Teoria da Computação (2009/2010)

Notas práticas

Fundamentos:

Σ **Alfabeto** : conjunto finito, não vazio, de **símbolos** (ou **letras**).

Σ^* **Fecho de Kleene** : conjunto de todas as palavras sobre o alfabeto Σ

⇨ Em termos algébricos:

Σ^+ é o **semigrupo** livre gerado por Σ, para a **concatenação** (operação binária associativa)

Σ^* é o **monóide** livre gerado por Σ, para a **concatenação** (operação binária associativa e com elemento **identidade** ε)

O Princípio da Indução Matemática:

Estabelecer a veracidade de uma Proposição P(X) definida sobre os elementos de um conjunto numerável X, em **duas** etapas:

1. **Caso Base**: Provar directamente para um, ou vários, elementos "pequenos" de X;
2. **Passo Indutivo**: Assumindo que P é verdadeira para todos os elementos "menores" que X e, utilizando esse facto, provar P(X).

⇨ **Para provar resultados por Indução Matemática, precisamos estabelecer definições indutivas dos conceitos associados, como por exemplo:**

Definição indutiva de **potência** de ordem n de uma palavra:

1. **Caso Base**:

$$w^0 = \varepsilon \qquad \forall\, w \in \Sigma^*$$

2. **Passo Indutivo:**

$$w^n = ww^{n-1} \qquad \forall\, w \in \Sigma^*, \quad \forall\, n \in \mathbb{N}, \text{ com } n > 0$$

Definição indutiva de **comprimento** de uma palavra:

1. **Caso Base**:

$$|\varepsilon| = 0$$

2. **Passo Indutivo:**

$$|wa| = |w| + 1 \qquad \forall\, w \in \Sigma^*, \forall\, a \in \Sigma$$

• Provar que:

O comprimento é **aditivo** relativamente à concatenação, isto é,

$$\boxed{|vw| = |v| + |w|}$$

Demonstração por **Indução sobre $|w|$** :

1. Caso Base:

Se $|w| = 0$, então $w = \varepsilon$

e portanto $|vw| = |v\varepsilon|$

$= |v|$ porque ε é o elemento identidade da concatenação

$= |v| + 0$ porque 0 é o elemento identidade da soma

$= |v| + |w|$ porque $|w| = 0$

2. Passo Indutivo:

Hipótese de Indução: $|vw| = |v| + |w|, \ \forall\, w : |w| = 1, 2, ..., \mathrm{n}$

Tese: $|vx| = |v| + |x|$, com $x = w\mathrm{a}$

Demonstração:

$|vx| = |vw\mathrm{a}|$ por construção, $x = w\mathrm{a}$

$= |vw| + 1$ pela definição de comprimento

$= |v| + |w| + 1$ por Hipótese de Indução

$= |v| + |x|$ pela construção de x e pela de comprimento.

Está assim provado o Passo Indutivo e, pelo Princípio da Indução, está demonstrada a proposição.

- **Provar que:**

$$\boxed{|w^n| = n\,|w|}\ ,\ \forall\, w \in \Sigma^*\ ,\ \forall\, n \in \mathbb{N}_0$$

Demonstração por **Indução sobre <u>n</u>** :

1. **Caso Base**:

Se $n = 0$, então $w^0 = \varepsilon$ pela definição de potência

e portanto $|w^0| = 0$ pela definição de comprimento

$= 0 \times |w|$ porque 0 é o elemento absorvente do produto

2. **Passo Indutivo:**

Hipótese de Indução: $|w^n| = n \times |w|$
e também $|w^k| = k \times |w|\ ,\ \forall\, k = 1, 2, \ldots, n$

Tese: $|w^{n+1}| = (n+1) \times |w|$

Demonstração:

$|w^{n+1}| = |\, w\, w^n|$ pela definição de potência

$= |w| + |w^n|$ pela propriedade anterior

$= |w| + n \times |w|$ por Hipótese de Indução

$= (n+1) \times |w|$ distributividade da soma

Está assim provado o Passo Indutivo e, pelo Princípio da Indução, está demonstrada a proposição.

Definição indutiva de **igualdade** ($\doteq$) de palavras:

{ Duas palavras são **iguais** quando uma for a cópia, letra a letra, da outra }

1. **Caso Base**:

   Se $|v| = |w| = 0$ então $v \doteq w$.

2. **Passo Indutivo:**

   Sejam $v, w \in \Sigma^*$ tais que $|v| > 0$ e $|w| > 0$, isto é, existem a, b $\in \Sigma$, iguais ou distintos, tais que $v = \mathrm{a}x$ e $w = \mathrm{b}y$.

   Se a = b e se $x \doteq y$ então $v \doteq w$.

