

3 LINGUAGENS E SUAS REPRESENTAÇÕES

⇒ CONCEITO

Uma **linguagem** formal L é um conjunto de sentenças formadas por símbolos tomados de algum alfabeto V ($L \subseteq V^*$).

EXEMPLO: o conjunto de sentenças válidas da língua portuguesa poderia ser definido como um subconjunto de $\{a, b, c, \dots, z\}^+$.

Uma linguagem pode ser:

- **finita:** seja $V = \{a, b\}$, tem-se
 $L_1 = \{w \mid w \in V^* \wedge |w| < 3\}$

$$\left. \begin{array}{l} L_2 = \emptyset \\ L_3 = \{\varepsilon\} \end{array} \right\} L_2 \neq L_3$$

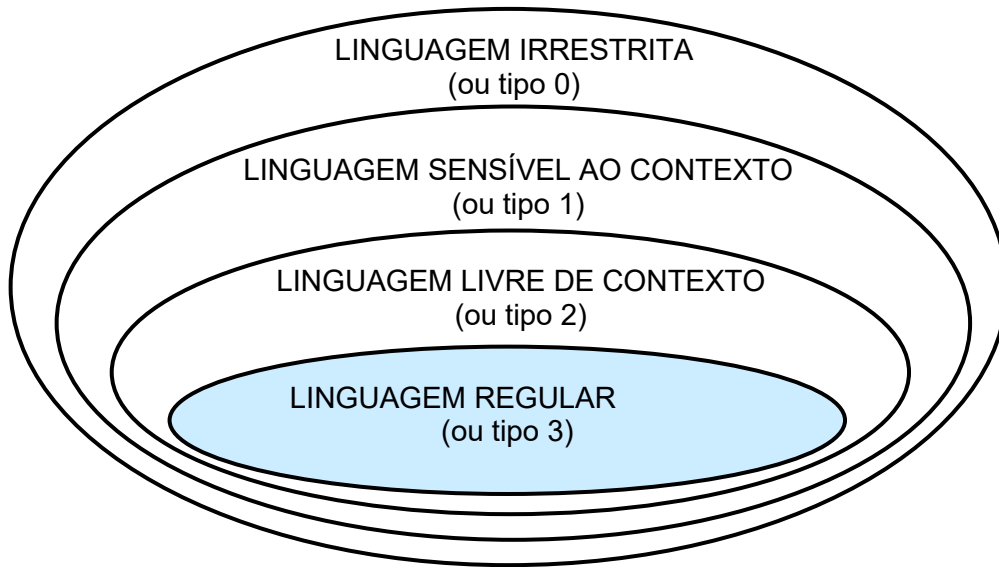
- **infinita:** seja $V = \{a, b\}$, tem-se
 $L_1 = \{w \mid w \in V^* \wedge |w| \bmod 2 = 0\}$
 $L_2 = \{w \mid w \text{ é uma palíndrome}\}$
 $L_3 = \text{linguagem de programação PASCAL}$

⇒ FORMAS DE ESPECIFICAÇÃO

- enumeração das sentenças
- descrição algébrica
- representação finita:
 - **reconhecedores:** autômatos finitos, máquinas de Turing, autômatos de pilha
 - **sistemas geradores:** gramática

Todo **reconhecedor** e todo **sistema gerador** pode ser representado por um **algoritmo**.

CLASSIFICAÇÃO DAS LINGUAGENS



⇒ LINGUAGEM REGULAR

- **FORMALISMOS:** *expressões regulares, autômatos finitos, gramáticas regulares.*
- **ALGORITMOS** (reconhecimento/geração) são **eficientes**, de **fácil implementação** e **pouca complexidade**.
- **APLICAÇÕES:** **analísadores léxicos**, busca e substituição de palavras em editores de texto.

⇒ LINGUAGEM LIVRE DE CONTEXTO


- **FORMALISMOS:** *autômatos de pilha, gramáticas livres de contexto.*
- **ALGORITMOS** (reconhecimento/geração) são **eficientes**.
- **APLICAÇÕES:** **especificação de linguagens de programação, analisadores sintáticos**, tradutores de linguagens e processadores de texto em geral; estruturação formal e análise computacional de linguagens naturais.

⇒ LINGUAGEM SENSÍVEL AO CONTEXTO

- **FORMALISMOS:** *máquinas de Turing com fita limitada, gramáticas sensíveis ao contexto.*

⇒ LINGUAGEM IRRESTRITA

- **FORMALISMOS:** *máquinas de Turing, gramáticas irrestritas*

- 
- **APLICAÇÕES:** "permitem explorar os limites da **capacidade de desenvolvimento de reconhecedores ou geradores de linguagens**, ou seja, estuda a solucionabilidade do problema da existência de algum reconhecedor ou gerador para determinada linguagem".



GRAMÁTICA

- Uma gramática é:
 - ✓ um sistema **gerador** de linguagens;
 - ✓ um sistema de reescrita;
 - ✓ um **dispositivo formal usado para especificar de maneira finita e precisa uma linguagem infinita.**

- **DEFINIÇÃO:** uma **gramática** define uma estrutura sobre um alfabeto de forma a permitir que apenas determinadas combinações de símbolos sejam consideradas sentenças.

