MAC 414
Autômatos, Computabilidade e Complexidade
aula 7 — 5/10/2020

Lema do Bombeamento

Lema

Se L é uma linguagem regular, então existe um inteiro N tal que, se $x \in L$ então para toda fatoração $x = yzw \in L$ com $|z| = N$, existe uma fatoração $z = tuv$, com $u \neq \lambda$ tal que para todo $n \geq 0$, $ytu^n vw \in L$ (ou seja, $ytu^* vw \subseteq L$).

Lema do Bombeamento

Lema

Se L é uma linguagem regular, então existe um inteiro N tal que, se $x \in L$ então para toda fatoração $x = yzw \in L$ com $|z| = N$, existe uma fatoração $z = tuv$, com $u \neq \lambda$ tal que para todo $n \geq 0$, $ytu^n vw \in L$ (ou seja, $ytu^* vw \subseteq L$).

Lema do Bombeamento

Lema

Se L é uma linguagem regular, então existe um inteiro N tal que, se $x \in L$ então para toda fatoração $x = yzw \in L$ com $|z| = N$, existe uma fatoração $z = tuv$, com $u \neq \lambda$ tal que para todo $n \geq 0$, $ytu^n vw \in L$ (ou seja, $ytu^* vw \subseteq L$).

Lema

Se L é uma linguagem regular, então existe um inteiro N tal que, se $x \in L$, existe uma fatoração $x = tuv$, com $|tu| \leq N$, tal que $u \neq \lambda$ e para todo $n \geq 0$, $tu^n v \in L$.

Para que serve?

Para que serve?

Provar que uma linguagem é regular:

Para que serve?

Provar que uma linguagem é regular: NUNCA!

Para que serve?

Provar que uma linguagem é regular: NUNCA!

Provar que uma linguagem não é regular.

1. AnBn

Para que serve?

Provar que uma linguagem é regular: NUNCA!

Provar que uma linguagem não é regular.

1. AnBn
2. $\{x \in (a+b)^* \mid |x|_a > 1000|x|_b\}$

Para que serve?

Provar que uma linguagem é regular: NUNCA!

Provar que uma linguagem não é regular.

1. AnBn
2. $\{x \in (a+b)^* \mid |x|_a > 1000|x|_b\}$
3. $\{x \in a^* \mid |x| \text{ é quadrado perfeito}\}$

Para que serve?

Provar que uma linguagem é regular: NUNCA!

Provar que uma linguagem não é regular.

1. AnBn
2. $\{x \in (a+b)^* \mid |x|_a > 1000|x|_b\}$
3. $\{x \in a^* \mid |x| \text{ é quadrado perfeito}\}$
4. $\{x : |x|_a = |x|_b\} + (a+b)^*(aaa+bbb)(a+b)^*$

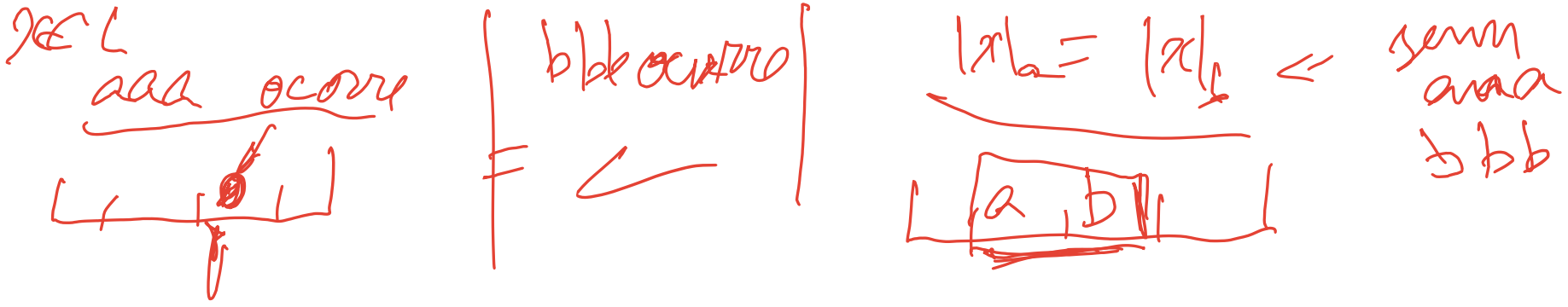
Um caso diferente

$$\{x : |x|_a = |x|_b\} + (a+b)^*(aaa+bbb)(a+b)^*$$

Um caso diferente

$$L = \{x : |x|_a = |x|_b\} + (a+b)^*(aaa+bbb)(a+b)^*$$

Satisfaz o Lema do Bombeamento com $N = 4$:



Um caso diferente

$$\{x : |x|_a = |x|_b\} + (a+b)^*(aaa+bbb)(a+b)^*$$

Satisfaz o Lema do Bombeamento com $N = 4$:

Se L é regular, então para toda linguagem regular R, $L \cap R$ é regular.

Um caso diferente

$$\{x : |x|_a = |x|_b\} + (a+b)^*(aaa+bbb)(a+b)^*$$

Satisfaz o Lema do Bombeamento com $N = 4$:

Se L é regular, então para toda linguagem regular R, $L \cap R$ é regular. Qual R?

Um caso diferente

$$\{x : |x|_a = |x|_b\} + (a+b)^*(aaa+bbb)(a+b)^*$$

Satisfaz o Lema do Bombeamento com $N = 4$:

Se L é regular, então para toda linguagem regular R, $L \cap R$ é regular. Qual R?

$R = (aab)^*(abb)^*$

$L \cap R = \{(aab)^n (abb)^n \mid n \geq 0\}$

Minimização de autômatos

Minimização de autômatos

Objetivo:

- Mostrar que para cada linguagem regular existe um AD *especialmente* mínimo.

Minimização de autômatos

Objetivo:

- Mostrar que para cada linguagem regular existe um AD *especialmente* mínimo.
- Esse autômato é único, a menos de isomorfismo.

Minimização de autômatos

Objetivo:

- Mostrar que para cada linguagem regular existe um AD *especialmente* mínimo.
- Esse autômato é único, a menos de isomorfismo.
- Mostrar um algoritmo polinomial que computa o autômato mínimo a partir de um AD dado.

Minimização de autômatos

Objetivo:

- Mostrar que para cada linguagem regular existe um AD *especialmente* mínimo.
- Esse autômato é único, a menos de isomorfismo.
- Mostrar um algoritmo polinomial que computa o autômato mínimo a partir de um AD dado.

Minimização de autômatos

Objetivo:

- Mostrar que para cada linguagem regular existe um AD *especialmente* mínimo.
- Esse autômato é único, a menos de isomorfismo.
- Mostrar um algoritmo polinomial que computa o autômato mínimo a partir de um AD dado.

O que é especial é a teoria algébrica, que **não** funciona para AND.

Relações de equivalência

Relações de equivalência

Só para lembrar:

Relações de equivalência

Só para lembrar:

A conjunto, uma relação binária $\sim$ é um subconjunto de $A \times A$; escrevemos $x \sim y$ para denotar $(x, y) \in \sim$.

Relações de equivalência

Só para lembrar:

A conjunto, uma relação binária $\sim$ é um subconjunto de $A \times A$; escrevemos $x \sim y$ para denotar $(x, y) \in \sim$.

Definição mais útil

$\sim$ é relação de equivalência se existe alguma função definida em A tal que
$x \sim y$ sse x e y têm a mesma imagem.

Relações de equivalência

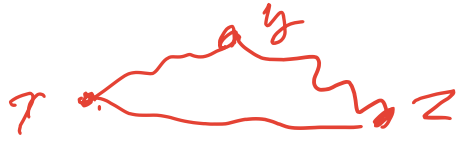
Só para lembrar:

A conjunto, uma relação binária $\sim$ é um subconjunto de $A \times A$; escrevemos $x \sim y$ para denotar $(x, y) \in \sim$.

Definição mais útil

$\sim$ é relação de equivalência se existe alguma função definida em A tal que
$x \sim y$ sse x e y têm a mesma imagem.

Definição interna: $\sim$ é relação de equivalência se for reflexiva, simétrica e transitiva.



Quociente

Quociente

Classe de equivalência: $[x] = \{y \in A \mid y \sim x\}$.

Quociente

Classe de equivalência: $[x] = \{y \in A \mid y \sim x\}$.
Conjunto quociente: $A/\sim = \{[x] \mid x \in A\}$.

Quociente

Classe de equivalência: $[x] = \{y \in A \mid y \sim x\}$.
Conjunto quociente: $A/\sim = \{[x] \mid x \in A\}$.
Índice de $\sim$: $|A/\sim|$, o número de classes.

Quociente

Classe de equivalência: $[x] = \{y \in A \mid y \sim x\}$.

Conjunto quociente: $A/\sim = \{[x] \mid x \in A\}$.

Índice de $\sim$: $|A/\sim|$, o número de classes.

A projeção canônica $\pi : A \to A/\sim$, definida por
$\pi(x) = [x]$ tem a propriedade $x \sim y$ sse $\pi(x) = \pi(y)$.

Quociente

Classe de equivalência: $[x] = \{y \in A \mid y \sim x\}$.
Conjunto quociente: $A/\sim = \{[x] \mid x \in A\}$.
Índice de $\sim$: $|A/\sim|$, o número de classes.
A projeção canônica $\pi : A \to A/\sim$, definida por
$\pi(x) = [x]$ tem a propriedade $x \sim y$ sse $\pi(x) = \pi(y)$.
Como definir uma função sobre o quociente

- Para dar a imagem de uma classe, escolhe-se um elemento da classe, e define-se a imagem a partir dele.

Quociente

Classe de equivalência: $[x] = \{y \in A \mid y \sim x\}$.
Conjunto quociente: $A/\sim = \{[x] \mid x \in A\}$.
Índice de $\sim$: $|A/\sim|$, o número de classes.
A projeção canônica $\pi : A \to A/\sim$, definida por
$\pi(x) = [x]$ tem a propriedade $x \sim y$ sse $\pi(x) = \pi(y)$.
Como definir uma função sobre o quociente

- Para dar a imagem de uma classe, escolhe-se um elemento da classe, e define-se a imagem a partir dele.
- Mostra-se que o resultado não depende de qual elemento foi escolhido antes. (jargão: a função foi bem-definida).

Congruências

Congruências

Em Álgebra, o termo congruência se refere a uma relação de equivalência que preserva as operações.

Congruências

Em Álgebra, o termo congruência se refere a uma relação de equivalência que preserva as operações.

Ex: Em $\mathbb{Z}$, temos $x \sim y \mod n$ se $n|x-y$.
Respeitar as operações: se $x \sim y \mod n$, $z \sim w \mod n$ então $x \circ z \sim y \circ w \mod n$,
onde $\circ$ pode ser $+$ ou $\times$.

Congruências

Em Álgebra, o termo congruência se refere a uma relação de equivalência que preserva as operações.

Ex: Em $\mathbb{Z}$, temos $x \sim y \mod n$ se $n|x-y$.
Respeitar as operações: se $x \sim y \mod n$, $z \sim w \mod n$ então $x \circ z \sim y \circ w \mod n$,
onde $\circ$ pode ser $+$ ou $\times$.

Ex: Se V é um espaço vetorial e H é um subespaço, define-se em V: $x \sim y \mod H$ se $x - y \in H$. As operações que são preservadas são soma de vetores e multiplicação por escalar.

Congruências em semiautômato

Congruências em semiautômato

Num semiautômato já usamos a notação qx para denotar o resultado da operação $\delta : K \times \Sigma^* \to K$. Podemos pensar em multiplicação à direita por uma palavra como análogo à multiplicação por escalar.

Congruências em semiautômato

Num semiautômato já usamos a notação qx para denotar o resultado da operação $\delta : K \times \Sigma^* \to K$. Podemos pensar em multiplicação à direita por uma palavra como análogo à multiplicação por escalar.

ATENÇÃO: *vamos aceitar temporariamente autômatos e semiautômatos com conjunto infinito de estados.*

Congruências em semiautômato

Num semiautômato já usamos a notação qx para denotar o resultado da operação $\delta : K \times \Sigma^* \to K$. Podemos pensar em multiplicação à direita por uma palavra como análogo à multiplicação por escalar.

ATENÇÃO: *vamos aceitar temporariamente autômatos e semiautômatos com conjunto infinito de estados.*

Uma congruência num semiatômato $\mathscr{A} = (K, \Sigma, \delta, s)$ é uma relação de equivalência $\sim$ em K tal que para toda palavra $x \in \Sigma^*$, $p \sim q \Rightarrow px \sim qx$.

Congruências em semiautômato

Num semiautômato já usamos a notação qx para denotar o resultado da operação $\delta : K \times \Sigma^* \to K$. Podemos pensar em multiplicação à direita por uma palavra como análogo à multiplicação por escalar.

ATENÇÃO: *vamos aceitar temporariamente autômatos e semiautômatos com conjunto infinito de estados.*

Uma congruência num semiatômato $\mathscr{A} = (K, \Sigma, \delta, s)$ é uma relação de equivalência $\sim$ em K tal que para toda palavra $x \in \Sigma^*$, $p \sim q \Rightarrow px \sim qx$.

Prop: Uma relação de equivalência $\sim$ em K é uma congruência sse para toda $\sigma \in \Sigma$, $p \sim q \Rightarrow p\sigma \sim q\sigma$.

