MAC 414
Autômatos, Computabilidade e Complexidade
aula 8 — 7/10/2020

Congruências em semiautômatos

Congruências em semiautômatos

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 semiautômatos

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 semiautômatos

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

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, $[q]\sigma = [q\sigma]$.

Congruências em semiautômatos

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

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, $[q]\sigma = [q\sigma]$.

Para toda $x \in \Sigma^*$, $[q]x = [qx]$.

Congruência em autômatos

Congruência em autômatos

Uma congruência num AD é uma congruência do respectivo semiautômato que satisfaz
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 satisfaz *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.

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

Proposição

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

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

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

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

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.

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

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

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 correspondente 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

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 correspondente é:

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

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

Este autômato é muuuuuuuuuuito especial.

$\mathscr{A}_L$ é o autômato minimal

Teorema

Se $\mathscr{A}$ é um AD que reconhece L, então existe um homomorfismo $\mathscr{A} \to \mathscr{A}_L$.

$\mathscr{A}_L$ é o autômato minimal

Teorema

Se $\mathscr{A}$ é um AD que reconhece L, então existe um homomorfismo $\mathscr{A} \to \mathscr{A}_L$.

$\mathscr{A}_L$ é o autômato minimal

Teorema

Se $\mathscr{A}$ é um AD que reconhece L, então existe um homomorfismo $\mathscr{A} \to \mathscr{A}_L$.

Corolário

Se $\mathscr{A}_L$ é finito, ele minimiza o número de estados entre ADs reconhecendo L e é único a menos de isomorfismo.

Corolário

Se $\mathscr{A}_L$ é finito, ele minimiza o número de estados entre ADs reconhecendo L e é único a menos de isomorfismo.

Corolário

Se $\mathscr{A}_L$ é finito, ele minimiza o número de estados entre ADs reconhecendo L e é único a menos de isomorfismo.

Dem. do corolário: O homomorfismo do Teorema é sobrejetor, logo $|\mathscr{A}_L| \leq |\mathscr{A}|$. Se $\mathscr{A}'$ é outro autômato mínimo para L, existe homomorfismo sobrejetor $\mathscr{A}' \to \mathscr{A}$. Como o número de estados é igual e finito, esse homomorfismo é bijetor, logo isomorfismo. □

Teorema

Se $\mathscr{A}$ é um AD que reconhece L, então existe um homomorfismo $\mathscr{A} \to \mathscr{A}_L$.

□

Teorema

Se $\mathscr{A}$ é um AD que reconhece L, então existe um homomorfismo $\mathscr{A} \to \mathscr{A}_L$.

Dem: Vamos primeiro definir em Σ^* a relação $\equiv$ por $x \equiv y$ sse $sx = sy$ (em $\mathscr{A}$). É fácil ver que $\equiv$ é uma congruência. Além disso, se $x \equiv y$, então $x \sim_L y$.

□

Teorema

Se $\mathscr{A}$ é um AD que reconhece L, então existe um homomorfismo $\mathscr{A} \to \mathscr{A}_L$.

Dem: Vamos primeiro definir em Σ^* a relação $\equiv$ por $x \equiv y$ sse $sx = sy$ (em $\mathscr{A}$). É fácil ver que $\equiv$ é uma congruência. Além disso, se $x \equiv y$, então $x \sim_L y$. Vamos agora definir φ. Se $p \in K$, escolha x tal que $sx = p$ e defina $\varphi(p) = [x]$. Isso não depende da escolha de x: se $sy = p$, então $y \equiv x$, logo $[y] = [x]$.

□

Uma caracterização

Teorema

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

Uma caracterização

Teorema

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

Dem: Se $\sim_L$ tem índice finito, $\mathscr{A}_L$ é um autômato finito; como ele reconhece L, L é regular.

Uma caracterização

Teorema

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

Dem: Se $\sim_L$ tem índice finito, $\mathscr{A}_L$ é um autômato finito; como ele reconhece L, L é regular.
Se L é regular, seja $\mathscr{A}$ um AD reconhecendo L.
Então,