{ Por comodidade, escrevemos $v = w$ }

⇨ **Comparar com Algoritmo Recorrente para verificar se duas Listas Lineares são iguais.**

Definição indutiva de palavra **inversa** (Reversa):

$$w^R = \begin{cases} w & \text{se } |w| = 0 \\ \mathrm{a}(v^R) & \text{se } w = v\mathrm{a}, \text{ com } \mathrm{a} \in \Sigma,\ v \in \Sigma^* \end{cases}$$

- **Provar que:**

$$\boxed{(xy)^R = y^R x^R}, \; \forall\, x, y \in \Sigma^*$$

Demonstração por **Indução sobre $|y|$** :

1. **Caso Base**:

   Se $|y| = 0$ então y = ε

   $(xy)^R = (x\varepsilon)^R = x^R = \varepsilon\, x^R = y^R x^R$.

2. **Passo Indutivo:**

   Hipótese de Indução: $\forall\, y \,,\; |y| \leq$ n, $(xy)^R = y^R x^R$.

   Tese: $\forall\, y \,,\; |y| =$ n+1, $(xy)^R = y^R x^R$.

Seja $y = w$a, com $|w|$ = n e a $\in \Sigma$

$(xy)^R$	=	$(x(w\text{a}))^R$	definição de y
	=	$((xw)\text{a})^R$	associatividade da concatenação
	=	$\text{a}(xw)^R$	definição de palavra inversa
	=	$\text{a}(w^R x^R)$	hipótese de indução
	=	$(\text{a}w^R)x^R$	associatividade da concatenação
	=	$(w\text{a})^R x^R$	definição de palavra inversa
	=	$y^R x^R$	definição de y

□

- ## Provar que:

$$(\boldsymbol{w}^{\mathbf{R}})^{\mathbf{R}} = \boldsymbol{w} \quad , \forall\, w \in \Sigma^*$$

$$(\boldsymbol{w}^{\mathbf{R}})^{\mathbf{n}} = (\boldsymbol{w}^{\mathbf{n}})^{\mathbf{R}} \quad , \forall\, w \in \Sigma^*, \forall\, \mathrm{n} \in \mathbb{N}_0$$

$$\boldsymbol{x}^{\mathbf{n}} = \boldsymbol{y}^{\mathbf{n}} \Rightarrow \boldsymbol{x} = \boldsymbol{y} \quad , \forall\, x, y \in \Sigma^*, \forall\, \mathrm{n} \in \mathbb{N}$$

$$\boldsymbol{w} = \boldsymbol{w}^{\mathbf{R}} \Rightarrow \boldsymbol{w}^{\mathbf{n}} = (\boldsymbol{w}^{\mathbf{n}})^{\mathbf{R}} \quad , \forall\, w \in \Sigma^*, \forall\, \mathrm{n} \in \mathbb{N}_0$$

$$\boldsymbol{w}^{\mathbf{n}} = (\boldsymbol{w}^{\mathbf{n}})^{\mathbf{R}} \Rightarrow \boldsymbol{w} = \boldsymbol{w}^{\mathbf{R}} \quad , \forall\, w \in \Sigma^*, \forall\, \mathrm{n} \in \mathbb{N}$$

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

  quantas palavras existem de comprimento 2?
  ... e de comprimento 3?
  ... e de comprimento n?

  Estabeleça uma fórmula e demonstre-a, por Indução.

- E se Σ for um alfabeto com k símbolos?

- **Palíndromos / Capicuas**

{ A palavra **capicua** provém do catalão
capicua (cap + i + cua), que significa: “cabeça e cauda” }
{ A palavra **palíndromo** provém do grego
palíndromos, que significa: “que corre para trás” }

Como definir palíndromo?

$\{ w \in \Sigma^* \mid \exists\, x \in \Sigma^* : \boldsymbol{w = x\, x^{\mathbf{R}}} \}$ define os palíndromos pares

$\{ w \in \Sigma^* \mid \exists\, x \in \Sigma^*, \exists\, \mathrm{a} \in \Sigma : \boldsymbol{w = x\, \mathbf{a}\, x^{\mathbf{R}}} \}$
define os palíndromos ímpares

Para o caso geral, definimos **palíndromo** como:

$\boldsymbol{\{ w \in \Sigma^* \mid w = w^{\mathbf{R}} \}}$

Utilizando a definição anterior,

- **Provar que:**

1. Se $\boldsymbol{w}$ for um palíndromo então $\boldsymbol{w^{\mathbf{n}}}$ é um palíndromo, $\forall\, \mathrm{n} \in \mathbb{N}_0$

2. Se $\boldsymbol{w^{\mathbf{n}}}$ for um palíndromo então $\boldsymbol{w}$ é um palíndromo, $\forall\, \mathrm{n} \in \mathbb{N}$

- **Estabeleça uma Definição Indutiva de Palíndromo.**

Conversão Binário $\mapsto$ Decimal :

