Teoria de Linguagem

Autômatos Finitos com Movimentos Vazios

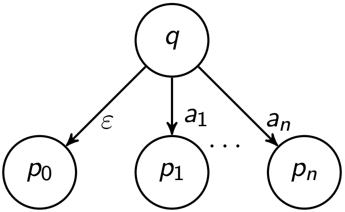
Vinicius H. S. Durelli
✉ durelli@ufsj.edu.br

UFSJ
UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

Organização

Autômatos finitos com movimentos vazios (AFNs$_\varepsilon$)

Movimentos vazios **generalizam** os movimentos não deteminísticos (Menezes 2011). Basicamente, um **movimento vazio é uma transição sem a leitura de um símbolo da fita**.



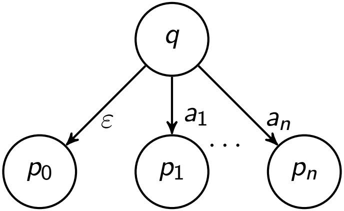
- Transições vazias podem ser interpretadas como um não determinismo interno ao autômato.
- Facilitam algumas construções e demonstrações relacionadas com os autômatos.
- **Não aumentam o poder computacional**.

Diagrama de um AFN$_\varepsilon$

A representação da função programa como um diagrama é análoga à do AFN. Por exemplo, suponha que:

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

O correspondente diagrama é ilustrado como:



➡ **Note**: ao processar uma entrada vazia, o AFN$_\varepsilon$ assume **simultaneamente os estados destino e origem**.

Definição formal

Excetuando-se δ, os componentes Σ, Q, q_0, e F são como na definição do AFN.

Definição → *Autômato Finito com Movimentos Vazios*

Um AFN$_\varepsilon$ $\mathcal{M}$ é uma quíntupla: $\mathcal{M} = (\Sigma, Q, \delta, q_0, F)$ onde:

- Σ representa o alfabeto de símbolos de entrada;
- Q é o conjunto finito de estados do autômato;
- δ função de transição ($\delta : Q \times (\Sigma \cup \varepsilon) \to 2^Q$) – a qual é uma função total;
  - Assim, para um estado p e o símbol especial ε:

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

    é uma transição do AFN$_\varepsilon$.
- $q_0 \in Q$ estado inicial;
- $F \subset Q$ representa o conjunto de estados finais.

❑

Definição formal

Excetuando-se δ, os componentes Σ, Q, q_0, e F são como na definição do AFN.

Definição → *Autômato Finito com Movimentos Vazios*

Um AFN$_\varepsilon$ $\mathcal{M}$ é uma quíntupla: $\mathcal{M} = (\Sigma, Q, \delta, q_0, F)$ onde:

- Σ representa o alfabeto de símbolos de entrada;
- Q é o conjunto finito de estados do autômato;
- δ função de transição ($\delta : Q \times (\Sigma \cup \varepsilon) \rightarrow 2^Q$) – a qual é uma função total;
  - Assim, para um estado p e o símbol especial ε:

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

    é uma transição do AFN$_\varepsilon$.
- $q_0 \in Q$ estado inicial;
- $F \subset Q$ representa o conjunto de estados finais.

❑

Computação vazia. . .

Definição → *Computação Vazia (Função FECHO-ε)*

➥ A computação vazia a partir de **um estado**, denotada por:

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

é definida como segue:

$$\delta_\varepsilon(q) = \{q\} \cup \delta(q, \varepsilon) \cup \left(\bigcup_{p \in \delta(q,\varepsilon)} \delta_\varepsilon(p) \right)$$

(Portanto, $\delta_\varepsilon(q) = \{q\}$, se $\delta(q, \varepsilon) = \emptyset$, ou seja, é indefinida.)

➥ A partir de um **conjunto de estados finito**, denotada por:

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

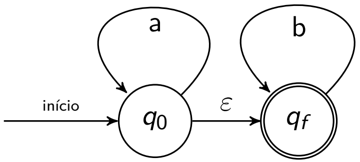
é tal que:

$$\delta^*_\varepsilon(P) : \bigcup_{q \in P} \delta_\varepsilon(q)$$

❑

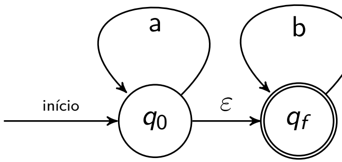
Exemplo (1)

Considere o AFN$_\varepsilon$ $\mathcal{M}_e = (\{a, b\}, \{q_0, q_f\}, \delta, q_0, \{q_f\})$ a seguir:



Exemplo (1)

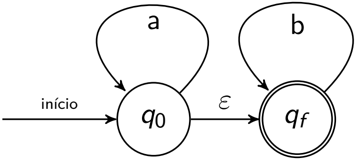
Considere o AFN$_\varepsilon$ $\mathcal{M}_e = (\{a, b\}, \{q_0, q_f\}, \delta, q_0, \{q_f\})$ a seguir:



- Qual a linguagem aceita por $\mathcal{M}_e$?

Exemplo (1)

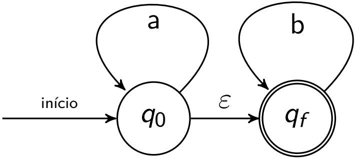
Considere o AFN$_\varepsilon$ $\mathcal{M}_e = (\{a, b\}, \{q_0, q_f\}, \delta, q_0, \{q_f\})$ a seguir:



- Qual a linguagem aceita por $\mathcal{M}_e$?

$$L_e = \{w \mid \text{qualquer } a \text{ antecede qualquer } b\}$$

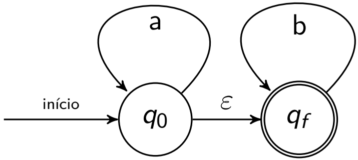
Exemplo (2)



☞ Para entender a computação de um AFN$_\varepsilon$ é necessário primeiro definir a computação resultante de transições vazias (a partir de um estado ou de um conjunto de estados). Computando o FECHO-ε:

- $\delta_\varepsilon(q_0) =$

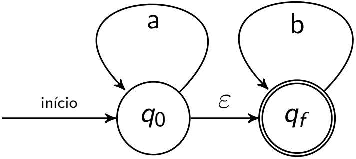
Exemplo (2)



☞ Para entender a computação de um AFN$_\varepsilon$ é necessário primeiro definir a computação resultante de transições vazias (a partir de um estado ou de um conjunto de estados). Computando o FECHO-ε:

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

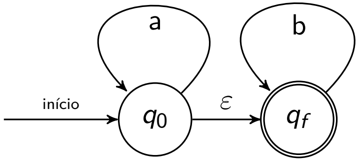
Exemplo (2)



☞ Para entender a computação de um AFN$_\varepsilon$ é necessário primeiro definir a computação resultante de transições vazias (a partir de um estado ou de um conjunto de estados). Computando o FECHO-ε:

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

Exemplo (2)



☞ Para entender a computação de um AFN$_\varepsilon$ é necessário primeiro definir a computação resultante de transições vazias (a partir de um estado ou de um conjunto de estados). Computando o FECHO-ε:

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

Função programa estendida

Definição → *Função Programa Estendida (para AFN$_\varepsilon$s)*

Seja $\mathcal{M} = (\Sigma, Q, \delta, q_0, F)$ um AFN$_\varepsilon$, a função programa de $\mathcal{M}$ denotada por:

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

é a função programa $\delta : Q \times (\Sigma \cup \varepsilon) \rightarrow 2^Q$ estendida para um conjunto finito de estados e para uma palavra e pode ser indutivamente definida como segue:

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

❑

Exercício

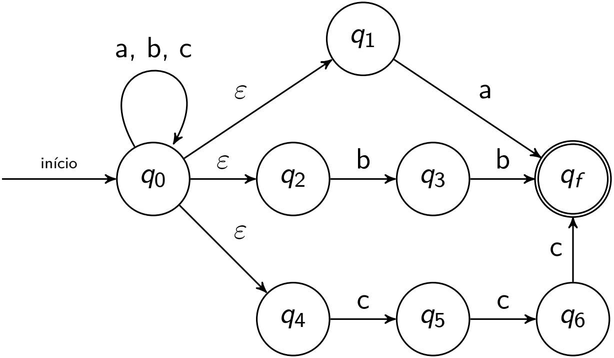
Considerando o alfabeto $\Sigma = \{a, b, c\}$:

Exercício ①**:** Crie um AFN-ε que aceita a linguagem descrita a seguir. $L_{e1} = \{w \ : \ \text{w possui } a, bb \text{ ou } ccc \text{ como sufixo}\}$

Exercício

Considerando o alfabeto $\Sigma = \{a, b, c\}$:

Exercício ①: Crie um AFN-ε que aceita a linguagem descrita a seguir. $L_{e1} = \{w \;:\; \text{w possui } a,\ bb \text{ ou } ccc \text{ como sufixo}\}$



Exemplificando a função programa estendida...

Considerando o AFN$_\varepsilon$ anterior, a computação da palavra *abb* a partir do estado q_0 é como segue:

$$\delta^*(\{q_0\}, abb) = \text{FECHO-}\varepsilon(\{r \mid r \in \delta(s, b) \text{ e } s \in \delta^*(\{q_0\}, ab), \text{ onde:}$$
$$\delta^*(\{q_0\}, ab) = \text{FECHO-}\varepsilon(\{r \mid r \in \delta(s, b) \text{ e } s \in \delta^*(\{q_0\}, a), \text{ onde:}$$
$$\delta^*(\{q_0\}, a) = \text{FECHO-}\varepsilon(\{r \mid r \in \delta(s, a) \text{ e } s \in \delta^*(\{q_0\}, \varepsilon)$$

Como $\delta(\{q_0\}, \varepsilon) = \text{FECHO-}\varepsilon(\{q_0\}) = \{q_0, q_1, q_2, q_4\}$ tem-se que:

$$\delta^*(\{q_0\}, a) = \{q_0, q_1, q_2, q_4, q_f\}$$
$$\delta^*(\{q_0\}, ab) = \{q_0, q_1, q_2, q_3, q_4\}$$
$$\delta^*(\{q_0\}, abb) = \{q_0, q_1, q_2, q_3, q_4, q_f\}$$

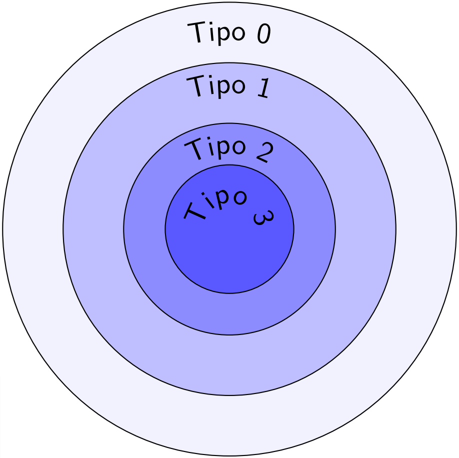
Um AFN$_\varepsilon$ tem mais poder computacional que um AFN?

Conforme mencionado, AFNs$_\varepsilon$ facilitam algumas construções e demonstrações (relacionadas com AFDs e AFNs). Porém...

- Não aumentam o poder computacional (Sipser 2012); ou seja, tais máquinas têm a mesma capacidade de reconhecimento.
- Conforme será mostrado a seguir, **qualquer AFN$_\varepsilon$ pode ser simulado por um AFN**.

Teorema

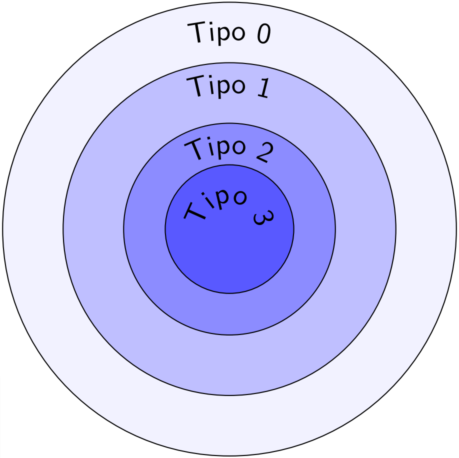
A classes dos AFNs$_\varepsilon$ é equivalente à classe dos AFNs.



