

# IPRJ – CONCEITOS DE LINGUAGENS DE PROGRAMAÇÃO

## LISTA DE EXERCÍCIOS 05

- 1) Quais são os critérios utilizados para avaliar uma linguagem de programação? Descreva brevemente cada um dos critérios.
- 2) Quais as principais características da linguagem Prolog?
- 3) Quais são as etapas do processo de compilação? Descreva as principais tarefas executadas em cada uma das etapas.
- 4) Qual a diferença entre a sintaxe e a semântica de uma linguagem de programação?
- 5) Considere a seguinte gramática em notação BNF:

$$\begin{aligned} \langle S \rangle &\rightarrow \langle A \rangle a \langle B \rangle b \\ \langle A \rangle &\rightarrow \langle A \rangle b \mid b \\ \langle B \rangle &\rightarrow a \langle B \rangle \mid a \end{aligned}$$

Indique quais das seguintes sentenças estão na linguagem gerada por essa gramática. Justifique a sua resposta apresentando a derivação da sentença.

- a) baab
  - b) bbbab
  - c) bbaaaaa
  - d) bbaab
  - e) aaabbbbaabbb
- 6) Escreva uma gramática em notação BNF para representar uma estrutura de repetição no seguinte formato:

```
while (x < y) && (y <= 10) do
  instrucoes
fim
```

Onde a expressão lógica pode ser formada por operadores relacionais (>, <, >=, <=, ==, !=) e operadores lógicos (&&, ||).

Prove que a gramática é válida apresentando a derivação do exemplo anterior.

7) Considere a seguinte gramática em notação BNF:

```
<tdecl> → <type> <varlist> ;  
<varlist> → <var> | <varlist> , <var>  
<var> → a | b | c | d | e | f | g  
<type> → static <basicctype> | <basicctype>  
<basicctype> → int | bool | string
```

Apresente uma derivação à extrema esquerda e construa a árvore de análise para cada uma das seguintes sentenças:

- a) int a, b;
- b) static string a, b, c;
- c) bool a, b, c, d, e;

8) Prove que a seguinte gramática é ambígua:

```
<S> → <A>  
<A> → <A> + <A> | <id>  
<id> → a | b | c
```

9) Considerando a existência de um analisador léxico implementado em C capaz de reconhecer os seguintes tokens:

INT\_LIT: número inteiro;  
IDENT: identificador de variável;  
ADD\_OP: operador de soma;  
SUB\_OP: operador de subtração;  
LEFT\_PAREN: parêntese esquerdo;  
RIGHT\_PAREN: parêntese direito;

Os quais são referentes a seguinte gramática:

```
<termo> → <fator> + <fator>  
          | <fator> - <fator>  
<fator> → identificador  
          | numero inteiro  
          | ( <termo> )
```

- a) De acordo com a descrição dos tokens e da gramática, apresente a implementação em C das funções que descrevem a gramática em um analisador sintático baseado no método de análise descendente recursiva.
- b) Simule a execução das funções do analisador léxico e apresente a árvore de análise gerada para o seguinte programa de entrada:  $(n1 + n2) - 25$