Consideremos o alfabeto binário $\Sigma = \{0, 1\}$.
O conjunto das palavras de Σ^+ pode ser interpretado como a **Representação Binária** dos números naturais, como por exemplo:

$$1001 \mapsto 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9$$

$$11001010 \mapsto 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^3 + 1 \times 2^1 = 202$$

Definição: $\forall\, w \in \{0, 1\}^+$

$$w = b_k \dots b_1\, b_0 \mapsto \sum_{i=0}^{k} b_i\, 2^i = n \in \mathbb{N}_0$$

Teorema: (Conversão **Binário $\mapsto$ Decimal da esquerda para a direita**)

1. **Caso Base**:

   $\mathbf{0} \in \{0, 1\} \mapsto \mathbf{0} \in \mathbb{N}_0$

   $\mathbf{1} \in \{0, 1\} \mapsto \mathbf{1} \in \mathbb{N}_0$

2. **Passo Indutivo:**

   Se $w \in \{0, 1\}^+ \mapsto n \in \mathbb{N}_0$ então $\boldsymbol{w0 \mapsto 2 \times n}$

   $\boldsymbol{w1 \mapsto 2 \times n + 1}$

Exemplo: 11001 $\mapsto$ 25

1	1	= 1
1	$2 \times 1 + 1$	= 3
0	2×3	= 6
0	2×6	= 12
1	$2 \times 12 + 1$	= 25

Demonstração:

1. **Caso Base**:

$$\mathbf{0} \in \{0, 1\} \mapsto 0 \times 2^0 = \mathbf{0}$$
$$\mathbf{1} \in \{0, 1\} \mapsto 1 \times 2^0 = \mathbf{1}$$

2. **Passo Indutivo:**

Se $\quad w = b_k \ldots b_1 b_0 \mapsto \sum_{i=0}^{k} b_i 2^i = n$

então $\boldsymbol{w0} = b_k \ldots b_1 b_0 0 \mapsto \sum_{i=0}^{k} b_i 2^{i+1} + 0 \times 2^0 =$

$$2 \times \sum_{i=0}^{k} b_i 2^i = \mathbf{2 \times n}$$

$\boldsymbol{w1} = b_k \ldots b_1 b_0 1 \mapsto \sum_{i=0}^{k} b_i 2^{i+1} + 1 \times 2^0 =$

$$2 \times \sum_{i=0}^{k} b_i 2^i + 1 = \mathbf{2 \times n + 1}$$

- **Para cada uma das Linguagens seguintes, construa o respectivo Autómato Finito Determinista.**
  Defina cada AFD na forma $A = (Q, \Sigma, \delta, q_0, F)$ **e estabeleça os correspondentes Diagramas e Tabelas de Transições:**

1. $\{ w \in \{0, 1\}^* \mid w$ é a representação binária de um número **par** $\}$

2. $\{ w \in \{0, 1\}^* \mid w$ é a representação binária de um número **par** e começa por 1 $\}$

3. $\{ w \in \{0, 1\}^* \mid w$ é a representação binária de um número **par** sem 0's à esquerda $\}$

4. $\{ w \in \{0, 1\}^* \mid w$ é a representação binária de um múltiplo de 4 $\}$

5. $\{ w \in \{0, 1\}^* \mid w$ é a representação binária de 2^n, com $n \geq 0$, e começa por 1 $\}$

6. $\{ w \in \{0, 1\}^* \mid w$ é a representação binária de um múltiplo de 3 $\}$

7. $\{ w \in \{0, 1\}^* \mid w$ representa um múltiplo de 5 e começa por 1 $\}$

8. $\{ w \in \{0, 1\}^* \mid w^R$ representa um múltiplo de 5 $\}$

9. $\{ w \in \{a, b\}^* \mid w$ começa por a $\}$

10. $\{ w \in \{a, b\}^* \mid w$ não começa por a $\}$

11. $\{ w \in \{a, b\}^* \mid w$ começa por ab $\}$

12. $\{ w \in \{0, 1\}^* \mid 001$ é parte de $w \}$

13. $\{ w \in \{0, 1\}^* \mid w$ não contém $001\}$

14. $\{ w \in \{a, b\}^* \mid w = a^n$ com n ímpar $\}$

15. $\{ w \in \{a, b\}^* \mid w = a^n b \text{ com } n \geq 0 \}$

16. $\{ w \in \{a, b\}^* \mid w = a^m b^n \text{ com } m, n \geq 0 \}$

17. $\{ w \in \{a, b\}^* \mid w = a^m b^n \text{ onde } m, n \geq 0 \text{ são números pares} \}$

