Chapt. 5 Normal Forms of CFGs.

Normal forms of CFGs facilitate further studies of CFLs: discussions become simpler if the given CFG satisfies certain properties.

We discuss algorithms for:

- Eliminating useless symbols
- Eliminating ε-productions
- Eliminating chain productions
- Converting a CFG into Chomsky normal form.

5.1. Elimination of Useless Symbols

Useless symbols are symbols that are not used in any terminal derivation.

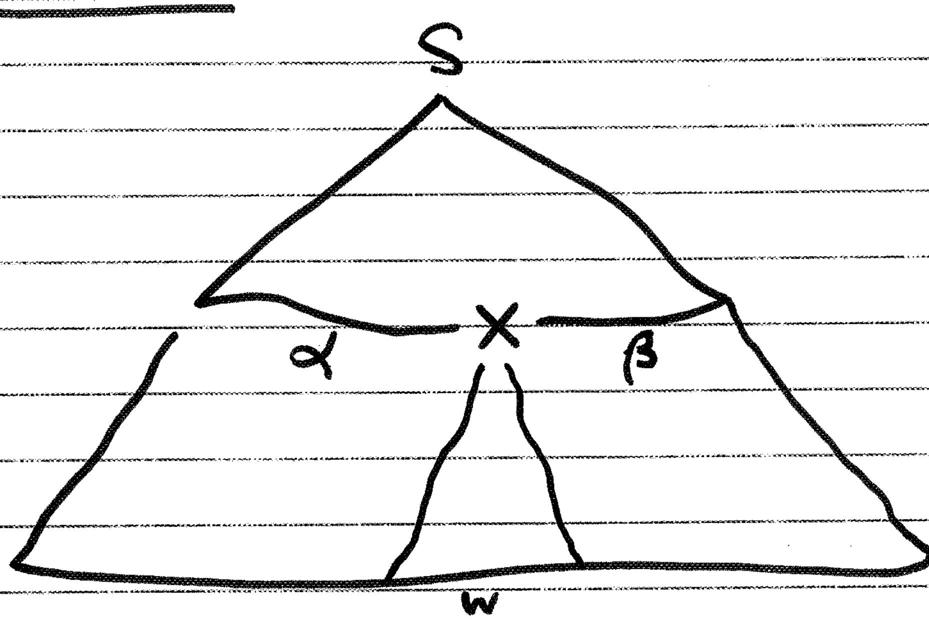
Let $G = (V, \Sigma, R, S)$ be a CFG.

A grammar symbol $X \in V \cup \Sigma$ is <u>useful</u> if there exists a derivation of the form

$$S \stackrel{*}{\Longrightarrow} \alpha X \beta \stackrel{*}{\Longrightarrow} w$$

where $\alpha, \beta \in (V \cup \Sigma)^*$, $w \in \Sigma^*$.

Illustration:



Thus, X is useful iff X appears in some terminal derivation tree.

If X is not useful, it's <u>useless</u>

Observation.

In $S \stackrel{*}{\Longrightarrow} \alpha X \beta \stackrel{*}{\Longrightarrow} w$

there are 2 subderivations:

- $S \stackrel{*}{\Longrightarrow} \alpha X \beta$, and
- $X \stackrel{*}{\Longrightarrow} v$, where v is substring of w. □

A nonterminal $A \in V$ is <u>productive</u> if there is a derivation $A \xRightarrow{*} x$, $x \in \Sigma^*$

A grammar symbol $X \in V \cup \Sigma$ is <u>reachable</u> if there is a derivation

$$S \xRightarrow{*} \alpha X \beta, \quad \alpha, \beta \in (V \cup \Sigma)^*$$

<u>Proposition</u>. Let $G = (V, \Sigma, R, S)$ be a CFG. Then all symbols in $V \cup \Sigma$ are useful iff

(1) $\forall A \in V$: A is productive

(2) $\forall X \in V \cup \Sigma$: X is reachable

<u>Pf.</u> "$\Longrightarrow$". Suppose all symbols are useful. Let $A \in V$. Since A is useful, there is a derivation

$$S \xRightarrow{*} \alpha A \beta \xRightarrow{*} w$$

by def. of useful symbols.

Thus $A \xRightarrow{*} v$, where v is a substring of w. Hence A is productive.

Now let $X \in V \cup \Sigma$ be any gram. symbol. Again, since X is useful, there is a derivation of form

$$S \overset{*}{\Longrightarrow} \alpha X \beta \overset{*}{\Longrightarrow} w$$

Clearly, X is reachable.

"$\Longleftarrow$". Suppose all nonterminals in V are productive and all gram. symbols in $V \cup \Sigma$ are reachable. We want to show that all symbols in G are useful.

