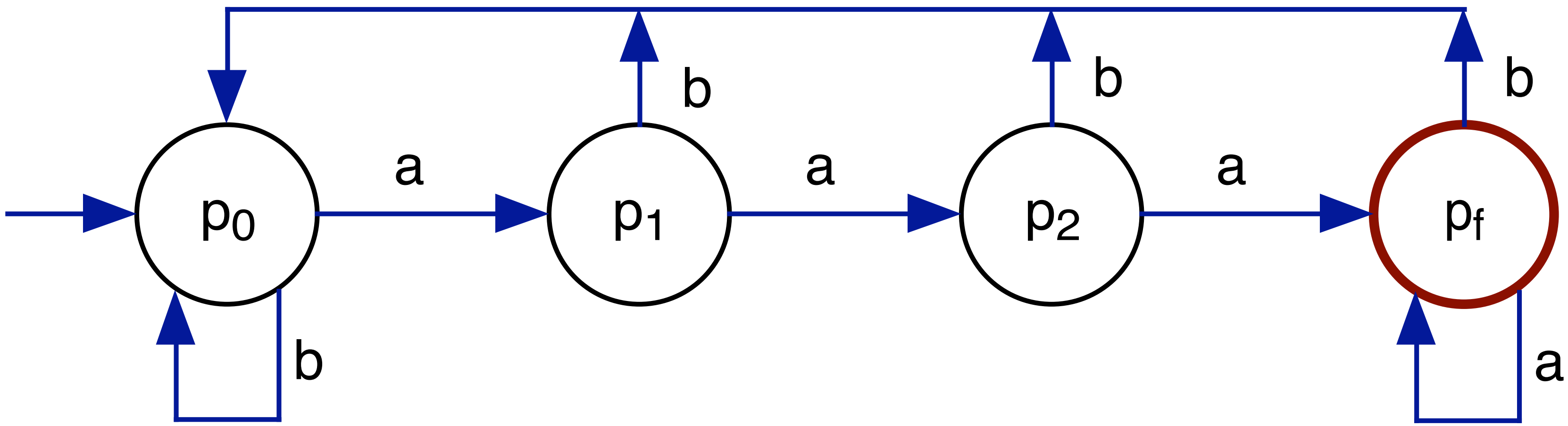
δ_{6D}	a	b
$p_0 = \langle q_0 \rangle$	$\langle q_0 q_1 \rangle$	$\langle q_0 \rangle$
$p_1 = \langle q_0 q_1 \rangle$	$\langle q_0 q_1 q_2 \rangle$	$\langle q_0 \rangle$
$p_2 = \langle q_0 q_1 q_2 \rangle$	$\langle q_0 q_1 q_2 q_f \rangle$	$\langle q_0 \rangle$
$p_f = \langle q_0 q_1 q_2 q_f \rangle$	$\langle q_0 q_1 q_2 q_f \rangle$	$\langle q_0 \rangle$



3 – Linguagens Regulares

3.5 Autômato Finito com Movimentos Vazios

◆ Movimentos vazios

- generalizam os movimentos não-determinísticos

◆ Movimento vazio

- transição sem leitura de símbolo algum da fita
- interpretado como um não-determinismo interno ao autômato
  - * transição encapsulada
  - * excetuando-se por uma eventual mudança de estados
  - * nada mais pode ser observado

◆ Algumas vantagens

- facilita algumas construções e demonstrações

◆ Poder computacional p/ autômatos finitos

- *não* aumenta o poder de reconhecimento de linguagens
- qualquer aAFNε pode ser simulado por um AFD

Def: Autômato Finito com Movimentos Vazios - AFNε

$$M = (\Sigma, Q, \delta, q_0, F)$$

- Σ - alfabeto (de símbolos) de entrada
- Q - conjunto de estados possíveis
- δ - (função) programa ou função de transição (função parcial)