18. $L = \{ awa \mid \forall\, w \in \{a, b\}^* \}$

19. $L^2 = \{ axaaya \mid \forall\, x, y \in \{a, b\}^* \}$

20. $\{ w \mid \forall\, w \in \{a, b\}^* \}$

21. $\{ w \in \{a, b\}^* \mid |w| \text{ é um número par} \}$

22. $\{ w \in \{a, b\}^* \mid |w| \textbf{ mod } 3 = 0 \}$

23. $\{ w \in \{a, b\}^* \mid n_a(w) \text{ é um número ímpar} \}$

24. $\{ w \in \{a, b\}^* \mid n_a(w) \textbf{ mod } 3 = 0 \}$

25. $\{ w \in \{a, b\}^* \mid n_a(w) \textbf{ mod } 3 > n_b(w) \textbf{ mod } 3 \}$

26. $\{ w \in \{a, b\}^* \mid w \text{ começa e acaba com a mesma letra} \}$

27. $\{ w \in \{0, 1\}^* \mid w \text{ não tem 0's consecutivos} \}$

28. $\{ w \in \{a, b\}^* \mid w \text{ onde nenhuma letra aparece repetida 3 vezes consecutivas} \}$

29. $\{ (10)^n \mid n \geq 0 \} \cup \{ (01)^n \mid n \geq 0 \}$

30. $\{ w \in \{0, 1\}^* \mid w \text{ tem igual número de (10)'s e de (01)'s} \}$

31. $\{ w \in \{0, 1\}^* \mid \text{se } w \text{ começa por 0 não tem 0's repetidos e se } w \text{ começa por 1 não tem 1's repetidos} \}$

⇨ Sobre AFD's reconhecedores de Linguagens Complementares:

Como a **Linguagem** de um AFD, $A = (Q, \Sigma, \delta, q_0, F)$

é o conjunto das palavras **reconhecidas** por A,

$$L(A) = \{ w \in \Sigma^* \mid \delta^{\wedge}(q_0, w) \in F \}$$

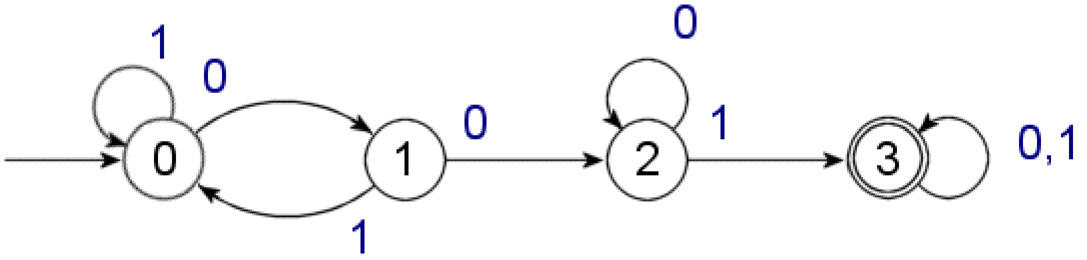
então, o conjunto das palavras **não reconhecidas** por A é,

$$\overline{L(A)} = \Sigma^* \setminus L(A) = \{ w \in \Sigma^* \mid \delta^{\wedge}(q_0, w) \notin \mathbf{F} \}$$

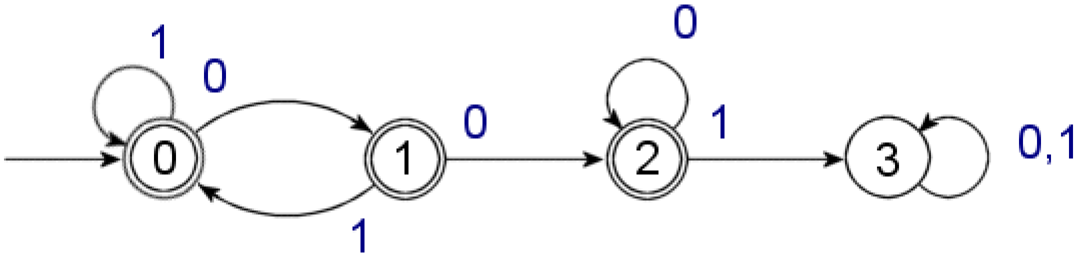
$$= \{ w \in \Sigma^* \mid \delta^{\wedge}(q_0, w) \in \mathbf{Q} \setminus \mathbf{F} \}$$

Como por exemplo:

$\{ w \in \{0, 1\}^* \mid 001$ é parte de $w \}$



$\{ w \in \{0, 1\}^* \mid w$ não contém $001\}$



⇨ A extensão $\delta^{\wedge}$ e o reconhecimento de palavras:

Definição: $\delta^{\wedge} : Q \times \Sigma^{*} \longrightarrow Q$

$$\begin{cases} \delta^{\wedge}(q, \varepsilon) = q \\ \delta^{\wedge}(q, w) = \delta(\delta^{\wedge}(q, x), a), \ \forall\, w = xa \end{cases}$$

- **Provar que:**

$$\boxed{\boldsymbol{\delta^{\wedge}(q, xy) = \delta^{\wedge}(\delta^{\wedge}(q, x), y)}}\,, \ \forall\, q \in Q, \ \forall\, x,y \in \Sigma^{*}$$

(Demonstrar por Indução sobre |y|)

- **Seja** $A = (Q, \Sigma, \delta, q_0, F)$ **um AFD onde** $\exists\, \mathbf{q} \in \mathbf{Q}$ **tal que,**
  $\forall\, a \in \Sigma$, $\boldsymbol{\delta(q, a) = q}$.
  Mostrar que $\forall\, w \in \Sigma^{*}$, $\boldsymbol{\delta^{\wedge}(q, w) = q}$.

