

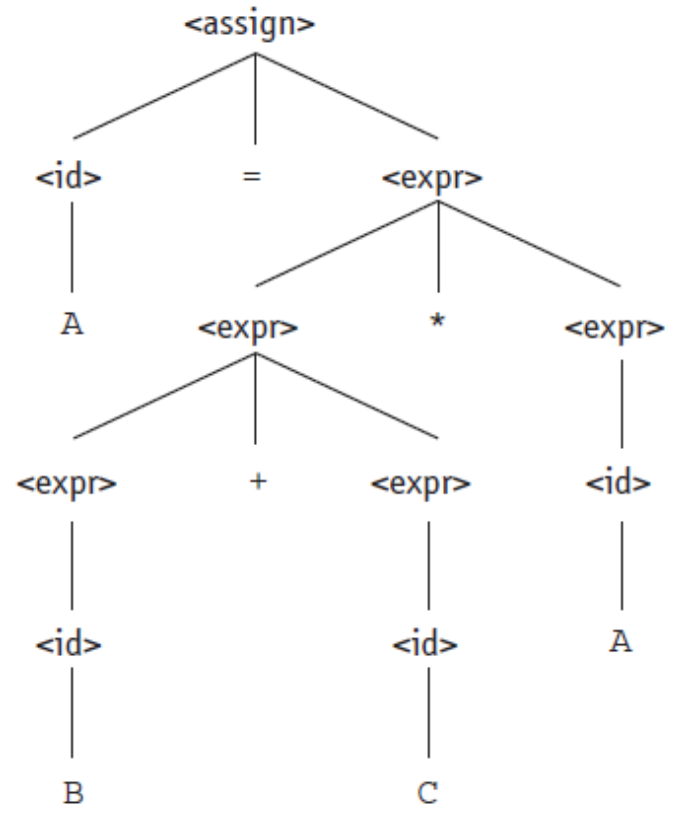
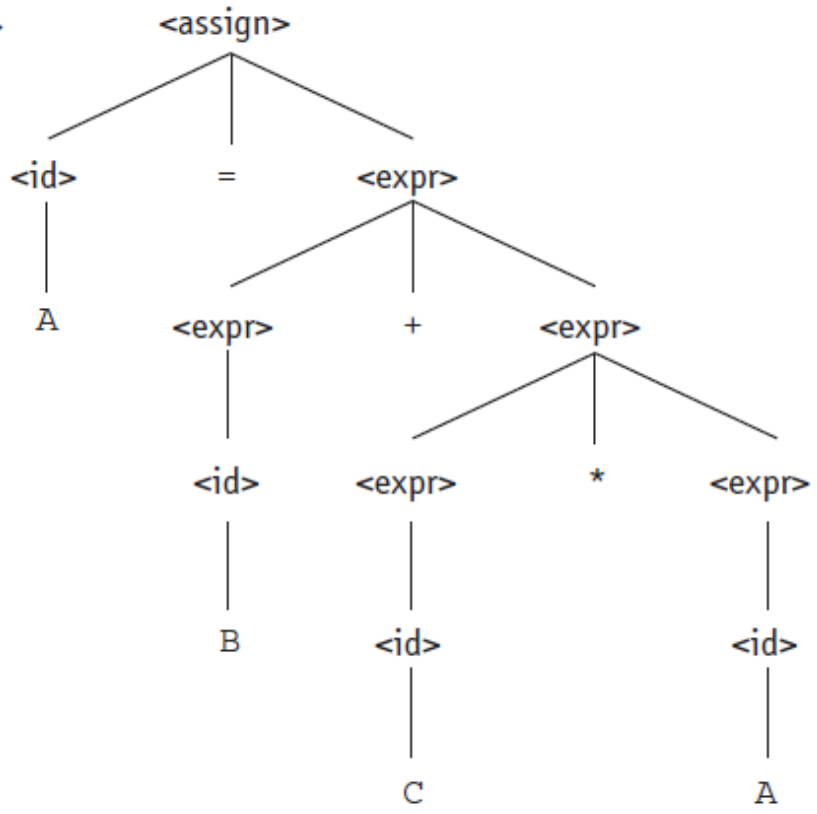
Sintaxis de Lenguajes de Programación

- Ambigüedad,
- Precedencia,
- Asociatividad

Ambigüedad en gramáticas

- Una gramática es ambigua si genera una forma sentencial con dos o más árboles de análisis sintáctico distintos.
 - Si la gramática genera una sentencia con más de una derivación a la izquierda.
 - Si la gramática genera una sentencia con más de una derivación a la derecha.
- Cada derivación con una gramática no ambigua tiene un único árbol de análisis sintáctico, a pesar de ella poder ser representada por derivaciones diferentes (izquierda o derecha).

