

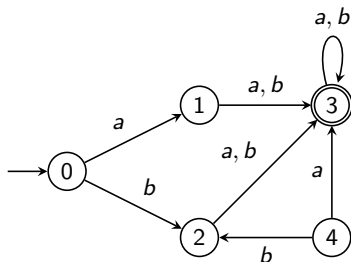
Minimización de autómatas

Clase 08

IIC 2223

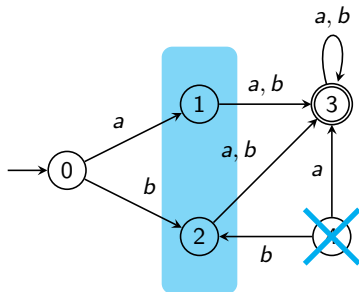
Prof. Cristian Riveros

¿cómo minimizo un autómata finito?



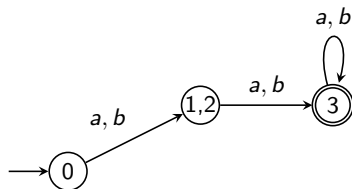
1. Eliminar estados inaccesibles.
 - Fácil de realizar y no cambia el lenguaje del autómata finito.
2. Colapsar estados “equivalentes”.
 - ¿cómo sabemos cuales estados colapsar y cuales no?

¿cómo minimizo un autómata finito?



1. Eliminar estados inaccesibles.
 - Fácil de realizar y no cambia el lenguaje del autómata finito.
2. Colapsar estados “equivalentes”.
 - ¿cómo sabemos cuales estados colapsar y cuales no?

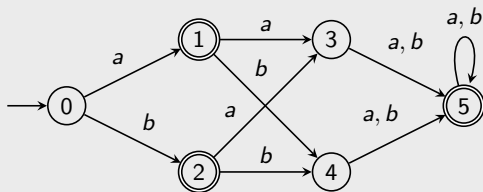
¿cómo minimizo un autómata finito?



1. Eliminar estados inaccesibles.
 - Fácil de realizar y no cambia el lenguaje del autómata finito.
2. Colapsar estados “equivalentes”.
 - ¿cómo sabemos cuales estados colapsar y cuales no?

¿cómo sabemos cuales estados colapsar y cuales no?

Ejemplo

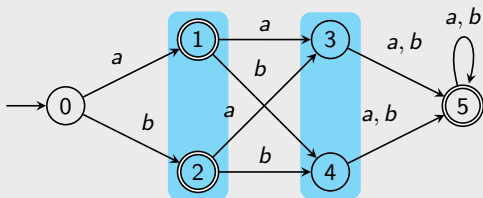


- Podemos colapsar estados 1 y 2.
- Podemos colapsar estados 3 y 4.

¿podemos colapsar más estados?

¿cómo sabemos cuales estados colapsar y cuales no?

Ejemplo

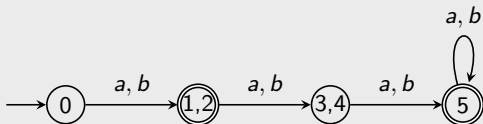


- Podemos colapsar estados 1 y 2.
- Podemos colapsar estados 3 y 4.

¿podemos colapsar más estados?

¿cómo sabemos cuales estados colapsar y cuales no?

Ejemplo

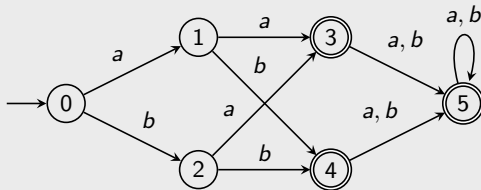


- Podemos colapsar estados 1 y 2.
- Podemos colapsar estados 3 y 4.

¿podemos colapsar más estados?

¿cómo sabemos cuales estados colapsar y cuales no?

