

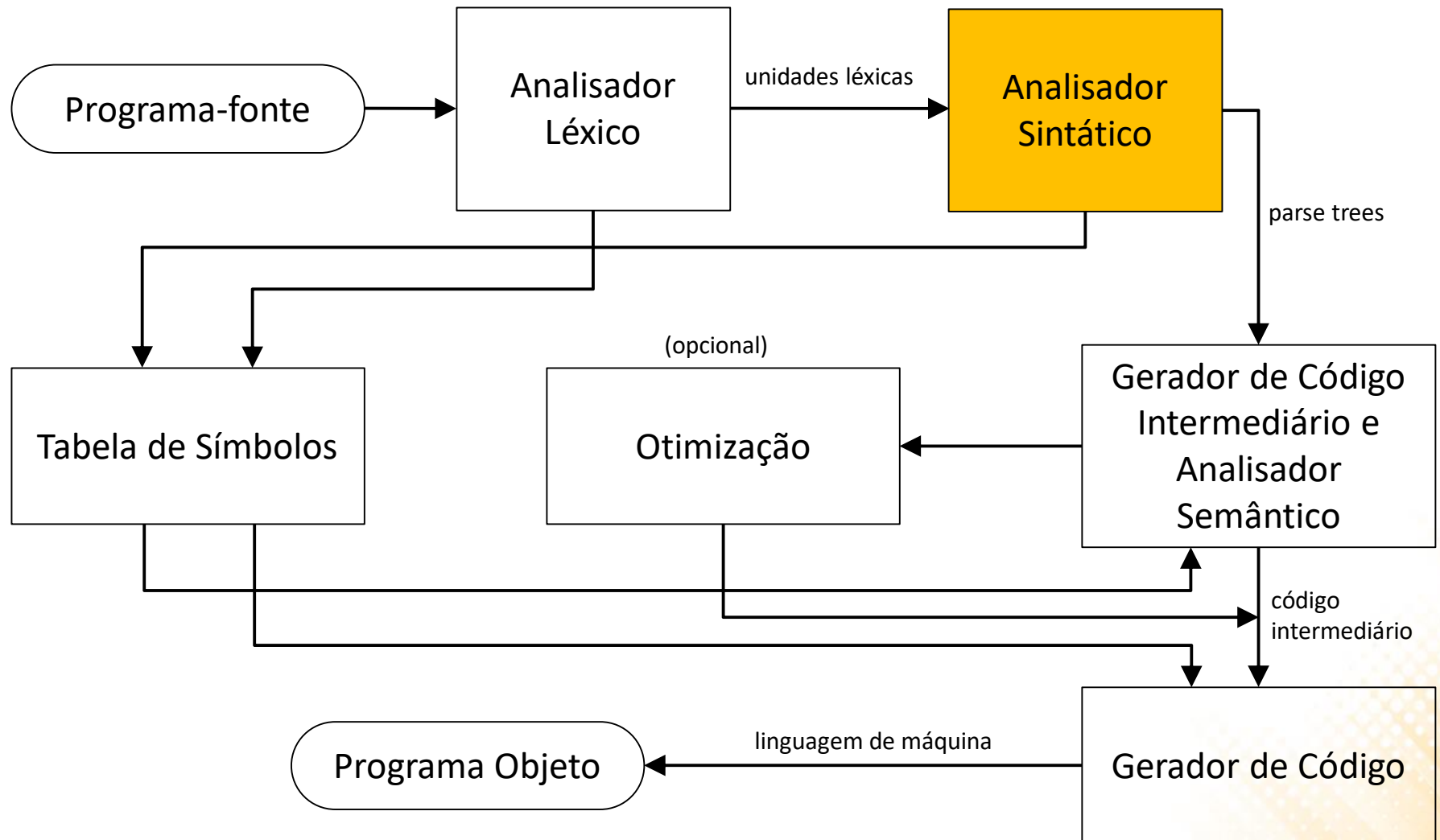
Compiladores

Aula 06 – Construção de Árvores Sintáticas Abstratas (Implementação)

