Linguagens Formais e Autômatos (LFA)

Aula de 02/10/2013

Máquinas de Moore e Mealy
Implementação e exercícios

©Luiz M. Afonso e Clarisse S. de Souza, 2013

Recapitulando a aula anterior

- **Transdutores finitos** são extensões de AFDs que, a partir da leitura dos símbolos da cadeia de entrada, formada por símbolos do alfabeto Σ, gravam símbolos na cadeia de saída, pertencentes ao alfabeto Δ

- **Máquina de Moore**: função de transdução definida sobre os **estados** do autômato (λ: $Q \rightarrow \Delta^*$)

- **Máquina de Mealy**: função de transdução definida sobre as transições do autômato (λ: $Q \times \Sigma \rightarrow \Delta^*$)

©Luiz M. Afonso e Clarisse S. de Souza, 2013

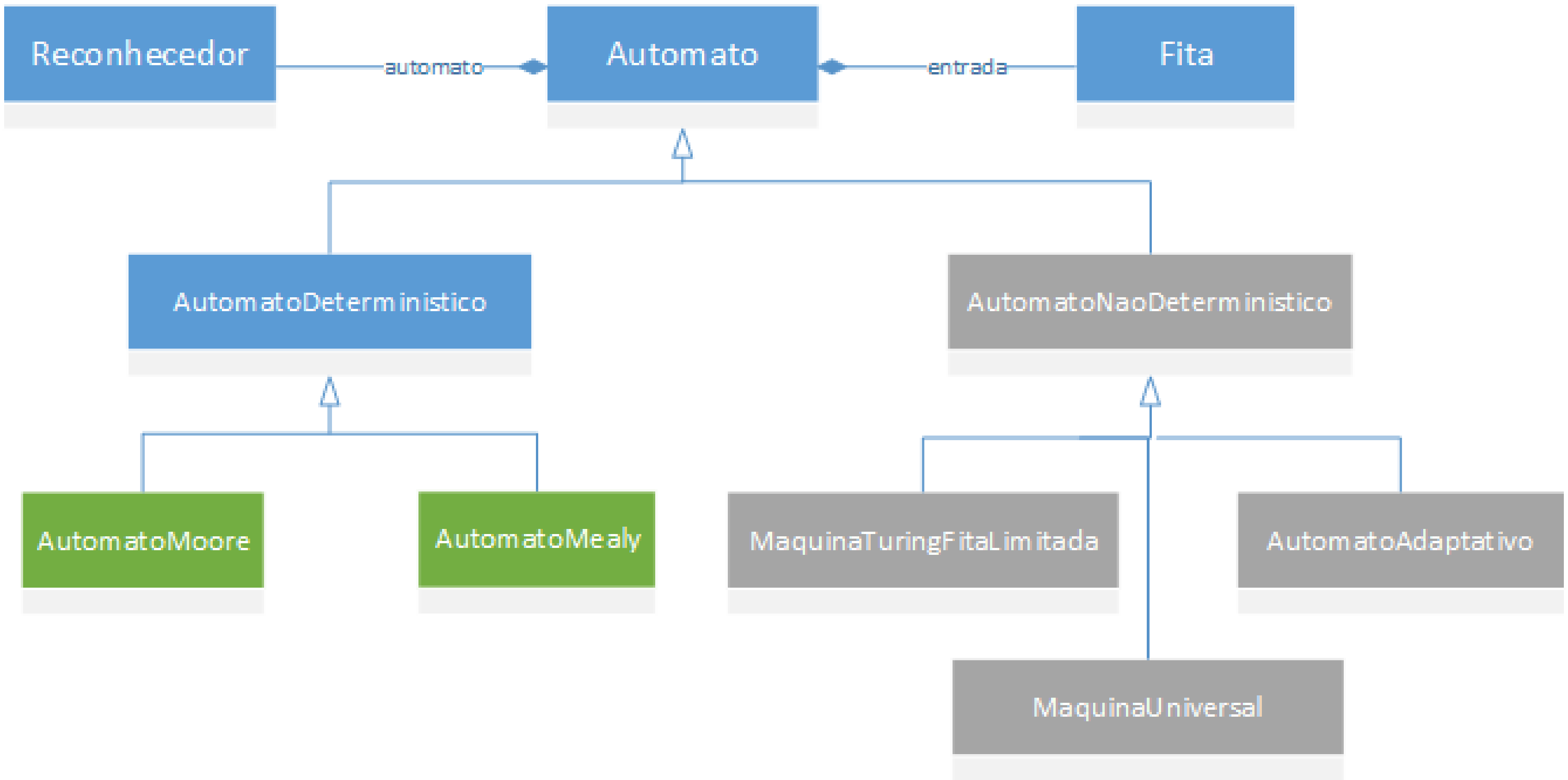
Conteúdo da aula 16

- Visão geral da implementação das máquinas de Moore e Mealy em Ruby

- Exercícios sobre transdutores finitos, extraídos do livro-texto (Ramos,2009 cap.3)

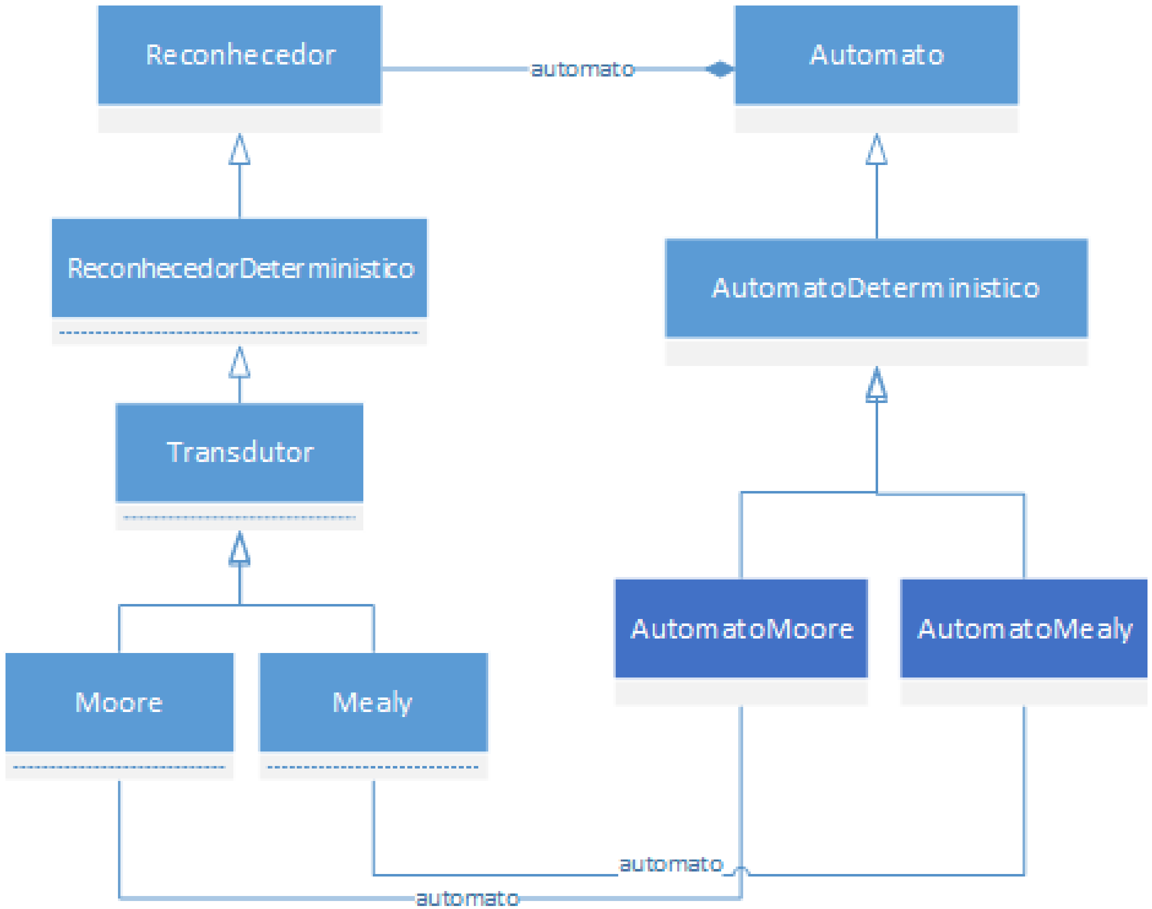
©Luiz M. Afonso e Clarisse S. de Souza, 2013