Um AFN$_\varepsilon$ tem mais poder computacional que um AFN?

Conforme mencionado, AFNs$_\varepsilon$ facilitam algumas construções e demonstrações (relacionadas com AFDs e AFNs). Porém...

- Não aumentam o poder computacional (Sipser 2012); ou seja, tais máquinas têm a mesma capacidade de reconhecimento.
- Conforme será mostrado a seguir, **qualquer AFN$_\varepsilon$ pode ser simulado por um AFN**.



Teorema

A classes dos AFNs$_\varepsilon$ é equivalente à classe dos AFNs.

Prova (1)

Convertendo um AFN$_\varepsilon$ em um AFN equivalente

A prova consiste em mostrar que a partir de um AFN$_\varepsilon$ $\mathcal{M}_\varepsilon$ é possível construir um AFN $\mathcal{M}_N$.

Ideia central

Construir uma função programa sem movimentos vazios (i.e., δ_N) na qual o conjunto de estados destinos de cada transição não vazia é ampliado com todos os estados possíveis que podem ser atingidos por transições vazias.

✍ Dado um AFN$_\varepsilon$ $\mathcal{M}_\varepsilon = (\Sigma, Q, \delta, q_0, F)$. A meta é construir um AFN $\mathcal{M}_N = (\Sigma, Q_N, \delta_N, q_0, F_N)$ tal que $L(\mathcal{M}_\varepsilon) = L(\mathcal{M}_N)$. A construção de $\mathcal{M}_N$ é como segue:

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

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

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

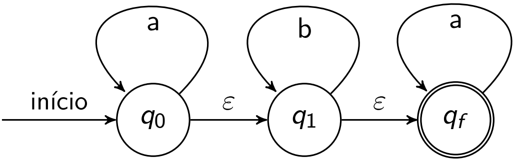
$$\delta_\varepsilon(q) \cap F \neq \emptyset$$

A demonstração que o AFN $\mathcal{M}_N$ simula o AFN$_\varepsilon$ $\mathcal{M}_\varepsilon$ é feita por indução no tamanho da palavra.

Prova (3)

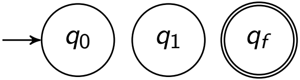
Convertendo um AFN$_\varepsilon$ em um AFN equivalente

Considere o AFN$_\varepsilon$ $\mathcal{M}_\varepsilon = (\{a, b\}, \{q_0, q_1, q_f\}, \delta, q_0, \{q_f\})$ a seguir:



Calculando F_N...

Antes de calcular F_N:



Conforme mencionado, F_N é o conjunto de todos os estados pertencentes a Q, tal que $\delta_\varepsilon(q) \cap F \neq \emptyset$. Então:

$$
\begin{aligned}
F_N &= \{q_0, q_1, q_f\} \text{ pois:}\\
\delta_\varepsilon(q_0) &= \{q_0, q_1, q_f\}\\
\delta_\varepsilon(q_1) &= \{q_1, q_f\}\\
\delta_\varepsilon(q_f) &= \{q_f\}
\end{aligned}
$$

Depois de calcular F_N:

Calculando δ_N...

Seguindo a abordagem proposta anteriormente, δ_N é tal que:

Na construção de δ_N note que:

$$\delta^*(\{q_0\}, \varepsilon) = \{q_0, q_1, q_f\}$$
$$\delta^*(\{q_1\}, \varepsilon) = \{q_1, q_f\}$$
$$\delta^*(\{q_f\}, \varepsilon) = \{q_f\}$$

$$\begin{aligned}\delta_N(q_0, a) = \delta^*(\{q_0\}, a) &= \delta_\varepsilon(\{r \mid r \in \delta(s, a) \text{ e } s \in \delta^*(\{q_0\}, \varepsilon)\}) \\ &= \{q_0, q_1, q_f\}\end{aligned}$$

Calculando δ_N...