- **Seja** $A = (Q, \Sigma, \delta, q_0, F)$ **um AFD onde** $\exists\, \mathbf{a} \in \Sigma$ **tal que,**
  $\forall\, q \in Q$, $\boldsymbol{\delta(q, a) = q}$.

  a) Mostrar que $\forall\, n \geq 0$, $\boldsymbol{\delta^{\wedge}(q, a^{n}) = q}$.

  b) Estabelecer uma condição suficiente para que $\{a\}^{*} \subseteq L(A)$.

- **Seja** A = (Q, Σ, δ, q_0, {q_f}) **tal que,** $\forall$ a $\in$ Σ, $\boldsymbol{\delta(q_0, a) = \delta(q_f, a)}$.

  a) **Mostrar que** $\forall\, w \neq \varepsilon$, $\hat{\delta}(q_0, w) = \hat{\delta}(q_f, w)$.

  b) **Mostrar que se** $x \neq \varepsilon$, $x \in L(A)$ **então** $\forall\, k > 0$, $x^k \in L(A)$.

- **Seja** A = (Q, {a, b}, δ, q_0, F) **um AFD.**

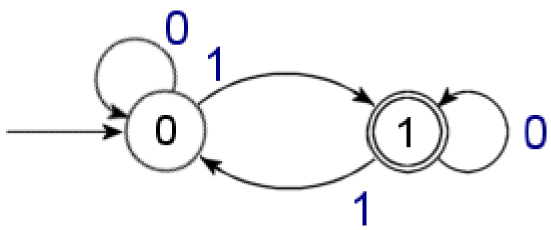
  a) **Demonstrar por Indução que:**
  se $\exists\, q \in Q$ **tal que** $\hat{\delta}(q, aa) = q$
  então também $\hat{\delta}(q, a^n) = q$ **para qualquer** $n \geq 0$ **par.**

  b) **Estabelecer condições sobre esse** $q \in Q$
  para que $\{a^n \mid n \geq 0 \text{ par}\} \subseteq L(A)$.

  c) **Estabelecer condições para que**
  $L(A) = \{a^m b^n \mid m, n \geq 0 \text{ pares}\}$.

- **Seja** A = ({q_0, q_1}, {0, 1}, δ, q_0, {q_1})

  com



  provar que $L(A) = \{ w \in \{0, 1\}^* \mid n_1(w) \bmod 2 = 1 \}$.

		0	1
$\rightarrow$	q_0	q_0	q_1
*	q_1	q_1	q_0

Por definição, $L(A) = \{ w \in \{0, 1\}^* \mid \hat{\delta}(q_0, w) = q_1 \}$

O reconhecimento das palavras com um número **ímpar** de 1's termina no estado $\mathbf{q_1}$ (**aceitação**) e o reconhecimento das palavras com um número **par** de 1's termina no estado $\mathbf{q_0}$ (**não aceitação**).

Basta portanto provar que:

$\forall\, w \in \{0, 1\}^*$, $\hat{\delta}(q_0, w) = q_1 \iff n_1(w) \bmod 2 = 1$

(o que é equivalente a, $\hat{\delta}(q_0, w) = q_0 \iff n_1(w) \bmod 2 = 0$)

Demonstração por Indução sobre $|w|$:

1. **Caso Base**:

   Se $|w| = 0$ então $w = \varepsilon$.

   Nesse caso $\hat{\delta}(q_0, \varepsilon) = q_0$ e $n_1(\varepsilon) = 0 = 0 \bmod 2$.

2. **Passo Indutivo:**

   Hipótese de Indução: $\hat{\delta}(q_0, x) = q_1 \iff n_1(x) \bmod 2 = 1$,

   para todo o $x : |x| < |w|$.

   Tese de Indução: $\hat{\delta}(q_0, w) = q_1 \iff n_1(w) \bmod 2 = 1$,

   para $w = x\text{a}$ com $\text{a} \in \{0, 1\}$.

Demonstração:

<u>Se</u> **a = 0** : então $\mathbf{n_1(w) = n_1(x)}$, ou seja, $n_1(w)$ **mod** $2 = n_1(x)$ **mod** 2.

