

Formas Normais

- Simplificações na gramática – definir um formato para as regras.
- Eliminação da Recursividade no Símbolo Inicial

Seja $G = (V, \Sigma, P, S)$. Existe $G' = (V', \Sigma, P', S')$ na qual:

1. $L(G) = L(G')$,
2. S' não é recursivo.

- Prova:
 - basta fazer $G' = (V \cup \{S'\}, \Sigma, P \cup \{S' \rightarrow S\}, S')$.

Formas Normais

➤ Exemplo:

$$S \rightarrow a S \mid A A \mid A C$$
$$A \rightarrow a A \mid \lambda$$
$$B \rightarrow b B \mid b S$$
$$C \rightarrow c C \mid \lambda$$

G'?

Eliminação de Produções Lambda

➤ Objetivo:

- evitar formas sentenciais com variáveis removidas por produções lambda.

➤ Exemplo:

$$\begin{aligned} S &\rightarrow S a B \mid a B \\ B &\rightarrow b B \mid \lambda \end{aligned}$$

$$S \Rightarrow SaB \Rightarrow SaBaB \Rightarrow aBaBaB \Rightarrow a\lambda aBaB \Rightarrow aa\lambda aB \Rightarrow aaa\lambda$$

Eliminação de Produções Lambda

- Variável anulável:
 - símbolo não-terminal que gera λ .
- Gramática não-contrátil:
 - não contém símbolos anuláveis,
 - a forma sentencial nunca é reduzida pela aplicação da produção.
- Gramática essencialmente não-contrátil:
 - único símbolo que pode ser anulável é o símbolo inicial.

Construção do Conjunto de Símbolos Anuláveis

Entrada: (V, Σ, P, S) . Saída: $C = \{ A \in V \mid A \xRightarrow{*} \lambda \}$

para cada produção $A \rightarrow \lambda$ de G inclua A em C

repita

para cada símbolo X de C faça

se existe uma produção $X \rightarrow w$ com $w \in C^*$ então

inclua X em C

fim se

fim para

até que nenhum símbolo seja incluído em C

Eliminação de Produções Lambda

1. elimine a possível recursão de S criando o símbolo inicial S' .
2. construa o conjunto C dos símbolos anuláveis da gramática.
3. se $S' \in C$ então crie a produção $S' \rightarrow \lambda$.
4. elimine toda produções da forma $A \rightarrow \lambda$ (exceto $S' \rightarrow \lambda$) ou $A \overset{*}{\Rightarrow} w \overset{*}{\Rightarrow} \lambda$.
1. para cada produção da forma $B \rightarrow uAw$, adicionar uma nova produção $B \rightarrow uw$.
2. repetir 4, 5, 6 caso alguma produção da forma $A \rightarrow \lambda$ tenha sido introduzida.

Eliminação de Regras de Cadeias

- Regra de cadeia:
 - regra da forma $A \rightarrow B$, $B \in V$.
- Cadeia:
 - derivação da forma $A_1 \Rightarrow A_2 \Rightarrow A_3 \Rightarrow \dots \Rightarrow A_n$, $n \geq 2$.
- Algoritmo:
 - entrada: G essencialmente não-contrátil.
 - eliminar toda produção da forma $A \rightarrow B$.
 - para cada regra da forma $B \rightarrow u$, adicionar a regra $A \rightarrow u$.

Eliminação de Regras de Cadeias

➤ Exemplo:

$$S \rightarrow A C A \mid C A \mid A A \mid A C \mid A \mid C \mid \lambda$$
$$A \rightarrow a A a \mid a a \mid B \mid C$$
$$B \rightarrow b B \mid b$$
$$C \rightarrow c C \mid c$$

Eliminação de Símbolos Inúteis

➤ Símbolo alcançável:

- x é alcançável se $S \xRightarrow{*} u x v$

➤ Símbolo útil:

- x é útil se $S \xRightarrow{*} u x v \xRightarrow{*} w, w \in \Sigma^*$.

