

Teoria da Computação

Eliminação de Recursão à Esquerda

Prof. Jefferson Magalhães de Moraes

Eliminar Recursão à Esquerda

- Uma gramática $G = (N, T, P, S)$ tem recursão à esquerda se existe $A \in N$ tal que
 - $A \rightarrow A\alpha, \alpha \in (N \cup T)^*$
- Uma gramática $G = (N, T, P, S)$ tem recursão à direita se existe $A \in N$ tal que
 - $A \rightarrow \alpha A, \alpha \in (N \cup T)^*$
- A recursão é dita direta se a derivação acima for em um passo, isto é:
 - G tem recursão direta à esquerda se existe produção $A \rightarrow A\alpha \in P$
 - G tem recursão direta à direita se existe produção $A \rightarrow \alpha A \in P$

Eliminar a Recursão à Esquerda

- A importância de eliminar a recursividade à esquerda é que alguns tipos de compiladores podem executar o processo de reconhecimento como chamadas de rotinas (procedimentos ou funções). Assim, uma gramática recursiva à esquerda, tal como por exemplo $A \rightarrow Aa \mid a$, acaba gerando um laço infinito $A \Rightarrow Aa \Rightarrow Aaa \Rightarrow Aaaa \Rightarrow \dots$ e o processo de reconhecimento não finaliza nunca.

Eliminar a Recursão à Esquerda

Algoritmo:

1. recursões diretas: substituir cada regra

$A \rightarrow A\alpha_1 | A\alpha_2 | \cdots | A\alpha_n | \beta_1 | \beta_2 | \cdots | \beta_m$, onde nenhum β_i começa por A , por:

$$A \rightarrow \beta_1 A' | \beta_2 A' | \cdots | \beta_m A'$$

$$A' \rightarrow \alpha_1 A' | \alpha_2 A' | \cdots | \alpha_n A' | \varepsilon$$

Eliminar Recursão à esquerda

- Para Recursões indiretas:
 - Transformar produções com recursão indireta em recursão direta (fazer as substituições necessárias)

Eliminar Recursão à esquerda

- Exemplo: $G = (N, T, P, S)$

- $P: S \rightarrow A a$

- $A \rightarrow S b \mid c A \mid a$

- Solução:

- $P': S \rightarrow A a$

- $A \rightarrow A a b \mid c A \mid a$

- $P'': S \rightarrow A a$

- $A \rightarrow c A A' \mid a A'$

- $A' \rightarrow a b A' \mid \epsilon$