Uma caracterização

Teorema

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

Dem: Se $\sim_L$ tem índice finito, $\mathscr{A}_L$ é um autômato finito; como ele reconhece L, L é regular.
Se L é regular, seja $\mathscr{A}$ um AD reconhecendo L.
Então,

$$|\sim_L|$$

Uma caracterização

Teorema

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

Dem: Se $\sim_L$ tem índice finito, $\mathscr{A}_L$ é um autômato finito; como ele reconhece L, L é regular.
Se L é regular, seja $\mathscr{A}$ um AD reconhecendo L.
Então,

$$|\sim_L| = |\mathscr{A}_L|$$

Uma caracterização

Teorema

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

Dem: Se $\sim_L$ tem índice finito, $\mathscr{A}_L$ é um autômato finito; como ele reconhece L, L é regular.
Se L é regular, seja $\mathscr{A}$ um AD reconhecendo L.
Então,

$$|\sim_L| = |\mathscr{A}_L| \leq |\mathscr{A}|$$

Uma caracterização

Teorema

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

Dem: Se $\sim_L$ tem índice finito, $\mathscr{A}_L$ é um autômato finito; como ele reconhece L, L é regular.
Se L é regular, seja $\mathscr{A}$ um AD reconhecendo L.
Então,

$$|\sim_L| = |\mathscr{A}_L| \leq |\mathscr{A}| < \infty\,.$$

□

Algoritmo de minimização

Dado: Autômato determinístico acessível $\mathscr{A}$, reconhecendo a linguagem L.
Devolve: $\mathscr{A}_L$ e o homomorfismo $\mathscr{A} \to \mathscr{A}_L$.

Algoritmo de minimização

Dado: Autômato determinístico acessível $\mathscr{A}$, reconhecendo a linguagem L.
Devolve: $\mathscr{A}_L$ e o homomorfismo $\mathscr{A} \to \mathscr{A}_L$.

Dois algoritmos conhecidos:

1. Algoritmo de Moore (1956): Mais conceitual. $\mathcal{O}(|\Sigma| n^2)$, onde $n = |\mathscr{A}|$.

Algoritmo de minimização

Dado: Autômato determinístico acessível $\mathscr{A}$, reconhecendo a linguagem L.
Devolve: $\mathscr{A}_L$ e o homomorfismo $\mathscr{A} \to \mathscr{A}_L$.

Dois algoritmos conhecidos:

1. Algoritmo de Moore (1956): Mais conceitual. $\mathcal{O}(|\Sigma| n^2)$, onde $n = |\mathscr{A}|$.
2. Algoritmo de Hopcroft (1960): Mais esperto, $\mathcal{O}(|\Sigma| n \lg n)$.

Algoritmo de minimização

Dado: Autômato determinístico acessível $\mathscr{A}$, reconhecendo a linguagem L.
Devolve: $\mathscr{A}_L$ e o homomorfismo $\mathscr{A} \to \mathscr{A}_L$.

Dois algoritmos conhecidos:

1. Algoritmo de Moore (1956): Mais conceitual. $\mathcal{O}(|\Sigma| n^2)$, onde $n = |\mathscr{A}|$.
2. Algoritmo de Hopcroft (1960): Mais esperto, $\mathcal{O}(|\Sigma| n \lg n)$.

Vamos ver o de Moore.

Aproximações de $\sim_{\mathscr{A}}$

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

Aproximações de $\sim_{\mathscr{A}}$

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

Para $k \geq 0$, definimos

$p \sim_k q$ se para toda $x \in \Sigma^{\leq k}$, $\quad px \in F \Leftrightarrow qx \in F$.

Todas essas relações são de equivalência.

Aproximações de $\sim_{\mathscr{A}}$

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

Para $k \geq 0$, definimos

$p \sim_k q$ se para toda $x \in \Sigma^{\leq k}$, $\quad px \in F \Leftrightarrow qx \in F$.

Todas essas relações são de equivalência.

Em particular, $\sim_0$ tem duas classes, F e $K \setminus F$.

Aproximações de $\sim_{\mathscr{A}}$

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

Para $k \geq 0$, definimos

$p \sim_k q$ se para toda $x \in \Sigma^{\leq k}$, $\quad px \in F \Leftrightarrow qx \in F$.

Todas essas relações são de equivalência.

Em particular, $\sim_0$ tem duas classes, F e $K \setminus F$.

Lema: Para $k > 0$, $p \sim_k q$ sse

1. $p \sim_{k-1} q$, e
2. para todo $\sigma \in \Sigma$, $p\sigma \sim_{k-1} q\sigma$.

Aproximações de $\sim_{\mathscr{A}}$

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

Para $k \geq 0$, definimos

$p \sim_k q$ se para toda $x \in \Sigma^{\leq k}$, $\quad px \in F \Leftrightarrow qx \in F$.

Todas essas relações são de equivalência.

Em particular, $\sim_0$ tem duas classes, F e $K \setminus F$.

