

# **Gramática Livre do Contexto**

## Linguagens Formais A

Prof. Giovani Rubert Libreotto

# Introdução

- O estudo de Linguagens Livres de Contexto pode ser abordado através dos formalismos:
  - operacional ou reconhecedor (Autômato com pilha)
  - axiomático ou gerador (Gramática Livre de Contexto)

# Histórico

- GLC foram inicialmente desenvolvidas por linguistas para representar linguagens naturais:

$\langle \text{frase} \rangle$	$\longrightarrow$	$\langle \text{grupo nominal} \rangle \langle \text{grupo verbal} \rangle$
$\langle \text{grupo nominal} \rangle$	$\longrightarrow$	$\langle \text{artigo} \rangle \langle \text{adjetivo} \rangle \langle \text{nome} \rangle$
$\langle \text{artigo} \rangle$	$\longrightarrow$	“o”
$\langle \text{adjetivo} \rangle$	$\longrightarrow$	“pequeno”
$\langle \text{nome} \rangle$	$\longrightarrow$	“príncipe”

# Histórico

- Descobriu-se que GLC não são adequadas para linguagens como o inglês.
- Na mesma época, cientistas da computação desenvolviam a notação de Backus e Naur (BNF), para descrever linguagens de programação, equivalente às GLC, com algumas abreviações.

# Classe das Linguagens Livres de Contexto

- Compreende um universo mais amplo de linguagens do que a classe das linguagens regulares
- Os algoritmos reconhecedores são relativamente simples e eficientes
  - Exemplos de aplicação: analisadores sintáticos, tradutores de linguagens e processadores de textos em geral.

# Relação entre as classes de linguagens

Universo de Todas as Linguagens

Linguagens Livre do Contexto

Linguagens Regulares

# Gramáticas Livres de Contexto (GLC)

- As linguagens livres de contexto tem uma grande importância para definir a sintaxe de linguagens de programação;
- As gramáticas livres de contexto são as gramáticas associadas às linguagens livres de contexto;

# Gramáticas Livres de Contexto (GLC)

- Estruturas de bloco (com *begin* e *end* associados) ou parênteses aninhados bem balanceados não podem ser escritas com linguagens regulares, e sim com linguagens livres de contexto;
- Os autômatos a pilha são as máquinas que permitem reconhecer as linguagens livres de contexto.

# Gramática Livre de Contexto

- É uma gramática  $G = (V, T, P, S)$ , com a restrição de que qualquer produção é da forma  $A \rightarrow \alpha$ , onde  $A$  é uma variável e  $\alpha$  pode conter variáveis ou terminais, em qualquer ordem.
- Uma linguagem é dita Linguagem Livre de Contexto (LLC) ou Tipo 2 se for gerada por uma Gramática Livre de Contexto.

# Exemplos de GLCs

- Duplo balanceamento:  $L_1 = \{ a^n b^n \mid n \geq 0 \}$ 
  - $G_1 = (\{ S \}, \{ a, b \}, P_1, S)$ ,  $P_1 = \{ S \rightarrow aSb \mid \epsilon \}$
  - Exemplo: geração da palavra aabb
    - $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aa\epsilon bb = aabb$

# Exemplos de GLCs

- O duplo balanceamento é um exemplo clássico no estudo das LLCs, pois permite a implementação de estruturas平衡eadas como:
  - a) blocos do tipo BEGIN – END
    - $S \rightarrow \text{begin } S \text{ end} \mid L \mid \epsilon$ , onde L gera uma lista de comandos por exemplo
  - b) Linguagens com parênteses平衡ados na forma ()
    - $S \rightarrow ( S ) \mid E \mid \epsilon$ , onde E gera uma expressão aritmética por exemplo

# Exemplos

- Expressões com operadores e operandos simples
  - $G2 = (\{ E \}, \{ +, *, (, ), x \}, P2, E)$ , onde:
  - $P2 = \{ E \rightarrow E+E \mid E^*E \mid (E) \mid x \}$
- a expressão  $(x+x)^*x$  pode ser gerada pela seguinte sequência de derivação:
  - $E \Rightarrow E^*E \Rightarrow (E)^*E \Rightarrow (E+E)^*E \Rightarrow (x+E)^*E \Rightarrow (x+x)^*E \Rightarrow (x+x)^*x$

