

Autómatas y Lenguajes Formales

Tema 5: Minimización de autómatas, propiedades de cerradura de lenguajes regulares

Dr. Favio Ezequiel Miranda Perea
favio@ciencias.unam.mx

Facultad de Ciencias UNAM¹

3 de febrero de 2020

¹Con el apoyo del proyecto PAPIME PE102117



Eliminación de Estados Inaccesibles

- Sea $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AFD. Decimos que un estado $q \in Q$ es accesible si y sólo si existe $w \in \Sigma^*$ tal que $\delta^*(q_0, w) = q$.

Eliminación de Estados Inaccesibles

- Sea $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AFD. Decimos que un estado $q \in Q$ es accesible si y sólo si existe $w \in \Sigma^*$ tal que $\delta^*(q_0, w) = q$.
- Es decir, q es accesible si y sólo si el procesamiento de alguna cadena termina en el estado q .



Eliminación de Estados Inaccesibles

- Sea $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AFD. Decimos que un estado $q \in Q$ es accesible si y sólo si existe $w \in \Sigma^*$ tal que $\delta^*(q_0, w) = q$.
- Es decir, q es accesible si y sólo si el procesamiento de alguna cadena termina en el estado q .
- El conjunto de estados accesibles de un autómata M se denota $Acc(M)$.

Eliminación de Estados Inaccesibles

- Sea $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AFD. Decimos que un estado $q \in Q$ es accesible si y sólo si existe $w \in \Sigma^*$ tal que $\delta^*(q_0, w) = q$.
- Es decir, q es accesible si y sólo si el procesamiento de alguna cadena termina en el estado q .
- El conjunto de estados accesibles de un autómata M se denota $Acc(M)$.
- Si un estado no es accesible decimos que es inaccesible.



Cálculo de estados accesibles

- Es claro que el conjunto $Acc(M)$ puede construirse de manera algorítmica, por ejemplo como sigue:



Cálculo de estados accesibles

- Es claro que el conjunto $Acc(M)$ puede construirse de manera algorítmica, por ejemplo como sigue:
 - ▶ $A_N := \{q_0\}$, $A_V := \emptyset$.



Cálculo de estados accesibles

- Es claro que el conjunto $Acc(M)$ puede construirse de manera algorítmica, por ejemplo como sigue:
 - ▶ $A_N := \{q_0\}$, $A_V := \emptyset$.
 - ▶ Mientras que $A_N \neq A_V$



Cálculo de estados accesibles

- Es claro que el conjunto $Acc(M)$ puede construirse de manera algorítmica, por ejemplo como sigue:
 - ▶ $A_N := \{q_0\}$, $A_V := \emptyset$.
 - ▶ Mientras que $A_N \neq A_V$
 - ★ $A_V := A_N$



Cálculo de estados accesibles

- Es claro que el conjunto $Acc(M)$ puede construirse de manera algorítmica, por ejemplo como sigue:
 - ▶ $A_N := \{q_0\}$, $A_V := \emptyset$.
 - ▶ Mientras que $A_N \neq A_V$
 - ★ $A_V := A_N$
 - ★ $A_N := A_N \cup \{q \in Q \mid \delta(p, a) = q \text{ } a \in \Sigma, p \in A_N\}$



Cálculo de estados accesibles

- Es claro que el conjunto $Acc(M)$ puede construirse de manera algorítmica, por ejemplo como sigue:

- ▶ $A_N := \{q_0\}$, $A_V := \emptyset$.
- ▶ Mientras que $A_N \neq A_V$
 - ★ $A_V := A_N$
 - ★ $A_N := A_N \cup \{q \in Q \mid \delta(p, a) = q \text{ } a \in \Sigma, p \in A_N\}$
- ▶ $Acc(M) := A_N$.

