$

$$A \rightarrow X_1 \dots X_{i-1} X_{i+1} \dots X_n \in P_1.$$

(Only if) If $A \Rightarrow_{G_1}^k w$, $A \Rightarrow_G^* w$ and $w \neq \varepsilon$.

basis If $A \Rightarrow_{G_1} w$, $w \neq \varepsilon$ (G_1 is ε -free).

$$A \Rightarrow_G \alpha, \alpha \Rightarrow_G^* w \text{ (\varepsilon-rules only, } |\alpha| \geq |w|)$$

induction Assume $A \Rightarrow_{G_1} Z_1 \dots Z_n \Rightarrow_{G_1}^{k-1} w = x_1 \dots x_n$, $\forall Z_i \Rightarrow_{G_1}^* x_i$.

$\exists A \rightarrow Z_1 \dots Z_n \in P_1$ comes from $A \rightarrow X_1 \dots X_m \in P$, ($m \geq n$).

$$\therefore A \Rightarrow_G X_1 \dots X_m \Rightarrow_G^* Z_1 \dots Z_n \text{ (\varepsilon-rules only)}$$

$$\Rightarrow_G^* x_1 \dots x_n \text{ and } \forall x_i \neq \varepsilon \quad \text{by IH}(Z_i \Rightarrow_{G_1}^{<k} x_i)$$

$$= x \text{ and } x \neq \varepsilon.$$

7.1.4 Eliminating Unit productions

$A \rightarrow B$ is called a **unit production**, if $A, B \in N$.

(A, B) is called a **unit pair**, if $A \Rightarrow^* B$.

Theorem 7.11 Following algorithm finds exactly unit pairs.

basis (A, A) is a **unit pair**.

induction If (A, B) is a **unit pair** and $B \rightarrow C \in P$,
 (A, C) is a **unit pair**.

Proof Number of derivation steps unit pair is found.

basis Zero steps. $A = B$, (A, A) is added in **basis**.

induction Assume $A \Rightarrow^n C$. Then $\exists B, A \Rightarrow^{n-1} B \Rightarrow C$.

$\therefore (A, C)$ is in unit pair(IH) and the **induction** rule $B \rightarrow C \in P$
 adds (A, B) in unit pair.

Theorem 7.13 Let $G = (N, \Sigma, P, S)$ be a cfg. Then

$\exists G_1 = (N, \Sigma, P_1, S)$ that has **no** unit productions and $L(G_1) = L(G)$,
 $P_1 = \{A \rightarrow \alpha \mid (A, B) \text{ is a unit pair, } B \rightarrow \alpha \in P, \alpha \notin N\}$.

Proof $A \Rightarrow_G^* w$ if and only if $A \Rightarrow_{G_1}^* w$.

If $A \rightarrow \alpha \in P_1, \alpha \notin N. \therefore$ Non-unit productions.

(If) If $A \rightarrow \alpha \in P_1, A \rightarrow \alpha \in P$ or $A \Rightarrow_G^* B \Rightarrow_G \alpha$.

\therefore If $A \rightarrow \alpha \in P_1, A \Rightarrow_G^* \alpha$.

\therefore If $A \Rightarrow_{G_1}^* w, A \Rightarrow_G^* w$.

(Only if) If $A \Rightarrow_G^* w, A \Rightarrow_{lm} G^* w$ in G .

Assume $A = \alpha_0 \Rightarrow_{lm} \alpha_1 \Rightarrow_{lm} \alpha_2 \dots \Rightarrow_{lm} \alpha_n = w$ in G .

$0 \leq \forall i < n,$

1) If $\alpha_i \Rightarrow_{lm} \alpha_{i+1}$ by non unit production in $G, \alpha_i \Rightarrow_{lm} G_1 \alpha_{i+1}$.

2) If $\alpha_i \Rightarrow_{lm} \alpha_{i+1}$ in G by unit production,

$i < \exists k, \exists i \leq \forall j < k, \alpha_j \Rightarrow_{lm} G \alpha_{j+1}$ by unit productions
and finally $\alpha_k \Rightarrow_{lm} \alpha_{k+1}$ by non unit production

$\therefore \alpha_i \Rightarrow_{lm} G^* \alpha_{k+1}$.

