

3 – Linguagens Livres de Contexto

Aula 12

Sumário

Capítulo 3 – Linguagens Livres de Contexto

3.1. Gramáticas Livres de Contexto

3.1.1. Árvore de Derivação

3.1.2. Ambiguidade

3.1.3. Simplificação de Gramáticas Livres de Contexto

3.1.4. Recursão à Esquerda

3.2. Autômato com pilha

3.3. Propriedades das Linguagens Livres de Contexto

Sumário

Capítulo 3 – Linguagens Livres de Contexto

3.1. Gramáticas Livres de Contexto

3.1.1. Árvore de Derivação

3.1.2. Ambiguidade

3.1.3. Simplificação de Gramáticas Livres de Contexto

3.1.4. Recursão à Esquerda

3.2. Autômato com pilha

3.3. Propriedades das Linguagens Livres de Contexto

Hierarquia de Chomsky

Tipo 0. Gramáticas com estruturas de frase

Tipo 1. Gram. Sensíveis ao contexto

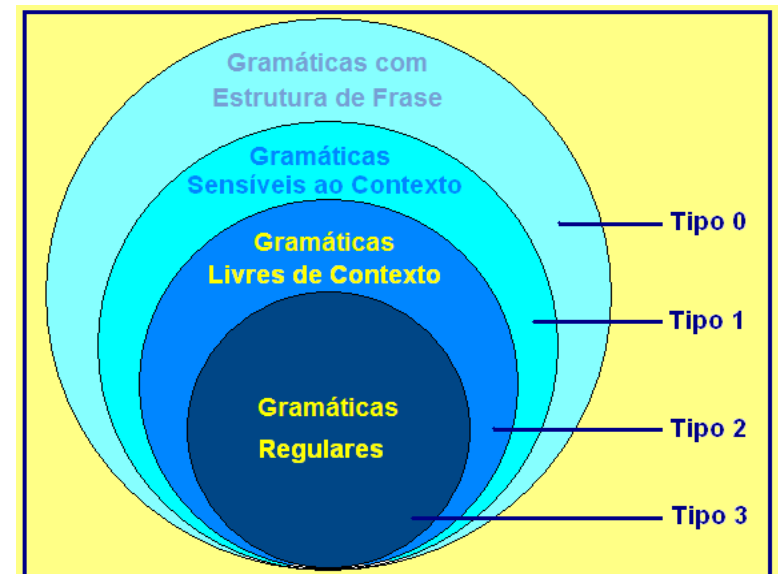
- Nenhuma das regras de produção pode **reduzir o comprimento** da forma sentencial que for substituída
- Se $\alpha \rightarrow \beta$ então $|\alpha| \leq |\beta|$

Tipo 2. **Gram. Livres de contexto**

- As regras tem apenas **uma Variável** do lado esquerdo
- **Não** pode ter **terminal** do lado **esquerdo**
- Ex:
 - $A \rightarrow \beta$
 - $Aa \rightarrow \beta$, **não pode**

Tipo 3. Gram. Regulares

- Deve ser **Linear à direita** ou **à esquerda**
- Ex:
 - $A \rightarrow aB \mid a$
 - $B \rightarrow Ba \mid a$
 - $A \rightarrow ABa$, **Não pode**



Linguagens Livres de Contexto

- Estudo da classe de linguagens tipo 2
- É de fundamental importância
 - Tem um universo mais abrangente que as LR
 - Entretanto é uma classe relativamente restrita
 - Fácil definir linguagens que não pertencem a ela
 - Tem melhor aplicabilidade às linguagens de programação
 - Analisadores sintáticos
 - Tradutores de linguagens
 - Processadores de texto em geral

Linguagens Livres de Contexto

- Tem melhor aplicabilidade às linguagens de programação
 - Analisadores sintáticos
 - Tradutores de linguagens
 - Processadores de texto em geral
 - Ex:
 - Parênteses balanceados
 - begin begin end end
 - {...}
 - ((...))
 - Blocos e estruturas de controle
 - If <cond> then if <cond> then ... else else

Linguagens Livres de Contexto

- Estudaremos duas abordagens (Formalismos):

1. Autômato com Pilha

- Formalismo operacional ou reconhecedor
- Análogo ao Autômato finito não-determinístico
- Memória auxiliar: PILHA
- Pode ser lida ou gravada

2. Gramática livre de contexto

- Formalismo axiomático ou gerador
- Gramática com restrições da forma das regras de produção

Gramática Livre de Contexto

- DEFINIÇÃO FORMAL:

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

- V conjunto finito de símbolos, **variáveis** ou **não-terminais**
- T conjunto finito de símbolos, **terminais**
- P **produções**
 - $(V \cup T)^+ \rightarrow (V \cup T)^+$
 - Qualquer regra é da forma $A \rightarrow \alpha$
 - **A** é variável de V
 - **α** é palavra de $(V \cup T)^*$
- O lado esquerdo é composto por uma única variável

Livre de contexto??

