Teoría de Lenguajes

Clase Teórica

Minimización Autómatas Finitos

Segundo cuatrimestre 2025

Bibliografía: Capítulo 4, *Introduction to Automata Theory, Languages and Computation*, J. Hopcroft, R. Motwani, J. Ullman, Second Edition, Addison Wesley, 2001.

Teorema

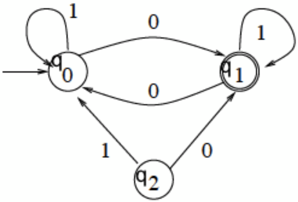
Para todo autómata finito determinimístico hay otro que reconoce el mismo lenguaje y tiene una cantidad mímina de estados.

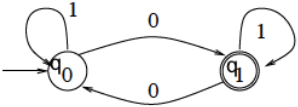
Demostración.

Fuerza bruta, y algoritmo para determinar si dos autómatas finitos reconocen el mismo lenguaje. □

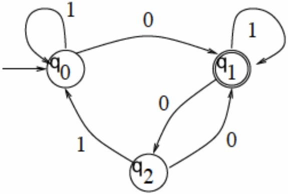
Ejemplos de minimización de AFD

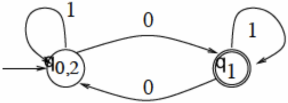
$M =< Q, \Sigma, \delta, q_0, F >$. El estado q_2 es inaccesible, entonces puede ser quitado. $\mathcal{L}(M) = (1^*01^*)(01^*01^*)^*$.

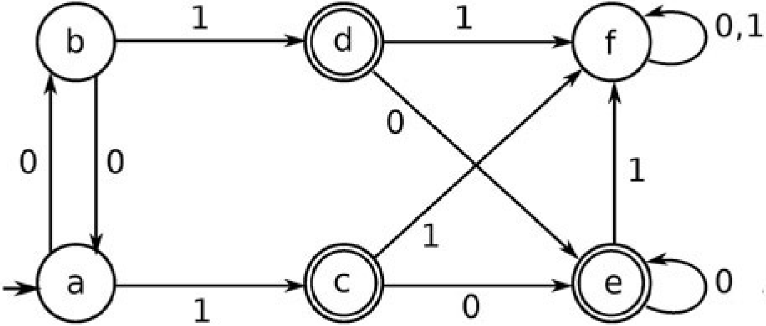




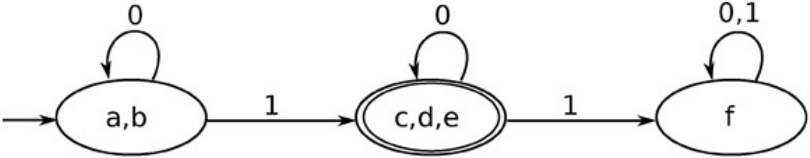
$M' =< Q, \Sigma, \delta', q_0, F >$. No hay estados inaccesibles.
$\mathcal{L}(M') = (1^*01^*)(01^*01^*)^*$.







b
d
f
a
c
e
0,1
0
1



a,b
c,d,e
f
0
0,1
1

Definición

Sea $M =< Q, \Sigma, \delta, q_0, F >$ *un AFD. Definimos* $\equiv$ *la relación de indistinguibilidad sobre* Q*: dos estados* $q, r \in Q$ *son indistinguibles, que denotamos* $q \equiv r$ *, cuando*

$$\forall \alpha \in \Sigma^*, (\widehat{\delta}(q, \alpha) \in F \text{ si y solo si } \widehat{\delta}(r, \alpha) \in F).$$

Observación

Todo par de estados indistinguibles, al consumir cualquier cadena $\alpha \in \Sigma^*$, llegan a otro par de estados indistinguibles:

$$\textit{Si } q \equiv r \quad \textit{entonces } \forall \alpha \in \Sigma^*, \big(\widehat{\delta}(q,\alpha) \equiv \widehat{\delta}(r,\alpha)\big)$$

Demostración. Supongamos $q \equiv r$ pero $\exists \alpha \in \Sigma^*, (\widehat{\delta}(q,\alpha) \not\equiv \widehat{\delta}(r,\alpha))$. Entonces existe una cadena β que distingue $\widehat{\delta}(q,\alpha)$ de $\widehat{\delta}(r,\alpha)$:

$$\exists \beta \in \Sigma^*, (\widehat{\delta}(\widehat{\delta}(q,\alpha),\beta) \in F \,\wedge\, \widehat{\delta}(\widehat{\delta}(r,\alpha),\beta) \notin F)$$

o viceversa. Esto equivale a decir que

$$\widehat{\delta}(q,\alpha\beta) \in F \,\wedge\, \widehat{\delta}(r,\alpha\beta) \notin F$$

o viceversa. Pero entonces $q \not\equiv r$, y arribamos a una contradicción.