Cálculo de estados accesibles

- Es claro que el conjunto $Acc(M)$ puede construirse de manera algorítmica, por ejemplo como sigue:
 - ▶ $A_N := \{q_0\}$, $A_V := \emptyset$.
 - ▶ Mientras que $A_N \neq A_V$
 - ★ $A_V := A_N$
 - ★ $A_N := A_N \cup \{q \in Q \mid \delta(p, a) = q \text{ } a \in \Sigma, p \in A_N\}$
 - ▶ $Acc(M) := A_N$.
- Los estados inaccesibles en un autómata son ínutilles y pueden ser eliminados sin afectar el lenguaje de aceptación



Eliminación de estados inaccesibles

Autómata equivalente

- Dado un $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un AFD $M' = \langle Q', \Sigma', \delta', q'_0, F' \rangle$ equivalente a M que contiene únicamente a los estados accesibles de M , es decir, $Q' = Acc(M)$ y por lo tanto no contiene estados inaccesibles.



Eliminación de estados inaccesibles

Autómata equivalente

- Dado un $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un AFD $M' = \langle Q', \Sigma', \delta', q'_0, F' \rangle$ equivalente a M que contiene únicamente a los estados accesibles de M , es decir, $Q' = Acc(M)$ y por lo tanto no contiene estados inaccesibles.
- Basta definir M' como sigue:



Eliminación de estados inaccesibles

Autómata equivalente

- Dado un $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un AFD $M' = \langle Q', \Sigma', \delta', q'_0, F' \rangle$ equivalente a M que contiene únicamente a los estados accesibles de M , es decir, $Q' = Acc(M)$ y por lo tanto no contiene estados inaccesibles.
- Basta definir M' como sigue:
 - ▶ $Q' = Acc(M)$



Eliminación de estados inaccesibles

Autómata equivalente

- Dado un $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un AFD $M' = \langle Q', \Sigma', \delta', q'_0, F' \rangle$ equivalente a M que contiene únicamente a los estados accesibles de M , es decir, $Q' = Acc(M)$ y por lo tanto no contiene estados inaccesibles.
- Basta definir M' como sigue:
 - ▶ $Q' = Acc(M)$
 - ▶ $\Sigma' = \Sigma$



Eliminación de estados inaccesibles

Autómata equivalente

- Dado un $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un AFD $M' = \langle Q', \Sigma', \delta', q'_0, F' \rangle$ equivalente a M que contiene únicamente a los estados accesibles de M , es decir, $Q' = Acc(M)$ y por lo tanto no contiene estados inaccesibles.
- Basta definir M' como sigue:
 - ▶ $Q' = Acc(M)$
 - ▶ $\Sigma' = \Sigma$
 - ▶ $\delta' = \delta|_{Q'}$



Eliminación de estados inaccesibles

Autómata equivalente

- Dado un $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un AFD $M' = \langle Q', \Sigma', \delta', q'_0, F' \rangle$ equivalente a M que contiene únicamente a los estados accesibles de M , es decir, $Q' = Acc(M)$ y por lo tanto no contiene estados inaccesibles.
- Basta definir M' como sigue:
 - ▶ $Q' = Acc(M)$
 - ▶ $\Sigma' = \Sigma$
 - ▶ $\delta' = \delta|_{Q'}$
 - ▶ $q'_0 = q_0$



Eliminación de estados inaccesibles

Autómata equivalente

- Dado un $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un AFD $M' = \langle Q', \Sigma', \delta', q'_0, F' \rangle$ equivalente a M que contiene únicamente a los estados accesibles de M , es decir, $Q' = Acc(M)$ y por lo tanto no contiene estados inaccesibles.
- Basta definir M' como sigue:
 - ▶ $Q' = Acc(M)$
 - ▶ $\Sigma' = \Sigma$
 - ▶ $\delta' = \delta|_{Q'}$
 - ▶ $q'_0 = q_0$
 - ▶ $F' = F \cap Q'$



Eliminación de estados inaccesibles

Autómata equivalente

- Dado un $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un AFD $M' = \langle Q', \Sigma', \delta', q'_0, F' \rangle$ equivalente a M que contiene únicamente a los estados accesibles de M , es decir, $Q' = Acc(M)$ y por lo tanto no contiene estados inaccesibles.
- Basta definir M' como sigue:
 - ▶ $Q' = Acc(M)$
 - ▶ $\Sigma' = \Sigma$
 - ▶ $\delta' = \delta|_{Q'}$
 - ▶ $q'_0 = q_0$
 - ▶ $F' = F \cap Q'$
- La prueba de la equivalencia $L(M) = L(M')$ es inmediata y se deja como ejercicio.



