Tema 3: Propiedades de los Conjuntos Regulares

Serafín Moral

Universidad de Granada

Octubre, 2019

Contenido

- Lema de bombeo y aplicaciones:
  - Demostrar que un lenguaje no es regular.
- Operaciones con conjuntos regulares: complementario, intersección, diferencia, homomorfismos.
- Algoritmos para autómatas:
  - Lenguaje vacío-no vacío
  - Lenguaje finito-infinito
  - Igualdad de lenguajes de dos autómatas
- Minimización de autómatas: estados indistinguibles.

Lema de Bombeo

Sea L un conjunto regular, entonces *existe* un $n \in \mathbb{N}$ tal que $\forall z \in L$, *si* $|z| \geq n$, entonces z *se puede* expresar de la forma $z = uvw$ donde

1. $|uv| \leq n$
2. $|v| \geq 1$
3. $(\forall i \geq 0) \quad uv^i w \in L$

además n puede ser el número de estados de cualquier autómata que acepte el lenguaje L.

Comentarios

- Es un lema.
- Es útil para demostrar que un determinado lenguaje no es regular.
- Es una condición necesaria para los conjuntos regulares.
- No es una buena guía para descubrir si un lenguaje es o no regular.

Demostración

Sea $M = (Q, A, \delta, q_0, F)$ un autómata finito determinista que acepta el lenguaje L y n su número de estados.
Supongamos $z \in L$, $\quad z = a_1 a_2 \ldots a_m$ con $m \geq n$.
Como M acepta el lenguaje L y $z \in L$, tenemos que al leer z en M se llega desde el estado inicial a un estado final.

Sea $z' = a_1 a_2 \ldots a_n$ la palabra formada por los n primeros símbolos de z.

Consideremos el vector de estados $(q_{i_0}, q_{i_1}, \ldots, q_{i_n})$ donde q_{i_0} es el estado inicial, q_0, y $q_{i_j} = \delta(q_{i_{j-1}}, a_j)$

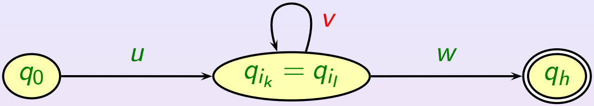
Hay n estados distintos y el vector es de longitud $n+1$: algún estado se debe de repetir. Supongamos que se repiten $q_{i_k} = q_{i_l}$.

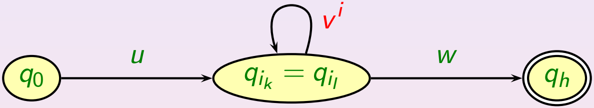
La descomposición es

$u = a_1 \ldots a_k$, $v = a_{k+1} \ldots a_l$, $w = a_{l+1} \ldots a_m$

Como al leer v pasamos por un ciclo (nos movemos de un estado a él mismo), tenemos que $\delta^*(q_0, uv^i w) = \delta^*(q_0, uvw) = \delta^*(q_0, z) \in F$. Y como M acepta L, thenemos que $uv^i w \in L$.

En el Diagrama de Transición





Pertenecen al lenguaje porque llega al mismo estado final que z.

$|uv| \leq n$ porque el ciclo se produce como máximo al leer n símbolos.

$|v| = l - k \geq 1$ porque en el vector de estados siempre se leía un símbolo para pasar al siguiente estado.

Un lenguaje no es regular

$\forall n \in \mathbb{N}$, existe una palabra $z \in L$, con $|z| \geq n$ tal que para toda descomposición

$$z = uvw$$

Si se verifica

- $|uv| \leq n$
- $|v| \geq 1$

entonces

$$\exists i \in \mathbb{N}, \text{tal que } uv^i w \notin L$$

$\{0^j1^j : j \geq 0\}$ no es regular

$\forall n \in \mathbb{N}$, existe una palabra $z \in L$, con $|z| \geq n$, $\quad z = 0^n1^n$
tal que para toda descomposición $z = uvw$. Si se verifica

- $|uv| \leq n$
- $|v| \geq 1$

entonces tenemos que $u = 0^k, v = 0^l, w = 0^{n-k-l}1^n$, con $l \geq 1$.

$$\exists i \in \mathbb{N},\ \text{tal que } uv^iw \notin L$$

Haciendo $i = 2, uv^2w = 0^k0^{2l}0^{n-k-l}1^n = 0^{n+l}1^n \notin L$.