Seguindo a abordagem proposta anteriormente, δ_N é tal que:

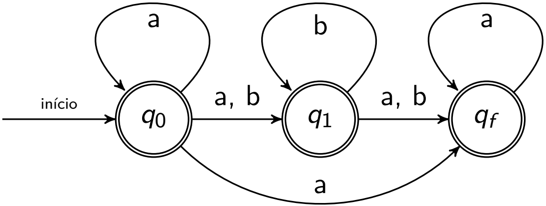
$$
\begin{aligned}
\delta_N(q_0, a) &= \delta^*(\{q_0\}, a) = \delta_\varepsilon(\{r \mid r \in \delta(s, a) \text{ e } s \in \delta^*(\{q_0\}, \varepsilon)\}) \\
&= \{q_0, q_1, q_f\} \\
\delta_N(q_0, b) &= \delta^*(\{q_0\}, b) = \delta_\varepsilon(\{r \mid r \in \delta(s, b) \text{ e } s \in \delta^*(\{q_0\}, \varepsilon)\}) \\
&= \{q_1, q_f\} \\
\delta_N(q_1, a) &= \delta^*(\{q_1\}, a) = \delta_\varepsilon(\{r \mid r \in \delta(s, a) \text{ e } s \in \delta^*(\{q_1\}, \varepsilon)\}) \\
&= \{q_f\} \\
\delta_N(q_1, b) &= \delta^*(\{q_1\}, b) = \delta_\varepsilon(\{r \mid r \in \delta(s, b) \text{ e } s \in \delta^*(\{q_1\}, \varepsilon)\}) \\
&= \{q_1, q_f\} \\
\delta_N(q_f, a) &= \delta^*(\{q_f\}, a) = \delta_\varepsilon(\{r \mid r \in \delta(s, a) \text{ e } s \in \delta^*(\{q_f\}, \varepsilon)\}) \\
&= \{q_f\} \\
\delta_N(q_f, b) &= \delta^*(\{q_f\}, b) = \delta_\varepsilon(\{r \mid r \in \delta(s, b) \text{ e } s \in \delta^*(\{q_f\}, \varepsilon)\}) \\
&= \emptyset
\end{aligned}
$$

AFN resultante...

Tabela de transições:

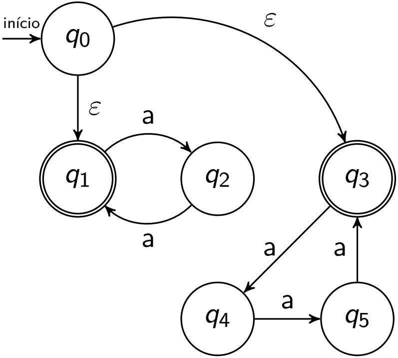
δ_N	a	b
q_0	$\{q_0, q_1, q_f\}$	$\{q_1, q_f\}$
q_1	$\{q_f\}$	$\{q_1, q_f\}$
q_f	$\{q_f\}$	$\emptyset$

Diagrama do AFN:



Exercício (1)

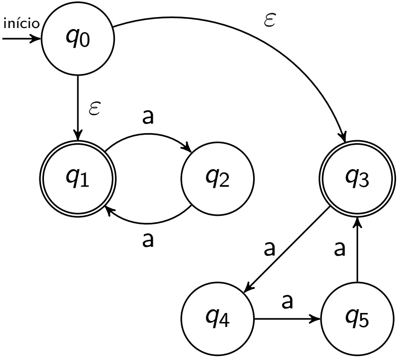
Considere o AFN-ε abaixo:



- Qual a linguagem aceita pela autômato?

Exercício (1)

Considere o AFN-ε abaixo:

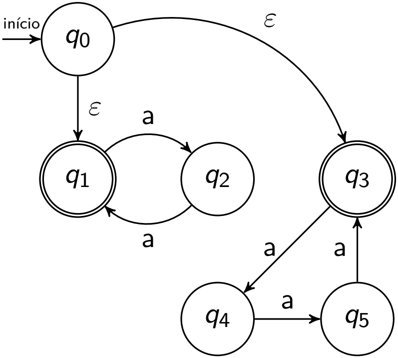


- Qual a linguagem aceita pela autômato?

$$L_e = \{a^n \mid n \text{ é par ou múltiplo de } 3\}$$

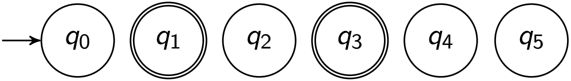
Exercício (2)

Exercício ❷: Converta o AFN-ε abaixo em um AFN.



