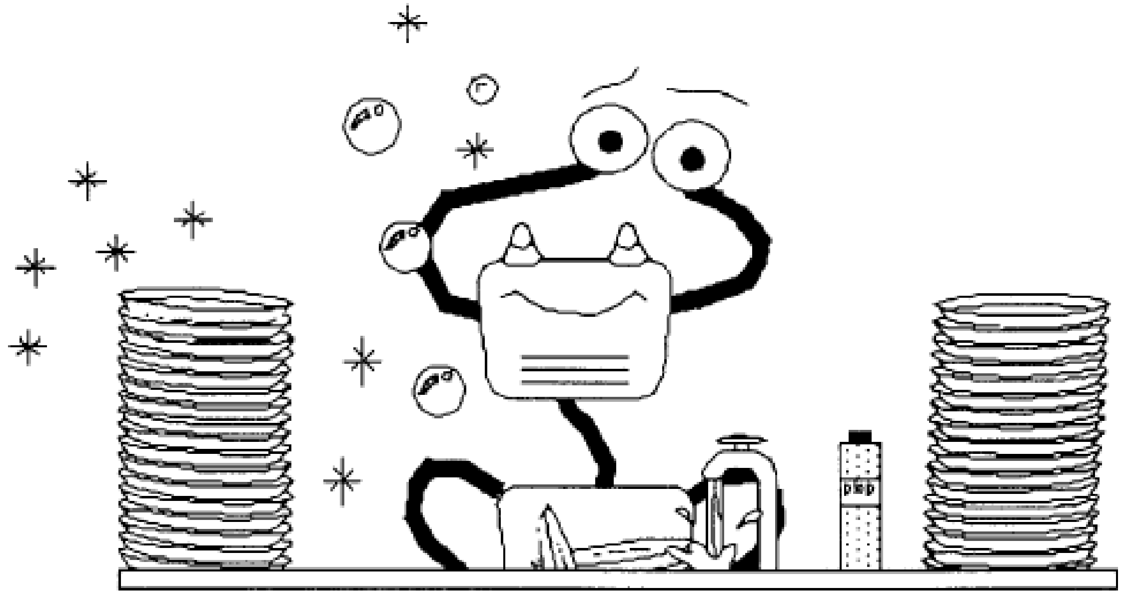
Correção da prova!

Parte II – Linguagens Livre do contexto



Previously on Computation Theory...

- Descrevemos linguagens através de autômatos finitos e expressões regulares: formas diferentes, mas equivalentes.
- Sabemos que algumas linguagens não são possíveis de serem descritas por eles, p.e. $\{0^n1^n \mid n >= 0\}$ (ver Lema do bombeamento, aula 06).

Introdução

- Gramática livre do contexto: método mais poderoso de descrição de linguagens.
- Estruturas recursivas.
- LLC são simples e eficientes.
- No estudo de linguagens humanas: GLC's podem captar aspectos importantes da relação entre substantivos, verbos e preposições.

Introdução

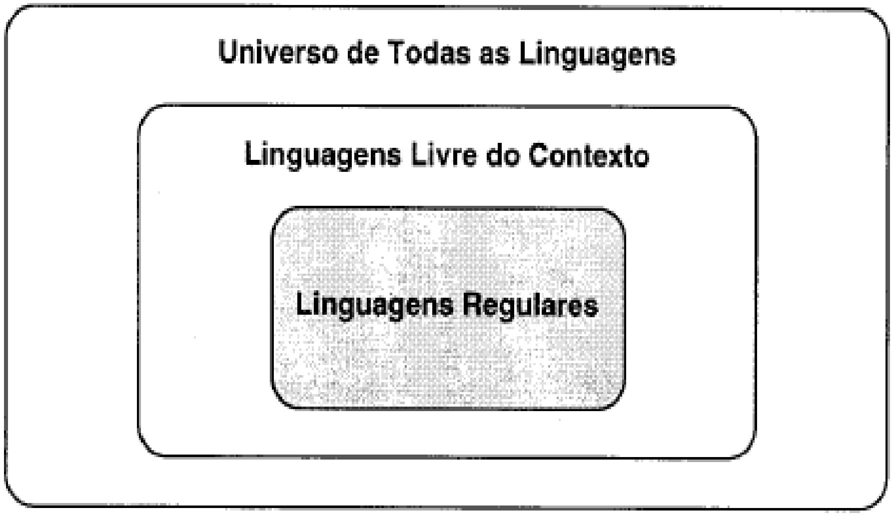
- Aplicação: especificação e compilação de linguagens de programação.
  - Elaborar uma grámatica para a linguagem: primeiro passo para um compilador e interpretador.
  - ***Parser***: extrai o significado do programa para gerar o código compilado ou interpreta a execução.
  - GLC's facilitam a produção do *parser*.