Lema: Para $k > 0$, $p \sim_k q$ sse

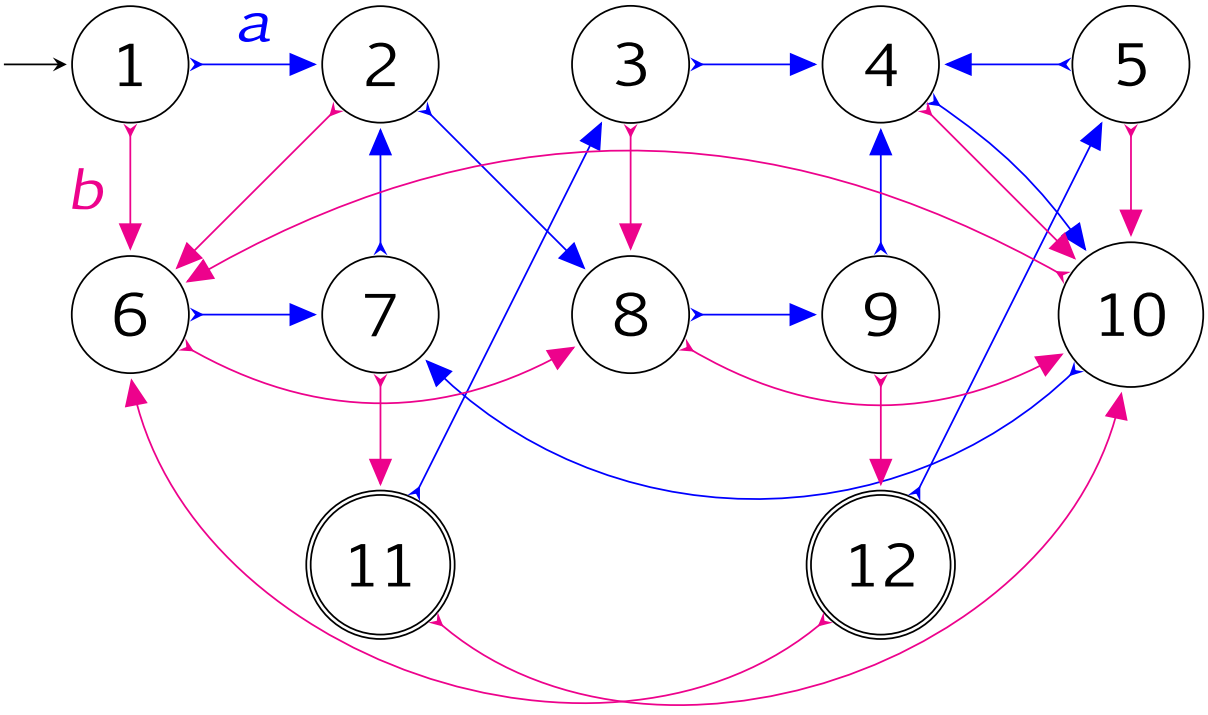
1. $p \sim_{k-1} q$, e
2. para todo $\sigma \in \Sigma$, $p\sigma \sim_{k-1} q\sigma$.

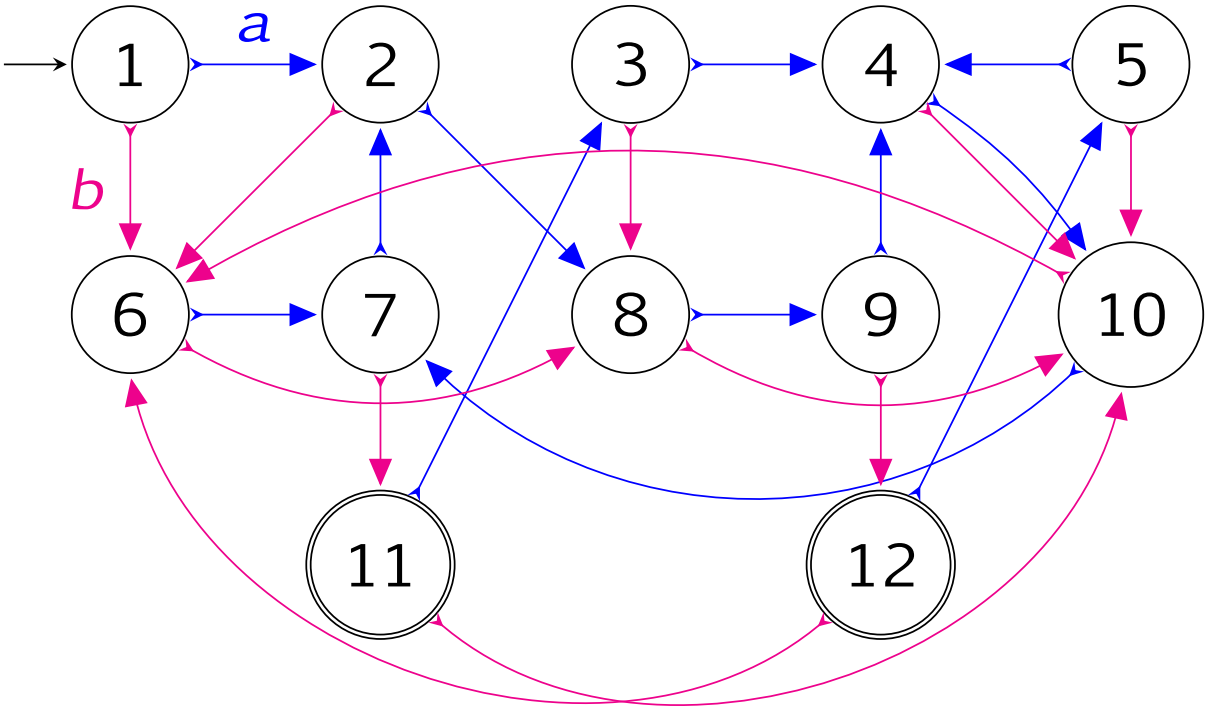
Lema: Se $\sim_k = \sim_{k+1}$, então $\sim_k = \sim_{\mathscr{A}}$.

