Formal languages and Automata
形式语言与自动机

Chapter6 SIMPLIFICATION OF CONTEXT-FREE GRAMMARS AND NORMAL FORMS

Beijing University of Posts and Telecommunications

北京邮电大学

刘咏彬 liiuyb@bupt.edu.cn

2023.11

TWO IMPORTANT NORMAL FORMS —Chomsky Normal Form

- Chomsky Normal Form（CNF，乔姆斯基范式）

  The string on the right of a production consist of no more than two symbols.

- **DEFINITION 6.4**

  A context-free grammar is in Chomsky normal form if all productions are of the form

$$A \to BC$$

or

$$A \to a,$$

where A, B, C are in V , and a is in T.

Chomsky Normal Form

- **EXAMPLE 6.7**

The grammar

$$S \to AS|a,$$
$$A \to SA|b$$

is in Chomsky normal form.

The grammar

$$S \to AS|AAS,$$
$$A \to SA|aa$$

is not; both productions $S \to AAS$ and $A \to aa$ violate the conditions of **Definition 6**

Chomsky Normal Form

- THEOREM 6.6

  Any context-free grammar $G = (V, T, S, P)$ with $\lambda \notin \mathrm{L(G)}$ has an equivalent grammar $\widehat{G} = (\widehat{V}, \widehat{T}, S, \widehat{P})$ in Chomsky normal form.

- Proof

  Because of Theorem 6.5, we can assume without loss of generality that G has no λ-productions and no unit-productions.（前提：G是经过简化后的CFG文法)

  The construction of $\widehat{G}$ will be done in two steps.

Chomsky Normal Form — construction step 1

- Step 1: Removes all terminals from productions whose right side has length greater than one

Construct a grammar $G_1 = (V_1, \mathrm{T}, \mathrm{S}, P_1)$ from G by considering all productions in P in the form

$$\mathrm{A} \to x_1 x_2 \cdots x_n, \tag{6.5}$$

where each x_i is a symbol either in V or in T.

If $\mathrm{n} = 1$, then x_1 must be a terminal since we have no unit-productions. In this case, put the production into P_1.

If $\mathrm{n} \geq 2$, introduce new variables B_a for each $\mathrm{a} \in \mathrm{T}$. For each production of P in the form (6.5) we put into P_1 the production

$$\mathrm{A} \to C_1 C_2 \cdots C_n,$$

$$\mathrm{S} \to \mathrm{AS} | \mathrm{AAS}, \quad \mathrm{A} \to \mathrm{SA} | \mathrm{aa}$$

where

$$C_i = x_i \text{ if } x_i \text{ is in } \mathrm{V},$$

and

$$C_i = B_a \text{ if } x_i = \mathrm{a}.$$

For every B_a we also put into P_1 the production

B_a

a.

Chomsky Normal Form — construction step 1

At the end of this step we have a grammar G_1 all of whose productions have the form

$$A \rightarrow a, \tag{6.6}$$

or

$$A \rightarrow C_1 C_2 \cdots C_n, \tag{6.7}$$

where $C_i \in V_1$.

Chomsky Normal Form — construction step 2

- Step 2: Reduce the length of the right sides

First we put all productions of the form (6.6) as well as all the productions of the form (6.7) with $\mathrm{n} = 2$ into $\widehat{P}$.
For $\mathrm{n} > 2$, we introduce new variables $D_1, D_2, \ldots$and put into $\widehat{P}$ the productions

$$A \rightarrow C_1 D_1,$$

$$D_1 \rightarrow C_2 D_2, .$$

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

$\mathrm{S} \rightarrow \mathrm{AS} | \mathrm{AAS},$
$\mathrm{A} \rightarrow \mathrm{SA} | \mathrm{aa}$

Obviously, the resulting grammar $\widehat{G}$ is in Chomsky normal form. Repeated applications of Theorem 6.1 will show that $\mathrm{L}(G_1) = \mathrm{L}(\widehat{G})$, so that $\mathrm{L}(\widehat{G}) = \mathrm{L}(G)$

EXAMPLE 6.8

Convert the grammar with productions

$$\begin{aligned} S &\rightarrow ABa, \\ A &\rightarrow aab, \\ B &\rightarrow Ac \end{aligned}$$

to Chomsky normal form.

