

Forma Normal de Greibach

(1)

Uma gramática livre do contexto está na Forma Normal de Greibach se todas as suas produções são da forma:

$$A \rightarrow a\alpha$$

Na qual A é uma variável de V , a é um terminal de T , e α é uma palavra de V^* .

Portanto, a palavra vazia não pertence à linguagem gerada por uma gramática na FNG.

O algoritmo a seguir transforma uma GLC qualquer, cuja linguagem gerada não possua a palavra vazia, em uma gramática na FNG. O algoritmo é dividido nas seguintes etapas:

a) Etapa 1: Simplificação da gramática. Análoga à correspondente etapa do algoritmo referente à FNC.

b) Etapa 2: Renomeação das variáveis em uma ordem crescente qualquer. As variáveis da gramática são renomeadas em uma ordem crescente qualquer, como, por exemplo, A_1, A_2, \dots, A_n , em que n é o cardinal do conjunto de variáveis.

Diferentes critérios de renomeação podem resultar em diferentes gramáticas na FNG.

Entretanto, todas são equivalentes (geram a mesma linguagem);

②) Etapa 3: Transformação de produções para a forma $A_r \rightarrow A_s^{\alpha}$, em que $r \leq s$.
As produções são modificadas, garantindo que a primeira variável do lado direito é maior ou igual que a do lado esquerdo, considerando-se a ordenação da etapa anterior. As produções $A_r \rightarrow A_s^{\alpha}$ tais que $r > s$ são modificadas, substituindo-se a variável A_s pelas suas correspondentes produções ($A_s \rightarrow B_1 | \dots | B_m$), resultando em $A_r \rightarrow B_1^{\alpha} | \dots | B_m^{\alpha}$ e assim sucessivamente.
Entretanto, como o conjunto de variáveis é finito, existe um limite para as produções crescentes, que pode ser a geração de um terminal ($A_r \rightarrow \alpha$) ou de uma recursão ($A_r \rightarrow A_r$);

③) Etapa 4: Exclusão das Recursões da forma $A_r \rightarrow A_r^{\alpha}$. As recursões à esquerda podem existir originalmente na gramática, ou serem geradas pela etapa anterior. A eliminação da recursão à esquerda pode ser realizada, introduzindo-se variáveis auxiliares e incluindo-se recursão à direita ($B_r \rightarrow^{\alpha} B_r$);

④) Etapa 5: Um terminal no início do lado direito de cada produção. Após a execução da etapa anterior, todas as produções da forma $A_r \rightarrow A_s^{\alpha}$ são tais que $r < s$. Consequentemente, as produções da maior variável A_n só podem iniciar por um terminal no lado direito. Assim, se, em $A_{n-1} \rightarrow A_n^{\alpha}$, for substituído A_n pelas suas correspondentes produções (exemplo: $A_n \rightarrow \alpha\beta$), o lado direito das produções de A_{n-1} também

iniciará por um terminal (exemplo: $A_{n-1} \rightarrow a\beta_a^B$).
A repetição do algoritmo para A_{n-2}, \dots, A_1 resultará em produções exclusivamente da forma
 $A_r \rightarrow aa$.

f) Etapa 6: Produções na forma $A \rightarrow aa$ em que a é composta por variáveis. É análoga à correspondente etapa do algoritmo relativo à forma normal de Chomsky.

Definição do Algoritmo

Seja $G = (V, T, P, S)$ uma gramática livre do contexto, tal que $\epsilon \notin \text{GEN}(G)$ ou $\epsilon \notin L(G)$. O algoritmo para transformar na forma normal de Greibach é como segue:

Etapa 1: Simplificação da gramática. As seguintes simplificações:

- * Produções vazias;
- * Produções que substituem variáveis;
- * Símbolos inúteis (opcional);

devem ser realizadas, usando os algoritmos de simplificação apresentados anteriormente e resultando na gramática:

$$G_1 = (V_1, T_1, P_1, S)$$

Etapa 2: Renomeação das variáveis em uma ordem crescente qualquer. A gramática resultante desta etapa é (suponha que A_i corresponde à renomeação de S): ④

$$G_2 = (V_2, T_2, P_2, A_i)$$

Etapas 3 e 4: Transformações de produções para a forma $A_r \rightarrow A_s^\alpha$, na qual $r \leq s$ e exclusão das recursões da forma $A_r \rightarrow A_r^\alpha$. A gramática resultante destas duas etapas realizadas em conjunto é:

$$G_3 = (V_3, T_1, P_3, A_i)$$

Onde V_3, P_3 são construídos conforme o algoritmo abaixo, supondo que o cardinal de V_2 é n (a cada ciclo, suponha $B_r \notin V_3$).

