

Relatório
Processamento de Linguagens

Trabalho Prático 1 – Exercício 1

Alunos:
a16443 – Francisco Moreira Rebêlo
a21958– Gabriel Jablonsky
a23525 – Diogo Parente

Docente:
Óscar Ribeiro

Escola Superior de Tecnologia
Licenciatura em Engenharia de Sistemas Informáticos - PL

Barcelos, 15 de abril de 2023

Índice

1. Alínea A	4
1.1. Expressão regular definida:	4
1.1.1. Explicação:	4
1.1.2. Parte 1	4
1.1.3. Parte 2	4
1.1.4. Parte 3	4
1.1.5. Parte 4	4
1.1.6. Teste com programa	4
2. Alínea B + D	5
3. Alínea C	5
3.1. Tabela de transição	5
3.2. Algoritmo para testes	6
3.2.1. Resultados obtidos	6

1. Alínea A

1.1. Expressão regular definida:

`((\+|-)[\d]+\.[\d]+[E]+[\d]+)|([\d]+\.[\d]+[E]+[\d]+)|((\+|-)([\d]+\.[\d]+[E](\+|-)+[\d]+)|([\d]+\.[\d]+[E](\+|-)[\d]+)))`

1.1.1. Explicação:

1.1.2. Parte 1

`((\+|-)[\d]+\.[\d]+[E]+[\d]+)` → Números reconhecidos: +10.5E45

1.1.3. Parte 2

`([\d]+\.[\d]+[E]+[\d]+)` → Números reconhecidos: 10.10E45

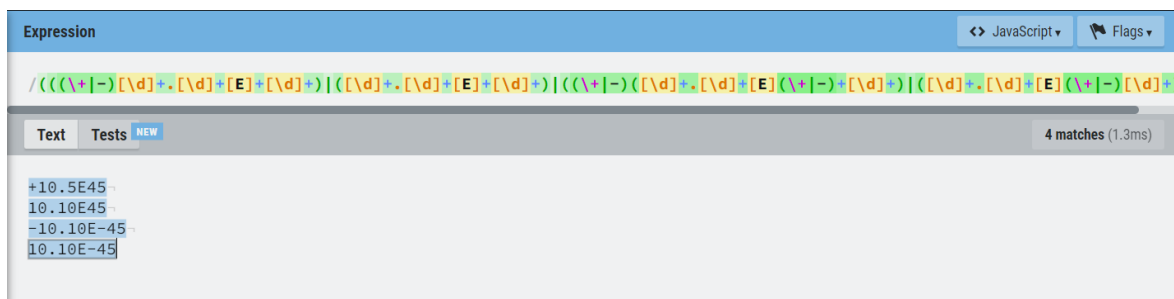
1.1.4. Parte 3

`(\+|-)([\d]+\.[\d]+[E](\+|-)+[\d]+)` → Números reconhecidos: -10.10E-45

1.1.5. Parte 4

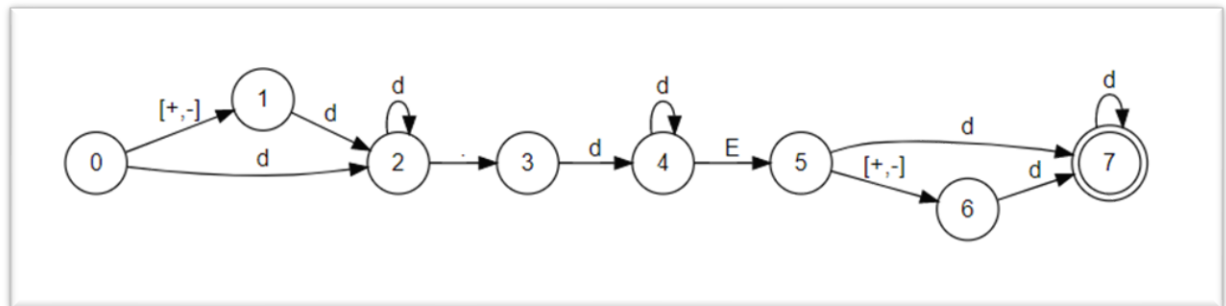
`([\d]+\.[\d]+[E](\+|-)[\d]+)` → Números reconhecidos: 10.10E-45

1.1.6. Teste com programa rexegr



2. Alínea B + D

2.1. Autómato determinista em graphviz



3. Alínea C

3.1. Tabela de transição

	[+,-]	\d	\d	\d	.	\d	\d	E	[+,-]	\d	\d	\d
Q0	Q1	Q2										
Q1			Q2									
Q2				Q2	Q3							
Q3						Q4						
Q4							Q4	Q5				
Q5									Q6	Q7		
Q6											Q7	
Q7												Q7

3.2. Algoritmo para testes.

```
V = {"+", "-", "range (10)", "."}
Q = {"q0", "q1", "q2", "q3", "q4", "q5", "q6", "q7"}
delta = {"q0": {"+": "q1", "-": "q1", "range (10)": "q2"},
        "q1": {"range (10)": "q2"},
        "q2": {"range (10)": "q2", ".": "q3"},
        "q3": {"range (10)": "q4"},
        "q4": {"range (10)": "q4", "E": "q5"},
        "q5": {"+": "q6", "-": "q6", "range (10)": "q7"},
        "q6": {"range (10)": "q7"},
        "q7": {"range (10)": "q7", "E": "q5"},
        }
q0 = "q0"
F = {"q7"}

def reconhece(palavra:str):
    def reconhecedigitos0_9(c):
        return c.isdigit() and c in "0123456789"

    estado_atual= "q0"
    tam = len(palavra)
    i = 0
    while (i<tam) and (estado_atual != "Erro"):
        simbolo_atual = palavra[i]
        if simbolo_atual in V:
            estado_atual = delta[estado_atual][simbolo_atual]
        elif reconhecedigitos0_9(simbolo_atual):
            estado_atual = delta[estado_atual]["range (10)"]
        elif simbolo_atual == "E":
            estado_atual = delta[estado_atual]["E"]
        else:
            estado_atual = "Erro"
        i+=1
    return (estado_atual in F) and (i==tam)

for exemplo in ["12.5", "-12.5", "12.55E-10", "-12.45E-10", "125",
"12.55E10"]:
    print(f"'{exemplo}'\t{reconhece(exemplo)}")
```

3.2.1. Resultados obtidos

```
PS C:\Users\diogo> & C:/Users/diogo/AppData/Local/Programs/Python/Python311/python.exe c:/Users/diogo/Downloads/teste5.py
'12.5' False
'-12.5' False
'12.55E-10' True
'-12.45E-10' True
'125' False
'12.55E10' True
PS C:\Users\diogo>
```