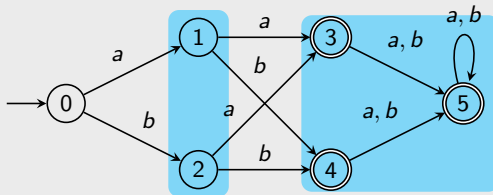
Ejemplo



- Podemos colapsar estados 1 y 2.
- Podemos colapsar estados 3, 4 y 5.

¿cómo sabemos cuales estados colapsar y cuales no?

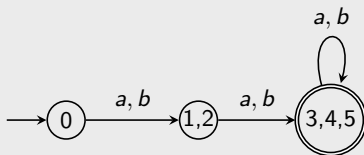
Ejemplo



- Podemos colapsar estados 1 y 2.
- Podemos colapsar estados 3, 4 y 5.

¿cómo sabemos cuales estados colapsar y cuales no?

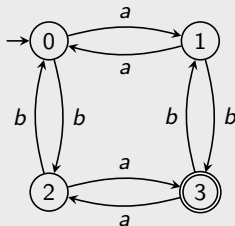
Ejemplo



- Podemos colapsar estados 1 y 2.
- Podemos colapsar estados 3, 4 y 5.

¿cómo sabemos cuales estados colapsar y cuales no?

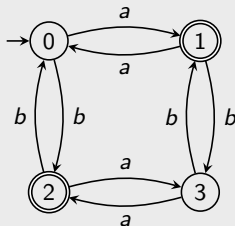
Ejemplo



¿qué estados podemos colapsar?

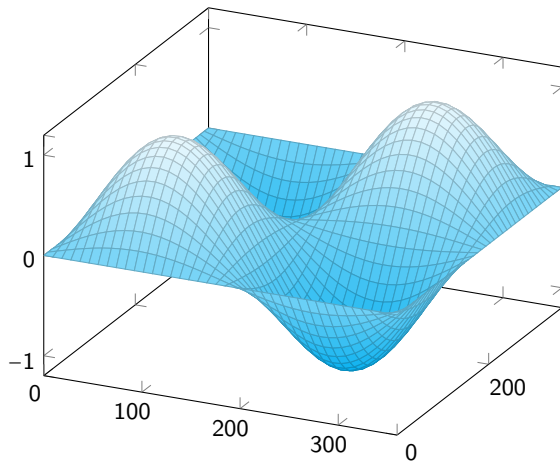
¿cómo sabemos cuales estados colapsar y cuales no?

Ejemplo



¿y ahora? ¿qué estados podemos colapsar?

Minimización de estados



Estrategia: “achicar” nuestro autómata colapsando estados.

Outline

Colapsar estados

Algoritmo de minimización

Outline

Colapsar estados

Algoritmo de minimización

Función de transición extendida

Definición

Sea $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un DFA.

Se define la **función de transición extendida**

$$\hat{\delta} : Q \times \Sigma^* \rightarrow Q$$

inductivamente como:

$$\begin{aligned}\hat{\delta}(q, \epsilon) &\stackrel{\text{def}}{=} q \\ \hat{\delta}(q, w \cdot a) &\stackrel{\text{def}}{=} \delta(\hat{\delta}(q, w), a)\end{aligned}$$

¿cuándo podemos colapsar dos estados?

Sea $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un DFA y $p, q \in Q$.

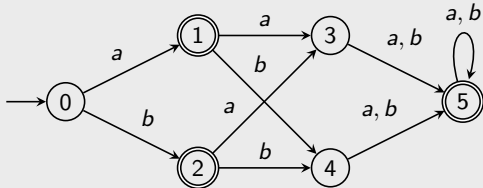
Definición

Decimos que p y q son **indistingibles** ($p \approx_{\mathcal{A}} q$) si:

$$p \approx_{\mathcal{A}} q \quad \text{ssi} \quad (\hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F), \text{ para todo } w \in \Sigma^*.$$

Decimos que p y q son **distingibles** si NO son indistingibles ($p \not\approx_{\mathcal{A}} q$).