O semiautômato quociente

O semiautômato quociente

Dados $\mathscr{A}$ e $\sim$, podemos construir

$$\mathscr{A}/{\sim} = \left(K/{\sim}, \Sigma, \bar{\delta}, [s]\right)$$

onde $\bar{\delta}([q], \sigma) = [\delta(q, \sigma)]$, ou seja, $[q]\sigma = [q\sigma]$.

O semiautômato quociente

Dados $\mathscr{A}$ e $\sim$, podemos construir

$$\mathscr{A}/\!\sim = \left(K/\!\sim, \Sigma, \bar{\delta}, [s]\right)$$

onde $\bar{\delta}([q], \sigma) = [\delta(q, \sigma)]$, ou seja, $[q]\sigma = [q\sigma]$.

Vale, pelo método usual, que para toda $x \in \Sigma^*$,
$[q]x = [qx]$.

Homomorfismo

Homomorfismo

Um homomorfismo $\varphi : (K, \Sigma, \delta, s) \to (K', \Sigma, \delta', s')$ é uma função $\varphi : K \to K'$ tal que $\varphi(s) = s'$ e para todos q,x, $\varphi(qx) = \varphi(q)x$.

Homomorfismo

Um homomorfismo $\varphi : (K, \Sigma, \delta, s) \to (K', \Sigma, \delta', s')$ é uma função $\varphi : K \to K'$ tal que $\varphi(s) = s'$ e para todos q,x, $\varphi(qx) = \varphi(q)x$.

Algumas propriedades:

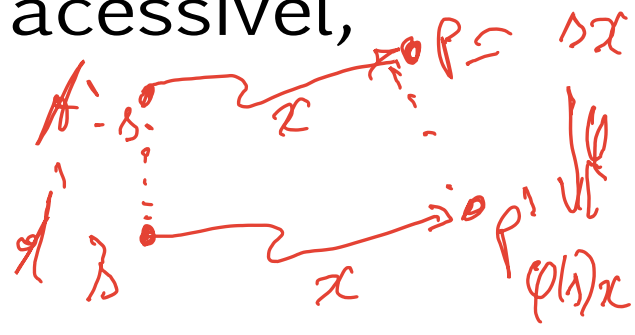
1. Se $\varphi : K \to K'$ é tal que $\varphi(s) = s'$ e para todos q,σ, $\varphi(q\sigma) = \varphi(q)\sigma$, φ é homomorfismo.

Homomorfismo

Um homomorfismo $\varphi : (K, \Sigma, \delta, s) \to (K', \Sigma, \delta', s')$ é uma função $\varphi : K \to K'$ tal que $\varphi(s) = s'$ e para todos q,x, $\varphi(qx) = \varphi(q)x$.

Algumas propriedades:

1. Se $\varphi : K \to K'$ é tal que $\varphi(s) = s'$ e para todos q,σ, $\varphi(q\sigma) = \varphi(q)\sigma$, φ é homomorfismo.
2. Se φ é homomorfismo e $\mathscr{A}'$ é acessível, então φ é sobrejetor.



Homomorfismo

Um homomorfismo $\varphi : (K, \Sigma, \delta, s) \to (K', \Sigma, \delta', s')$ é uma função $\varphi : K \to K'$ tal que $\varphi(s) = s'$ e para todos q,x, $\varphi(qx) = \varphi(q)x$.

Algumas propriedades:

1. Se $\varphi : K \to K'$ é tal que $\varphi(s) = s'$ e para todos q,σ, $\varphi(q\sigma) = \varphi(q)\sigma$, φ é homomorfismo.
2. Se φ é homomorfismo e $\mathscr{A}'$ é acessível, então φ é sobrejetor.
3. Se $\mathscr{A}$ é acessível, existe no máximo um homomorfismo $\mathscr{A} \to \mathscr{A}'$.

Homomorfismo

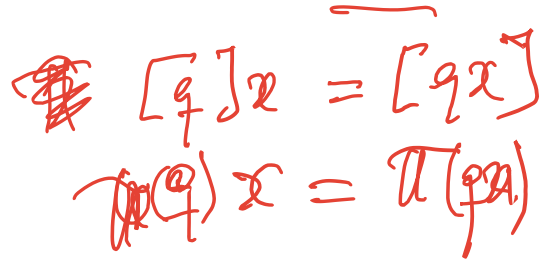
Um homomorfismo $\varphi : (K, \Sigma, \delta, s) \to (K', \Sigma, \delta', s')$ é uma função $\varphi : K \to K'$ tal que $\varphi(s) = s'$ e para todos q,x, $\varphi(qx) = \varphi(q)x$.

