

Desafio: Implementação de Autômatos

Disciplina: Linguagem de Programação

Conceitos de Autômatos

Autômatos Finitos

- ▶ O Autômato Finito é uma máquina, reconhecedora de palavras ou cadeia de caracteres, que sempre retorna uma resposta SIM (cadeia reconhecida) ou NÃO (cadeia não reconhecida)



AFD – Definição Formal

▶ Definição formal de um Autômato Finito Determinístico:

- ▶ Um AFD M é uma **5-upla**:

$$M = (Q, \Sigma, \delta, q_0, F), \text{ onde:}$$

- ▶ **Q** : conjunto finito de **estados** S
- ▶ **Σ** : alfabeto de **símbolos de entrada**
- ▶ **δ** : Funções de transição que mapeiam um par estado-símbolo de entrada em um novo estado
- ▶ **q_0** : um **estado inicial** S_0
- ▶ **F** : um conjunto de **estados finais** para aceitação de cadeias

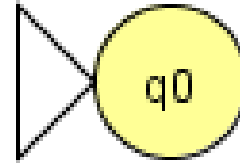
▶ Reconhecimento de cadeias válidas:

- ▶ Uma **cadeia é reconhecida** se existe um percurso do **estado inicial** até um **estado final**
-



AFD – Representação Gráfica

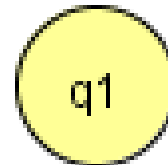
- ▶ Um AFD pode ser representado por meio de um diagrama similar ao de “máquinas de estados finitos”



Estado inicial

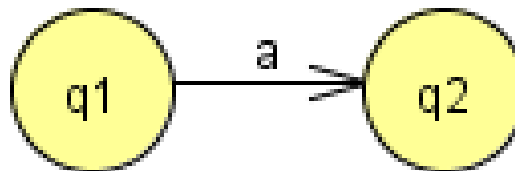


Estado final



Demais estados

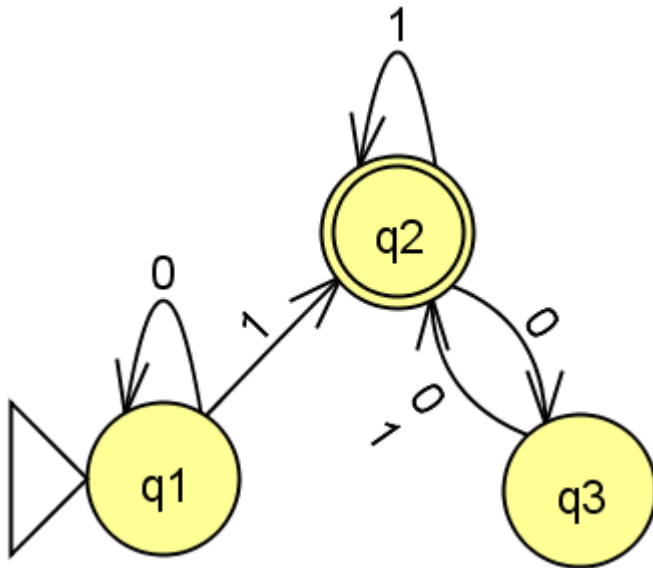
- ▶ Serve como uma representação mais intuitiva das transições



Transição $\delta(q1, a) = q2$

AFD – Exemplo

Um autômato finito M_1



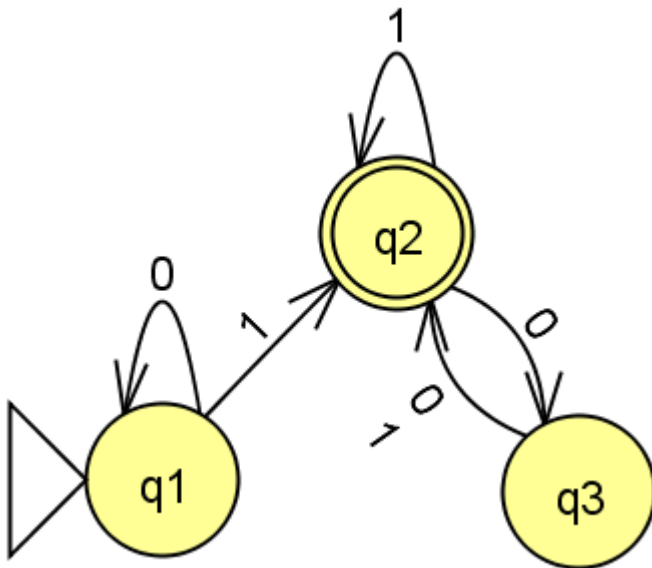
1 é uma cadeia válida?
10 é uma cadeia válida?