Eliminación de estados inaccesibles

Autómata equivalente

- Dado un $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un AFD $M' = \langle Q', \Sigma', \delta', q'_0, F' \rangle$ equivalente a M que contiene únicamente a los estados accesibles de M , es decir, $Q' = Acc(M)$ y por lo tanto no contiene estados inaccesibles.
- Basta definir M' como sigue:
 - ▶ $Q' = Acc(M)$
 - ▶ $\Sigma' = \Sigma$
 - ▶ $\delta' = \delta|_{Q'}$
 - ▶ $q'_0 = q_0$
 - ▶ $F' = F \cap Q'$
- La prueba de la equivalencia $L(M) = L(M')$ es inmediata y se deja como ejercicio.
- Debido a este resultado de ahora en adelante podemos suponer que un autómata no tiene estados inaccesibles.



Equivalencias de estados

Minimización

- Decimos que dos estados $q, q' \in Q$ de un AFD son equivalentes $q \equiv q'$ si y sólo si:

$$\forall w \in \Sigma^* (\delta^*(q, w) \in F \Leftrightarrow \delta^*(q', w) \in F)$$



Equivalencias de estados

Minimización

- Decimos que dos estados $q, q' \in Q$ de un AFD son equivalentes $q \equiv q'$ si y sólo si:

$$\forall w \in \Sigma^* (\delta^*(q, w) \in F \Leftrightarrow \delta^*(q', w) \in F)$$

- Es decir, $\delta^*(q, w), \delta^*(q', w)$ son ambos finales o ambos no finales.



\equiv es un relación de equivalencia

Minimización de AFD

La relación \equiv entre estados es una relación de equivalencia, es decir cumple lo siguiente:

- Reflexividad: $q \equiv q$.



\equiv es un relación de equivalencia

Minimización de AFD

La relación \equiv entre estados es una relación de equivalencia, es decir cumple lo siguiente:

- Reflexividad: $q \equiv q$.
- Simetria: si $q \equiv q'$ entonces $q' \equiv q$.



\equiv es un relación de equivalencia

Minimización de AFD

La relación \equiv entre estados es una relación de equivalencia, es decir cumple lo siguiente:

- Reflexividad: $q \equiv q$.
- Simetria: si $q \equiv q'$ entonces $q' \equiv q$.
- Transitividad: si $q \equiv q'$ y $q' \equiv q''$ entonces $q \equiv q''$.



\equiv es un relación de equivalencia

Minimización de AFD

La relación \equiv entre estados es una relación de equivalencia, es decir cumple lo siguiente:

- Reflexividad: $q \equiv q$.
- Simetria: si $q \equiv q'$ entonces $q' \equiv q$.
- Transitividad: si $q \equiv q'$ y $q' \equiv q''$ entonces $q \equiv q''$.



\equiv es un relación de equivalencia

Minimización de AFD

La relación \equiv entre estados es una relación de equivalencia, es decir cumple lo siguiente:

- Reflexividad: $q \equiv q$.
- Simetria: si $q \equiv q'$ entonces $q' \equiv q$.
- Transitividad: si $q \equiv q'$ y $q' \equiv q''$ entonces $q \equiv q''$.

Adicionalmente la función de transición δ es compatible con \equiv , en el siguiente sentido:

Si $q \equiv q'$ entonces $\forall a \in \Sigma (\delta(q, a) \equiv \delta(q', a))$



Partición del conjunto de estados

Minimización de AFD

La relación de equivalencia \equiv genera una **partición** del conjunto de estados dada por las clases de equivalencia de cada estado definidas como:

$$[q] := \{p \in Q \mid q \equiv p\}$$



Partición del conjunto de estados

Minimización de AFD

La relación de equivalencia \equiv genera una **partición** del conjunto de estados dada por las clases de equivalencia de cada estado definidas como:

$$[q] := \{p \in Q \mid q \equiv p\}$$

Es decir, los conjuntos $[q]$ cumplen lo siguiente:

- $\forall q \in Q ([q] \neq \emptyset)$.