$$\delta: Q \times (\Sigma \cup \{ \varepsilon \}) \rightarrow 2^Q$$

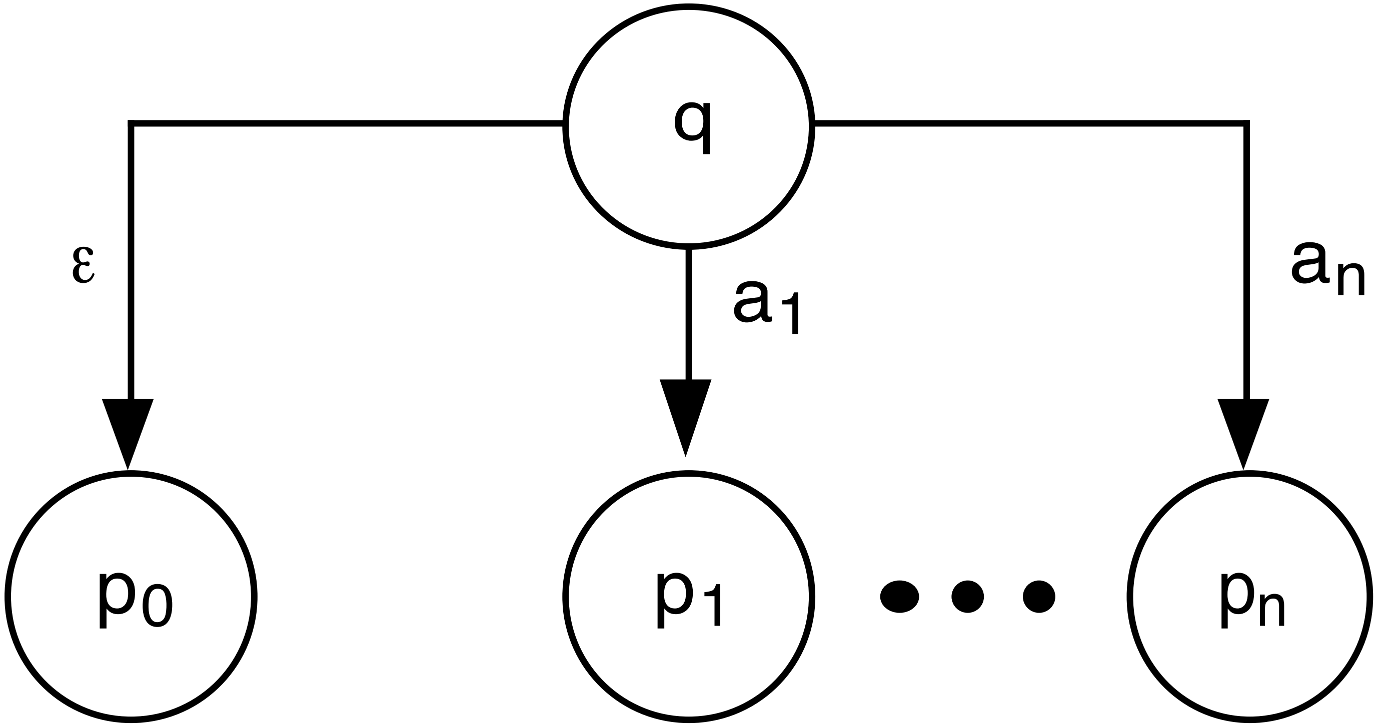
* movimento vazio ou transição vazia

$$\delta(p, \varepsilon) = \{ q_1, q_2, \ldots, q_n \}$$

- q_0 - elemento distinguido de Q: estado inicial
- F - subconjunto de Q: conjunto de estados finais

◆ Autômato como diagrama

$\delta(q, \varepsilon) = \{ p_0 \}$ $\delta(q, a_1) = \{ p_1 \}$ … $\delta(q, a_n) = \{ p_n \}$



◆ Computação de um AFNε

- análoga à de um AFN

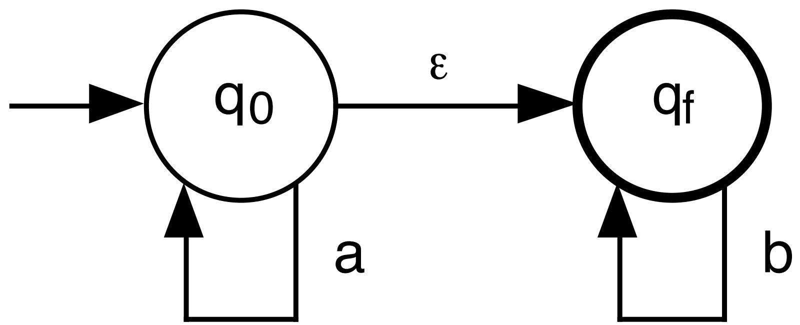
◆ Processamento de uma transição vazia

- não-determinístico
- assume simultaneamente os estados destino e origem
- origem de um movimento vazio: caminho alternativo

Exp: AFNε: a's antecedem b's

$$M_7 = (\{ a, b \}, \{ q_0, q_f \}, \delta_7, q_0, \{ q_f \})$$

δ_7	a	b	ε
q_0	$\{ q_0 \}$	-	$\{ q_f \}$
q_f	-	$\{ q_f \}$	



◆ Antes de definir computação

- computação de transições vazias a partir de
  - ∗ um estado
  - ∗ um conjunto finito de estados