¿cuál par de estados son indistingibles?



(Paréntesis): relaciones de equivalencia



(Paréntesis): relaciones de equivalencia

Definición

Una relación \approx_R sobre un conjunto X se dice de **equivalencia** si:

- **reflexiva**: $\forall p \in X. p \approx_R p$
- **simétrica**: $\forall p, q \in X. \text{ si } p \approx_R q \text{ entonces } q \approx_R p.$
- **transitiva**: $\forall p, q, r \in X. \text{ si } p \approx_R q \text{ y } q \approx_R r, \text{ entonces } p \approx_R r.$

Para un elemento $p \in X$ se define su **clase de equivalencia** según \approx_R como:

$$[p]_{\approx_R} = \{q \mid q \approx_R p\}$$

Una función $f : X \rightarrow X$ se dice **bien definida** sobre \approx_R si:

$$p \approx_R q \text{ entonces } f(p) \approx_R f(q)$$

La relación $\approx_{\mathcal{A}}$ es una relación de equivalencia

Definición

Decimos que p y q son **indistinguibles** ($p \approx_{\mathcal{A}} q$) si:

$$p \approx_{\mathcal{A}} q \quad \text{ssi} \quad (\hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F), \text{ para todo } w \in \Sigma^*$$

Propiedades

- $\approx_{\mathcal{A}}$ es una **relación de equivalencia** entre estados.
 - **reflexiva:** $\forall p \in Q. p \approx_{\mathcal{A}} p$
 - **simétrica:** $\forall p, q \in Q. \text{ si } p \approx_{\mathcal{A}} q \text{ entonces } q \approx_{\mathcal{A}} p.$
 - **transitiva:** $\forall p, q, r \in Q. \text{ si } p \approx_{\mathcal{A}} q \text{ y } q \approx_{\mathcal{A}} r, \text{ entonces } p \approx_{\mathcal{A}} r.$

La relación $\approx_{\mathcal{A}}$ es una relación de equivalencia

Definición

Decimos que p y q son **indistinguibles** ($p \approx_{\mathcal{A}} q$) si:

$$p \approx_{\mathcal{A}} q \quad \text{ssi} \quad (\hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F), \text{ para todo } w \in \Sigma^*$$

Propiedades

- $\approx_{\mathcal{A}}$ es una **relación de equivalencia** entre estados.
- Cada estado $p \in Q$ esta en exactamente una clase de equivalencia:

$$[p]_{\approx_{\mathcal{A}}} = \{q \mid q \approx_{\mathcal{A}} p\}$$

- Para todo $a \in \Sigma$ la función $\delta(\cdot, a) : Q \rightarrow Q$ esta **bien definida** sobre $\approx_{\mathcal{A}}$.

$$p \approx_{\mathcal{A}} q \text{ entonces } \delta(p, a) \approx_{\mathcal{A}} \delta(q, a)$$

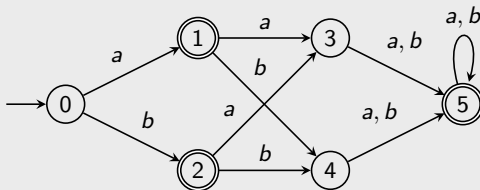
El autómata cuociente

Para un DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ se define el DFA:

$$\mathcal{A} / \approx = (Q_{\approx}, \Sigma, \delta_{\approx}, q_{\approx}, F_{\approx})$$

- $Q_{\approx} = \{ [p]_{\approx_{\mathcal{A}}} \mid p \in Q \}$
- $\delta_{\approx}([p]_{\approx_{\mathcal{A}}}, a) = [\delta(p, a)]_{\approx_{\mathcal{A}}}$
- $q_{\approx} = [q_0]_{\approx_{\mathcal{A}}}$
- $F_{\approx} = \{ [p]_{\approx_{\mathcal{A}}} \mid p \in F \}$

¿cuál es el autómata cuociente para \mathcal{A} ?