Introdução

- Coleção de linguagens associadas à gramáticas livre do contexto são **linguagens livres do contexto (llc).**
- LLC's são definidas a partir de um formalismo gerador (gramática) e um reconhecedor (autômato), como:
  - GLC: onde as regras de produção são definidas mais livremente que em GR.
  - Autômato com pilha (*pushdown automata*): estrutura básica e análoga ao AF, incluindo uma memória auxiliar tipo pilha (pode ser lida ou gravada) e a facilidade do não-determinismo.

Gramática livre do contexto

- Gramáticas consistem em uma coleção de regras de substituição (produção).
- GLC é uma 4-tupla G = (V,T,P,S) com a restrição de conter, no lado esquerdo da produção, exatamente uma variável.



Gramática livre do contexto

- "Livre do contexto" refere-se à possibilidade de representar a mais geral classe de linguagens cuja produção é da forma $A \rightarrow a$.
  - A variável A deriva a sem depender (livre) de qualquer análise de símbolos que o precedem ou sucedem A (contexto) na palavra que está sendo derivada.
- Uma LR é uma LLC.

Gramática livre do contexto

- Exemplo: Duplo balanceamento.

$$L1 = \{a^n b^n \mid n >= 0\}$$

a $G1 = (\{S\},\{a,b\},P=\{S \rightarrow aSb,\ S \rightarrow \varepsilon\},S)$ pode produzir aaabbb, por exemplo.

- Esse exemplo é importante pois permite fazer analogia para problemas do tipo:
  - Linguagens bloco-estruturadas $begin^n end^n$
  - Linguagens com parênteses balanceados $(^n)^n$

Gramática livre do contexto

- Exemplo: Expressões aritméticas.

  $G2 = (\{E\}, \{+,*,[,],x\}, P2, E)$

  $P2 = \{E \rightarrow E+E \mid E*E \mid [E] \mid x\}$

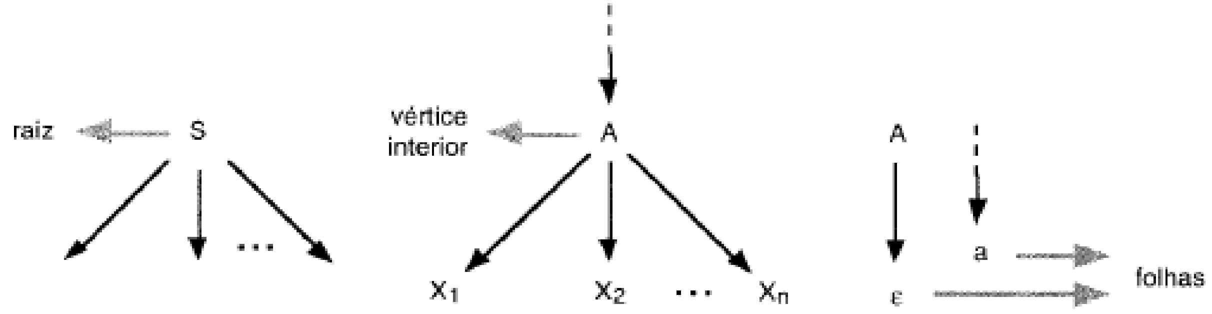
  G2 permite gerar [x+x]*x:

  $E \rightarrow E*E \rightarrow [E]*E \rightarrow [E+E]*E \rightarrow [x+E]*E \rightarrow [x+x]*E \rightarrow [x+x]*x$.

- É possível gerar a mesma expressão com outra sequência de derivação?