Def: Computação Vazia

$M = (\Sigma, Q, \delta, q_0, F)$

Computação Vazia ou Função Fecho Vazio (*um* estado)

$$\delta\varepsilon: Q \rightarrow 2^Q$$

- indutivamente definida

  * $\delta\varepsilon(q) = \{ q \}$, se $\delta(q, \varepsilon)$ é indefinida
  * $\delta\varepsilon(q) = \{ q \} \cup \delta(q, \varepsilon) \cup (\bigcup_{p \in \delta(q,\varepsilon)} \delta\varepsilon(p))$, caso contrário

Computação Vazia ou Função Fecho Vazio (*conjunto* de estados)

$$\delta\varepsilon^*: 2^Q \rightarrow 2^Q$$

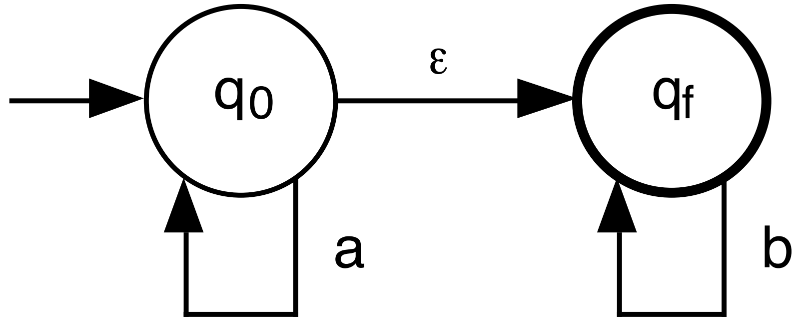
- tal que

$$\delta\varepsilon^*(P) = \bigcup_{q \in P} \delta\varepsilon(q)$$

◆ Por simplicidade, $\delta\varepsilon$ e $\delta\varepsilon^*$

- ambas denotadas por $\delta\varepsilon$

Exp: Computação Vazia



- $\delta\varepsilon(q_0) = \{ q_0, q_f \}$
- $\delta\varepsilon(q_f) = \{ q_f \}$
- $\delta\varepsilon(\{ q_0, q_f \}) = \{ q_0, q_f \}$

◆ Computação de um AFNε para uma entrada w

- sucessiva aplicação da função programa
- para cada símbolo de w (da esquerda para a direita)
- cada passo de aplicação *intercalado* com computações vazias
- até ocorrer uma condição de parada

◆ Assim, antes de processar a próxima transição

- determinar
  - todos os demais estados atingíveis
  - exclusivamente por movimentos vazios

Def: Função Programa Estendida, Computação

$M = (\Sigma, Q, \delta, q_0, F)$ AFNε

$$\delta^*: 2^Q \times \Sigma^* \rightarrow 2^Q$$

indutivamente definida

- $\delta^*(P, \varepsilon) = \delta\varepsilon(P)$
- $\delta^*(P, wa) = \delta\varepsilon(R)$ onde $R = \{ r \mid r \in \delta(s, a)$ e $s \in \delta^*(P, w) \}$

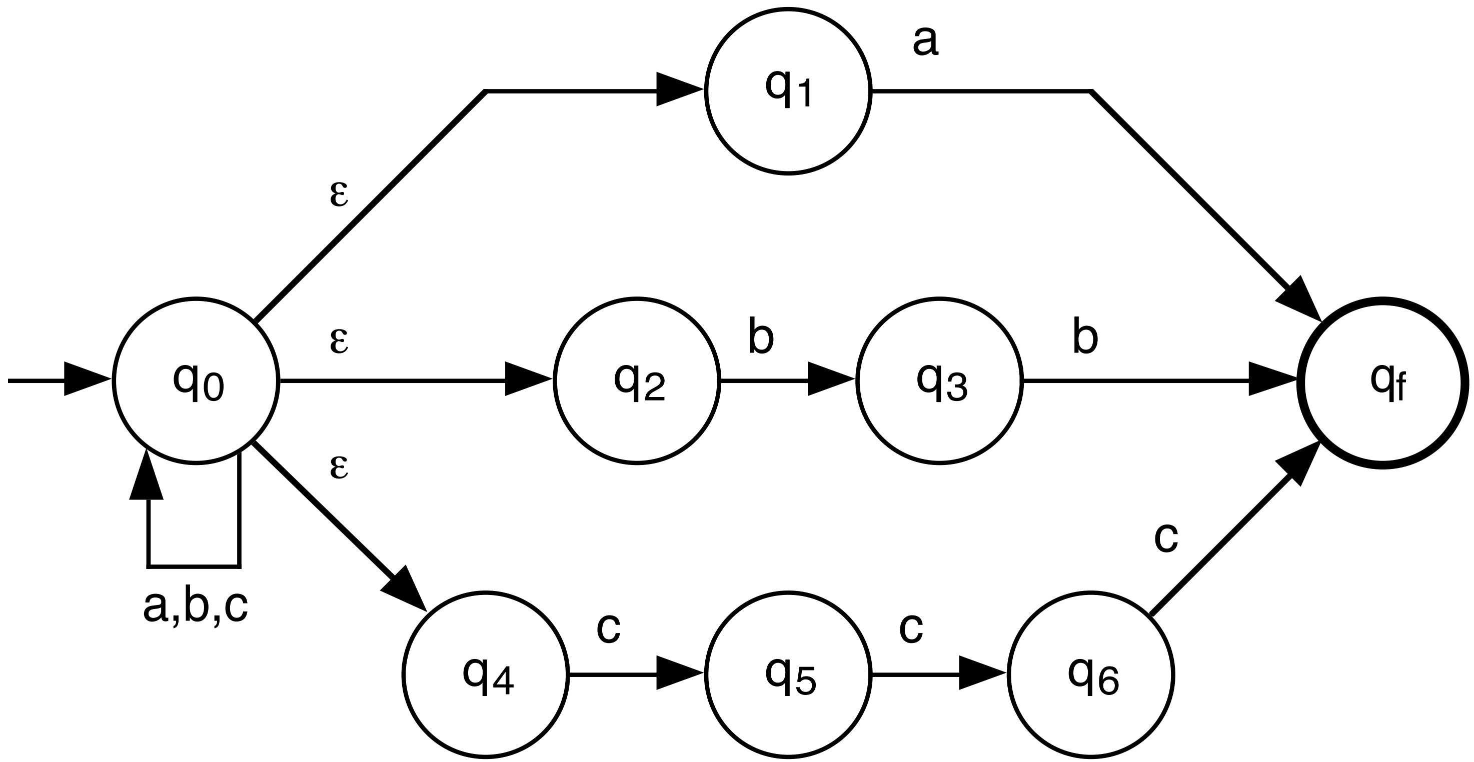
◆ Parada do processamento, Ling. Aceita/Rejeitada