Teorema

La indistinguibilidad $\equiv$ *es una relación de equivalencia.*

Demostración.

- reflexividad: Debemos ver que para todo $q \in Q$, $q \equiv q$.

$$q \equiv q \text{ si y solo si } \forall \alpha \in \Sigma^*, (\widehat{\delta}(q, \alpha) \in F \Leftrightarrow \widehat{\delta}(q, \alpha) \in F)$$

  y esta doble implicación es siempre verdadera.

- simetría: Supongamos $q \equiv r$. Entonces

$$\forall \alpha \in \Sigma^*, (\widehat{\delta}(q, \alpha) \in F \Leftrightarrow \widehat{\delta}(r, \alpha) \in F). \text{ Luego,}$$
$$\forall \alpha \in \Sigma^*, (\widehat{\delta}(r, \alpha) \in F \Leftrightarrow \widehat{\delta}(q, \alpha) \in F). \text{ Por lo tanto, } r \equiv q.$$

- transitividad: Supongamos $q \equiv r$ and $r \equiv s$. Entonces,

$$\forall \alpha \in \Sigma^*, (\widehat{\delta}(q, \alpha) \in F \Leftrightarrow \widehat{\delta}(r, \alpha) \in F), y$$
$$\forall \alpha \in \Sigma^*, (\widehat{\delta}(r, \alpha) \in F \Leftrightarrow \widehat{\delta}(s, \alpha) \in F)$$

  Por lo tanto, $\forall \alpha \in \Sigma^*, (\widehat{\delta}(q, \alpha) \in F \Leftrightarrow \widehat{\delta}(s, \alpha) \in F)$. Es decir, $q \equiv s$.

Definición

Si A es un conjunto y $\sim$ una relación de equivalencia sobre A, entonces las clases de equivalencia forman una partición del conjunto A.

Las clases de equivalencia de la relación $\sim$ determinan un nuevo conjunto, denominado conjunto cociente y denotado $A/\sim$.

Estados accesibles

Definición

El estado p de AFD $M = (Q, \Sigma, \delta, q_0, F)$ es accesible si existe $w \in \Sigma^*,\ p = \widehat{\delta}(q_0, w)$.

Algoritmo que computa el conjunto de estados accesibles

```
Input M
Output conjunto de estados accesibles

accesibles:={q0}
nuevos:= {q0}
repetir
       temp:=∅
       para cada q en nuevos
             para  todo c en Σ
                   temp:= temp∪{δ(q,c)}
       nuevos:=temp - accesibles
       accesibles:= accesibles ∪ nuevos
hasta (nuevos = ∅)
```

Definición (Autómata Finito Determinístico Mínimo)

Sea $M = < Q, \Sigma, \delta, q_0, F >$ *un AFD sin estados inaccesibles, el AFD mínimo equivalente* $M_{min} = < Q_{min}, \Sigma, \delta_{min}, q_{min_0}, F_{min} >$ *es*

$$\begin{aligned} Q_{min} &= (Q \,/ \equiv) \ \textit{(las clases de equivalencia de } \equiv \textit{)} \\ \delta_{min}\left(\left[q\right], a\right) &= \left[\delta\left(q, a\right)\right] \\ q_{min_0} &= [q_0] \\ F_{min} &= \{[q] \in Q_{min} : q \in F\} \end{aligned}$$

Veamos que $\mathcal{L}(M) = \mathcal{L}(M_{min})$.
Mostremos que para toda $\alpha \in \Sigma^*$,

$$\widehat{\delta}(q_0, \alpha) \in F \text{ si y solo si } \widehat{\delta}_{min}([q_0], \alpha) \in F_{min}$$

Alcanza con ver que

$$[\widehat{\delta}(q_0, \alpha)] = \widehat{\delta}_{min}([q_0], \alpha)$$

Lo demostramos en el siguiente lema.

