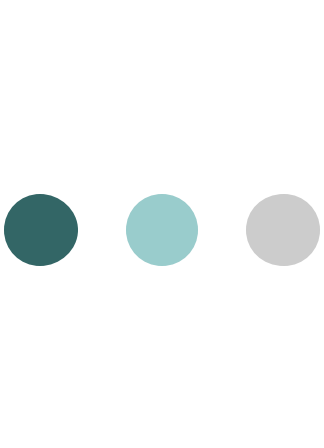


Análise Sintáctica

Compiladores, Aula Nº 11

João M. P. Cardoso

Manuseando estruturas if-then-else

Start ? Stat

Stat ? IF Expr THEN Stat ELSE Stat

Stat ? IF Expr THEN Stat

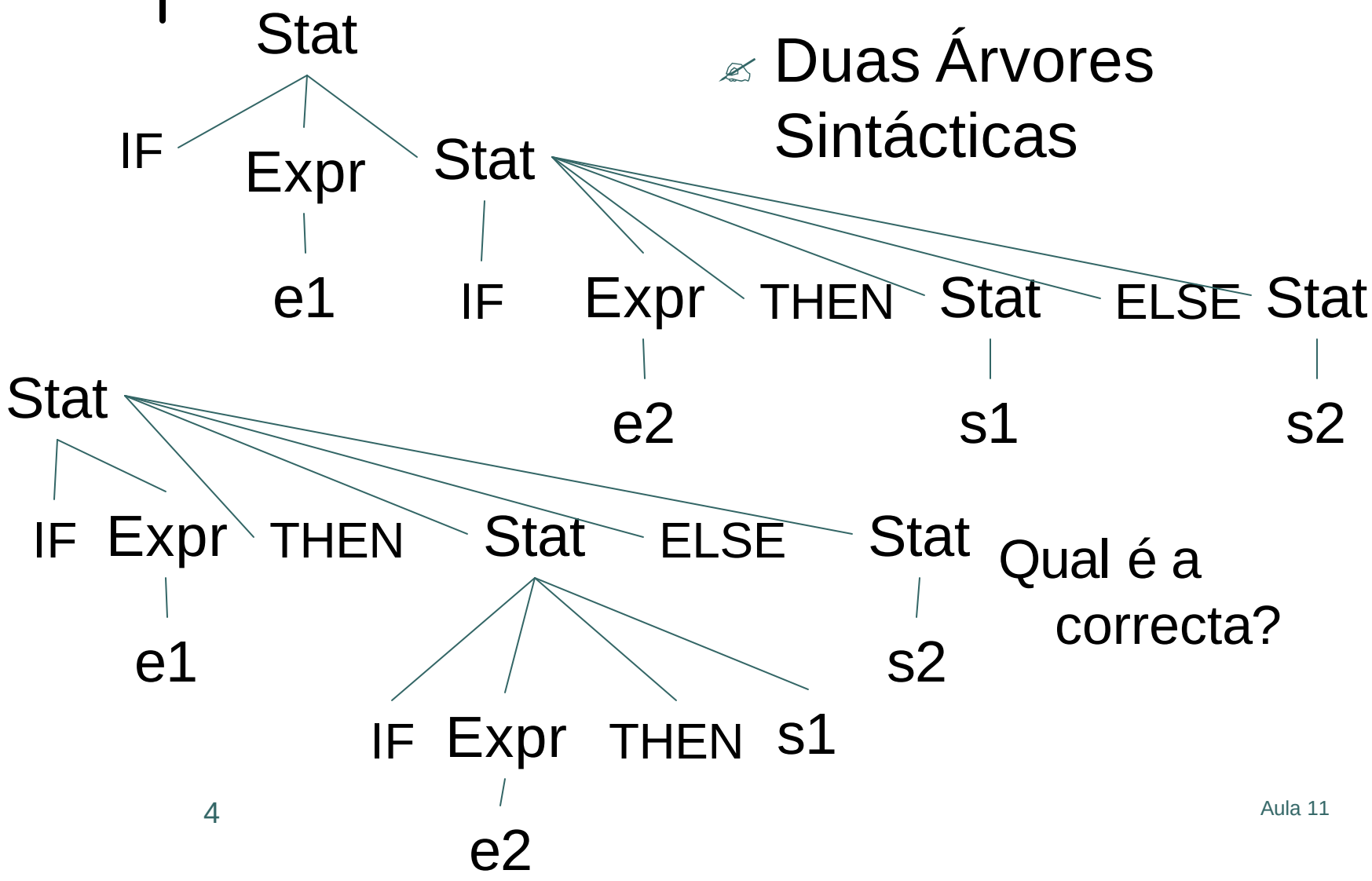
Stat ? ...