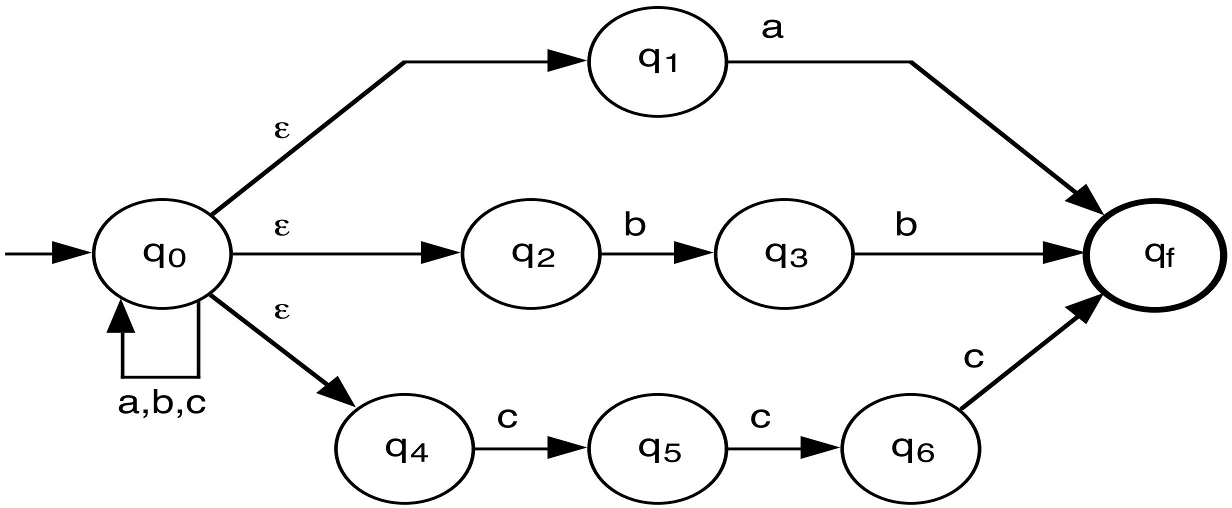
- análoga à do autômato finito não-determinístico

Exp: Computação Vazia, Computação

$L_8 = \{ w \mid w$ possui como sufixo a ou bb ou ccc $\}$

$M_8 = (\{ a, b, c \}, \{ q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_f \}, \delta_8, q_0, \{ q_f \})$





$\delta^*(\{ q_0 \}, abb) = \delta\varepsilon(\{ r \mid r \in \delta(s, b) \text{ e } s \in \delta^*(\{ q_0 \}, ab) \})$ (1)

$\delta^*(\{ q_0 \}, ab) = \delta\varepsilon(\{ r \mid r \in \delta(s, b) \text{ e } s \in \delta^*(\{ q_0 \}, a) \})$ (2)