Edirlei Soares de Lima

<edirlei.lima@universidadeeuropeia.pt>

Processo de Compilação



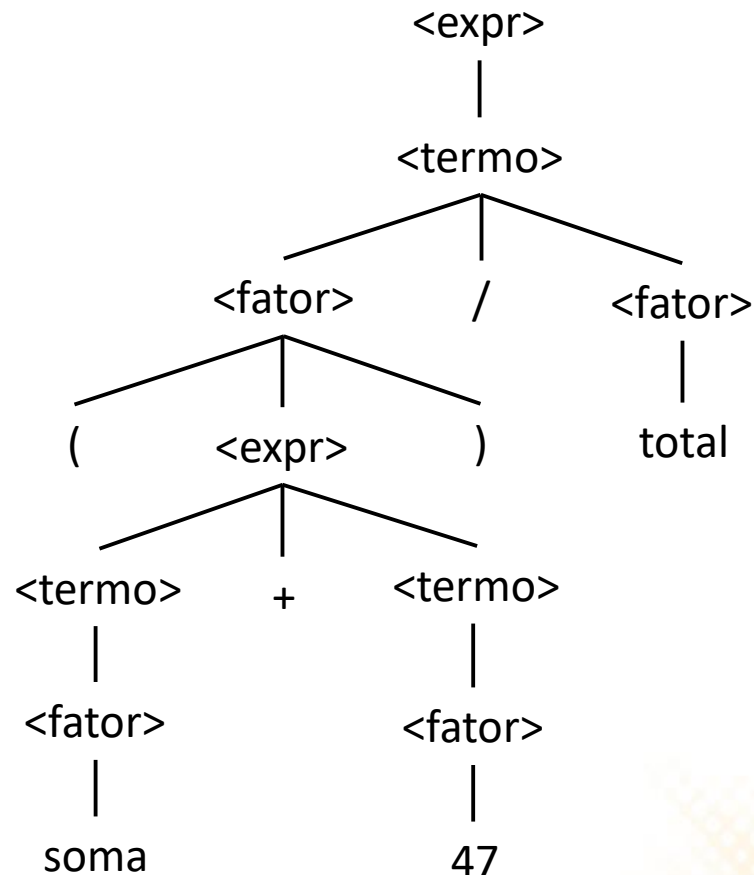
Análise Sintática

- Os analisadores sintáticos constroem árvores de análise sintática (**parse trees**) para os programas dados.
 - Em alguns casos, a parse tree é construída implicitamente (apenas o resultado de percorrer a árvore é gerado);
- Objetivos da Análise Sintática:
 - Verificar o programa de entrada para determinar se ele está sintaticamente correto;
 - Produzir árvores de análise (parse trees).

Análise Sintática

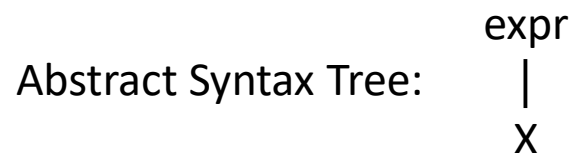
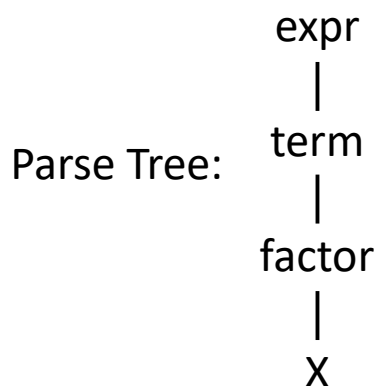
- Saída:

```
Token: 25, Lexema: (  
Enter <expr>: 25  
Enter <termo>: 25  
Enter <fator>: 25  
Token: 11, Lexema: soma  
Enter <expr>: 11  
Enter <termo>: 11  
Enter <fator>: 11  
Token: 21, Lexema: +  
Exit <fator>: 11  
Exit <termo>: 21  
Token: 10, Lexema: 47  
Enter <termo>: 10  
Enter <fator>: 10  
Token: 26, Lexema: )  
Exit <fator>: 10  
Exit <termo>: 26  
Exit <expr>: 11  
Token: 24, Lexema: /  
Exit <fator>: 25  
Token: 11, Lexema: total  
Enter <fator>: 11  
Token: -1, Lexema: EOF  
Exit <fator>: 11  
Exit <termo>: -1  
Exit <expr>: 25
```



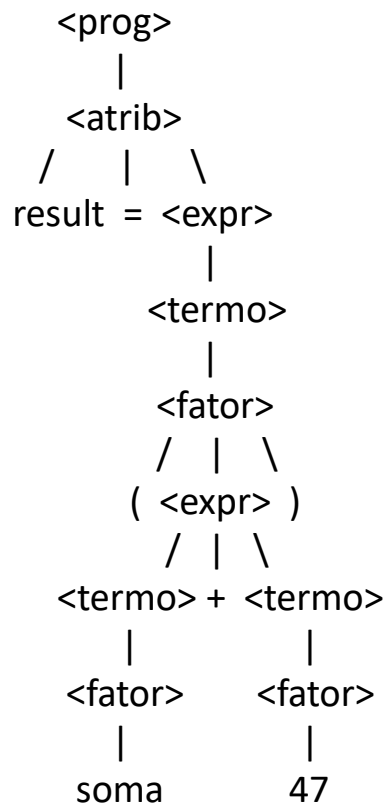
Árvore de Sintaxe

- Após a análise, os **detalhes de derivação** não são necessários para fases subsequentes do processo de compilação.
- O Analisador Semântico remove as produções intermediárias para criar uma **árvore sintática abstrata** (*abstract syntax tree*).



