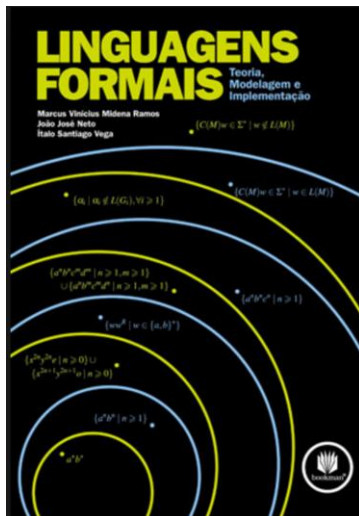


TEORIA: AUTÔMATOS FINITOS NÃO-DETERMINÍSTICOS COM TRANSIÇÕES VAZIAS



Nossos **objetivos** nesta aula são:

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



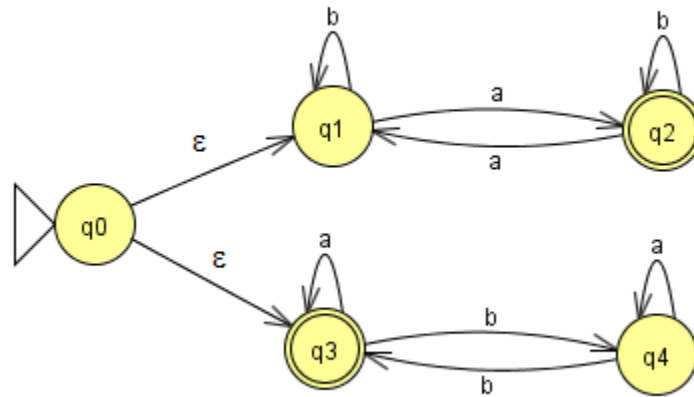
Para esta semana, usamos como referência a **Seção 3.3 (Autômatos Finitos, somente autômatos não-determinísticos com transições vazias)** do nosso livro da referência básica:

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 COM TRANSIÇÕES VAZIAS (ϵ -afnd)

- Um **autômato finito não-determinístico com transições vazias** (ϵ -afnd) M é um autômato finito não determinístico onde podemos ter transições com a palavra vazia (ϵ).
- Um ϵ -afnd também pode ser representado por um grafo orientado, como mostrado na próxima página.



- Assim como nos autômatos finitos não-determinísticos, para os autômatos finitos não-determinísticos com transições vazias podemos ter o conceito de linguagem **aceita** por um ϵ -afnd M:

$$L(M) = \{ \omega \in \Sigma^* \mid (q_0, \omega, q), q \in F \}$$

, onde (q_0, ω, q) representa as sucessivas aplicações da relação de transição para cada letra da palavra ω .

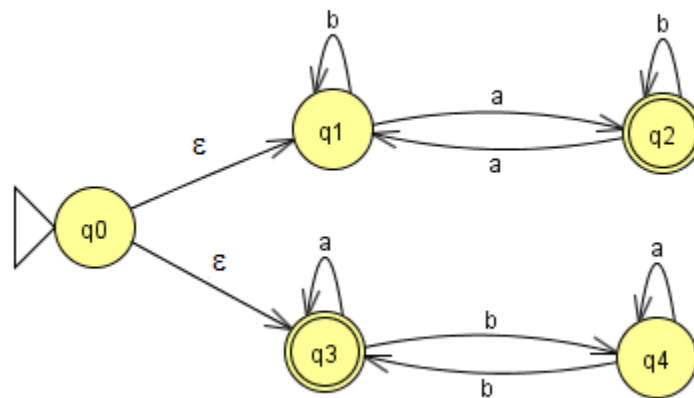
No caso do exemplo acima, $L(M) = \{ \omega \in \{a,b\}^* \mid \omega \text{ tem um número ímpar de } a\text{'s ou um número par de } b\text{'s} \}$

- Notação: **NRec(Σ)** = conjunto de todas as linguagens sobre Σ aceitáveis por **autômatos finitos não-determinísticos**
 ϵ -**NRec(Σ)** = conjunto de todas as linguagens sobre Σ aceitáveis por **autômatos finitos não-determinísticos com transições vazias**

TEOREMA: $\text{NRec}(\Sigma) = \epsilon\text{-NRec}(\Sigma)$

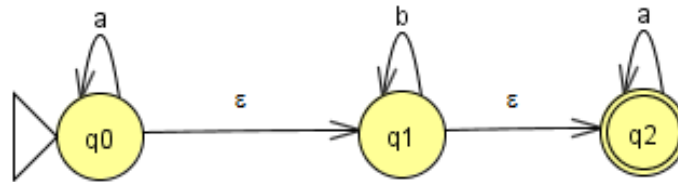
- Este teorema diz, essencialmente, que os afnds e os ϵ -afnds têm o mesmo poder computacional, ou seja, reconhecem e aceitam as mesmas linguagens.
- Esboço da prova do Teorema:*
 - $\text{NRec}(\Sigma) \subseteq \epsilon\text{-NRec}(\Sigma)$: é trivial, pois todo afnd é um ϵ -afnd.
 - $\epsilon\text{-NRec}(\Sigma) \subseteq \text{NRec}(\Sigma)$: esta inclusão é baseada no algoritmo de eliminação de transições vazias.

- Algoritmo de conversão ϵ -afnd \rightarrow afnd (Algoritmo de Eliminação de Transições Vazias)



EXERCÍCIO TUTORIADO

Transforme o ε -afnd abaixo para um afnd:



EXERCÍCIO COM DISCUSSÃO PAREADA

(a) Construa um ε -afnd que reconheça todas as palavras sobre o alfabeto $\Sigma = \{a,b\}$ que tenham 1 ou mais segmentos **ab**.

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

PROBLEMA

Os **analísadores léxicos** discutidos na aula anterior ficam mais simples de serem especificados via autômatos finitos determinísticos com transições vazias. Por exemplo, vamos considerar novamente a seguinte entrada de um programa em Java:

soma = -356;

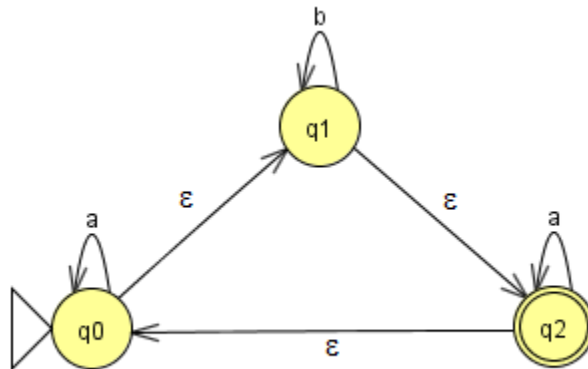
Ela seria separada por um analisador léxico em soma (**identificador**), = (**operador**), -356 (**número**) e ; (**delimitador**).

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

EXERCÍCIOS EXTRA-CLASSE

1. Construa um ε -afnd que reconheça todas as palavras sobre o alfabeto $\Sigma = \{0,1\}$ que comecem por 1 e que representem números pares.
2. Construa um ε -afnd que reconheça todas as palavras sobre o alfabeto $\Sigma = \{a,b\}$ que possuam o segmento aa ou o segmento bb.

3. Transforme o ϵ -afnd abaixo para um afnd:



4. Transforme o afnd da questão (3) para um afd.