- Sendo nossas produções do tipo
 - $A \rightarrow \alpha$
- Em uma derivação qualquer, A deriva em α
 - Sem depender de qualquer análise
 - Não depende dos símbolos que *antecedem* ou que a *sucedem*, isto é “contexto”
 - Afinal regras desse tipo não pode: $Aa \rightarrow \beta$, *não pode*
 - Como não dependem, pode-se dizer que são “livres de contexto”

Gramática livre de contexto

- Seja a seguinte linguagem

$$L = \{ a^n b^n \mid n \geq 0 \}$$

- Esta linguagem é regular?

Gramática livre de contexto

- Seja a seguinte linguagem

$$L = \{ a^n b^n \mid n \geq 0 \}$$

- Esta linguagem é regular?
- *R: Pelo teorema que vimos no capítulo anterior, se conseguirmos criar um Automato para esta linguagem ela será regular*
- Chamamos esta linguagem de duplo balanceamento
 - Muito utilizado em linguagens de programação

Gramática livre de contexto

- Seja a seguinte linguagem

$$L = \{ a^n b^n \mid n \geq 0 \}$$

$$G = (\{S\}, \{a,b\}, P, S)$$

GLC

$S \rightarrow aSb$

$S \rightarrow \varepsilon$

Gramática livre de contexto

- Seja a seguinte linguagem

$$L = \{ a^n b^n \mid n \geq 0 \}$$

$$G = (\{S\}, \{a,b\}, P, S)$$

GLC

$S \rightarrow aSb$

$S \rightarrow \varepsilon$

- Derivação da palavra aabb

$$S \rightarrow aSb \rightarrow aaSbb \rightarrow aabb$$

Gramática livre de contexto

- Expressões Aritméticas

$L = \{\text{expressões aritméticas com colchetes balanceados, dois operadores e um operando}\}$

- $G = (\{ E \}, \{ +, *, [,], x \}, P, E)$
- Derivação da palavra $[x + x] * x$

GLC

$E \rightarrow E + E$

$E \rightarrow E * E$

$E \rightarrow [E]$

$E \rightarrow x$

$E \Rightarrow E * E \Rightarrow [E] * E \Rightarrow [E + E] * E$

$\Rightarrow [x + E] * E \Rightarrow [x + x] * E \Rightarrow [x + x] * x$

Gramática livre de contexto

- Expressões Aritméticas

$L = \{\text{expressões aritméticas com colchetes balanceados, dois operadores e um operando}\}$

- $G = (\{ E \}, \{ +, *, [,], x \}, P, E)$
- Derivação da palavra $[x + x] * x$

GLC

$E \rightarrow E + E$

$E \rightarrow E * E$

$E \rightarrow [E]$

$E \rightarrow x$

$$\begin{aligned} E &\Rightarrow E * E \Rightarrow [E] * E \Rightarrow [E + E] * E \\ &\Rightarrow [x + E] * E \Rightarrow [x + x] * E \Rightarrow [x + x] * x \end{aligned}$$

- Existe outras formas de fazer essa derivação? Quais?

