

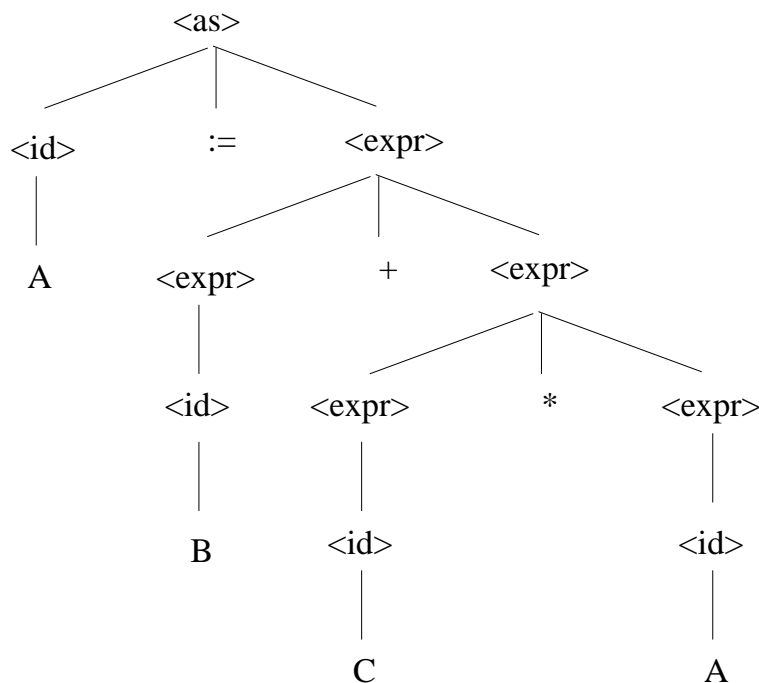
Ambiguity: Consider the following grammar for simple assignment statements:

$\langle \text{as} \rangle \rightarrow \langle \text{id} \rangle := \langle \text{expr} \rangle$

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

$\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{expr} \rangle$
 $\quad \mid \langle \text{expr} \rangle * \langle \text{expr} \rangle$
 $\quad \mid (\langle \text{expr} \rangle)$
 $\quad \mid \langle \text{id} \rangle$

We can parse $A := B + C * A$ in two ways:



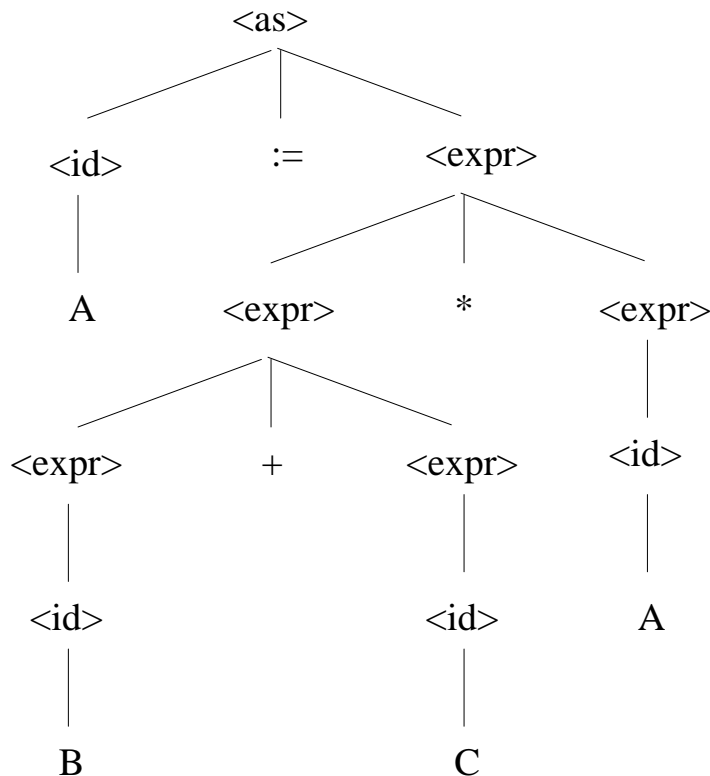
Suggests that $B + (C * A)$ is computed

$\langle \text{as} \rangle \rightarrow \langle \text{id} \rangle := \langle \text{expr} \rangle$

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

$\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{expr} \rangle$
 $\quad \quad \quad \mid \langle \text{expr} \rangle * \langle \text{expr} \rangle$
 $\quad \quad \quad \mid (\langle \text{expr} \rangle)$
 $\quad \quad \quad \mid \langle \text{id} \rangle$

But we can also build the parse tree:



Suggests that $(B + C) * A$ is computed

If any sentence in $L(G)$ has more than one parse tree, then we say that G is *ambiguous*.

Can we give a grammar for this language of assignments that is **not** ambiguous?

If so, we prefer to correlate the parse tree with the intended meaning; in this case, operator precedence.

Here is a new grammar that defines exactly the same language as does the previous grammar.