Partición del conjunto de estados

Minimización de AFD

La relación de equivalencia \equiv genera una **partición** del conjunto de estados dada por las clases de equivalencia de cada estado definidas como:

$$[q] := \{p \in Q \mid q \equiv p\}$$

Es decir, los conjuntos $[q]$ cumplen lo siguiente:

- $\forall q \in Q ([q] \neq \emptyset)$.
- $\forall p, q \in Q ([q] = [p] \text{ ó } [q] \cap [p] = \emptyset)$.



Partición del conjunto de estados

Minimización de AFD

La relación de equivalencia \equiv genera una **partición** del conjunto de estados dada por las clases de equivalencia de cada estado definidas como:

$$[q] := \{p \in Q \mid q \equiv p\}$$

Es decir, los conjuntos $[q]$ cumplen lo siguiente:

- $\forall q \in Q ([q] \neq \emptyset)$.
- $\forall p, q \in Q ([q] = [p] \text{ ó } [q] \cap [p] = \emptyset)$.
- $\bigcup_{q \in Q} [q] = Q$.



El autómata cociente

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un autómata M^{min} , equivalente a M y que tiene un número mínimo de estados. Este autómata se define como $M^{min} = \langle Q_m, \Sigma, \delta_m, [q_0], F_m \rangle$ donde:

- $Q_m := Q / \equiv := \{[q] \mid q \in Q^M\}$



El autómata cociente

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un autómata M^{\min} , equivalente a M y que tiene un número mínimo de estados. Este autómata se define como $M^{\min} = \langle Q_m, \Sigma, \delta_m, [q_0], F_m \rangle$ donde:

- $Q_m := Q / \equiv := \{[q] \mid q \in Q^M\}$
- $[q_0]$ es el estado inicial.



El autómata cociente

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un autómata M^{\min} , equivalente a M y que tiene un número mínimo de estados. Este autómata se define como $M^{\min} = \langle Q_m, \Sigma, \delta_m, [q_0], F_m \rangle$ donde:

- $Q_m := Q / \equiv := \{[q] \mid q \in Q^M\}$
- $[q_0]$ es el estado inicial.
- $F_m := \{[q] \mid q \in F\}$



El autómata cociente

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un autómata M^{\min} , equivalente a M y que tiene un número mínimo de estados. Este autómata se define como $M^{\min} = \langle Q_m, \Sigma, \delta_m, [q_0], F_m \rangle$ donde:

- $Q_m := Q / \equiv := \{[q] \mid q \in Q^M\}$
- $[q_0]$ es el estado inicial.
- $F_m := \{[q] \mid q \in F\}$
- $\delta_m : Q \times \Sigma \rightarrow Q$ se define como

$$\delta_m([q], a) = [\delta(q, a)]$$



El autómata cociente

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un autómata M^{\min} , equivalente a M y que tiene un número mínimo de estados. Este autómata se define como $M^{\min} = \langle Q_m, \Sigma, \delta_m, [q_0], F_m \rangle$ donde:

- $Q_m := Q / \equiv := \{[q] \mid q \in Q^M\}$
- $[q_0]$ es el estado inicial.
- $F_m := \{[q] \mid q \in F\}$
- $\delta_m : Q \times \Sigma \rightarrow Q$ se define como

$$\delta_m([q], a) = [\delta(q, a)]$$



El autómata cociente

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ existe un autómata M^{\min} , equivalente a M y que tiene un número mínimo de estados. Este autómata se define como $M^{\min} = \langle Q_m, \Sigma, \delta_m, [q_0], F_m \rangle$ donde:

- $Q_m := Q / \equiv := \{[q] \mid q \in Q^M\}$
- $[q_0]$ es el estado inicial.
- $F_m := \{[q] \mid q \in F\}$
- $\delta_m : Q \times \Sigma \rightarrow Q$ se define como

$$\delta_m([q], a) = [\delta(q, a)]$$

M^{\min} se conoce también como el autómata cociente de M denotado también con M / \equiv .