# Exemplos

- Base para as gramáticas de operandos e expressões como:
  - $S \rightarrow S \text{ and } S \mid S \text{ or } S \mid \text{not } S \mid (S) \mid X$
  - $X \rightarrow E > E \mid E < E \mid E = E \mid E \neq E \mid \text{True} \mid \text{False}$
- OBS: as regras para o símbolo E são as definidas no slide anterior

# Outro Exemplo de Gramática

$\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{expr} \rangle$

|  $\langle \text{expr} \rangle - \langle \text{expr} \rangle$

|  $(\langle \text{expr} \rangle)$

|  $\langle \text{const} \rangle$

$\langle \text{const} \rangle \rightarrow \langle \text{const} \rangle \langle \text{const} \rangle | 0 | 1 | 2 | 3 | 4 | 5 |$   
 $6 | 7 | 8 | 9$

# Derivação

- Verificar se uma frase faz parte da linguagem gerada pela gramática, envolve sucessivas substituições da cadeia de símbolos que ocorre do lado esquerdo da produção pela sua construção sintática correspondente, partindo do símbolo inicial.

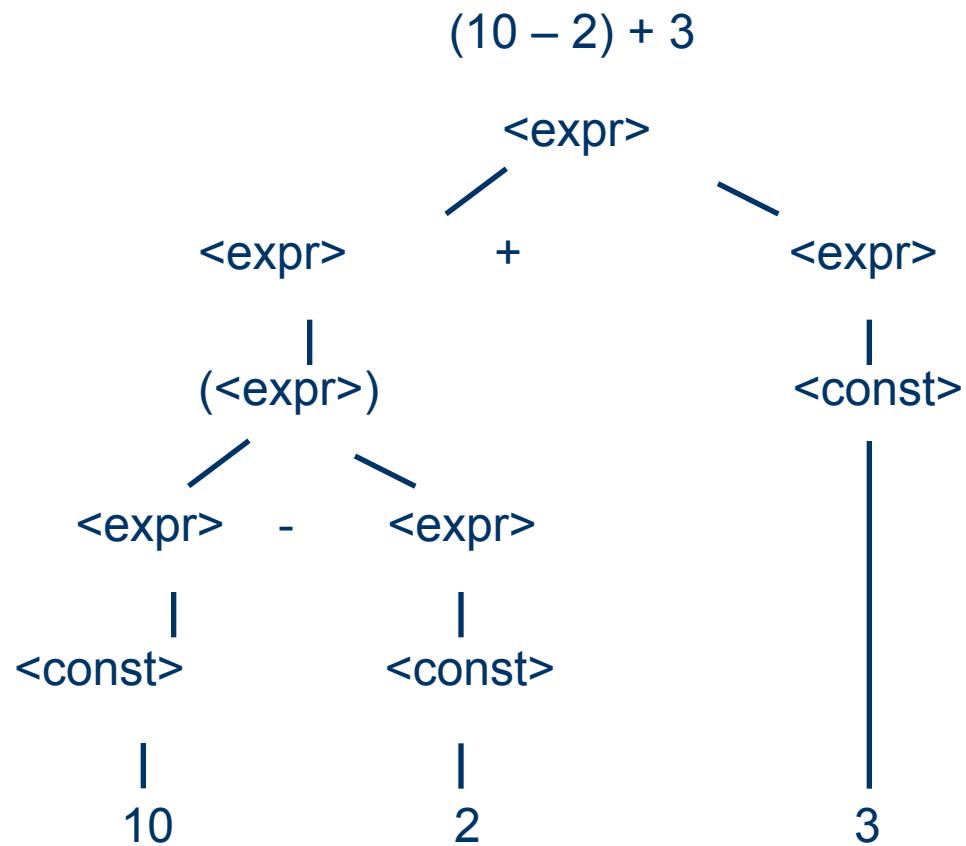
# Derivação

- Essa substituição é chamada derivação sendo normalmente denotada pelo símbolo  $\Rightarrow$ .