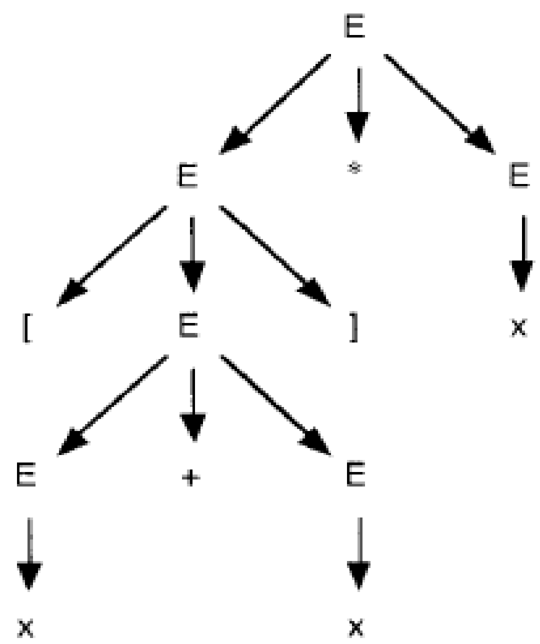
Árvore de derivação

- Derivação de palavras na forma de árvore.
  - A **raiz** é o símbolo inicial da gramática.
  - Os **vértices interiores** obrigatoriamente são variáveis.
  - Um vértice folha é um símbolo terminal, ou o símbolo vazio. O símbolo vazio é o **único** filho de seu pai.



Árvore de derivação

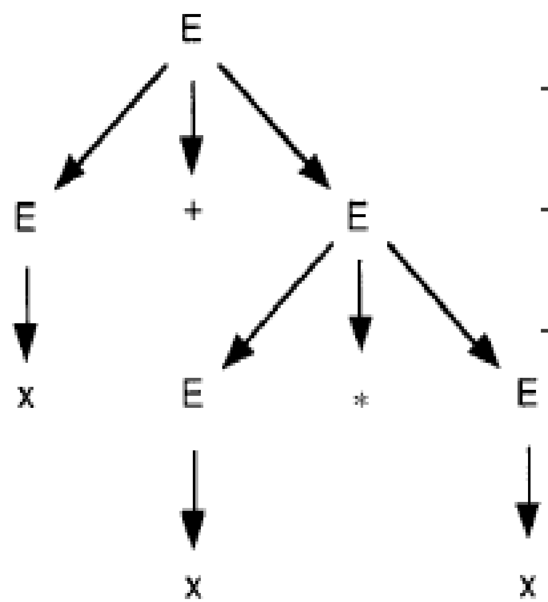
- Exemplo: Voltando à expressão aritmética [x+x]*x.



Árvore de derivação

- Exemplo: Árvore de derivação x Derivações

palavra = x+x*x



- E → E+E

→ x+E → x+E*E → x+x*E → x+x*x

→ E+E*E → E+E*x → E+x*x → x+x*x

→ E+E*E → x+E*E → x+x*E → x+x*x

...

Árvore de derivação

- **Derivação mais à esquerda (direita)**
  - É a sequência de produção aplicada sempre à variável mais à esquerda (direita)

Mais à esquerda:

$$E \rightarrow E+E \rightarrow x+E \rightarrow x+E*E \rightarrow x+x*E \rightarrow x+x*x$$

Mais à direita:

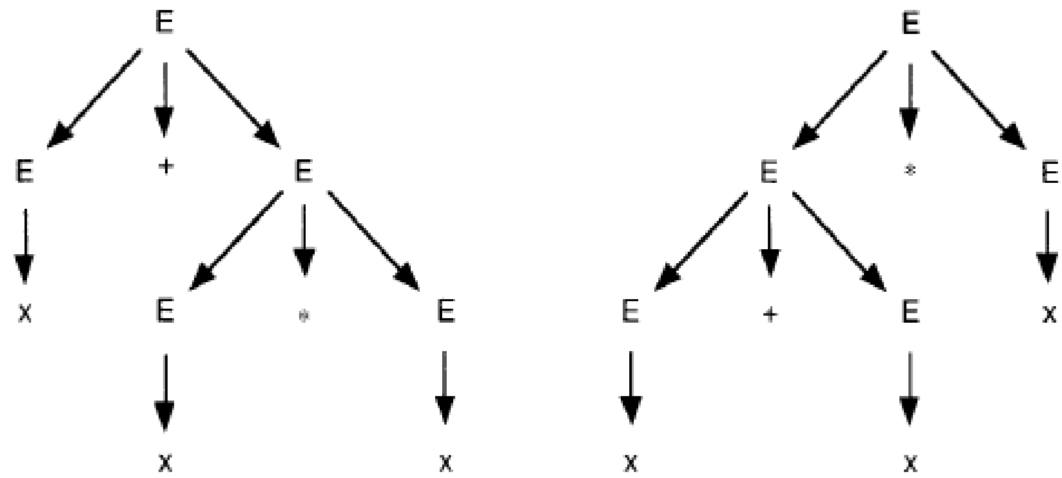
$$E \rightarrow E+E \rightarrow E+E*E \rightarrow E+E*x \rightarrow E+x*x \rightarrow x+x*x$$

