

## Compiladores – Análise Sintática (Introdução e Top-Down)

Carlos Henrique Vieira Marques Veeck

Baseado na apresentação de Leopoldo Teixeira (IF688)

---

### O que é Análise Sintática

- Recebe tokens do analisador léxico (lexer).
- Constrói uma árvore sintática (parse tree).
- Verifica se a estrutura do programa segue a gramática da linguagem.

Resultados possíveis:

- Correto: string está bem formada.
  - Incorreto: há erro de sintaxe.  
Obs: Mesmo com sintaxe correta, podem existir erros semânticos.
- 

### Gramáticas

- Definidas por regras de produção.
- A linguagem gerada por uma gramática  $G$  é chamada  $L(G)$ .
- Usadas para descrever estruturas sintáticas (expressões, comandos etc).

Exemplo de gramática:

$\text{exp} \rightarrow \text{exp} + \text{exp} \mid \text{exp} - \text{exp} \mid \text{digit}$

$\text{digit} \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

---

### Expressões Regulares vs. Gramáticas

Expressões Regulares:

- Usadas na análise léxica.
- Mais simples e eficientes.

- Descrevem padrões lineares.

Gramáticas Livres de Contexto (GLC):

- Usadas na análise sintática.
  - Lidam com estruturas aninhadas (como parênteses, blocos, etc).
  - Mais poderosas e expressivas.
- 

### **Derivação**

- Processo de gerar strings a partir do símbolo inicial.
  - Substitui não-terminais pelas produções.
  - Resulta em uma string da linguagem e pode ser usada para construir a árvore sintática.
- 

### **Parsing (Análise Sintática)**

- Recebe uma string e constrói a árvore sintática correspondente.
- Inverso da derivação.

Complexidade:

- Teórica:  $O(n^3)$
  - Na prática: parsing pode ser feito em tempo linear ( $O(n)$ ).
- 

### **Tipos de Parsing**

1. Métodos universais: funcionam para qualquer gramática, mas são lentos.
2. Top-down: constroem a árvore da raiz para as folhas.
3. Bottom-up: constroem a árvore das folhas para a raiz.

---

## Parsing Top-Down

- Começa do símbolo inicial.
- Lê tokens da esquerda para a direita.
- Decide qual produção usar com base no próximo token.

Resultados:

- Sucesso: todos os símbolos da árvore são terminais e tokens foram consumidos.
- Erro: não há correspondência entre entrada e a árvore parcialmente construída.

---

## Problemas de Parsing Top-Down

- Pode exigir backtracking (voltar e tentar outra produção).
- Isso causa ineficiência.

---

## Parsing Preditivo

- Tipo de top-down que **evita backtracking**.
- O primeiro token já indica qual produção aplicar.