$\{0^{j^2} : j \geq 0\}$ no es regular

$\forall n \in \mathbb{N}$, existe una palabra $z \in L$, con $|z| \geq n$, $\quad z = 0^{n^2}$
tal que para toda descomposición $z = uvw$. Si se verifica

- $|uv| \leq n$
- $|v| \geq 1$

entonces tenemos que $u = 0^k, v = 0^l, w = 0^{n^2-l-k}$, con $l \geq 1, l \leq n$.
$\exists i \in \mathbb{N}$, tal que $uv^iw \notin L$
Haciendo $i = 2, uv^2w = 0^k0^{2l}0^{n^2-l-k} = 0^{n^2+l}$
Como $(n+1)^2 - n^2 = n^2 + 2n + 1 - n^2 = 2n + 1 > n \geq l$, tenemos que $n^2 < n^2 + l < (n+1)^2$ y $uv^2w = 0^{n^2+l} \notin L$

$\{u \in \{0,1\}^* : u = u^{-1}\}$ no es regular

$\forall n \in \mathbb{N}$, existe una palabra $z \in L$, con $|z| \geq n$, $\quad z = 0^n 1^n 0^n$
tal que para toda descomposición $z = uvw$. Si se verifica

- $|uv| \leq n$
- $|v| \geq 1$

entonces tenemos que $u = 0^k, v = 0^l, w = 0^{n-k-l}1^n 0^n$, con $l \geq 1$.

$$\exists i \in \mathbb{N}, \text{ tal que } uv^i w \notin L$$

Haciendo $i = 2, uv^2w = 0^k 0^{2l} 0^{n-k-l} 1^n 0^n = 0^{n+l} 1^n 0^n \notin L$

Contraejemplo

La condición del lema de bombeo es necesaria, pero no suficiente

El lenguaje $L = \{a^l b^j c^k \ : \ (l = 0) \vee (j = k)\}$ no es regular, pero satisface la condición necesaria que aparece en el lema de bombeo

Se verifica la condición del lema de bombeo para $n = 2$.
Si $z \in L$ y $|z| \geq 2$ entonces $z = a^l b^j c^k$ con $l = 0$ ó $j = k$.
Una descomposición de z se puede obtener de la siguiente forma:

- $u = \varepsilon$
- v es el primer símbolo de z
- w es z menos su primer símbolo

Contraej. Cont.

Caben dos posibilidades:

a) $l = 0$. En este caso $z = b^j c^k$ Se verifican las tres condiciones

exigidas en el lema de bombeo.

1. $|uv| = 1 \leq n = 2$
2. $|v| = 1 \geq 1$
3. Si $i \geq 0$ entonces $uv^i w$ sigue siendo una sucesión de b seguida de una sucesión de c y por tanto una palabra de L.

Contraej. Cont.

b) $l \geq 1$. En ese caso $z = a^l b^j c^j (l \geq 1)$, y también aquí se verifican las tres condiciones:

1. $|uv| = 1 \leq n = 2$
2. $|v| = 1 \geq 1$
3. Si $i \geq 0$ entonces $uv^i w$ sigue siendo una cacesión de a seguida de una sucesión de b y otra de c, en la que la cantidad de b es igual que la cantidad de c, y por tanto, una palabra de L.

