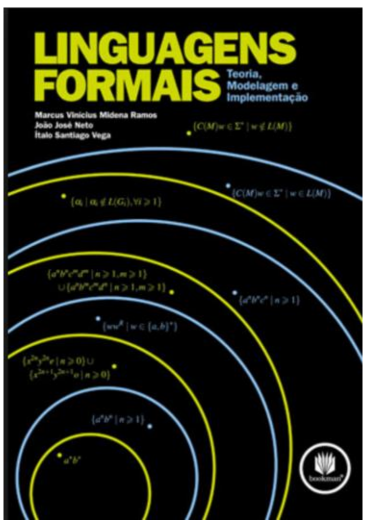
FACULDADE DE COMPUTAÇÃO E INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
Linguagens Formais e Autômatos – Aula 04 – 2º SEMESTRE/2012
Prof. Luciano Silva

TEORIA: AUTÔMATOS FINITOS NÃO-DETERMINÍSTICOS

Nossos **objetivos**nesta aula são:

- conhecer o conceito de autômatos finitos não-determinísticos
- praticar com autômatos finitos não-determinísticos



Para esta semana, usamos como referência a **Seção3.3 (Autômatos Finitos, somente autômatos não-determinísticos)** do nosso livro da referência básica:

RAMOS, M.V.M., JOSÉ NETO, J., VEJA, I.S. **Linguagens Formais: Teoria, Modelagem e Implementação**. Porto Alegre: Bookman, 2009.

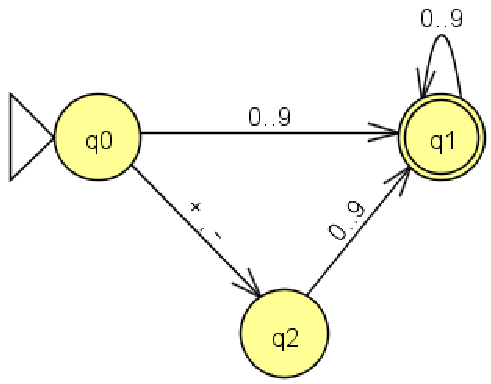
Não deixem de ler esta seção depois desta aula!

TEORIA: AUTÔMATOS FINITOS NÃO-DETERMINÍSTICOS

- Um **autômato finito não-determinístico** (afnd) M é uma 5-upla M = $(Q, \Sigma, \delta, q_0, F)$ onde:
  - Q é um conjunto finito de estados
  - Σ é um alfabeto de entrada
  - δ é uma **relação** de transição $Q \times \Sigma \times Q$
  - q_0 é o estado inicial, $q_0 \in Q$
  - F é um conjunto de estados finais, $F \subseteq Q$

  Esta definição diz, essencialmente, que a partir de um estado podemos ter mais de uma transição com a mesma letra ou não ter uma transição com uma determinada letra.

- Um afnd também pode ser representado por um grafo orientado, como mostrado no exemplo abaixo:



  Neste exemplo, observe que não temos transições com todos os símbolos do alfabeto em todos os estados.

- Assim como nos autômatos finitos determinísticos, para os autômatos finitos não determinísticospodemoster o conceito de linguagem **aceita** por um afnd M:

$$L(M) = \{ \omega \in \Sigma^* \mid (q0,\omega,q),\ q \in F \}$$

  , onde $(q0,\omega,q)$ representa as sucessivas aplicações da relação de transição para cada letra da palavra ω.

  No caso do exemplo acima, $L(M) = \{ \omega \in \{+ , -, 0..9\}^* \mid \omega$representa um número inteiro com ou sem sinal $\}$

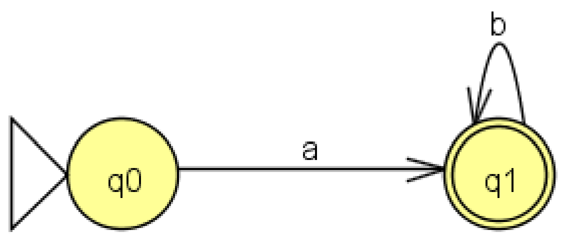
- Notação: **Rec(Σ)** = conjunto de todas as linguagens sobre Σ reconhecíveis por **autômatos finitos determinísticos**
  NRec(Σ) = conjunto de todas as linguagens sobre Σ reconhecíveis por **autômatos finitos não-determinísticos**

TEOREMA: Rec(Σ) = NRec(Σ)

- Este teorema diz, essencialmente, que os afds e os afnds têm o mesmo poder computacional, ou seja, reconhecem e aceitam as mesmas linguagens.

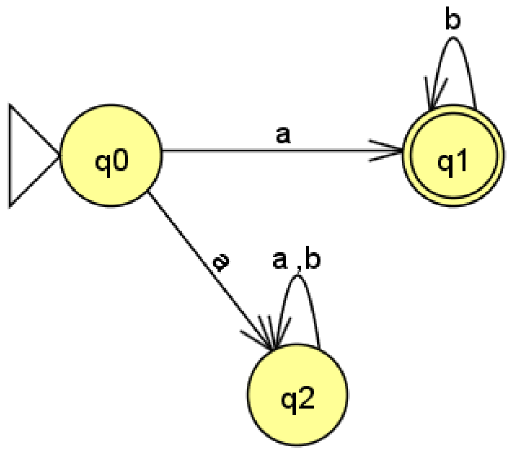
- *Esboço da prova do Teorema*:
    - **Rec(Σ)⊆NRec(Σ)**: é trivial, pois todo afd é um afnd.
    - **NRec(Σ)⊆Rec(Σ)**: esta inclusão é baseada no algoritmo de construção dos subconjuntos, que permite converter um afnd em um afd.

- **Algoritmo de conversão afnd → afd (Algoritmo de Construção dos Sub-Conjuntos)**



EXERCÍCIO TUTORIADO

Transforme o afnd abaixo para um afd:



EXERCÍCIO COM DISCUSSÃO EM DUPLAS

(a) Construa um afnd que reconheça todas as palavras sobre o alfabeto $\Sigma = \{a,b\}$ que comecem pelo segmento ab.

(b) Transforme o afnd do item (a) para um afd.

PROBLEMA

Os **analisadores léxicos** discutidos na aula anterior ficam mais simples de serem especificados via autômatos finitos não-determinísticos, devido a não-necessidade de se fazer transições com todos os símbolos do alfabeto.

Por exemplo, vamos considerar novamente a seguinte entrada de um programa em Java:

soma = -356;

Ela seria separada por um analisador léxico em soma (**identificador**), = (**operador**), -356 (**número**) e ; (**delimitador**).

Especifique um afnd que seja capaz de reconhecer todos os itens léxicos presentes neste tipo de construção em Java.

EXERCÍCIOS EXTRA-CLASSE

1. Construa um afnd que reconheça todas as palavras sobre o alfabeto $\Sigma = \{a,b\}$ que comecem pelo segmento aa.

2. Construa um afnd que reconheça todas as palavras sobre o alfabeto $\Sigma = \{a,b\}$ que possuam o segmento aa ou o segmento bb (ou ambos).

3. Construa um afnd que reconheça todas as palavras sobre o alfabeto $\Sigma = \{a,b\}$ que possuam um número par de a´s e de b´s.

4. Construa um afnd que reconheça todas as palavras sobre o alfabeto $\Sigma = \{a,b\}$ que possuam um número ímpar de a´s.

5. Converta o afnd abaixo para um afd correspondente, eliminando os estados inacessíveis: