Teoria de Linguagem

Gramáticas Livres de Contexto

Vinicius H. S. Durelli
✉ durelli@ufsj.edu.br

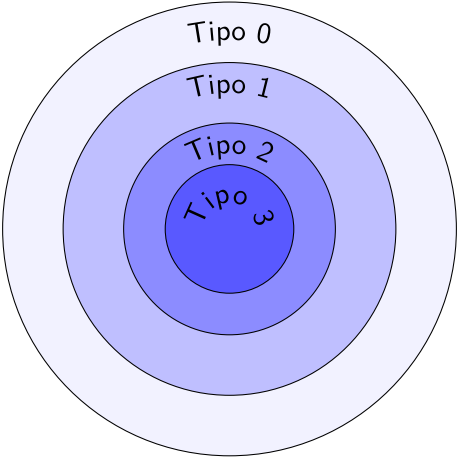
UFSJ
UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

Organização

Linguagens livres de contexto (Tipo 2)...

Em relação as linguagens regulares, linguagens livres de contexto compreendem um universo mais amplo.

- Aborda questões típicas de linguagens de programação como, por exemplo, **parênteses balanceados**.
- A classe das linguagens livres de contexto contêm propriamente a classe das linguagens regulares (Sipser 2012).



Formalismos...

O estudo das linguagens livres de contexto é abordado usando os seguintes formalismos (Menezes 2011):

- **Autômato com pilha**: formalismo operacional ou reconhecedor cuja estrutura básica é semelhante à do autômato finito não determinístico.
- **Gramática livre de contexto**: formalismo axiomático ou gerador que, conforme o nome indica, é uma gramática com restrições na forma das regras de produção.

Formalismos...

O estudo das linguagens livres de contexto é abordado usando os seguintes formalismos (Menezes 2011):

- **Autômato com pilha**: formalismo operacional ou reconhecedor cuja estrutura básica é semelhante à do autômato finito não determinístico.
- **Gramática livre de contexto**: formalismo axiomático ou gerador que, conforme o nome indica, é uma gramática com restrições na forma das regras de produção.

Gramáticas livres de contexto (1)

Uma Gramática Livre de Contexto (GLC) é formada por **quatro componentes**:

1. Um conjunto finito de **símbolos** que formam as **palavras** da linguagem sendo definida. Esse alfabeto é normalmente denominado conjunto de **símbolos terminais**.
2. Um conjunto finito de **variáveis**, denominadas **não terminais**. Cada variável representa uma linguagem, i.e., um conjunto de símbolos.

Gramáticas livres de contexto (2)

Uma Gramática Livre de Contexto (GLC) é formada por **quatro componentes**:

3. Uma das variáveis representa a linguagem sendo definida como um todo: essa variável é denominada **símbolo inicial**. As outras variáveis representam classes adicionais de **palavras** que auxiliam na definição da linguagem do símbolo inicial.
4. Um conjunto finito de **produções** ou **regras** que representam a definição recursiva da linguagem. Cada regra consiste de:
   - Uma variável;
   - O símbolo que denota a regra $\rightarrow$;
   - Uma **palavra** com zero ou mais **terminais e não terminais**.

Definição → *Gramática Livre de Contexto (GLC)*

Uma gramática livre de contexto G é uma gramática:

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

Com a restrição de que qualquer regra de produção de P é da forma:

$$A \rightarrow \alpha$$

onde A é uma variável de V e α uma palavra de $(V \cup T)^*$. ❑

Uma GLC pode ser formalmente definida como uma quádrupla $G = (V, T, P, S)$, onde V é um conjunto de variáveis, T os terminais, P representa o conjunto de produções (i.e., $V \times \{V \cup T\}^*$) e $S \in V$ é o símbolo inicial.

Definições formais: GLC e linguagem livre de contexto (2)

Definição → *Linguagem Livre de Contexto (Tipo 2)*

Seja G uma GLC. A linguagem gerada por G:

$$GERA(G) = \{w \in T^* \mid S \overset{+}{\Longrightarrow} w\}$$

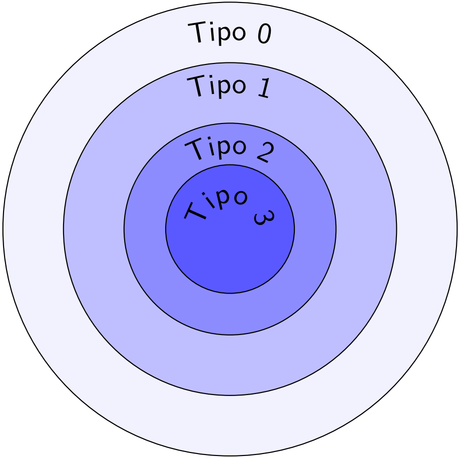
é dita uma linguagem livre de contexto ou linguagem tipo 2. ❑

Sobre o nome "livre de contexto"...

☞ O nome **"livre de contexto"** deve-se ao fato de representar a mais geral classe de linguagens cuja produção é da forma:

$$A \rightarrow \alpha$$

Em uma derivação, a variável A deriva α **sem depender** ("livre") de qualquer análise dos símbolos que antecedem ou sucedem A ("contexto") na palavra sendo derivada (Menezes 2011).



1. Contextualização
   - Gramáticas livres de contexto
2. Definição formal
3. Exemplo
4. Derivações
   - Árvore de derivação
   - Derivações mais à esquerda (direita)
5. GLC ambígua
6. Considerações finais

Exemplo (1)

Considere a linguagem dos palíndromos L_{pal}. Para simplificar a discussão, considere somente os palíndromos do alfabeto $\{0, 1\}$.

Formalmente, uma palavra w é um palíndromo se e somente se $w = w^R$.

* L_{pal} pode ser expressada por meio de uma definição recursiva:

Exemplo (1)

Considere a linguagem dos palíndromos L_{pal}. Para simplificar a discussão, considere somente os palíndromos do alfabeto $\{0, 1\}$.

Formalmente, uma palavra w é um palíndromo se e somente se $w = w^R$.

* L_{pal} pode ser expressada por meio de uma definição recursiva:

> - **Base:** ϵ, 0 e 1 são palíndromos.
> - **Indução:** Se w é um palíndromo, $0w0$ e $1w1$ também são.
>
> Nenhuma *palavra* de 0s e 1s é um palíndromo a menos que ela seja definida de acordo com essas regras base e de indução.

Exemplo (2)

Tal definição recursiva de linguagem pode ser facilmente expressada utilizando uma GLC. As regras que definem L_{pal} são:

$P \rightarrow \varepsilon$
$P \rightarrow 0$
$P \rightarrow 1$
$P \rightarrow 0P0$
$P \rightarrow 1P1$

- As três primeiras regras formam a base.
- As outras duas produções foram a parte indutiva da definição.

Por exemplo, a regra quatro afirma que a partir de qualquer *palavra* w pertencente à classe P, é possível formar a *palavra* $0w0$ que também pertence à linguagem.

Exemplo (2)

Tal definição recursiva de linguagem pode ser facilmente expressada utilizando uma GLC. As regras que definem L_{pal} são:

$P \rightarrow \varepsilon$
$P \rightarrow 0$
$P \rightarrow 1$
$P \rightarrow 0P0$
$P \rightarrow 1P1$

- As três primeiras regras formam a base.
- As outras duas produções foram a parte indutiva da definição.

Por exemplo, a regra quatro afirma que a partir de qualquer *palavra* w pertencente à classe P, é possível formar a *palavra* $0w0$ que também pertence à linguagem.

Exemplo (3)

☞ Quando uma variável tem mais de uma produção, uma barra vertical (i.e., $|$) pode ser usada para separar as produções.

Portanto, o conjunto de produções da gramática ilustrada anteriormente pode ser escrito como:

$$P \rightarrow 0P0 \mid 1P1 \mid 0 \mid 1 \mid \varepsilon$$

Exercícios (1)

Considerando o alfabeto $\Sigma = \{1, 0\}$:

Exercício ①: Crie uma GLC que aceita a linguagem descrita abaixo:
$L_{e1} = \{w \ : \ w = w^R \text{ e } |\text{w}| \text{ é par}\}$

Exercício ②: Desenvolva uma GLC que aceita a linguagem denotada abaixo:
$L_{e2} = \{w \ : |\text{w}| \text{ é ímpar e o símbolo do meio é } 1\}$

Exercícios (1)

Considerando o alfabeto $\Sigma = \{1, 0\}$:

Exercício ①: Crie uma GLC que aceita a linguagem descrita abaixo:
$L_{e1} = \{w \ : \ w = w^R$ e |w| é par$\}$

$S \rightarrow 0S0 \mid 1S1 \mid \varepsilon$

Exercício ②: Desenvolva uma GLC que aceita a linguagem denotada abaixo:
$L_{e2} = \{w \ :$ |w| é ímpar e o símbolo do meio é 1$\}$

Exercícios (1)

Considerando o alfabeto $\Sigma = \{1, 0\}$:

Exercício ①: Crie uma GLC que aceita a linguagem descrita abaixo:
$L_{e1} = \{w \; : \; w = w^R \text{ e } |\text{w}| \text{ é par}\}$

$S \rightarrow 0S0 \mid 1S1 \mid \varepsilon$

Exercício ②: Desenvolva uma GLC que aceita a linguagem denotada abaixo:
$L_{e2} = \{w \; : |\text{w}| \text{ é ímpar e o símbolo do meio é } 1\}$

$S \rightarrow 0S0 \mid 0S1 \mid 1S0 \mid 1S1 \mid 1$

Derivações (1)

O processo fundamental para geração de uma *palavra* é a **aplicação de regras**, tal processo é comumente denominado **derivação** (Hopcroft et al. 2006).

> A aplicação de uma regra $A \to w$ sobre uma *palavra* $uAv \in \{V \cup T\}^*$ produz uwv. Os prefixos u e v definem o contexto em que a regra é aplicada.

☞ Os **prefixos e sufixos não definem restrições** sobre como e quando uma regra pode ser aplicada.

Derivações (2)

Definição → *Derivação*

Uma palavra $w \in \{V \cup T\}^*$ é derivada de $A \in \{V \cup T\}^*$ se existe uma sequência finita de regras que transformam A em w:

$$A \implies w_1 \implies w_2 \implies \ldots \implies w$$

Note que $\implies$ pode ser expandido para representar **zero, uma ou mais derivações**. Para tal, o símbolo * é usado para denotar **"zero ou mais derivações"**.

- **Base:** para qualquer *string* $\alpha \in \{V \cup T\}$, diz-se que $\alpha \overset{*}{\underset{G}{\implies}} \alpha$. Isso é, qualquer *string* deriva ela mesma.
- **Indução:** Se $\alpha \overset{*}{\underset{G}{\implies}} \beta$ e $\beta \overset{*}{\underset{G}{\implies}} \gamma$, então $\alpha \overset{*}{\underset{G}{\implies}} \gamma$.

Exercícios (2)

✎ Considere um caso clássico e de fundamental importância na computação: **duplo balanceamento**.

$$L_{bal} = \{a^n b^n \mid n \geqslant 0\}$$

❸ Defina uma GLC que gere L_{bal}:

Exercícios (2)

✎ Considere um caso clássico e de fundamental importância na computação: **duplo balanceamento**.

$$L_{bal} = \{a^n b^n \mid n \geqslant 0\}$$

❸ Defina uma GLC que gere L_{bal}:

$$G_{bal} = (\{S\}, \{a, b\}, P_{bal}, S)$$

$S \rightarrow \varepsilon \mid aSb$

❹ Demostre a sequência de derivação para a palavra *aabb*:

✎ Considere um caso clássico e de fundamental importância na computação: **duplo balanceamento**.

$$L_{bal} = \{a^n b^n \mid n \geqslant 0\}$$

❸ Defina uma GLC que gere L_{bal}:

$$G_{bal} = (\{S\}, \{a, b\}, P_{bal}, S)$$

$S \rightarrow \varepsilon \mid aSb$

❹ Demostre a sequência de derivação para a palavra *aabb*:

$$S \implies aSb \implies aaSbb \implies aa\varepsilon bb = aabb$$

Exercícios (2)

❺ Defina uma GLC que gere **expressões aritméticas**.
A GLC deve ser composta de expressões aritméticas contendo parênteses balanceados, dois operadores (i.e., $+$ e $\div$) e um operando.

Exercícios (2)

❺ Defina uma GLC que gere **expressões aritméticas**.
A GLC deve ser composta de expressões aritméticas contendo parênteses balanceados, dois operadores (i.e., $+$ e $\div$) e um operando.

$$G_{exp} = (\{E\}, \{+, \div, (,), x\}, P_{exp}, E)$$

$E \rightarrow E + E \mid E \div E \mid (E) \mid x$

❻ Demostre a sequência de derivação para a palavra $(x + x) \div x$:

Exercícios (2)

❺ Defina uma GLC que gere **expressões aritméticas**.
A GLC deve ser composta de expressões aritméticas contendo parênteses balanceados, dois operadores (i.e., $+$ e $\div$) e um operando.

$$G_{exp} = (\{E\}, \{+, \div, (,), x\}, P_{exp}, E)$$

$E \rightarrow E + E \mid E \div E \mid (E) \mid x$

❻ Demostre a sequência de derivação para a palavra $(x + x) \div x$:

$$E \implies E \div E \implies (E) \div E \implies (E + E) \div E \implies$$
$$(x + E) \div E \implies (x + x) \div E \implies (x + x) \div x$$