Revisão da arquitetura de implementação do livro



©Luiz M. Afonso e Clarisse S. de Souza, 2013

Implementação de transdutores - visão detalhada



©Luiz M. Afonso e Clarisse S. de Souza, 2013

Novas classes

- Transdutor
- Máquina de Moore
  - Moore
  - AutomatoMoore
  - MovimentacaoMoore
- Máquina de Mealy:
  - Mealy
  - AutomatoMealy
  - MovimentacaoMealy

©Luiz M. Afonso e Clarisse S. de Souza, 2013

Classe Transdutor (moore/Transdutor.rb)

```
class Transdutor < ReconhecedorDeterministico
    attr_accessor :lambda

    def instanciarEstruturaEspecifica()
        @lambda = {}
    end

    def adicionarLambda( traducao )
        @lambda.update( traducao )
    end

    def traduzir()
        analisar()
    end

end
```


©Luiz M. Afonso e Clarisse S. de Souza, 2013

Classe Moore (moore/Moore.rb)

```
class Moore < Transdutor
    def instanciarAutomato( estadoInicial, estadosFinais )
        @automato = AutomatoMoore.new( estadoInicial, estadosFinais )
        @automato.criarVinculo( self )
        @resultado = ''
    end

    def traduzirEstado()
        print( @lambda[ @automato.consulta.estadoCorrente?() ] )
    end
end
```

©Luiz M. Afonso e Clarisse S. de Souza, 2013

Classe AutomatoMoore (moore/AutomatoMoore.rb)

```
class AutomatoMoore < AutomatoDeterministico
    attr_accessor :moore

    def criarVinculo( moore )
        @moore = moore
    end

    def instanciarMovimentacao()
        @movimentacao = MovimentacaoMoore.new( self )
    end
end
```

©Luiz M. Afonso e Clarisse S. de Souza, 2013

Classe MovimentacaoMoore (moore/servico/MovimentacaoMoore.rb)

```
class MovimentacaoMoore < MovimentacaoDeterministica
    def mover( estadosSeguintes )
        @automato.moore.traduzirEstado()
        super( estadosSeguintes )
    end
end
```

©Luiz M. Afonso e Clarisse S. de Souza, 2013

Classe Mealy (mealy/Mealy.rb)

```
class Mealy < Transdutor
    def instanciarAutomato( estadoInicial, estadosFinais )
        @automato = AutomatoMealy.new( estadoInicial, estadosFinais )
        @automato.criarVinculo( self )
    end

    def traduzirTransicao()
        print @lambda[ [@automato.consulta.estadoCorrente?(),
                  @automato.entrada.ler()] ]
    end
end
```

©Luiz M. Afonso e Clarisse S. de Souza, 2013

Classe AutomatoMealy (mealy/AutomatoMealy.rb)

```
class AutomatoMealy < AutomatoDeterministico
    attr_accessor :mealy

    def criarVinculo( mealy )
        @mealy = mealy
    end

    def instanciarMovimentacao()
        @movimentacao = MovimentacaoMealy.new( self )
    end
end
```

©Luiz M. Afonso e Clarisse S. de Souza, 2013

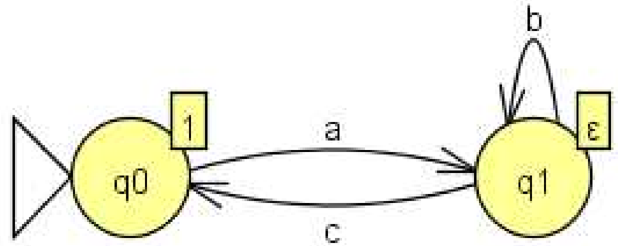
Classe MovimentacaoMealy (mealy/servico/MovimentacaoMealy.rb)

```
class MovimentacaoMealy < MovimentacaoDeterministica
    def mover( estadosSeguintes )
        @automato.mealy.traduzirTransicao()
        super( estadosSeguintes )
    end
end
```

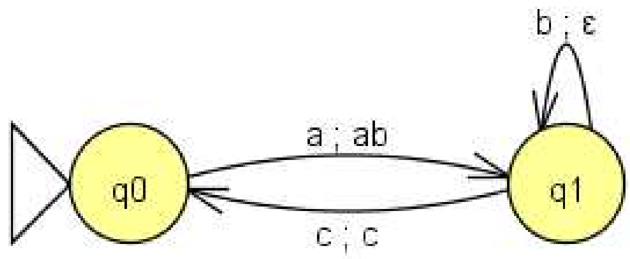
©Luiz M. Afonso e Clarisse S. de Souza, 2013