Ambiguidade

- Uma mesma palavra associada a duas ou mais árvores de derivação.
- Para algumas aplicações de desenvolvimento e otimização de alguns algoritmos de reconhecimento é conveniente que a gramática não seja ambígua.
- Nem sempre é possível eliminar a ambiguidade.
- É mais fácil definir linguagens para as quais qualquer GLC é ambígua.

Ambiguidade

- **Definição**: uma GLC é dita uma *Gramática ambígua*, se existe uma palavra que possua duas ou mais árvores de derivação.
- Exemplo: a palavra x+x*x pode ser gerada por árvores distintas, portanto é ambígua.



Ambiguidade

- **Definição**: Uma linguagem é uma *Linguagem Inerentemente Ambígua* se qualquer GLC que a define é ambígua.
- Exemplo: $L3 = \{w \mid w = a^n b^n c^m d^m, n>=1, m>=1\}$
  $L4 = \{w \mid w = a^n b^m c^m d^n, n>=1, m>=1\}$

Simplificando GLC

- É possível simplificar algumas produções sem reduzir o poder de geração das GLC.
- São simplificações:
  - Exclusão de variáveis ou terminais não-usados para gerar palavras.
  - Exclusão de produções vazias na forma $A \rightarrow \varepsilon$ (se a palavra vazia pertence à linguagem, é incluída uma produção vazia específica para tal fim).
  - Exclusão de produções na forma $A \rightarrow B$.

Simplificando GLC

- Símbolos inúteis: Exclui-se as produções que fazem referência a estes símbolos, e os próprios símbolos.
  - Etapa 1: garante que qualquer variável gere terminais.
  - Etapa 2: garante que qualquer símbolo é atingível a partir do símbolo inicial.
- A ordem das etapas é rígida.

Simplificando GLC

- Exemplo: $G = (V,T,P,S) = (\{S,A,B,C\}, \{a,b,c\},P,S)$, onde $P = \{S \to aAa \mid bBb,\ A \to a|S,\ C \to c\}$
- E1 (qualquer variável gera palavra de terminais):

$V_1 = \varnothing;$

repita

$\quad V_1 = V_1 \cup \{ A \mid A \to \alpha \in P \text{ e } \alpha \in (T \cup V_1)^* \}$

até que o cardinal de V_1 não aumente;

iteração	variáveis
início	$\varnothing$
1	$\{A, C\}$
2	$\{A, C, S\}$
3	$\{A, C, S\}$

Simplificando GLC

- Exemplo: $G = (V,T,P,S) = (\{S,A,B,C\}, \{a,b,c\},P,S)$, onde $P = \{S \to aAa \mid bBb,\ A \to a|S,\ C \to c\}$
- E1 (qualquer variável gera palavra de terminais):

$V_1 = \varnothing$;
repita
    $V_1 = V_1 \cup \{ A \mid A \to \alpha \in P \text{ e } \alpha \in (T \cup V_1)^* \}$
até que o cardinal de V_1 não aumente;

iteração	variáveis
início	$\varnothing$
1	$\{A, C\}$
2	$\{A, C, S\}$
3	$\{A, C, S\}$

- Para na 3a iteração e $S \to bBb$ é excluída.

Simplificando GLC

- Exemplo: $G = (V,T,P,S) = (\{S,A,B,C\}, \{a,b,c\},P,S)$, onde $P = \{S \rightarrow aAa \mid bBb,\ A \rightarrow a \mid S,\ C \rightarrow c\}$
- E2 (qualquer símbolo é atingido a partir de S – inicial).

```
T₂ = ∅;
V₂ = { S };
repita
      V₂ = V₂ ∪ { A | X → αAβ ∈ P₁, X ∈ V₂ };
      T₂ = T₂ ∪ { a | X → αaβ ∈ P₁, X ∈ V₂ }
até   que os cardinais de V₂ e T₂ não aumentem;
```

iteração	variáveis	terminais
início	$\{S\}$	$\varnothing$
1	$\{S, A\}$	$\{a\}$
2	$\{S, A\}$	$\{a\}$

Simplificando GLC

- Exemplo: $G = (V,T,P,S) = (\{S,A,B,C\}, \{a,b,c\},P,S)$, onde $P = \{S \rightarrow aAa \mid bBb,\ A \rightarrow a \mid S,\ C \rightarrow c\}$
- E2 (qualquer símbolo é atingido a partir de S – inicial).

