

# Compiladores

## BNF e Grafo Sintático

# Linguagem Livre de Contexto

- Em geral, uma linguagem de programação pertence a uma classe de linguagens chamadas de linguagens livres de contexto.
- Uma das maneiras de se definirem tais linguagens é por meio das gramáticas livres de contexto.

# Gramática

- Definição: Gramática

- É uma quádrupla ordenada  $G = (V, T, P, S)$ , na qual:
  - »  $V$ : conjunto finito de símbolos variáveis ou não-terminais
  - »  $T$ : conjunto finito de símbolos terminais
  - »  $P$ : conjunto finito de pares, denominados regras de produção tal que a primeira componente é a palavra de  $(V \cup T)^+$  e a segunda componente é palavra de  $(V \cup T)^*$
  - »  $S$ : elemento de  $V$  denominado variável inicial
- Regras de Produção: uma regra de produção  $(\alpha, \beta)$  é representada por  $\alpha \rightarrow \beta$ .

As regras de produção definem as condições de geração das palavras da linguagem.

# Gramática Livre de Contexto

- GLC é É uma quádrupla  $G = (\Omega, \Sigma, P, S)$ , na qual:
  - $\Sigma$  – é o alfabeto sobre a linguagem definida;
  - $\Omega$  – é um conjunto não vazio de símbolos não terminais;
  - $P$  – é um conjunto de produções da forma  $A \rightarrow \alpha$ , onde  $A \in \Omega$  e  $\alpha \in (\Sigma \cup \Omega)^*$ ;
  - $S$  – é o símbolo inicial da gramática,  $S \in \Omega$

# Exemplo de GLC

$G_1 = \langle \{S, A, B\}, \{a, b, c\}, S, P \rangle$ , onde  $P$  é:

$$(1) \quad S \rightarrow AB$$

$$(2) \quad A \rightarrow aAb$$

$$(3) \quad A \rightarrow \lambda$$

$$(4) \quad B \rightarrow cB$$

$$(5) \quad B \rightarrow \lambda$$

# Exemplo de GLC

Iniciamos com $S$	aplicamos (1)	obtemos $AB$
temos $AB$	aplicamos (2)	obtemos $aAbB$
temos $aAbB$	aplicamos (2)	obtemos $aaAbbB$
temos $aaAbbB$	aplicamos (3)	obtemos $aabbB$
temos $aabbB$	aplicamos (4)	obtemos $aabbcB$
temos $aabbcB$	aplicamos (4)	obtemos $aabbccB$
temos $aabbccB$	aplicamos (5)	obtemos $aabbcc$

# Forma de Backus-Naur (BNF)

- A **Forma de Backus-Naur** (BNF) é uma outra maneira de se definir linguagens livres de contexto.
- Ela é semelhante a uma gramática livre de contexto, mas permite que o lado direito das produções possua alguns operadores.
  - seleção
  - opcional
  - repetição 0 ou mais vezes
  - repetição 1 ou mais vezes

# BNF - Seleção

- $(\alpha \mid \beta)$  – Um dos elementos entre parênteses ( $\alpha$  ou  $\beta$ ) pode ser utilizado na aplicação da produção.

- $S \rightarrow a(b \mid c \mid d)e$

$$S \Rightarrow abe$$

- $S \Rightarrow ace$

$$S \Rightarrow ade$$

$$S \rightarrow abe$$

- $S \rightarrow ace$

$$S \rightarrow ade$$

# BNF – Seleção - Exemplo

•  $S \rightarrow (c(aSa \mid bSb)c \mid \lambda)$

$S \rightarrow \lambda$

•  $S \rightarrow caSac$

$S \rightarrow cbSbc$

•  $cacacbbcacac$

•  $S \Rightarrow caSac \Rightarrow cacaSacac \Rightarrow cacacbSbcacac \Rightarrow cacacbbcacac$

# BNF – Opcional

- $[\alpha]$  – O que estiver entre os colchetes pode ser utilizado ou não na aplicação da produção.

