

Capítulo II – Gramáticas

II.1 – Motivação

- **O que é uma Gramática?**

- Um sistema gerador de linguagens;
- Um sistema de reescrita;
- Uma maneira finita de representar uma linguagem;
- Um dispositivo formal usado para especificar de maneira finita e precisa uma linguagem potencialmente infinita.

- **Qual a finalidade de uma gramática?**

- Definir o subconjunto de V^* que forma uma determinada linguagem.

- **Exemplo intuitivo de uma Gramática**

(um subconjunto da gramática da língua portuguesa)

$\langle \text{sentença} \rangle ::= \langle \text{sujeito} \rangle \langle \text{verbo} \rangle \langle \text{objeto} \rangle$

$\langle \text{sujeito} \rangle ::= \langle \text{substantivo} \rangle$

| $\langle \text{artigo} \rangle \langle \text{substantivo} \rangle$

| $\langle \text{artigo} \rangle \langle \text{adjetivo} \rangle \langle \text{substantivo} \rangle$

$\langle \text{substantivo} \rangle ::= \text{compilador} \mid \text{tradutor} \mid \text{codigo}$

| $\text{interpretador} \mid \text{programa} \mid \text{resultado}$

$\langle \text{artigo} \rangle ::= \text{o} \mid \text{um}$

$\langle \text{adjetivo} \rangle ::= \text{eficiente} \mid \text{bom} \mid \text{mau} \mid \text{melhor} \mid \text{pior}$

$\langle \text{verbo} \rangle ::= \text{gera} \mid \text{traduz} \mid \text{compila} \mid \text{interpreta}$

$\langle \text{objeto} \rangle ::= \langle \text{substantivo} \rangle$

| $\langle \text{artigo} \rangle \langle \text{substantivo} \rangle$

| $\langle \text{artigo} \rangle \langle \text{adjetivo} \rangle \langle \text{substantivo} \rangle$

| ϵ (* vazio *)

II.2 – Definição Formal de Gramática

$$G = (V_n, V_t, P, S)$$

onde:

V_n – conjunto finito de símbolos não-terminais.

V_t – conjunto finito de símbolos terminais.

convenções: $V_n \cap V_t = \varnothing$ e $V_n \cup V_t = V$

P – conjunto finito de pares (α, β) denominados produções (ou regras gramaticais ou de sintaxe).

$$P = \{ \alpha ::= \beta \mid \alpha \in V^* V_n V^* \wedge \beta \in V^* \}$$

S – símbolo $\in V_n$, é o símbolo inicial da gramática.

Exemplo: Formalizando o subconjunto da gramática da língua portuguesa apresentado, teríamos:

$G_{\text{portugues}} = (V_n, V_t, P, S)$, onde:

$V_n = \{ \langle \text{sentença} \rangle, \langle \text{sujeito} \rangle, \langle \text{substantivo} \rangle, \langle \text{artigo} \rangle, \langle \text{adjetivo} \rangle, \langle \text{verbo} \rangle, \langle \text{objeto} \rangle \}$

$V_t = \{ \text{compilador, tradutor, código, interpretador} \mid \text{programa, resultado, o, um, eficiente, bom, mau, melhor, pior, gera, traduz, compila, interpreta} \}$

P = é o conjunto das regras gramaticais apresentado

$S = \langle \text{sentença} \rangle$

Exemplos e exercícios:

1 - Construa gramáticas que representem as seguintes linguagens:

- a) Inteiros positivos menores que 1000
- b) Inteiros positivos
- c) Inteiros positivos pares
- d) Números reais

2 – Dado $V_T = \{ a, b \}$, construa uma gramática cuja linguagem gerada seja:

- a) O conjunto de sentenças cujo último símbolo seja igual o primeiro
- b) O conjunto de sentenças de tamanho ímpar
- c) O conjunto de sentenças com número par de b's
- d) O conjunto de sentenças com #a's divisível por 3
- e) O conjunto de sentenças que não possuam a's consecutivos
- f) O conjunto de sentenças onde o #a's seja igual o #b's e todos os a's precedam todos os b's
- g) O conjunto de sentenças onde o #a's = #b's

Notação a ser utilizada neste curso:

::= - ->

V_N – Letras de “A” a “T” e palavras escritas com letras maiúsculas

V_T – Letras de “a” a “t”, palavras escritas com letras minúsculas, dígitos e caracteres especiais