Arquivos de teste

- moore/CasoUso.rb



- mealy/CasoUso.rb



Exercícios sobre transdutores finitos

- A seguir, alguns exercícios sobre o tema selecionados do livro-texto
- Os exercícios podem ser feitos em dupla
- As soluções serão discutidas na sequência

©Luiz M. Afonso e Clarisse S. de Souza, 2013

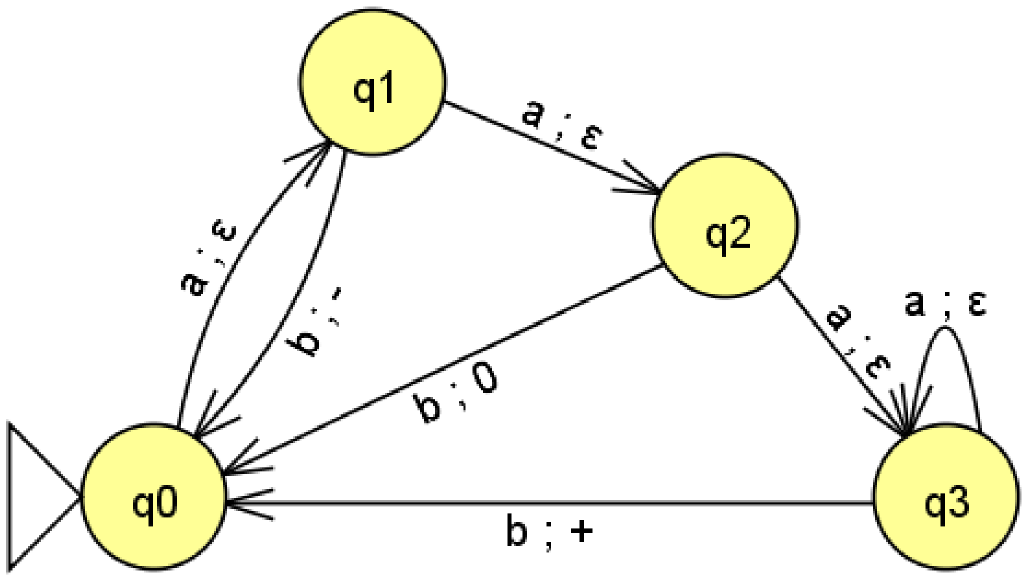
Ex. 108 (Ramos p.288)

A expressão regular (a+b)* representa uma linguagem cujas sentenças são sequências arbitrárias de um ou mais símbolos a terminadas por um símbolo b. São exemplos de sentenças dessa linguagem: ε , ab, aaababaab e aaaab. Defina formalmente um transdutor finito que aceite essa linguagem como entrada e gere na saída cadeias sobre {-,0,+}, da seguinte forma:

- Para cada subcadeia $\alpha \in a^+$, se $|\alpha|=1$ então o símbolo '-' é emitido na saída;
- Se $|\alpha|=2$, o símbolo '0' é emitido na saída;
- Se $|\alpha| \geq 3$, o símbolo '+' é emitido na saída.
- Exemplos:
  - ε produz ε
  - *ab* produz –
  - *aaababaab* produz +-0

©Luiz M. Afonso e Clarisse S. de Souza, 2013

Ex. 108 – solução 1 (Mealy)