Exercícios (3)

✎ Considerando novamente a gramática desenvolvida anteriormente e que denota a linguagem a seguir: $L_{e2} = \{w \in \{0,1\} \mid$ o tamanho de w é impar e o símbolo do meio é $1\}$.

$$G_{L_{e2}} = (\{S\}, \{0,1\}, P_{L_{e2}}, S)$$

$S \rightarrow 0S0 \mid 1S1 \mid 0S1 \mid 1S0 \mid 1$

❼ Demostre a sequência de derivação para a palavra 1011111:

Exercícios (3)

✎ Considerando novamente a gramática desenvolvida anteriormente e que denota a linguagem a seguir: $L_{e2} = \{w \in \{0,1\} \mid$ o tamanho de w é impar e o símbolo do meio é $1\}$.

$$G_{L_{e2}} = (\{S\}, \{0,1\}, P_{L_{e2}}, S)$$

$S \rightarrow 0S0 \mid 1S1 \mid 0S1 \mid 1S0 \mid 1$

❼ Demostre a sequência de derivação para a palavra 1011111:

$$S \implies 1S1 \implies 10S11 \implies 101S111 \implies 1011111$$

Árvores de derivação (*parse trees*) são importantes para a definição de **linguagens de programação e seus compiladores**.

Analisador Sintático (*Parser*)

Podemos imaginar o analisador sintático de um compilador como **"um dispositivo que tenta determinar se existe uma derivação da sentença de entrada de acordo com uma gramática livre de contexto"**.

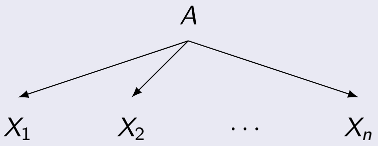
Em algumas aplicações (e.g., compiladores e processadores de texto) é conveniente representar a derivação de palavras na forma de árvore:

- Raiz: símbolo inicial;
- Folhas: símbolos terminais.

Árvore de derivação: definição

Definição → *Árvore de derivação*

- A raiz é o símbolo inicial da gramática;
- Os vértices interiores são variáveis. Se A é um vértice interior e $X_1, X_2, \ldots X_n$ são os “filhos” de A, então:
  - $A \to X_1, X_2, \ldots X_n$ é uma produção da gramática;
  - os vértices $X_1, X_2, \ldots X_n$ são ordenados da esquerda para a direita.



- Um vértice folha é um símbolo terminal ou ε.

Exemplo

Considere a gramática que gera L_{bal}: $G_{bal} = (\{S\}, \{a, b\}, P_{bal}, S)$

$$S \rightarrow \varepsilon \mid aSb$$

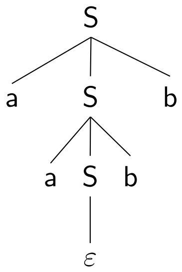
Exemplo de árvore de derivação para a palavra *aabb*:

Exemplo

Considere a gramática que gera L_{bal}: $G_{bal} = (\{S\}, \{a, b\}, P_{bal}, S)$

$$S \rightarrow \varepsilon \mid aSb$$

Exemplo de árvore de derivação para a palavra *aabb*:



Exercícios (3)

❺ Considerando a gramática G_{exp} (abaixo), gere a árvore de derivação para a palavra $(x + x) \div x$:

$$G_{exp} = (\{E\}, \{+, \div, (,), x\}, P_{exp}, E)$$

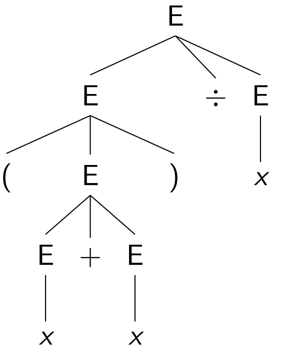
$E \rightarrow E + E \mid E \div E \mid (E) \mid x$

Exercícios (3)

❺ Considerando a gramática G_{exp} (abaixo), gere a árvore de derivação para a palavra $(x + x) \div x$:

$$G_{exp} = (\{E\}, \{+, \div, (,), x\}, P_{exp}, E)$$

$E \rightarrow E + E \mid E \div E \mid (E) \mid x$



Árvore de derivação × derivações

> Uma única árvore de derivação pode representar derivações distintas de uma mesma palavra (Menezes 2011).