Algumas propriedades:

1. Se $\varphi : K \to K'$ é tal que $\varphi(s) = s'$ e para todos q,σ, $\varphi(q\sigma) = \varphi(q)\sigma$, φ é homomorfismo.
2. Se φ é homomorfismo e $\mathscr{A}'$ é acessível, então φ é sobrejetor.
3. Se $\mathscr{A}$ é acessível, existe no máximo um homomorfismo $\mathscr{A} \to \mathscr{A}'$.
4. Composição de homomorfismos é homomorfismo.

Quocientes, de novo

1. Se $\sim$ é uma congruência em $\mathscr{A}$, então π e um homomorfismo $\mathscr{A} \to \mathscr{A}/\sim$.



Quocientes, de novo

1. Se $\sim$ é uma congruência em $\mathscr{A}$, então π e um homomorfismo $\mathscr{A} \to \mathscr{A}/\sim$.
2. Suponha que $\mathscr{A}$ e $\mathscr{A}'$ são acessíveis e $\varphi : \mathscr{A} \to \mathscr{A}'$ é um homomorfismo. Seja $\sim$ a relação de equivalência dada por φ. Então $\mathscr{A}/\sim$ é isomorfo a $\mathscr{A}'$.

$[x] \longleftrightarrow \varphi(x)$

Quocientes, de novo

1. Se $\sim$ é uma congruência em $\mathscr{A}$, então π e um homomorfismo $\mathscr{A} \to \mathscr{A}/\sim$.
2. Suponha que $\mathscr{A}$ e $\mathscr{A}'$ são acessíveis e $\varphi : \mathscr{A} \to \mathscr{A}$ é um homomorfismo. Seja $\sim$ a relação de equivalência dada por φ.
   Então $\mathscr{A}/\sim$ é isomorfo a $\mathscr{A}'$.

Quocientes, de novo

1. Se $\sim$ é uma congruência em $\mathscr{A}$, então π e um homomorfismo $\mathscr{A} \to \mathscr{A}/\!\sim$.
2. Suponha que $\mathscr{A}$ e $\mathscr{A}'$ são acessíveis e $\varphi : \mathscr{A} \to \mathscr{A}$ é um homomorfismo. Seja $\sim$ a relação de equivalência dada por φ. Então $\mathscr{A}/\!\sim$ é isomorfo a $\mathscr{A}'$.

Daqui para a frente, quando falarmos de autômatos e semi-autômatos, vamos supor *acessível* implicitamente.

Congruência em autômatos

Congruência em autômatos

Uma congruência num AD é uma congruência do respectivo semiautômato que preserva a "operação" adicional $p \in F$.

Congruência em autômatos

Uma congruência num AD é uma congruência do respectivo semiautômato que preserva a "operação" adicional $p \in F$.

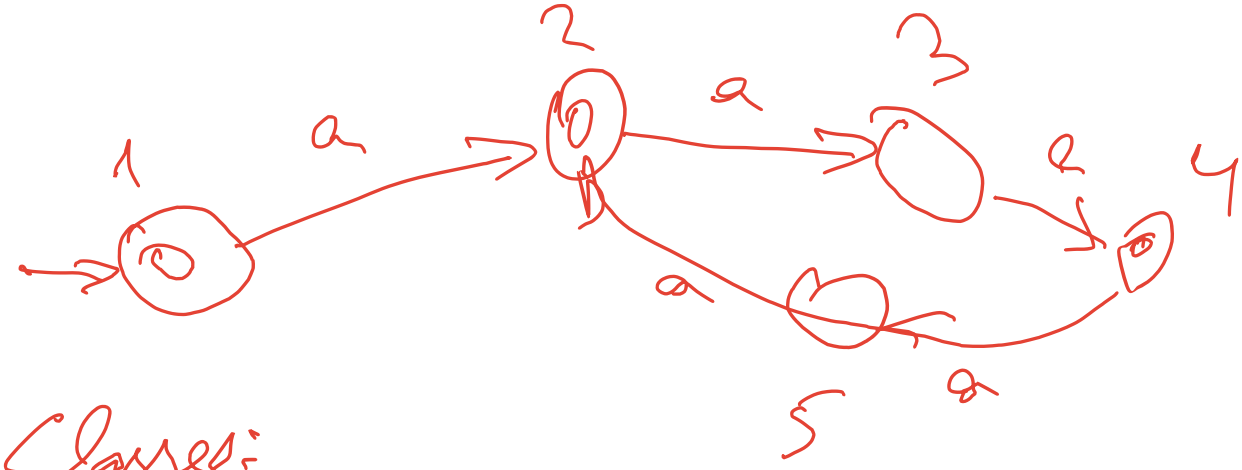
Isso implica que se $p \sim q$, então ambos são finais ou ambos são não-finais.

