



Gramática

Prof. Me. Stéfano Schwenck Borges Vale Vita

Material adaptado UEM-PR:

Prof. Yandre Maldonado e Gomes da Costa



Gramática

- Mecanismo gerador que permite definir formalmente uma linguagem;
- Através de uma gramática pode-se gerar todas as sentenças da linguagem definida por ela;
- Modelo muito aplicado na especificação de linguagens computacionais;



Gramática

- Formalmente, gramática é uma quádrupla $G = (V, T, P, S)$, onde:
 - **V** é um conjunto finito de símbolos não-terminais (ou variáveis);
 - **T** é um conjunto finito de símbolos terminais disjundo de V ;
 - **P** é um conjunto finito de pares, denominados regras de produção tal que a primeira componente é palavra de $(V \cup T)^+$ e a segunda componente é palavra de $(V \cup T)^*$;
 - **S** é um elemento de V , denominado símbolo inicial (ou símbolo de partida).



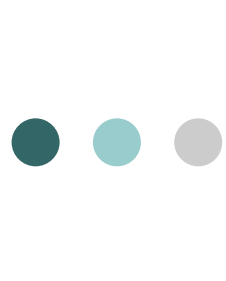
Gramática

- Os símbolos de T equivalem aqueles que aparecem nos programas de uma linguagem de programação. É o alfabeto em cima do qual a linguagem é definida;
- Os elementos de V são símbolos auxiliares que são criados para permitir a definição das regras da linguagem. Eles correspondem à “categorias sintáticas” da linguagem definida:
 - Português: sentença, predicado, verbo, ...;
 - Pascal: programa, bloco, procedimento, ...;



Gramática

- Uma regra de produção (α , β) é representada por $\alpha \rightarrow \beta$;
- As regras de produção definem as condições de geração das sentenças;
- A aplicação de uma regra de produção é denominada derivação;
- Uma regra $\alpha \rightarrow \beta$ indica que α pode ser substituído por β sempre que α aparecer;
- Enquanto houver símbolo não-terminal na cadeia em derivação, esta derivação não terá terminado;



Gramática

- O símbolo inicial é o símbolo através do qual deve iniciar o processo de derivação de uma sentença;
- Observações:
 - $V \cap T = \emptyset$
 - Os elementos de T são os terminais. Procuraremos representá-los por letras minúsculas (a, b, c, d, ...)
 - Os elementos de V são os não-terminais. Procuraremos representá-los por letras maiúsculas (A, B, C, D, ...)
 - As cadeias mistas, isto é, aquelas que contém símbolos de V e símbolos de T (cadeias pertencentes à $(V \cup T)^*$) serão representadas por letras gregas (α , β , γ , δ , ...)



Gramática

- Exemplo:

- $G_1 = (\{S, A, B\}, \{a, b\}, P, S)$ onde:
 - $P = \{$
 - 1) $S \rightarrow AB$
 - 2) $A \rightarrow a$
 - 3) $B \rightarrow b$ $\}$
- Quais as cadeias terminais geradas por esta gramática?

Gramática

○ Descrição de linguagens:

$G_1 = (\{A, B\}, \{a, b, c\}, P, A)$ onde:
 $P = \{$ 1) $A \rightarrow aB$
2) $B \rightarrow bB$
3) $B \rightarrow c$ $\}$

$G_2 = (\{S, A, B, C\}, \{a, b, c\}, P, S)$ onde:
 $P = \{$ 1) $S \rightarrow A$
2) $A \rightarrow BaC$
3) $B \rightarrow bB$
4) $B \rightarrow b$
5) $C \rightarrow cC$
6) $C \rightarrow c$ $\}$

- A linguagem gerada pela gramática G_1 descrita acima, poderia ser expressa da seguinte forma: $L(G_1) = \{ab^n c, n \geq 0\}$
- A linguagem gerada pela gramática G_2 descrita acima, poderia ser expressa da seguinte forma: $L(G_2) = \{b^m a c^n, m \geq 1, n \geq 1\}$



Gramática

- Geração direta (\Rightarrow):

- Considere $\alpha, \beta, \gamma, \delta \in (V \cup T)^*$

Uma cadeia $\alpha\gamma\beta$ gera diretamente (\Rightarrow)
uma cadeia $\alpha\delta\beta$ sse:

$$\gamma \rightarrow \delta \in P$$



Gramática

■ Geração (\Rightarrow^*):

– Considere $\alpha, \beta, \gamma, \delta \in (V \cup T)^*$

Uma cadeia α gera (\Rightarrow^*) uma cadeia β sse:

$\exists \gamma_1, \gamma_2, \dots, \gamma_n$ tal que

$$\alpha \Rightarrow \gamma_1 \Rightarrow \gamma_2 \Rightarrow \dots \Rightarrow \gamma_n \Rightarrow \beta \quad n \geq 0$$