$T_2 = \varnothing$;
$V_2 = \{ S \}$;
repita
  $V_2 = V_2 \cup \{ A \mid X \rightarrow \alpha A \beta \in P_1, X \in V_2 \}$;
  $T_2 = T_2 \cup \{ a \mid X \rightarrow \alpha a \beta \in P_1, X \in V_2 \}$
até que os cardinais de V_2 e T_2 não aumentem;

iteração	variáveis	terminais
início	$\{S\}$	$\varnothing$
1	$\{S, A\}$	$\{a\}$
2	$\{S, A\}$	$\{a\}$

A produção $C \rightarrow c$ é excluída pois C e c não pertencem aos novos conjuntos de variáveis e terminais.

Simplificando GLC

- Exemplo: $G = (V,T,P,S) = (\{S,A,B,C\}, \{a,b,c\},P,S)$, onde $P = \{S \rightarrow aAa \mid bBb, A \rightarrow a|S, C \rightarrow c\}$

- Depois de E1 e E2, a gramática resultante é:

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

$$P = \{S \rightarrow aAa, A \rightarrow a|S\}.$$

Simplificando GLC

- Produções vazias: Pode determinar modificações diversas nas produções das gramática.
  - Etapa 1: parte de produções que geram ε diretamente e sucessivamente determina as variáveis que indiretamente geram ε.
  - Etapa 2: para cada produção que possui uma variável que gera palavra vazia, é determinada uma transição adicional sem esta variável.
  - Etapa 3: se a palavra vazia pertence ao dicionário, então será incluída uma produção para gerar essa palavra

Simplificando GLC

- Exemplo: $G = (\{S,X,Y\},\{a,b\},P,S)$, onde $P = \{S \rightarrow aXa \mid bXb \mid \varepsilon,\ X \rightarrow a \mid b \mid Y,\ Y \rightarrow \varepsilon\}$
- E1 (conjunto de variáveis que geram ε):

$V_\varepsilon = \{ A \mid A \rightarrow \varepsilon \}$;

repita

$V_\varepsilon = V_\varepsilon \cup \{ X \mid X \rightarrow X_1 \ldots X_n \in P$ tal que $X_1, \ldots, X_n \in V_\varepsilon \}$

até que o cardinal de V_ε não aumente;

iteração	V_ε
início	{S, Y}
1	{S, Y, X}
2	{S, Y, X}

Simplificando GLC

- Exemplo: $G = (\{S,X,Y\},\{a,b\},P,S)$, onde $P = \{S \rightarrow aXa \mid bXb \mid \varepsilon, X \rightarrow a \mid b \mid Y, Y \rightarrow \varepsilon \}$
- E2 (conjunto de produções sem produções vazias)

$P_1 = \{ A \rightarrow \alpha \mid \alpha \neq \varepsilon \}$;
repita
    para toda $A \rightarrow \alpha \in P_1$ e $X \in V_\varepsilon$ tal que $\alpha = \alpha_1 X \alpha_2$ e $\alpha_1\alpha_2 \neq \varepsilon$
    faça $P_1 = P_1 \cup \{ A \rightarrow \alpha_1\alpha_2 \}$
até que o cardinal de P_1 não aumente;

iteração	produções
inicial	$\{ S \rightarrow aXa \mid bXb, X \rightarrow a \mid b \mid Y \}$
1	$\{ S \rightarrow aXa \mid bXb \mid aa \mid bb, X \rightarrow a \mid b \mid Y \}$
2	$\{ S \rightarrow aXa \mid bXb \mid aa \mid bb, X \rightarrow a \mid b \mid Y \}$

Simplificando GLC

- Exemplo: $G = (\{S,X,Y\},\{a,b\},P,S)$, onde $P = \{S \rightarrow aXa \mid bXb \mid \varepsilon, X \rightarrow a \mid b \mid Y, Y \rightarrow \varepsilon \}$
- E3 (inclusão da palavra vazia): A palavra vazia pertence à linguagem $S \rightarrow \varepsilon$ é incluída no conjunto de produções.