Ya conocemos las siguientes propiedades:

- *Unión:* Si L_1 y L_2 son conjuntos regulares, entonces $L_1 \cup L_2$ es regular.
- *Concatenación:* Si L_1 y L_2 son regulares, entonces $L_1 L_2$ es regular.
- *Clausura de Kleene:* Si L es regular, entonces L^* es regular.

La primera propiedad se obtiene considerando que si L_1 y L_2 son conjuntos regulares entonces tienen dos expresiones regulares, $\mathbf{r_1}$ y $\mathbf{r_2}$. Entonces $\mathbf{r_1} + \mathbf{r_2}$ es una expresión regular para $L_1 \cup L_2$, y como la unión tiene una expresión regular, entonces es regular. Análogamente se demuestran las otras dos propiedades.

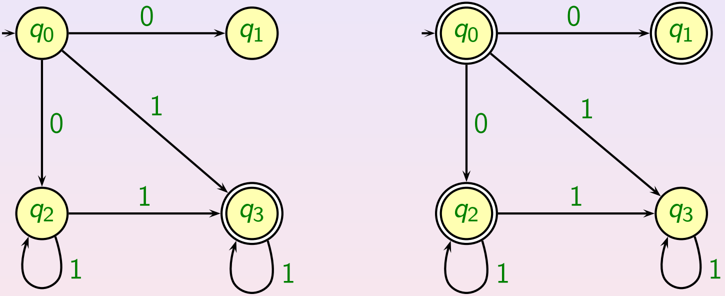
Complementario

Propiedad

Si $L \subseteq A^*$ es un lenguaje regular entonces $\overline{L} = A^* \setminus L$ es regular.

Basta con considerar que si $M = (Q, A, \delta, q_0, F)$ es un autómata finito determista que acepta el lenguaje L, entonces $M' = (Q, A, \delta, q_0, Q \setminus F)$ acepta el lenguaje complementario $A^* \setminus L$.

Esta operación no es válida en autómatas no deterministas



011 es aceptada en ambos autómatas
100 no es aceptada en ninguno de los autómatas

Intersección

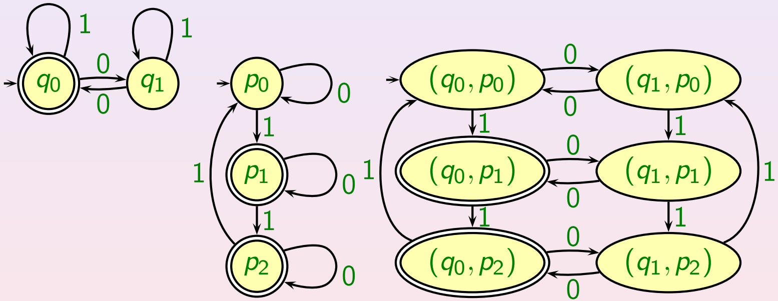
Propiedad

Si L_1 y L_2 son dos lenguajes regulares sobre el alfabeto A, entonces $L_1 \cap L_2$ es regular.

Es inmediato ya que $L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})}$.
Existe también una demostración constructiva. Si $M_1 = (Q_1, A, \delta_1, q_0^1, F_1)$ es un autómata finito determinístico que acepta L_1, y $M_2 = (Q_2, A, \delta_2, q_0^2, F_2)$ es un autómata que acepta L_2, entonces
$M = (Q_1 \times Q_2, A, \delta, (q_0^1, q_0^2), F_1 \times F_2)$
donde $\delta((q_i, q_j), a) = (\delta_1(q_i, a), \delta_2(q_j, a))$, acepta el lenguaje $L_1 \cap L_2$.
A este autómata se le llama autómata producto.

Ejemplo

Construir el autómata que acepta las palabras con un número de ceros que es múltiplo de 2 y un número de unos que no sea múltiplo de 3.