As required by the construction of Theorem 6.6, the grammar does not have any λ-productions or any unit-productions.

In Step 1, we introduce new variables B_a, B_b, B_c and use the algorithm to get

$$\begin{aligned} S &\rightarrow ABB_a, \\ A &\rightarrow B_aB_aB_b, \\ B &\rightarrow AB_c, \\ B_a &\rightarrow a, \\ B_b &\rightarrow b, \\ B_c &\rightarrow c. \end{aligned}$$

As required by the construction of Theorem 6.6, the grammar does not have any λ-productions or any unit-productions.

In Step 1, we introduce new variables B_a, B_b, B_c and use the algorithm to get

$$\begin{aligned} S &\rightarrow ABB_a, \\ A &\rightarrow B_aB_aB_b, \\ B &\rightarrow AB_c, \\ B_a &\rightarrow a, \\ B_b &\rightarrow b, \\ B_c &\rightarrow c. \end{aligned}$$

In the second step, we introduce additional variables to get the first two productions into normal form and we get the final result

$$\begin{aligned} S &\rightarrow AD_1, \\ D_1 &\rightarrow BB_a, \\ A &\rightarrow B_aD_2, \\ D_2 &\rightarrow B_aB_b, \\ B &\rightarrow AB_c, \\ B_a &\rightarrow a, \\ B_b &\rightarrow b, \\ B_c &\rightarrow c. \end{aligned}$$

TWO IMPORTANT NORMAL FORMS
—Greibach Normal Form

- Greibach Normal Form（GNF，格里巴克范式）

- **DEFINITION 6.5**

  A context-free grammar is said to be in Greibach normal form if all productions have the form

  $$A \rightarrow ax,$$

  where $a \in T$ and $x \in V^*$.

  (a是1个终极符开头，x是任意变量的串)

EXAMPLE 6.9

The grammar

$$
\begin{aligned}
S &\rightarrow AB, \\
A &\rightarrow aA \,|bB|\, b, \\
B &\rightarrow b
\end{aligned}
$$

is not in Greibach normal form. However, using the substitution given by Theorem 6.1, we immediately get the equivalent grammar

$$
\begin{aligned}
S &\rightarrow aAB \,|bBB|\, bB, \\
A &\rightarrow aA \,|bB|\, b, \\
B &\rightarrow b,
\end{aligned}
$$

which is in Greibach normal form.

EXAMPLE 6.10

Convert the grammar

$$S \rightarrow abSb|aa$$

into Greibach normal form.

Here we can use a device similar to the one introduced in the construction of Chomsky normal form. We introduce new variables A and B that are essentially synonyms for a and b, respectively. Substituting for the terminals with their associated variables leads to the equivalent grammar

$$\begin{aligned} S &\rightarrow aBSB|aA, \\ A &\rightarrow a, \\ B &\rightarrow b, \end{aligned}$$

which is in Greibach normal form.

补充：

- $A \rightarrow A0 \mid 1 \mid 2$

- $A \rightarrow B1 \mid 0$,

  $B \rightarrow A0 \mid 1$

补充：递归的定义

- 定义：递归(recursive)
- 如果G中存在形如$A \Rightarrow^{n} \alpha A \beta$的派生，则称该派生是关于变量A递归的，简称为递归派生。
  - 当n=1时，称该派生关于变量A直接递归(directly recursive)，简称为直接递归派生。形如$A \rightarrow \alpha A \beta$的产生式是变量A的直接递归的(directly recursive)产生式。
  - 当$n \geq 2$时，称该派生是关于变量A的间接递归(indirectly recursive)派生。简称为间接递归派生。
  - 当$\alpha = \varepsilon$时，称相应的(直接/间接)递归为(直接/间接)左递归(left-recursive)；
  - 当$\beta = \varepsilon$时，称相应的(直接/间接)递归为(直接/间接)右递归(right-recursive)。