Sumário

Capítulo 3 – Linguagens Livres de Contexto

3.1. Gramáticas Livres de Contexto

3.1.1. *Árvore de Derivação*

3.1.2. Ambiguidade

3.1.3. Simplificação de Gramáticas Livres de Contexto

3.1.4. Recursão à Esquerda

3.2. Autômato com pilha

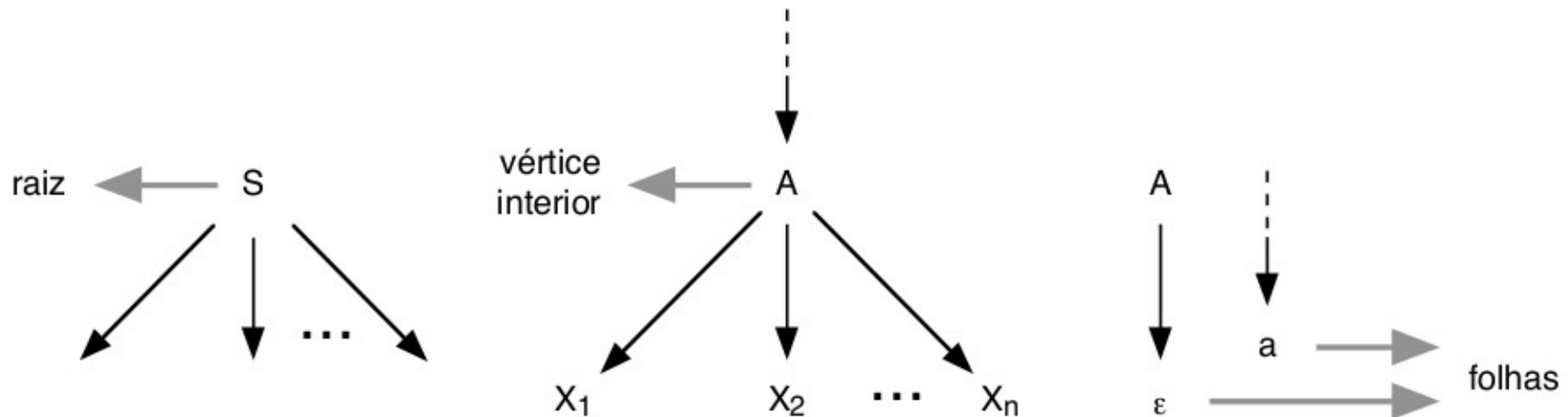
3.3. Propriedades das Linguagens Livres de Contexto

Árvore de derivação

- Árvore de derivação
 - representa a derivação de uma palavra na forma de árvore
 - parte do símbolo inicial como a **raiz**
 - termina em símbolos terminais como **folhas**
- Conveniente em muitas aplicações
 - Compiladores
 - Processadores de texto

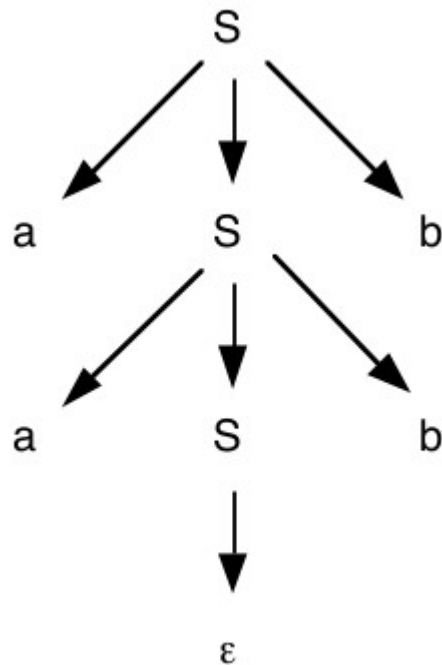
Árvore de derivação

- Definição:
 - Raiz**: símbolo inicial
 - Vértices** interiores: variáveis
 - se A é um vértice interior e X_1, X_2, \dots, X_n são os "filhos" de A
 - $A \rightarrow X_1, X_2, \dots, X_n$ é uma produção da gramática
 - X_1, X_2, \dots, X_n são ordenados da esquerda para a direita
 - Vértice folha ou folha**: terminal ou o símbolo vazio
 - se vazio: único filho de seu pai ($A \rightarrow \varepsilon$)