# Derivação

```
<expressão>
⇒ <expr> + <expr>
⇒ (<expr>) + <expr>
⇒ (<expr> - <expr>) + <expr>
⇒ (<const> - <expr>) + <expr>
⇒ (<const><const> - <expr>) + <expr>
⇒ (1<const> - <expr>) + <expr>
⇒ (10 - <expr>) + <expr>
⇒ (10 - <const>) + <expr>
...
⇒ (10 - 2) + 3
```

# Árvore Sintática

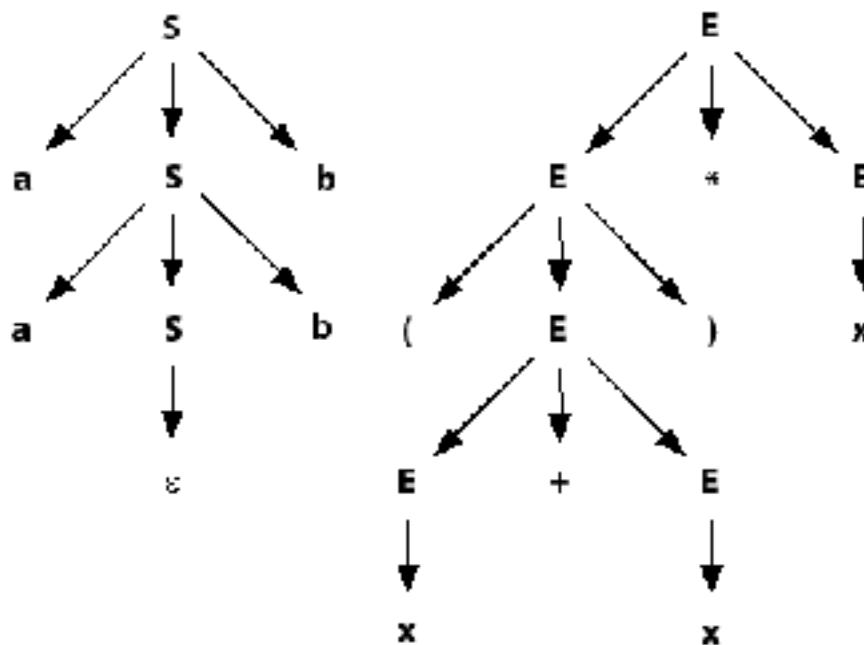


# Árvores de Derivação

- É conveniente representar a derivação de palavras em uma árvore de derivação, onde:
  - A raiz é o símbolo inicial da gramática;
  - Os vértices interiores obrigatoriamente são variáveis;
  - Se  $A$  é um vértice interior e  $X_1, X_2, \dots, X_n$  são os filhos de  $A$ , então  $A \rightarrow X_1X_2\dots X_n$  é uma produção da gramática.
  - Um vértice folha é um símbolo terminal ou o símbolo vazio  $\epsilon$  (neste caso,  $\epsilon$  é filho único)

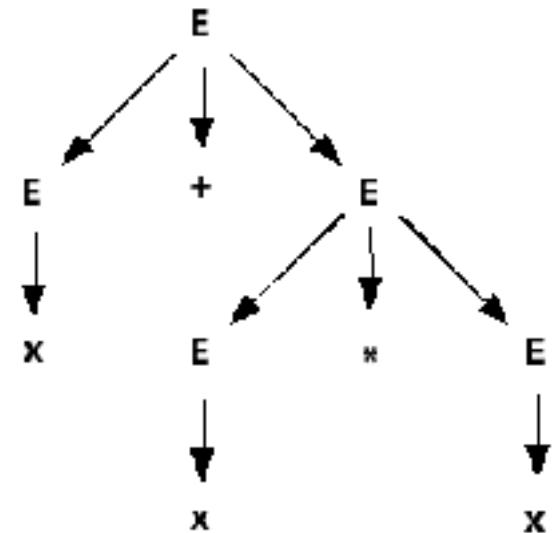
## Exemplo 3

- Árvores de derivação para as palavras aabb e  $(x+x)^*x$  (exemplos 1 e 2) :