<u>Se</u> $n_1(x)$ **mod** $2 = 0 \Leftrightarrow \delta^{\wedge}(q_0, x) = q_0$, por Hip. de Indução.

Nesse caso também,

$\delta^{\wedge}(q_0, w) = \delta^{\wedge}(q_0, x0) = \delta(\delta^{\wedge}(q_0, x), 0) = \delta(q_0, 0) = q_0$.

<u>Se</u> $n_1(x)$ **mod** $2 = 1 \Leftrightarrow \delta^{\wedge}(q_0, x) = q_1$, por Hip. de Indução.

Nesse caso também,

$\delta^{\wedge}(q_0, w) = \delta^{\wedge}(q_0, x0) = \delta(\delta^{\wedge}(q_0, x), 0) = \delta(q_1, 0) = q_1$.

Portanto $\delta^{\wedge}(q_0, w) = q_1 \Leftrightarrow n_1(w)$ **mod** $2 = 1$.

<u>Se</u> **a = 1** : então $\mathbf{n_1(w) = n_1(x) + 1}$.

<u>Se</u> $n_1(x)$ **mod** $2 = 0 \Leftrightarrow \delta^{\wedge}(q_0, x) = q_0$, por Hip. de Indução.

Nesse caso, $n_1(w)$ **mod** $2 = (n_1(x)+1)$ **mod** $2 = 1$, e

$\delta^{\wedge}(q_0, w) = \delta^{\wedge}(q_0, x1) = \delta(\delta^{\wedge}(q_0, x), 1) = \delta(q_0, 1) = q_1$.

<u>Se</u> $n_1(x)$ **mod** $2 = 1 \Leftrightarrow \delta^{\wedge}(q_0, x) = q_1$, por Hip. de Indução.

Nesse caso, $n_1(w)$ **mod** $2 = (n_1(x)+1)$ **mod** $2 = 0$, e

$\delta^{\wedge}(q_0, w) = \delta^{\wedge}(q_0, x1) = \delta(\delta^{\wedge}(q_0, x), 1) = \delta(q_1, 1) = q_0$.

Portanto $\delta^{\wedge}(q_0, w) = q_1 \Leftrightarrow n_1(w)$ **mod** $2 = 1$.

⇨ **Nota:** **Na maior parte dos casos, sendo dados um Autómato A e uma Linguagem L, para provar que L = L(A), é necessário provar que:** $L \subseteq L(A)$
$L \supseteq L(A)$

- **Seja** $L_0 = \{ (10)^n \mid \forall\, n \geq 0 \}$

  a) **Construa um AFD para reconhecer L_0 e chame-lhe A_0.**

  b) **Demonstre que $L_0 = L(A_0)$.**

- **Seja** $L_1 = \{ w \in \{0, 1\}^* \mid w \text{ começa e acaba em } 0 \}$

  a) **Construa um AFD para reconhecer L_1 e chame-lhe A_1.**

  b) **Demonstre que $L_1 = L(A_1)$.**

- **Sejam,**

  $A_2 = (\{p, q\}, \{0, 1\}, \{ \delta(p, 1) = \delta(q, 0) = q \}, p, \{q\})$

  $L_2 = \{ 10^n \mid \forall\, n \geq 0 \}$

  Demonstre que $L_2 = L(A_2)$.

Autómatos Finitos Não-Deterministas com Transições-ε (AFND-ε)

- **Para cada uma das Linguagens seguintes, construa um Autómato Não-Determinista, possivelmente com Transições-ε. (Tente obter o AFND-ε mais simples que conseguir)**

1. $\{ w \in \{a, b\}^* \mid w = a^m b^n \text{ com } m, n \geq 0 \}$

2. $\{ w \in \{a, b, c\}^* \mid w = a^m b^n c^p \text{ com } m, n, p \geq 0 \}$

3. $\{ w \in \{a, b\}^* \mid w = a^m b^n \text{ onde } m, n \geq 0 \text{ são números pares} \}$

4. $\{ awa \mid \forall\, w \in \{a, b\}^* \}$

5. $\{ w \in \{0, 1\}^* \mid w$ é a representação binária de um número **par** $\}$

6. $\{ w \in \{0, 1\}^* \mid w$ acaba em 01 $\}$

7. $\{ w \in \{0, 1\}^* \mid w$ acaba em 0 ou em 11 $\}$

8. $\{ w \in \{0, 1\}^* \mid w$ acaba em 00 ou em 11 $\}$

9. $\{ w \in \{0, 1\}^* \mid w$ começa por 010 **ou** acaba em 110$\}$

10. $\{ w \in \{0, 1\}^* \mid w$ começa por 010 **e** acaba em 110$\}$

11. $\{ w \in \{0, 1\}^* \mid w$ tem, pelo menos, duas ocorrências de 01 **e** acaba em 11$\}$

12. $\{ab, abc\}^*$ (só com 3 estados)

