# Algoritmo de minimización de autómatas

Clase 11

IIC2223 / IIC2224

Prof. Cristian Riveros

# Outline

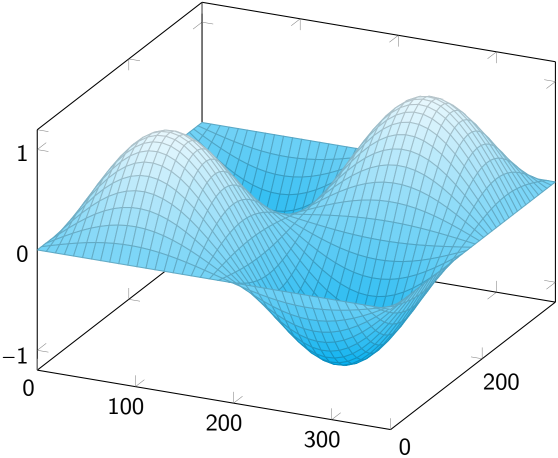
Colapsar estados (recordatorio)

Autómata cuociente

Algoritmo de minimización

# Outline

# Minimización de estados




Estrategia: "reducir" nuestro autómata colapsando estados.

# Función de transición extendida

## Definición

Sea $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un DFA.
Se define la **función de transición extendida**

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

inductivamente como:

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

# ¿cuándo podemos colapsar dos estados?

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

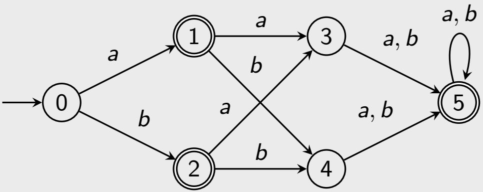
## Definición

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

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

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

¿cuál par de estados son indistinguibles?



# (Paréntesis): relaciones de equivalencia

# (Paréntesis): relaciones de equivalencia

## Definición

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

- **reflexiva**: $\forall p \in X.\ p \approx p$
- **simétrica**: $\forall p, q \in X.$ si $p \approx q$ entonces $q \approx p$.
- **transitiva**: $\forall p, q, r \in X.$ si $p \approx q$ y $q \approx r$, entonces $p \approx r$.

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

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

Una función $f : X \to X$ se dice **bien definida** sobre $\approx$ si:

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

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

## Definición

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

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

## 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 $\delta(\cdot, a) : Q \to Q$ esta **bien definida** sobre $\approx_{\mathcal{A}}$.

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

# Outline

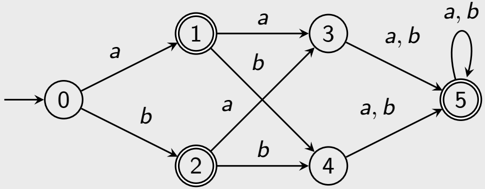
# El autómata cuociente

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

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

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

¿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:

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

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

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

# El autómata cuociente define el mismo lenguaje

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

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

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

## Teorema

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

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

# El autómata cuociente 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 $\rho$ 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 $\rho_\approx$ 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 cuociente define el mismo lenguaje

## Demostración

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

Por inducción sobre $i$.

**Caso base:** $[p_0]_{\approx_\mathcal{A}} = [q_0]_{\approx_\mathcal{A}} = q_\approx = X_0$ (por definición)

**Caso inductivo:** Suponemos que $[p_i]_{\approx_\mathcal{A}} = X_i$. (HI)

$$
\begin{aligned}
X_{i+1} &= \delta_\approx(X_i, a_{i+1}) && \text{(definición } \rho_\approx\text{)} \\
&= \delta_\approx([p_i]_{\approx_\mathcal{A}}, a_{i+1}) && \text{(HI)} \\
&= [\delta(p_i, a_{i+1})]_{\approx_\mathcal{A}} && \text{(definición } \delta_\approx\text{)} \\
&= [p_{i+1}]_{\approx_\mathcal{A}} && \text{(definición } \rho\text{)}
\end{aligned}
$$

# El autómata cuociente define el mismo lenguaje

## Demostración

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

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

($\Rightarrow$) Definición de $F_{\approx}$.

($\Leftarrow$) Suponemos que $[p_n]_{\approx_{\mathcal{A}}} \in F_{\approx}$.

Existe $p \in F$ tal que $[p_n]_{\approx_{\mathcal{A}}} = [p]_{\approx_{\mathcal{A}}}$. (definición $F_{\approx}$)

Como $p_n \approx_{\mathcal{A}} p$, entonces:

$$(\hat{\delta}(p_n, w) \in F \iff \hat{\delta}(p, w) \in F), \text{ para todo } w \in \Sigma^*.$$

Si escogemos $w = \epsilon$: $(\hat{\delta}(p_n, \epsilon) = p_n \in F \iff \hat{\delta}(p, \epsilon) = p \in F)$.

