



Análise Sintáctica

Compiladores, Aula N° 10

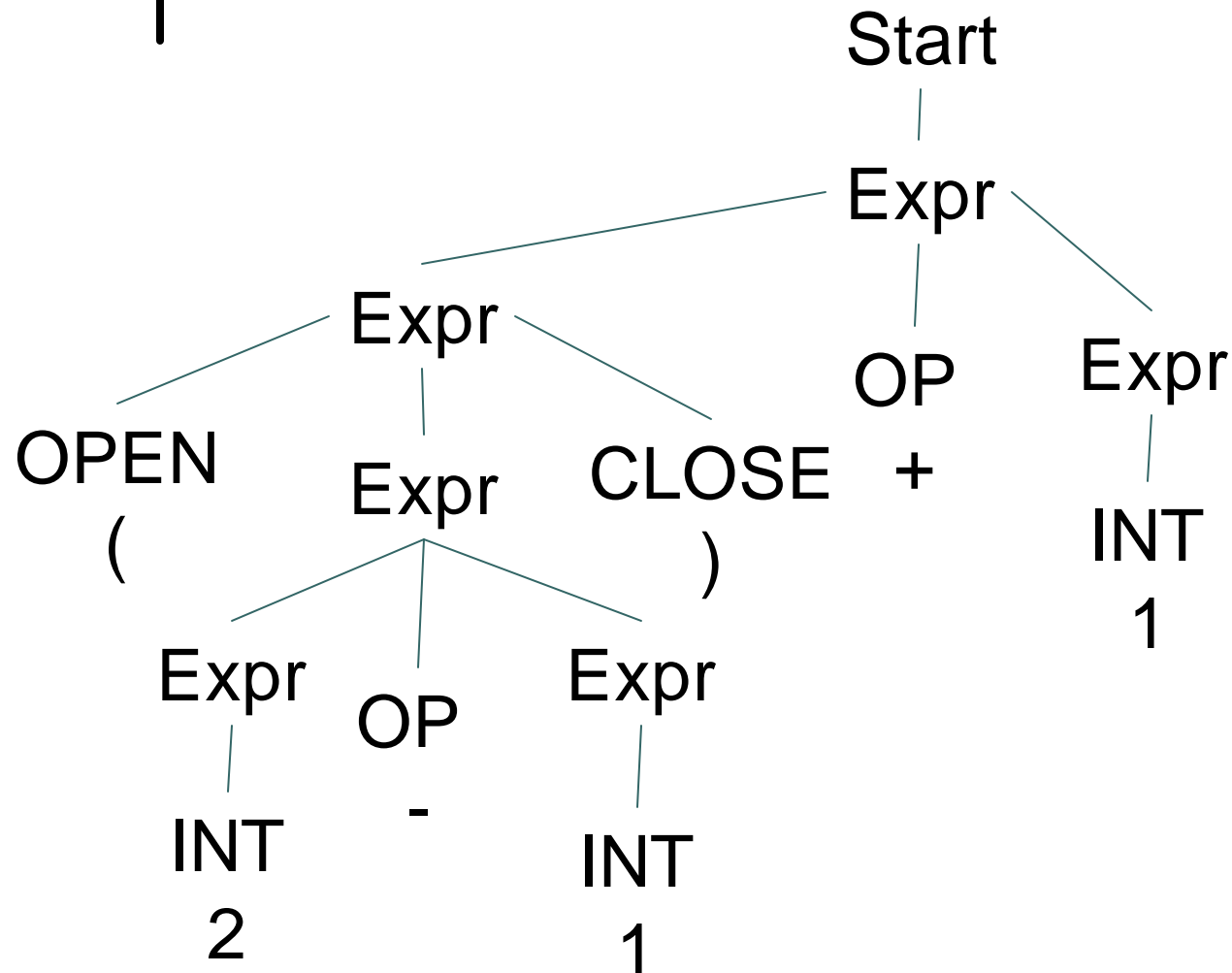
João M. P. Cardoso



Árvore Sintáctica

- ✍ Nódos internos: símbolos não-terminais
- ✍ Folhas: símbolos terminais
- ✍ Laços:
 - ✍ De símbolos não-terminais do LHS da produção
 - ✍ A nós do RHS da produção
- ✍ Captura a derivação da frase (String)