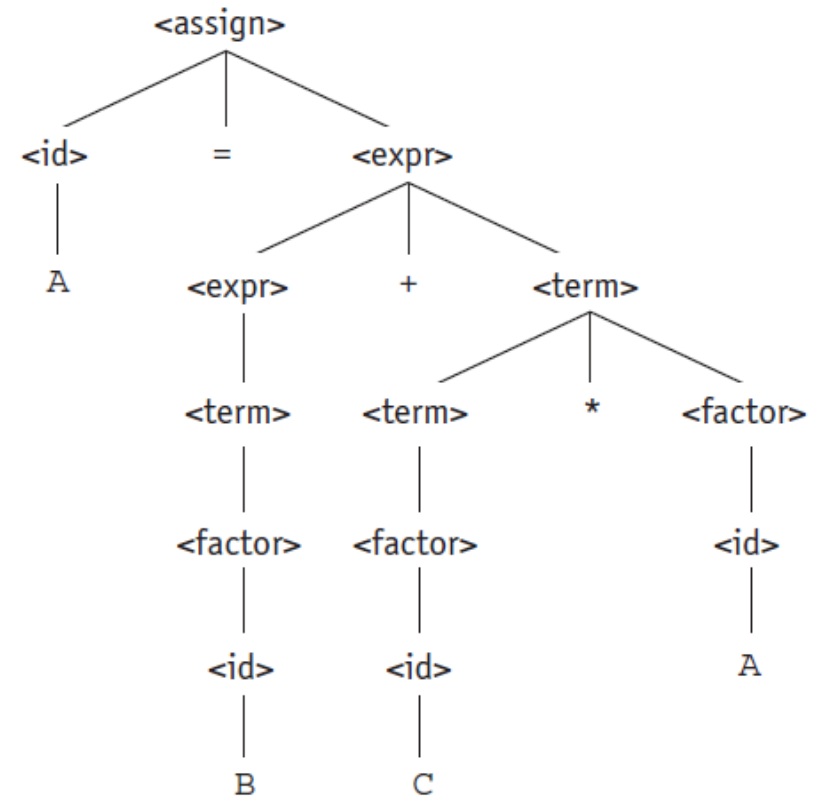
$\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\langle \text{id} \rangle \rightarrow A | B | C$
 $\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{expr} \rangle$
 | $\langle \text{expr} \rangle * \langle \text{expr}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



Una gramática de expresión no ambigua

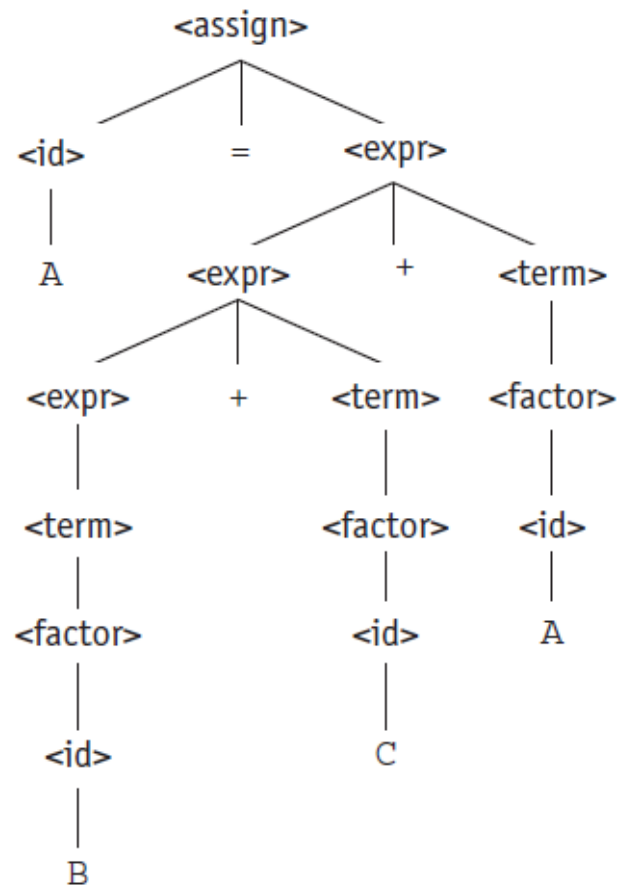
- Si usamos el árbol de análisis sintáctico para indicar niveles de precedencia, podemos eliminar la ambigüedad.