$\langle \text{as} \rangle \rightarrow \langle \text{id} \rangle := \langle \text{expr} \rangle$

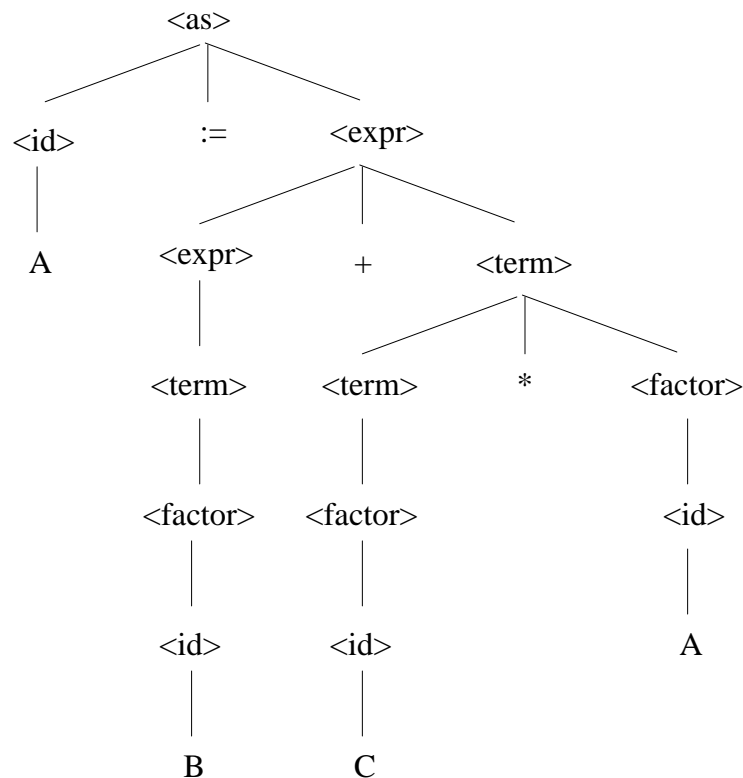
$\langle \text{id} \rangle \rightarrow A \mid B \mid C$

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

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

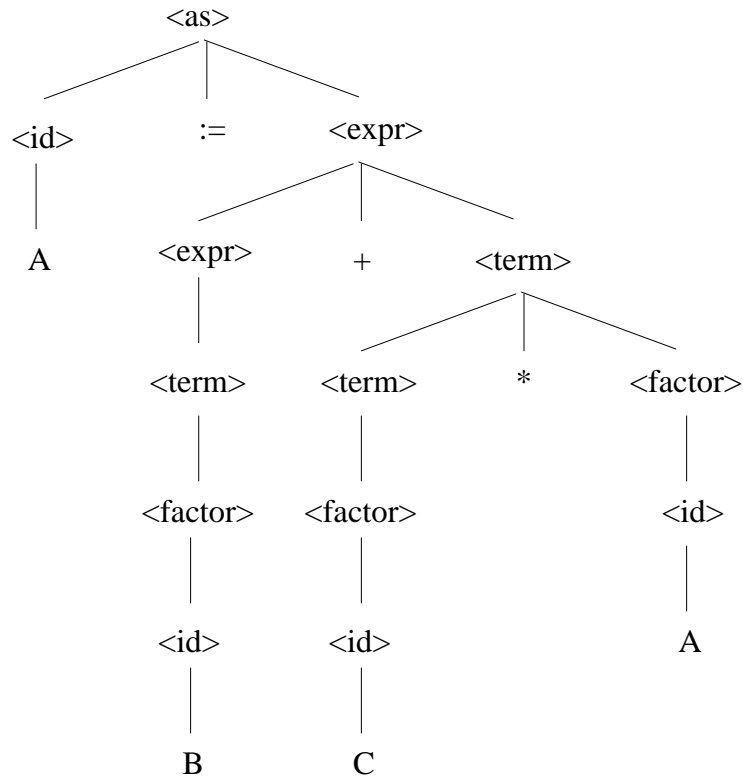
$\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle) \mid \langle \text{id} \rangle$

But there is only one parse tree for $A := B+C*A$



Note that $*$ has precedence over $+$

The unique parse tree



corresponds to more than one derivation:

- | | |
|---|---|
| $\langle as \rangle \Rightarrow \langle id \rangle := \langle expr \rangle$ | $\langle as \rangle \Rightarrow \langle id \rangle := \langle expr \rangle$ |
| $\Rightarrow A := \langle expr \rangle$ | $\Rightarrow A := \langle expr \rangle$ |
| $\Rightarrow A := \langle expr \rangle + \langle term \rangle$ | $\Rightarrow A := \langle expr \rangle + \langle term \rangle$ |
| $\Rightarrow A := \langle term \rangle + \langle term \rangle$ | $\Rightarrow A := \langle term \rangle + \langle term \rangle$ |
| $\Rightarrow A := \langle factor \rangle + \langle term \rangle$ | $\Rightarrow A := \langle factor \rangle + \langle term \rangle$ |
| $\Rightarrow A := \langle id \rangle + \langle term \rangle$ | $\Rightarrow A := \langle id \rangle + \langle term \rangle$ |
| $\Rightarrow A := B + \langle term \rangle$ | $\Rightarrow A := B + \langle term \rangle$ |
| $\Rightarrow A := B + \langle term \rangle * \langle factor \rangle$ | $\Rightarrow A := B + \langle term \rangle * \langle factor \rangle$ |
| $\Rightarrow A := B + \langle factor \rangle * \langle factor \rangle$ | $\Rightarrow A := B + \langle term \rangle * \langle id \rangle$ |
| $\Rightarrow A := B + \langle id \rangle * \langle factor \rangle$ | $\Rightarrow A := B + \langle term \rangle * A$ |
| $\Rightarrow A := B + C * \langle factor \rangle$ | $\Rightarrow A := B + \langle factor \rangle * A$ |
| $\Rightarrow A := B + C * \langle id \rangle$ | $\Rightarrow A := B + \langle id \rangle * A$ |
| $\Rightarrow A := B + C * A$ | $\Rightarrow A := B + C * A$ |

Canonical