Árvore Sintáctica para (2-1)+1





Ambiguidade numa Gramática

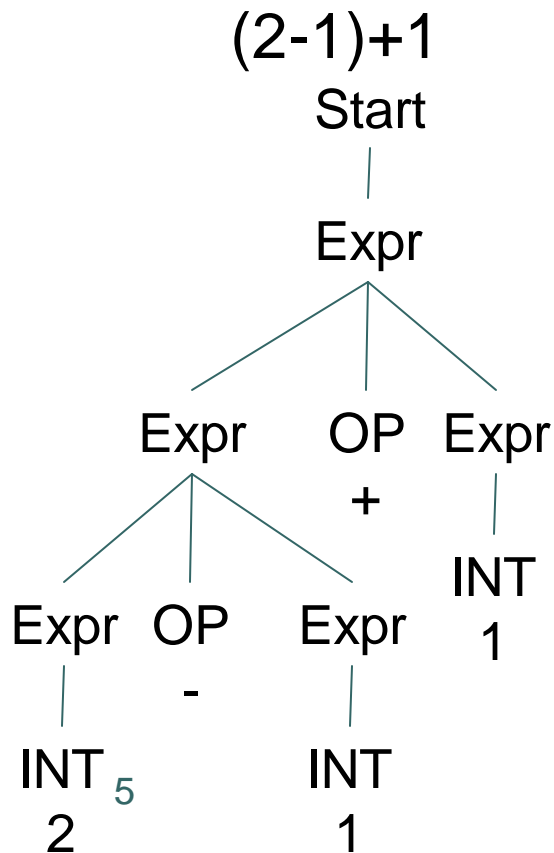
- ✍ Múltiplas derivações (como consequência: múltiplas árvores sintáticas) para a mesma String
- ✍ Derivação e árvore sintática reflecte usualmente a semântica do programa
- ✍ Ambiguidade na gramática reflecte muitas das vezes ambiguidades na semântica da linguagem (considerada indesejável)



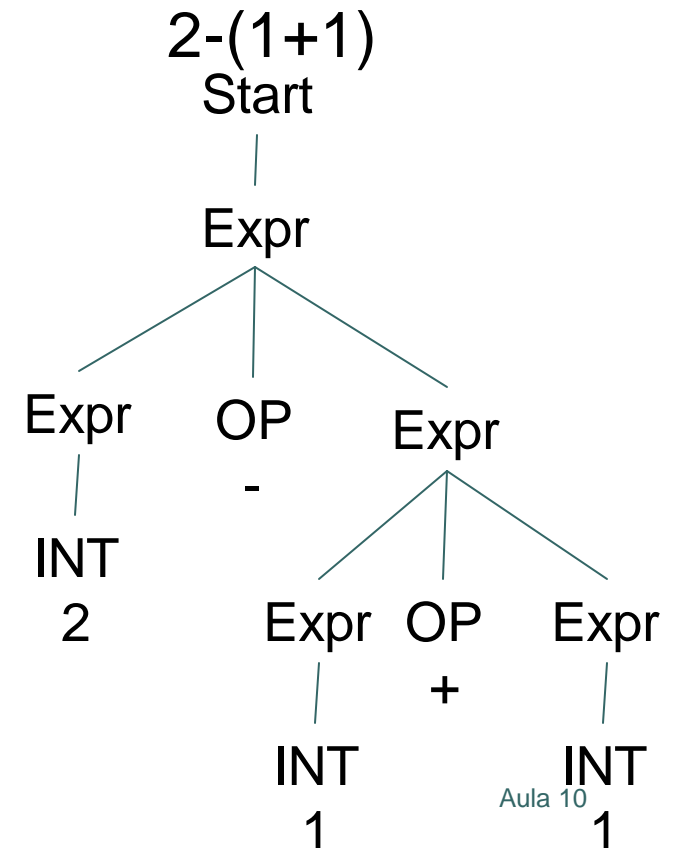
Exemplo de ambiguidade

✍ Duas árvores sintáticas para 2-1+1

Árvore correspondente a



Árvore correspondente a





Eliminação de ambiguidade

- ✍ Solução: modificar gramática
- ✍ Faz todos os operadores com associação à esquerda