消除直接左递归

- 左递归对语言句子的分析是不利的，一般要消除文法中的左递归。思路是，将左递归变成右递归。
- 对于任意的CFG $G=(V,T,S,P)$，G中所有A的产生式

$$
\begin{cases}
A \to \beta_1 \mid \beta_2 \mid \ldots \mid \beta_m \\
A \to A\alpha_1 \mid A\alpha_2 \mid \ldots \mid A\alpha_n
\end{cases}
$$

观察：

$$A \to 1 \mid 2$$
$$A \to A0$$

可以被等价地替换为产生式组

$$
\begin{cases}
A \to \beta_1 \mid \beta_2 \mid \ldots \mid \beta_m \\
A \to \beta_1 B \mid \beta_2 B \mid \ldots \mid \beta_m B \\
B \to \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_n \\
B \to \alpha_1 B \mid \alpha_2 B \mid \ldots \mid \alpha_n B
\end{cases}
$$

CFG->GNF step1

- **THEOREM 6.7**

For every context-free grammar G with $\lambda \notin L(G)$, there exists an equivalentgrammar $\widehat{G}$ in Greibach normal form.

Proof: 设G为化简过的文法，下面分3步进行规范化处理：

step1：构造$G_1=(V_1,T,P_1,S)$,使得$L(G_1)=L(G)$,

G_1中的产生式都化成如下形式的产生式：

$A \to A_1A_2 \ldots A_m$

$A \to aA_1A_2 \ldots A_{m-1}$

$A \to a$

其中，A，A_1，A_2，…，$A_m \in V_1$，$a \in T$，$m \geq 2$。

对于P的每个产生式$A \to \alpha$，如果$\alpha \in T \cup V^+ \cup TV^+$（符合范式要求），则直接将$A \to \alpha$放入$P_1$；否则，对$A \to \alpha$进行如下处理：

设$\alpha = X_1X_2 \ldots X_m$则对每一个$X_i$，$i \geq 2$，如果$X_i = a \in T$，则引入新变量$A_a$（放入$V_1$）和产生式$A_a \to a$（放入$P_1$），用$A_a$替换产生式$A \to \alpha$中的$X_i$，然后将处理后的形如$A \to A_1A_2 \ldots A_m$或者$A \to aA_1A_2 \ldots A_{m-1}$的产生式放入$P_1$

CFG->GNF step2 Remove left recursive

step2：消除左递归（包括间接左递归和直接左递归）

- step2-1：对V_1中的变量标记顺序：

设$V_1=(A_1,A_2,\ldots\ldots A_m)$，构造$G_2=(V_2,T,P_2,S)$，使得$L(G_2)=L(G_1)$，且$G_2$中的产生式都是形如：

$A_i \to A_j\alpha \quad i<j$

$A_i \to a\alpha$

$B_i \to \alpha$

理解：i,j代表变量定义的顺序，i<j表示产生式右侧变量一定是在产生式左侧变量后边定义的

举例：
$A \to B\alpha$
$B \to C\beta$
$C \to A\psi$

其中，$V_2=V_1\cup\{B_1, B_2, \ldots, B_n\}$，$V_1\cap\{B_1, B_2, \ldots, B_n\}=\Phi$。$\{B_1, B_2, \ldots, B_n\}$是在文法改造过程中引入的新变量，$\alpha\in V_2^*$，$a\in T$

CFG->GNF step2 Remove left recursive

step2-2: ：输入： $G_1=(V_1,T,P_1,S)$ 输出： $G_2=(V_2,T,P_2,S)$

排序，使 $A_i \to A_j\alpha$ i<=j

① for k=1 to m do
  begin
② for j=1 to k-1 do
③ for每个形如$A_k \to A_j\alpha$的产生式do
  begin
④ 标记产生式$A_k \to A_j\alpha$。
  设$A_j \to \gamma_1|\gamma_2|...|\gamma_n$为所有$A_j$产生式
  根据Theorom 6-1，将产生式组$A_k \to \gamma_1\alpha|\gamma_2\alpha|...|\gamma_n\alpha$添加到产生式集合$P_2$中；
  end

消除 $A_k \to A_k\alpha$

⑤ 设$A_k \to A_k\alpha_1|A_k\alpha_2|...|A_k\alpha_p$是所有的右部第一个字符为$A_k$的$A_k$产生式，$A_k \to \beta_1|\beta_2|...|\beta_q$是所有其他的$A_k$产生式。根据消除直接左递归的方法，标记所有的$A_k$产生式，
  并引入新的变量B，将下列产生式添加到产生式集合P_2中：
  $A_k \to \beta_1|\beta_2|...|\beta_q$
  $A_k \to \beta_1B|\beta_2B|...|\beta_qB$
  $B \to \alpha_1|\alpha_2|...|\alpha_p$
  $B \to \alpha_1B|\alpha_2B|...|\alpha_pB$
  end
⑥ 将P_1中未被标记的产生式全部添加到产生式集合P_2中

CFG->GNF step3

step3：

设$V_2=V_1\cup(B_1,B_2,\ldots\ldots B_n)$,构造$G_3=(V_3,T,P_3,S)$,使得$L(G_3)=L(G_2)$，根据THEOREM 6.1(产生式代入)，构造等价文法G_3。

输入：$G_2=(V_2,T,P_2,S)$

输出：$G_3=(V_3,T,P_3,S)$

主要步骤：

① for k=m-1 to 1 do (m是V_1中变量的个数)

② if $A_k\to A_j\beta\in P_2$ & j>k then

③ for 所有的A_j产生式$A_j\to\gamma$ do

将产生式$A_k\to\gamma\beta$放入P_3；

④ for k= 1 to n do (n是V_2中新引入的变量的个数)

⑤ 用P_3中的产生式将所有的B_k产生式替换成满足GNF要求的形式。

GNF 例子

将下列文法转化为GNF

$A_1 \to A_2 \mathrm{b} A_3 | aA_1$
$A_2 \to A_3 \mathrm{c} A_3 | b$
$A_3 \to A_1 \mathrm{c} A_3 | A_2 \mathrm{bb} | \mathrm{a}$

$A_1 \to A_2 \mathrm{B} A_3 | aA_1$
$A_2 \to A_3 \mathrm{C} A_3 | b$
$A_3 \to A_1 \mathrm{C} A_3 | A_2 \mathrm{BB} | \mathrm{a}$
$\mathrm{B} \to b$
$\mathrm{C} \to \mathrm{c}$

step1：将产生式都化成下面形式:

$A \to A_1 A_2 \ldots A_m$

$A \to aA_1 A_2 \ldots A_{m-1}$

$A \to a$

引入变量B和C，对文法进行改造，将文法产生式右侧不在最左侧的终结符号替换为变量。

Greibach Normal Form

间接左递归

$A_1 \to A_2BA_3|aA_1$
$A_2 \to A_3CA_3|b$
$A_3 \to A_1CA_3|A_2BB|a$
$B \to b$
$C \to c$

$A_1 \to A_2BA_3|aA_1$
$A_2 \to A_3CA_3|b$
$A_3 \to A_2BA_3CA_3|aA_1CA_3|A_2BB|a$
$B \to b$
$C \to c$

step2目标：$A_i \to A_j\alpha$　$i<j$

$A_i \to a\alpha$
$B_i \to \alpha$

1.由于$A_1 \to A_2BA_3|\alpha A_1$和$A_2 \to A_3CA_3|b$满足定理证明中step2的要求，所以暂时不做处理

2.$A_3 \to A_1CA_3$不满足要求，所以用A_1产生式的所有的候选式替换产生式$A_3 \to A_1CA_3$中所有的A_1，得到所有的A_3产生式

Greibach Normal Form

$A_1 \to A_2BA_3|aA_1$
$A_2 \to A_3CA_3|b$
$A_3 \to A_2BA_3CA_3|aA_1CA_3|A_2BB|a$
$B \to b$
$C \to c$

step2 目标：$A_i \to A_j\alpha \quad i<j$

$A_i \to a\alpha$
$B_i \to \alpha$

再用A_2，产生式的所有候选式替换产生式$A_3 \to A_2BA_3CA_3$和$A_3 \to A_2BB$中的A_2，得到所有的A_3产生式

$A_1 \to A_2BA_3|aA_1$
$A_2 \to A_3CA_3|b$
$A_3 \to A_3CA_3BA_3CA_3|b\,BA_3CA_3|aA_1CA_3|A_3CA_3BB|bBB|a$
$B \to b$
$C \to c$