Árvore de derivação

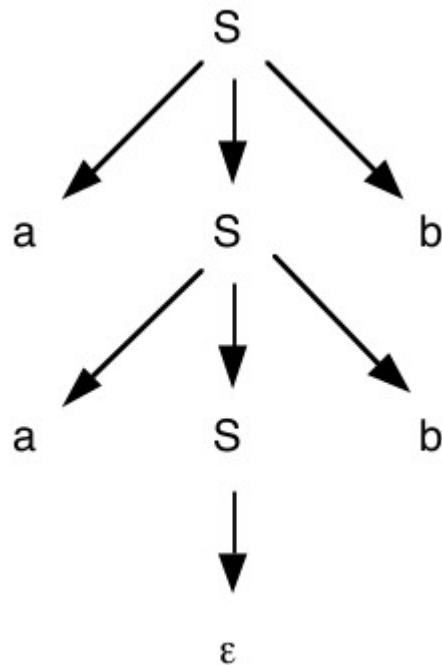
- Exemplo 1
 - Derivando **aabb**



Árvore de derivação

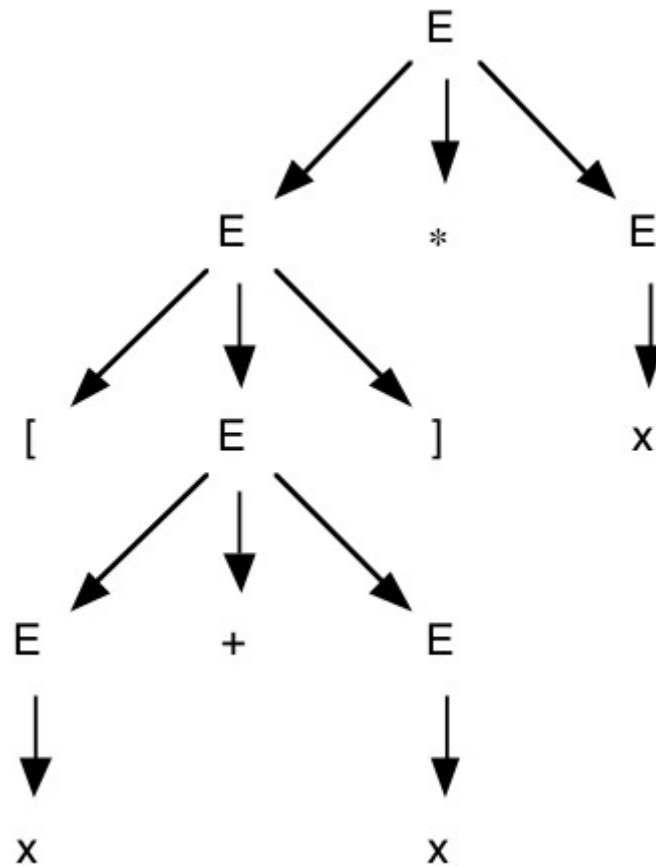
- Exemplo 1

- Derivando **aabb**



- Exemplo 2

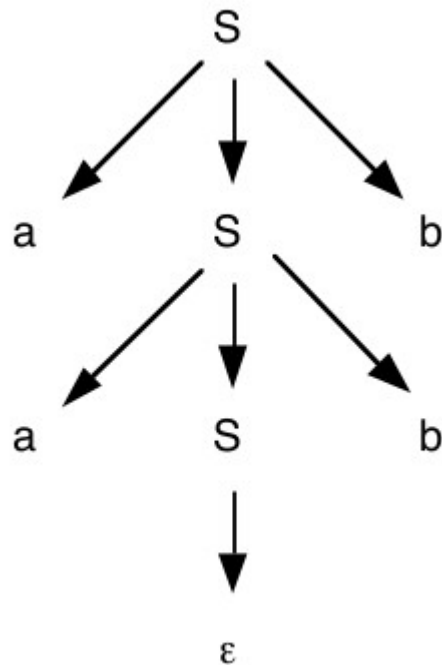
- Derivando **$[x + x] * x$**



Árvore de derivação

- Exemplo 1

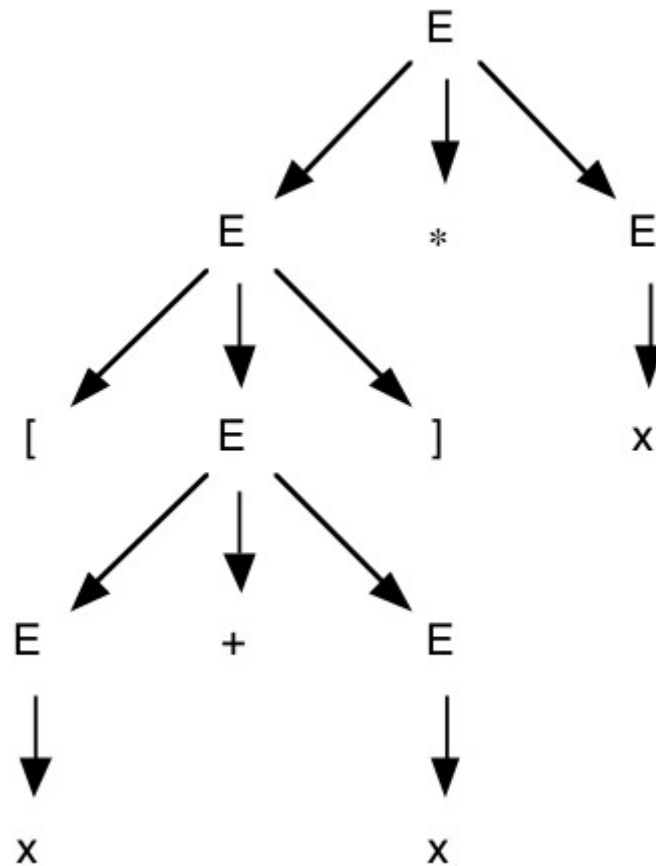
- Derivando **aabb**



Note que podemos fazer derivações **DISTINTAS** para uma mesma **PALAVRA**

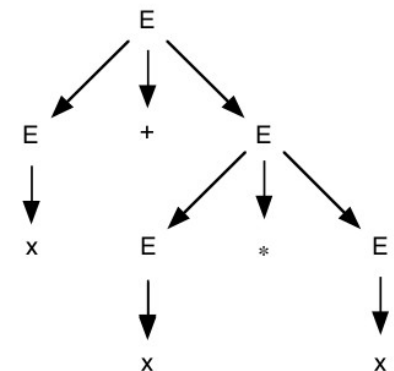
- Exemplo 2

- Derivando **$[x + x] * x$**



Árvore de derivação

- Uma derivação pode ser classificada de acordo com a forma de aplicação
 - Mais a direita, ou
 - Mais a esquerda
- Derivação mais a direita