$T = (Q, \Sigma, \Delta, \delta, \lambda, q0, F)$

$Q = \{q0,q1,q2,q3\}$

$\Sigma = \{a,b\}$

$\Delta = \{-,0,+\}$

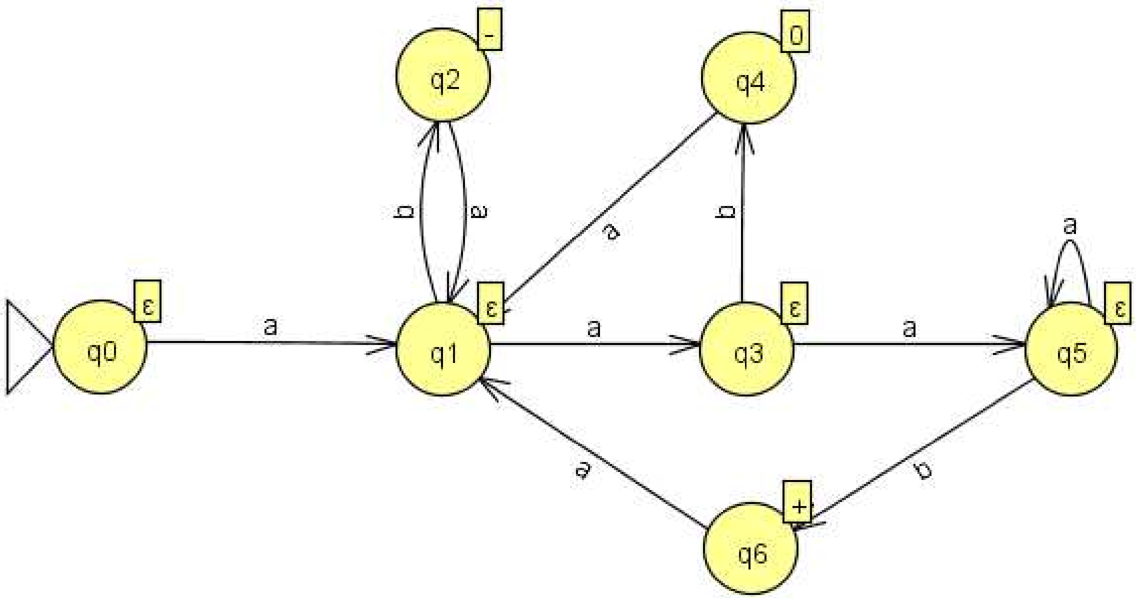
$\delta=\{(q0,a)\rightarrow q1, (q1,a)\rightarrow q2,$
$(q1,b)\rightarrow q0, (q2,a)\rightarrow q3,$
$(q2,b)\rightarrow q0, (q3,a)\rightarrow q3,$
$(q3,b)\rightarrow q0\}$

$\lambda=\{(q0,a)\rightarrow \varepsilon, (q1,a)\rightarrow \varepsilon,$
$(q1,b)\rightarrow -, (q2,a)\rightarrow \varepsilon,$
$(q2,b)\rightarrow 0, (q3,a)\rightarrow \varepsilon,$
$(q3,b)\rightarrow +\}$

$F = \{q0\}$

©Luiz M. Afonso e Clarisse S. de Souza, 2013

Ex. 108 – solução 2 (Moore)



$T = (Q, \Sigma, \Delta, \delta, \lambda, q0, F)$

$Q = \{q0,q1,q2,q3,q4,q5,q6\}$

$\Sigma = \{a,b\}$

$\Delta = \{-,0,+\}$

$\delta=\{(q0,a)\rightarrow q1, (q1,a)\rightarrow q3, (q1,b)\rightarrow q2, (q2,a)\rightarrow q1, (q3,a)\rightarrow q5, (q3,b)\rightarrow q4, (q4,a)\rightarrow q1, (q5,a)\rightarrow q5, (q5,b)\rightarrow q6, (q6,a)\rightarrow q1\}$

$\lambda=\{ q0\rightarrow \varepsilon, q1\rightarrow \varepsilon, q2\rightarrow -, q3\rightarrow \varepsilon, q4\rightarrow 0, q5\rightarrow \varepsilon, q6\rightarrow +\}$

$F = \{q0,q2,q4,q6\}$

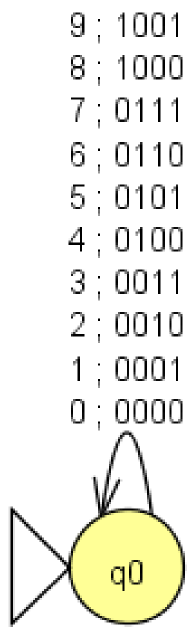
Ex. 112 – Binary Coded Decimal

Construa um transdutor finito que aceite como entrada a linguagem dos números inteiros decimais maiores ou iguais a zero, e gere na saída a representação equivalente em BCD – Binary Coded Decimal (0 → 0000, 1 → 0001, ... 9 → 1001).