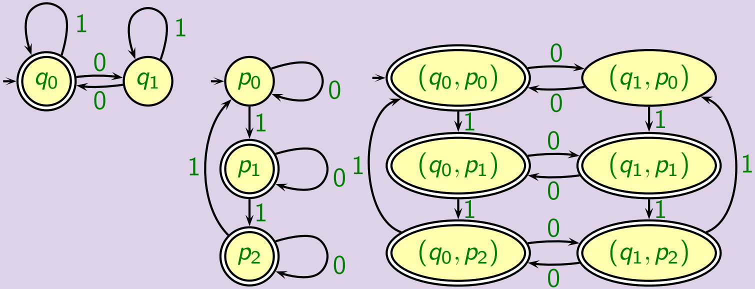


Autómata Producto para la Unión

Para construir un autómata que acepte la unión de los lenguajes aceptados por dos autómatas con el autómata producto, basta con hacer finales las parejas de estados en las que, al menos, uno de ellos es final.

Ejemplo

Construir el autómata que acepta las palabras con un número de ceros que es múltiplo de 2 o un número de unos que no sea múltiplo de 3.



Homomorfismo

Propiedad

Si A y B son alfabetos y $f : A^* \longrightarrow B^*$ un homomorfismo entre ellos, **entonces** si $L \subseteq A^*$ es un lenguaje regular, $f(L) = \{f(u) \in B^* : u \in L\}$ es también un lenguaje regular.

Basta con comprobar que se puede conseguir una expresión regular para $f(L)$ partiendo de una expresión regular para L: Basta con substituir cada símbolo, a, de L, por la correspondiente palabra $f(a)$.

Ejemplo

Si $A = \{0,1,2,3,4,5,6,7,8,9\}$ y $B = \{0,1\}$ y f es el homomorfismo dado por

$f(0) = 0000, \quad f(1) = 0001, \quad f(2) = 0010, \quad f(3) = 0011$
$f(4) = 0100, \quad f(5) = 0101, \quad f(6) = 0110, \quad f(7) = 0111$
$f(8) = 1000, \quad f(9) = 1001$

Si $L \subseteq A^*$ es el lenguaje regular dado por la expresión regular $(1+2)^*9$,
$f(L)$ viene dado por la expresión regular $(0001+0010)^*1001$.

Aplicación

Las propiedades que hemos visto nos pueden servir para demostrar que un lenguaje no es regular.
Como ejemplo, lo vamos a aplicar a demostrar que
$L = \{a^i b^j c^k : (i = 0) \vee (j = k)\}$ no es regular.
La haremos por reducción al absurdo al absurdo, considerando que L es regular y llegando a una contradicción.
Entonces $L' = L \cap L_1$ donde $L_1 = \{a^i b^j c^k : i > 0, j, k \geq 0\}$ es regular.
$L' = L \cap L_1 = \{a^i b^j c^k : (i > 0) \wedge (j = k)\}$

Aplicación: L no es regular

$L' = \{a^i b^j c^k : (i > 0) \wedge (j = k)\}$ es regular.
Si L' es regular, y f es el homomorfismo entre $\{a, b, c\}^*$ y $\{0, 1\}^*$ dado por

$$f(a) = \varepsilon, \quad f(b) = 0, \quad f(c) = 1$$

entonces $f(L')$ es regular.
Pero $f(L') = \{0^k 1^k : k \geq 0\}$ y sabemos que este conjunto no es regular.
Por la tanto hemos encontrado una contradicción y la hipótesis de que L es regular es falsa.

Propiedad

Si A y B son alfabetos y $f : A^* \longrightarrow B^*$ es un homomorfismo, entonces si $L \subseteq B^*$ es un conjunto regular, también lo es $f^{-1}(L) = \{u \in A^* : f(u) \in L\}$.

Supongamos que $M = (Q, B, \delta, q_0, F)$ que acepta el lenguaje L
Entonces el autómata $\overline{M} = (Q, A, \overline{\delta}, q_0, F)$ donde

$$\overline{\delta}(q, a) = \delta^*(q, f(a)),$$

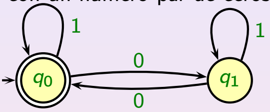
acepta el lenguaje $f^{-1}(L)$.

