

Curso: Engenharia de Computação

Linguagens Formais e Compiladores

Prof. Clayton J A Silva, MSc
clayton.silva@professores.ibmec.edu.br



A descrição geral da sintaxe de LP

Árvores de análise sintática

Seja uma gramática para definir uma atribuição simples

Exemplo 3.2 Uma gramática para sentenças de atribuição simples.

$\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

$\langle \text{id} \rangle \rightarrow A \mid B \mid C$

$\langle \text{expr} \rangle \rightarrow \langle \text{id} \rangle + \langle \text{expr} \rangle$

$\mid \langle \text{id} \rangle * \langle \text{expr} \rangle$

$\mid (\langle \text{expr} \rangle)$

$\mid \langle \text{id} \rangle$

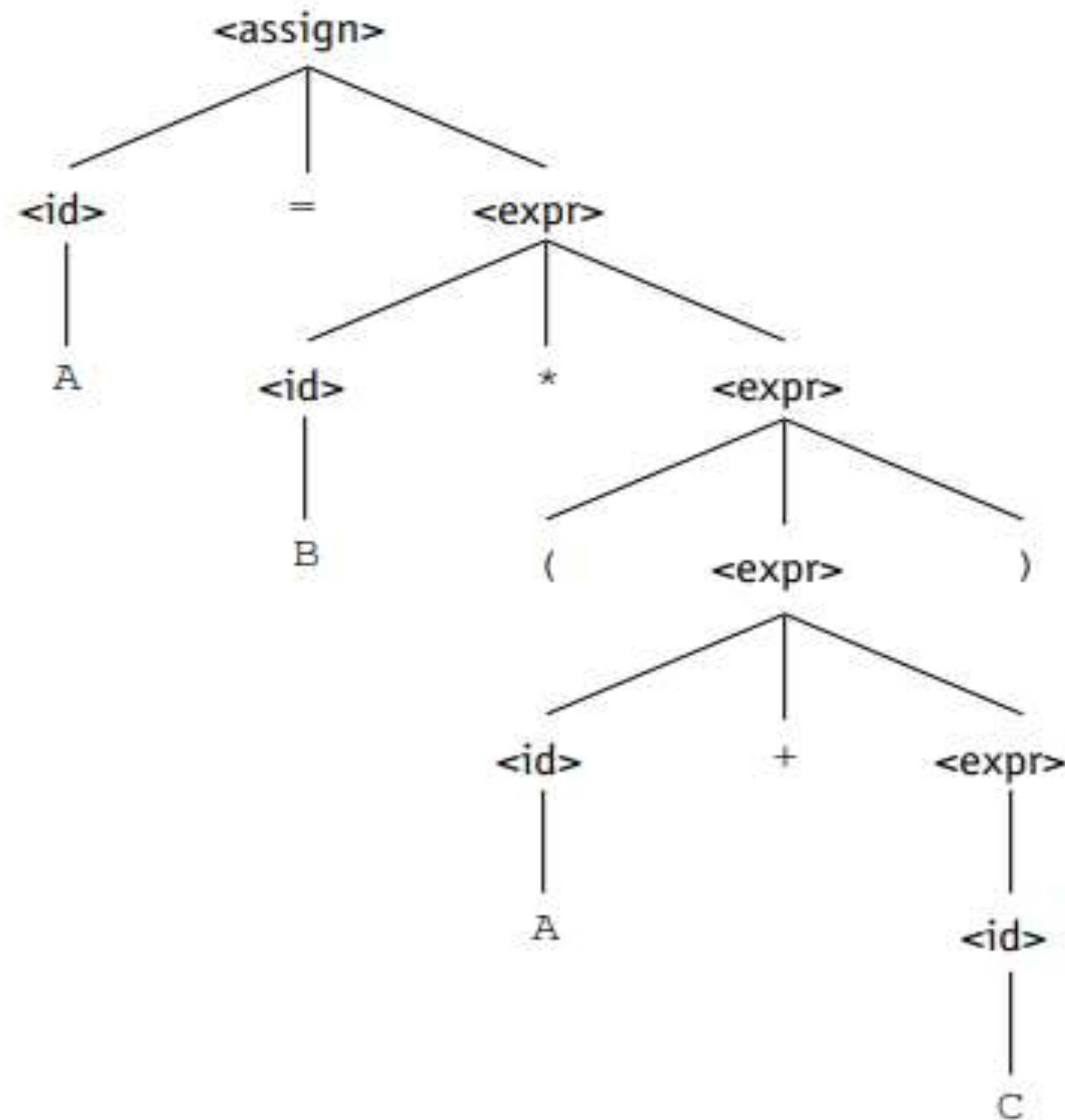
Árvores de análise sintática

...que gera a derivação mais à esquerda

$\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\langle \text{id} \rangle \rightarrow A \mid B \mid C$
 $\langle \text{expr} \rangle \rightarrow \langle \text{id} \rangle + \langle \text{expr} \rangle$
 $\mid \langle \text{id} \rangle * \langle \text{expr} \rangle$
 $\mid (\langle \text{expr} \rangle)$
 $\mid \langle \text{id} \rangle$