Minimalidad de M^{min}

- Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ el autómata cociente M/\equiv es el autómata mínimo equivalente a M . Es decir, se tiene $L(M) = L(M/\equiv)$ y no existe un autómata equivalente a M con menos estados que M/\equiv .



Minimalidad de M^{min}

- Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ el autómata cociente M/\equiv es el autómata mínimo equivalente a M . Es decir, se tiene $L(M) = L(M/\equiv)$ y no existe un autómata equivalente a M con menos estados que M/\equiv .
- La equivalencia entre M y M^{min} se sigue de la siguiente propiedad:



Minimalidad de M^{min}

- Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ el autómata cociente M/\equiv es el autómata mínimo equivalente a M . Es decir, se tiene $L(M) = L(M/\equiv)$ y no existe un autómata equivalente a M con menos estados que M/\equiv .
- La equivalencia entre M y M^{min} se sigue de la siguiente propiedad:

Sean $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AFD y M^{min} su autómata cociente. Para cualesquiera $q \in Q$, $w \in \Sigma^*$ se cumple

$$\delta_m^*([q], w) = [\delta^*(q, w)]$$



Minimalidad de M^{min}

- Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ el autómata cociente M/\equiv es el autómata mínimo equivalente a M . Es decir, se tiene $L(M) = L(M/\equiv)$ y no existe un autómata equivalente a M con menos estados que M/\equiv .
- La equivalencia entre M y M^{min} se sigue de la siguiente propiedad:

Sean $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AFD y M^{min} su autómata cociente. Para cualesquiera $q \in Q$, $w \in \Sigma^*$ se cumple

$$\delta_m^*([q], w) = [\delta^*(q, w)]$$

La prueba es por inducción sobre w .



k-equivalencia

Minimización de AFD

- Definimos la relación de *k*-equivalencia para cualquier $k \in \mathbb{N}$ como sigue:

$$\forall w \in \Sigma^*, |w| \leq k (\delta^*(q, w) \in F \Leftrightarrow \delta^*(q', w) \in F)$$



k-equivalencia

Minimización de AFD

- Definimos la relación de *k*-equivalencia para cualquier $k \in \mathbb{N}$ como sigue:

$$\forall w \in \Sigma^*, |w| \leq k (\delta^*(q, w) \in F \Leftrightarrow \delta^*(q', w) \in F)$$

- Es decir, para cualquier cadena w de longitud menor o igual que k , los estados $\delta^*(q, w), \delta^*(q', w)$ son ambos finales o ambos no finales.



k -equivalencia

Minimización de AFD

- Definimos la relación de k -equivalencia para cualquier $k \in \mathbb{N}$ como sigue:

$$\forall w \in \Sigma^*, |w| \leq k (\delta^*(q, w) \in F \Leftrightarrow \delta^*(q', w) \in F)$$

- Es decir, para cualquier cadena w de longitud menor o igual que k , los estados $\delta^*(q, w), \delta^*(q', w)$ son ambos finales o ambos no finales.
- \equiv_k es una relación de equivalencia cuyas clases se denotan con $[q]_k$, es decir $[q]_k = \{p \in Q \mid q \equiv_k p\}$



Propiedades

k -equivalencia

La relación de k -equivalencia cumple las siguientes propiedades:

P1 $q \equiv q'$ si y sólo si $\forall k \in \mathbb{N}(q \equiv_k q')$.



Propiedades

k -equivalencia

La relación de k -equivalencia cumple las siguientes propiedades:

P1 $q \equiv q'$ si y sólo si $\forall k \in \mathbb{N}(q \equiv_k q')$.

P2 $q \equiv_0 q'$ si y sólo si $q, q' \in F$ ó $q, q' \in Q - F$



Propiedades

k -equivalencia

La relación de k -equivalencia cumple las siguientes propiedades:

P1 $q \equiv q'$ si y sólo si $\forall k \in \mathbb{N}(q \equiv_k q')$.