$$P_3 = P_2$$

Para r variando de 1 até n

faca

para s variando de 1 até $r-1$

faca para toda $A_r \rightarrow A_s^\alpha \in P_3$

faca excluir $A_r \rightarrow A_s^\alpha$ de P_3 ;

para toda $A_s \rightarrow \beta \in P_3$

faca $P_3 = P_3 \cup \{A_r \rightarrow \beta\}$

para toda $A_r \rightarrow A_r^\alpha \in P_3$

faca excluir $A_r \rightarrow A_r^\alpha$ de P_3 ;

$V_3 = V_3 \cup \{B_r\}$;

$P_3 = P_3 \cup \{B_r \rightarrow \alpha\} \cup \{B_r \rightarrow \alpha B_r\}$;

recursão

para toda $A_r \rightarrow \phi \in P_3$ tal que ϕ não ⁽⁵⁾
inicia por A_r e
alguma $A_r \rightarrow A_r d$ foi
excluída.

faca $P_3 = P_3 \cup \{A_r \rightarrow \phi B_r\};$

Etapa 5: um terminal no inicio do lado direito de
cada produção. A gramática resultante desta
etapa é:

$$G_4 = (V_3, T_2, P_4, A_i)$$

onde P_4 é construído conforme o algoritmo ⁽²⁾ abaixo:

Também é necessário garantir que as produções
relativas às variáveis auxiliares B_r iniciam
por um terminal do lado direito, conforme o
algoritmo ⁽³⁾.

Algoritmo ⁽²⁾: etapa 5.1

$$P_4 = P_3;$$

para r variando de $n-1$ até 1 e toda $A_r \rightarrow A_s \alpha \in P_4$
faca excluir $A_r \rightarrow A_s \alpha$ de P_4 ;

para toda $A_s \rightarrow \beta$ de P_4 ,

faca $P_4 = P_4 \cup \{A_r \rightarrow \beta \alpha\};$

Algoritmo ⁽³⁾: etapa 5.2

para toda $B_r \rightarrow A_s \beta_r$

faca excluir $B_r \rightarrow A_s \beta_r$ de P_4 ;

para toda $A_s \rightarrow \alpha \alpha$

faca $P_4 = P_4 \cup \{B_r \rightarrow \alpha \alpha \beta_r\};$

Etapa 6: Produções na forma $A \rightarrow a^\alpha$ na qual α é composta por variáveis. É análoga à correspondente etapa do algoritmo relativo à FNC. (6)

Exemplo: Considere a GLC:

$G = (\{S, A\}, \{a, b\}, P, S)$ onde:

$$P = \{S \rightarrow AA \mid a, \\ A \rightarrow SS \mid b\}$$

Etapa 1: Simplificação da Gramática: já está simplificada.

Etapa 2: Renomeação das variáveis em uma ordem crescente qualquer. As variáveis S e A são renomeadas para A_1 e A_2 , respectivamente. As produções da gramática ficam como segue:

$$A_1 \rightarrow A_2 A_2 \mid a$$

$$A_2 \rightarrow A_1 A_1 \mid b$$

Etapa 3 e 4: Transformação de produções para a forma $A_r \rightarrow A_s \alpha$, na qual $r \leq s$ e exclusão das regras da forma $A_r \rightarrow A_r \alpha$. A produção $A_2 \rightarrow A_1 A_1$ necessita ser modificada. As produções da gramática ficam com segue:

$$A_2 \rightarrow A_1 A_1$$

$$r \leq s \text{ falso}$$

move $A_2 \rightarrow A_1 A_1$

substituição

$$A_2 \rightarrow B$$

$$A_2 \rightarrow B$$

$$A_1 \rightarrow A_2 A_2 \mid a$$

$$A_1 \rightarrow A_2 A_2 \mid a$$

$$A_2 \rightarrow \underline{A_2 A_2 A_1} \mid a A_1 \mid b$$

$$\boxed{\begin{array}{l} A_1 \rightarrow A_2 A_2 \\ A_r \rightarrow A_s \alpha \\ r \leq s < s = 2 \\ r \leq s \end{array}}$$

Mas $A_r \rightarrow A_r \alpha$
não é regra
deixa como está.

$$A_2 \rightarrow A_1 A_1$$

$$r \leq s \text{ falso}$$

$$r = 2 < s = 2$$

na mesma dificuldade.

