

# Resolvendo Problemas de Ambiguidade

Vamos usar de exemplo a mesma gramática de expressões aritméticas

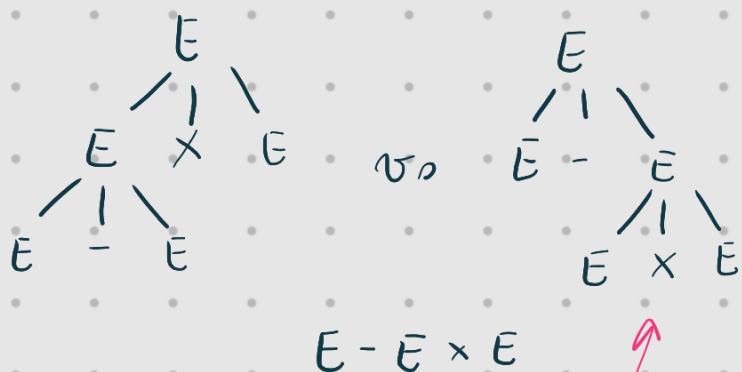
$$\begin{aligned} E &\rightarrow n \\ E &\rightarrow ( E ) \\ E &\rightarrow E - E \\ E &\rightarrow E \div E \end{aligned}$$

} ela possui ambiguidade  
no caso  $n - n \div n$

árvores possíveis:

Problema 1:

Precedência

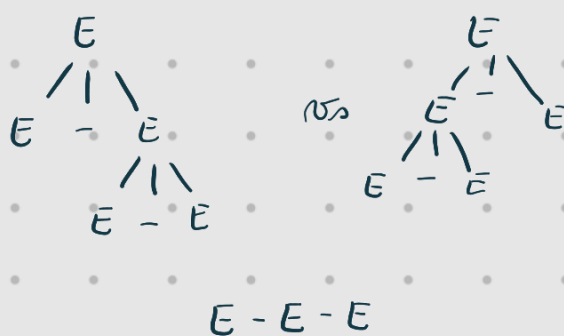


a certa regra essa  
pela matemática

Para resolver, vamos  
dizer que  $\times$  tem  
precedência mais forte  
que  $-$

Problema 2

Associatividade



Consideramos que o  
operador  $-$  é  
associativo à esquerda

Como resolver a ambiguidade?

$$E \rightarrow T - T$$

$$T \rightarrow F \times F$$

$$F \rightarrow n$$

$$F \rightarrow ( E )$$

Essa versão não é ambígua, mas não aceita todas  
as frases que deriva

- como:
- o  $n - n$
  - o  $n \times n$
  - o  $n$
  - o  $(n)$

Então vamos modificar para:

$$E \rightarrow T - T$$

$$E \rightarrow T$$

$$T \rightarrow F \times F$$

$$T \rightarrow F$$

$$F \rightarrow n$$

$$F \rightarrow (E)$$

Porém nesse também falta coisas:

- o  $n - n - n$
- o  $n \times n \times n$

Uma solução seria:

$$E \rightarrow E - T$$

$$E \rightarrow T$$

$$T \rightarrow T \times F$$

$$T \rightarrow F$$

$$F \rightarrow n$$

$$F \rightarrow (E)$$

sequência de  $T$ , separados por '-'  
associação esquerda

sequência de  $F$ , separados por ' $\times$ '  
associação esquerda

expressões atômicas.

o menos preciso  
que isso



$\hookrightarrow n - n - m$

para  $10 / 2 / 3$ , normalmente associamos a esquerda  
também  $(10 / 2) / 3$

para  $+$  e  $\times$  tanto faz a associação, mas normalmente  
é à esquerda

para derivação e associação normalmente é à direita

$$2^{3^4} \Rightarrow (2^3)^4$$

$2^{(3^4)} \rightarrow$  mais esolva

Qualidade deriva se não associativa mas muitas linguagens de programação fazem ser associativa à esquerda

$$X == Y == Z$$

faz

$$(booleano) == Z$$

$$\Downarrow$$
$$X == Y$$

No caso:

$$X \rightarrow X a$$

$$X \rightarrow b$$

$\Rightarrow$

$$X \rightarrow b R$$

$$R \rightarrow \epsilon$$

$$R \rightarrow a R$$

fun parse X()  
b = come("b")

l =



while prox == 'a':

a = come("a")

l =



Mas antes vamos resolver:

$$\left[ \begin{array}{l} X \rightarrow X - m \\ X \rightarrow m \end{array} \right.$$

$$X \rightarrow X - m$$

$$X \rightarrow m$$

$$X \rightarrow (X)$$