 - Sequencia de produções aplicada sempre à variável mais a **direita** da palavra
 - Ex: $x+x*x$
 - $E \Rightarrow E+E \Rightarrow E+E*E \Rightarrow E+E*x \Rightarrow E+x*x \Rightarrow x+x*x$
- Derivação mais a esquerda
 - Sequencia de produções aplicada sempre à variável mais a **esquerda** da palavra
 - Ex: $x+x*x$
 - $E \Rightarrow E*E \Rightarrow E+E*E \Rightarrow x+E*E \Rightarrow x+x*E \Rightarrow x+x*x$



Sumário

Capítulo 3 – Linguagens Livres de Contexto

3.1. Gramáticas Livres de Contexto

3.1.1. Árvore de Derivação

3.1.2. Ambiguidade

3.1.3. Simplificação de Gramáticas Livres de Contexto

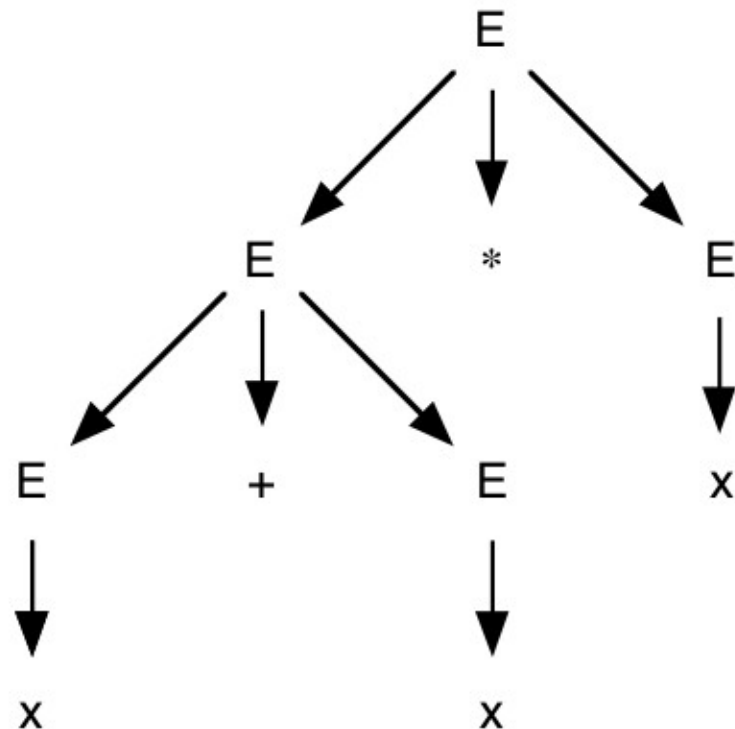
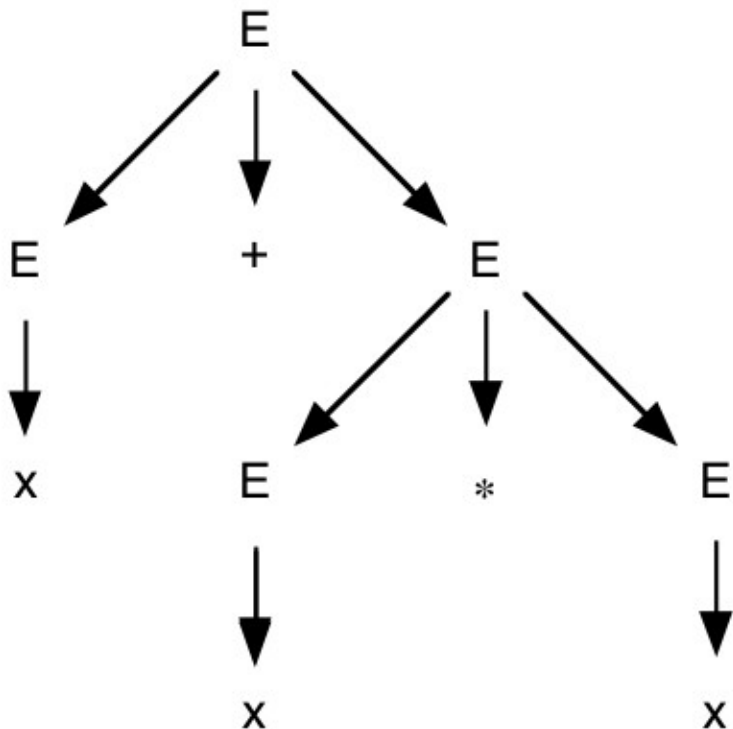
3.1.4. Recursão à Esquerda