$\langle \text{assign} \rangle \Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\Rightarrow A = \langle \text{expr} \rangle$
 $\Rightarrow A = \langle \text{id} \rangle * \langle \text{expr} \rangle$
 $\Rightarrow A = B * \langle \text{expr} \rangle$
 $\Rightarrow A = B * (\langle \text{expr} \rangle)$
 $\Rightarrow A = B * (\langle \text{id} \rangle + \langle \text{expr} \rangle)$
 $\Rightarrow A = B * (A + \langle \text{expr} \rangle)$
 $\Rightarrow A = B * (A + \langle \text{id} \rangle)$
 $\Rightarrow A = B * (A + C)$

$\langle \text{assign} \rangle \Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\Rightarrow A = \langle \text{expr} \rangle$
 $\Rightarrow A = \langle \text{id} \rangle * \langle \text{expr} \rangle$
 $\Rightarrow A = B * \langle \text{expr} \rangle$
 $\Rightarrow A = B * (\langle \text{expr} \rangle)$
 $\Rightarrow A = B * (\langle \text{id} \rangle + \langle \text{expr} \rangle)$
 $\Rightarrow A = B * (A + \langle \text{expr} \rangle)$
 $\Rightarrow A = B * (A + \langle \text{id} \rangle)$
 $\Rightarrow A = B * (A + C)$