- A gramática resultante é a seguinte:

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

  $P = \{S \rightarrow aXa \mid bXb \mid aa \mid bb \mid \varepsilon, X \rightarrow a \mid b \mid Y\}$
- Obs.: Repare que Y, agora, é um símbolo inútil.

Simplificando GLC

- Produções na forma A → B: Esse tipo de produção não acrescenta informação; se A → B e B → a, isso pode ser substituído por A → a.
  - Etapa 1 - Construção do fecho de cada variável: o fecho de uma variável é o conjunto de variáveis que podem substituí-la transitivamente. Se A → B e B → C, B e C pertencem ao fecho de A.
  - Etapa 2 – Exclusão das produções na forma A → B. Substitui as produções desse tipo de forma ao símbolo terminal ser atingível a partir de A, através de seu fecho.

Simplificando GLC

- Exemplo: G = ({S,X,Y},{a,b},P,S), onde P = {S → aXa | bXb | ε, X → a | b | S | ε }
- E1 (contrução do fecho):

  FECHO-S = Ø

  FECHO-X = {S}

Simplificando GLC

- Exemplo: G = ({S,X,Y},{a,b},P,S), onde P = {S → aXa | bXb | ε, X → a | b | S | ε }
- E2 (exclusão de A → B):

iteração	produções
inicial	$\{S \to aXa \mid bXb, X \to a \mid b \mid \varepsilon\}$
S	$\{S \to aXa \mid bXb, X \to a \mid b \mid \varepsilon\}$
X	$\{S \to aXa \mid bXb, X \to a \mid b \mid \varepsilon \mid aXa \mid bXb\}$

- A gramática resultante é:

G = ({S,X,Y},{a,b},P,S), onde

P = {S → aXa | bXb, X → a | b | aXa | bXb | ε }

Simplificando GLC

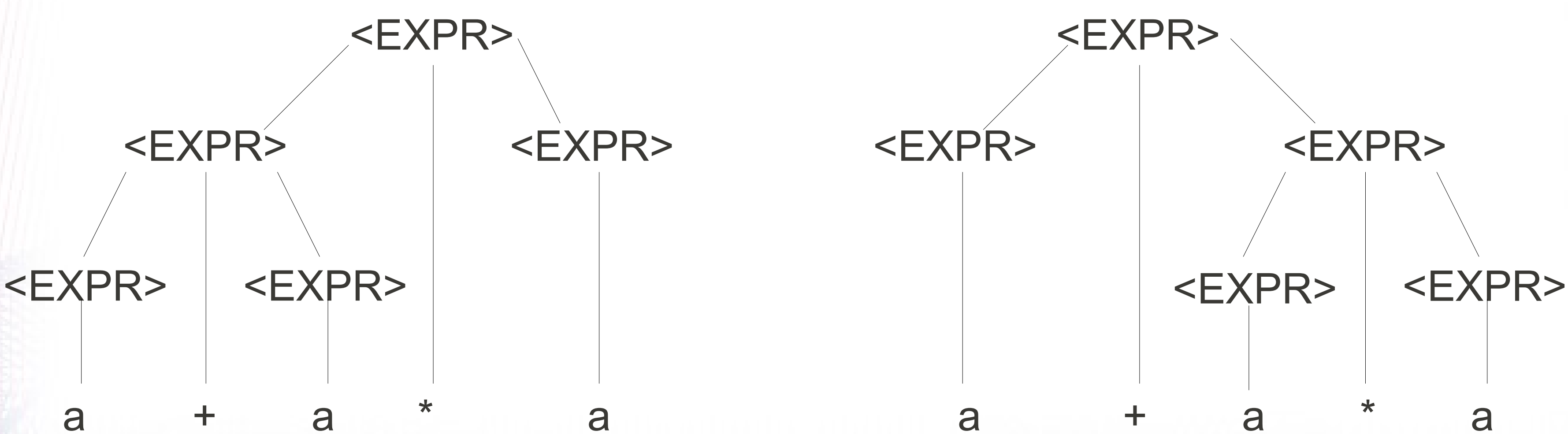
- Simplificações combinadas:
  - Nem todas as sequências de combinações surtem o efeito desejado.
  - Em uma gramática sem símbolos inúteis, mas com produções $A \rightarrow B$, ao excluir esse tipo de produção, o algoritmo pode gerar símbolos inúteis.
  - Sequências recomendadas de exclusão:
    - Produções vazias;
    - Produções $A \rightarrow B$;
    - Símbolos inúteis.