Solução (1): Calculando F_N...

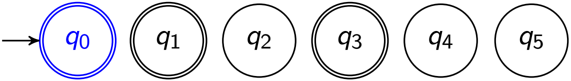
Antes de calcular F_N:



F_N é o conjunto de todos os estados tal que $\delta_\varepsilon(q) \cap F \neq \emptyset$:

$$
\begin{aligned}
F_N &= \{q_0, q_1, q_3\} \text{ pois:}\\
\delta_\varepsilon(q_0) &= \{q_0, q_1, q_3\}\\
\delta_\varepsilon(q_1) &= \{q_1\}\\
\delta_\varepsilon(q_3) &= \{q_3\}
\end{aligned}
$$

Depois de calcular F_N:



Solução (2)

Seguindo a abordagem, δ_N é tal que:

Na construção de δ_N note que:

$$\delta^*(\{q_0\}, \varepsilon) = \{q_0, q_1, q_3\}$$

$$\begin{aligned}\delta_N(q_0, a) &= \delta^*(\{q_0\}, a) = \delta_\varepsilon(\{r \mid r \in \delta(s, a) \text{ e } s \in \delta^*(\{q_0\}, \varepsilon)\}) \\ &= \{q_2, q_4\}\end{aligned}$$

Solução (2)

Seguindo a abordagem, δ_N é tal que:

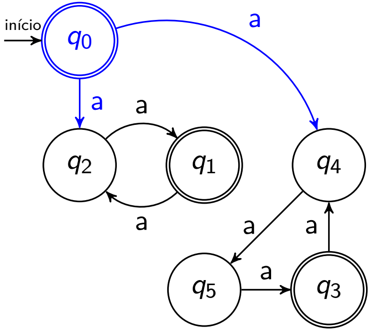
$$\begin{aligned}
\delta_N(q_0, a) &= \delta^*(\{q_0\}, a) = \delta_\varepsilon(\{r \mid r \in \delta(s, a) \text{ e } s \in \delta^*(\{q_0\}, \varepsilon)\}) \\
&= \{q_2, q_4\} \\
&\vdots \\
\delta_N(q_5, a) &= \delta^*(\{q_5\}, a) = \delta_\varepsilon(\{r \mid r \in \delta(s, a) \text{ e } s \in \delta^*(\{q_5\}, \varepsilon)\}) \\
&= \{q_3\}
\end{aligned}$$

Solução (2)

Seguindo a abordagem, δ_N é tal que:

$$\begin{aligned}\delta_N(q_0, a) &= \delta^*(\{q_0\}, a) = \delta_\varepsilon(\{r \mid r \in \delta(s, a) \text{ e } s \in \delta^*(\{q_0\}, \varepsilon)\}) \\ &= \{q_2, q_4\} \\ &\vdots \\ \delta_N(q_5, a) &= \delta^*(\{q_5\}, a) = \delta_\varepsilon(\{r \mid r \in \delta(s, a) \text{ e } s \in \delta^*(\{q_5\}, \varepsilon)\}) \\ &= \{q_3\}\end{aligned}$$

Solução:



1. Autômato finito com movimentos vazios (AFN$_\varepsilon$s)
   - Definição formal
2. Exemplo
3. Função programa estendida (para AFN$_\varepsilon$s)
4. Equivalência entre AFN$_\varepsilon$ e AFN
5. **Considerações finais**

Considerações finais...

Na aula de hoje nós vimos:

- Autômatos finitos com movimentos vazios (AFNs$_\varepsilon$);
  - Equivalência entre AFNs$_\varepsilon$ e AFNs;
- Função de transição estendida (para AFNs$_\epsilon$).

Na **próxima aula**: minimização de autômatos.

Referências

Menezes, Paulo Blauth (2011). *Linguagens Formais e Autômatos*. 6th ed. Livros Didáticos Informática da UFRGS. Bookman, p. 256.

Sipser, Michael (2012). *Introduction to the Theory of Computation*. 3rd ed. Cengage Learning, p. 480.

☺**Próxima aula: exercício(s) sobre o conteúdo da aula de hoje!** ☺