➤ Símbolo Inútil:

- x é inútil se x não é alcançável, ou
- x não deriva um string de terminais.

Conjunto de Símbolos que Derivam String de Terminais

para cada produção $A \rightarrow w$ ($w \in \Sigma^*$) faça

 inclua A em T

fim para

repita

 para cada símbolo não terminal A faça

 se existe a produção $A \rightarrow w$, $w \in (\Sigma \cup T)^*$ então

 inclua A em T

 fim se

 fim para

até que nenhum símbolo seja incluído em T

Remoção de Símbolos que não Derivam Strings de Terminais

1. construa o conjunto T da gramática.
2. elimine as produções que contêm variáveis que não pertencem a T .
3. elimine as variáveis que não pertencem a T .
4. elimine de Σ os terminais que não aparecem em nenhuma produção.

Remoção de Símbolos que não Derivam Strings de Terminais

Exemplo:

$$S \rightarrow A C \mid B S \mid B$$
$$A \rightarrow a A \mid a F$$
$$B \rightarrow C F \mid b$$
$$C \rightarrow c C \mid D$$
$$D \rightarrow a D \mid B D \mid C$$
$$E \rightarrow a A \mid B S A$$
$$F \rightarrow b B \mid b$$

Conjunto de Símbolos Alcançáveis

inclua S em R

repita

para cada símbolo A de R faça

para cada produção $A \rightarrow w$ faça

inclua em R todos os não-terminais de w

fim para

fim para

até que nenhum símbolo seja incluído em R

Remoção de Símbolos Inúteis

1. elimine de G os símbolos que não derivam terminais.
2. construa o conjunto R dos símbolos alcançáveis.
3. elimine as produções que contêm símbolos que não pertencem a R .
4. elimine de V os símbolos que não pertencem a R .
5. elimine de Σ os símbolos que não aparecem em nenhuma produção.

Eliminação de Símbolos Inúteis

Exemplo:

$$S \rightarrow A C \mid B S \mid B$$
$$A \rightarrow a A \mid a F$$
$$B \rightarrow C F \mid b$$
$$C \rightarrow c C \mid D$$
$$D \rightarrow a D \mid B D \mid C$$
$$E \rightarrow a A \mid B S A$$
$$F \rightarrow b B \mid b$$

Remoção de Símbolos Inúteis

➤ Ordem:

1. remoção de símbolos que não geram terminais,
2. remoção de símbolos não alcançáveis.

A ordem é importante?

➤ Exemplo:

$$S \rightarrow a \mid A B$$
$$A \rightarrow b$$

Forma Normal de Chomsky

- Uma gramática $G = (V, \Sigma, P, S)$ está na forma normal de Chomsky se todas as suas produções são da forma:
 - $S \rightarrow \lambda$
 - $A \rightarrow a$
 - $A \rightarrow B C \quad B, C \in V - \{S\}$
- Nestas gramáticas toda árvore de parse é uma árvore binária.

Transformação para FN Chomsky

- 1) eliminar a recursão do símbolo inicial.
- 2) eliminar as transições lambda.
- 3) eliminar as regras de cadeias.
- 4) eliminar os símbolos inúteis.

Transformação para FN Chomsky

- 5) para cada produção $A \rightarrow w$
para cada terminal a de w faça
crie a produção $A' \rightarrow a$ (se esta produção não existir)
substitua a por A' em w
fim para
fim para
- 6) para cada produção $A \rightarrow w$ com $|w| > 2$ faça
 $w = B w'$
substitua a produção por $A \rightarrow B T$ e $T \rightarrow w'$
fim para

Transformação para FN Chomsky

➤ Exemplo:

$$S \rightarrow a A B C \mid a$$
$$A \rightarrow a A \mid a$$
$$B \rightarrow b b B \mid b c$$
$$C \rightarrow c C \mid c$$