Consider a grammar symbol $X \in V \cup \Sigma$. Since X is reachable, there is a derivation of form

$$S \overset{*}{\Longrightarrow} \alpha X \beta \quad , \alpha, \beta \in (V \cup \Sigma)^*.$$

Since all nonterminals are prod., for every nonterminal A in $\alpha X \beta$, there is a derivation of form

$$A \overset{*}{\Longrightarrow} u \quad , u \in \Sigma^*.$$

Thus, there is a derivation of form

$$S \overset{*}{\Longrightarrow} \alpha X \beta \overset{*}{\Longrightarrow} w, \quad w \in \Sigma^*.$$

Therefore X is useful. $\square$

Thus, in order for G to have only useful symbols we make sure:

- All nonterminals are productive
- All grammar symbols are reachable.

<u>Algorithm</u> /* Computing product. nonterm */

<u>**Input.**</u> A CFG $G = (V, \Sigma, R, S)$

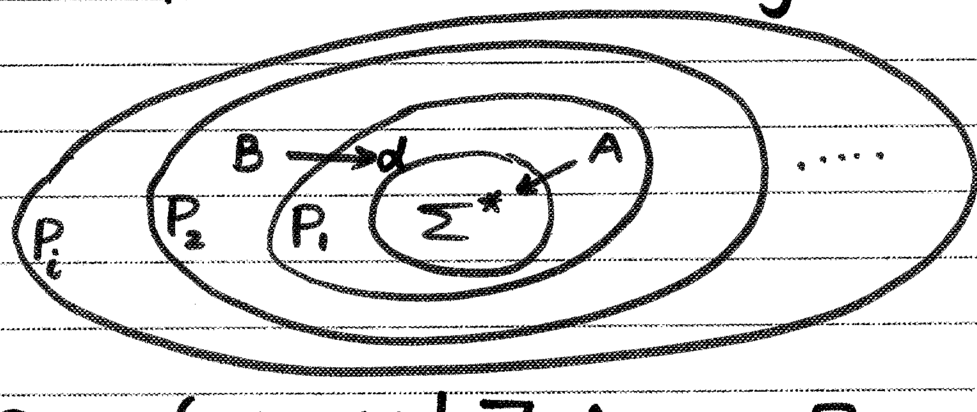
<u>**Output.**</u> $P \subseteq V$, set of prod. nonterm.

<u>**Method.**</u>

<u>Idea</u>: A is prod. if $\exists\ A \overset{*}{\Rightarrow} v$

Such a derivation $A \overset{*}{\Rightarrow} v$ is of length 1 or 2 or ....

So proceed inductively:



$P_1 := \{ A \in V \mid \exists\ A \to v \in R:\ v \in \Sigma^* \}$

$i := 1;$

<u>repeat</u> $i := i + 1;$

$P_i := P_{i-1} \cup \{ B \in V \mid \exists\ B \to \alpha \in R:\ \alpha \in (P_{i-1} \cup \Sigma)^* \}$

<u>until</u> $P_i = P_{i-1};$

$P := P_i$. □

Ex 1. G: $S \to AC \mid BS \mid B$
$A \to aA \mid aF$
$B \to CF \mid b$
$C \to cC \mid D$
$D \to aD \mid BD \mid C$
$E \to aA \mid BSA$
$F \to bB \mid b$

$P_1 = \{B, F\}$
$P_2 = \{S, A, B, F\}$
$P_3 = \{S, A, B, E, F\} = P_4$

Thus, prod. nonterm. are S, A, B, E, F and C, D are nonproductive. □

Q. Is the above algor. correct?

Proposition. A nonterm. $A \in P$ iff A is productive.

Pf. "$\Rightarrow$". Let $A \in P$. Then $A \in P_i$ for some i. By induction on i one can show that A is productive.

Basis. $i = 1$. Then $A \to v$, $v \in \Sigma^*$. By def. A is productive.

Inductive step.

Ind. hyp. Suppose that for some $i \geq 1$: $A \in P_i \Longrightarrow A$ is product.

Consider $B \in P_{i+1}$. From algor there exists $B \to \alpha \in R, \alpha \in (P_i \cup \Sigma)^*$

Let $\alpha = u_1 A_1 \dots u_k A_k u_{k+1}$, where $A_1, \dots, A_k \in P_i$, $u_1, u_2, \dots, u_{k+1} \in \Sigma^*$. As $A_1, \dots, A_k$ are productive by the ind. hyp, we have

$$A_1 \overset{*}{\Longrightarrow} v_1 \quad , \dots , \quad A_k \overset{*}{\Longrightarrow} v_k$$

Thus, $\quad B \overset{*}{\Longrightarrow} u_1 v_1 \dots u_k v_k u_{k+1} \in \Sigma^*$

i.e., B is productive.

"$\Longleftarrow$". Suppose A is productive, i.e. there is $A \overset{*}{\Longrightarrow} v$, $v \in \Sigma^*$. We show by induction on length of $A \overset{*}{\Longrightarrow} v$ that $A \in P$.

