

Compiladores

Revisão de Linguagens Formais
Prof. Tales Viegas

<http://facebook.com/ProfessorTalesViegas>

Definições Indutivas

Proposição: elaborar uma gramática (conjunto de regras) que permita especificar todas as expressões possíveis com a utilização de a e b como operandos e '+', '*', '(' e ')' como operadores.

Solução:

1. a e b são expressões;
2. Se α e β são expressões, então as sequências $\alpha + \beta$ e $\alpha * \beta$ são expressões;
3. Se α é uma sequência de símbolos que é uma expressão, então (α) é uma expressão
4. Expressões são todas e somente as sequências de símbolos obtidas pelas aplicações das regras 1 a 3 um número finito de vezes.

Definições Indutivas

Uma definição mais concisa seria dada pela segunda solução:

Segunda Solução:

E = conjunto de expressões válidas;

1. a e b estão em E ;
2. Se α e β estão em E , então $\alpha + \beta$ e $\alpha * \beta$ estão em E ;
3. Se α está em E , então (α) está em E .

Obs: aqui aparece a necessidade de formalizar as soluções: necessidade de uma meta-linguagem que permita elaborar as regras da gramática que gera a linguagem desejada.

Forma de Backus Naur

Teríamos:

$E ::= a$

$E ::= b$

$E ::= E + E$

$E ::= E * E$

$E ::= (E)$

Ou:

$E ::= a$

| b

| $E + E$

| $E * E$

| (E)

Gramáticas

Uma gramática livre de contexto (GLC) é composta de quatro componentes:

- um conjunto de tokens, conhecidos como símbolos terminais;
- um conjunto de não-terminais;
- um conjunto de produções, usadas para definir as regras particulares à gramática em questão;
- um símbolo de partida para a utilização das regras.

Gramáticas

Formalmente as gramáticas podem ser caracterizadas como quádruplas ordenadas:

$$G = (N, T, P, S)$$

- ▶ **N** = vocabulário não terminal da gramática G (corresponde ao conjunto de todos os elementos simbólicos dos quais a gramática se vale para definir as leis de formação das sentenças da linguagem)
- ▶ **T** = vocabulário terminal contendo os símbolos ou átomos dos quais as sentenças da linguagem são constituídas. Dá-se o nome de terminais aos elementos de T.

Gramáticas

- ▶ P = conjunto de todas as leis de formação utilizadas pela gramática para definir a linguagem.
- ▶ Para tanto, cada construção parcial, representada por um não-terminal, é definida como um conjunto de regras de formação relativas à definição do não-terminal a ela referente.
- ▶ A cada uma destas regras de formação que compõem o conjunto P dá-se o nome de produção da gramática.
- ▶ Cada produção P tem a forma:
 $\alpha \Rightarrow \beta \quad \alpha \in (N \cup T)^+; \beta \in (N \cup T)^*$

onde α é, no caso geral, uma cadeia contendo no mínimo um não-terminal

Gramáticas

- ▶ **S** é um elemento de N, cuja propriedade ser o não-terminal que dá início ao processo de geração de sentenças. S é dito o símbolo inicial ou o axioma da gramática.

Gramáticas

No exemplo dado, teríamos:

$$N = \{E\}$$

$$T = \{a, b, +, *, (,)\}$$

P = regras já apresentadas para a definição

$$S = E$$

Obs: ao conjunto de cadeias geradas pela gramática dá-se o nome de linguagem.

Gramáticas Regulares

Gramáticas regulares são gramáticas onde as produções são apenas da forma:

$$A \rightarrow \alpha$$

:Gramática linear à direita

$$A \rightarrow \alpha A$$

ou

$$A \rightarrow \alpha$$

$$A \rightarrow A \alpha$$

:Gramática linear à esquerda

Gramáticas Regulares e GLC

- ▶ Toda *gramática regular* é uma *gramática livre de contexto*;
- ▶ As produções das *gramáticas regulares* são mais simples do que as produções das *gramáticas livres de contexto*;
- ▶ As *gramáticas livres de contexto* permitem a construção de *árvore sintáticas* mais complexas.

$$E \rightarrow T \mid E + T \mid E - T$$

$$T \rightarrow F \mid T^* F \mid T / F$$

$$F \rightarrow i \mid F^{**} i$$

A Gramática Regular equivalente seria:

$$E \rightarrow iM \mid i$$

$$M \rightarrow + E \mid - E \mid ^* E \mid / E \mid ^{**} E$$

Exercícios

1. Escreva uma gramática que reconheça identificadores iniciados por letra, podendo possuir depois zero ou mais dígitos ou letras.
2. Escreva uma Gramática Regular para reconhecimento de números binários
3. Escreva uma Gramática para reconhecer comentários em C ANSI (iniciados por /* e terminados por */)
4. Escreva uma Gramática para reconhecer números reais (Ex: 1212.121)