- ▶ $S = \{q1, q2, q3\}$
- ▶ $\Sigma = \{0, 1\}$
- ▶ $S_0 = \{q1\}$
- ▶ $F = \{q3\}$

Estado	Símbolo de entrada	
	0	1
q1	{q1}	{q2}
q2	{q3}	{q2}
q3	{q2}	{q2}

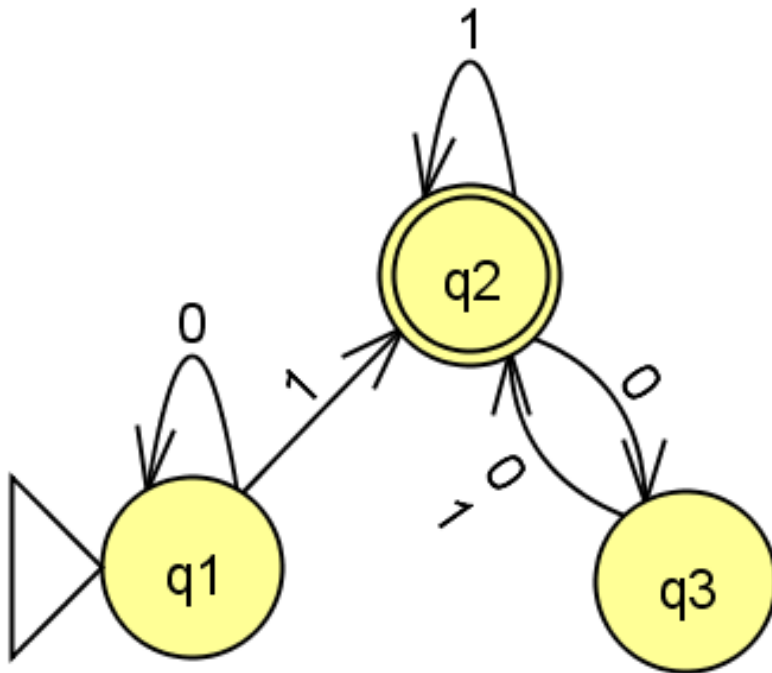
AFD – Exemplo

Entrada 1101



1. Inicia no estado **q1**
2. Lê **1**, segue transição de **q1** para **q2**
3. Lê **1**, segue transição de **q2** para **q2**
4. Lê **0**, segue transição de **q2** para **q3**
5. Lê **1**, segue transição de **q3** para **q2**
6. Para com saída e reconhece

AFD – Exemplo



- Testar: 1, 01, 11, 0101 (em M_1)

Implementação do Autômato

```
//Estado final q2
int q2(int simbolo){
    exibe(2, simbolo);
    switch(cadeia[simbolo++){
        case '1' :
            return q2(simbolo);
        case '0' :
            return q3(simbolo);
        case '\0' :
            return 1; //caso fim de string, retorna VALIDA
        default :
            return 0; //INVALIDA
    }
}
```

```
//Estado inicial q1
int q1(int simbolo){
    exibe(1, simbolo);
    switch(cadeia[simbolo++){
        case '0' :
            return q1(simbolo);
        case '1' :
            return q2(simbolo);
        default :
            return 0; //INVALIDA
    }
}
```

```
//Estado q3
int q3(int simbolo){
    exibe(3, simbolo);
    switch(cadeia[simbolo++){
        case '0' :
            return q2(simbolo);
        case '1' :
            return q2(simbolo);
        default :
            return 0; //INVALIDA
    }
}
```