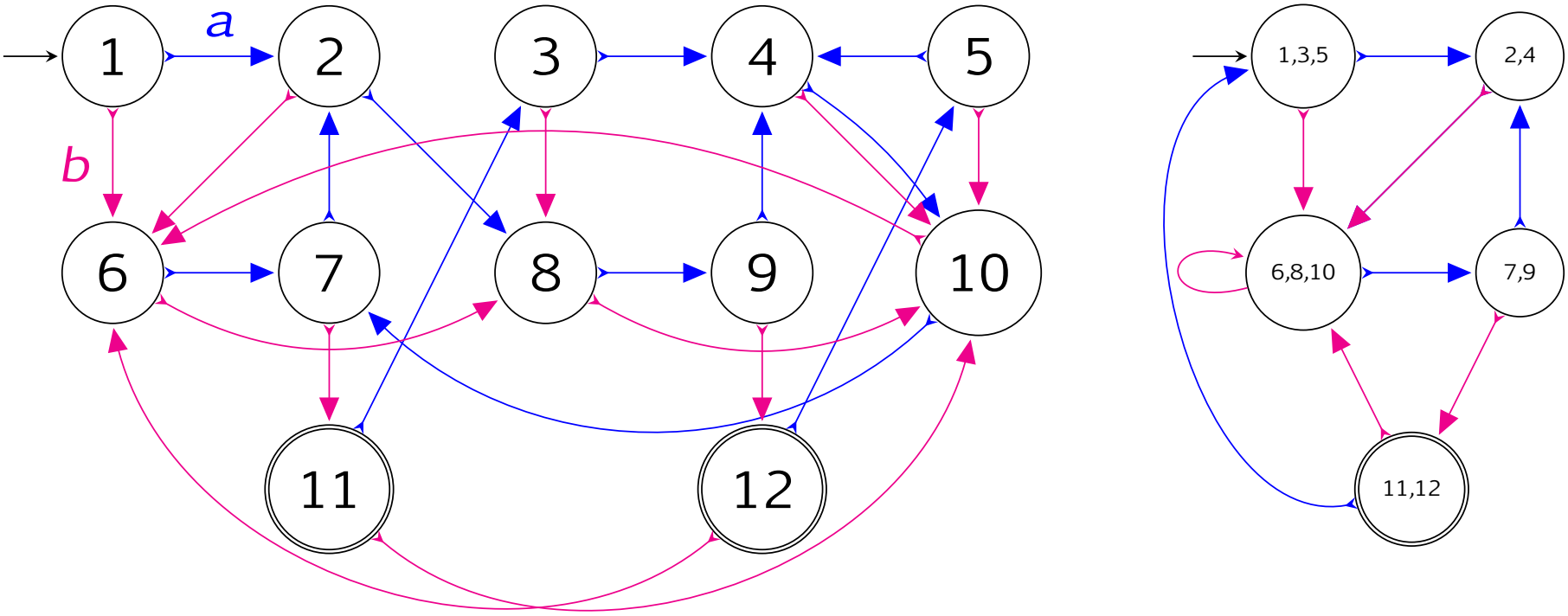
Provas

O Algoritmo

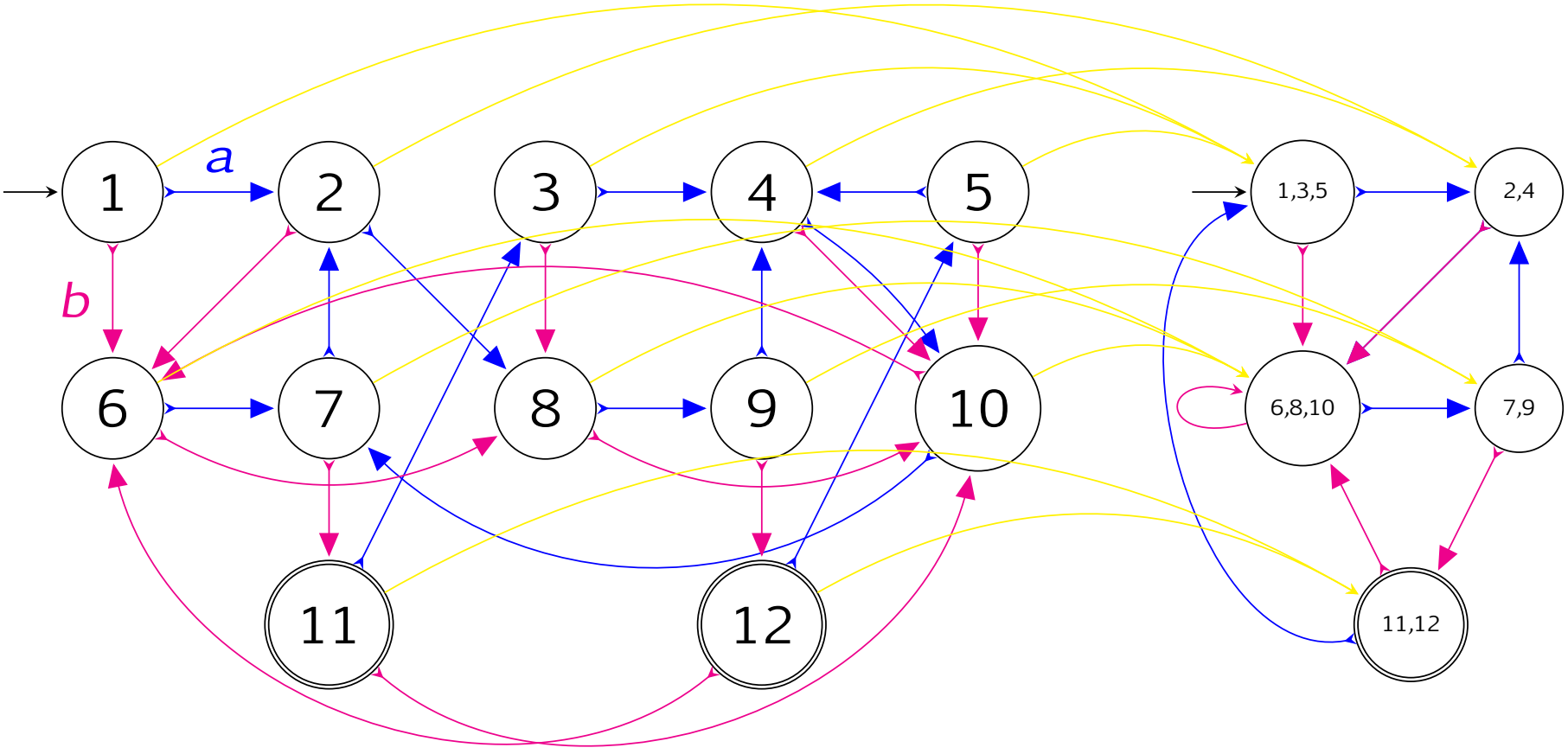
1. Mantém uma partição **Til** de K. Inicialmente **Til** tem blocos F e $K \setminus F$.
2. Para cada estado p, compute $([p\sigma])_{\sigma \in \Sigma}$
3. Compute **NovoTil** quebrando os blocos de **Til**
4. Se **NovoTil**$\neq$**Til**, faça **NovoTil**$\leftarrow$**Til** e volte para (2)
5. Devolva **Til**


a
b
1
2
3
4
5
6
7
8
9
10
11
12


1
a
2
3
4
5
b
6
7
8
9
10
11
12


a
b
1
2
3
4
5
6
7
8
9
10
11
12

1,3,5
2,4
6,8,10
7,9
11,12


a
b
1
2
3
4
5
6
7
8
9
10
11
12
1,3,5
2,4
6,8,10
7,9
11,12

Mais sobre o autômato minimal

Temos exemplos de ANDs para os quais a construção dos subconjuntos é exponencial.

`www.ime.usp.br/~am/414-10/automatos-ruins.pdf`