$\delta^*(\{ q_0 \}, a) = \delta\varepsilon(\{ r \mid r \in \delta(s, a) \text{ e } s \in \delta^*(\{ q_0 \}, \varepsilon) \})$ (3)

Como:

$\delta^*(\{ q_0 \}, \varepsilon) \} = \delta\varepsilon(\{ q_0 \}) = \{ q_0, q_1, q_2, q_4 \}$ considerado em (3)

$\delta^*(\{ q_0 \}, a) = \{ q_0, q_1, q_2, q_4, q_f \}$ considerado em (2)

$\delta^*(\{ q_0 \}, ab) = \{ q_0, q_1, q_2, q_3, q_4 \}$ considerado em (1)

Resulta na computação: $\delta^*(\{ q_0 \}, abb) = \{ q_0, q_1, q_2, q_3, q_4, q_f \}$

Teorema: Equivalência entre AFN e AFNε

Classe dos Autômatos Finitos com Movimentos Vazios é equivalente à Classe dos Autômatos Finitos Não-Determinísticos

Prova: (*por indução*)

Mostrar que

- a partir de um AFNε M qualquer
- construir um AFN M_N que realiza as mesmas computações
- M_N simula M

AFNε → AFN

- construção de uma função programa sem movimentos vazios
- conjunto de estados destino de cada transição não-vazia
  - ampliado com os demais estados possíveis de serem atingidos exclusivamente por transições vazias

$M = (\Sigma, Q, \delta, q_0, F)$ um AFNε qualquer. AFN construído

$$M_N = (\Sigma, Q, \delta_N, q_0, F_N)$$

- $\delta_N: Q \times \Sigma \to 2^Q$ é tal que

$$\delta_N(q, a) = \delta^*(\{ q \}, a)$$

- F_N é o conjunto de todos os estados q pertencentes a Q

$$\delta\varepsilon(q) \cap F \neq \varnothing$$

  - ∗ estados que atingem estados finais via computações vazias

Demonstração que, de fato, o AFN M_N simula o AFNε M

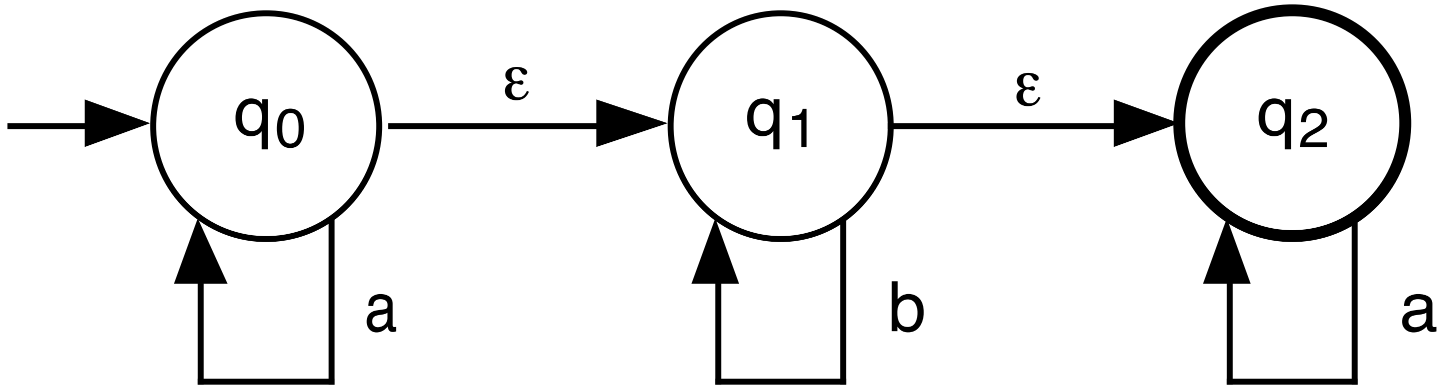
- indução no tamanho da palavra
- exercício