Árvores de análise sintática

As árvores geram de forma fácil as derivações das sentenças de uma linguagem

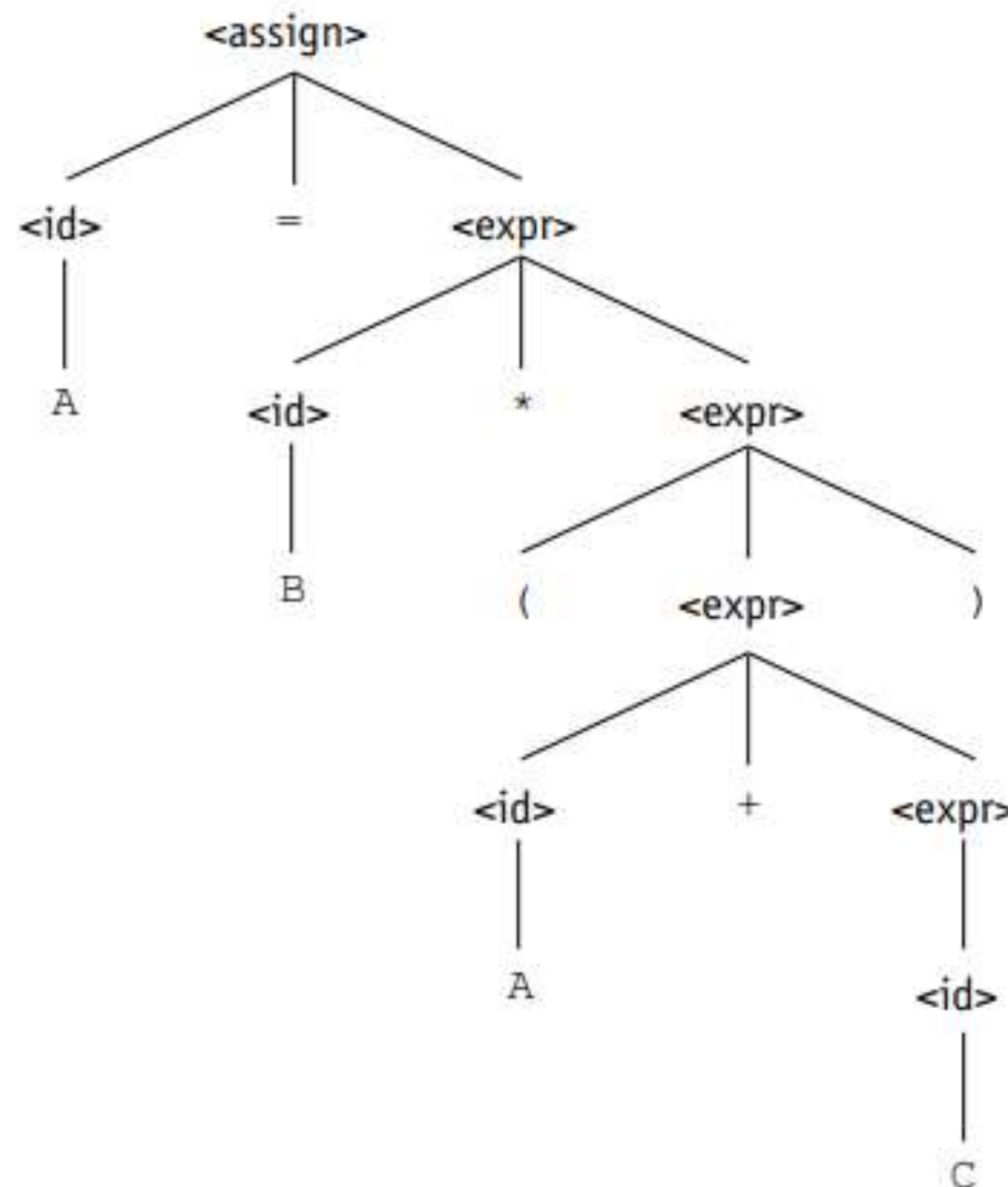
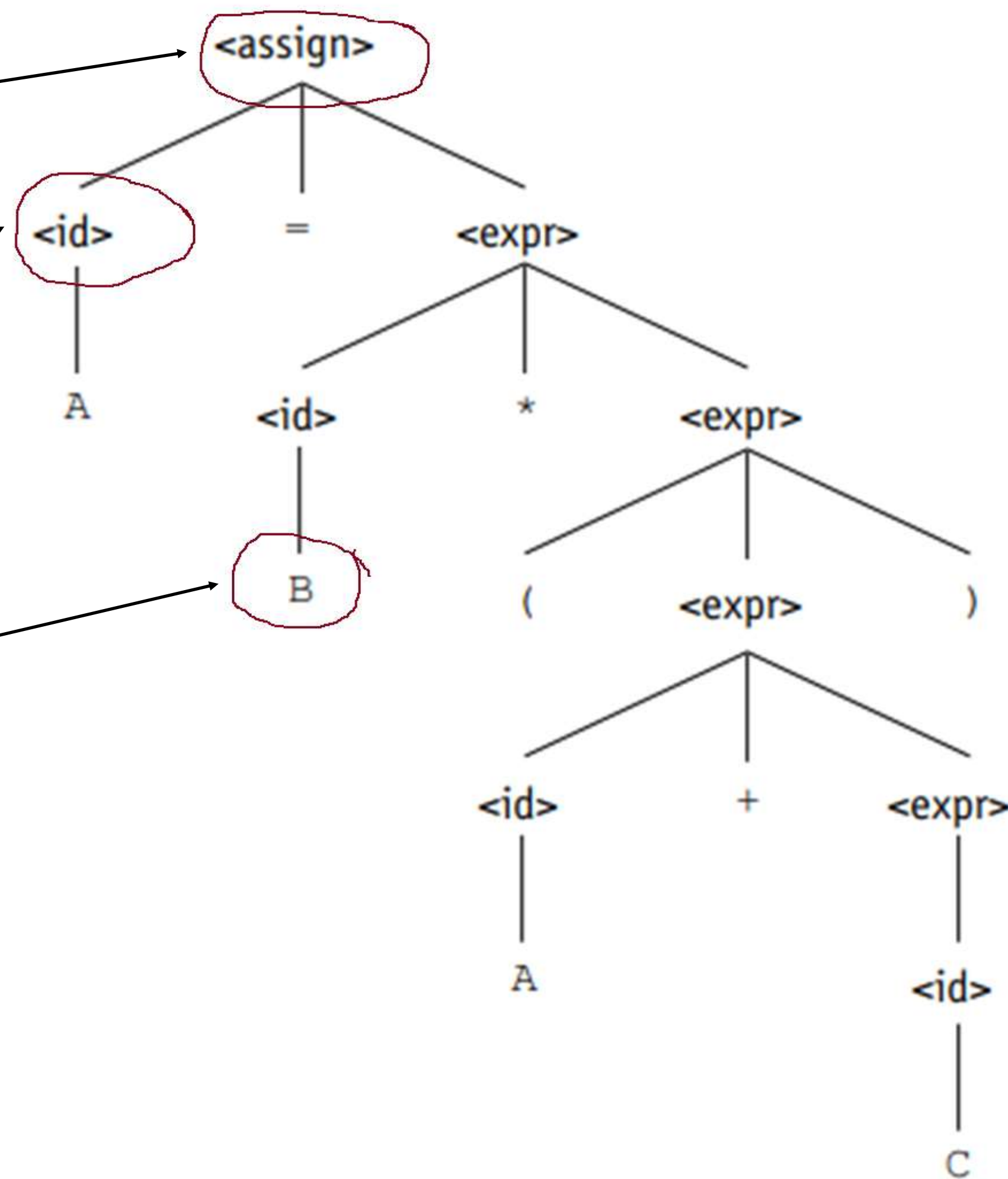