Exercícios

- Dado G = ({<EXPR>,<TERM>,<FACTOR>}, {a,+,*,(,)}, R, <EXPR>}, onde = {<EXPR> → <EXPR>+<EXPR> | <EXPR>*<EXPR> | (<EXPR>) | a} responda:
  - Para a palavra a+a*a, G é ambígua? Prove.

Exercícios

- Dado G = ({<EXPR>,<TERM>,<FACTOR>}, {a,+,*,(,)}, R, <EXPR>}, onde = {<EXPR> → <EXPR>+<EXPR> | <EXPR>*<EXPR> | (<EXPR>) | a} responda:
  - Para a palavra a+a*a, G é ambígua? Prove.
  - Sim.



Exercícios

- Considere a gramática G = ({S,X,Y,Z,A,B}, {a,b,u,v},P,S), onde

$$
\begin{aligned}
P = \{\ & S \rightarrow XYZ, \\
& X \rightarrow AXA \mid BXB \mid Z \mid \varepsilon, \\
& Y \rightarrow AYB \mid BYA \mid Z \mid \varepsilon, \\
& A \rightarrow a,\ B \rightarrow b \\
& Z \rightarrow Zu \mid Zv \mid \varepsilon\ \}
\end{aligned}
$$

- Qual a linguagem gerada?
- Simplifique a gramática.

Formas normais

Formas normais

- Estabelecem restrições rígidas na definição das produções, sem reduzir o poder de geração das GLC.
- Usadas no desenvolvimento de algoritmos(reconhecedores de linguagem) e na prova de teoremas.
  - FN de Chomsky, onde as produções são da forma: $A \rightarrow BC$ ou $A \rightarrow a$
  - FN de Greibach, onde as produções são da forma: $A \rightarrow a\alpha$ (onde α é uma palavra de variáveis)

FN de Chomsky

- Qualquer LLC é gerada por uma GLC na forma normal de Chomsky.
- A conversão segue 3 etapas:
    1) Simplificação da gramática;
    2) Transformação do lado direito das produções de comprimento maior ou igual a dois;
    3) Transformação do lado direito das produções de comprimento maior ou igual a três, em produções com exatamente duas variáveis.

FN de Chomsky

- E1:
  - Exclui produções vazias;
  - Exclui produções do tipo $A \to B$ (se o lado direito da produção tiver só um símbolo, ele deve ser terminal); e
  - Exclui opcionalmente símbolos inúteis.
  - Use os algoritmos de simplificação mostrados anteriormente!

FN de Chomsky

- E2:
  - Garante que o lado direito das produções de comprimento maior ou igual a 2 seja composto exclusivamente por variáveis.
  - Exclui um terminal substituindo-o por uma variável intermediária:

    $A \rightarrow aB$, torna-se:

    $A \rightarrow TB$, $T \rightarrow a$

FN de Chomsky

- E3:
  - Garante que o lado direito das produções de comprimento maior do que 1 seja composto exclusivamente por duas variáveis.

    $A \rightarrow BCD$

    $A \rightarrow BF$

    $F \rightarrow CD$

FN de Chomsky

- Exemplo: Transformação GLC para FNC.

  $G = (\{E\},\{+,*,[,],x\},P=\{E \rightarrow E+E \mid E*E \mid [E] \mid x\},E)$

- E1: tem algo para excluir (vazias, inúteis ou $A \rightarrow B$)???

FN de Chomsky

- Exemplo: Transformação GLC para FNC.

  G = ({E},{+,*,[,],x},P={E → E+E | E*E | [E] | x},E)
- E1: tem algo para excluir (vazias, inúteis ou A → B)??? Não1
- E2: E → E+E torna-se E→EME, onde M → +

  E → E*E torna-se E→EVE, onde V → *

  E → [E] torna-se E→C¹EC², onde C¹→[e C²→].

  E → x (não mexe).

FN de Chomsky

- Exemplo: Transformação GLC para FNC.

  G = ({E},{+,*,[,],x},P={E $\rightarrow$ E+E | E*E | [E] | x},E)

- E3: E $\rightarrow$ EME | EVE | C^1EC^2 torna-se:

  $E \rightarrow ED^1 \mid ED^2 \mid C^1D^3$

  $D^1 \rightarrow ME$

  $D^2 \rightarrow VE$

  $D^3 \rightarrow EC^2$

FN de Chomsky

- Exemplo: Transformação GLC para FNC.