Basis. $A \overset{*}{\Longrightarrow} v$ is of length 1.

Then $A \to v \in R$. Hence $A \in P_1$

Ind. Step.

Ind. Hyp. Suppose for derivations $A \overset{*}{\Longrightarrow} v$ of length $\leq n$ that $A \in P$

Consider $A \xRightarrow{n+1} v$. Then it's of form

$$A \Rightarrow u_1 A_1 \dots u_k A_k u_{k+1} \xRightarrow{n} v.$$

It follows that

$$A_1 \xRightarrow{n_1} v_1 \ , \dots , \ A_k \xRightarrow{n_k} v_k$$

where $n_1, \dots, n_k \leq n$. By ind. hyp $A_1, \dots, A_k \in P_i$ for some i.

Hence, $A \in P_{i+1}$ by algor. $\square$

<u>Lem 1.</u> There is an algor. that constr. for a given CFG G an equiv. CFG G' whose nonterminals are productive.

<u>Pf.</u> Use the above algor. to compute the set of prod. nonterm. P for a given CFG $G = (V, \Sigma, R, S)$. Then remove all nonprod. nonterm. and any production involving a nonproductive nonterm., i.e.

$$G' = (V' = P, \Sigma, R', S)$$

where

$$R' = R \cap P \times (P \cup \Sigma)^* \qquad \square$$

Ex 1. (con't) G' :

$$\begin{aligned} S &\to BS \mid B \\ A &\to aA \mid aF \\ B &\to b \\ E &\to aA \mid BSA \\ F &\to bB \mid b \quad \square \end{aligned}$$

Ex 2. G :

$$\begin{aligned} S &\to ABC \mid BaB \\ A &\to aA \mid BaC \mid aaa \\ B &\to bBb \mid a \\ C &\to CA \mid AC \end{aligned}$$

$P_1 = \{A, B\}$

$P_2 = \{S, A, B\} = P_3$

$P = \{S, A, B\}$

Removing nonproductive nonterm. gives: G':

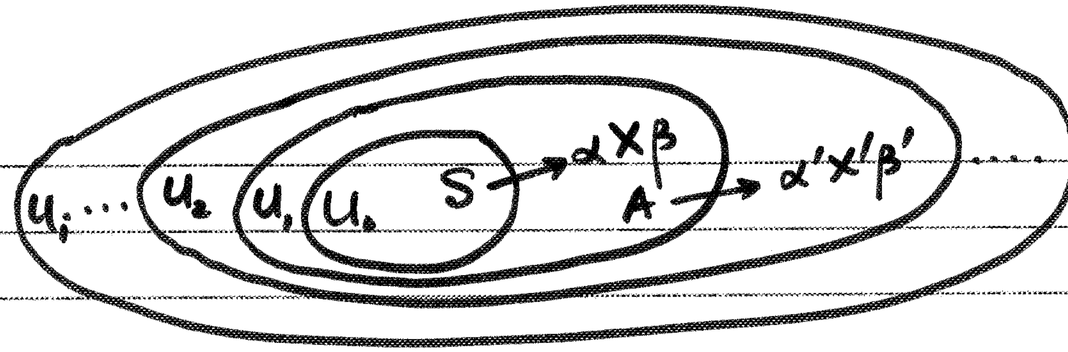
$$\begin{aligned} S &\to BaB \\ A &\to aA \mid aaa \\ B &\to bBb \mid a \quad \square \end{aligned}$$

Q. How to compute reachable symbols?

By def. X is reachable if $\exists$

$$S \stackrel{*}{\Rightarrow} \alpha X \beta.$$

As before proceed ind. on length of $S \stackrel{*}{\Rightarrow} \alpha X \beta$ to compute set of reachable symbols



Algorithm. /* Computing reachable symbols */

Input. A CFG $G = (V, \Sigma, R, S)$

Output. $U \subseteq V \cup \Sigma$, the set of reachable symbols.

Method.

$U_0 := \{S\}$; $i := 0$;

repeat $i := i+1$;

$U_i := U_{i-1} \cup \{ X \in V \cup \Sigma \mid \exists\, A \to \alpha X \beta \in R : A \in U_{i-1} \}$

until $U_i = U_{i-1}$;

$U := U_i$ □

Ex 1. (con't) G':

$S \to BS \mid B$
$A \to aA \mid aF$
$B \to b$
$E \to aA \mid BSA$
$F \to bB \mid b$

$U_0 = \{S\}$
$U_1 = \{S, B\}$
$U_2 = \{S, B, b\} = U_3$
$U = \{S, B, b\}$

G'':

$S \to BS \mid B$
$B \to b$

□

Proposition. $X \in U$ iff X is reachable.