Congruência em autômatos

Uma congruência num AD é uma congruência do respectivo semiautômato que preserva a "operação" adicional $p \in F$.

Isso implica que se $p \sim q$, então ambos são finais ou ambos são não-finais.

Dada uma congruência, podemos definir o autômato quociente, tomando como estados finais as classes dos estados finais originais.

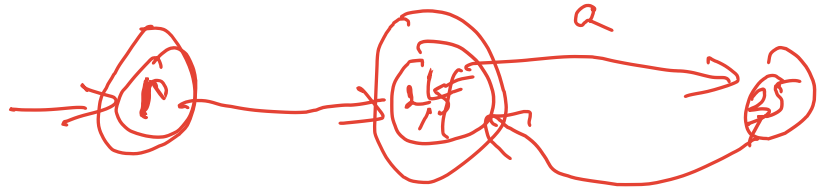


Classes:

{1}

{3,5}

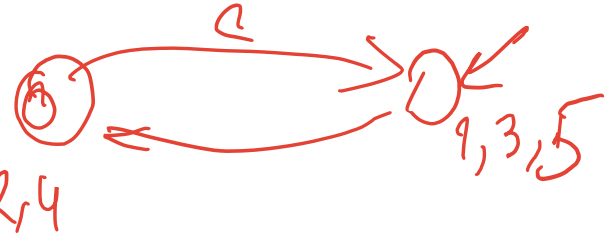
{2,4}



1 não final

{1,3,5}

{2,4}



Homomorfismos de autômatos

Homomorfismos de autômatos

Um homomorfismo $\varphi : (K, \Sigma, \delta, s, F) \to (K', \Sigma, \delta', s', F')$ é um homomorfismo de semiautômatos tal que:

$p \in F \Rightarrow \varphi(p) \in F'$ e $p \in K \setminus F \Rightarrow \varphi(p) \in K' \setminus F'$.

Homomorfismos de autômatos

Um homomorfismo $\varphi : (K, \Sigma, \delta, s, F) \to (K', \Sigma, \delta', s', F')$ é um homomorfismo de semiautômatos tal que:

$p \in F \Rightarrow \varphi(p) \in F'$ e $p \in K \setminus F \Rightarrow \varphi(p) \in K' \setminus F'$.

Equivalentemente: $p \in F \Leftrightarrow \varphi(p) \in F'$.

Homomorfismos de autômatos

Um homomorfismo $\varphi : (K, \Sigma, \delta, s, F) \to (K', \Sigma, \delta', s', F')$ é um homomorfismo de semiautômatos tal que:

$p \in F \Rightarrow \varphi(p) \in F'$ e $p \in K \setminus F \Rightarrow \varphi(p) \in K' \setminus F'$.

Equivalentemente: $p \in F \Leftrightarrow \varphi(p) \in F'$.

Continua valendo:

1. Composição funciona.

Homomorfismos de autômatos

Um homomorfismo $\varphi : (K, \Sigma, \delta, s, F) \to (K', \Sigma, \delta', s', F')$ é um homomorfismo de semiautômatos tal que:

$p \in F \Rightarrow \varphi(p) \in F'$ e $p \in K \setminus F \Rightarrow \varphi(p) \in K' \setminus F'$.

Equivalentemente: $p \in F \Leftrightarrow \varphi(p) \in F'$.

Continua valendo:

1. Composição funciona.
2. A projeção canônica em um quociente é um homomorfismo.

Homomorfismos de autômatos

Um homomorfismo $\varphi : (K, \Sigma, \delta, s, F) \to (K', \Sigma, \delta', s', F')$ é um homomorfismo de semiautômatos tal que:

$p \in F \Rightarrow \varphi(p) \in F'$ e $p \in K \setminus F \Rightarrow \varphi(p) \in K' \setminus F'$.

Equivalentemente: $p \in F \Leftrightarrow \varphi(p) \in F'$.

