

## Exercícios de Análise Léxica

1 – Vamos projetar um diagrama de estados para reconhecer os literais de ponto flutuante de sua linguagem de programação favorita?

*Sugestão de diagrama de estados (autômato finito)*

Formato: tabela de transição e estados

*Estado Inicial: q0*

Transição	Leitura
q0 -> q1	dígito
q1 -> q2	dígito
q2 -> q2	dígito
q1 -> q3	ponto ( . )
q2 -> q3	ponto ( . )
q3 -> q4	dígito
q4 -> q4	dígito

a) Você poderia melhorar este diagrama de estados? Caso sim, faça sua sugestão. Além disso, utilizando o **JFlap** faça o diagrama de estados proposto e teste.

*Temos aqui algumas sugestões de literais de ponto flutuante para testar (Java), mas você pode usar também as suas.*

123.54

0.003

4.132492653589793

20.

.123

123

abcd

b) Agora temos uma sugestão de código em java que representa o diagrama de estados. Você quer testar o mesmo? Verificar se há adaptações a serem feitas? Corrigir se for necessário? Ou prefere fazer um código em sua linguagem e programação favorita? Também, se você propôs o seu diagrama de estados, espera-se o seu código. Fique à vontade.

*Obs.: Você precisa de um arquivo de entrada para testar os literais.*

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class ReconhecedorLiteralPontoFlutuante {
    private static BufferedReader in_fp;
    private static char proximoCaractere;
    private static int estadoAtual;

    public static void main(String[] args) {

        try {
            in_fp = new BufferedReader(new FileReader("entrada.txt"));
            String line;
            while ((line = in_fp.readLine()) != null) {
                System.out.println("Testando: " + line);
                if (reconhecerLiteralPontoFlutuante(line)) {
                    System.out.println("Literal de ponto flutuante válido: " + line);
                } else {
                    System.out.println("Literal de ponto flutuante inválido: " + line);
                }
            }
        } catch (IOException e) {
            System.out.println("Erro ao ler o arquivo.");
        }
    }
}
```

```

    }

    }

    private static boolean reconhecerLiteralPontoFlutuante(String entrada) {
        estadoAtual = 0;
        for (int i = 0; i < entrada.length(); i++) {
            char caractere = entrada.charAt(i);
            transicao(caractere);
        }
        return estadoAtual == 4; // Estado FINAL
    }

    private static void transicao(char caractere) {
        switch (estadoAtual) {
            case 0:
                if (Character.isDigit(caractere)) {
                    estadoAtual = 1;
                } else {
                    estadoAtual = -1; // Estado inválido
                }
                break;
            case 1:
                if (Character.isDigit(caractere)) {
                    estadoAtual = 2;
                } else if (caractere == '.') {
                    estadoAtual = 3;
                } else {
                    estadoAtual = -1; // Estado inválido
                }
                break;
            case 2:
                if (Character.isDigit(caractere)) {
                    // Permanece no estado 2
                } else if (caractere == '.') {
                    estadoAtual = 3;
                } else {
                    estadoAtual = -1; // Estado inválido
                }
                break;
            case 3:
                if (Character.isDigit(caractere)) {
                    estadoAtual = 4;
                } else {
                    estadoAtual = -1; // Estado inválido
                }
                break;
            case 4:
                if (Character.isDigit(caractere)) {
                    // Permanece no estado 4
                } else {
                    estadoAtual = -1; // Estado inválido
                }
                break;
        }
    }
}

```

**2 – Com base no exercício anterior, tenho duas sugestões para você:**

**a) projetar um diagrama de estados para reconhecer uma das formas de comentários das linguagens de programação mais conhecidas. Sugestão: inicia com /\* e termina com \*/.**

**Ou**

**b) projetar um diagrama de estados para reconhecer identificadores de linguagem de programação de sua preferência.**

Posteriormente a isso: faça o diagrama de estados no JFlap, implemente da linguagem de programação de sua preferência e faça os testes.