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



Asociatividad de operadores

- Asociatividad de un operador también puede ser indicada por una gramática



Una gramática no ambigua para if-then-else

`<if_stmt> → if <logic_expr> then <stmt>
 | if <logic_expr> then <stmt> else <stmt>`

Si `<stmt> → <if_stmt>`, esa gramática sería ambigua.

Considere la sentencia:

`if <logic_expr> then if <logic_expr> then <stmt> else <stmt>`

¿Cómo quedarían
los posibles árboles?

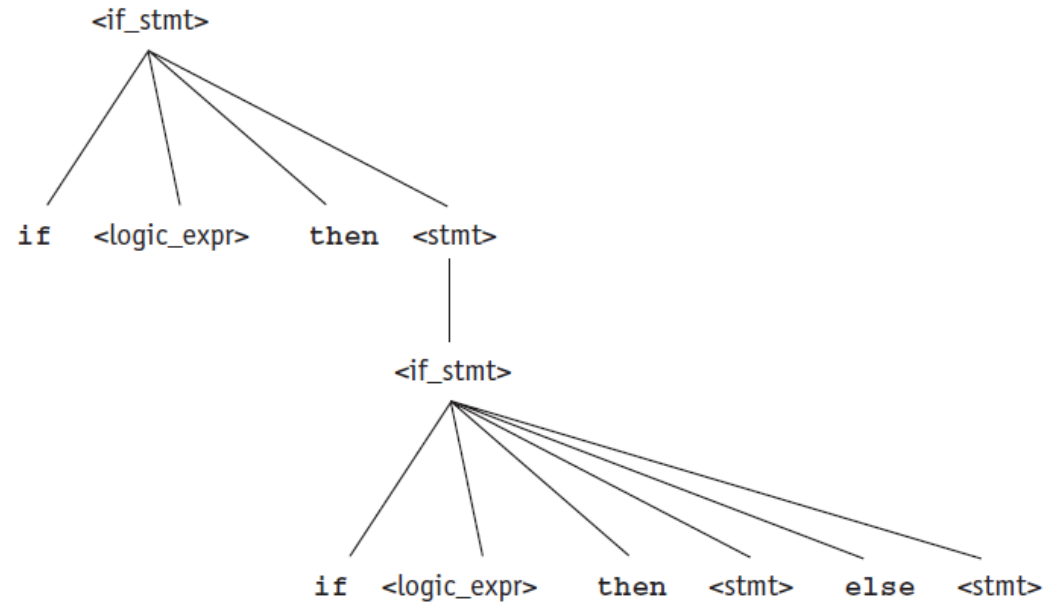
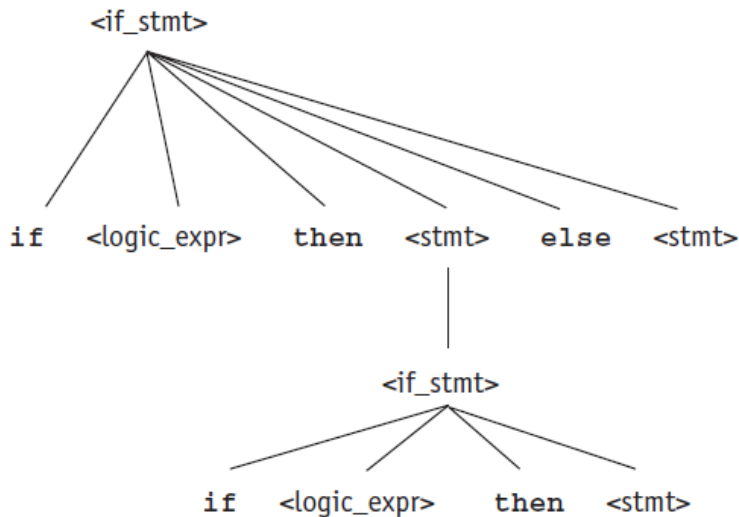
```
if done == true  
  then if denom == 0  
    then quotient = 0;  
    else quotient = num / denom;
```