Por exemplo, a cadeia de entrada 308 deve gerar na saída a cadeia 001100001000.

©Luiz M. Afonso e Clarisse S. de Souza, 2013

Ex. 112 – solução com Mealy



Obs: esse é um caso típico em que a escolha do tipo de transdutor influencia diretamente a facilidade de resolução do problema.

©Luiz M. Afonso e Clarisse S. de Souza, 2013

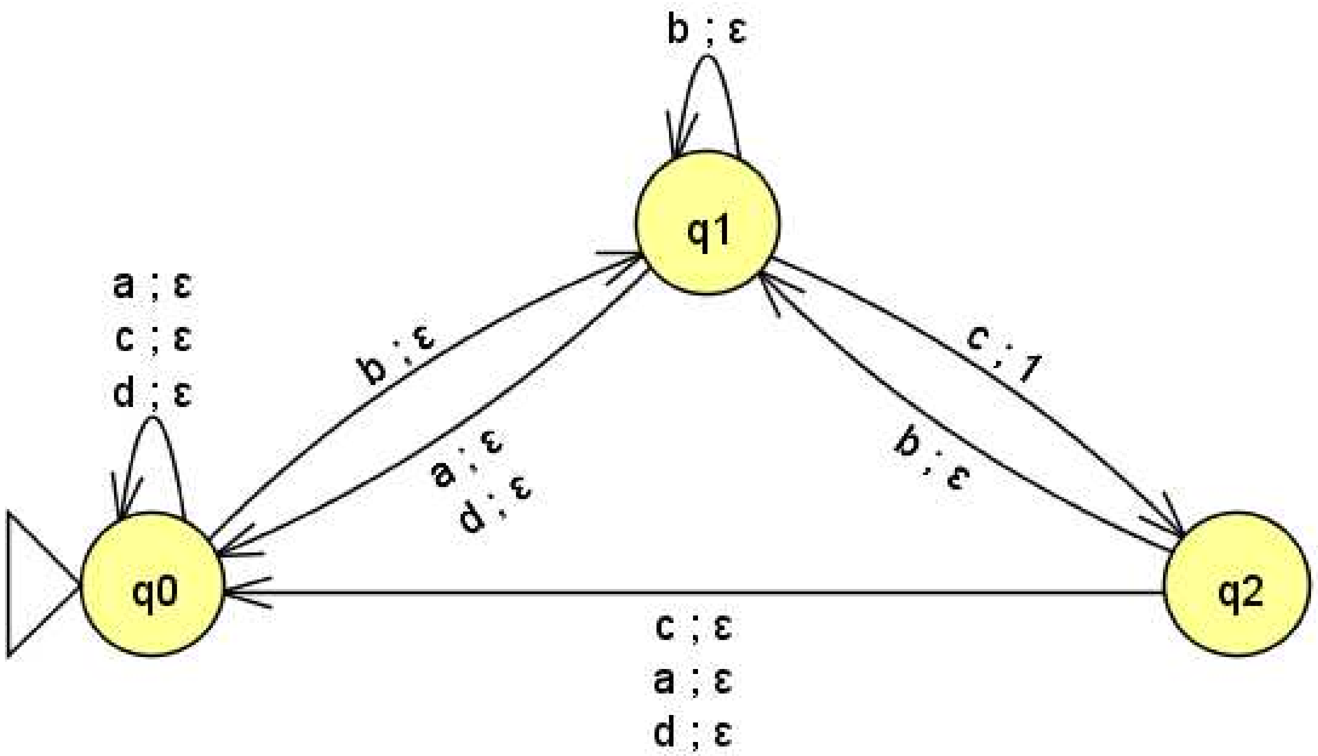
Ex. 119

Construa um transdutor finito (Mealy ou Moore) para a linguagem de entrada (a|b|c|d)*, gerando a linguagem de saída L $\subseteq$ 1*, de tal forma que a quantidade de símbolos '1' na cadeia de saída indique a quantidade de subcadeias da forma bcd* presentes na cadeia de entrada.