Gramática

- Assim, em G_1 (slide 7) temos:
 - $S \Rightarrow^* ab$
 - $S \Rightarrow^* aB$
 - $AB \Rightarrow^* ab$
- E em G_2 , temos:
 - $\langle \text{sentença} \rangle \Rightarrow^* \text{o peixe mordeu a } \langle \text{subst.} \rangle$
 - $\langle \text{complemento} \rangle \Rightarrow^* \text{o } \langle \text{substantivo} \rangle$



Gramática

○ Definições:

- Forma sentencial: uma cadeia $\alpha \in (V \cup T)^*$ é uma forma sentencial de uma gramática sse $S \xRightarrow{*} \alpha$, ou seja, α é um “embrião” para alguma sentença gerada pela gramática, ou a própria sentença.
- Sentença: uma forma sentencial α , é uma sentença de G sse $\alpha \in T^*$. Portanto, as cadeias terminais geradas pela gramática são as sentenças de G .



Gramática

- Exemplo: dada a gramática

$G_4 = (\{A, B\}, \{a, b, c\}, P, A)$ onde:

$P = \{ 1) A \rightarrow aB$

$2) B \rightarrow bB$

$3) B \rightarrow c \quad \}$

Indique se as seguintes cadeias são sentenças ou formas sentenciais produzidas por G_4 :

	Sentença	Forma Sentencial
abc		
A		
a		
abB		
abbbB		
abbbc		



Tipos de Gramática

- Tipo 1 - exemplo:

$S \rightarrow aSBC$

$S \rightarrow aBC$

$CB \rightarrow BC$

$aB \rightarrow ab$

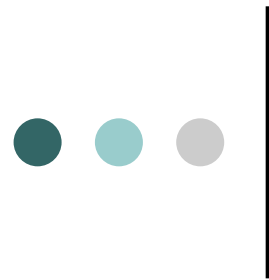
$bB \rightarrow bb$

$bC \rightarrow bc$

$cC \rightarrow cc$

- Qual é a linguagem gerada por esta gramática?

$L(G) = \{a^n b^n c^n | n > 0\}$



Tipos de Gramática

- Tipo 2 - exemplo 1:

$$S \rightarrow AB$$

$$A \rightarrow 0A11$$

$$A \rightarrow \lambda$$

$$B \rightarrow 0B$$

$$B \rightarrow \lambda$$

- Qual é a linguagem gerada por esta gramática?

$$L(G) = \{0^n 1^{2n} 0^m \mid n \geq 0, m \geq 0\}$$



Tipos de Gramática

- Tipo 2 – exemplo 2:

$S \rightarrow aB$

$S \rightarrow bA$

$A \rightarrow a$

$A \rightarrow aS$

$A \rightarrow bAA$

$B \rightarrow b$

$B \rightarrow bS$

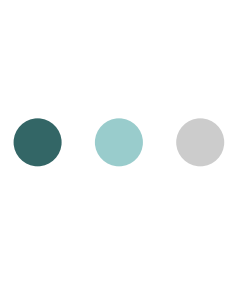
$B \rightarrow aBB$

- Qual é a linguagem gerada por esta gramática?

$L(G) = \{w \in \{a,b\}^+ \mid w \text{ contém número de } a\text{'s igual ao número de } b\text{'s}\}$

ou

$L(G) = \{w \in \{a,b\}^+ \mid |w|_a = |w|_b\}$



Tipos de Gramática

- Tipo 3 - exemplo 3:

$$N \rightarrow +D|-D$$
$$D \rightarrow 0|1|2|3|4|5|6|7|8|9|1E|2E|3E|4E|5E|6E|7E|8E|9E$$
$$E \rightarrow 0|1|2|3|4|5|6|7|8|9|0E|1E|2E|3E|4E|5E|6E|7E|8E|9E$$

- Números inteiros com sinal sem ocorrência de zeros à esquerda.

Descrição simplificada do Pascal em EBNF:

```
<programa> ::= PROGRAM <identificador> ; <bloco> .
<bloco> ::= <declarações> BEGIN <comando> END | BEGIN <comando> END
<comando> ::= <variável> := <expressão> ; <comando>
               | READ <variável> ; <comando>
               | WRITE <expressão> ; <comando>
               | IF <expressão> THEN <comando> ELSE <comando>
               | FOR <identificador> := <expressão> TO <expressão> DO <comando>
               | BEGIN <comando> END
<declarações> ::= VAR <lista_identif.> : <tipo>;
               | TYPE <identificador> = <tipo> ;
<expressão>   ::= <variável>
               | <identificador>
               | <constante>
               | <expressão> <operador binário> <expressão>
<variável>    ::= <identificador>
<operador_bin> ::= + | - | * | / | = | > | < | <> | <= | >=
<tipo>        ::= INTEGER
               | REAL
               | STRING
<constante>   ::= <identificador> | <número>
<lista_identif.> ::= <identificador> { , <identificador> }
<identificador> ::= <letra> { <letra> | <dígito> }
<número>       ::= <dígito> { <dígito> }
<dígito>       ::= 0|1|2|3|4|5|6|7|8|9
<letra>        ::= a|b|c|d|e|f ... x|y|z|A|B|C|D ... X|Y|Z
```