P2 $q \equiv_0 q'$ si y sólo si $q, q' \in F$ ó $q, q' \in Q - F$

P3 $[q]_0 = F$ si y sólo si $q \in F$.



Propiedades

k -equivalencia

La relación de k -equivalencia cumple las siguientes propiedades:

P1 $q \equiv q'$ si y sólo si $\forall k \in \mathbb{N} (q \equiv_k q')$.

P2 $q \equiv_0 q'$ si y sólo si $q, q' \in F$ ó $q, q' \in Q - F$

P3 $[q]_0 = F$ si y sólo si $q \in F$.

P4 Si $q \equiv_k q'$ entonces $q \equiv_{k-1} q'$



Propiedades

k -equivalencia

La relación de k -equivalencia cumple las siguientes propiedades:

P1 $q \equiv q'$ si y sólo si $\forall k \in \mathbb{N} (q \equiv_k q')$.

P2 $q \equiv_0 q'$ si y sólo si $q, q' \in F$ ó $q, q' \in Q - F$

P3 $[q]_0 = F$ si y sólo si $q \in F$.

P4 Si $q \equiv_k q'$ entonces $q \equiv_{k-1} q'$

P5 $[q]_k \subseteq [q]_{k-1}$



Propiedades

k -equivalencia

P6 Si $q \equiv_k q'$ entonces $\forall a \in \Sigma (\delta(q, a) \equiv_{k-1} \delta(q', a))$



Propiedades

k -equivalencia

P6 Si $q \equiv_k q'$ entonces $\forall a \in \Sigma (\delta(q, a) \equiv_{k-1} \delta(q', a))$

P7 $q \equiv_k q'$ si y sólo si $q \equiv_{k-1} q'$ y $\forall a \in \Sigma (\delta(q, a) \equiv_{k-1} \delta(q', a))$



Propiedades

k -equivalencia

P6 Si $q \equiv_k q'$ entonces $\forall a \in \Sigma (\delta(q, a) \equiv_{k-1} \delta(q', a))$

P7 $q \equiv_k q'$ si y sólo si $q \equiv_{k-1} q'$ y $\forall a \in \Sigma (\delta(q, a) \equiv_{k-1} \delta(q', a))$

P8 Sea $P_k = \{[q]_k \mid q \in Q\}$ la partición dada por la relación \equiv_k para cualquier $k \in \mathbb{N}$.

Si $P_k = P_{k-1}$ para alguna k entonces $P_k = P_m$ para toda $m \geq k$.



Construcción del autómata mínimo

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ el AFD mínimo asociado puede construirse como sigue:

- $Q := Q - \text{estados inaccesibles desde } q_0.$



Construcción del autómata mínimo

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ el AFD mínimo asociado puede construirse como sigue:

- $Q := Q - \text{estados inaccesibles desde } q_0.$
- $k := 0$ Construir la partición $P_0 = \{F, Q - F\}.$



Construcción del autómata mínimo

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ el AFD mínimo asociado puede construirse como sigue:

- $Q := Q - \text{estados inaccesibles desde } q_0.$
- $k := 0$ Construir la partición $P_0 = \{F, Q - F\}.$

Repetir

Construcción del autómata mínimo

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ el AFD mínimo asociado puede construirse como sigue:

- $Q := Q - \text{estados inaccesibles desde } q_0.$
- $k := 0$ Construir la partición $P_0 = \{F, Q - F\}.$

Repetir

- ▶ $k := k + 1$



Construcción del autómata mínimo

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ el AFD mínimo asociado puede construirse como sigue:

- $Q := Q - \text{estados inaccesibles desde } q_0.$
- $k := 0$ Construir la partición $P_0 = \{F, Q - F\}.$

Repetir

- ▶ $k := k + 1$
- ▶ Construir P_k a partir de P_{k-1} manteniendo a dos estados q, q' en la misma clase si y sólo si para toda $a \in \Sigma$, los estados $\delta(q, a)$ y $\delta(q', a)$ estaban en la misma clase en P_{k-1} .