FIGURA 3.1

Uma árvore de análise sintática para a sentença simples $A = B * (A + C)$.

- A árvores de análise mostram a **derivação** da regra de atribuição
- A **raiz** contém a abstração a ser representada
- Os **ramos** são definidos pelos **vértices internos** ou **nós**, rotulados pelos símbolos não terminais
- As **folhas** de cada ramo sempre serão rotuladas pelos símbolos terminais

**FIGURA 3.1**

Uma árvore de análise sintática para a sentença simples $A = B * (A + C)$.

Ambiguidade

Uma gramática é dita **ambígua** quando gera uma sentença que admite **mais de uma árvore sintática** para a mesma sentença (*)

Exemplo 3.3 Uma gramática ambígua para sentenças de atribuição simples.

$$\begin{aligned}\langle \text{assign} \rangle &\rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle \\ \langle \text{id} \rangle &\rightarrow A \mid B \mid C \\ \langle \text{expr} \rangle &\rightarrow \langle \text{expr} \rangle + \langle \text{expr} \rangle \\ &\mid \langle \text{expr} \rangle * \langle \text{expr} \rangle \\ &\mid (\langle \text{expr} \rangle) \\ &\mid \langle \text{id} \rangle\end{aligned}$$

Uma sentença pode ter mais de uma **derivação**, ainda assim pertencer a uma gramática não ambígua