Lema

Para todo $q \in Q$ *y* $\alpha \in \Sigma^*$, $\widehat{\delta}_{min}([q], \alpha) = [\widehat{\delta}(q, \alpha)]$.

Demostración. Por inducción la estructura de α.
Caso base $\alpha = \lambda$. Por definición de $\widehat{\delta}$ y de $\widehat{\delta}_{min}$,
$\widehat{\delta}(q, \lambda) = q$ y $\widehat{\delta}_{min}([q], \lambda) = [q]$
Concluimos que, $\widehat{\delta}_{min}([q], \lambda)] = [\widehat{\delta}(q, \lambda)] = [q]$.

Caso inductivo $\alpha = \beta a$ con $\beta \in \Sigma^*, a \in \Sigma$.
Asumamos que vale para β, es decir

$$\widehat{\delta}_{min}([q], \beta) = [\widehat{\delta}(q, \beta)]$$

Luego,

$$\begin{array}{ll}
\widehat{\delta}_{min}([q], \beta a) & = \delta_{min}(\widehat{\delta}_{min}([q], \beta), a), \qquad \text{def. } \widehat{\delta}_{min} \\
\delta_{min}(\widehat{\delta}_{min}([q], \beta), a) & = \delta_{min}([\widehat{\delta}(q, \beta)], a), \qquad \text{HI} \\
\delta_{min}([\widehat{\delta}(q, \beta)], a) & = \left[\delta(\widehat{\delta}(q, \beta), a)\right], \qquad \text{def. } \delta_{min} \\
\left[\delta(\widehat{\delta}(q, \beta), a)\right] & = \left[\widehat{\delta}(q, \beta a)\right], \qquad \text{def. } \widehat{\delta}
\end{array}$$

□

Notación: $\Sigma^{\leq k}$ es el conjunto de todas las palabras sobre Σ de longitud menor o igual que k.

Definición (Indistinguibilidad de orden k: $\overset{k}{\equiv}$)

Sea $M =< Q, \Sigma, \delta, q_0, F >$ un AFD sin estados inaccesibles, y sea k un entero no negativo. Definimos $\overset{k}{\equiv} \subseteq Q \times Q$,

$$p \overset{k}{\equiv} q \text{ si y solo si } \forall \alpha \in \Sigma^{\leq k} \left(\widehat{\delta}\,(p, \alpha) \in F \Leftrightarrow \widehat{\delta}\,(q, \alpha) \in F \right).$$

Ejemplo. Dado AFD $M = (Q, \Sigma, \delta, q_0, F)$ con $\Sigma = \{a, b\}$,

$$(Q/\overset{0}{\equiv}) = \{F, Q \setminus F\}$$

$$(Q/\overset{1}{\equiv}) = \left\{ \begin{array}{l} F \cap \{p : \delta(p,a) \in F \wedge \delta(p,b) \in F\}, \\ F \cap \{p : \delta(p,a) \in F \wedge \delta(p,b) \notin F\}, \\ F \cap \{p : \delta(p,a) \notin F \wedge \delta(p,b) \in F\}, \\ F \cap \{p : \delta(p,a) \notin F \wedge \delta(p,b) \notin F\}, \\ (Q \setminus F) \cap \{p : \delta(p,a) \in F \wedge \delta(p,b) \in F\}, \\ (Q \setminus F) \cap \{p : \delta(p,a) \in F \wedge \delta(p,b) \notin F\}, \\ (Q \setminus F) \cap \{p : \delta(p,a) \notin F \wedge \delta(p,b) \in F\}, \\ (Q \setminus F) \cap \{p : \delta(p,a) \notin F \wedge \delta(p,b) \notin F\} \end{array} \right\}$$

Teorema (Propiedades de la indistinguibilidad de orden k)