If $A \Rightarrow_{lm} G^* w, A \Rightarrow_{lm} G^* w$.

7.1.5 Chomsky Normal Form(CNF)

1. $S \rightarrow \varepsilon \in P$, or
2. $A \rightarrow BC \in P$ where $B, C, \in N$, or
3. $A \rightarrow a \in P$ where $a \in \Sigma$.

Theorem 7.16 Let $G = (N, \Sigma, P, S)$ be a CFG. There is a CFG G_1 such that G_1 is CNF and $L(G) = L(G_1)$.

Proof

1. Eliminate useless symbols and productions.

2. Eliminate ε -rules.

3. Eliminate unit production.

No ε -productions and no unit productions.

If $\varepsilon \in L(G)$, $S \rightarrow \varepsilon \in P_1$.

$A \rightarrow a \in P$, CNF.

$A \rightarrow X_1 \dots X_n \in P$ where $n \geq 2$, $X_i \in N \cup \Sigma$.

$\forall X_i \in \Sigma$, $B_a \rightarrow a \in P_1$ and replace X_i by B_a .

$A \rightarrow C_1 \dots C_n \in P$ where $n \geq 2$, $C_i \in N$.

If $n = 2$, CNF.

$A \rightarrow C_1 \dots C_n \in P$ where $n \geq 3$, $C_i \in N$.

$A \rightarrow C_1 D_1 \in P_1$,

$D_1 \rightarrow C_2 D_2 \in P_1$,

...

$D_{n-3} \rightarrow C_{n-2} D_{n-2} \in P_1$,

$$D_{n-2} \rightarrow C_{n-1}C_n \in P_1.$$

Proof G_1 is CNF is trivial.

1) If $A \rightarrow X_1 \dots X_k \in P$, $A \Rightarrow_{G_1}^+ X_1 \dots X_k$.

If $A \Rightarrow_G^* w \in \Sigma^*$, $A \Rightarrow_{G_1}^* w \in \Sigma^*$.

2) If $A \Rightarrow_{G_1}^* w \in \Sigma^*$ and consider the parse tree of w in G_1 .

Convert the parse tree into the parse tree of w in G .

i) $A \rightarrow C_1 D_1, \dots, D_{n-3} \rightarrow C_{n-2} D_{n-2}, D_{n-2} \rightarrow C_{n-1} C_n$ into

$$A \rightarrow C_1 \dots C_{n-1} C_n. \text{ (Fig. 7.4)}$$

ii) $B_a \rightarrow a$ into a

$\therefore L(G) = L(G_1)$.

Regular grammar(normal form)

$A \rightarrow aB \text{ or } b$ $A, B \in N, a, b \in \Sigma.$ *right linear*

$A \rightarrow Ba \text{ or } b$ $A, B \in N, a, b \in \Sigma.$ *left linear*

Extended regular grammar

$A \rightarrow xB \text{ or } y$ $A, B \in N, x, y \in \Sigma^*.$ *right linear*

$A \rightarrow Bx \text{ or } y$ $A, B \in N, x, y \in \Sigma^*.$ *left linear*

Context-free grammar(Chomsky's normal form)

$A \rightarrow BC \text{ or } a$ $A, B, C \in N, a \in \Sigma.$

Context free grammar(extended)

$A \rightarrow \alpha$ $A \in N, \alpha \in (N \cup \Sigma)^*.$

7.2 The Pumping Lemma for context-free Languages

7.2.1 The size of parse tree

Theorem 7.17 Let $G = (N, \Sigma, P, S)$ be a Chomsky Normal Form context-free grammar and consider a parse tree for $w \in L(G)$. If n is the length (# of edges) of the longest path in the parse tree, $|w| \leq 2^{n-1}$.

Proof Induction on n ,

i) $n = 1$, $w \in \Sigma$, $|w| = 1 \leq 2^{1-1} = 1$.

ii) $n > 1$, $S \rightarrow AB$ is the root of the tree.