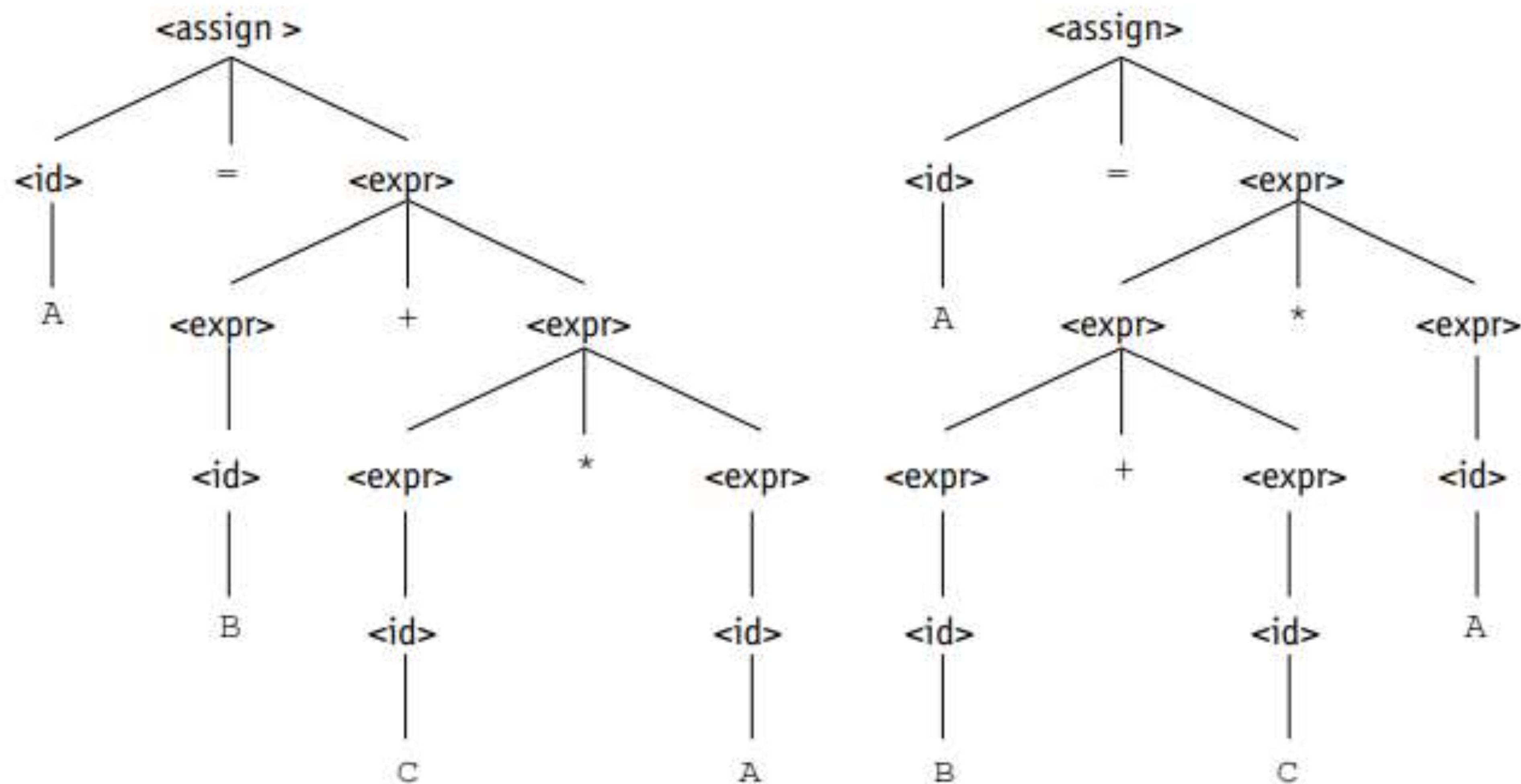


FIGURA 3.2

Duas árvores diferentes de análise sintática para a mesma sentença, $A = B + C * A$.

Precedência entre operadores

Quem eu resolvo primeiro? $B + C$ ou $C * A$?