1. $\overset{k}{\equiv}$ *es una relación de equivalencia*
2. $\overset{k+1}{\equiv} \subseteq \overset{k}{\equiv}$
3. *Si* $Q - F \neq \emptyset$ *y* $F \neq \emptyset$ *entonces* $\left(Q/ \overset{0}{\equiv}\right) = \{Q - F, F\}$.
4. $p \overset{k+1}{\equiv} r \Leftrightarrow \forall a \in \Sigma, \delta\left(p, a\right) \overset{k}{\equiv} \delta\left(r, a\right)$.
5. *Si* $\left(\overset{k+1}{\equiv} = \overset{k}{\equiv}\right)$ *entonces* $\forall n \geq 0, \left(\overset{k+n}{\equiv} = \overset{k}{\equiv}\right)$

Demostración

1. $\overset{k}{\equiv}$ es una relación de equivalencia: ejercicio.

2. $\overset{k+1}{\equiv} \subseteq \overset{k}{\equiv}$. Supongamos $p \overset{k+1}{\equiv} q$.
   Si $\forall \alpha \in \Sigma^*(|\alpha| \leq k+1)$ entonces $\left(\widehat{\delta}\,(p,\alpha) \in F \Leftrightarrow \widehat{\delta}\,(q,\alpha) \in F\right)$.
   Por lo tanto,
   Si $\forall \alpha \in \Sigma^*, \quad (|\alpha| \leq k)$ entonces $\left(\widehat{\delta}\,(p,\alpha) \in F \Leftrightarrow \widehat{\delta}\,(q,\alpha) \in F\right)$.
   Por definición de $\overset{k}{\equiv}$, $p \overset{k}{\equiv} q$.

3. Supongamos $Q - F \neq \emptyset$ y $F \neq \emptyset$.
Debemos ver que $\left(Q/ \overset{0}{\equiv}\right) = \{Q - F, F\}$.

$$\begin{aligned}
\left(Q/ \overset{0}{\equiv}\right) &= \Big\{\{q \in Q : \widehat{\delta}(q, \lambda) \notin F\}, \{q \in Q : \widehat{\delta}(q, \lambda) \in F\}\Big\} \\
&= \{\{q \in Q : q \notin F\}, \{q \in Q : q \in F\}\} \\
&= \{Q - F, F\}.
\end{aligned}$$

4. Debemos probar $p \overset{k+1}{\equiv} r$ si y solo si $\forall a \in \Sigma, \delta(p,a) \overset{k}{\equiv} \delta(r,a)$.

$\Rightarrow$) Supongamos $p \overset{k+1}{\equiv} r$ pero no es cierto que $\forall a \in \Sigma, \delta(p,a) \overset{k}{\equiv} \delta(r,a)$.
Entonces $\exists a \in \Sigma, \exists \alpha \in \Sigma^*$, $(|\alpha| \leq k) \wedge$

$$\left(\widehat{\delta}\left(\delta\left(p,a\right),\alpha\right) \in F\right) \wedge \left(\widehat{\delta}\left(\delta\left(r,a\right),\alpha\right) \notin F\right).$$ O viceversa.

Por lo tanto $\left(\widehat{\delta}\left(p,a\alpha\right) \in F\right) \wedge \left(\widehat{\delta}\left(r,a\alpha\right) \notin F\right)$. O viceversa.
Entonces, $p \overset{k+1}{\not\equiv} r$, ya que $|a\alpha| \leq k+1$, contradiciendo $p \overset{k+1}{\equiv} r$.

$\Leftarrow$) Demostramos el contrapositivo. Supongamos que $p \overset{k+1}{\not\equiv} q$. Entonces
$\exists \alpha = a\alpha'$, con $|\alpha| \leq k+1$ que distingue p de q, o sea que

$$\left(\widehat{\delta}\left(\delta\left(p,a\right),\alpha'\right) \in F\right) \wedge \left(\widehat{\delta}\left(\delta\left(q,a\right),\alpha'\right) \notin F\right).$$ O viceversa

Por lo tanto $\delta(p,a) \overset{k}{\not\equiv} \delta(q,a)$.

5. Debemos probar que si $\left(\overset{k+1}{\equiv} = \overset{k}{\equiv}\right)$ entonces $\forall n \geq 0$, $\left(\overset{k+n}{\equiv} = \overset{k}{\equiv}\right)$:

Inducción en n.