Pf. Similar to previous proposition □

Lemma 2. There is an algor. that constr. for a given CFG G an equiv. CFG G' whose symbols are all reachable

Pf. Let $G = (V, \Sigma, R, S)$.
Use the above algor. to compute U, the set of reachable symbols.
Then remove all unreachable symbols and any productions involving an unreachable symbols, i.e. $G' = (V', \Sigma', R', S)$ where

- $V' = V \cap U$, $\Sigma' = \Sigma \cap U$,
- $R' = R \cap V' \times U^*$ □

Ex 2. (con't) G':

$$S \to BaB$$
$$A \to aA \mid aaa$$
$$B \to bBb \mid a$$

$U_0 = \{S\}$
$U_1 = \{S, B, a\}$
$U_2 = \{S, B, a, b\} = U_3$
$U = \{S, B, a, b\}$

G'':

$$S \to BaB$$
$$B \to bBb \mid a$$

□

<u>**Theorem**</u>. There is an algor. that constructs for a given CFG G an equiv. CFG G′ containing useful symbols only.

<u>Pf.</u> We apply Lem 1 first to remove all nonproductive nonterminals. We then apply Lem 2 to remove all unreachable symbols.

In the resulting CFG G′ all symb. are reachable. Moreover, the nonterminals remain productive.

From the first prop. it follows that all symbols in G′ are useful □

<u>Remark</u>. In producing the equiv. CFG G′ containing useful symbols only, we have to apply first Lem. 1 followed by Lem 2.

Reversing this order may yield a wrong result.

Ex. Consider the CFG G:

$$S \to AB \mid a$$
$$A \to a$$

$\xRightarrow{\text{Lemma 2:}}$ $U = \{S, A, B, a\}$

So, G': $S \to AB \mid a$
$A \to a$

$\xRightarrow{\text{Lemma 1}}$ $P = \{S, A\}$

So, the resulting CFG G'' is:

$$S \to a$$
$$A \to a$$

Now in G'', A becomes unreach. $\square$

Ex. Removing useless symbols in G:

$$S \to AB \mid AC$$
$$A \to aAb \mid bAa \mid a$$
$$B \to bbA \mid aaB \mid AB$$
$$C \to abCa \mid aDb$$
$$D \to bD \mid aC \qquad E \to bE \mid \varepsilon$$

$\xRightarrow{\text{Lem 1.}}$ $P_1 = \{A, E\}$
$P_2 = \{A, B, E\}$
$P_3 = \{S, A, B, E\} = P_4$

So, $P = \{S, A, B, E\}$ and

G': $S \to AB$
$A \to aAb \mid bAa \mid a$
$B \to bbA \mid aaB \mid AB$
$E \to bE \mid \varepsilon$

Lem 2 $\Rightarrow$ $U_0 = \{S\}$

$U_1 = \{S, A, B\}$

$U_2 = \{S, A, B, a, b\} = U_3$

Thus, $U = \{S, A, B, a, b\}$

and G'': $S \to AB$

$A \to aAb \mid bAa \mid a$

$B \to bbA \mid aaB \mid AB$

$\square$

5.2. Elimination of ε-productions

A production of form $A \to \varepsilon$ is called an ε-production

Goal: As ε-productions produce nothing we want to get rid of them without affecting $L(G)$.

The presence of an ε-production $A \to \varepsilon$ means that A may be erased if it appears in a sentential form.

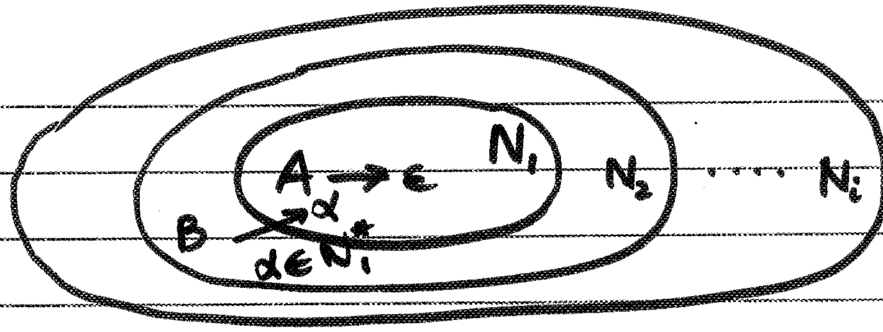
Def. A nonterm. $A \in V$ is *nullable* if there is a derivation $A \overset{*}{\Rightarrow} \varepsilon$ (i.e., A is erasable).