Uma **gramática G** é definida como sendo uma quádrupla $G = (V_N, V_T, P, S)$, onde:

V_N é um conjunto finito de símbolos denominados **símbolos não-terminais**, usados na descrição da linguagem;

V_T é um conjunto finito de símbolos denominados **símbolos terminais**, ou seja, os símbolos propriamente ditos;

P é conjunto finito de pares (α, β) denominados **regras de produção** (ou regras gramaticais) que relacionam os símbolos terminais e não-terminais e são representadas por $\alpha ::= \beta$ ou $\alpha \rightarrow \beta$ (onde α e β são cadeias sobre V , com α envolvendo pelo menos um símbolo pertencente a V_N e β é eventualmente vazia);

S é o **símbolo inicial** da gramática a partir do qual as sentenças de uma linguagem podem ser geradas.

- **EXEMPLO:** a linguagem dos números inteiros sem sinal é gerada pela seguinte gramática G:

$$G = (V_N, V_T, P, S)$$

onde

$$V_N = \{N, D\}$$

$$V_T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$P = \left\{ \begin{array}{ll} N \rightarrow D N & \\ N \rightarrow D & \\ D \rightarrow 0 & D \rightarrow 5 \\ D \rightarrow 1 & D \rightarrow 6 \\ D \rightarrow 2 & D \rightarrow 7 \\ D \rightarrow 3 & D \rightarrow 8 \\ D \rightarrow 4 & D \rightarrow 9 \end{array} \right.$$

$$S = N$$

- **DERIVAÇÃO (REDUÇÃO):** a utilização de gramáticas pode ser formalizada por duas operações de substituição:
 - ✓ **derivação:** é a operação que consiste na substituição de uma sentença ou parte dela por outra de acordo com as regras de produção da gramática, no sentido $\text{SÍMBOLO INICIAL} \Rightarrow \text{SENTENÇA}$.
 - ✓ **redução:** é a operação que consiste na substituição de uma sentença ou parte dela por outro de acordo com as regras de produção da gramática, no sentido $\text{SENTENÇA} \Leftarrow \text{SÍMBOLO INICIAL}$.
 - ✓ sucessivos passos de derivação (redução) são definidos da seguinte forma:
 - $\alpha \Rightarrow_G \beta$ **derivação em um passo** ou direta: α deriva diretamente β , se e somente se $\alpha \rightarrow \beta \in P$;
 - $\alpha \Rightarrow_{G^*} \beta$ **derivação em zero ou mais passos:** α deriva em zero ou mais passos β , se e somente se existirem sequências $\alpha_1, \alpha_2, \alpha_3 \dots \alpha_n$ tais que $\alpha \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \alpha_3 \Rightarrow \dots \Rightarrow \alpha_n \Rightarrow \beta$;
 - $\alpha \Rightarrow_{G^+} \beta$ **derivação em um ou mais passos:** α deriva em um ou mais passos β , se e somente se for necessário pelo menos um passo na derivação.

A **derivação** é a operação adequada à **geração de sentenças**, enquanto **redução** é a operação adequada ao **reconhecimento de sentenças**.

Assim, pode-se definir formalmente uma **linguagem** gerada por uma gramática $G = (V_N, V_T, P, S)$, denotada por $L(G)$ como sendo: $L(G) = \{x \mid x \in V_T^* \text{ e } S \Rightarrow_{G^+} x\}$.

- **EQUIVALÊNCIA ENTRE GRAMÁTICAS:** $G_1 \equiv G_2 \leftrightarrow L(G_1) = L(G_2)$

$ \begin{array}{l} G_1 \\ V_N = \{S, A\} \\ V_T = \{a, b\} \\ P = \{ \\ \quad S \rightarrow a A \\ \quad S \rightarrow \varepsilon \\ \quad A \rightarrow b S \\ \quad \} \\ S = S \end{array} $	$ \begin{array}{l} G_2 \\ V_N = \{S\} \\ V_T = \{a, b\} \\ P = \{ \\ \quad S \rightarrow a b S \\ \quad S \rightarrow \varepsilon \\ \quad \} \\ S = S \end{array} $
--	---

Qual a linguagem gerada por G_1 ? Qual a linguagem gerada por G_2 ?

- **NOTAÇÃO**

- ✓ $\alpha \rightarrow \beta_1, \alpha \rightarrow \beta_2, \dots \alpha \rightarrow \beta_n$ podem ser abreviadas para $\alpha \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$
- ✓ os **símbolos não-terminais** serão sempre representados por letras maiúsculas {A, B, ... T};
- ✓ os **símbolos terminais** serão sempre representados por letras minúsculas {a, b, ... t}, dígitos, ou caracteres especiais;
- ✓ as **formas sentenciais** são compostas por **não-terminais e/ou terminais** e serão representadas por letras gregas $\{\alpha, \beta, \delta, \dots\}$;
- ✓ as **sentenças** são compostas por **terminais** e serão representadas por letras minúsculas {u, v, ... z}.