出现直接左递归，下面消除直接左递归

Greibach Normal Form

第二步目标：$A_i \to A_j\alpha \quad i<j$

$A_i \to a\alpha$
$B_i \to \alpha$

$$
\begin{aligned}
&A_1 \to A_2BA_3|aA_1\\
&A_2 \to A_3CA_3|b\\
&A_3 \to A_3CA_3BA_3CA_3|b\ BA_3CA_3|aA_1CA_3|A_3CA_3BB|bBB|a\\
&B \to b\\
&C \to c
\end{aligned}
$$

引入变量B_1，消除A_3产生式中的左递归

$$
\begin{aligned}
&A_1 \to A_2BA_3|aA_1\\
&A_2 \to A_3CA_3|b\\
&A_3 \to b\ BA_3CA_3\ |\ aA_1CA_3|bBB|a\\
&A_3 \to b\ BA_3CA_3\ B_1|\ aA_1CA_3B_1|bBBB_1|aB_1\\
&B_1 \to CA_3BA_3CA_3\ |\ CA_3BB\\
&B_1 \to CA_3BA_3CA_3\ B_1|\ CA_3BBB_1\\
&B \to b\\
&C \to c
\end{aligned}
$$

回顾：消除直接左递归

$$
\begin{cases}
A \to A\alpha_1\ |A\alpha_2\ |\ldots|A\alpha_n\\
A \to \beta_1\ |\beta_2\ |\ldots|\beta_m
\end{cases}
$$

可以被等价地替换为产生式组

$$
\begin{cases}
A \to \beta_1\ |\beta_2\ |\ldots|\beta_m\\
A \to \beta_1 B\ |\beta_2\ B\ |\ldots|\beta_m\ B\\
B \to \alpha_1\ |\alpha_2\ |\ldots|\alpha_n\\
B \to \alpha_1 B\ |\alpha_2\ B\ |\ldots|\alpha_n\ B
\end{cases}
$$

Greibach Normal Form

$A_1 \rightarrow A_2BA_3|aA_1$
$A_2 \rightarrow A_3CA_3|b$
$A_3 \rightarrow b\ BA_3CA_3\ |\ aA_1CA_3|bBB|a$
$A_3 \rightarrow b\ BA_3CA_3\ B_1|\ aA_1CA_3B_1|bBBB_1|aB_1$
$B_1 \rightarrow CA_3BA_3CA_3\ |\ CA_3BB$
$B_1 \rightarrow CA_3BA_3CA_3\ B_1|\ CA_3BBB_1$
$B \rightarrow b$
$C \rightarrow c$

$A_1 \rightarrow A_2BA_3|aA_1$
$A_2 \rightarrow b\ BA_3CA_3\ CA_3|\ aA_1CA_3CA_3|bBBCA_3|$
$b\ BA_3CA_3\ B_1CA_3|\ aA_1CA_3B_1CA_3|\ aCA_3$
$bBBB_1CA_3|bBBB_1CA_3|b$
$A_3 \rightarrow b\ BA_3CA_3\ |\ aA_1CA_3|bBB|a$
$A_3 \rightarrow b\ BA_3CA_3\ B_1|\ aA_1CA_3B_1|bBBB_1|aB_1$
$B_1 \rightarrow CA_3BA_3CA_3\ |\ CA_3BB$
$B_1 \rightarrow CA_3BA_3CA_3\ B_1|\ CA_3BBB_1$
$B \rightarrow b$
$C \rightarrow c$

第三步目标：$A \rightarrow aA_1A_2 \ldots A_m$ $(m \geq 1)$
$A \rightarrow a$

第三步：

1.具有最大下标的A_3已经满足GNF的要求
2.将这些产生式带入还不满足要求的A_2产生式，使得A_2产生式都满足GNF的要求

Greibach Normal Form

$A_1 \rightarrow A_2 BA_3 | aA_1$
$A_2 \rightarrow b\ BA_3CA_3\ CA_3 |\ aA_1CA_3CA_3 | bBBCA_3 |$
$\quad b\ BA_3CA_3\ B_1\ CA_3 |\ aA_1CA_3B_1CA_3 |$
$\quad bBBB_1CA_3 | bBBB_1CA_3 | b$
$A_3 \rightarrow b\ BA_3CA_3 \ |\ aA_1CA_3 | bBB | a$
$A_3 \rightarrow b\ BA_3CA_3\ B_1 |\ aA_1CA_3B_1 | bBBB_1 | aB_1$
$B_1 \rightarrow CA_3BA_3CA_3 \ |\ CA_3BB$
$B_1 \rightarrow CA_3BA_3CA_3\ B_1 |\ CA_3BBB_1$
$B \rightarrow b$
$C \rightarrow c$