Q. Can we compute the set of nullable nonterminals for a given CFG G?

Idea. A derivation $A \overset{*}{\Rightarrow} \varepsilon$ is

- either of length 1 : $A \Rightarrow \varepsilon$
- or of form $A \Rightarrow A_1 \dots A_k \overset{*}{\Rightarrow} \varepsilon$

  where $A_1 \overset{*}{\Rightarrow} \varepsilon, \dots, A_k \overset{*}{\Rightarrow} \varepsilon$ are shorter derivations.



$$N_i = \{ A \in V \mid \exists A \to \alpha \in R : \alpha \in N_{i-1}^* \} \cup N_{i-1}$$

<u>Algorithm</u> /* Computing set of nullable nonterminals */

<u>Input.</u> A CFG $G = (V, \Sigma, R, S)$

<u>Output.</u> $N \subset V$, the set of nullable nonterminals

<u>Method.</u> $N_1 := \{ A \in V \mid \exists A \to \varepsilon \in R \}$

$i := 1;$

<u>repeat</u> $i := i+1;$

$N_i := N_{i-1} \cup \{ A \in V \mid \exists A \to \alpha : \alpha \in N_{i-1}^* \}$

<u>until</u> $N_i = N_{i-1}$;

$N := N_i$ □

<u>Ex:</u> G : $S \to ACA$

$A \to aAa \mid B \mid C$

$B \to bB \mid b$

$C \to cC \mid \varepsilon$

$N_1 = \{ C \}$

$N_2 = \{ A, C \}$

$N_3 = \{ S, A, C \} = N_4$ $N = \{ S, A, C \}$ □

Q. How to get rid of ε-productions without affecting $L(G)$?

Idea. Consider e.g. a production

$$A \longrightarrow \alpha_0 A_1 \alpha_1 A_2 \alpha_2 \in R$$

where A_1, A_2 are nullable symbols on the right-hand side and $\alpha_0, \alpha_1, \alpha_2$ contain no nullable symbols.

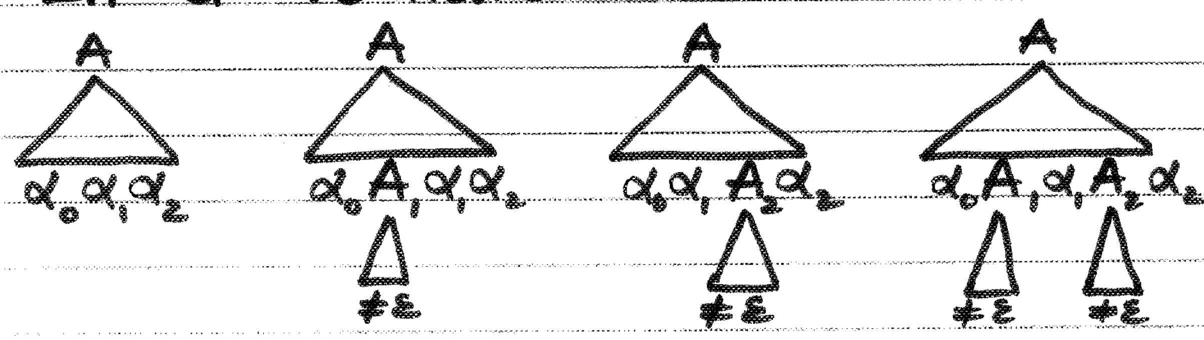
Since A_1, A_2 are erasable, in G

$$A \overset{*}{\underset{G}{\Longrightarrow}} \alpha_0 \alpha_1 \alpha_2 \mid \alpha_0 A_1 \alpha_1 \alpha_2 \mid \alpha_0 \alpha_1 A_2 \alpha_2 \mid \alpha_0 A_1 \alpha_1 A_2 \alpha_2$$

Thus, after eliminating ε-prod., in order to preserve $L(G)$, we must add the productions

$$A \longrightarrow \alpha_0 \alpha_1 \alpha_2 \mid \alpha_0 A_1 \alpha_1 \alpha_2 \mid \alpha_0 \alpha_1 A_2 \alpha_2$$

In G' we have derivation trees:



<u>Thm</u>. There is an algorithm that constr. for a given CFG G an "equiv." CFG G' s.t. $L(G') = L(G) - \{\varepsilon\}$ and G' contains <u>no</u> ε-productions.

<u>Pf</u>. Let $G = (V, \Sigma, R, S)$. First we compute N = set of nullable symb. of G. Then $G' = (V, \Sigma, R', S)$ where R' is constructed as follows:

- Take all productions from R, except ε-productions
- For a production $A \rightarrow X_1 \ldots X_m \in R$ add to R' the productions of form $A \longrightarrow \beta_1 \ldots \beta_m$

  where:

  - if $X_i \notin N$, then $\beta_i = X_i$
  - if $X_i \in N$, then $\beta_i = \varepsilon$ or $\beta_i = X_i$, i.e. X_i is erased or it stays
  - $\beta_1 \ldots \beta_m \neq \varepsilon$, i.e., the resulting production is not ε-prod