Árvore Sintáctica

- Considere o enunciado:
  - if e_1 then if e_2 then s_1 else s_2

Árvore Sintáctica

- Duas Árvores Sintácticas



Leituras alternativas

- Gramática ambígua
  - Árvore sintáctica 1

    if e_1

    if e_2 s_1

    else s_2
  - Árvore sintáctica 2

    if e_1

    if e_2 s_1

    else s_2

Gramática modificada

Goal ? Stat

Stat ? WithElse

Stat ? LastElse

WithElse ? IF Expr THEN WithElse ELSE WithElse

WithElse ? ...

LastElse ? IF Expr THEN Stat

LastElse ? IF Expr THEN WithElse ELSE LastElse

Gramática modificada

- Ideia básica: controlar quando um IF sem ELSE pode ocorrer
  - No nível de topo do enunciado
  - Ou como último numa sequência de enunciados if then else if then ...

Analisador Sintáctico

- Converte programas numa árvore sintáctica
- Pode ser programado do zero!
- Ou construído automaticamente por um gerador de "parsers"
  - Aceitam uma gramática como entrada
  - Produzem um analisador sintáctico como resultado
- Problema prático
  - A Árvore Sintáctica para a gramática modificada é complicada
  - Gostaríamos de começar com uma árvore sintáctica mais intuitiva

Solução

- Sintaxe Abstracta versus Concreta
  - Sintaxe abstracta corresponde ao meio intuitivo de pensar a estrutura do programa
    - Omite detalhes como palavras-chave supérfluas que estão lá para tornar a linguagem ambígua
    - Sintaxe abstracta pode ser ambígua
  - Sintaxe Concreta corresponde à gramática completa para utilizada para analisar sintacticamente a linguagem
- Os analisadores sintácticos são muitas das vezes escritos para produzirem Árvores Sintácticas Abstractas (ASTs)

ASTs (Árvores Sintácticas Abstractas)

- Começar com uma gramática intuitiva mas ambígua
- Modificar a gramática para a tornar não-ambígua
  - Árvores sintácticas concretas
  - Menos intuitivas
- Converter as árvores sintácticas concretas em ASTs
  - Correspondem à gramática intuitiva para a linguagem
  - Mais simples de manipular pelo programa

Exemplo

Gramática não-ambígua

OP1 = + | -

OP2 = * | /

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

OPEN = (

CLOSE =)