$A_1 \rightarrow b\ BA_3CA_3\ CA_3BA_3 |\ aA_1CA_3CA_3BA_3 | bBBCA_3BA_3 |$
$\quad b\ BA_3CA_3\ B_1\ CA_3BA_3 |\ aA_1CA_3B_1CA_3BA_3 |\ aCA_3BA_3\ |$
$\quad bBBB_1CA_3BA_3 | bBBB_1CA_3BA_3 | bBA_3 | aA_1$
$A_2 \rightarrow b\ BA_3CA_3\ CA_3 |\ aA_1CA_3CA_3 | bBBCA_3 |$
$\quad b\ BA_3CA_3\ B_1\ CA_3 |\ aA_1CA_3B_1CA_3 |$
$\quad bBBB_1CA_3 | bBBB_1CA_3 | b$
$A_3 \rightarrow b\ BA_3CA_3 \ |\ aA_1CA_3 | bBB | a$
$A_3 \rightarrow b\ BA_3CA_3\ B_1 |\ aA_1CA_3B_1 | bBBB_1 | aB_1$
$B_1 \rightarrow CA_3BA_3CA_3 \ |\ CA_3BB$
$B_1 \rightarrow CA_3BA_3CA_3\ B_1 |\ CA_3BBB_1$
$B \rightarrow b$
$C \rightarrow c$

第三步目标：$A \rightarrow aA_1A_2 \ldots A_m \quad (m \geq 1)$

$A \rightarrow a$

1.A_2已经满足GNF的要求
2.将这些产生式带入还不满足要求的A产生式，使得A_1产生式都满足GNF的要求

Greibach Normal Form

$A_1 \rightarrow b\ BA_3CA_3\ CA_3BA_3|\ aA_1CA_3CA_3BA_3|bBBCA_3BA_3|$
$\quad b\ BA_3CA_3\ B_1CA_3BA_3|\ aA_1CA_3B_1CA_3BA_3|$
$\quad bBBB_1CA_3BA_3|bBBB_1CA_3BA_3|bBA_3|aA_1$
$A_2 \rightarrow b\ BA_3CA_3\ CA_3|\ aA_1CA_3CA_3|bBBCA_3|$
$\quad b\ BA_3CA_3\ B_1CA_3|\ aA_1CA_3B_1CA_3|$
$\quad bBBB_1CA_3|bBBB_1CA_3|b$
$A_3 \rightarrow b\ BA_3CA_3\ |\ aA_1CA_3|bBB|a$
$A_3 \rightarrow b\ BA_3CA_3\ B_1|\ aA_1CA_3B_1\ |bBBB_1\ |aB_1$
$B_1 \rightarrow CA_3BA_3CA_3\ |\ CA_3BB$
$B_1 \rightarrow CA_3BA_3CA_3\ B_1|\ CA_3BBB_1$
$B \rightarrow b$
$C \rightarrow c$

$A_1 \rightarrow b\ BA_3CA_3\ CA_3BA_3|\ aA_1CA_3CA_3BA_3|bBBCA_3BA_3|$
$\quad b\ BA_3CA_3\ B_1CA_3BA_3|\ aA_1CA_3B_1CA_3BA_3|\ aCA_3BA_3\ |$
$\quad bBBB_1CA_3BA_3|bBBB_1CA_3BA_3|bBA_3|aA_1$
$A_2 \rightarrow b\ BA_3CA_3\ CA_3|\ aA_1CA_3CA_3|bBBCA_3|$
$\quad b\ BA_3CA_3\ B_1CA_3|\ aA_1CA_3B_1CA_3|$
$\quad bBBB_1CA_3|bBBB_1CA_3|b$
$A_3 \rightarrow b\ BA_3CA_3\ |\ aA_1CA_3|bBB|a$
$A_3 \rightarrow b\ BA_3CA_3\ B_1|\ aA_1CA_3B_1\ |bBBB_1\ |aB_1$
$B_1 \rightarrow cA_3BA_3CA_3\ |cA_3BB|cA_3BA_3CA_3\ B_1|cA_3BBB_1$
$B \rightarrow b$
$C \rightarrow c$

