

# UNIVERSIDADE FEDERAL DO PAMPA

## ENGENHARIA DE COMPUTAÇÃO

Prof. Ana Paula Lüdtke Ferreira

### Linguagens Formais

Trabalho Prático T1 – 2018/1

**Aluno:**

Seja a linguagem de programação  $L$ , gerada pela seguinte gramática livre de contexto:

$\langle \text{programa} \rangle ::= \langle \text{principal} \rangle \langle \text{lista-de-funções} \rangle$   
 $\langle \text{principal} \rangle ::= \text{main}[\langle \text{params} \rangle](\langle \text{corpo} \rangle)$   
 $\langle \text{lista-de-funções} \rangle ::= \langle \text{função} \rangle$   
 $\langle \text{lista-de-funções} \rangle ::= \langle \text{função} \rangle \langle \text{lista-de-funções} \rangle$   
 $\langle \text{função} \rangle ::= \langle \text{id} \rangle [(\langle \text{corpo} \rangle)]$   
 $\langle \text{função} \rangle ::= \langle \text{id} \rangle [\langle \text{params} \rangle](\langle \text{corpo} \rangle)$   
 $\langle \text{params} \rangle ::= \langle \text{id} \rangle, \langle \text{params} \rangle$   
 $\langle \text{params} \rangle ::= \langle \text{id} \rangle$   
 $\langle \text{corpo} \rangle ::= \langle \text{id} \rangle []$   
 $\langle \text{corpo} \rangle ::= \langle \text{id} \rangle [\langle \text{args} \rangle]$   
 $\langle \text{corpo} \rangle ::= \text{if}[\langle \text{cond} \rangle, \langle \text{corpo} \rangle, \langle \text{corpo} \rangle]$   
 $\langle \text{args} \rangle ::= \langle \text{arg} \rangle$   
 $\langle \text{args} \rangle ::= \langle \text{arg} \rangle, \langle \text{args} \rangle$   
 $\langle \text{arg} \rangle ::= \langle \text{exp} \rangle$   
 $\langle \text{arg} \rangle ::= \langle \text{corpo} \rangle$   
 $\langle \text{exp} \rangle ::= \langle \text{num} \rangle$   
 $\langle \text{exp} \rangle ::= \langle \text{id} \rangle$   
 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{exp} \rangle +$   
 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{exp} \rangle -$   
 $\langle \text{exp} \rangle ::= \langle \text{corpo} \rangle$   
 $\langle \text{cond} \rangle ::= \langle \text{exp} \rangle \langle \text{exp} \rangle >$   
 $\langle \text{cond} \rangle ::= \langle \text{exp} \rangle \langle \text{exp} \rangle <$   
 $\langle \text{cond} \rangle ::= \langle \text{exp} \rangle \langle \text{exp} \rangle =$   
 $\langle \text{cond} \rangle ::= \langle \text{exp} \rangle \langle \text{exp} \rangle <>$   
 $\langle \text{id} \rangle ::= \langle \text{letra} \rangle \langle \text{seqsimb} \rangle$   
 $\langle \text{num} \rangle ::= \langle \text{digito} \rangle$   
 $\langle \text{num} \rangle ::= \langle \text{digito} \rangle \langle \text{num} \rangle$   
 $\langle \text{letra} \rangle ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z$   
 $\langle \text{digito} \rangle ::= 0|1|2|3|4|5|6|7|8|9$   
 $\langle \text{seqsimb} \rangle ::= \varepsilon$   
 $\langle \text{seqsimb} \rangle ::= \langle \text{letra} \rangle \langle \text{seqsimb} \rangle$   
 $\langle \text{seqsimb} \rangle ::= \langle \text{digito} \rangle \langle \text{seqsimb} \rangle$

1. “Compile” os seguintes trechos de código, dizendo se são programas válidos na linguagem  $L$  e quais geram erros. No caso de gerar erros, diga o que, no programa, está em desacordo com a gramática.

- (a) `main[x] (if[x>0, x, -1 * x])`
- (b) `main[x] (fatorial[x])`  
`fatorial[x] (if[x>1, fatorial[x 1 -], 1])`
- (c) `main[n] (fib(n))`  
`fib[k] (if[k 1 =, 0,`  
`if[k 2 =, 1,`  
`if[k 2 >, fib(n 1 -) fib(n 2 -) +]])`
- (d) `main[a,b,c] (fatorial[a b + c *])`  
`fatorial[x] (if[x 1 >, fatorial[x 1 -], 1])`

2. Escreva um programa que reconheça programas escritos na linguagem  $L$ . Os passos para executar essa operação são os seguintes:

- (a) Escreva um programa (em C, C++, Java ou Python), que leia um arquivo de texto contendo um código de programa na linguagem  $L$ .
- (b) Escreva um analisador léxico (autômato finito determinístico) que separe os itens do programa em identificadores, números, operadores e demais símbolos delimitadores da linguagem, conforme a gramática dada.
- (c) A partir da gramática dada e dos elementos identificados no passo anterior, escreva um analisador sintático determinístico que reconheça programas escritos na linguagem  $L$ .
- (d) No caso do programa apresentado como entrada possuir erros de sintaxe, o analisador sintático deverá ser capaz de reconstruir a derivação usada, mostrando qual foi o erro que ocorreu.

3. Apresente o trabalho realizado na forma de um relatório, explicitando o projeto do sistema, os algoritmos utilizados, a bibliografia consultada, o código fonte produzido, os resultados de análise para todos os programas apresentados no item 1 deste trabalho e qualquer outro aspecto considerado digno de nota.

A execução do trabalho é INDIVIDUAL.

Data da entrega: até 16/07/2018.

Data da apresentação (horário marcado com a professora): até 19/07/2018.

Observação: inconsistências ou omissões encontradas na gramática deverão ser resolvidas pelo/a discente e justificadas no relatório apresentado.