Ejemplo

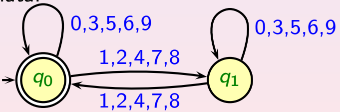
En el homomorfismo f entre $A = \{0,1,2,3,4,5,6,7,8,9\}$ y $B = \{0,1\}$ dado por:

$f(0) = 0000$, $f(1) = 0001$, $f(2) = 0010$, $f(3) = 0011$, $f(4) = 0100$,
$f(5) = 0101$, $f(6) = 0110$, $f(7) = 0111$, $f(8) = 1000$, $f(9) = 1001$

El conjunto L de las palabras con un número par de ceros es:



y el conjunto de palabras $u \in A^*$ tales que $f(u)$ tiene un número par de ceros (es decir $f^{-1}(L)$) es regular con autómata:



Aplicación

Esta propiedad también se puede usar para demostrar que un lenguaje no es regular por reducción al absurdo.

Ejemplo

Si $A = B = \{0, 1\}$ y f es el homomorfismo dado por

$$f(0) = 00, \qquad f(1) = 11$$

entonces el lenguaje $L = \{0^{2k}1^{2k} : k \geq 0\}$ no es regular, porque si lo fuese su imagen inversa, $f^{-1}(L) = \{0^k1^k : k \geq 0\}$ sería también regular y no lo es.

Cociente

Teorema

Si R es un conjunto regular y L un lenguaje culquiera, entonces el cociente de lenguajes $R/L = \{u \in A^* : \exists v \in L \text{ verificando } uv \in R\}$ es un conjunto regular.

Sea $M = (Q, A, \delta, q_0, F)$ un autómata finito determinístico que acepta el lenguaje R.
Entonces R/L es aceptado por el autómata

$$M' = (Q, A, \delta, q_0, F')$$

donde $F' = \{q \in Q : \exists y \in L \text{ tal que } \delta^*(q, y) \in F\}$

¡¡Esta demostración no es constructiva!!

Algoritmos

Algoritmo

Existe un algoritmo para determinar si el lenguaje aceptado por un autómata es vacío

Basta eliminar estados inaccesibles (mediante un recorrido por el grafo a partir del estado inicial) y comprobar si quedan estados finales.

El lenguaje aceptado por un autómata es finito o infinito: Se suponen eliminados los estados inaccesibles y se eliminan los estados de error o estados desde los que no se pueden llegar a estado finales.

Se puede hacer recorriendo el grafo en sentido contrario a los arcos y empezando en los estados finales.

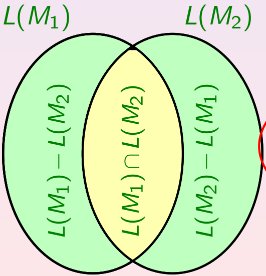
Se comprueba si en el grafo resultante quedan ciclos.

Algoritmos: Igualdad

Algoritmo

Existe un algoritmo para el problema: **Dados dos autómatas finitos M_1 y M_2 comprobar si aceptan el mismo lenguaje:**

Basta con construir el autómata que acepta el lenguaje
$(L(M_1) \setminus L(M_2)) \cup (L(M_2) \setminus L(M_1)) = (L(M_1) \cap \overline{L(M_2)}) \cup (L(M_2) \cap \overline{L(M_1)})$
Después se comprueba si el lenguaje aceptado por este autómata es vacío.



La mejor forma de hacer este autómata es con el autómata producto, haciendo finales las parejas de estados en las que un estado es final y el otro no.

