Aula 9: Linguagens Livres de Contexto

Prof. Lucio A. Rocha

Engenharia de Computação
Universidade Tecnológica Federal do Paraná, UTFPR
Campus Apucarana, Brasil

2º semestre / 2023

Sumário

Seção 1

Linguagens Livres de Contexto

Linguagens Livres de Contexto

- Linguagens Livres de Contexto (LLC ou Tipo 2):
  - Universo de linguagens mais amplo do que as linguagens regulares.
  - Parênteses, chaves, colchetes balanceados.
  - Blocos estruturados.
  - Típico de linguagens de programação.
- Lida com algoritmos reconhecedores e geradores.

Linguagens Livres de Contexto

- Aplicações:
  - Linguagens de programação
  - Analisadores sintáticos
  - Tradutores de linguagens
  - Processadores de texto
- Hierarquia de Chomsky
  - Contém a classe das linguagens regulares
- Relativamente restrita
  - Fácil definir linguagens que não pertencem a essa linguagem.

Linguagens Livres de Contexto

- LLCs são abordadas com os seguintes formalismos:
  - Gramática Livre de Contexto (axiomático ou gerador)
    - Restrições na forma das regras de produção
    - Mais livre que na gramática regular
  - Autômato com pilha (operacional ou reconhecedor)
    - Equivalente ao AFN
    - Possui uma memória auxiliar do tipo pilha
    - Pode ser lida ou gravada

Linguagens Livres de Contexto

- Com relação à Gramática Livre de Contexto (GLC):
  - Árvore de derivação:
    - Representação de uma palavra no formato de árvore.
    - Símbolo sentencial é a raiz.
    - Símbolos terminais são as folhas.
  - Gramática Ambígua:
    - Ao menos uma ou mais palavras com duas ou mais árvores de derivação.
  - Simplificação de Gramática (produções):
    - Sem reduzir o poder de geração de palavras.
  - Forma normal:
    - Restrições rígidas na forma das produções.

Subseção 2

Gramática Livre de Contexto

Gramática Livre de Contexto

- Def.: Gramática Livre de Contexto (GLC):
  - $G = (V_T, V_N, \mathbb{P}, S_i)$
- V_T: símbolos terminais.
- V_N: símbolos não-terminais.
- $\mathbb{P}$: regras de produção (função parcial):

$$A \to \alpha \mid \alpha \in (V_T \cup V_N)^*$$

- Ou seja, o lado esquerdo das produções contém apenas uma variável.
- α: é uma palavra.
- S_i: é o símbolo sentencial.

Gramática Livre de Contexto

- Linguagem G gerada pela GLC:

$$GERA(G) = \{w \in {V_T}^* \mid S_i \Rightarrow^+ \ w\}$$

- Portanto, toda Linguagem Regular é também uma Linguagem Livre de Contexto.

Gramática Livre de Contexto

- Gramática com Duplo balanceamento:
- Analogia com o Duplo balanceamento em linguagens de programação.
  - Exemplos:
  - $begin^n end^n$, $if^n fi^n$, $for^n\ done^n$
  - $\{^n\ \}^n, [^n\ ]^n$, etc.

Gramática Livre de Contexto

- Exemplo: Gramática com Duplo balanceamento:
  Seja a linguagem L:

$$L = \{0^n 1^n \mid n \geq 0\}$$
$$G = (\{0,1\}, \{Z\}, \mathbb{P}, Z)$$
$$\mathbb{P} = \{Z \rightarrow 0Z1 \mid Z \rightarrow \varepsilon\}$$

- $GERA(G) = L$
- Derivação: **w=0011**:

$$Z \Rightarrow 0Z1 \Rightarrow 00Z11 \Rightarrow 00\varepsilon 11 = 0011$$

Gramática Livre de Contexto

- Exemplo: GLC para Expressões Aritméticas
  Seja a linguagem L:
  L=expressões aritméticas com colchetes balanceados, dois operandos e um operador.

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

- $GERA(G) = L$
- Derivação: **w=[x+x]*x**:

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

Subseção 3

Árvore de Derivação

Árvore de Derivação

- Árvore: estrutura de dados que possui:
  - Nó raiz.
  - Subárvores.
  - Nós folha.

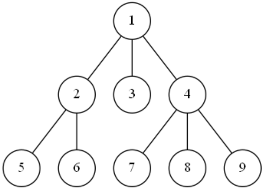
- 

Figura: Árvore de Derivação.

Árvore de Derivação

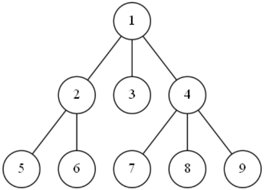


Figura: Árvore de Derivação.

- Número de subárvores determina o grau do nó:
  - Nó 1: grau 3.
  - Nó 2: grau 2.
  - Nó 3: grau 0 (zero).
- Hierarquia: Nó 2 é filho do nó 1.