Gramática Original

Start ? Expr

Expr ? Expr OP Expr

Expr ? INT

Expr ? OPEN Expr CLOSE

Gramática Modificada

Start ? Expr

Expr ? Expr OP INT

Expr ? INT

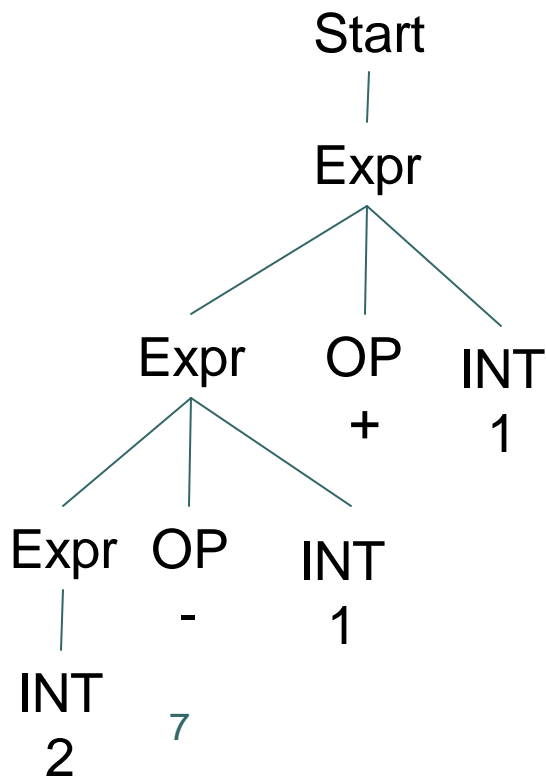
Expr ? OPEN Expr CLOSE



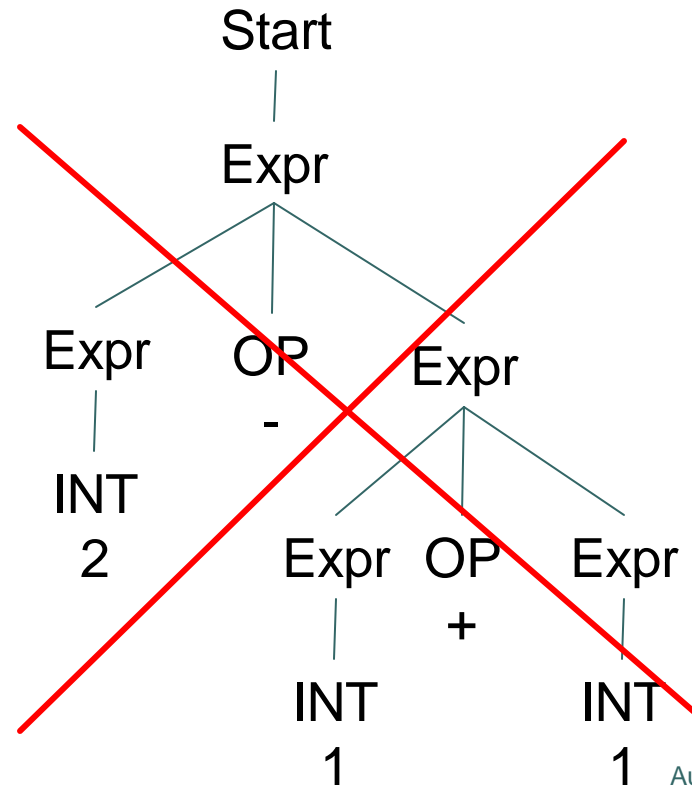
Árvore sintáctica para a gramática

✍ Apenas uma árvore sintáctica para: 2-1+1

Árvore sintáctica válida

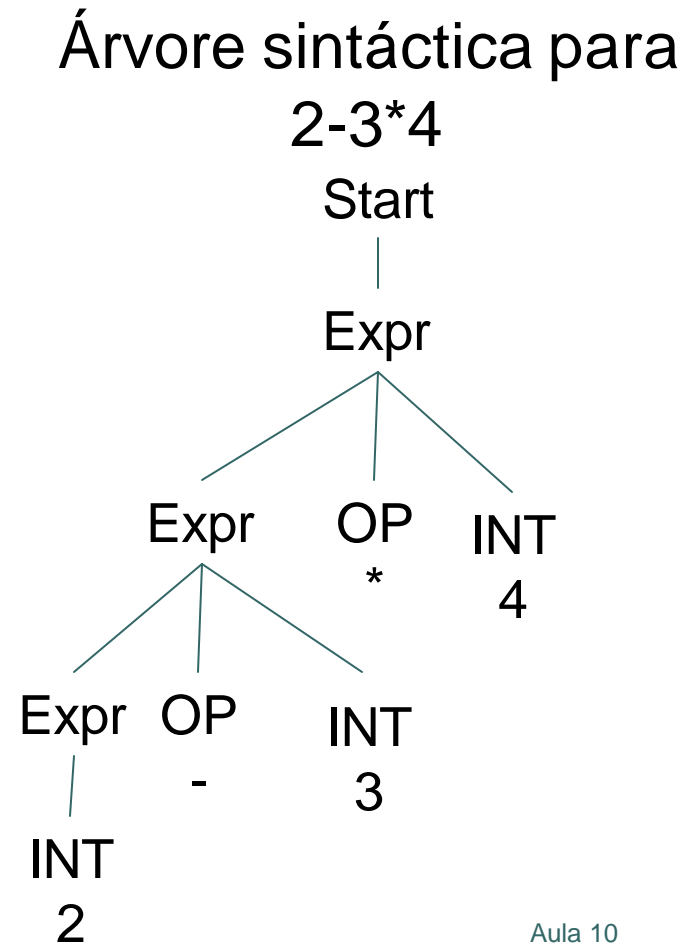


Árvore sintáctica inválida



Violação de prioridade

- ✍ Todos os operadores associam à esquerda
- ✍ Viola prioridade de * sobre +
- ✍ $2-3*4$ associa como $(2-3)*4$





Resolver prioridade

Gramática Original

OP = + | - | * | /

INT = [0-9] [0-9]*

OPEN = (

CLOSE =)

Start ? Expr

Expr ? Expr OP INT

Expr ? INT

Expr ? OPEN Expr CLOSE

Gramática Modificada

OP1 = + | -

OP2 = * | /

INT = [0-9] [0-9]*

OPEN = (

CLOSE =)