3.2. Autômato com pilha

3.3. Propriedades das Linguagens Livres de Contexto

GLC Ambígua

- Definição: Uma gramática ambígua é ambígua se existe uma palavra associada a *duas ou mais árvores* de derivação
- Exemplo:
 - Seja a palavra $x + x * x$



GLC Ambígua

- É possível construir mais de uma derivação a direita ou esquerda
- Seja a palavra $x + x * x$
- *À esquerda*
 - $E \Rightarrow E + E \Rightarrow x + E \Rightarrow x + E * E \Rightarrow x + x * E \Rightarrow x + x * x$
 - $E \Rightarrow E * E \Rightarrow E + E * E \Rightarrow x + E * E \Rightarrow x + x * E \Rightarrow x + x * x$
- *À direita*
 - $E \Rightarrow E + E \Rightarrow E + E * E \Rightarrow E + E * x \Rightarrow E + x * x \Rightarrow x + x * x$
 - $E \Rightarrow E * E \Rightarrow E * x \Rightarrow E + E * x \Rightarrow E + x * x \Rightarrow x + x * x$

GLC Ambígua

- Seja a gramática G

Gramática

$S \rightarrow \mathbf{aABe}$

$A \rightarrow \mathbf{Abc} \mid \mathbf{b}$

$B \rightarrow \mathbf{d}$

Esta gramática é ambígua?

GLC Ambígua

- Qual das derivações são válidas para a gramática **G**?

Gramática

$S \rightarrow aXa$
 $X \rightarrow \varepsilon \mid bY$
 $Y \rightarrow \varepsilon \mid cXc \mid d$

Derivação 1

$S \rightarrow$
 $axa \rightarrow$
 aa

Derivação 2

$S \rightarrow$
 $axa \rightarrow$
 $abYa \rightarrow$
 $acXca \rightarrow$
 $acca$

Derivação 3

$S \rightarrow$
 $axa \rightarrow$
 $abYa \rightarrow$
 $abcXcda \rightarrow$
 $abccda$

Derivação 4

$S \rightarrow$
 $axa \rightarrow$
 $abYa \rightarrow$
 $abcXca \rightarrow$
 $abcbYcaabc \rightarrow$
 $bdca$

Esta gramática é ambígua?

GLC Ambígua

- Exercício 3.5, livro. Seja a gramática **G** abaixo:
- $G = (\{S\}, \{a, b\}, P, S)$

GLC
$S \rightarrow SS \mid aSa \mid bSb \mid \epsilon$

- Qual a linguagem gerada?
- Esta gramática é ambígua?
- Para a palavra aabbaaaa
 - Construa uma árvore de derivação
 - Para a árvore construída, determine a derivação mais à esquerda e mais à direita

GLC Ambígua

- Seja a gramática **G** abaixo:

GLC

$\text{expr_list} \rightarrow \text{expr_list expr_ptv} \mid \text{expr_ptv};$

$\text{expr_ptv} \rightarrow \text{expr PTVIRG};$

$\text{expr} \rightarrow \text{expr MAIS expr} \mid \text{expr MENOS expr} \mid \text{INTEIRO};$

- Faça algumas derivações;
- Esta gramática é ambígua?