



# **Processamento de Linguagens**

## **Trabalho Prático 01 - Exercício 1**

### **Linguagens Regulares – Análise Léxica**

Trabalho elaborado por:

Marcos Alberto Lopes de Vasconcelos, nºa18568

Diogo Miguel Barbosa de Oliveira, nº20468

Tomás Fernandes Ferreira, nºa20457

## Exercício 1

a)

$(\backslash+|-)?\backslash d+\backslash.\backslash d+([eE](\backslash+|-)?\backslash d+)?$

- $(\backslash+|-)?$  - utilização opcional do caractere '+' ou '-' no início da expressão regular. O caracter '|' é utilizado para separar as duas opções, ou uma ou outra. O caracter '?' define que o grupo é opcional e que pode ter ou não um sinal.
- $\backslash d+$  - uma ou mais ocorrências de um dígito numérico. O ' $\backslash d$ ' é a abreviação de qualquer dígito numérico de 0 a 9 e o '+' significa que pode haver uma ou mais ocorrências do dígito.
- $\backslash.$  - utilização de um ponto '.'. Com o caracter '\ ' é para ser interpretado literalmente como um ponto.
- $[eE]$  - utilização do caractere 'e' ou 'E' usado para notação científica.

b) e c)

AFND

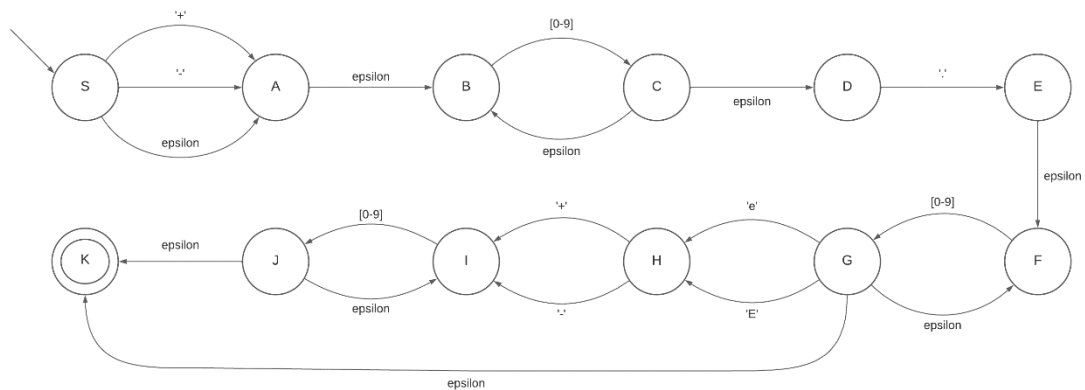
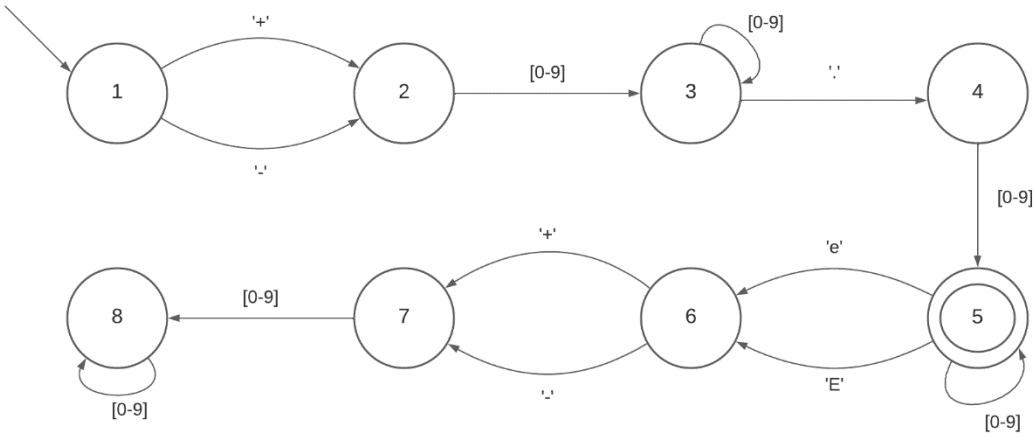


Tabela de transição

Tabela de Transição						
Estado	+	-	.	E	e	[0-9]
>SAB(1)	AB(2)	AB(2)	\	\	\	\
AB(2)	\	\	\	\	\	BCD(3)
BCD(3)	\	\	EF(4)	\	\	BCD(3)
EF(4)	\	\	\	\	\	EFGK(5)
*EFGK(5)	\	\	\	GH(6)	GH(6)	*EFGK(5)
GH(6)	HI(7)	HI(7)	\	\	\	\
HI(7)	\	\	\	\	\	IJK(8)
*IJK(8)						*IJK(8)

AFD



## Algoritmo para alguns exemplos

```
algoritmo.py X
algoritmo.py > ...
1  import re
2
3  er = r'(\+|-)?\d+\.\d+([e|E](\+|-)?\d+)?'
4  er_regex = re.compile(er)
5
6  t1 = "-3423.12413432"
7  t2 = "+12312.58912"
8  t3 = "3.14"
9
10 list_tocheck = [+36.423, -3.45e7, 5.41e-8, -0.45E22, 4.56, t1, t2, t3]
11 list_tocheck = [str(x) for x in list_tocheck]
12
13 for element in list_tocheck:
14     if er_regex.fullmatch(element):
15         print(f"{element} é um exemplo válido de constante com virgula flutuante")
16     else:
17         print(f"{element} não é um exemplo válido de constante com virgula flutuante")
```

d)

Grafo utilizando a linguagem graphviz

```
grafo.py X
grafo.py > ...
1  import graphviz
2
3  dot = graphviz.Graph()
4  dot.attr(rankdir='LR')
5  dot.node('', shape='none')
6  dot.node('1')
7  dot.node('2')
8  dot.node('3')
9  dot.node('4')
10 dot.node('5', shape='doublecircle')
11 dot.node('6')
12 dot.node('7')
13 dot.node('8', shape='doublecircle')
14
15 dot.edge('', '1', dir='forward')
16 dot.edge('1', '2', label='+', dir='forward')
17 dot.edge('1', '2', label='-', dir='forward')
18 dot.edge('2', '3', label='[0-9]', dir='forward')
19 dot.edge('3', '3', label='[0-9]', dir='forward')
20 dot.edge('3', '4', label='.', dir='forward')
21 dot.edge('4', '5', label='[0-9]', dir='forward')
22 dot.edge('5', '5', label='[0-9]', dir='forward')
23 dot.edge('5', '6', label='e', dir='forward')
24 dot.edge('5', '6', label='E', dir='forward')
25 dot.edge('6', '7', label='+', dir='forward')
26 dot.edge('6', '7', label='-', dir='forward')
27 dot.edge('7', '8', label='[0-9]', dir='forward')
28 dot.edge('8', '8', label='[0-9]', dir='forward')
29
30 dot.render('grafo', format='pdf')
```