Construcción del autómata mínimo

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ el AFD mínimo asociado puede construirse como sigue:

- $Q := Q - \text{estados inaccesibles desde } q_0.$
- $k := 0$ Construir la partición $P_0 = \{F, Q - F\}.$

Repetir

- ▶ $k := k + 1$
- ▶ Construir P_k a partir de P_{k-1} manteniendo a dos estados q, q' en la misma clase si y sólo si para toda $a \in \Sigma$, los estados $\delta(q, a)$ y $\delta(q', a)$ estaban en la misma clase en P_{k-1} .

hasta que $P_k = P_{k-1}.$



Construcción del autómata mínimo

Minimización de AFD

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ el AFD mínimo asociado puede construirse como sigue:

- $Q := Q - \text{estados inaccesibles desde } q_0.$
- $k := 0$ Construir la partición $P_0 = \{F, Q - F\}.$

Repetir

- ▶ $k := k + 1$
- ▶ Construir P_k a partir de P_{k-1} manteniendo a dos estados q, q' en la misma clase si y sólo si para toda $a \in \Sigma$, los estados $\delta(q, a)$ y $\delta(q', a)$ estaban en la misma clase en P_{k-1} .

hasta que $P_k = P_{k-1}$.

- En tal caso P_k es la partición generada por \equiv ,
 $P_k = Q / \equiv = \{[q] \mid q \in Q\}.$



Correctud del algoritmo de minimización

- La correctud del algoritmo anterior es consecuencia de la siguiente propiedad:



Correctud del algoritmo de minimización

- La correctud del algoritmo anterior es consecuencia de la siguiente propiedad:

Si M es un AFD entonces la sucesión de particiones $P_0, P_1, \dots, P_k, \dots$ generadas por las clases de k -equivalencia de estados se estaciona,



Correctud del algoritmo de minimización

- La correctud del algoritmo anterior es consecuencia de la siguiente propiedad:

Si M es un AFD entonces la sucesión de particiones $P_0, P_1, \dots, P_k, \dots$ generadas por las clases de k -equivalencia de estados se estaciona,
es decir, existe un $n \in \mathbb{N}$ tal que $\forall k \geq n, P_k = P_n$.



Correctud del algoritmo de minimización

- La correctud del algoritmo anterior es consecuencia de la siguiente propiedad:

Si M es un AFD entonces la sucesión de particiones $P_0, P_1, \dots, P_k, \dots$ generadas por las clases de k -equivalencia de estados se estaciona,

es decir, existe un $n \in \mathbb{N}$ tal que $\forall k \geq n, P_k = P_n$.

Más aún $n \leq |Q|$, es decir, n es a lo más el número de estados de M .



Propiedades de cerradura

Lenguajes regulares

Las propiedades de cerradura nos permiten construir nuevos lenguajes regulares a partir de lenguajes ya conocidos por medio de algunas operaciones entre lenguajes.



Propiedades de cerradura

Lenguajes regulares

Las propiedades de cerradura nos permiten construir nuevos lenguajes regulares a partir de lenguajes ya conocidos por medio de algunas operaciones entre lenguajes.

Si L, M son lenguajes regulares entonces:

- $L \cup M$ es regular.



Propiedades de cerradura

Lenguajes regulares

Las propiedades de cerradura nos permiten construir nuevos lenguajes regulares a partir de lenguajes ya conocidos por medio de algunas operaciones entre lenguajes.

Si L, M son lenguajes regulares entonces:

- $L \cup M$ es regular.
- LM es regular.



Propiedades de cerradura

Lenguajes regulares

Las propiedades de cerradura nos permiten construir nuevos lenguajes regulares a partir de lenguajes ya conocidos por medio de algunas operaciones entre lenguajes.

Si L, M son lenguajes regulares entonces:

- $L \cup M$ es regular.
- LM es regular.
- L^* es regular.



Propiedades de cerradura

Lenguajes regulares

Las propiedades de cerradura nos permiten construir nuevos lenguajes regulares a partir de lenguajes ya conocidos por medio de algunas operaciones entre lenguajes.