<u>Claim</u>. $\forall w \in \Sigma^+$: $A \overset{*}{\underset{G}{\Longrightarrow}} w$ iff $A \overset{*}{\underset{G'}{\Longrightarrow}} w$

i.e. $L(G') = L(G) - \{\varepsilon\}$

<u>Pf</u> (Claim) By ind. on length of derivations. $\square$

From Claim, the theorem follows. $\square$

<u>Ex</u>. (con't) G:

$$S \to ACA$$
$$A \to aAa \mid B \mid C$$
$$B \to bB \mid b$$
$$C \to cC \mid \varepsilon$$

Nullable symbols: S, A, C

G':

$$S \to ACA \mid CA \mid AA \mid AC \mid A \mid C$$
$$A \to aAa \mid aa \mid B \mid C$$
$$B \to bB \mid b$$
$$C \to cC \mid c$$

$\square$

<u>Thm</u>. There is an algorithm that constr. for a given CFG G an "equiv" CFG G' s.t.

(1) $L(G') = L(G) - \{\varepsilon\}$

(2) G' contains no ε-productions

(3) G' contains only useful symb.

<u>Pf</u>: Remove ε-prod and then remove useless symbols. $\square$

Ex. G:

$$
\begin{aligned}
S &\to ABCBCD \\
A &\to CD \\
B &\to Cb \\
C &\to A \mid \varepsilon \\
D &\to bD \mid \varepsilon
\end{aligned}
$$

Comp. nullable nonterminals:

$$N_1 = \{C, D\}$$

$$N_2 = \{A, C, D\} = N_3$$

$$N = \{A, C, D\}$$

Thus, G': $S \to ABCBCD \mid$

erase 1 symb $\Rightarrow$ $\left\{ \begin{array}{l} BCBCD \mid \\ ABBCD \mid \\ ABCBD \mid \\ ABCBC \mid \end{array} \right.$

erase 2 symb $\Rightarrow$ $\left\{ \begin{array}{ll} BBCD & \mid ABBC \\ BCBD & \mid ABCB \\ BCBC & \mid \\ ABBD & \end{array} \right.$

erase 3 symb $\Rightarrow$ $\left\{ \begin{array}{ll} BBD & \mid BCB \\ ABB & \mid BBC \end{array} \right.$

erase 4 symb $\Rightarrow$ $\{$ BB

$$
\begin{aligned}
A &\to CD \mid C \mid D \\
B &\to Cb \mid b \\
C &\to A \\
D &\to bD \mid b
\end{aligned}
$$

$\square$

5.3 Elimination of Chain Productions.

A production of form $A \rightarrow B$, where $A, B \in V$, is chain (unit, single) prod.

Using chain productions one may get derivations of form

$$A \Rightarrow B \Rightarrow C \Rightarrow \cdots \Rightarrow F$$

which is not desirable.

Q. How to get rid of chain productions without affecting $L(G)$?

Define for $A \in V$:

$$\text{CHAIN}(A) := \{ B \in V \mid \exists \text{ derivation } A \overset{*}{\Rightarrow} B \text{ of length} \geq 1 \text{ using only chain prod.} \}$$

Illustration.

(form of a linear tree)

Q. Given G and $A \in V$, can we comp. $\text{CHAIN}(A)$?

Ex. G :

$$
\begin{aligned}
S &\to A \mid CB \\
A &\to C \mid D \\
B &\to bB \mid b \\
C &\to cC \mid c \\
D &\to dD \mid d
\end{aligned}
$$

$\mathrm{CHAIN}(S) = \{A, C, D\}$

$\mathrm{CHAIN}(A) = \{C, D\}$

$\mathrm{CHAIN}(C) = \emptyset = \mathrm{CHAIN}(D) = \mathrm{CHAIN}(B)$ □

Thm. There is an algorithm that constr. for a given CFG G an equiv. CFG G' s.t.

(1) $L(G') = L(G) - \{\varepsilon\}$

(2) G' contains no ε-prod.

(3) G' contains no chain prod

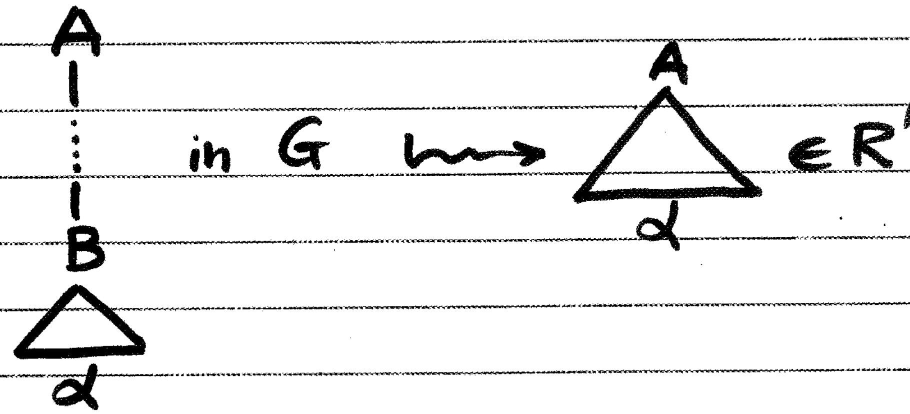
(4) All symbols in G' are useful

Pf. W.l.o.g. we assume $\varepsilon \notin L(G)$ and G contains no ε-productions.