Autómata Minimal

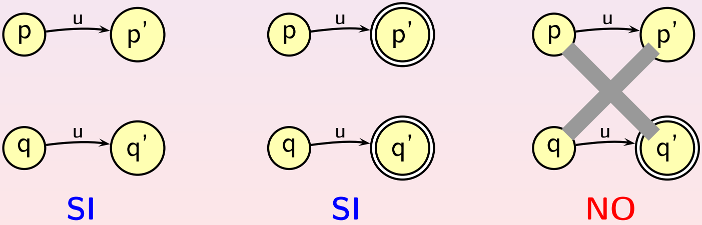
Un autómata finito determinista M se dice minimal si no exite otro autómata con menos estados que él y que acepte el mismo lenguaje.

- La primera condición para que un autómata sea minimal es que no tenga estados inaccesibles.
- Vamos a ver primero una condición para que un autómata sin estados indistinguibles sea minimal: que no tenga estados indistinguibles.
- A continuación veremos un algoritmo que, dado un autómata M calcula un autómata minimal que acepta el mismo lenguaje.

Un concepto básico para minimizar autómatas es el de estados indistinguibles.

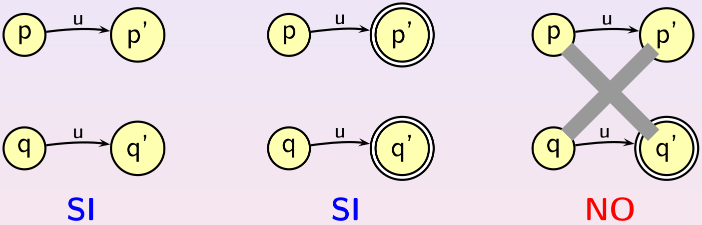
Si $M = (Q, A, \delta, q_0, F)$ es un autómata finito determinista y p, q son dos estados de Q, decimos que p y q son indistinguibles si y solo si se cumple que

$$\forall u \in A^*, \quad (\delta^*(p,u) \in F \Leftrightarrow \delta^*(q,u) \in F)$$



Parejas de Estados Distinguibles

- Si p, g son dos estados si no son indistinguibles diremos que son distinguibles



- Para que p, q sean distinguibles debe de existir $u \in A^*$ tal que en el conjuntos $\{\delta^*(p,u), \delta^*(q,u)\}$ haya un estado final y otro no final.

Relación de indistinguibilidad

Propiedades

La relación ser indistinguible de es una relación de equivalencia en el conjunto Q de estados: es reflexiva, simétrica y transitiva.

- Es reflexiva, porque para $u \in A^*$, $\delta^*(p,u) \in F \Leftrightarrow \delta^*(p,u) \in F$.
- Es simétrica: si p y q son indistinguibles $\forall u \in A^*$, $\delta^*(p,u) \in F \Leftrightarrow \delta^*(q,u) \in F$, entonces $\forall u \in A^*$, $\delta^*(q,u) \in F \Leftrightarrow \delta^*(p,u) \in F$ y q,p son indistinguibles.
- Es transitiva, porque si p,q y q,r son indistinguibles, entonces $\forall u \in A^*, \delta^*(p,u) \in F \Leftrightarrow \delta^*(q,u) \in F$ y $\delta^*(q,u) \in F \Leftrightarrow \delta^*(r,u) \in F$, por lo tanto $\forall u \in A^*, \delta^*(p,u) \in F \Leftrightarrow \delta^*(r,u) \in F$, y tenemos que p,r son indistinguibles.

Por tanto el conjunto de estados Q se puede particionar en clases de equivalencia: Sea $[q]$ la clase de equivalencia que contienes a q (y a todos los estados que son indistinguibles de q.
Si p,q son indistinguibles, entonces $[p] = [q]$.

Propiedad

Un estado final y un no final son siempre distinguibles: al leer la palabra vacía en un caso se llega a un estado final y en otro a un no final.

Propiedad

Si $a \in A$, y los estados $\delta(p, a)$ y $\delta(q, a)$ son distinguibles, entonces p y q son distinguibles.

Propiedad

Si p y q son dos estados indistinguibles, entonces para todo $a \in A$, los estados $\delta(p, a)$ y $\delta(q, a)$ son también indistinguibles.

