

Relatório - Compiladores

Luan Bodner do Rosário

24 de Abril de 2016

Introdução

Gramática

$\langle \text{type} \rangle ::= \text{'vazio'}$
 $| \text{'inteiro'}$
 $| \text{'flutuante'}$

$\langle \text{variableDec} \rangle ::= \langle \text{type} \rangle \text{' : 'id'}$

$\langle \text{operationsExp} \rangle ::= \langle \text{equalityExp} \rangle$

$\langle \text{equalityExp} \rangle ::= \langle \text{relationalExp} \rangle \langle \text{equalityExpTransformed} \rangle$

$\langle \text{equalityExpTransformed} \rangle ::= \text{NULL}$
 $| \text{'='} \langle \text{relationalExp} \rangle \langle \text{equalityExpTransformed} \rangle$

$\langle \text{relationalExp} \rangle ::= \langle \text{additiveExp} \rangle \langle \text{relationalExpTransformed} \rangle$

$\langle \text{relationalExpTransformed} \rangle ::= \text{NULL}$
 $| < \langle \text{additiveExp} \rangle \langle \text{relationalExpTransformed} \rangle$
 $| > \langle \text{additiveExp} \rangle \langle \text{relationalExpTransformed} \rangle$
 $| <= \langle \text{additiveExp} \rangle \langle \text{relationalExpTransformed} \rangle$
 $| >= \langle \text{additiveExp} \rangle \langle \text{relationalExpTransformed} \rangle$

$\langle \text{additiveExp} \rangle ::= \langle \text{multiplicativeExp} \rangle \langle \text{additiveExpTransformed} \rangle$

$\langle \text{additiveExpTransformed} \rangle ::= \text{NULL}$
 $| + \langle \text{multiplicativeExp} \rangle \langle \text{additiveExpTransformed} \rangle$
 $| - \langle \text{multiplicativeExp} \rangle \langle \text{additiveExpTransformed} \rangle$

$\langle \text{multiplicativeExp} \rangle ::= \langle \text{factor} \rangle \langle \text{multiplicationExpTransformed} \rangle$

$\langle \text{multiplicationExpTransformed} \rangle ::= \text{NULL}$
 $| * \langle \text{factor} \rangle \langle \text{multiplicationExpTransformed} \rangle$
 $| / \langle \text{factor} \rangle \langle \text{multiplicationExpTransformed} \rangle$

$\langle \text{factor} \rangle ::= \text{'('} \langle \text{operationsExp} \rangle \text{')'}$
 $| \text{'numberFloat'}$
 $| \text{'numberInt'}$
 $| \text{'id'}$

$\langle \text{prototypeDef} \rangle ::= \text{'('} \langle \text{paramFunction} \rangle \text{')'}$

$\langle \text{paramFunction} \rangle ::= \text{NULL}$
 $\quad | \langle \text{variableDec} \rangle \text{ ',' } \langle \text{paramFunction} \rangle$
 $\quad | \langle \text{variableDec} \rangle$

$\langle \text{functionDec} \rangle ::= \langle \text{type} \rangle \text{ 'id' } \langle \text{prototypeDef} \rangle \langle \text{compoundStmt} \rangle \text{ 'fim'}$

$\langle \text{prototypeCall} \rangle ::= \text{NULL}$
 $\quad | \langle \text{operationsExp} \rangle \text{ ',' } \langle \text{prototypeCall} \rangle$
 $\quad | \langle \text{operationsExp} \rangle$

$\langle \text{functionCall} \rangle ::= \text{'id' ' (' } \langle \text{paramCall} \rangle \text{ ') '}$

$\langle \text{iterationExp} \rangle ::= \text{'repita' } \langle \text{compoundStmt} \rangle \text{'até' } \langle \text{operationsExp} \rangle \text{'fim'}$

$\langle \text{selectionExp} \rangle ::= \text{'se' } \langle \text{operationsExp} \rangle \text{'então' } \langle \text{compoundStmt} \rangle \text{'fim'}$
 $\quad | \text{'se' } \langle \text{operationsExp} \rangle \text{'então' } \langle \text{compoundStmt} \rangle \text{'senão'}$
 $\langle \text{compoundStmt} \rangle \text{'fim'}$

$\langle \text{ioTypes} \rangle ::= \text{'id'}$
 $\quad | \text{'numberInt'}$
 $\quad | \text{'numberFloat'}$
 $\quad | \langle \text{functionCall} \rangle$

$\langle \text{attributionExp} \rangle ::= \text{'id' ':' } \langle \text{operationsExp} \rangle$

$\langle \text{returnCom} \rangle ::= \text{'retorna' ' (' } \langle \text{operationsExp} \rangle \text{ ') '}$

$\langle \text{readCom} \rangle ::= \text{'leia' ' (' 'id' ') '}$

$\langle \text{writeCom} \rangle ::= \text{'escreve' ' (' } \langle \text{operationsExp} \rangle \text{ ') '}$

$\langle \text{compoundStmt} \rangle ::= \langle \text{expression} \rangle \langle \text{compoundStmt} \rangle$
 $\quad | \langle \text{expression} \rangle$

$\langle \text{expression} \rangle ::= \langle \text{selectionExp} \rangle$
 $\quad | \langle \text{iterationExp} \rangle$
 $\quad | \langle \text{functionCall} \rangle$
 $\quad | \langle \text{readCom} \rangle$
 $\quad | \langle \text{writeCom} \rangle$
 $\quad | \langle \text{returnCom} \rangle$
 $\quad | \langle \text{variableDec} \rangle$
 $\quad | \langle \text{attributionExp} \rangle$