Start ? Expr

Expr ? Expr OP1 Term

Expr ? Term

Term ? OPEN Expr CLOSE

Term ? Term OP2 INT

Term ? INT

Gramática intuitiva mas ambígua

OP = * | / | + | -

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

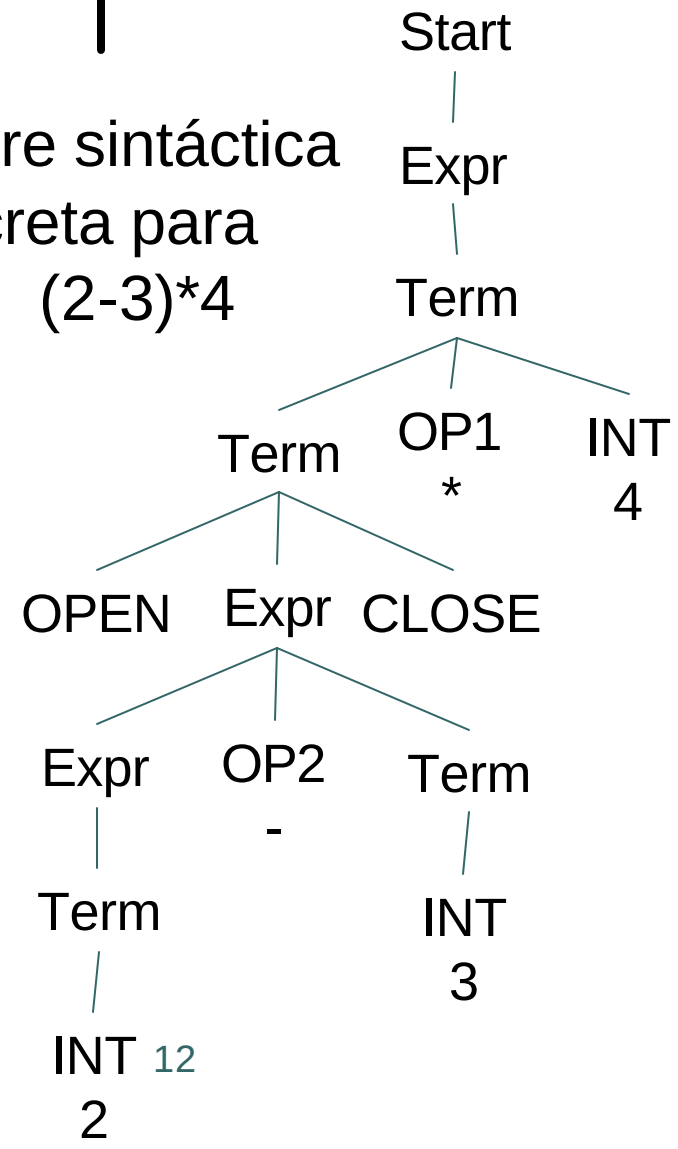
Start ? Expr

Expr ? Expr OP Expr

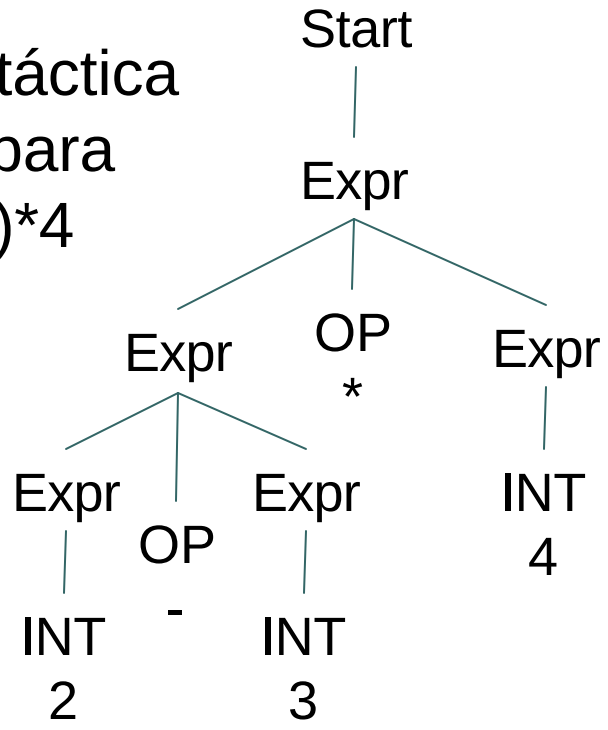
Expr ? INT

Exemplo

Árvore sintáctica concreta para (2-3)*4



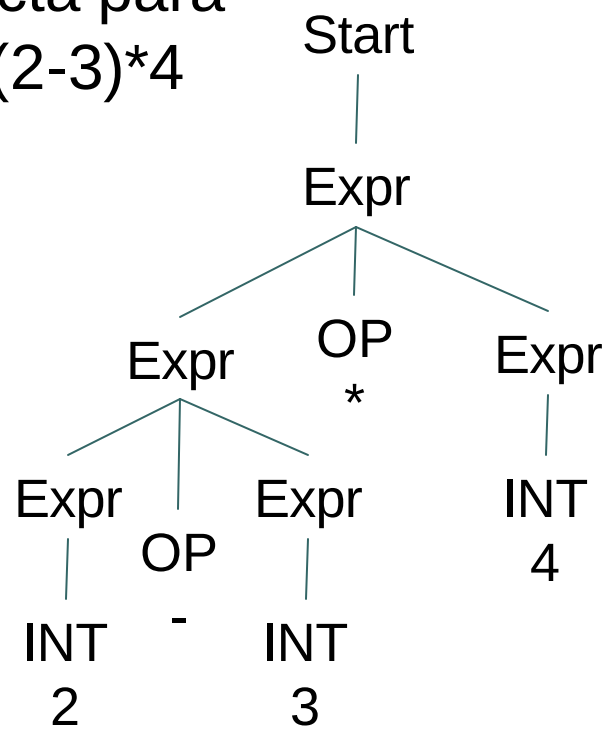
Árvore sintáctica abstracta para (2-3)*4



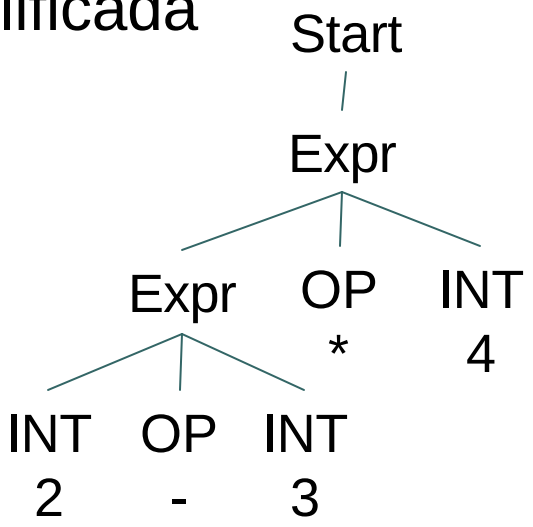
- Utiliza gramática intuitiva
- Elimina terminais supérfluos
  - OPEN, CLOSE, etc.

Exemplo

Árvore sintáctica abstracta para (2-3)*4



AST para (2-3)*4
Ainda mais simplificada



Sumário

- Níveis da estrutura lexicais e sintácticos
  - Lexicais – expressões regulares e autómatos
  - Sintácticos – gramáticas
- Ambiguidades na gramática
  - Gramáticas modificadas
  - Árvores Sintácticas Abstractas (ASTs)
- Papel generativo versus papel reconhecedor
  - Generativo mais conveniente para especificação
  - Reconhecedor requerido na implementação