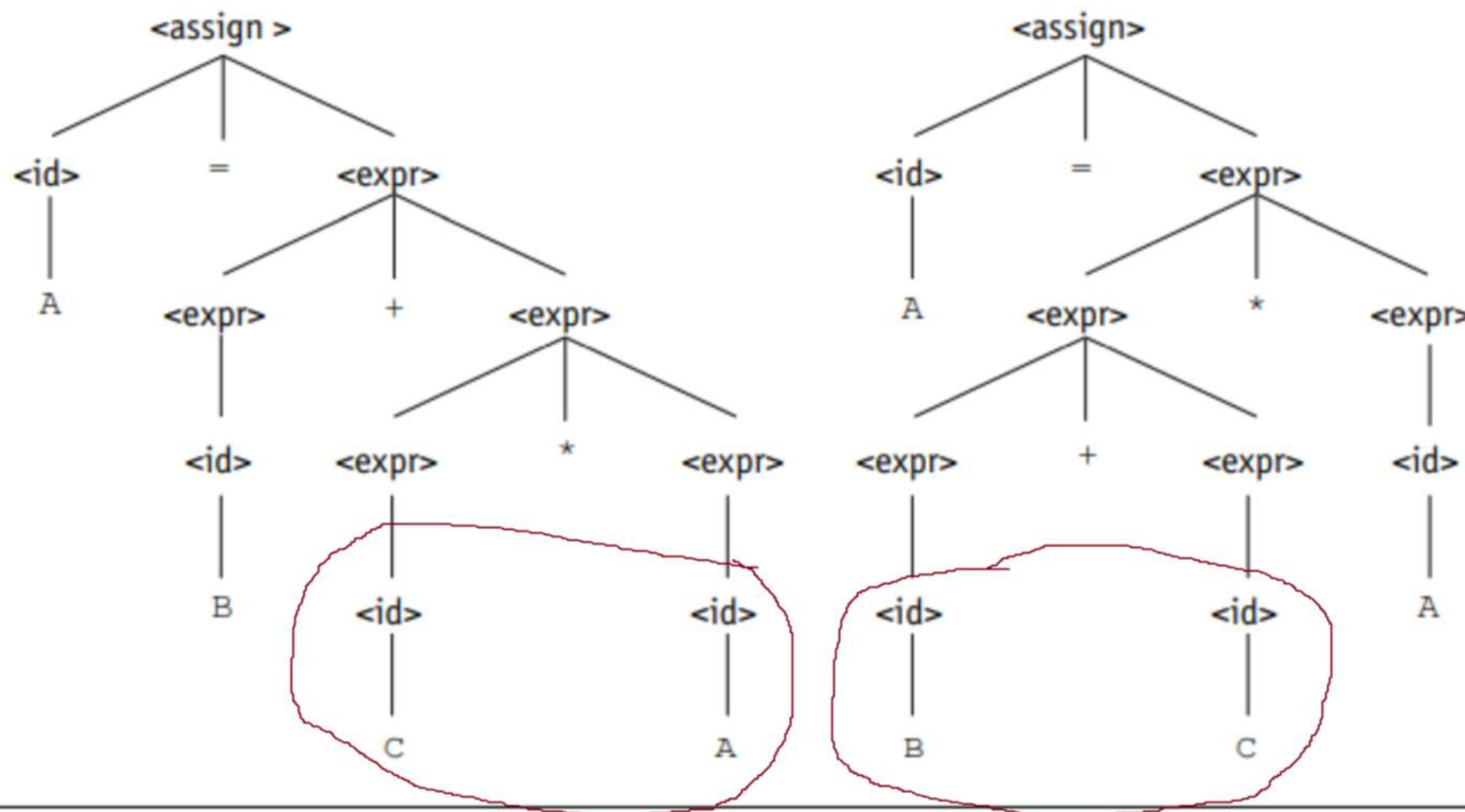


FIGURA 3.2

Duas árvores diferentes de análise sintática para a mesma sentença, $A = B + C * A$.

Precedência entre operadores

- Uma gramática pode ser escrita de modo a evoluir as folhas do nível mais alto para um nível inferior, da esquerda para a direita
- Uma forma de estabelecer precedência para que não haja ambiguidade é **estabelecer abstrações diferentes para operadores de precedência diferente**
- A ordem correta é especificada por meio de símbolos não terminais separados

Precedência entre operadores

Exemplo 3.4 Uma gramática não ambígua para expressões.

$\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

$\langle \text{id} \rangle \rightarrow A \mid B \mid C$

$\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle$
 $\mid \langle \text{term} \rangle$

$\langle \text{term} \rangle \rightarrow \langle \text{term} \rangle * \langle \text{factor} \rangle$
 $\mid \langle \text{factor} \rangle$

$\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle)$
 $\mid \langle \text{id} \rangle$

Precedência entre operadores

A derivação da sentença $A = B + C * A$

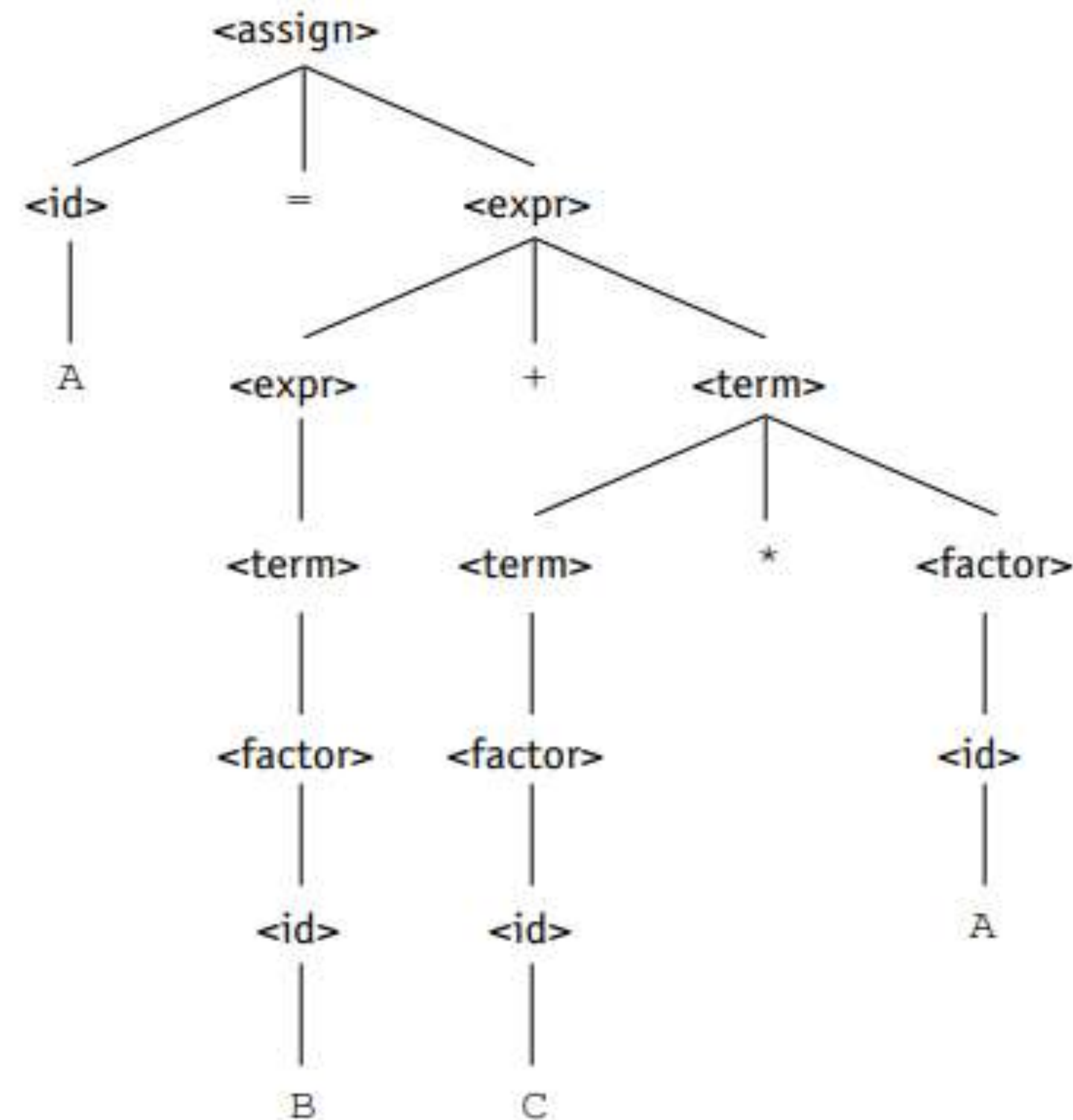
$\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\langle \text{id} \rangle \rightarrow A \mid B \mid C$
 $\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle$
 $\quad \mid \langle \text{term} \rangle$
 $\langle \text{term} \rangle \rightarrow \langle \text{term} \rangle * \langle \text{factor} \rangle$
 $\quad \mid \langle \text{factor} \rangle$
 $\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle)$
 $\quad \mid \langle \text{id} \rangle$