  $G = (\{E\},\{+,*,[,],x\},P=\{E \rightarrow E+E \mid E*E \mid [E] \mid x\},E)$

- A gramática resultante é:

  $G = (\{E,M,V,C^1,C^2,D^1,D^2,D^3\},\{+,*,[,],x\},P,E\}$

  $P=\{E \rightarrow ED^1 \mid ED^2 \mid C^1D^3 \mid x,$

  $D^1 \rightarrow ME,\ D^2 \rightarrow VE,\ D^3 \rightarrow EC^2,$

  $M \rightarrow +, V \rightarrow *, C^1 \rightarrow [,\ C^2 \rightarrow]\}$

FN de Greibach

- Qualquer GLC que não possua palavra vazia pode ser convertida em uma gramática na forma normal de Greibach.
- A conversão segue 6 etapas:
    1. Simplificação da gramática;
    2. Renomeção das variáveis em ordem crescente qualquer;
    3. Transformação de produções para a forma $Ar \rightarrow As\alpha$, onde $r <= s$.

FN de Greibach

- Qualquer GLC que não possua palavra vazia pode ser convertida em uma gramática na forma normal de Greibach.
- A conversão segue 6 etapas:

4. Exclusão das recursões da forma $A_r \rightarrow A_r\alpha$.
5. Um terminal no início do lado direito de cada produção.
6. Produções na forma $A \rightarrow a$

FN de Greibach

- Exemplo: Tranforme a GLC em FNG.

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

  $P = \{ S \rightarrow AA \mid a,\ A \rightarrow SS \mid b\}$
- E1 (simplificando G): Já está simplificada!
- E2 (renomear em ordem crescente): $S \rightarrow A1$ e $A \rightarrow A2$

  $P = \{A1 \rightarrow A2A2 \mid a,\ A2 \rightarrow A1A1 \mid b\}$

FN de Greibach

- Exemplo: Tranforme a GLC em FNG.

  G = ({S,A},{a,b},P,S)

  P = {A1 → A2A2 | a, A2 → A1A1|b}

- E3 (tranformando produções): transformando A2 → A1A1:

  A1 → A2A2 | a

  A2 → A2A2A1 | aA1 | b

FN de Greibach

- Exemplo: Tranforme a GLC em FNG.

  G = ({S,A},{a,b},P,S)

  P = {A1 → A2A2 | a, A2 → A1A1|b}

- E3 (tranformando produções): A2 → A2A2A1 | aA1 | b tem uma recursão e deve ser tratada:

  A1 → A2A2 | a

  A2 → aA1 | b| aA1B | bB

  B → A2A1 |A2A1B

FN de Greibach

- Exemplo: Tranforme a GLC em FNG.

  G = ({S,A},{a,b},P,S)

  P = {A1 → A2A2 | a, A2 → A1A1|b}

- E5 (terminal no início):

  A1 → aA1A2 | bA2 | aA1BA2 | bBA2 | a

  A2 → aA1 | b | aA1B | bB

  B → aA1A2 | bA1 | aA1BA1 | bA1 | aA1A1B | bA1B | aA1BA1B | bBA1B

FN de Greibach

- Exemplo: Tranforme a GLC em FNG.

  G = ({S,A},{a,b},P,S)

  P = {A1 → A2A2 | a, A2 → A1A1|b}

- E6 (palavra composta exclusivamente por variáveis): todas já estão nessa forma.

FN de Greibach

- Exemplo: Tranforme a GLC em FNG.

  G = ({S,A},{a,b},P,S)

  P = {A1 → A2A2 | a, A2 → A1A1|b}

- Forma normal resultante: G = ({A1,A2,B}, {a,b},P,S)

  P = {**A1 → aA1A2 | bA2 | aA1BA2 | bBA2 | a**
  A2 → aA1 | b | aA1B | bB

  B → aA1A2 | bA1 | aA1BA1 | bA1 | aA1A1B | bA1B | aA1BA1B | bBA1B}

Recursão à esquerda

- Em algoritmos reconhecedores, p.e., é desejável que a gramática não seja recursiva à esquerda (uma variável derivando ela mesma, direta ou indiretamente, como spimbolo mais à direita da subpalavra gerada).

$$A \rightarrow^{+} A\alpha$$

- Para retirar essa recursão execute os 4 primeiros passos da Forma Normal de Greibach.