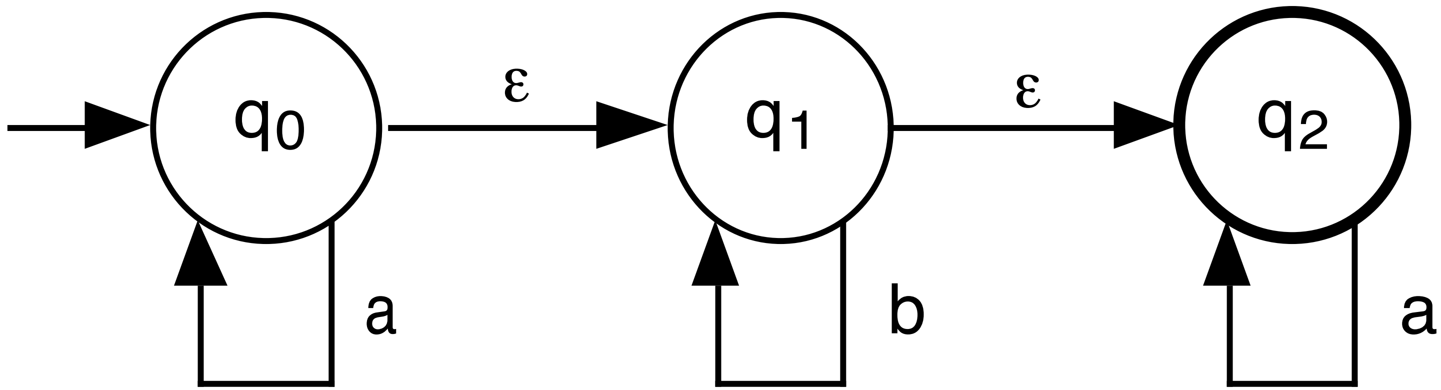
- **Portanto, linguagem aceita por AFNε**
  - é Linguagem Regular ou Tipo 3

Exp: Construção de um AFN a partir de um AFNε

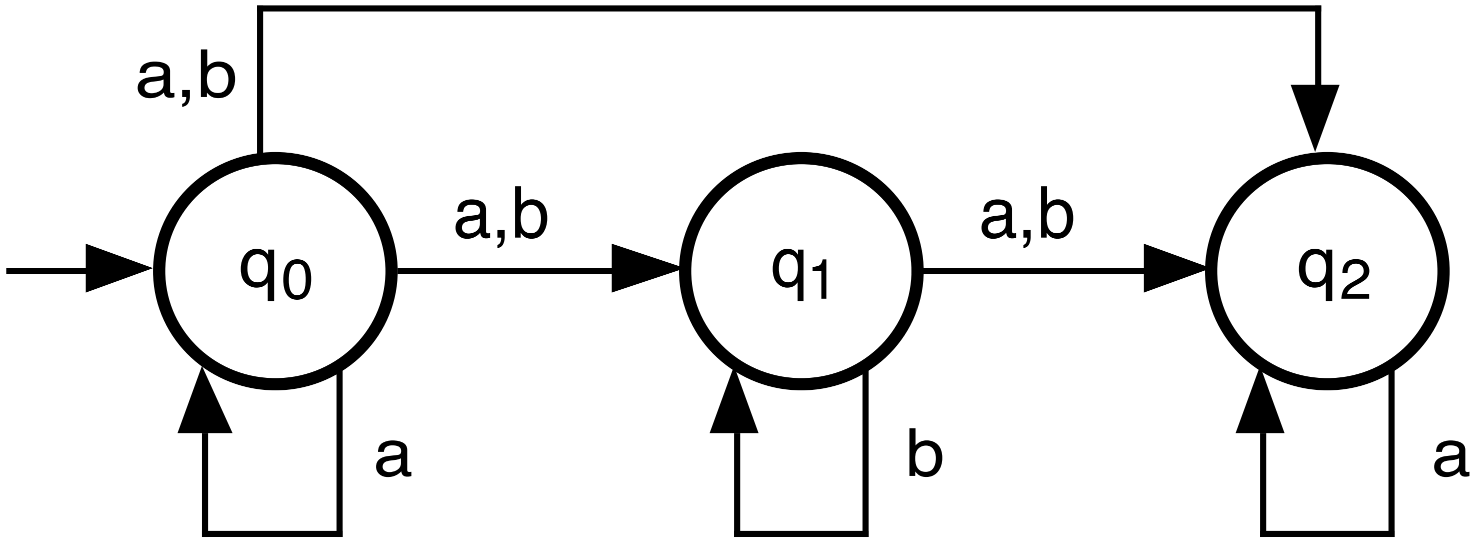
AFNε - $M_9 = (\{ a, b \}, \{ q_0, q_1, q_2 \}, \delta_9, q_0, \{ q_2 \})$