R' is constructed from R by:

- for all $A, B \in V$, $B \in \mathrm{CHAIN}(A)$ and $B \to \alpha \in R$ which is not a chain prod. add prod. $A \to \alpha$ to R'

Illustration.



Then remove all chain productions.

~~Finally remove all useless symb.~~

Clearly $L(G') = L(G)$ $\square$

Ex. (con't) G:
$$\begin{aligned} S &\to A \mid CB \\ A &\to C \mid D \\ B &\to bB \mid b \\ C &\to cC \mid c \\ D &\to dD \mid d \end{aligned}$$

$CHAIN(S) = \{A, C, D\}$

$CHAIN(A) = \{C, D\}$ $\quad CHAN(B) = \emptyset$

$CHAIN(C) = \emptyset = CHAIN(D)$

G':
$$\begin{aligned} S &\to cC \mid c \mid dD \mid d \mid CB \\ B &\to bB \mid b \\ C &\to cC \mid c \\ D &\to dD \mid d \\ A &\to \cancel{cC \mid c} \mid dD \mid d \end{aligned}$$

(Note that A is now useless) $\square$

Ex. G: $S \to ACA \mid CA \mid AA \mid AC \mid A \mid C$

$A \to aAa \mid aa \mid B \mid C$

$B \to bB \mid b$

$C \to cC \mid c$

$\text{CHAIN}(S) = \{A, C, B\}$

$\text{CHAIN}(A) = \{B, C\}$

$\text{CHAIN}(B) = \emptyset = \text{CHAIN}(C)$

G': $S \to ACA \mid CA \mid AA \mid AC$

$aAa \mid aa$

$bB \mid b$

$cC \mid c$

$A \to aAa \mid aa$

$bB \mid b$

$cC \mid c$

$B \to bB \mid b$

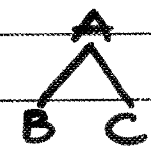
$C \to cC \mid c$

Note all symbols remain useful. □

5.4. Chomsky Normal Form.

A CFG G is in Chomsky normal form if every production is of form

(i) $A \longrightarrow BC$



(ii) $A \longrightarrow a$

where $A, B, C \in V$, $a \in \Sigma$.

Remark. We assume all symbols are useful and $\varepsilon \notin L(G)$.

Thm. There is an algor. that constr. for a given CFG G a CNF CFG G' s.t.

$$L(G') = L(G) - \{\varepsilon\}$$

Pf. W.l.o.g. we may assume that G contains no ε-prod. nor chain prod. Furthermore, all symbols are useful.

Step 1. /* Normalizing right-hand sides of productions */

- For each $a \in \Sigma$ introduce a new nonterminal D_a and a new production $D_a \longrightarrow a$

- Replace each prod. $A \to \alpha$, $|\alpha| \geq 2$ by $A \to \bar{\alpha}$, where $\bar{\alpha}$ is obtained from α by substituting each occurrence of $a \in \Sigma$ by D_a

<u>Ex:</u> G: $S \to aBbAa \mid BA$

$A \to aAb \mid a$

$B \to aBA \mid b$

Step 1 gives:

$S \to D_a B D_b A D_a \mid BA$

$A \to D_a A D_b \mid a$

$B \to D_a B A \mid b$

$D_a \to a$ $\quad$ $D_b \to b$ $\square$

Observe that after Step 1, if $A \to \alpha$ is a prod, then:

<u>Case 1.</u> $|\alpha| = 1$. Then $\alpha = a$ for some $a \in \Sigma$.