Caso base: $n = 0$. Trivial ya que $\overset{k+0}{\equiv} = \overset{k}{\equiv}$.

Caso inductivo. Suponemos cierto para n con $n \geq 0$:

Si $\left(\overset{k+1}{\equiv} = \overset{k}{\equiv}\right)$ entonces $\left(\overset{k+n}{\equiv} = \overset{k}{\equiv}\right)$.

Debemos probar que

Si $\left(\overset{k+1}{\equiv} = \overset{k}{\equiv}\right)$ entonces $\left(\overset{k+n+1}{\equiv} = \overset{k}{\equiv}\right)$.

Supongamos $\left(\overset{k+1}{\equiv} = \overset{k}{\equiv}\right)$. Veamos $\forall q, r \in Q$, $\left(q \overset{k+n+1}{\equiv} r \Leftrightarrow q \overset{k}{\equiv} r\right)$.

$$
\begin{aligned}
q \overset{k+n+1}{\equiv} r &\Leftrightarrow \left(\forall a \in \Sigma,\ \delta(q,a) \overset{k+n}{\equiv} \delta(r,a)\right) && \text{por definición } \overset{k+n}{\equiv} \\
&\Leftrightarrow \left(\forall a \in \Sigma,\ \delta(q,a) \overset{k}{\equiv} \delta(r,a)\right) && \text{por HI} \\
&\Leftrightarrow q \overset{k+1}{\equiv} r && \text{por definición } \overset{k+1}{\equiv}. \\
&\Leftrightarrow q \overset{k}{\equiv} r && \text{por suposición } \left(\overset{k+1}{\equiv} = \overset{k}{\equiv}\right).
\end{aligned}
$$

Recordemos

Sea $M =< Q, \Sigma, \delta, q_0, F >$ un AFD sin estados inaccesibles.
El AFD mínimo equivalente $M_{min} =< Q_{min}, \Sigma, \delta_{min}, q_{min_0}, F_{min} >$ es

$$\begin{aligned} Q_{min} &= (Q \,/ \equiv) \text{ (las clases de equivalencia de } \equiv \text{)} \\ \delta_{min}([q], a) &= [\delta(q, a)] \\ q_{min_0} &= [q_0] \\ F_{min} &= \{[q] \in Q_{min} : q \in F\} \end{aligned}$$

Algoritmo de minimización de un AFD (algoritmo de Moore)

`Input` `AFD` $M = (Q, \Sigma, \delta, q_0, F)$
`Output` $(Q/\equiv)$

$P := \{Q\}$
$i := 0$
mientras $\Big(P \neq \{X/\overset{i}{\equiv} \;:\; X \in P\}\Big)$

$\qquad P := \{X/\overset{i}{\equiv} \;:\; X \in P\}$
$\qquad i := i + 1$

`return` P

Definimos $M_{min} = (Q_{min}, \Sigma, \delta_{min}, q_{min_0}, F_{min})$,
$Q_{min} = Q/\equiv$,
$\delta_{min}([q], a) = [\delta(q, a)]$,
$q_{min_0} = [q_0]$,
$F_{min} = \{[q] : q \in F\}$

Teorema

Sea M_{min} un autómata mínimo. Entonces, cualquier AFD M' que reconozca el mismo lenguaje tiene al menos tantos estados como M_{min}. Es decir,

$$\forall M', \left(\textit{Si } \mathcal{L}\left(M_{min}\right) = \mathcal{L}\left(M'\right) \textit{ entonces } |Q_{min}| \leq |Q'|\right)$$

Para demostrarlo usaremos el siguiente lema.

Lema

Sean AFDs $M =< Q, \Sigma, \delta, q_0, F >$ *y* $M' =< Q', \Sigma, \delta', q_0', F' >$ *y* M *no posee estados inaccesibles. Si todo par de cadenas que conducen desde el estado incial a estados diferentes en* M *conducen también a estados diferentes en* M'*, entonces*

$$|Q| \leq |Q'| \, .$$

Definiremos una función inyectiva f de Q en Q'.