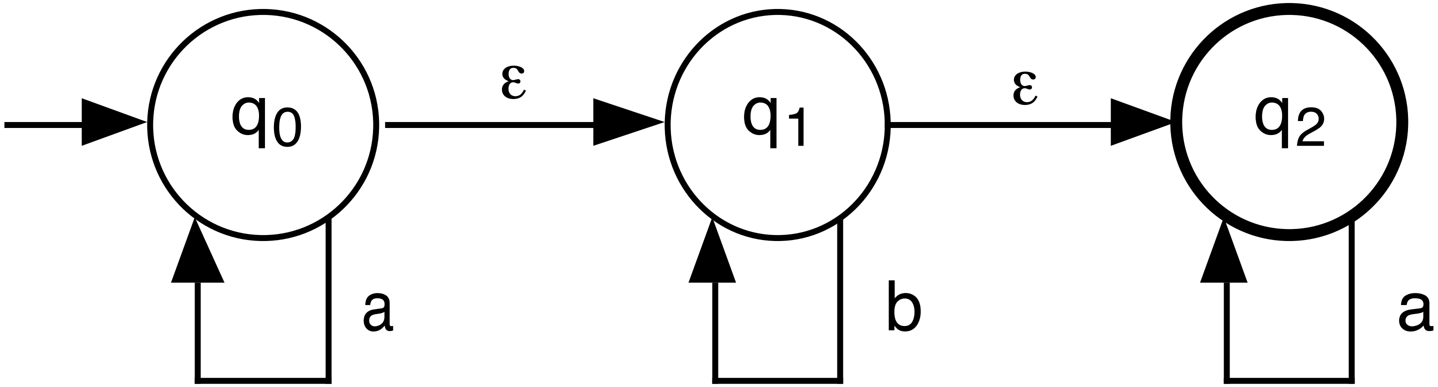
δ_9	a	b	ε
q_0	$\{ q_0 \}$	-	$\{ q_1 \}$
q_1	-	$\{ q_1 \}$	$\{ q_2 \}$
q_2	$\{ q_2 \}$	-	-





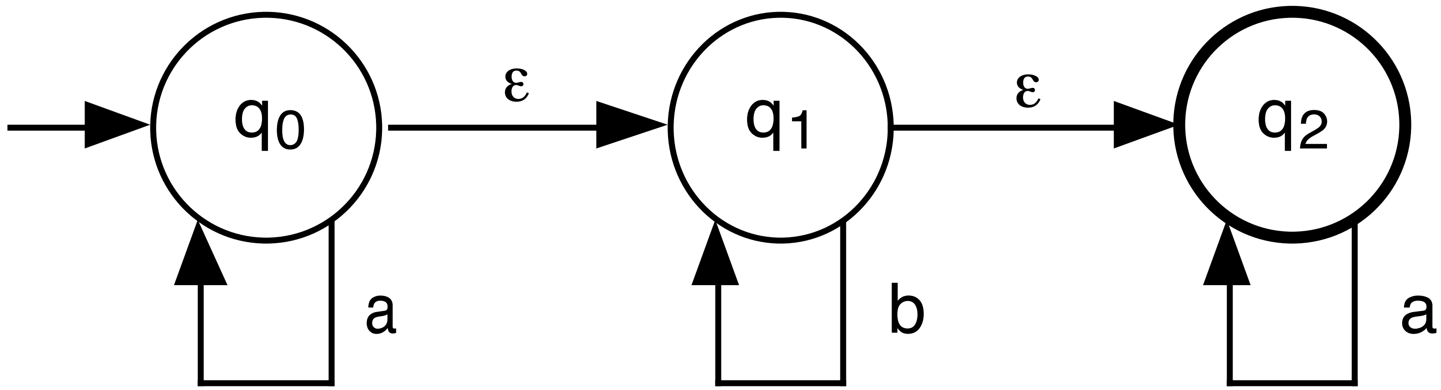
$M_{9_N} = (\{ a, b \}, \{ q_0, q_1, q_2 \}, \delta_{9_N}\, q_0, F_N)$





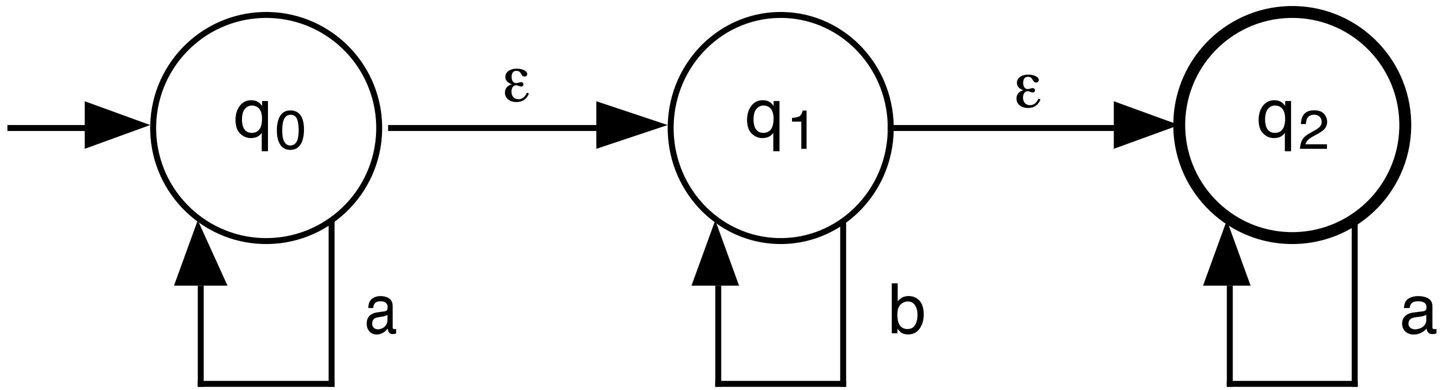
$F_N = \{ q_0, q_1, q_2 \}$

- $\delta\varepsilon(q_0) = \{ q_0, q_1, q_2 \}$
- $\delta\varepsilon(q_1) = \{ q_1, q_2 \}$
- $\delta\varepsilon(q_2) = \{ q_2 \}$