El autómata cuociente

Para un DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ se define el DFA:

$$\mathcal{A}/\approx = (Q_{\approx}, \Sigma, \delta_{\approx}, q_{\approx}, F_{\approx})$$

- $Q_{\approx} = \{ [p]_{\approx_{\mathcal{A}}} \mid p \in Q \}$
- $\delta_{\approx}([p]_{\approx_{\mathcal{A}}}, a) = [\delta(p, a)]_{\approx_{\mathcal{A}}}$
- $q_{\approx} = [q_0]_{\approx_{\mathcal{A}}}$
- $F_{\approx} = \{ [p]_{\approx_{\mathcal{A}}} \mid p \in F \}$

¿está \mathcal{A}/\approx bien definido? ¿depende la definición de δ_{\approx} de p ?

El autómata cuociente define el mismo lenguaje

Teorema

Para todo autómata finito determinista \mathcal{A} se cumple que:

$$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}/\approx)$$

El autómata cociente define el mismo lenguaje

Demostración

Sea $w = a_1 a_2 \dots a_n \in \Sigma^*$.

PD: $w \in \mathcal{L}(\mathcal{A}) \Leftrightarrow w \in \mathcal{L}(\mathcal{A}/\approx)$

Existe una ejecución ρ de \mathcal{A} sobre w :

$$\rho : p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} p_n$$

- $p_0 = q_0$.
- $\delta(p_i, a_{i+1}) = p_{i+1} \quad \forall i \in \{0, \dots, n-1\}$.

Existe una ejecución ρ_\approx de \mathcal{A}/\approx sobre w :

$$\rho_\approx : X_0 \xrightarrow{a_1} X_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} X_n$$

- $X_0 = q_\approx$.
- $\delta_\approx(X_i, a_{i+1}) = X_{i+1} \quad \forall i \in \{0, \dots, n-1\}$.

¿qué relación hay entre p_i y X_i ?

El autómata cociente define el mismo lenguaje

Demostración

PD: $[p_i]_{\approx_{\mathcal{A}}} = X_i \quad \forall i \in \{0, \dots, n\}$

Por inducción sobre i .

(¿cómo?)

PD: $p_n \in F$ si, y solo si, $[p_n]_{\approx_{\mathcal{A}}} \in F_{\approx}$

(¿cómo?)

Por lo tanto, $w \in \mathcal{L}(\mathcal{A}) \Leftrightarrow w \in \mathcal{L}(\mathcal{A}/\approx)$



Outline

Colapsar estados

Algoritmo de minimización

Algoritmo para encontrar las clases de equivalencia

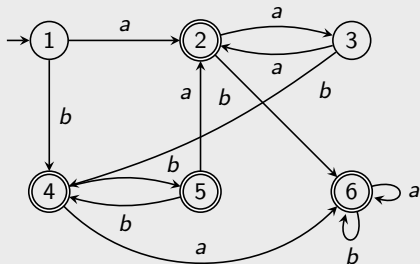
Buscamos los pares que son **distingibles**.

Busqueda de clase de estados distinguibles

1. Construya una tabla con los pares $\{p, q\}$ inicialmente sin marcar.
2. Marque $\{p, q\}$ si $p \in F$ y $q \notin F$ o viceversa.
3. Repita este paso hasta que no hayan mas cambios:
 - Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ estan marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.
4. Al terminar, $p \not\sim_{\mathcal{A}} q$ ssi la entrada $\{p, q\}$ está marcada.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles

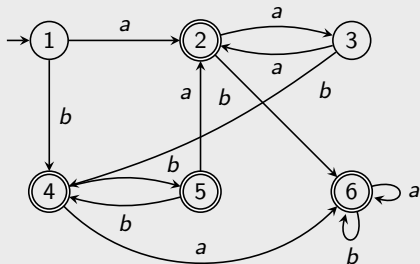


	①	②	③	④	⑤	⑥
①						
②						
③						
④						
⑤						
⑥						

1. Construya una tabla con los pares $\{p, q\}$ inicialmente sin marcar.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles

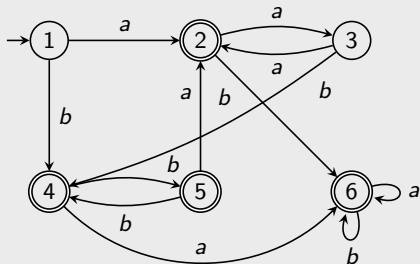


	①	②	③	④	⑤	⑥
①						
②						
③						
④						
⑤						
⑥						

1. Construya una tabla con los pares $\{p, q\}$ inicialmente sin marcar.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles

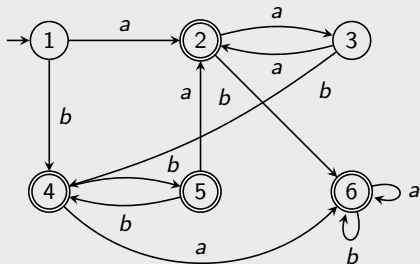


	①	②	③	④	⑤	⑥
①						
②						
③						
④						
⑤						
⑥						

2. Marque $\{p, q\}$ si $p \in F$ y $q \notin F$ o viceversa.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles

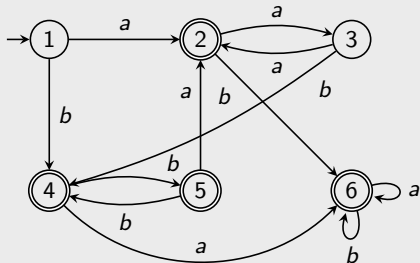


	①	②	③	④	⑤	⑥
①		✓		✓	✓	✓
②			✓			
③				✓	✓	✓
④						
⑤						
⑥						

2. Marque $\{p, q\}$ si $p \in F$ y $q \notin F$ o viceversa.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles



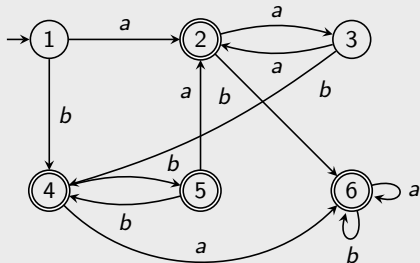
	①	②	③	④	⑤	⑥
①		✓		✓	✓	✓
②			✓			
③				✓	✓	✓
④						
⑤						
⑥						

3. Repita este paso hasta que no hayan mas cambios:

- Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ están marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles



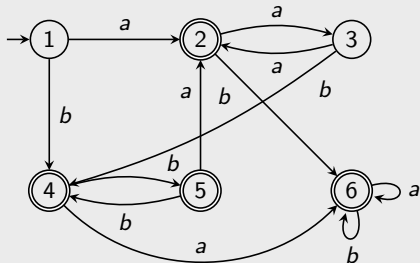
	①	②	③	④	⑤	⑥
①		✓		✓	✓	✓
②			✓	a (3,6)	a (3,2)	a (3,6)
③				✓	✓	✓
④						
⑤						
⑥						

3. Repita este paso hasta que no hayan mas cambios:

- Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ están marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles



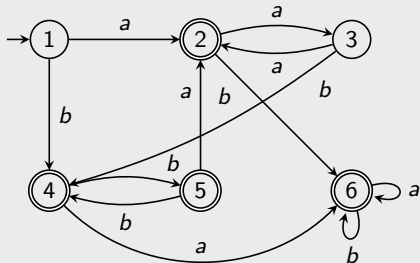
	①	②	③	④	⑤	⑥
①		✓		✓	✓	✓
②			✓	✓	✓	✓
③				✓	✓	✓
④						
⑤						
⑥						