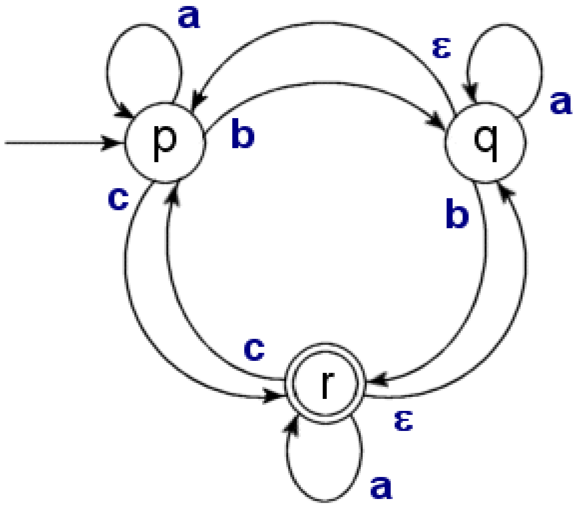
13. …

⇨ A eliminação das transições-ε por construção dos fechos-ε:

Exemplo: Um AFND-ε <u>sem</u> transições múltiplas,
A = {{p, q, r}, {a, b, c}, δ, p, {r}}

onde,

	δ	ε	a	b	c
⟶	p	∅	{p}	{q}	{r}
	q	{p}	{q}	{r}	∅
*	r	{q}	{r}	∅	{p}



fecho-ε(p) = {p}
fecho-ε(q) = {p, q}
fecho-ε(r) = {p, q, r}

Construção de um AFD equivalente: D = {Q_D, {a, b, c}, δ_D, q_D, F_D}

- **Os estados de D são constituídos pelos fechos-ε dos estados de A,**

  Q_D = {{p}, {p, q}, {p, q, r}}

- **Construção da função de transição δ_D :**

$\delta_D(\{p\}, a) = \text{fecho-}\varepsilon(\delta(\{p\}, a)) = \text{fecho-}\varepsilon(\{p\}) = \{p\}$
$\delta_D(\{p\}, b) = \text{fecho-}\varepsilon(\delta(\{p\}, b)) = \text{fecho-}\varepsilon(\{q\}) = \{p, q\}$
$\delta_D(\{p\}, c) = \text{fecho-}\varepsilon(\delta(\{p\}, c)) = \text{fecho-}\varepsilon(\{r\}) = \{p, q, r\}$

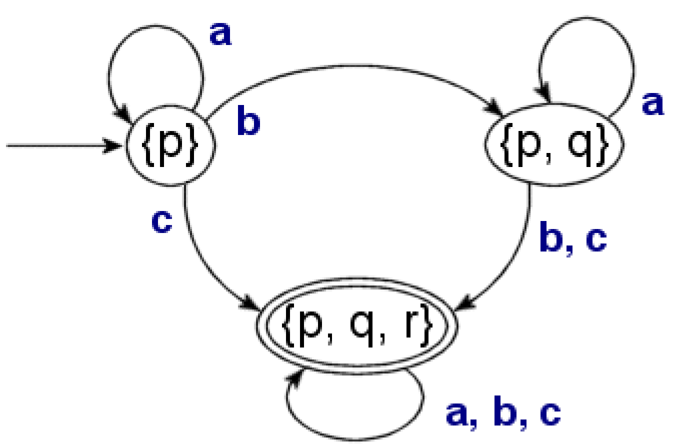
$\delta_D(\{p, q\}, a) = \text{fecho-}\varepsilon(\{p\}) \cup \text{fecho-}\varepsilon(\{q\}) = \{p\} \cup \{p, q\} = \{p, q\}$
$\delta_D(\{p, q\}, b) = \text{fecho-}\varepsilon(\{q\}) \cup \text{fecho-}\varepsilon(\{r\}) = \{p, q\} \cup \{p, q, r\} = \{p, q, r\}$
$\delta_D(\{p, q\}, c) = \text{fecho-}\varepsilon(\{r\}) \cup \emptyset = \{p, q, r\}$

$\delta_D(\{p, q, r\}, a) = \text{fecho-}\varepsilon(\{p\}) \cup \text{fecho-}\varepsilon(\{q\}) \cup \text{fecho-}\varepsilon(\{r\}) = \{p, q, r\}$
$\delta_D(\{p, q, r\}, b) = \text{fecho-}\varepsilon(\{q\}) \cup \text{fecho-}\varepsilon(\{r\}) \cup \emptyset = \{p, q, r\}$
$\delta_D(\{p, q, r\}, c) = \text{fecho-}\varepsilon(\{r\}) \cup \emptyset \cup \text{fecho-}\varepsilon(\{p\}) = \{p, q, r\}$

- **Estado Inicial** $q_D = \text{fecho-}\varepsilon(\{p\}) = \{p\}$
- **Estados Finais** $F_D = \{\{p, q, r\}\}$