Exemplos de entradas e correspondentes saídas:

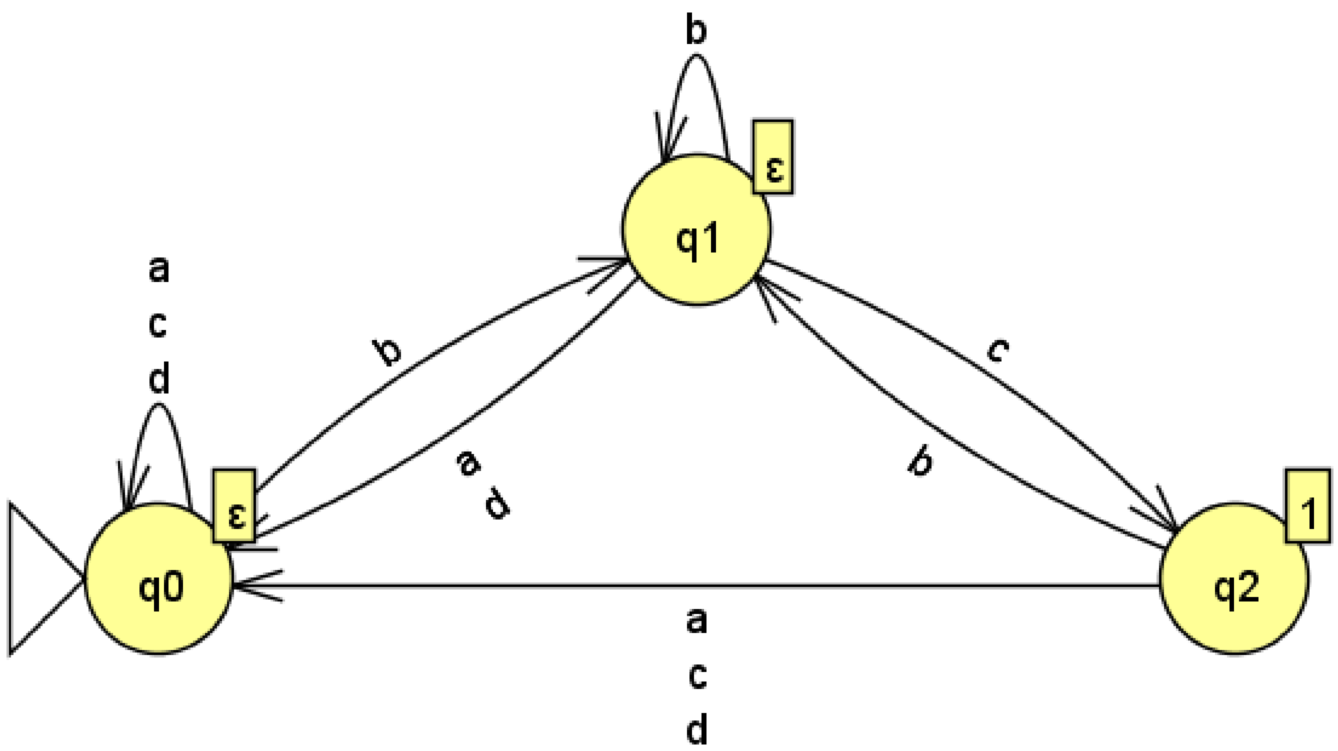
- abcdacbcdddbcacbcd gera 1111
- bcdabcddcaa gera 11
- aaaacdb gera ε

Ex. 119 – solução com Mealy



©Luiz M. Afonso e Clarisse S. de Souza, 2013

Ex. 119 – solução com Moore



©Luiz M. Afonso e Clarisse S. de Souza, 2013

Ex. 122 – para casa

Considere as linguagens de entrada Le e de saída Ls definidas a seguir:

- Le = {a,b,c}*
- Ls $\subseteq$ {a,b,c,3,4,5}*

Obtenha um transdutor finito que efetue o mapeamento de w$\in$Le para w'$\in$Ls, de tal forma que w' seja uma representação compacta da cadeia w, conforme o seguinte critério: toda subcadeia presente na cadeia de entrada w que contenha 3, 4 ou 5 símbolos repetidos em sequência deverá ser substituída, na cadeia de saída w', pela subcadeia correspondente formada pelo símbolo que se repete e o número 3, 4 ou 5. São exemplos de transdução: $\varepsilon \rightarrow \varepsilon$, a → a, cccc → c4, abca → abca, ccccccccb → c5c3b e aaaabcaaabb → a4bca3bb.

©Luiz M. Afonso e Clarisse S. de Souza, 2013