Consideremos primero la función $g : Q \to \Sigma^*$ definida por $g(q)$ es la cadena que da el camino mínimo desde q_0 a q;

$$g\left(q\right) = \min_{long-lex} \left\{ \alpha \in \Sigma^* : \widehat{\delta}\left(q_0, \alpha\right) = q \right\}$$

Ahora sí, sea $f : Q \to Q'$ tal que $f(q)$ es el estado al que llego en M' empezando en q_0' usando el camino mínimo en M dado por $g(q)$,

$$f\left(q\right) = \widehat{\delta'}\left(q_0', g\left(q\right)\right).$$

Veamos que f es inyectiva. Sean $p, q \in Q$ diferentes.
Entonces, $p = \widehat{\delta}\left(q_0, g\left(p\right)\right) \neq \widehat{\delta}\left(q_0, g\left(q\right)\right) = q$.
Y, supusimos acerca de M' ,
$\forall \alpha, \beta \in \Sigma^*, \Big(\text{ Si } \widehat{\delta}\left(q_0, \alpha\right) \neq \widehat{\delta}\left(q_0, \beta\right) \text{ entonces } \widehat{\delta'}\left(q_0', \alpha\right) \neq \widehat{\delta'}\left(q_0', \beta\right) \Big).$
Entonces, $\widehat{\delta'}\left(q_0', g\left(p\right)\right) \neq \widehat{\delta'}\left(q_0', g\left(q\right)\right).$
Usando la definición de f, tenemos $f(p) \neq f(q)$.
Concluímos $f : Q \to Q'$ es inyectiva, y por lo tanto $|Q| \leq |Q'|$.
$\square$

Demostración del Teorema.

Por el absurdo.
Supongamos que $\exists M'$ tal que $\mathcal{L}(M_{min}) = \mathcal{L}(M')$ y $|Q_{min}| > |Q'|$.
El contrareciproco del lema anterior dice:

> Si $|Q_{min}| > |Q'|$ entonces hay dos cadenas que conducen a estados diferentes en M_{min} pero al mismo estado en M'.

Apliquémoslo. Entonces existen $\alpha, \beta \in \Sigma^*$, $\alpha \neq \beta$ tales que

$$\left(\widehat{\delta}_{min}\left(q_0, \alpha\right) \neq \widehat{\delta}_{min}\left(q_0, \beta\right)\right) \wedge \left(\widehat{\delta'}\left(q_0', \alpha\right) = \widehat{\delta'}\left(q_0', \beta\right)\right)$$

Dado que $\widehat{\delta}_{min}\left(q_0, \alpha\right)$ y $\widehat{\delta}_{min}\left(q_0, \beta\right)$ son distinguibles por pertenecer al autómata M_{min} hay una cadena $\gamma \in \Sigma^*$

$$\widehat{\delta}_{min}\left(q_0, \alpha\gamma\right) \in F \wedge \widehat{\delta}_{min}\left(q_0, \beta\gamma\right) \notin F.$$

Entonces, $\alpha\gamma \in \mathcal{L}\left(M_{min}\right) \Leftrightarrow \beta\gamma \notin \mathcal{L}\left(M_{min}\right)$.
Por otro lado, como $\widehat{\delta'}\left(q_0', \alpha\right) = \widehat{\delta'}\left(q_0', \beta\right)$,

$$\widehat{\delta'}\left(q_0', \alpha\gamma\right) \in F \wedge \widehat{\delta'}\left(q_0', \beta\gamma\right) \in F, \text{ o ambos } \notin F,$$

Entonces, $\alpha\gamma \in \mathcal{L}\left(M'\right) \Leftrightarrow \beta\gamma \in \mathcal{L}\left(M'\right)$.
Por lo tanto $\mathcal{L}\left(M_{min}\right) \neq \mathcal{L}\left(M'\right)$, lo contradiciendo hipótesis $\mathcal{L}\left(M'\right) = \mathcal{L}\left(M_{min}\right)$. Concluimos $|Q_{min}| \leq |Q'|$. □

Algoritmos de Minimización de AFD

AFD $\langle Q, \Sigma, \delta, q, F \rangle$, con $|Q| = n$ y $|\Sigma| = s$.