Si L, M son lenguajes regulares entonces:

- $L \cup M$ es regular.
- LM es regular.
- L^* es regular.
- L^+ es regular.



Propiedades de cerradura

Lenguajes regulares

Las propiedades de cerradura nos permiten construir nuevos lenguajes regulares a partir de lenguajes ya conocidos por medio de algunas operaciones entre lenguajes.

Si L, M son lenguajes regulares entonces:

- $L \cup M$ es regular.
- LM es regular.
- L^* es regular.
- L^+ es regular.
- \bar{L} es regular



Propiedades de cerradura

Lenguajes regulares

Las propiedades de cerradura nos permiten construir nuevos lenguajes regulares a partir de lenguajes ya conocidos por medio de algunas operaciones entre lenguajes.

Si L, M son lenguajes regulares entonces:

- $L \cup M$ es regular.
- LM es regular.
- L^* es regular.
- L^+ es regular.
- \bar{L} es regular
- $L \cap M$ es regular.



Propiedades de cerradura

Lenguajes regulares

Las propiedades de cerradura nos permiten construir nuevos lenguajes regulares a partir de lenguajes ya conocidos por medio de algunas operaciones entre lenguajes.

Si L, M son lenguajes regulares entonces:

- $L \cup M$ es regular.
- LM es regular.
- L^* es regular.
- L^+ es regular.
- \bar{L} es regular
- $L \cap M$ es regular.
- $L - M$ es regular.



$$L = \{a^i b^j \mid i \neq j\}$$

Lenguajes no-regulares

- Supongamos que L es regular.



$$L = \{a^i b^j \mid i \neq j\}$$

Lenguajes no-regulares

- Supongamos que L es regular.
- a^*b^* claramente es regular.



$$L = \{a^i b^j \mid i \neq j\}$$

Lenguajes no-regulares

- Supongamos que L es regular.
- a^*b^* claramente es regular.
- Por propiedades de cerradura $a^*b^* - L$ también es regular.



$$L = \{a^i b^j \mid i \neq j\}$$

Lenguajes no-regulares

- Supongamos que L es regular.
- a^*b^* claramente es regular.
- Por propiedades de cerradura $a^*b^* - L$ también es regular.
- Pero $a^*b^* - L = \{a^i b^i \mid i \in \mathbb{N}\}$ que no es regular.



$$L = \{a^i b^j \mid i \neq j\}$$

Lenguajes no-regulares

- Supongamos que L es regular.
- a^*b^* claramente es regular.
- Por propiedades de cerradura $a^*b^* - L$ también es regular.
- Pero $a^*b^* - L = \{a^i b^i \mid i \in \mathbb{N}\}$ que no es regular.
- Por lo tanto L no puede ser regular.



$$L = \{wb^n \mid |w| = n, n \geq 1\}$$

Lenguajes no-regulares

- Supongamos que L es regular.



$$L = \{wb^n \mid |w| = n, n \geq 1\}$$

Lenguajes no-regulares

- Supongamos que L es regular.
- a^*b^* claramente es regular.



$$L = \{wb^n \mid |w| = n, n \geq 1\}$$

Lenguajes no-regulares

- Supongamos que L es regular.
- a^*b^* claramente es regular.
- Por propiedades de cerradura $L \cap a^*b^*$ también es regular.



$$L = \{wb^n \mid |w| = n, n \geq 1\}$$

Lenguajes no-regulares

- Supongamos que L es regular.
- a^*b^* claramente es regular.
- Por propiedades de cerradura $L \cap a^*b^*$ también es regular.
- Pero $L \cap a^*b^* = \{a^i b^i \mid i \in \mathbb{N}\}$ que no es regular.



$$L = \{wb^n \mid |w| = n, n \geq 1\}$$

Lenguajes no-regulares

- Supongamos que L es regular.
- a^*b^* claramente es regular.
- Por propiedades de cerradura $L \cap a^*b^*$ también es regular.
- Pero $L \cap a^*b^* = \{a^i b^i \mid i \in \mathbb{N}\}$ que no es regular.
- Por lo tanto L no puede ser regular.