- **EXEMPLO:** a gramática G que gera a linguagem dos números inteiros sem sinal:

$$G = (V_N, V_T, P, S)$$

onde

$$V_N = \{N, D\}$$

$$V_T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$P = \left\{ \begin{array}{ll} N \rightarrow D N & \\ N \rightarrow D & \\ D \rightarrow 0 & D \rightarrow 5 \\ D \rightarrow 1 & D \rightarrow 6 \\ D \rightarrow 2 & D \rightarrow 7 \\ D \rightarrow 3 & D \rightarrow 8 \\ D \rightarrow 4 & D \rightarrow 9 \end{array} \right\}$$

$$S = N$$

pode ser escrita da seguinte forma:

$$\begin{array}{l} N \rightarrow D N \mid D \\ D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \end{array}$$

⇒ **GRAMÁTICA REGULAR (GR)**

- Uma **gramática regular** é tipo mais simples de gramática na hierarquia de Chomsky e pode ser utilizada para especificar a **parte léxica** de linguagens de programação.

- Uma **gramática regular** é uma quádrupla $G = (V_N, V_T, P, S)$, onde:

$$P = \{A \rightarrow w B \mid A \in V_N, w \in V_T^*, B \in (V_N \cup \{\varepsilon\})\}$$

Assim, toda regra de produção tem uma das seguintes formas:

$$\begin{aligned} A &\rightarrow w B \\ A &\rightarrow B w \\ A &\rightarrow w \end{aligned}$$

ou seja, uma gramática regular admite apenas regras de produção constituídas por **sequências de terminais** seguidas (precedidas) ou não por apenas **um não-terminal**.

- **EXEMPLOS:**

G_1

$$\begin{aligned} A &\rightarrow a B \\ B &\rightarrow b a B \mid \varepsilon \end{aligned}$$

G_2

$$A \rightarrow A b a \mid a$$

G_3

$$\begin{aligned} A &\rightarrow a B \\ B &\rightarrow b C \mid \varepsilon \\ C &\rightarrow a B \end{aligned}$$

G_4

$$\begin{aligned} A &\rightarrow B a \mid a \\ B &\rightarrow A b \end{aligned}$$

Quais as linguagens geradas por cada uma das gramáticas?

A gramática G que gera a linguagem dos números inteiros sem sinal é uma gramática regular?



GRAMÁTICA LIVRE DE CONTEXTO (GLC)

- Uma **gramática livre de contexto** é utilizada para especificar a **parte sintática** de linguagens de programação.
- Uma **gramática livre de contexto** é uma quádrupla $G = (V_N, V_T, P, S)$, onde:

$$P = \{A \rightarrow \alpha \mid A \in V_N, \alpha \text{ é uma sentença em } (V_N \cup V_T)^*\}$$

ou seja, uma gramática livre de contexto admite apenas regras de produção cujo o lado esquerdo contém exatamente um **não-terminal**.

- **EXEMPLO:**

$$S \rightarrow a S b \mid \varepsilon$$

Qual a linguagem gerada pela gramática acima?

A gramática G que gera a linguagem dos números inteiros sem sinal é livre de contexto?

- **livre de contexto** significa: "A deriva α sem depender ('livre') de qualquer análise dos símbolos que antecedem ou sucedem A ('contexto') na sentença que está sendo derivada".

⇒ **GRAMÁTICA SENSÍVEL AO CONTEXTO (GSC)**

- Uma **gramática sensível ao contexto** é uma quádrupla $G = (V_N, V_T, P, S)$, onde:
 $P = \{\alpha \rightarrow \beta \mid \alpha \text{ é uma sentença em } (V_N \cup V_T)^+ \text{ com no mínimo um não-terminal, } \beta \text{ é uma sentença em } (V_N \cup V_T)^*, |\alpha| \leq |\beta| \text{ exceto para } S \rightarrow \varepsilon, \text{ sendo que } S \text{ não pode estar presente no lado direito de nenhuma produção}\}$

ou seja, uma gramática sensível ao contexto admite regras de produção contendo não-terminais e terminais tanto do lado esquerdo como do lado de direito desde que a cada derivação o tamanho da sentença derivada não diminua, exceto quando a sentença vazia é gerada.

- **EXEMPLO:**

$$\begin{aligned} S &\rightarrow a S B C \mid a B C \\ B C &\rightarrow C B \\ C B &\rightarrow B C \\ B &\rightarrow b \\ C &\rightarrow c \end{aligned}$$

Qual a linguagem gerada pela gramática acima?

A gramática G que gera a linguagem dos números inteiros sem sinal é sensível ao contexto?

- **sensível ao contexto** significa: "o lado esquerdo das produções da gramática pode ser uma sentença de terminais e não-terminais, definindo um 'contexto' de derivação".

⇒ **GRAMÁTICA IRRESTRITA (GI)**

- Uma **gramática irrestrita** é uma quádrupla $G = (V_N, V_T, P, S)$, que não possui restrições quanto à forma das regras de produção.
- **EXEMPLO:** qualquer uma das gramáticas especificadas anteriormente é um exemplo de uma gramática irrestrita.