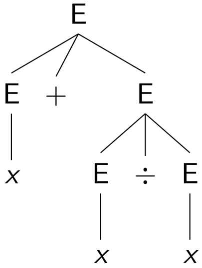
A palavra $x + x \div x$ pode ser gerada por diversas derivações distintas:

- Derivando sempre a **variável mais à esquerda**:

$$E \implies E + E \implies x + E \implies x + E \div E \implies x + x \div E \implies x + x \div x$$

- Derivando sempre a **variável mais à direita**:

$$E \implies E + E \implies E + E \div E \implies E + E \div x \implies E + x \div x \implies x + x \div x$$



Derivação mais à esquerda (direita)

Definição → *Derivação mais à Esquerda (Direita)*

Dada uma árvore de derivação, uma derivação mais à esquerda (respectivamente, derivação mais à direita) de uma palavra é a sequência de produções aplicadas sempre à variável mais à esquerda (direita) da palavra em cada passo da derivação (Menezes 2011). ❑

Exemplo de **derivação mais à esquerda**:

$$E \implies E+E \implies x+E \implies x+E\div E \implies x+x\div E \implies x+x\div x$$

Exemplo de **derivação mais à direita**:

$$E \implies E+E \implies E+E\div E \implies E+E\div x \implies E+x\div x \implies x+x\div x$$

GLC ambígua (1)

Uma mesma palavra pode ser associada a **duas ou mais árvores de derivação**, caracterizando uma **gramática ambígua**.

- Em várias aplicações, é conveniente que a gramática usada não seja ambígua.

Definição $\rightarrow$ *Gramática Ambígua*

Uma gramática é dita ambígua se existe pelo menos uma palavra que possua duas ou mais árvores de derivação nesta gramática. ❑

Alguma das GLC vistas anteriormente é ambígua?

GLC ambígua (1)

Uma mesma palavra pode ser associada a **duas ou mais árvores de derivação**, caracterizando uma **gramática ambígua**.

- Em várias aplicações, é conveniente que a gramática usada não seja ambígua.

Definição → *Gramática Ambígua*

Uma gramática é dita ambígua se existe pelo menos uma palavra que possua duas ou mais árvores de derivação nesta gramática. ❑

Alguma das GLC vistas anteriormente é ambígua?

Gramática ambígua (2)

A palavra $x + x \div x$ pode ser gerada por árvores distintas (ilustradas abaixo). Portanto, a gramática G_{exp} é ambígua.

Gramática ambígua (3)

Outra forma de determinar ambiguidade: verificando a existência de pelo menos uma palavra com duas ou mais derivação à esquerda (direita).

Exemplo: a palavra $x + x \div x$ possui mais de uma derivação à esquerda (respectivamente, à direita) como segue:

Derivações à esquerda:

1. $E \Longrightarrow E + E \Longrightarrow x + E \Longrightarrow x + E \div E \Longrightarrow x + x \div E \Longrightarrow x + x \div x$
2. $E \Longrightarrow E \div E \Longrightarrow E + E \div E \Longrightarrow x + E \div E \Longrightarrow x + x \div E \Longrightarrow x + x \div x$

Derivações à direita:

1. $E \Longrightarrow E + E \Longrightarrow E + E \div E \Longrightarrow E + E \div x \Longrightarrow E + x \div x \Longrightarrow x + x \div x$
2. $E \Longrightarrow E \div E \Longrightarrow E \div x \Longrightarrow E + E \div x \Longrightarrow E + x \div x \Longrightarrow x + x \div x$

Gramática ambígua (4)

Definição → *Gramática Ambígua* (Parte 2)

Uma GLC G é ambígua se existe pelo menos uma palavra tal que possui:

- duas ou mais derivações à esquerda; ou
- duas ou mais derivações à direita.

❑

Definição → *Linguagem Inerentemente Ambígua*

Uma linguagem é dita inerentemente ambígua se qualquer GLC que a define é ambígua. ❑

Considerações finais...

Na aula de hoje nós vimos:

- Gramáticas livres de contexto (GLC);
- Derivações;
- Árvores de derivação;
- Ambiguidade;

Na **próxima aula**: simplificando GLC.

Referências

Hopcroft, John E., Rajeev Motwani, & Jeffrey D. Ullman (2006). *Introduction to Automata Theory, Languages, and Computation*. 3rd ed. Pearson, p. 750.

Menezes, Paulo Blauth (2011). *Linguagens Formais e Autômatos*. 6th ed. Livros Didáticos Informática da UFRGS. Bookman, p. 256.

Sipser, Michael (2012). *Introduction to the Theory of Computation*. 3rd ed. Cengage Learning, p. 480.

☺**Próxima aula: exercício(s) sobre o conteúdo da aula de hoje!** ☺