

# Linguagens Formais e Autômatos

## Linguagem de Programação Utilizando Gramáticas Livres de Contexto

**Willian F. H. Schneider**  
**201710091**

As Gramáticas Livres de Contexto (GLCs) são empregadas para descrever linguagens que seguem estruturas livres de contexto. Uma GLC consiste em um conjunto de regras recursivas utilizadas para criar padrões de cadeias de caracteres. Essas gramáticas têm a capacidade de representar todas as linguagens regulares e mais algumas, embora não sejam capazes de abranger todas as linguagens possíveis.

O estudo das gramáticas livres de contexto é realizado em diversas áreas, incluindo ciência da computação teórica, design de compiladores e linguística. Elas desempenham um papel fundamental na descrição de linguagens de programação, e é possível gerar automaticamente programas de análise sintática em compiladores a partir dessas gramáticas.

As Gramáticas Livres de Contexto têm a capacidade de gerar linguagens livres de contexto. Esse processo é alcançado por meio da utilização de um conjunto de variáveis definidas recursivamente, inter-relacionadas por meio de um conjunto de regras de produção. A denominação "livre de contexto" é atribuída a essas gramáticas porque qualquer uma das regras de produção pode ser aplicada de forma independente do contexto, sem depender de quaisquer outros símbolos que possam ou não estar próximos a um símbolo específico ao qual a regra está sendo aplicada.

Uma Gramática Livre de Contexto  $\rightarrow G$ , é definida por quatro tuplas  $(V, \Sigma, R, S)$ , onde:

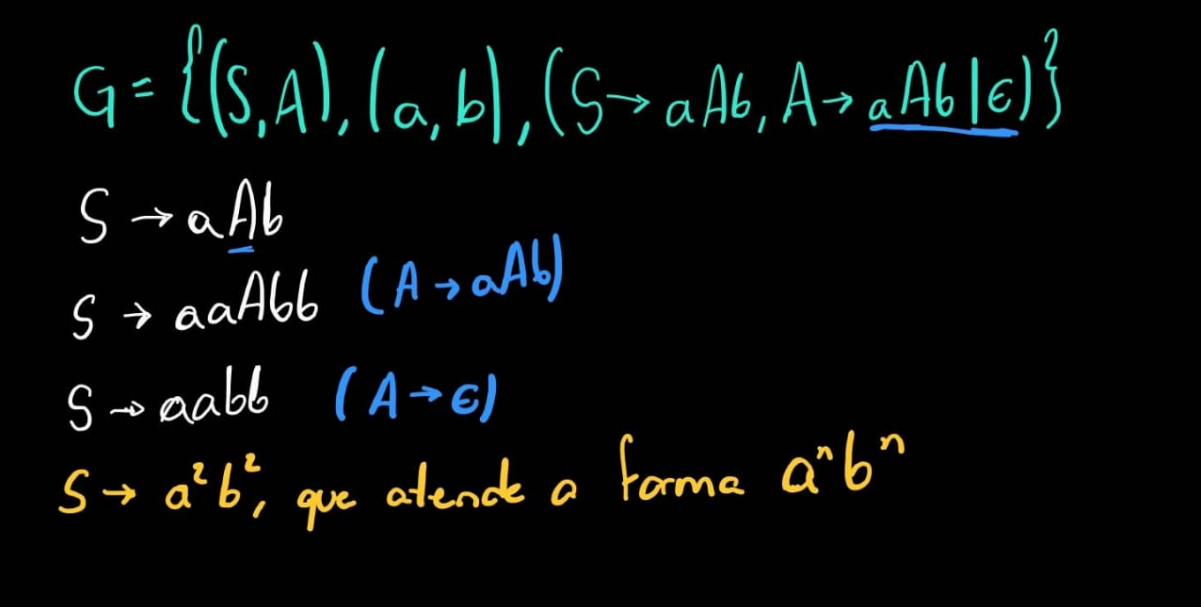
$V \rightarrow$  É uma coleção de variáveis ou símbolos não-terminais.

$\Sigma \rightarrow$  É um conjunto de símbolos terminais. O conjunto de terminais é o alfabeto da língua definido pela gramática  $G$ .

$R \rightarrow$  É um conjunto de regras de produção, onde cada regra de produção mapeia uma variável para uma *string*.

$S \rightarrow$  É o símbolo inicial usado para representar a frase inteira (ou programa).

O exemplo abaixo demonstra a geração de uma linguagem que possui o mesmo número de letras a e b, na forma de  $a^n b^n$ :



The image shows a handwritten derivation of the language  $a^n b^n$  using a context-free grammar. The grammar is defined as  $G = \{(S, A), (a, b), (S \rightarrow aAb, A \rightarrow aAb | \epsilon)\}$ . The derivation steps are as follows:

$$G = \{(S, A), (a, b), (S \rightarrow aAb, A \rightarrow aAb | \epsilon)\}$$
$$S \rightarrow aAb$$
$$S \rightarrow aaAbb \quad (A \rightarrow aAb)$$
$$S \rightarrow aabb \quad (A \rightarrow \epsilon)$$
$$S \rightarrow a^2 b^2, \text{ que atende a forma } a^n b^n$$

Como descrito no livro “Compiladores: Princípios, Técnicas e Ferramentas” (2008), muitas construções de linguagens de programação possuem uma estrutura inerentemente recursiva que pode ser identificada por gramáticas livres de contexto. Por exemplo, poderíamos ter um enunciado condicional que não pode ser especificado usando uma notação para expressões regulares, tal como:

se  $S_1$  e  $S_2$  são enunciados e  $E$  é uma expressão, então  
“if  $E$  then  $S_1$  else  $S_2$ ” é um enunciado.