

## CONSTRUÇÃO DE COMPILADORES I [BCC328]

### Primeira prova (2019–1)

22 de maio de 2019

Matrícula: \_\_\_\_\_

Nome: \_\_\_\_\_

Departamento de Computação  
Universidade Federal de Ouro Preto  
Prof. José Romildo Malaquias

**Questão 1.** Considere a seguinte especificação léxica:

loop	{ loop	}
end	{ end	}
:=	{ atrib	}
[+-]? ([0-9]+ "." [0-9]*   [0-9]* "." [0-9]+)	{ real	}
[+-]? [0-9]+	{ decimal	}
[+-]? 0 b [0-7]+	{ binário	}
[+-]? 0 x [0-9a-fA-F]+	{ hexadecimal	}
[a-zA-Z_][a-zA-Z0-9_]*	{ id	}
[ \t\n\r]+	{	}
# .*	{	}
.	{ erro	}

Observe que os caracteres brancos e os comentários são descartados.

Faça a análise léxica do programa a seguir, indicando a sequência de tokens encontrados.

Em sua resposta anote os tokens encontrados na linha da tabela correspondente à linha do programa em que ele foi encontrado.

1	program Main:
2	loop contador > 0
3	peso := peso + 0b1       # atualiza o peso do indivíduo
4	print -8439 456.17
5	end
6	loopwhile true do
7	print (-0x567beleza)
8	end

linha	tokens
1	
2	
3	
4	
5	
6	
7	
8	

**Questão 2.** Diferentes linguagens de programação usam diferentes notações para representar literais numéricos. Construa uma expressão regular para especificar os literais inteiros não negativos em VHDL, formados por uma sequência não vazia de dígitos decimais, e que podem incluir sublinhados (  ), úteis para agrupar dígitos facilitando a sua leitura e escrita. Porém um sublinhado não pode ocorrer como o primeiro ou último caracter do literal. Exemplos:

```
28
85_379_001
4_120_793_001
12_34_56
123_45678_9
```

**Questão 3.** Considere a seguinte gramática livre de contexto:

$$\begin{aligned} S &\rightarrow S S \\ S &\rightarrow \mathbf{a} \\ S &\rightarrow \mathbf{b} \end{aligned}$$

1. Dê evidências de que a gramática é ambígua.
2. Escreva uma gramática livre de contexto não ambígua equivalente à gramática dada.

**Questão 4.** Considere a seguinte gramática livre de contexto:

$$\begin{aligned}
 S &\rightarrow A \ \$ \\
 A &\rightarrow u \ B \ D \ z \\
 B &\rightarrow B \ v \\
 B &\rightarrow w \\
 D &\rightarrow E \ F \\
 E &\rightarrow y \\
 E &\rightarrow \\
 F &\rightarrow x \\
 F &\rightarrow
 \end{aligned}$$

- Determine quais símbolos não terminais são anuláveis.
- Determine o conjunto *FIRST* de cada símbolo não terminal.
- Determine o conjunto *FOLLOW* de cada símbolo não terminal.

NT	NULLABLE	FIRST	FOLLOW
S			
A			
B			
D			
E			
F			

**Questão 5.** Considere a seguinte gramática livre de contexto:

- $A \rightarrow S \ \$$
- $S \rightarrow X \ Y \ Z$
- $X \rightarrow a \ X \ b$
- $X \rightarrow$
- $Y \rightarrow c \ Y \ Z \ c \ X$
- $Y \rightarrow d$
- $Z \rightarrow e \ Z \ Y \ e$
- $Z \rightarrow f$

Os símbolos não terminais anuláveis, bem como os conjuntos *FIRST* e *FOLLOW* estão indicados na tabela a seguir.

NT	NULLABLE	FIRST	FOLLOW
A	F	a c d	
S	F	a c d	\$
X	V	a	b c d e f
Y	F	c d	e f
Z	F	e f	c d \$

Construa a tabela de análise sintática LL(1) para a gramática. Esta gramática é LL(1)? Justifique sua resposta.

**Observação:** utilize os números das regras de produção para preencher a tabela.