完成！

第三步目标：$A \rightarrow aA_1A_2\ldots A_m \quad (m \geq 1)$

$A \rightarrow a$

最后，将所有的B_1产生式变换成满足GNF要求的形式

A MEMBERSHIP ALGORITHM FOR CFG

- CYK algorithm （named after its originators J. Cocke, D. H. Younger, and T. Kasami)
- $O(|w|^3)$
- Prerequisite： the grammar is in Chomsky normal form （CNF)
- https://www.xarg.org/tools/cyk-algorithm/

CYK Algorithm

Assume that we have a grammar $\mathrm{G} = (\mathrm{V}, \mathrm{T}, \mathrm{S}, \mathrm{P})$ in Chomsky normal form and a string

$$\mathrm{w} = a_1 a_2 \cdots a_n .$$

We define substrings

$$w_{ij} = a_i \cdots a_j ,$$

and subsets of V

$$V_{ij} = \{ \mathrm{A} \in \mathrm{V} : \mathrm{A} \overset{*}{\Rightarrow} w_{ij} \} .$$

Clearly, $\mathrm{w} \in \mathrm{L}(\mathrm{G})$ if and only if $\mathrm{S} \in V_{1n}$.

CYK Algorithm

- To compute V_{ij} , observe that $A \in V_{ii}$ if and only if G contains a production $A \to a_i$.
- A derives wij if and only if there is a production $A \to BC$,with $B \overset{*}{\Rightarrow} w_{ik}$ and $C \overset{*}{\Rightarrow} w_{k+1,j}$ for some k with $i \leq k$, $k < j$.

$$V_{ij} = \bigcup_{\{k \in \{i, i+1, \ldots, j-1\}} \{A : A \to BC, \text{ with } B \in V_{ik}, C \in V_{k+1,j}\}$$

- An inspection of the indices in (6.8) shows that it can be used to computeall the Vij if we proceed in the sequence

  1. Compute V11, V22, ..., Vnn,
  2. Compute V12, V23, ..., Vn−1,n,
  3. Compute V13, V24, ..., Vn−2,n,

  and so on.

EXAMPLE 6.11

Determine whether the string $w = aabbb$ is in the language generated by the grammar

$$\begin{aligned} S &\to AB, \\ A &\to BB|a, \\ B &\to AB|b. \end{aligned}$$

First note that $w_{11} = a$, so V_{11} is the set of all variables that immediately derive a, that is, $V_{11} = \{A\}$. Since $w_{22} = a$, we also have $V_{22} = \{A\}$ and, similarly,

$$V_{11} = \{A\}, V_{22} = \{A\}, V_{33} = \{B\}, V_{44} = \{B\}, V_{55} = \{B\}.$$

Now we use (6.8) to get

$$V_{12} = \{A : A \to BC, B \in V_{11}, C \in V_{22}\}.$$

Since $V_{11} = \{A\}$ and $V_{22} = \{A\}$, the set consists of all variables that occur on the left side of a production whose right side is AA. Since there are none, V_{12} is empty. Next,

$$V_{23} = \{A : A \to BC, B \in V_{22}, C \in V_{33}\},$$

so the required right side is AB, and we have $V_{23} = \{S, B\}$. A straightforward argument along these lines then gives

$$\begin{aligned} V_{12} &= \varnothing, V_{23} = \{S, B\}, V_{34} = \{A\}, V_{45} = \{A\}, \\ V_{13} &= \{S, B\}, V_{24} = \{A\}, V_{35} = \{S, B\}, \\ V_{14} &= \{A\}, V_{25} = \{S, B\}, \\ V_{15} &= \{S, B\}, \end{aligned}$$

so that $w \in L(G)$.

END