Continua valendo:

1. Composição funciona.
2. A projeção canônica em um quociente é um homomorfismo.

Homomorfismos de autômatos

Um homomorfismo $\varphi : (K, \Sigma, \delta, s, F) \to (K', \Sigma, \delta', s', F')$ é um homomorfismo de semiautômatos tal que:

$p \in F \Rightarrow \varphi(p) \in F'$ e $p \in K \setminus F \Rightarrow \varphi(p) \in K' \setminus F'$.

Equivalentemente: $p \in F \Leftrightarrow \varphi(p) \in F'$.

Continua valendo:

1. Composição funciona.
2. A projeção canônica em um quociente é um homomorfismo.

Proposição

Se existe homomorfismo de um autômato em outro, então eles reconhecem a mesma linguagem.

Dem:

$$x \in L(\mathcal{A}) \iff sx \in F$$
$$\iff \varphi(sx) \in F'$$
$$\iff \varphi(s)\,x \in F'$$
$$\iff s'x \in F'$$
$$\iff x \in L(\mathcal{A}')$$

□

Dem:

Dem:

$$\begin{aligned} x \in L(\mathscr{A}) &\Leftrightarrow sx \in F \\ &\Leftrightarrow \varphi(sx) \in F' \\ &\Leftrightarrow \varphi(s)x \in F' \\ &\Leftrightarrow s'x \in F' \\ &\Leftrightarrow x \in L(\mathscr{A}'). \end{aligned}$$

□

A congruência natural

A congruência natural

Dado $\mathscr{A} = (K, \Sigma, \delta, s, F)$, vamos definir em K a relação binária

$$p \sim_{\mathscr{A}} q \text{ se para toda palavra } x, \quad px \in F \Leftrightarrow qx \in F.$$

A congruência natural

Dado $\mathscr{A} = (K, \Sigma, \delta, s, F)$, vamos definir em K a relação binária

$$p \sim_{\mathscr{A}} q \text{ se para toda palavra } x, \quad px \in F \Leftrightarrow qx \in F.$$

Prop: $\sim_{\mathscr{A}}$ é uma congruência em $\mathscr{A}$.

A congruência natural

Dado $\mathscr{A} = (K, \Sigma, \delta, s, F)$, vamos definir em K a relação binária

$$p \underset{\mathscr{A}}{\sim} q \text{ se para toda palavra } x, \quad px \in F \Leftrightarrow qx \in F.$$

Prop: $\underset{\mathscr{A}}{\sim}$ é uma congruência em $\mathscr{A}$.

Um AD $\mathscr{A}$ é reduzido se ele é acessível e $\underset{\mathscr{A}}{\sim}$ é a identidade.

A congruência natural

Dado $\mathscr{A} = (K, \Sigma, \delta, s, F)$, vamos definir em K a relação binária

$$p \sim_{\mathscr{A}} q \text{ se para toda palavra } x, \quad px \in F \Leftrightarrow qx \in F.$$

Prop: $\sim_{\mathscr{A}}$ é uma congruência em $\mathscr{A}$.

Um AD $\mathscr{A}$ é reduzido se ele é acessível e $\sim_{\mathscr{A}}$ é a identidade.

Prop: Para todo AD $\mathscr{A}$, $\mathscr{A}/\sim_{\mathscr{A}}$ é reduzido.

Dem:

$A = (Q, \Sigma, \delta, s, F)$

$A' = A/\sim_A \quad \sim = \sim \quad \equiv = \sim'$

$[p] \equiv [q] \iff \forall x \,(\, [p]x \in F' \iff [q]x \in F' \,)$

$\iff \forall x \,(\, [px] \in F' \iff [qx] \in F' \,)$

$\iff \forall x \,(\, px \in F \iff qx \in F \,)$

$\iff p \sim q$

$\iff [p] = [q]$ ∎

O autômato de uma linguagem

Σ^* como semiautômato:

O autômato de uma linguagem

Σ^* como semiautômato:

$K = \Sigma^*$, $s = \lambda$, $\delta(x, \sigma) = x\sigma$.

.

O autômato de uma linguagem

Σ^* como semiautômato:

$K = \Sigma^*$, $s = \lambda$, $\delta(x, y) = xy$.

.

O autômato de uma linguagem

Σ^* como semiautômato:
$K = \Sigma^*$, $s = \lambda$, $\delta(x, y) = xy$.

Dada $L \subseteq \Sigma^*$, definimos A_L colocando $F = L$:
$A_L = (\Sigma^*, \Sigma, \delta, \lambda, L)$.

.