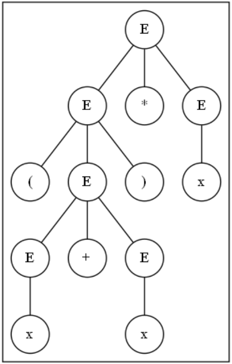
Árvore de Derivação

- Árvore Sintática: representação de derivações para reconhecimento de palavras em uma estrutura de árvore.
  - Raiz: símbolo sentencial.
  - Vértices intermediários: variáveis (símbolos não-terminais).
  - Folhas: símbolos terminais.

Árvore de Derivação

- Exemplo: Construir árvore sintática de reconhecimento da sentença
  $(x+x)*x$
  de acordo com a GLC G:

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



Árvore de Derivação

- Derivação canônica: escolha padronizada das derivações quando há mais de uma opção de derivação.
- Derivação canônica mais à esquerda:

$$E \Rightarrow^2 E * E \Rightarrow^3 (E) * E \Rightarrow^1 (E + E) * E \Rightarrow^4 (x + E) * E \Rightarrow^4 (x + x) * E \Rightarrow^4 (x + x) * x$$

- Derivação canônica mais à direita:

$$E \Rightarrow^2 E * E \Rightarrow^4 E * x \Rightarrow^3 (E) * x \Rightarrow^1 (E + E) * x \Rightarrow^4 (x + x) * x$$

Seção 2

Gramática ambígua

Gramática ambígua

- Gramática ambígua:

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

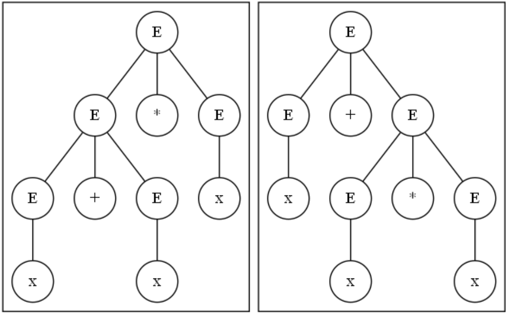
- 

Figura: Árvore de Derivação.

Gramática ambígua

- Portanto, temos 2 árvores sintáticas diferentes que reconhecem a mesma palavra.
- Como remover a ambiguidade: gerar uma nova sequência ordenada de produções (E seguido de M seguido de P).
- Ideia: <u>M</u>ultiplicação sempre <u>após</u> a soma.
- $E \rightarrow E + E$:

$$E \rightarrow E + M \text{ (}E\text{ com recursão nova)}$$
$$E \rightarrow M \text{ (}E\text{ sem recusão)}$$

- $E \rightarrow E * E$:

$$M \rightarrow M * P \text{ (\underline{M}ultiplicação com recursão nova)}$$
$$M \rightarrow P \text{ (\underline{M}ultiplicação sem recursão)}$$

Gramática ambígua

- A nova sequência de produções da gramática G_3 é como segue:
  - $\mathbb{P}: E \rightarrow E + M$
    $E \rightarrow M$
    $M \rightarrow M * P$
    $M \rightarrow P$
    $P \rightarrow (E)$ (P é feito após M)
    $P \rightarrow x$ (P é feito após M)
- Derivação de $x + x * x$:
- $E \Rightarrow^1 E{+}M \Rightarrow^6 x{+}M \Rightarrow^3 x{+}M{*}P \Rightarrow^4 x{+}P{*}P \Rightarrow^6 x{+}x{*}P \Rightarrow^6 x{+}x{*}x$

Gramática ambígua

- Remoção da Recursão à Esquerda:

$$A \rightarrow A\alpha$$
$$A \rightarrow x$$

- Pode gerar: $A \Rightarrow A\alpha \Rightarrow A\alpha\alpha...\alpha \Rightarrow x\alpha\alpha...\alpha$
- Analisador sintático preditivo (que faz derivações mais à esquerda) não pode operar com gramáticas que tenham produções à esquerda.

Gramática ambígua

- A remoção da Recursão à Esquerda é feita em 3 passos:

$$A \rightarrow A\alpha$$
$$A \rightarrow x$$

Trocar por:

$$A \rightarrow xB \text{ (}A \text{ com recursão nova)}$$
$$B \rightarrow \alpha B \text{ (}B \text{ com recursão)}$$
$$B \rightarrow \varepsilon \text{ (}B \text{ sem recursão)}$$

- Pode gerar: $A \Rightarrow xB \Rightarrow x\alpha B \Rightarrow x\alpha\alpha ... B \Rightarrow x\alpha\alpha ... \alpha$

Seção 3

Simplificação de Gramática

Simplificação de Gramática

- Símbolos inúteis:

$$G = (\{S, A, B, C\}, \{a, b, c\}, \mathbb{P}, S)$$
$$\mathbb{P} : \{S \to aAa \mid bBb,\ A \to a \mid S,\ C \to c\}$$

- Passo 1: Toda variável gera terminais, direta ou indiretamente: bBb pode ser excluída;
- Passo 2: Todo símbolo é atingível a partir do símbolo sentencial: $C \to c$ pode ser excluída.