Como $p \in F$, concluimos que $p_n \in F$.

Por lo tanto, $w \in \mathcal{L}(\mathcal{A})$ si, y solo si, $w \in \mathcal{L}(\mathcal{A}/\approx)$. ■

# Outline

# Algoritmo para encontrar las clases de equivalencia

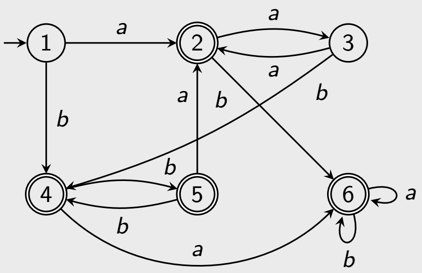
Buscamos los pares que son **distingibles**.

## Busqueda de clase de estados distingibles

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, $p \not\approx_{\mathcal{A}} q$ ssi la entrada $\{p, q\}$ está marcada.

# Algoritmo para encontrar las clases de equivalencia

## Encuentre los estados distinguibles

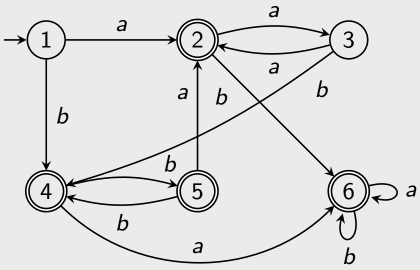



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |

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

# Algoritmo para encontrar las clases de equivalencia

## Encuentre los estados distinguibles

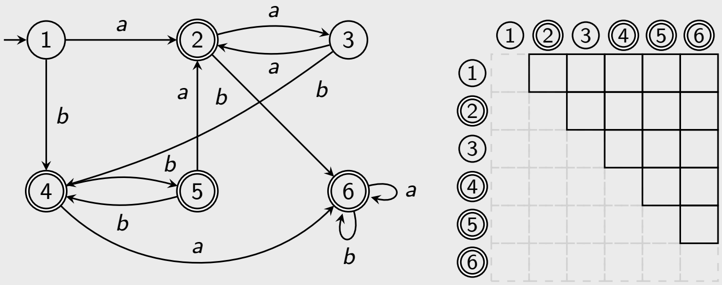



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |

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

# Algoritmo para encontrar las clases de equivalencia

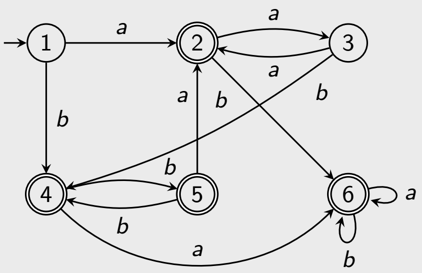
## Encuentre los estados distinguibles




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

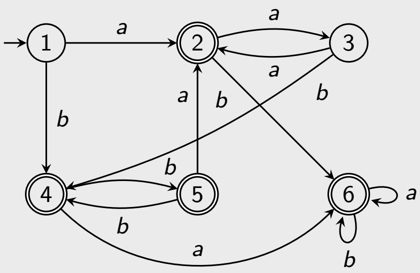



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | ✓ | | ✓ | ✓ | ✓ |
| 2 | | | ✓ | | | |
| 3 | | | | ✓ | ✓ | ✓ |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |

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

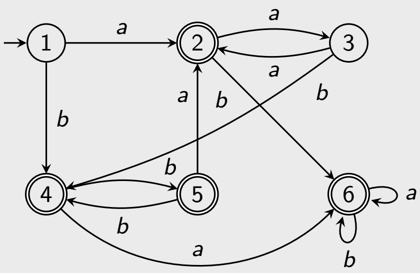



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | ✓ | | ✓ | ✓ | ✓ |
| 2 | | | ✓ | | | |
| 3 | | | | ✓ | ✓ | ✓ |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |

3. Repita este paso hasta que no hayan mas cambios:
   - Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ estan marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

# Algoritmo para encontrar las clases de equivalencia

## Encuentre los estados distinguibles

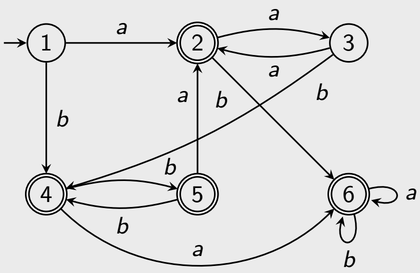



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | ✓ | | ✓ | ✓ | ✓ |
| 2 | | | ✓ | a (3,6) | a (3,2) | a (3,6) |
| 3 | | | | ✓ | ✓ | ✓ |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |

3. Repita este paso hasta que no hayan mas cambios:
   - Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ estan marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

# Algoritmo para encontrar las clases de equivalencia

## Encuentre los estados distinguibles

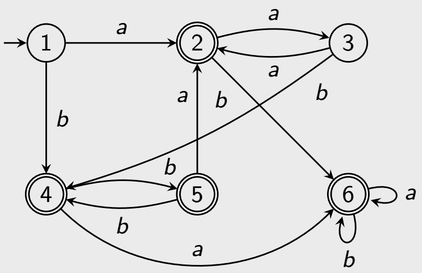



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | ✓ | | ✓ | ✓ | ✓ |
| 2 | | | ✓ | ✓ | ✓ | ✓ |
| 3 | | | | ✓ | ✓ | ✓ |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |

3. Repita este paso hasta que no hayan mas cambios:
   - Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ estan marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

# Algoritmo para encontrar las clases de equivalencia

## Encuentre los estados distinguibles

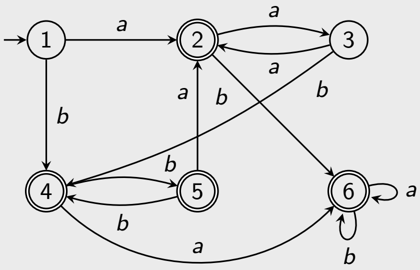



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | ✓ | | ✓ | ✓ | ✓ |
| 2 | | | ✓ | ✓ | ✓ | ✓ |
| 3 | | | | ✓ | ✓ | ✓ |
| 4 | | | | | a (6,2) | |
| 5 | | | | | | a (2,6) |
| 6 | | | | | | |

3. Repita este paso hasta que no hayan mas cambios:
   - Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ estan marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

# Algoritmo para encontrar las clases de equivalencia

## Encuentre los estados distinguibles

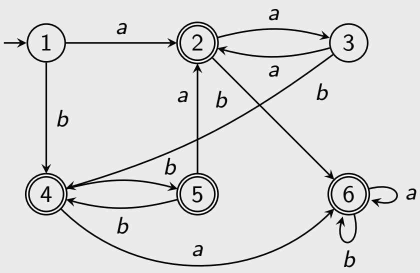



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | ✓ | | ✓ | ✓ | ✓ |
| 2 | | | ✓ | ✓ | ✓ | ✓ |
| 3 | | | | ✓ | ✓ | ✓ |
| 4 | | | | | ✓ | |
| 5 | | | | | | ✓ |
| 6 | | | | | | |

3. Repita este paso hasta que no hayan mas cambios:
   - Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ estan marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

# Algoritmo para encontrar las clases de equivalencia

## Encuentre los estados distinguibles

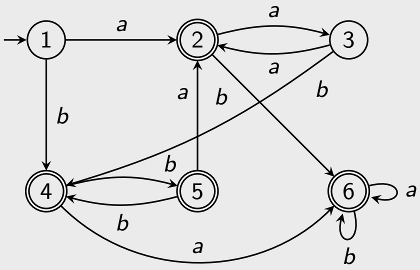



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | ✓ | | ✓ | ✓ | ✓ |
| 2 | | | ✓ | ✓ | ✓ | ✓ |
| 3 | | | | ✓ | ✓ | ✓ |
| 4 | | | | | ✓ | b (5,6) |
| 5 | | | | | | ✓ |
| 6 | | | | | | |

3. Repita este paso hasta que no hayan mas cambios:
   - Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ estan marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

# Algoritmo para encontrar las clases de equivalencia

## Encuentre los estados distinguibles

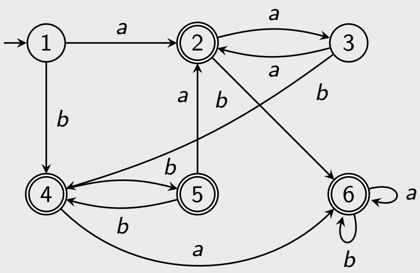



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | ✓ | | ✓ | ✓ | ✓ |
| 2 | | | ✓ | ✓ | ✓ | ✓ |
| 3 | | | | ✓ | ✓ | ✓ |
| 4 | | | | | ✓ | ✓ |
| 5 | | | | | | ✓ |
| 6 | | | | | | |

3. Repita este paso hasta que no hayan mas cambios:
   - Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ estan 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, $p \not\approx_{\mathcal{A}} q$ 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.

# Cierre de clase

En esta clase vimos:

1. Autómata cuociente.
2. Demostramos que es equivalente al autómata inicial.
3. Algoritmo de minimización encontrando estados distinguibles.

Varias preguntas sin resolver sobre minimización:

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

**Próximas clases** responderemos estas preguntas.