V_T^* - u, v, x, y, w, z

$\{V_N \cup V_T\}$ – U, V, X, Y, W, Z

$\{V_N \cup V_T\}^*$ - $\alpha, \beta, \gamma, \delta, \dots, \omega$ (exceto ϵ)

II.3 – Derivação e Redução

Definição: São operações de substituição que formalizam o uso de gramáticas

Seja $G = (V_n, V_t, P, S)$ uma gramática $\forall \alpha \gamma \in (V_n \cup V_t)^*$

- **Derivação em um passo (ou direta):**

$$\delta\alpha\gamma \Rightarrow \delta\beta\gamma \Leftrightarrow \alpha \rightarrow \beta \in P$$

- **Derivação em zero ou mais passos**

$$\alpha \stackrel{*}{\Rightarrow} \beta \Leftrightarrow \alpha \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_n \Rightarrow \beta; p/n \geq 0$$

obs. Se $\alpha \stackrel{*}{\Rightarrow} \beta$ em 0 (zero) passos, então $\alpha = \beta$.

- **Derivação em um ou mais passos**

$$\alpha \stackrel{+}{\Rightarrow} \beta \Leftrightarrow \alpha \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_n \Rightarrow \beta; p/n \geq 1$$

- **Redução em um passo (ou direta)**

$$\delta\alpha\gamma \Leftarrow \delta\beta\gamma$$

- **Redução em zero ou mais passos**

$$\delta\alpha\gamma \stackrel{*}{\Leftarrow} \delta\beta\gamma$$

- **Redução em um ou mais passos**

$$\delta\alpha\gamma \stackrel{+}{\Leftarrow} \delta\beta\gamma$$

II.4 - Sentença, Forma Sentencial e Linguagem

Seja $G = (V_n, V_t, P, S)$:

Sentença – seqüência de terminais produzida a partir do símbolo inicial de G

$$\mathbf{x} \mid \mathbf{x} \in V_t^* \wedge S \xRightarrow{+} \mathbf{x}$$

Forma Sentencial – seqüência de terminais e/ou não-terminais produzida a partir do símbolo inicial de G

$$\text{Se } S \Rightarrow \alpha \Rightarrow \beta \Rightarrow \dots \Rightarrow \gamma \Rightarrow \dots$$

então $\alpha, \beta, \dots, \gamma, \dots$ são formas sentenciais de G .

$$\alpha \mid \alpha \in V^* \wedge S \xRightarrow{*} \alpha$$

Linguagem – conjunto de sentenças derivadas a partir do símbolo inicial de G

$$L(G) = \{ \mathbf{x} \mid \mathbf{x} \in V_t^* \wedge S \xRightarrow{+} \mathbf{x} \}$$

Gramáticas Equivalentes – Duas gramáticas G_1 e G_2 são equivalentes entre si, se e somente se $L(G_1) = L(G_2)$.

$$G_1 \equiv G_2 \Leftrightarrow L(G_1) = L(G_2)$$

II.5 – Tipos de Gramáticas

(Classificação ou hierarquia de CHOMSKY)

Gramática Tipo 0:

(ou gramática sem restrições)

$G = (V_n, V_t, P, S)$, onde:

$$P = \{\alpha \rightarrow \beta \mid \alpha \in V^* V_n V^* \wedge \beta \in V^*\}$$

Gramática Tipo 1:

(ou Gramática Sensível ao Contexto – G.S.C.)

$G = (V_n, V_t, P, S)$, onde:

$$P = \{\alpha \rightarrow \beta \mid |\alpha| \leq |\beta|, \alpha \in V^* V_n V^* \wedge \beta \in V^+\}$$

Gramática Tipo 2:

(ou Gramática Livre de Contexto – G.L.C.)

$G = (V_n, V_t, P, S)$, onde:

$$P = \{A \rightarrow \beta \mid A \in V_n \wedge \beta \in V^+\}$$

Gramática Tipo 3:

(ou Gramática Regular – G.R.)

$G = (V_n, V_t, P, S)$, onde:

$$P = \{A \rightarrow a X \mid A \in V_n, a \in V_t \wedge X \in \{V_n \cup \{\epsilon\}\}\}$$

Observação: As linguagens representadas por G.S.C., G.L.C. e G.R. são denominadas, respectivamente, Linguagens Sensíveis ao Contexto (L.S.C.) Linguagens Livres de Contexto (L.L.C.) e Linguagens Regulares (L.R.).

II.6 – Sentença Vazia

Considerações:

1 - A motivação para o estudo de gramáticas foi à necessidade de se encontrar **representações finitas** para as linguagens.

2 - Se uma linguagem **L** possui uma **descrição finita**, então **$L_1 = L \cup \{\epsilon\}$** , também deveria possuir **descrição finita**.

3 - Pela definição dada, as G.S.C., G.L.C. e G.R. **não aceitam** produções da forma **$S \rightarrow \epsilon$** - logo, a **sentença vazia (ϵ)** não pode pertencer as L.S.C., L.L.C. ou L.R..

Redefinição de G.S.C. G.L.C. e G.R.

G.S.C., G.L.C. e G.R., podem ter a produção $S \rightarrow \epsilon$, desde que:

- 1 – **S seja o símbolo inicial da gramática;**
- 2 – **S não apareça no lado direito de nenhuma produção da gramática em questão.**

Observação: segundo esta redefinição, a produção $S \rightarrow \epsilon$ só poderá ser usada na derivação de ϵ (a sentença vazia).

Teorema II.1: “Se L é SC, LC ou REGULAR, então $L_1 = L \cup \{\epsilon\}$ e $L_2 = L - \{\epsilon\}$ serão do mesmo tipo”.

Lema II.1: “Se $G = (V_n, V_t, P, S)$ é uma GSC, então $\exists G_1$ SC | $L(G_1) = L(G) \wedge$ o símbolo inicial de G_1 não apareça no lado direito de nenhuma produção de G_1 ”.

Prova: Seja $G = (V_n, V_t, P, S)$ uma G.S.C.;

Seja $G_1 = (V_n \cup \{S_1\}, V_t, P_1, S_1)$, onde:

$P_1 = \{S_1 \rightarrow \alpha \mid S_1 \rightarrow \alpha \exists \leftrightarrow S \rightarrow \alpha \in P\} \cup P$ e

S_1 é o símbolo inicial de G_1 .

Para completar a prova, precisamos mostrar que $L(G_1) = L(G)$:

$$L(G_1) = L(G) \Leftrightarrow \begin{array}{l} 1 - L(G) \subseteq L(G_1) \\ 2 - L(G_1) \subseteq L(G) \end{array}$$

1 – Seja $w \in L(G) \therefore S \xRightarrow{+} w$

se $S \xRightarrow{+} w$

então $S \Rightarrow \alpha \xRightarrow{*} w$

se $S \Rightarrow \alpha$

então $S \rightarrow \alpha \in P$ e, por definição, $S_1 \rightarrow \alpha \in P_1$

Logo, $S_1 \Rightarrow \alpha \xRightarrow{*} w \therefore S_1 \xRightarrow{+} w \therefore L(G) \subseteq L(G_1)$

2 – Seja $w \in L(G_1) \therefore S_1 \xRightarrow{+} w$

se $S_1 \xRightarrow{+} w$

então $S_1 \Rightarrow \alpha \xRightarrow{*} w$

se $S_1 \Rightarrow \alpha$

então $S_1 \rightarrow \alpha \in P_1$ e, para que esta produção exista, P contem a produção $S \rightarrow \alpha$,

portanto:

$$S \Rightarrow \alpha \xRightarrow{*} w \therefore S \xRightarrow{+} w \therefore L(G_1) \subseteq L(G)$$

Conclusão: De 1 e 2, temos que $L(G_1) = L(G)$.

II.7 – Recursividade das G.S.C.

Definição: Uma gramática G é recursiva se existe um algoritmo que determine, para qualquer seqüência w , se w é ou não gerada por G .

Teorema II.2: “Se $G = (V_n, V_t, P, S)$ é uma G.S.C., então G é RECURSIVA”.

Prova: (* através de algoritmo *)

Seja $G = (V_n, V_t, P, S)$ uma G.S.C. | $S \rightarrow \varepsilon \notin P$;

Seja w uma seqüência $\forall |w| = n$;

Seja T_M o conjunto de formas sentenciais α |

$|\alpha| \leq N \wedge S \rightarrow \alpha$ em M passos.

Algoritmo II.1:

1 – $T_0 = \{ S \}$

2 – $M \leftarrow 1$

3 – $T_M \leftarrow T_{M-1} \cup \{ \alpha \mid \beta \rightarrow \alpha, \text{ onde } \beta \in T_{M-1} \wedge |\alpha| \leq n \}$

4 – se $T_M = T_{M-1}$

então fim

senão $M \leftarrow M+1$;

calcule novo T_M (volte ao passo 3).