

Compiladores — Folha laboratorial 3

DCC/FCUP

Outubro 2020

Parsing top-down

Exercício 1

Considere a gramática para expressões aritméticas fatorizada de forma a eliminar ambiguidade e recursão à esquerda:

$$\begin{array}{lll} E \rightarrow T E' & T \rightarrow F T' & \\ E' \rightarrow + T E' & T' \rightarrow * F T' & F \rightarrow \text{num} \\ E' \rightarrow - T E' & T' \rightarrow / F T' & F \rightarrow (E) \\ E' \rightarrow \varepsilon & T' \rightarrow \varepsilon & \end{array}$$

- (a) Determine o valor de `NULLABLE` e dos conjuntos `FIRST` e `FOLLOW` para os não-terminais desta gramática. (Sugestão: comece por determinar as repostas por inspeção das produções e só depois use as equações apresentadas na aulas para verificar se a sua intuição está correta.)
- (b) Mostre que a gramática acima é *LL(1)* calculado a tabela de *parsing* preditivo.

Exercício 2

Partindo do esqueleto de código fornecido, implemente um *parser* para esta gramática usando descida recursiva. O *parser* deve aceitar expressões bem formadas como `123*(45+6)` ou `(1+2*3)/(4+5)` e rejeitar expressões incorretas como `)12+53(` ou `(1++2*3)`.