Simplificação de Gramática

- Passo 1:

Iteração	Variáveis
1	ε
2	{A,C}
3	{A,C,S}

- Passo 2:

Iteração	Variável	Terminais
1	{S}	ε
2	{S,A}	a
3	{S,A}	a

Produções vazias

- Produções vazias:

$$G = (\{S, X, Y\}, \{a, b\}, \mathbb{P}, S)$$
$$\mathbb{P} : \{S \rightarrow aXa \mid bXb \mid \varepsilon,\ X \rightarrow a \mid b \mid Y,\ Y \rightarrow \varepsilon\}$$

- Passo 1:
  - a) Identificar as Variáveis que diretamente geram ε:

$$S \rightarrow \varepsilon$$
$$Y \rightarrow \varepsilon$$

  b) Identificar as Variáveis que indiretamente geram ε:

$$S \rightarrow X \rightarrow Y \rightarrow \varepsilon$$

Produções vazias

- Passo 2:
  - a) Excluir produções vazias:

$$S \to aXa \mid bXb$$
$$X \to a \mid b \mid Y$$

b) Nas produções não-vazias, se a produção tem variável que gera ε, adicionar nova produção sem essa variável:

$$S \to aXa \mid bXb \mid aa \mid bb$$
$$X \to a \mid b \mid Y$$

- Passo 3: incluir a palavra vazia, se necessário.
  - Neste exemplo, ε é palavra da linguagem.

$$S \to aXa \mid bXb \mid aa \mid bb \mid \varepsilon$$
$$X \to a \mid b \mid Y$$

  - Neste exemplo, Y é símbolo inútil.

Substituição de variáveis

- Objetivo: remover produções unitárias $A \rightarrow B$
- Se $A \rightarrow B, B \rightarrow C$ então $A \rightarrow C$
- a) Fecho transitivo de cada variável:
  - A alcança C exclusivamente com transições $X \rightarrow Y$
  - FECHO(A): conjunto de variáveis que podem substituir A transitivamente.
  - FECHO(A) = {B,C}
- b) Exclusão das produções que substituem variáveis:
  - Excluir $A \rightarrow B$ por $A \rightarrow \alpha$, onde α é alcançável através do FECHO(A).

Substituição de variáveis

- Exemplo:
  - $G = (\{a, b\}, \{S, X\}, \mathbb{P}, S)$
    $\mathbb{P} = \{S \rightarrow aXa|bXb,$
    $X \rightarrow a|b|S|\varepsilon\}$

1. FECHO(S) $= \emptyset$
   FECHO(X) $= \{\text{S}\}$
2. FECHO(S) $\Rightarrow \{S \rightarrow aXa|bXb, X \rightarrow a|b|\emptyset|\varepsilon\ \}$
   FECHO(X) $\Rightarrow \{S \rightarrow aXa|bXb, X \rightarrow a|b|aXa|bXb|\varepsilon\}$

Substituição de variáveis

- Exemplo:
- $G = (\{a, b\}, \{S, A, B\}, \mathbb{P}, S)$
  $\mathbb{P} = \{S \rightarrow Aa|B,$
  $A \rightarrow b|B$
  $B \rightarrow A|a$
  $\}$
- G_1:Grupo não-unitário:
  $S \rightarrow Aa$
  $A \rightarrow b$
  $B \rightarrow a$
- G_2: Grupo unitário $(P \rightarrow Q)$:
  $S \rightarrow B$
  $A \rightarrow B$ $(B \in G1$, adiciona $A \rightarrow a$ em $G_1)$
  $B \rightarrow A$

Substituição de variáveis

- G_1:Grupo não-unitário:
  $S \rightarrow Aa$
  $A \rightarrow b|a$
  $B \rightarrow a$
- G_2: Grupo unitário:
  $S \rightarrow B$
  $A \rightarrow B$
  $B \rightarrow A$ ($A \in G1$, adiciona $B \rightarrow b$ em G_1)

Substituição de variáveis

- G_1:Grupo não-unitário:
  $S \to Aa$
  $A \to b|a$
  $B \to a|b$
- G_2: Grupo unitário:
  $S \to B$ ($B \in G1$, adiciona $S \to a|b$ em G_1)
  $A \to B$
  $B \to A$

Substituição de variáveis

- G_1:Grupo não-unitário:
  $S \rightarrow Aa|a|b$
  $A \rightarrow b|a$
  $B \rightarrow a|b$ (Produção inútil)

Gramática resultante:

- $G = (\{a, b\}, \{S, A\}, \mathbb{P}, S)$
  $\mathbb{P} = \{S \rightarrow Aa|a|b,$
  $A \rightarrow b|a$
  $\}$