Una gramática no ambigua para if-then-else

$\langle \text{if_stmt} \rangle \rightarrow \text{if } \langle \text{logic_expr} \rangle \text{ then } \langle \text{stmt} \rangle$
 $\quad \mid \text{if } \langle \text{logic_expr} \rangle \text{ then } \langle \text{stmt} \rangle \text{ else } \langle \text{stmt} \rangle$

Considere la sentencia:

if $\langle \text{logic_expr} \rangle$ **then** **if** $\langle \text{logic_expr} \rangle$ **then** $\langle \text{stmt} \rangle$ **else** $\langle \text{stmt} \rangle$



Una gramática no ambigua para if-then-else

Una clausula else, cuando presente, tiene correspondencia con el then más próximo todavía no emparejado.

$\langle \text{stmt} \rangle \rightarrow \langle \text{matched} \rangle \mid \langle \text{unmatched} \rangle$

$\langle \text{matched} \rangle \rightarrow \text{if } \langle \text{logic_expr} \rangle \text{ then } \langle \text{matched} \rangle \text{ else } \langle \text{matched} \rangle$
| cualquier otra sentencia no if

$\langle \text{unmatched} \rangle \rightarrow \text{if } \langle \text{logic_expr} \rangle \text{ then } \langle \text{stmt} \rangle$
| if $\langle \text{logic_expr} \rangle$ then $\langle \text{matched} \rangle$ else
 $\langle \text{unmatched} \rangle$

BNF extendida (EBNF)

- Partes opcionales son delimitadas por corchetes []
 $\text{<proc_call>} \rightarrow \text{ident} [\text{<expr_list>}]$
- Partes alternativas de RHSs son colocadas entre parentesis y separadas con barras verticales
 $\text{<term>} \rightarrow \text{<term>} (+|-) \text{const}$
- Repeticiones (0 o más) son colocadas entre llaves { }
 $\text{<ident>} \rightarrow \text{letter} \{\text{letter|digit}\}$

BNF y EBNF

BNF:

$$\begin{aligned}\langle \text{expr} \rangle &\rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle \\ &\quad | \langle \text{expr} \rangle - \langle \text{term} \rangle \\ &\quad | \langle \text{term} \rangle \\ \langle \text{term} \rangle &\rightarrow \langle \text{term} \rangle * \langle \text{factor} \rangle \\ &\quad | \langle \text{term} \rangle / \langle \text{factor} \rangle \\ &\quad | \langle \text{factor} \rangle \\ \langle \text{factor} \rangle &\rightarrow \langle \text{exp} \rangle ** \langle \text{factor} \rangle \\ &\quad | \langle \text{exp} \rangle \\ \langle \text{exp} \rangle &\rightarrow (\langle \text{expr} \rangle) \\ &\quad | \text{id}\end{aligned}$$

EBNF:

$$\begin{aligned}\langle \text{expr} \rangle &\rightarrow \langle \text{term} \rangle \{ (+ | -) \langle \text{term} \rangle \} \\ \langle \text{term} \rangle &\rightarrow \langle \text{factor} \rangle \{ (* | /) \langle \text{factor} \rangle \} \\ \langle \text{factor} \rangle &\rightarrow \langle \text{exp} \rangle \{ ** \langle \text{exp} \rangle \} \\ \langle \text{exp} \rangle &\rightarrow (\langle \text{expr} \rangle) \\ &\quad | \text{id}\end{aligned}$$

Recientes variaciones en EBNF

- RHSs alternativos son colocados en líneas separadas
- Uso de dos puntos en vez de \rightarrow
- Uso de *opt* para partes opcionales
- Uso de *oneof* para elecciones