$\langle \text{assign} \rangle \Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\Rightarrow A = \langle \text{expr} \rangle$
 $\Rightarrow A = \langle \text{expr} \rangle + \langle \text{term} \rangle$
 $\Rightarrow A = \langle \text{term} \rangle + \langle \text{term} \rangle$
 $\Rightarrow A = \langle \text{factor} \rangle + \langle \text{term} \rangle$
 $\Rightarrow A = \langle \text{id} \rangle + \langle \text{term} \rangle$
 $\Rightarrow A = B + \langle \text{term} \rangle$
 $\Rightarrow A = B + \langle \text{term} \rangle * \langle \text{factor} \rangle$
 $\Rightarrow A = B + \langle \text{factor} \rangle * \langle \text{factor} \rangle$
 $\Rightarrow A = B + \langle \text{id} \rangle * \langle \text{factor} \rangle$
 $\Rightarrow A = B + C * \langle \text{factor} \rangle$
 $\Rightarrow A = B + C * \langle \text{id} \rangle$
 $\Rightarrow A = B + C * A$

Precedência entre operadores

A árvore sintática da sentença $A = B + C * A$

$\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\langle \text{id} \rangle \rightarrow A \mid B \mid C$
 $\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle$
 $\mid \langle \text{term} \rangle$
 $\langle \text{term} \rangle \rightarrow \langle \text{term} \rangle * \langle \text{factor} \rangle$
 $\mid \langle \text{factor} \rangle$
 $\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle)$
 $\mid \langle \text{id} \rangle$