- $S \rightarrow a[bcd]e$

- $S \Rightarrow ae$

- $S \Rightarrow abcde$

# BNF – Opcional - Exemplo

$$S \rightarrow |a[(A | B)c]d$$

- $A \rightarrow a[a]$

$$B \rightarrow [b]$$

$$S \Rightarrow ad$$

$$S \Rightarrow aAcd \Rightarrow aacd$$

- $S \Rightarrow aAcd \Rightarrow aaacd$

$$S \Rightarrow aBcd \Rightarrow abcd$$

$$S \Rightarrow aBcd \Rightarrow acd$$

# BNF – Opcional - Exemplo

$S \rightarrow |a[(A | B)c]d$

•  $A \rightarrow a[a]$

$B \rightarrow [b]$

$S \rightarrow ad$

$S \rightarrow aAcd$

$S \rightarrow aBcd$

•  $A \rightarrow aa$

$A \rightarrow a$

$B \rightarrow b$

$B \rightarrow \lambda$

# Repetição – 0 ou mais vezes

- $(\alpha)^*$  – O que estiver entre parêntese pode ser usado um número qualquer de vezes na aplicação da produção e pode também não ser usado (repetido nenhuma vez).

- $S \rightarrow a(b)^*c$

- $\{ac, abc, abbc, abbbc, \dots\}$

- Como qualquer outra BNF, podemos encontrar uma

GLC correspondente, como, por exemplo:

$$\begin{array}{l} S \rightarrow aZc \\ Z \rightarrow bZ \\ Z \rightarrow \lambda \end{array}$$

# Repetição – 0 ou mais vezes - Exemplo

- $S \rightarrow a|(b | c)^*d$
- Gera todas as cadeias que iniciam com  $a$ , terminam com  $d$  e têm entre estes o  $\lambda$  ou qualquer cadeia formada por  $b$  e  $c$ .
- $\{ad, abd, acd, abbd, abcd, acbd, accd, abbbd, \dots\}$

# Repetição – 1 ou mais vezes

- $(\alpha)^+$  – O que estiver entre parêntese pode ser usado uma ou mais vezes na aplicação da produção. A produção  $\alpha \rightarrow \beta(\gamma)^+\delta$  é equivalente a  $\alpha \rightarrow \beta\gamma(\gamma)^*\delta$ .

- $S \rightarrow a(b)^+c$

- $\{abc, abbc, abbbc, \dots\}$

- Como qualquer outra BNF, podemos encontrar uma

GLC correspondente, como, por exemplo:

$$\begin{array}{l} S \rightarrow aZc \\ Z \rightarrow bZ \\ Z \rightarrow b \end{array}$$

# Repetição – 1 ou mais vezes - Exemplo

- $S \rightarrow a(b \mid c)^+d$
- Gera todas as cadeias que iniciam com  $a$ , terminam com  $d$  e têm entre estes qualquer cadeia formada por  $b$  e  $c$ .
- $\{abd, acd, abbd, abcd, acbd, accd, abbbd, \dots\}$

# BNF novamente

- $\langle program \rangle \rightarrow \{ \langle statlist \rangle \}$
- $\langle statlist \rangle \rightarrow \lambda \mid \langle statement \rangle \langle statlist \rangle$
- $\langle statement \rangle \rightarrow \langle assignment \rangle \mid \langle conditional \rangle \mid \langle loop \rangle$
- $\langle conditional \rangle \rightarrow \mathbf{if} \langle expr \rangle \{ \langle statlist \rangle \} \mathbf{else} \{ \langle statlist \rangle \} \mid \mathbf{if} \langle expr \rangle \{ \langle statlist \rangle \}$
- $\langle assignment \rangle \rightarrow \mathbf{ident} = \langle expr \rangle;$   
 $\langle loop \rangle \rightarrow \mathbf{while} \langle expr \rangle \{ \langle statlist \rangle \}$
- $\langle expr \rangle \rightarrow \mathbf{ident} \mid \mathbf{numero} \mid ((\langle expr \rangle)) \mid \langle expr \rangle \langle oper \rangle \langle expr \rangle$
- $\langle oper \rangle \rightarrow + \mid - \mid * \mid / \mid < \mid > \mid <= \mid >= \mid == \mid !=$

# Grafo Sintático

- Outra maneira de representar uma linguagem, chamada de grafo sintático. Essa representação facilita a visualização do tipo de cadeias ou formas sentenciais que cada não-terminal pode gerar.