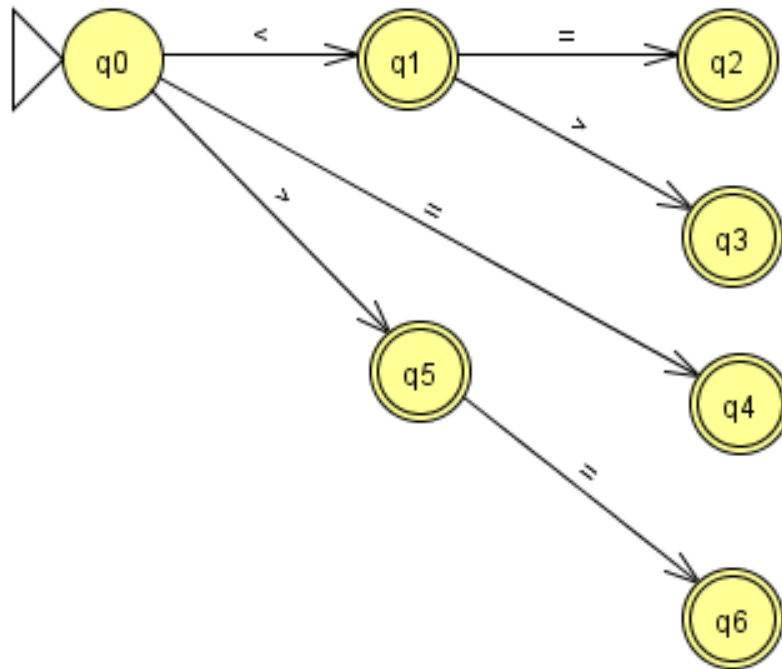


Exercícios Desafio

Implementar os Seguintes AFD

1) Reconhecer todos os operadores relacionais:

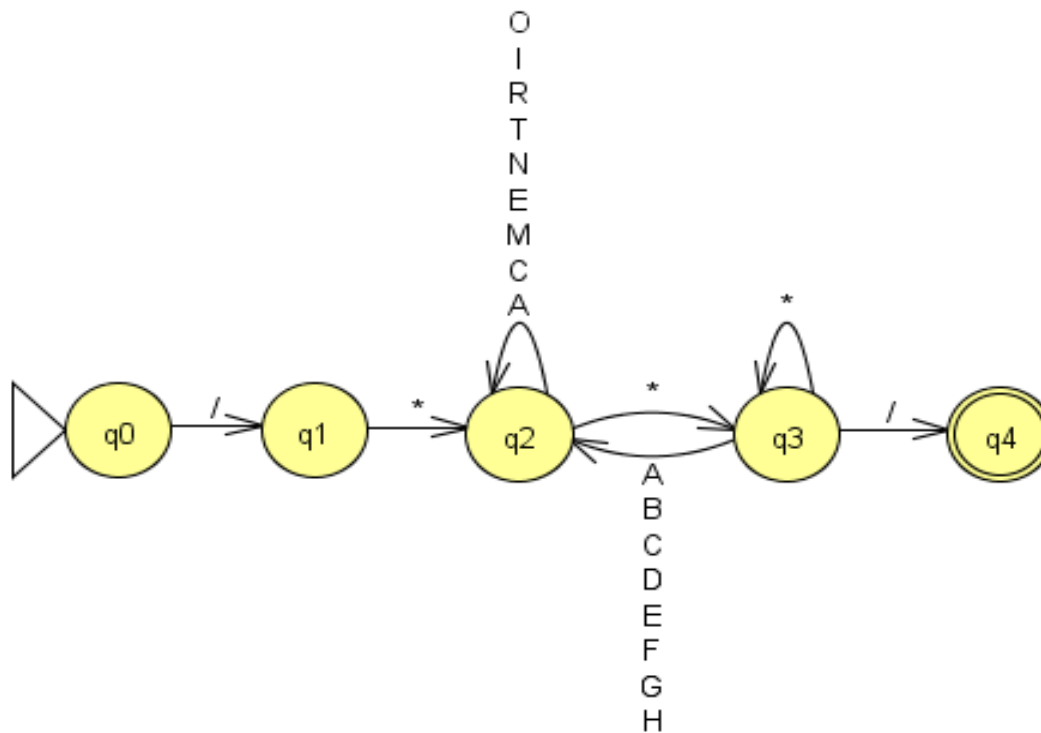
$>$, $>=$, $<$, $<=$, $=$, $<>$



Implementar os Seguintes AFD

2. Reconhecer comentários da forma:

`/*COMENTARIO****COMENTARIO*/`



Implementar os Seguintes AFD

3. Reconhecer nomes válidos para variáveis e números inteiros e ponto flutuante.