O autômato de uma linguagem

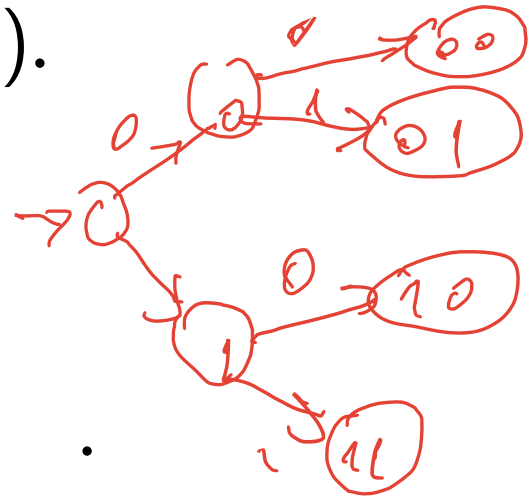
Σ^* como semiautômato:

$K = \Sigma^*, s = \lambda, \delta(x, y) = xy.$

Dada $L \subseteq \Sigma^*$, definimos A_L colocando $F = L$:

$$A_L = (\Sigma^*, \Sigma, \delta, \lambda, L).$$

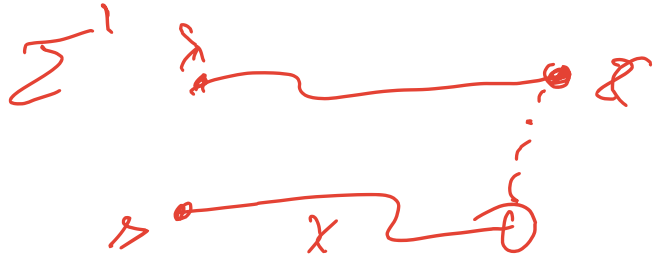
Prop: A_L reconhece L.



O autômato de uma linguagem

Σ^* como semiautômato:
$K = \Sigma^*, s = \lambda, \delta(x, y) = xy$.

Dada $L \subseteq \Sigma^*$, definimos A_L colocando $F = L$:
$A_L = (\Sigma^*, \Sigma, \delta, \lambda, L)$.



Prop: A_L reconhece L.
Prop: Se $\mathscr{A}$ reconhece L, então existe um homomorfismo $A_L \to \mathscr{A}$.

O autômato reduzido de uma linguagem

O autômato reduzido de uma linguagem

Vamos reduzir A_L. A construção toda pode ser descrita diretamente a partir de L.

O autômato reduzido de uma linguagem

Vamos reduzir A_L. A construção toda pode ser descrita diretamente a partir de L.

A relação $\sim_{A_L}$ pode ser denotada mais simplesmente por $\sim_L$, e é uma relação em Σ^*, dada por:

$x \sim_L y$ sse para toda palavra z, $xz \in L \Leftrightarrow yz \in L$.

O autômato reduzido de uma linguagem

Vamos reduzir A_L. A construção toda pode ser descrita diretamente a partir de L.

A relação $\sim_{A_L}$ pode ser denotada mais simplesmente por $\sim_L$, e é uma relação em Σ^*, dada por:

$x \sim_L y$ sse para toda palavra $z, xz \in L \Leftrightarrow yz \in L$.

O autômato reduzido é:

$$\mathscr{A}_L = (\Sigma^*/\sim_L, \Sigma, \delta, [\lambda], \pi(L))$$

onde $\delta([x], \sigma) = [x\sigma]$ e π é a projeção canônica.

O autômato reduzido de uma linguagem

Vamos reduzir A_L. A construção toda pode ser descrita diretamente a partir de L.

A relação $\sim_{A_L}$ pode ser denotada mais simplesmente por $\sim_L$, e é uma relação em Σ^*, dada por:

$$x \sim_L y \text{ sse para toda palavra } z, xz \in L \Leftrightarrow yz \in L.$$

O autômato reduzido é:

$$A_L \longrightarrow \mathscr{A}_L = (\Sigma^*/\sim_L, \Sigma, \delta, [\lambda], \pi(L))$$

onde $\delta([x], \sigma) = [x\sigma]$ e π é a projeção canônica.

Este autômato é muuuuuuuuuito especial.

Uma caracterização

Teorema

(Myhill-Nerode) Uma linguagem L é regular se e somente se $\sim_L$ tem índice finito.

Provar que $L = \{ (aab)^n (abb)^n \mid n \geq 0 \}$ não é regular

Vamos mostrar que

$n \neq m \Rightarrow (aab)^n \not\sim (aab)^m$

$z = (abb)^n$

$(aab)^n z = (aab)^n (abb)^n \in L$

$(aab)^m z = (aab)^m (abb)^n \notin L$