## Exemplo 4

- Árvore de Derivação x Derivações
- Possíveis derivações da palavra  $x+x^*x$ :
  - a)  $E \Rightarrow E+E \Rightarrow x+E \Rightarrow x+E^*E \Rightarrow x+x^*E \Rightarrow x+x^*x$
  - b)  $E \Rightarrow E+E \Rightarrow E+E^*E \Rightarrow E+E^*x \Rightarrow E+x^*x \Rightarrow x+x^*x$
  - c)  $E \Rightarrow E+E \Rightarrow E+E^*E \Rightarrow x+E^*E \Rightarrow x+x^*E \Rightarrow x+x^*x$
  - d) etc...

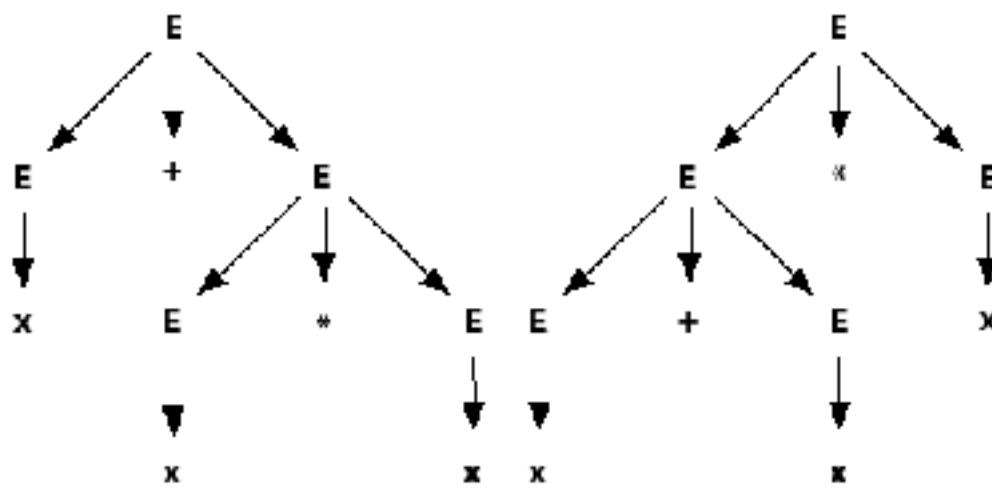


# Derivação mais à esquerda (ou direita)

- É a sequência de produção aplicada sempre à variável mais à esquerda (direita).
- No exemplo anterior, a representa uma derivação mais à esquerda, e b uma derivação mais à direita.

# Ambigüidade

- Uma Gramática Livre do Contexto é dita uma Gramática Ambígua, se existe uma palavra que possua duas ou mais árvores de derivação.



Considerando a gramática  
 $P_2 = \{ E \rightarrow E+E \mid E^*E \mid [E] \mid x \},$   
do exemplo anterior, a  
palavra  $x+x^*x$ :

# Ambiguidade

- A palavra  $x+x^*x$  possui mais de uma derivação à esquerda (direta):
  - a) Derivações mais à esquerda:
    - $E \Rightarrow E+E \Rightarrow x+E \Rightarrow x+E^*E \Rightarrow x+x^*E \Rightarrow x+x^*x$
    - $E \Rightarrow E^*E \Rightarrow E+E^*E \Rightarrow x+E^*E \Rightarrow x+x^*E \Rightarrow x+x^*x$
  - b) Derivações mais à direita:
    - $E \Rightarrow E+E \Rightarrow E+E^*E \Rightarrow E+E^*x \Rightarrow E+x^*x \Rightarrow x+x^*x$
    - $E \Rightarrow E^*E \Rightarrow E^*x \Rightarrow E+E^*x \Rightarrow E+x^*x \Rightarrow x+x^*x$

# **Gramática Livre do Contexto**

## Linguagens Formais A

Prof. Giovani Rubert Libreotto