Na construção de δ_{9_N}

- δ_9*$(\{ q_0 \}, \varepsilon) = \{ q_0, q_1, q_2 \}$
- δ_9*$(\{ q_1 \}, \varepsilon) = \{ q_1, q_2 \}$
- δ_9*$(\{ q_2 \}, \varepsilon) = \{ q_2 \}$



Assim, δ_{9_N} é tal que

$$\delta_{9_N}(q_0, a) = \delta_9^*(\{\, q_0 \,\}, a) =$$

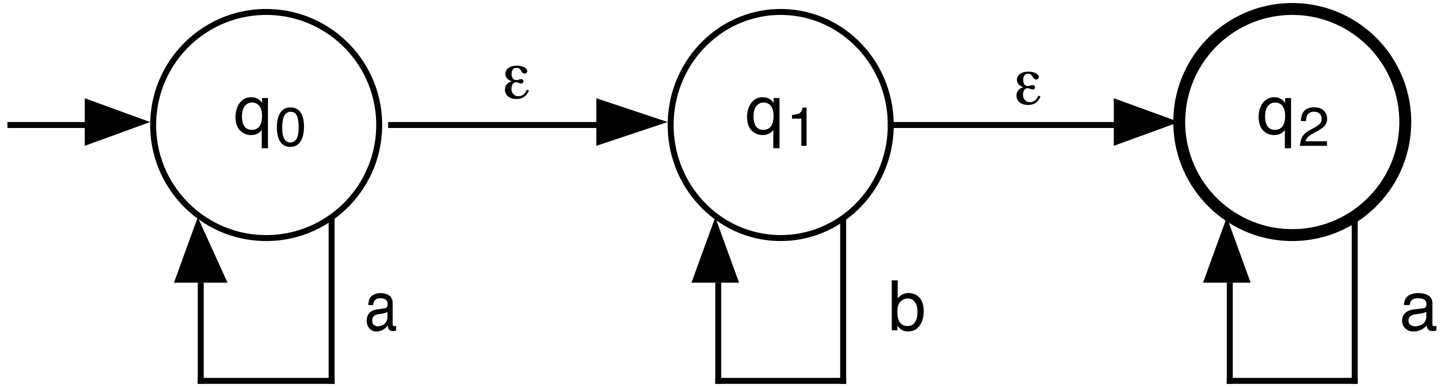
$$\delta\varepsilon(\{\, r \mid r \in \delta(s, a) \text{ e } s \in \underline{\delta}^*(\{\, q_0 \,\}, \varepsilon) \,\}) = \{\, q_0, q_1, q_2 \,\}$$

$$\delta_{9_N}(q_0, b) = \delta_9^*(\{\, q_0 \,\}, b) =$$

$$\delta\varepsilon(\{\, r \mid r \in \delta(s, b) \text{ e } s \in \underline{\delta}^*(\{\, q_0 \,\}, \varepsilon) \,\}) = \{\, q_1, q_2 \,\}$$

$$\delta_{9_N}(q_1, a) = \delta_9^*(\{\, q_1 \,\}, a) =$$

$$\delta\varepsilon(\{\, r \mid r \in \delta(s, a) \text{ e } s \in \underline{\delta}^*(\{\, q_1 \,\}, \varepsilon) \,\}) = \{\, q_2 \,\}$$



- $\delta 9_{N}(q_1, b) = \underline{\delta} 9^*(\{ q_1 \}, b) = \delta\varepsilon(\{ r \mid r \in \delta(s, b)$ e $s \in \underline{\delta}^*(\{ q_1 \}, \varepsilon) \}) = \{ q_1, q_2 \}$

- $\delta 9_{N}(q_2, a) = \underline{\delta} 9^*(\{ q_2 \}, a) = \delta\varepsilon(\{ r \mid r \in \delta(s, a)$ e $s \in \underline{\delta}^*(\{ q_2 \}, \varepsilon) \}) = \{ q_2 \}$

- $\delta 9_{N}(q_2, b) = \underline{\delta} 9^*(\{ q_2 \}, b) = \delta\varepsilon(\{ r \mid r \in \delta(s, b)$ e $s \in \underline{\delta}^*(\{ q_2 \}, \varepsilon) \})$ é indefinida