Two subtrees with roots A and B , respectively,

and assume $A \Rightarrow^* w_a$, $B \Rightarrow^* w_b$, and $w = w_a w_b$.

By induction hypothesis, $|w_a| \leq 2^{n-2}$ and $|w_b| \leq 2^{n-2}$.

$\therefore |w| \leq 2^{n-2} + 2^{n-2} = 2^{n-1}$.

7.2.2 Statement of the pumping Lemma

Theorem 7.18 (The pumping lemma for context-free languages)

Let L be a CFL. $\exists n \in \mathbb{N}$. \exists . if $\forall z \in L$ and $|z| \geq n$, then we write $z = uvwxy$

1) $|vwx| \leq n$,

the **first pump**

2) $vx \neq \varepsilon$,

nontrivial pump

3) $\forall i \geq 0, uv^iwx^iy \in L$.

pump

Proof

Since L is CFL, $\exists G = (N, \Sigma, P, S)$ where $L = L(G)$ and G is CNF.

Choose $n = 2^{|N|}$ and

suppose the longest path P of the parse tree for $z \in L$ is $k+1$.

$$n = 2^{|N|} \leq |z| \leq 2^{(k+1)-1} = 2^k. \text{ (Def. of } n, \text{ and Thm. 7.17)}$$

$$\therefore |N| \leq k.$$

Consider the longest path($k+1$), $(A_0, A_1, \dots, A_k, a)$

$$A_0 = S, 0 \leq \forall i \leq k: A_i \in N, a \in \Sigma. \text{ (Fig. 7.5)}$$

\therefore Since $|N| \leq k$, $1 \leq \exists i < \exists j \leq k \ .\exists. A_i = A_j = A$.

Assume $S \Rightarrow^* uA_i y \Rightarrow^* uvA_j xy \Rightarrow^* uvwxy$. (Fig. 7.6)

Note that $\underline{A} = A_i \Rightarrow^* vA_j x = \underline{vAx}$ or $\underline{A} = A_j \Rightarrow^* \underline{w}$, and $S \Rightarrow^* u\underline{A}y$.

\therefore (1) $S \Rightarrow^* uAy \Rightarrow^* uwy = uv^0wx^0y$; or

(2) Assume $S \Rightarrow^* uAy \Rightarrow^* uv^{i-1}wx^{i-1}y$ for $i \geq 1$, and

$$S \Rightarrow^* uAy \Rightarrow^* uvAxy \Rightarrow^* uvv^{i-1}wxx^{i-1}y = uv^iwx^i y. \text{ (Fig. 7.7)}$$

$\therefore S \Rightarrow^* uv^iwx^i y$ for $i \geq 0$. 3)

Since G is useful and ε -free(CNF) $v \neq \varepsilon$ and $x \neq \varepsilon$, $\therefore vx \neq \varepsilon$. 2)

We can select A_i to be the closest to the bottom of the tree, $k - i \leq |N|$,

Since the length of the longest path in A_i -subtree $\leq |N| + 1$,

$$\therefore |vwx| \leq n \text{ (Thm. 7.17)} \quad 1)$$

7.2.3 Application of the Pumping Lemma for CFL's

1. Pick L that we want to prove that L is **not** context-free.
2. “Adversary” pick n (any possible n)
3. Pick z , we may use n as a parameter
4. “Adversary” break z into $uvwxy$ \exists . $|vwx| \leq n$, $vx \neq \varepsilon$.
5. To “win” the game, find i \exists . $uv^iwx^iy \notin L$.

Context-free languages cannot match **more than two** groups of symbols for equality or inequality.

Example 7.19 $L = \{0^n 1^n 2^n \mid n \geq 1\}$

Let K be the adversary number, and $z = 0^n 1^n 2^n$,

For all breaks of z into $uvwxy$ \exists . $|vwx| \leq K$, $vx \neq \varepsilon$.

(1) $u = 0^{n-i}$, $vwx = 0^i 1^{n-i}$, and $y = 1^i 2^n$. Since $vx \neq \varepsilon$, $uwy \notin L$.

(2) $u = 0^n 1^{n-i}$, $vwx = 1^i 2^{n-i}$, and $y = 2^i$. Since $vx \neq \varepsilon$, $uwy \notin L$.

Two groups match cannot be interleaved.

Example 7.20 $L = \{0^i 1^j 2^i 3^j \mid i, j \geq 1\}$

Let n be the adversary number and $z = 0^n 1^n 2^n 3^n$.

For all breaks of z into $uvwxy$. \exists . $|vwx| \leq n$, $vx \neq \varepsilon$,

vwy : substring of at most two consecutive symbols

Nontrivial ($vx \neq \varepsilon$) pumping of v and x

Less than or equal to n symbols that is in vwx .

CFL's cannot match two strings of arbitrary length

Exercise 7.21 $L = \{ww \mid w \in (0+1)^*\}$

Consider $z = 0^n 1^n 0^n 1^n$.

7.3 Closure properties of Context Free Languages

Context-free languages are closed under

1. union,
2. concatenation,
3. closure,
4. substitution,
5. reversal

Let $G_A = (N_A, \Sigma_A, P_A, S_A)$ and $G_B = (N_B, \Sigma_B, P_B, S_B)$ be cfg's. Then

$$G_1 = (N_A \cup N_B \cup \{S_1\}, \Sigma_A \cup \Sigma_B, P_A \cup P_B \cup \{S_1 \rightarrow S_A \mid S_B\}, S_1)$$

$$\text{.}\exists. L(G_1) = L(G_A) \cup L(G_B),$$

$$G_2 = (N_A \cup N_B \cup \{S_2\}, \Sigma_A \cup \Sigma_B, P_A \cup P_B \cup \{S_2 \rightarrow S_A S_B\}, S_2)$$

$$\text{.}\exists. L(G_2) = L(G_A)L(G_B),$$

$$G_3 = (N_A \cup \{S_3\}, \Sigma_A, P_A \cup \{S_3 \rightarrow S_A S_3 \mid \varepsilon\}, S_3) \text{.}\exists. L(G_3) = L(G_A)^*,$$

$$G_5 = (N_A, \Sigma_A, \{A \rightarrow \alpha^R \mid A \rightarrow \alpha \in P\}, S_A) \text{.}\exists. L(G_5) = L(G_A)^R.$$

Context-free language is not closed under intersection

Example 7.26 We know that $L = \{0^n 1^n 2^n \mid n \geq 1\}$ in Ex. 7.19.

Consider

$$L_1 = \{0^n 1^n 2^i \mid n \geq 1, i \geq 1\}$$

$$G_1: S \rightarrow AB$$

$$A \rightarrow 0A1 \mid 01$$

$$B \rightarrow 2B \mid 2.$$

$$L_2 = \{0^i 1^n 2^n \mid n \geq 1, i \geq 1\}$$

$$G_2: S \rightarrow AB$$

$$A \rightarrow 0A \mid 0$$

$$B \rightarrow 1B2 \mid 12.$$

L_1 and L_2 are context-free but $L = L_1 \cap L_2 = \{0^n 1^n 2^n \mid n \geq 1\}$

is not context-free

counter example

Theorem 7.27 *If L is CFL and R is regular language, then*

$L \cap R$ is context-free.

Proof *Let $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z_P, F_P)$ be a PDA, $L(P) = L$, and*

$A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ be a FA, $L(A) = R$. Then

$P' = (Q_P \times Q_A, \Sigma, \Gamma, \delta, (q_P, q_A), Z_P, F_P \times F_A)$ where $a \in \Sigma \cup \{\varepsilon\}$ and

$\delta((q, p), a, X) = \{((r, s), \gamma) \mid s \in \delta_A^(p, a), (r, \gamma) \in \delta_P(q, a, X)\}$.*

