

# 1 Avaliação

## Linguagens de Programação 2020.2

Aluno: \_\_\_\_\_

1. As linguagens de programação (LP) podem ser avaliadas considerando diversos critérios de avaliação. Entre eles, podemos destacar a legibilidade, capacidade de escrita e a confiabilidade. Analise a seguinte lista de características das LP e relacione com os critérios de avaliação da linguagem que são afetados pela característica citada. Justifique sua resposta.

Característica	Legibilidade	Capacidade de Escrita	Confiabilidade
Na linguagem C, o símbolo '*' possui múltiplos significados	X	X	X
Na linguagem Fortran não possui registros para armazenar tipos diferentes de dados. Por isso, precisamos criar vários vetores, cada um para armazenar elementos de um determinado tipo.			
O operador ternário do C é uma forma abreviada de escrever condicionais if-else			
A linguagem C não possui o tipo booleano.			
A linguagem Pascal faz a verificação de acesso a posições válida em vetores em tempo de execução.			
A linguagem C possui dois tipos de dados estruturados: matrizes e registros. Os registros podem ser retorno de funções, mas matrizes não.			
A linguagem ADA utiliza marcadores <b>end if</b> para finalizar o bloco de seleção e <b>end loop</b> para finalizar a construção de laço.			

Explicação: A sobrecarga (overloading) de operadores reduz a simplicidade global da LP. Quando a simplicidade global da LP é afetada, a legibilidade, capacidade de escrita e confiabilidade são afetadas.

2. O trecho de código seguinte permite com uma única linha copiar um vetor de caracteres para um outro.

```
for (; *q = *p ; q++, p++) ;
```

Cite dois critérios de avaliação afetados no exemplo acima.

3. Uma linguagem de programação pode ser implementada usando o método de compilação ou interpretação. Qual é o método de implementação mais adequado para linguagens com a tipagem dinâmica. Justifique a sua resposta.

4. Considere a seguinte gramática:

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

- Mostre uma derivação mais à esquerda da sentença bbaab
- Mostre uma árvore de análise da sentença bbbaaab
- Descreva a linguagem gerada pela gramática acima usando uma expressão regular.
- Descreva a linguagem gerada pela gramática acima usando um autômato.

5. Considere a seguinte gramática:

$$\begin{aligned} \langle expr \rangle &\rightarrow \langle termo \rangle \langle expr' \rangle \\ \langle expr' \rangle &\rightarrow + \langle termo \rangle \langle expr' \rangle \mid \varepsilon \\ \langle termo \rangle &\rightarrow \langle fator \rangle \langle termo' \rangle \\ \langle termo' \rangle &\rightarrow * \langle fator \rangle \langle termo' \rangle \mid \varepsilon \\ \langle fator \rangle &\rightarrow (\langle expr \rangle) \mid \langle id \rangle \\ \langle id \rangle &\rightarrow A \mid B \mid C \end{aligned}$$

Mostre uma derivação mais à esquerda e uma árvore de análise para as sentenças considerando  $\langle expr \rangle$  como não terminal inicial e  $\varepsilon$  a string vazia:

- $(A+B)*C$
- $A*(B+C)$

6. Considere a seguinte gramática:

$$S \rightarrow iS \mid iSeS \mid \varepsilon$$

A gramática acima é ambígua. Mostre que o string  $ii\epsilon$  possui duas árvores de análise distintas.

Anulada Considere a seguinte gramática:

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

Mostre que a gramática acima é ambígua. Dica: A gramática acima descreve a linguagem  $L = \{a^*wc^k \mid w \in \{a,b\}^*, k = n_a(w)\}$ , onde  $n_a(w)$  representa a quantidade de a's em  $w$ .

7. Considere o seguinte descrição BNF:

```
prog      := func_list
func_list := func func_list | ε
func      := type id '(' param_list ')' '{' var_decl_list stmt_list '}'
stmt      := id '=' id ';'
stmt_list := stmt stmt_list | ε
var_decl  := type id ';'
var_decl_list := var_decl var_decl_list | ε
type      := int
param_list := param | param ',' param_list
param     := void | type id
```

Considere o seguinte trecho de código:

```
int main(void){
    int a;
    b = a;
}
```

(a) Faça a análise léxica do código acima indicando os tokens associados a cada lexema:

lexema	token
int	int
main	id
(	(
void	
)	
{	
int	
a	
;	
b	
=	
}	

(b) Mostre que o trecho de código acima pode ser gerado pela descrição BNF.

(c) Indique um erro semântico presente no código.

(d) O seguinte trecho de código está sintaticamente correto?

```
int main(void){
    int a, b, c;
    int v[5];
    b = a;
}
```

(e) Se o trecho de código do item anterior está sintaticamente incorreto, estenda a descrição BNF para que o código acima fique sintaticamente correto.

8. Considere a seguinte gramática de atributos:

Regra Sintática	Regra Semântica
$\langle E \rangle \rightarrow \langle E \rangle [1] + \langle T \rangle$	$\langle E \rangle .val \leftarrow \langle E \rangle [1].val + \langle T \rangle .val$
$\langle E \rangle \rightarrow \langle T \rangle$	$\langle E \rangle .val \leftarrow \langle T \rangle .val$
$\langle T \rangle \rightarrow \langle T \rangle [1] * \langle F \rangle$	$\langle T \rangle .val \leftarrow \langle T \rangle [1].val * \langle F \rangle .val$
$\langle T \rangle \rightarrow \langle F \rangle$	$\langle T \rangle .val \leftarrow \langle F \rangle .val$
$\langle F \rangle \rightarrow (\langle E \rangle)$	$\langle F \rangle .val \leftarrow \langle E \rangle .val$
$\langle F \rangle \rightarrow \langle num \rangle$	$\langle F \rangle .val \leftarrow \langle num \rangle .val$

Construa a árvore sintática anotada (completamente atribuída) para a expressão  $(8 + 3) * (2 + 5)$ .

9. Mostre a análise descendente dirigida pela tabela completa, incluindo o conteúdo da pilha de análise e a cadeia de entrada para a entrada  $(A+B)*C$  usando a tabela de análise descendente seguinte:

	a	+	*	(	)	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow a$			$F \rightarrow (E)$		

10. Mostre a análise ascendente dirigida por tabela completa, incluindo o conteúdo da pilha de análise, a cadeia de entrada e a ação para a entrada  $[[a];[a]]$  usando a gramática e a tabela de análise seguinte:

Gramática:

1.  $S \rightarrow [ L ]$
2.  $S \rightarrow a$
3.  $L \rightarrow L ; S$
4.  $L \rightarrow S$

Tabela de Análise Ascendente:

	ACTION					GO TO	
	;	[	]	a	\$	L	S
0		s2		s3			1
1					acc		
2		s2		s3		4	5
3	r2		r2		r2		
4	s6		s7				
5	r4		r4				
6		s2		s3			8
7	r1		r1		r1		
8	r3		r3				