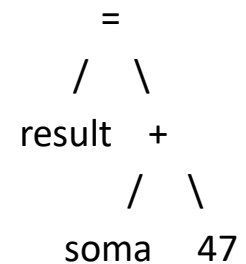
Árvore Sintática Abstrata

Árvore Sintática:



result = (soma + 47)

Árvore Sintática Abstrata:



Árvore Sintática Abstrata

- Para transformar uma árvore sintática em uma árvore sintática abstrata, é necessário:
 1. Remover nós que possuem um único filho. A árvore sintática abstrata nunca contém sequências de nós com apenas um filho;
 2. Remover detalhes sintáticos (como parênteses, vírgula, e ponto e vírgula);
 3. Os operadores (como +, -, x, e /) devem ser transformados em nós internos na árvore abstrata.

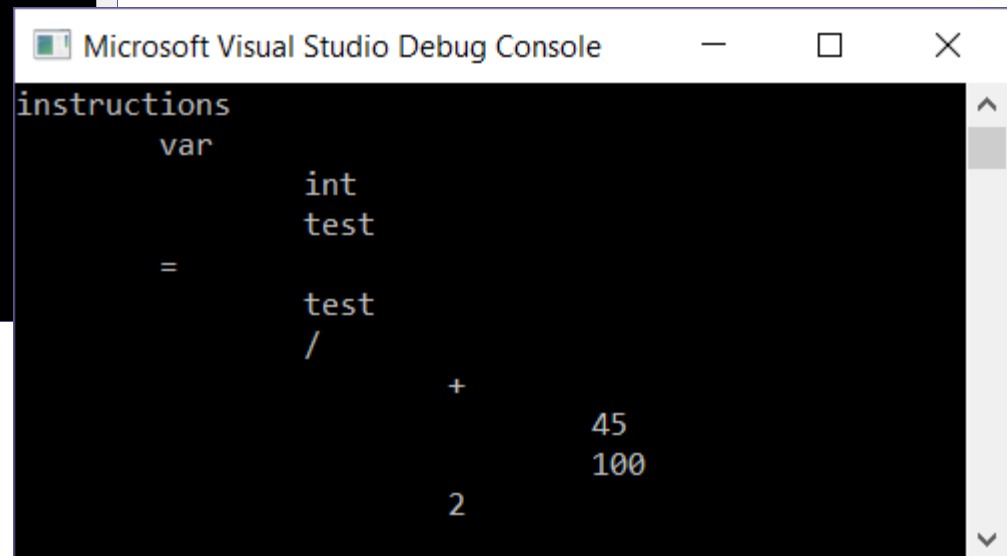
Construção da Árvore Sintática Abstrata (Implementação)

```
Node* CSTtoAST(Node *current)
{
    int i = 0;
    int x = 0;
    if (current == NULL)
        return NULL;
    while (current->children[i] != NULL)
    {
        current->children[i] = CSTtoAST(current->children[i]);
        i++;
    }
    ...
    if (i == 1)
    {
        if (current->type == -1)
        {
            Node* node = current->children[0];
            free(current);
            return node;
        }
    }
    ...
}
```


Construção da Árvore Sintática Abstrata (Implementação)

- Entrada:

```
int test;  
test = (45 + 100)/2;
```



Leitura Complementar

- Aho, A. V., Lam, M. S., Jeffrey, R. S.
Compiladores: Princípios, Técnicas e Ferramentas. 2ª edição, Pearson, 2007.
ISBN: 978-8588639249.
 - Capítulo 3: Lexical Analysis
 - Capítulo 4: Syntax Analysis
- Sebesta, R. W. **Conceitos de Linguagens de Programação.** 9ª edição Editora Bookman, 2011. ISBN: 978-8577807918.
 - Capítulo 3: Descrevendo a Sintaxe e a Semântica
 - Capítulo 4: Análise Léxica e Sintática