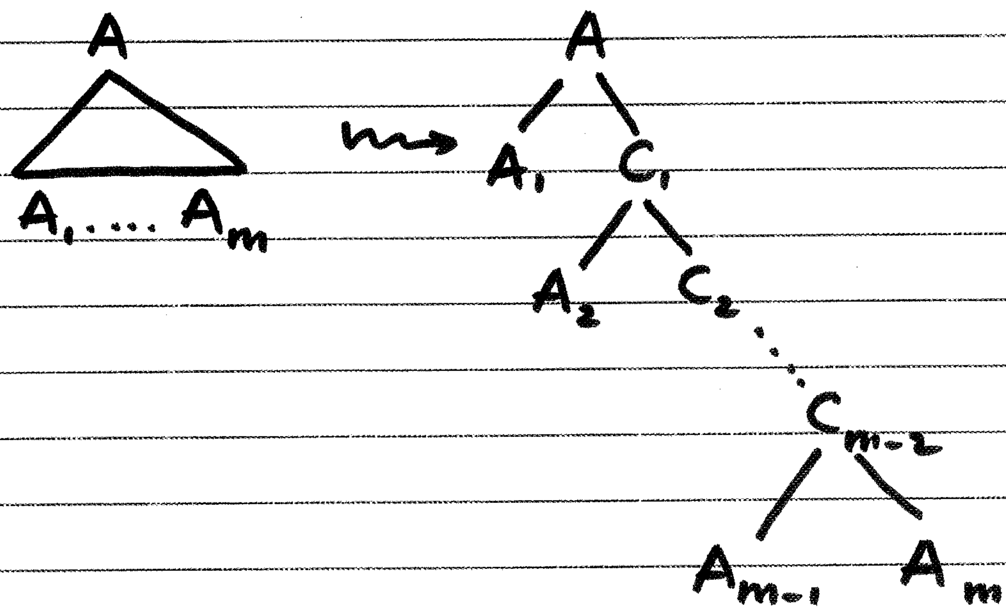
<u>Case 2.</u> $|\alpha| \geq 2$. Then $\alpha \in V^*$, i.e., α is a string of nonterminals.

Step 2. /* Converting prod. into binary form */

Let $A \to A_1 \dots A_m$, $m \geq 3$, be a production. Replace it by

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

where $C_1, \dots, C_{m-2}$ are new nonterm.



□

Ex. (con't)

$S \to D_a C_1 \mid BA$

$C_1 \to B C_2$ $\quad C_2 \to D_b C_3$ $\quad C_3 \to A D_a$

$A \to D_a E_1 \mid a$

$E_1 \to A D_b$

$B \to D_a F_1 \mid b$

$F_1 \to BA$

$D_a \to a$ $\qquad D_b \to b$ □

Ex. G: $S \rightarrow ACA \mid CA \mid AA \mid AC$
$\quad aAa \mid aa \mid bB \mid b \mid cC \mid c$

$A \rightarrow aAa \mid aa \mid bB \mid b \mid cC \mid c$

$B \rightarrow bB \mid b$

$C \rightarrow cC \mid c$

Step 1

$S \rightarrow ACA \mid CA \mid AA \mid AC$
$\quad D_a A D_a \mid D_a D_a \mid D_b B \mid b$
$\quad D_c C \mid c$

$A \rightarrow D_a A D_a \mid D_a D_a \mid D_b B \mid b$
$\quad D_c C \mid c$

$B \rightarrow D_b B \mid b$

$C \rightarrow D_c C \mid c$

$D_a \rightarrow a \quad D_b \rightarrow b \quad D_c \rightarrow c$

Step 2

$S \rightarrow AX \mid CA \mid AA \mid AC \mid D_a Y$
$\quad D_a D_a \mid D_b B \mid b$
$\quad D_c C \mid c$

$X \rightarrow CA$

$Y \rightarrow A D_a$

$A \rightarrow D_a Y \mid D_a D_a \mid D_b B \mid b \mid D_c C \mid c$

$B \rightarrow D_b B \mid b$

$C \rightarrow D_c C \mid c$ $\qquad \square$

$D_a \rightarrow a \quad D_b \rightarrow b \quad D_c \rightarrow c$

<u>Ex</u>. G: $S \to AACD$

$A \to aA \mid \varepsilon$

$C \to aC \mid a$

$D \to aD \mid Db \mid \varepsilon$

<u>Remove ε-productions:</u>

Nullable nonterm. : $\{A, D\}$

G_1: $S \to AACD \mid ACD \mid AAC \mid CD \mid AC \mid C$

$A \to aA \mid a$

$C \to aC \mid a$

$D \to aD \mid a \mid Db \mid b$

<u>Remove chain productions:</u>

G_2: $S \to AACD \mid ACD \mid AAC \mid CD \mid AC$

$aC \mid a$

$A \to aA \mid a$

$C \to aC \mid a$

$D \to aD \mid a \mid Db \mid b$

<u>Remove useless symbols:</u> No changes

<u>Convert to CNF:</u>

<u>Step1:</u> G_3: $S \to AACD \mid ACD \mid AAC \mid CD \mid AC$

$D_a C \mid a$

$A \to D_a A \mid a$

$C \to D_a C \mid a$

$D \to D_a D \mid a \mid DD_b \mid b$

$D_a \to a$ $D_b \to b$

Step2:

$S \rightarrow AX \quad X \rightarrow AY \quad Y \rightarrow CD$

$S \rightarrow AY$

$S \rightarrow AZ \quad Z \rightarrow AC$

$S \rightarrow CD \mid AC \mid D_a C \mid a$

$A \rightarrow D_a A \mid a$

$C \rightarrow D_a C \mid a$

$D \rightarrow D_a D \mid a \mid D D_b \mid b$

$D_a \rightarrow a \quad D_b \rightarrow b$