Induction

$(q_P, w, Z_P) \vdash^ (q, \varepsilon, \gamma)$ if and only if $((q_P, q_A), w, Z_P) \vdash^* ((q, p), \varepsilon, \gamma)$
and $q \in \delta_A^*(q_A, w)$.*

Theorem 7.29 *If L , L_1 , and L_2 are CFL's and R is regular language.*

- 1. $L - R$ is context-free.*
- 2. \bar{L} is not (necessary) context-free.*
- 3. $L_1 - L_2$ is not (necessary) context-free.*

7.4 Decision Properties of CFL's

<i>PDA by empty stack</i> \Leftrightarrow <i>PDA by final state</i>	Thm 6.9, 11	$O(n)$
<i>CFG</i> \Rightarrow <i>PDA</i>	Thm 6.13	$O(n)$
<i>PDA</i> \Rightarrow <i>CFG</i>	Thm 6.14	$O(n^3)$

<i>CFG</i> \Rightarrow <i>CNF</i>		$O(n^2)$
1. <i>Detecting reachable and generating symbol</i>		$O(n)$
<i>Eliminating useless symbols and productions</i>		$O(n)$
2. <i>Eliminating ε-production</i>		$O(2^k)$
<i>where k is maximum length of RHS</i>		$\therefore O(2^n)$
3. <i>Eliminating unit productions</i>		$O(n^2)$
4. <i>Replacing terminal symbols by nonterminal symbols</i>		$O(n)$
5. <i>Breaking length of RHS</i>		$O(n)$
\therefore 2' <i>Eliminating ε-production</i>	$2^2 O(n)$	$O(n)$

Membership problem CYK algorithm (Coke, Younger, Kasami)

Given $w = a_1 \dots a_n \in \Sigma^*$ and a cfg G in CNF, test if $w \in L(G)$ or not.

We can compute $X_{ij} = \{A \in N \mid A \Rightarrow^* a_i \dots a_j\}$, $1 \leq i \leq j \leq n$.

If $S \in X_{1n}$, $w \in L(G)$; otherwise $w \notin L(G)$.

How to compute X_{ij} . (w.l.o.g assume CNF)

basis $X_{ii} = \{A \mid A \rightarrow a_i \in P\}$

induction Assume $A \Rightarrow^* a_i \dots a_j$. Since $i < j$, and CNF (ϵ -free)

$A \rightarrow BC \in P$ where $B \Rightarrow^* a_i \dots a_k$ and $C \Rightarrow^* a_{k+1} \dots a_j$, $i \leq k < j$.

if $B \in X_{ik}$, $C \in X_{k+1,j}$, and $A \rightarrow BC \in P$; $A \in X_{ij}$.

Test $j-i$ pairs $(X_{ii}, X_{i+1,j})$, $(X_{i,i+1}, X_{i+2,j})$, \dots , $(X_{i,j-1}, X_{jj})$

Since for each $O(n^2)$ X_{ij} , test at most n pairs,

$\therefore O(n^3)$.

CYK algorithm in PASCAL style

for $i:=1$ to n do

for $j:=i$ to n do $X_{ij} := \emptyset$; (* initialize $O(n^2)$ *)

for $i:=1$ to n do (* basis $O(n)$ *)

for $\forall A \rightarrow a \in P$ do **if $a = a_i$ then** $X_{ii} := X_{ii} \cup \{A\}$;

for $k:=1$ to $n-1$ do

for $i:=1$ to $n-k$ do (* consider $X_{i,i+k}$ *)

for $j:=i$ to $i+k$ do (* recursion $O(n^3)$ *)

for $\forall A \rightarrow BC \in P$ do

if $(B \in X_{i,j})$ and $(C \in X_{j+1,i+k})$ then $X_{i,i+k} := X_{i,i+k} \cup \{A\}$;

Some undecidable problems on CFL's

- 1. Is a given CFG G ambiguous?*
- 2. Is a given CFL L is inherently ambiguous?*
- 3. Is the intersection of two CFL's are empty?*
- 4. Are two CFL's are same?*
- 5. Is given CFL L , $L = \Sigma^*$ where Σ is the alphabet of L .*