A produção $A_2 \rightarrow A_2 A_2 A_1$ contém uma recursão. ⑦
 Portanto, é necessário incluir uma variável auxiliar B , como segue:

$$A_1 \rightarrow A_2 A_2 | \bar{a}$$

$$A_2 \rightarrow \bar{a} A_2 | b | \underline{\bar{a} A_2 B} | \underline{b B}$$

$$B \rightarrow A_2 A_1 | A_2 A_1 B$$

$$(A_2) \rightarrow (A_2) A_2 A_1 \rightarrow \text{recursão}$$

$$A_2 \rightarrow A_2 d \quad r=2$$

Exclui a produção $A_2 \rightarrow A_2 A_2 A_1$

Inclui o $B_r \rightarrow B_2$ em V_3 .

Inclui as produções em P_3 :

$$\begin{aligned} B_r &\rightarrow \bar{a} \\ B_r &\rightarrow \bar{a} B_r \end{aligned}$$

$$B_2 \rightarrow A_2 A_1 | A_2 A_1 B_2$$

$$\begin{aligned} A_2 &\rightarrow \bar{a} A_1 \\ A_1 &\rightarrow \emptyset \end{aligned}$$

$$A_2 \rightarrow \frac{\bar{a} A_1}{\emptyset} B_2$$

$$\begin{aligned} A_2 &\rightarrow \bar{b} \\ A_2 &\rightarrow \emptyset \end{aligned}$$

$$A_2 \rightarrow \bar{b} B_2$$

Etapas 5: um terminal no início do lado direito de cada produção. O lado direito das produções da maior variável A_2 inicia por um terminal. Substituindo-se A_2 no lado direito de $A_1 \rightarrow A_2 A_2$ pelas suas correspondentes derivações, as produções de A_1 também iniciarão por um terminal:

$$A_1 \rightarrow \bar{a} A_1 A_2 | \bar{b} A_2 | \bar{a} A_1 B A_2 | \bar{b} B A_2 | \bar{a}$$

$$A_2 \rightarrow \bar{a} A_1 | \bar{b} | \bar{a} A_1 B | \bar{b} B$$

$$B \rightarrow A_2 A_1 | A_2 A_1 B.$$

As produções referentes à variável B também são modificadas:

$$\begin{aligned} B \rightarrow & \bar{a} A_1 A_2 | \bar{b} A_1 | \bar{a} A_1 B A_2 | \bar{b} B A_1 | \bar{a} A_1 A_1 B | \bar{b} A_1 B | \\ & \bar{a} A_1 B A_1 B | \bar{b} B A_1 B \end{aligned}$$



Etapa 6: Produções na forma $A \rightarrow a^\alpha$ na qual α é composta por variáveis. ⑧

Nenhum procedimento é necessário, pois as produções já se encontram nessa forma.

A gramática resultante, na FNG, é a seguinte:

$G_{FNG} = (\{A_1, A_2, B\}, \{a, b\}, P_{FNG}, A_1)$, na qual:

$P_{FNG} = \{ A_1 \rightarrow aA_1A_2 \mid bA_2 \mid aA_1BA_2 \mid bBA_2 \mid a,$
 $A_2 \rightarrow aA_1 \mid b \mid aA_1B \mid bB,$
 $B \rightarrow aA_1A_1 \mid bA_1 \mid aA_1BA_1 \mid bBA_1 \mid aA_1A_2B \mid$
 $bA_2B \mid aA_1BA_2B \mid bBA_2B \}$

}

5.

Recurso à Esquerda

(9)

Em diversas situações, como no desenvolvimento de algoritmos reconhecedores, é desejável que a gramática que define a linguagem não seja recursiva à esquerda.

Recurso à esquerda é a ocorrência da situação:

$$A \Rightarrow^+ A\alpha$$

Uma variável deriva ela mesma, de forma direta ou indireta, como o símbolo mais à esquerda de uma subpalavra gerada.

A transformação de uma gramática em uma equivalente sem recurso à esquerda pode ser realizada, executando-se as quatro primeiras ~~etapas~~ etapas do algoritmo da FNG.

Etapa 1: Simplificação da gramática;

Etapa 2: Renomeação das variáveis em ordem crescente;

Etapa 3: Transformação de produções para a forma $A_r \rightarrow A_s \alpha$, na qual $r \leq s$;

Etapa 4: Exclusão das recursões da forma $A_r \rightarrow A_r \alpha$.