3. Repita este paso hasta que no hayan mas cambios:

- Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ están marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles



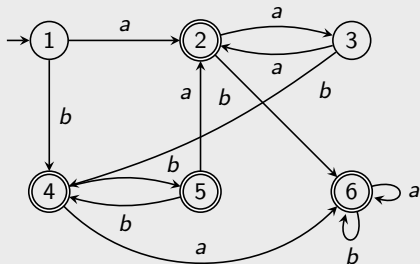
	①	②	③	④	⑤	⑥
①		✓		✓	✓	✓
②			✓	✓	✓	✓
③				✓	✓	✓
④					^a (6,2)	
⑤						^a (2,6)
⑥						

3. Repita este paso hasta que no hayan mas cambios:

- Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ están marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles



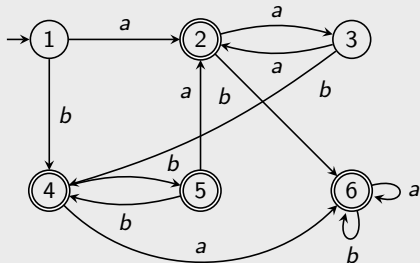
	①	②	③	④	⑤	⑥
①		✓		✓	✓	✓
②			✓	✓	✓	✓
③				✓	✓	✓
④					✓	
⑤						✓
⑥						

3. Repita este paso hasta que no hayan mas cambios:

- Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ están marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles



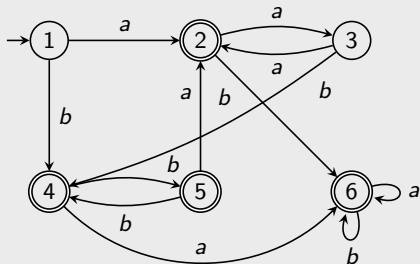
	①	②	③	④	⑤	⑥
①		✓		✓	✓	✓
②			✓	✓	✓	✓
③				✓	✓	✓
④					✓	b (5,6)
⑤						✓
⑥						

3. Repita este paso hasta que no hayan mas cambios:

- Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ están marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles



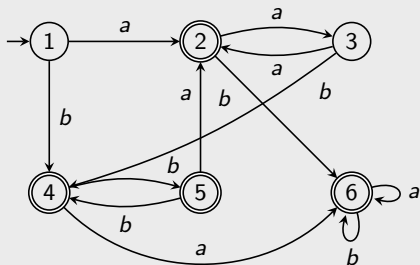
	①	②	③	④	⑤	⑥
①		✓		✓	✓	✓
②			✓	✓	✓	✓
③				✓	✓	✓
④					✓	✓
⑤						✓
⑥						

3. Repita este paso hasta que no hayan mas cambios:

- Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ están marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

Algoritmo para encontrar las clases de equivalencia

Encuentre los estados distinguibles



	①	②	③	④	⑤	⑥
①		✓		✓	✓	✓
②			✓	✓	✓	✓
③				✓	✓	✓
④					✓	✓
⑤						✓
⑥						

4. Al terminar, $p \not\sim_A q$ ssi la entrada $\{p, q\}$ está marcada.

Los pares indistinguibles serán todas las entradas **NO** marcadas.

Algunos detalles del algoritmo

- ¿siempre termina el algoritmo anterior?
- ¿és el algoritmo anterior correcto?
- ¿cómo construimos nuestro autómata cuociente?
- ¿cuál es el tiempo del algoritmo en relación a la cantidad de estados?

Demostración: ejercicio.

Varias preguntas sin resolver sobre minimización

1. ¿cómo sabemos si el autómata del algoritmo es un **mínimo**?
2. Dado L , ¿existe un **único** autómata mínimo?
3. Dado un \mathcal{A} , ¿és posible **construir** un autómata mínimo equivalente?

Estas preguntas las responderemos la próxima clase!