Complejidad tiempo peor caso:

Hopcroft (1971)	$O(ns \ \log n)$
Moore (1956)	$O(n^2 s)$
Brzozowski (1963)	$O(2^n)$

Algoritmo de minimización de Brzozowski

```
Input AFD M = (Q, Σ, δ, q_0, F) sin estados inaccesibles
Output AFD M_min

M^R := revertir M
M_D^R := determinizar M^R
(M_D^R)^R := revertir M_D^R
Output (M_D^R)^R
```

El algoritmo Brzozowski es correcto

M_D^R es la determinación de M^R, por lo tanto sus estados son conjuntos de estados de M^R. Dos estados $\mathcal{R}, \mathcal{S}$ de M_D^R difieren en al menos un estado q de M^R.

Supongamos $q \in \mathcal{R}$ y $q \notin \mathcal{S}$; entonces q aporta al menos una cadena al lenguaje aceptado desde $\mathcal{R}$, que no podría estar presente en el lenguaje aceptado desde $\mathcal{S}$ ya que esta cadena es exclusiva de q (ningún otro estado lo acepta).

Esto es válido para cada par de estados y, por lo tanto, cada estado se distingue de todos los demás. Entonces, M_D^R es un DFA con todos los estados distinguibles y alcanzables. M_D^R es mínimo para lenguaje reverso. Al invertirlo nuevamente obtenemos $(M_D^R)^R$ que es mínimo para lenguaje para el lenguaje aceptado por M. $\square$

La complejidad es exponencial en el tamaño de M.

Algoritmo de minimización de Hopcroft Sea AFD $(Q, \Sigma, \delta, q_0, F)$, donde Q no tiene inaccesibles.

```
P := {F, Q \ F};
W := {F};
while (W is not empty) do
     choose and remove a set A from W
     for each c in Σ do
          let X be the set of states for which a transition on c leads to a state in A
          for each set Y in P for which X ∩ Y is nonempty and Y \ X is nonempty do
               replace Y in P by the two sets X ∩ Y and Y \ X
               if Y is in W
                    replace Y in W by the same two sets
               else
                    if |X ∩ Y| <= |Y \ X|
                         add X ∩ Y to W
                    else
                         add Y \ X to W
          end;
     end;
end;
```

.

La complejidad peor caso es $O(n|\Sigma|\log n)$, donde $n = |Q|$.
Esta cota proviene de que cada una de las $n|\Sigma|$ transiciones participa en, a lo sumo, $O(\log n)$ pasos del algoritmo que realizan refinamiento, ya que en cada paso los conjuntos considerados de Q decrecen a la mitad de su tamaño.

Algoritmo de Hopcroft en página 161, *Introduction to Automata Theory, Languages and Computation*, J. Hopcroft, R. Motwani, J. Ullman, Second Edition, Addison Wesley, 2001.

Ejercicios

1. Dar un autómata AFD tal que $Q/\overset{2}{\equiv}$ sea distinto de $Q/\overset{3}{\equiv}$.
2. Sea un AFD $M = (Q, \Sigma, \delta, q_0, F)$. Mostrar que para todo entero $k \geq 0$,
$$((Q/\overset{k}{\equiv}))/\overset{k}{\equiv}) \text{ es igual a } Q/\overset{k}{\equiv}.$$
3. Consideremos el algoritmo de minimización de autómatas de Moore. Reemplacemos la instrucción $i = i + 1$ por la instrucción $i = i + 2$. ¿Terminará la ejecución del ciclo? En caso de que sí, ¿Con qué resultado?
4. Un transductor es un autómata con entrada y con salida (también llamado "Mealy machine"). Formalmente un transductor finito determinístico es una 7-upla $M = (Q, \Sigma, \Delta, \delta, \rho, q_0, F)$ donde Q, Σ, δ y q_0 son como en un DFA, Δ es el alfabeto de salida y ρ es la función que mapea $Q \times \Sigma$ en Δ.

   Es decir $\rho(q, a)$ es salida de la transición del estado q con entrada a.

   La salida de M con entrada $a_1 \ldots a_n$ es $\rho(q_0, a_1)\rho(q_1, a_2) \ldots \rho(q_{n-1}, a_n)$ donde q_0, $q_1 \ldots q_{n-1}$ es la secuencia de estados tal que $\delta(q_{i-1}, a) = q_i$ para $i = 1, \ldots, n$.

   ¿Cómo es el algoritmo de minimización para transductores?