- $A \rightarrow abc$

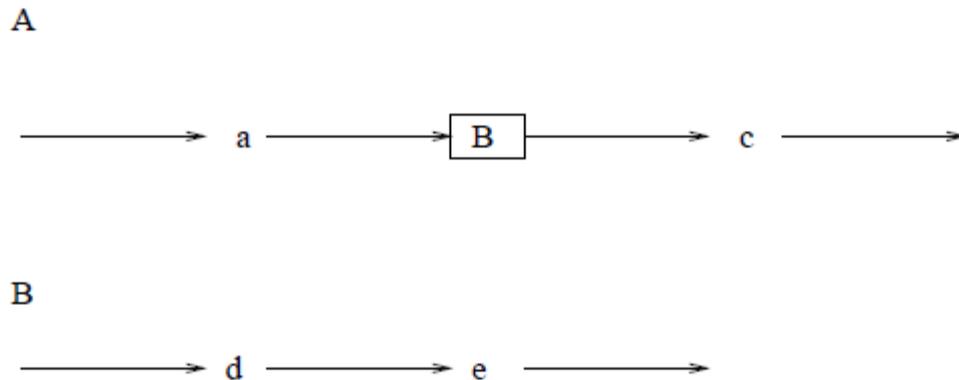


# Grafo Sintático

- Os nós correspondentes a símbolos não terminais são representados no diagrama por retângulos.

$$A \rightarrow aBc$$

$$B \rightarrow de$$

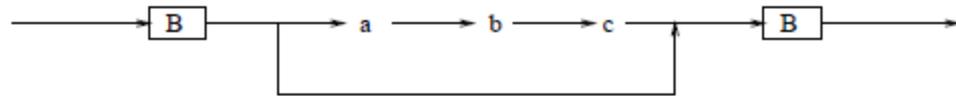


# Grafo Sintático

- Opcionais são uma seleção com o  $\lambda$ .

- $A \rightarrow B[abc]B$   
•  $B \rightarrow (de \mid \lambda)$

A



B



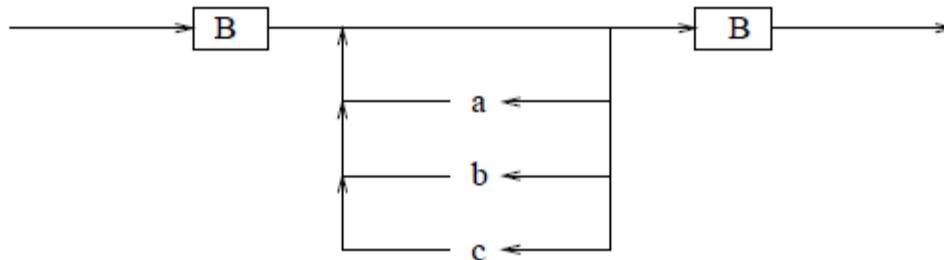
# Grafo Sintático

• E os operadores de repetição podem ser representados por meio de laços no grafo.

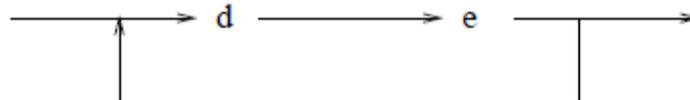
$$A \rightarrow B(a \mid b \mid c)^* B$$

$$B \rightarrow (de)^+$$

A



B



# Exercício

- Escrever na forma de grafo sintático

•  $\langle program \rangle \rightarrow \{ \langle statement \rangle^* \}$

•  $\langle statement \rangle \rightarrow \langle assignment \rangle \mid \langle conditional \rangle \mid \langle loop \rangle$

•  $\langle conditional \rangle \rightarrow$

**if**  $\langle expr \rangle \{ \langle statement \rangle^* \}$  [**else**  $\{ \langle statement \rangle^* \}$ ]

•  $\langle assignment \rangle \rightarrow$  **ident** =  $\langle expr \rangle$ ;

$\langle loop \rangle \rightarrow$  **while**  $\langle expr \rangle \{ \langle statement \rangle^* \}$

•  $\langle expr \rangle \rightarrow$  **ident** | **numero** | ( $\langle expr \rangle$ ) |

$\langle expr \rangle$   $\langle oper \rangle$   $\langle expr \rangle$

•  $\langle oper \rangle \rightarrow + \mid - \mid * \mid / \mid < \mid > \mid <= \mid >= \mid == \mid !=$