Donde,

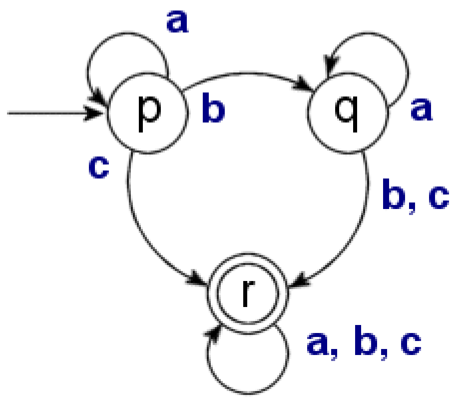
	δ_D	a	b	c
$\longrightarrow$	$\{p\}$	$\{p\}$	$\{p, q\}$	$\{p, q, r\}$
	$\{p, q\}$	$\{p, q\}$	$\{p, q, r\}$	$\{p, q, r\}$
$*$	$\{p, q, r\}$	$\{p, q, r\}$	$\{p, q, r\}$	$\{p, q, r\}$



Finalmente, fazendo $p \leftarrow \{p\}$
$q \leftarrow \{p, q\}$
$r \leftarrow \{p, q, r\}$ obtemos o AFD pretendido:

$D = \{\{p, q, r\}, \{a, b, c\}, \delta_D, p, \{r\}\}$, com

	δ_D	a	b	c
$\longrightarrow$	p	p	q	r
	q	q	r	r
*	r	r	r	r



Conversão AFND-ε ↦ AFD:

Entrada: $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$

Saída: $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$

Funções auxiliares:

$\text{vizinhos}(\ p \in Q_E,\ a \in \Sigma\ ) = \{\ q \in Q_E \mid \exists\ \delta_E(p, a)\ \}$

$\text{vizinhançaND}(\ S \subseteq Q_E,\ a \in \Sigma\ ) = \text{fecho-}\varepsilon(\ \bigcup_{s \in S} \text{vizinhos}(s, a)\ )$

O Algoritmo:

$q_D \leftarrow \text{fecho-}\varepsilon(\{q_0\})$

$Q_D \leftarrow Q_D \cup \{q_D\}$

para cada estado $q \in Q_D$ ainda não visitado

  visitar q

  para cada símbolo $a \in \Sigma$

    $S \leftarrow \text{vizinhançaND}(\{q\}, a)$

    $Q_D \leftarrow Q_D \cup S$

    inserir transição $\delta_D(q, a) = S$

para cada estado $q \in Q_D$

  se $q \in F_E$

  então $F_D \leftarrow F_D \cup \{q\}$

- **Para cada um dos AFND-ε's seguintes:**
  - **desenhe o respectivo Diagrama de Transições;**
  - **construa os conjuntos-potência e o consequente Diagrama de Transições;**
  - **calcule o fecho-ε de cada conjunto;**
  - **identifique os estados não acessíveis, o Estado Inicial e os Estados Finais;**
  - **construa o AFD equivalente;**
  - **verifique a necessidade de inclusão de "estados-ratoeira";**

- **Utilize o Algoritmo de Conversão AFND-ε ↦ AFD.**

1.

	δ	ε	0	1
$\longrightarrow *$	p	{r}	$\emptyset$	{q}
	q	$\emptyset$	{p, r}	{r}
	r	$\emptyset$	$\emptyset$	$\emptyset$

2.

	δ	ε	a	b	c
$\longrightarrow$	p	{q, r}	$\emptyset$	{q}	{r}
	q	$\emptyset$	{p}	{r}	{p, q}
$*$	r	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

3.

	δ	ε	a
$\longrightarrow$	p	$\emptyset$	{q}
$*$	q	{r}	$\emptyset$
	r	{p}	$\emptyset$

4.

	δ	ε	0	1
$\longrightarrow$	p	{q}	{q}	$\emptyset$
$*$	q	$\emptyset$	{p, r}	{q, r}
	r	$\emptyset$	{r}	{q}

5.

	δ	ε	a	b	c
$\longrightarrow *$	p	$\emptyset$	{q}	$\emptyset$	$\emptyset$
	q	$\emptyset$	$\emptyset$	{r}	$\emptyset$
	r	{p}	$\emptyset$	$\emptyset$	{p}

6.

	δ	0	1
$\longrightarrow$	p	{p, q}	{p}
	q	{r, s}	{t}
	r	{p, r}	{t}
$*$	s	$\emptyset$	$\emptyset$
$*$	t	$\emptyset$	$\emptyset$

7.

	δ	ε	0	1
$\longrightarrow$	p	{q}	$\emptyset$	{p, q}
	q	$\emptyset$	{r}	{q, r}
$*$	r	{q}	{r}	$\emptyset$