Associatividade de operadores

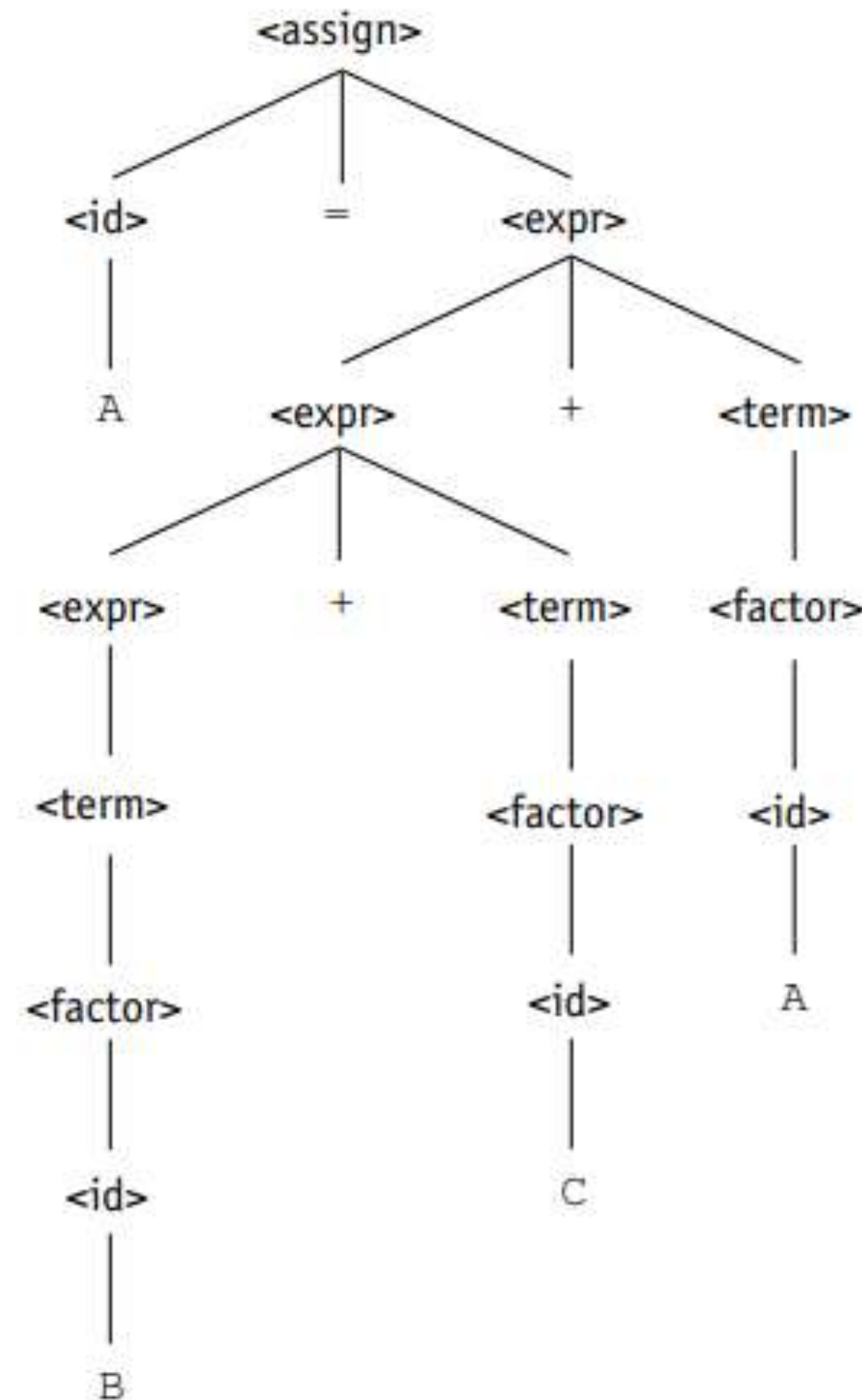
- Quando uma sentença inclui dois operadores de mesma precedência, uma regra semântica é necessária para especificar qual possui maior precedência – **associatividade**
- Por exemplo, seja a sentença $A = B + C + A$ e a gramática

```
<assign> → <id> = <expr>
<id> → A | B | C
<expr> → <expr> + <term>
        | <term>
<term> → <term> * <factor>
        | <factor>
<factor> → (<expr>)
          | <id>
```

Associatividade de operadores

- Sentença $A = B + C + A$

$\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\langle \text{id} \rangle \rightarrow A \mid B \mid C$
 $\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle$
 $\quad \quad \quad \mid \langle \text{term} \rangle$
 $\langle \text{term} \rangle \rightarrow \langle \text{term} \rangle * \langle \text{factor} \rangle$
 $\quad \quad \quad \mid \langle \text{factor} \rangle$
 $\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle)$
 $\quad \quad \quad \mid \langle \text{id} \rangle$



BNF Estendida (EBNF)

- Visam aumentar o poder descritivo da BNF
- Utiliza **metassímbolos**, como colchetes para melhorar a facilidade de escrita

Referências

- Sebesta, Robert W.; Conceitos de Linguagens de Programação; Capítulo 3.3; Bookman



IBMEC.BR

 /IBMEC

 IBMEC

 @IBMEC_OFICIAL

 @IBMEC

 **ibmec**