Start ? Expr

Expr ? Expr OP1 Term

Expr ? Term

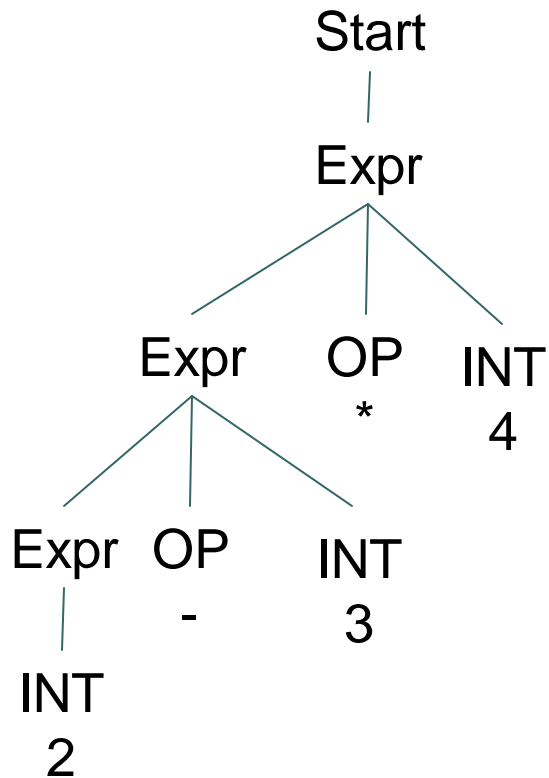
Expr ? OPEN Expr CLOSE

Term ? Term OP2 INT

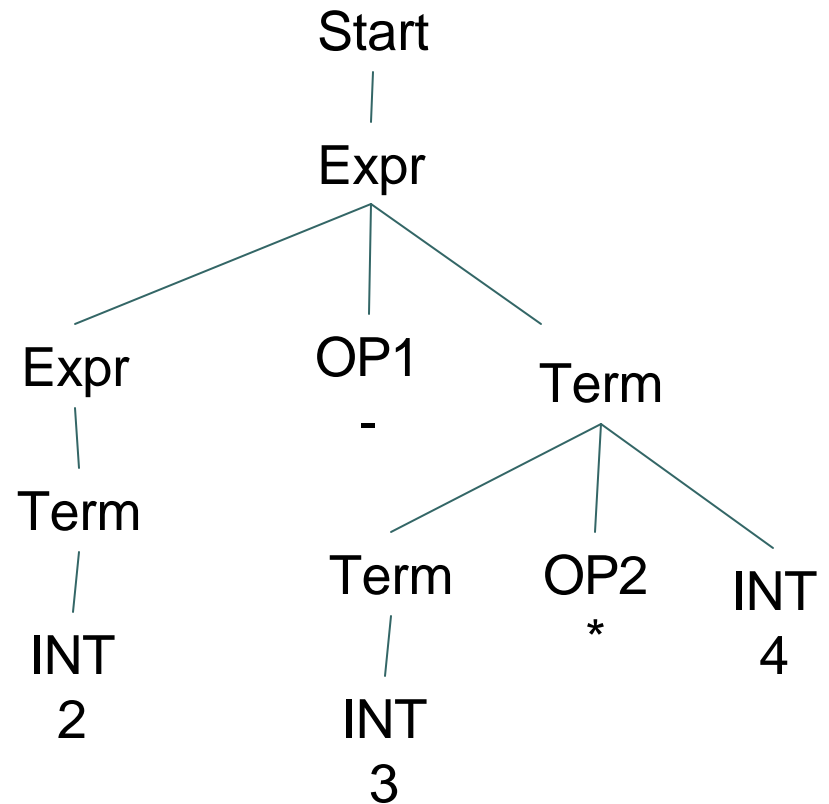
Term ? INT

Modificação nas Árvores Sintáticas

Velha Árvore sintáctica para
 $2-3*4$



Nova Árvore sintáctica para
 $2-3*4$





Ideia Geral

- ✍ Agrupar operadores por níveis de prioridade
 - ✍ * e / estão no nível de topo
 - ✍ + e - estão no nível a seguir
- ✍ Símbolo não-terminal para cada nível de prioridade
 - ✍ Term é não-terminal para * e /
 - ✍ Expr é não-terminal para + e -
- ✍ Pode-se fazer associatividade dos operadores à esquerda ou à direita em cada nível
- ✍ Generalizar para níveis arbitrários de prioridades



Exercícios (TPC)

- ✍ Especificar utilizando a representação BNF gramáticas correspondentes às expressões regulares: $[0-9]^+$ e $[0-9]^*$
- ✍ Dada a gramática:

NUM = $[0-9]^+$

ID = $[A-Za-Z][0-9A-Za-z]^*$

Expr ? Expr “+” Term | Expr “-” Term | Term

Term ? Term “*” Factor | Term “/” Factor | Factor

Factor ? Primary “^” Factor | Primary

Primary ? “-”Primary | Element

Element ? “(“ Expr “)” | NUM | ID

- ✍ Quais as árvores sintáticas para:

- $5-2*3$
- y^3