Propiedad

Propiedad

Si $u \in A^*$, y los estados $\delta^*(p,u)$ y $\delta^*(q,u)$ son distinguibles, entonces p y q son distinguibles.

Propiedad

Si p y q son dos estados indistinguibles, entonces para todo $u \in A^*$, los estados $\delta^*(p,u)$ y $\delta^*(q,u)$ son también indistinguibles.

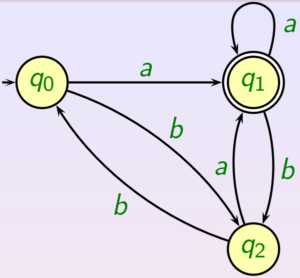
Autómata que agrupa estados indistinguibles

Si partimos de un autómata, $M = (Q, A, \delta, q_0, F)$, sea R es la relación de equivalencia de indistinguibilidad entre estados $[q]$ la clase de equivalencia asociada al estado q,
El nuevo autómata que agrupa los estados indistinguibles es:
$M_m = (Q_m, A, \delta_m, q_0^m, F_m)$ tiene los siguientes elementos,

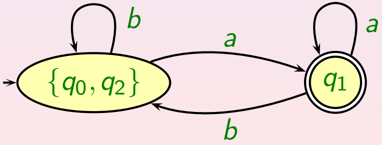
- $Q_m = \{[q] : q \text{ es accesible desde } q_0\}$
- $F_m = \{[q] : q \in F\}$
- $\delta_m([q], a) = [\delta(q, a)]$
- $q_0^m = [q_0]$

El autómata está bien definido, ya que si $[p] = [q]$, entonces p y q son indistinguibles, y por lo tanto, $\delta(p, a)$ y $\delta(q, a)$ también lo serán y $[\delta(p, a)] = [\delta(q, a)]$ y, por lo tanto, la definición de la función δ_m no depende del estado al que se le aplique δ.
Por otra parte, un estado final y otro no final son siempre distinguibles, no pueden estar en la misma clase, y F_m también está bien definido.

Ejemplo



q_0 y q_2 son indistinguibles. Autómata M_m:



Propiedad del Autómata M_m

Identidad de lenguajes

$M = (Q, A, \delta, q_0, F)$ y M_m es el autómata que agrupa los estados indistinguibles, entonces ambos aceptan el mismo lenguaje: $L(M) = L(M_n)$

La demostración es inmediata, ya que como $\delta_m([q], a) = [\delta(q, a)]$, entonces $\delta_m^*([q], u) = [\delta^*(q, u)]$.
Después de leer u desde q_0, M llegará a $\delta^*(q_0, u)$ y M_m si lee la misma palabra desde $[q_0]$ a $\delta_m^*([q_0], u) = [\delta^*(q_0, u)]$
Como $[\delta^*(q_0, u)] \in F_m$ si y solo si $\delta^*(q_0, u) \in F$, entonces $u \in L(M)$ si y solo si $u \in L(M_m)$.

Teorema

Si $M = (Q, A, \delta, q_0, F)$ y $M' = (Q', A, \delta', q'_0, F')$ aceptan el mismo lenguaje y $u, v \in A^*$ son tales que $\delta^*(q_0, u) = \delta^*(q_0, v)$, entonces $q'_i = \delta'^*(q'_0, u)$ y $q'_j = \delta'^*(q'_0, v)$ son indistinguibles en M'.

Supongamos $z \in A^*$, entonces
$\delta^*(q_0, uz) = \delta^*(\delta^*(q_0, u), z) = \delta^*(\delta^*(q_0, v), z) = \delta^*(q_0, vz)$.
Así, $uz \in L(M) \Leftrightarrow vz \in L(M)$.
Como $L(M) = L(M')$, entonces $uz \in L(M') \Leftrightarrow vz \in L(M')$.

Teorema

Si $M = (Q, A, \delta, q_0, F)$ y $M' = (Q', A, \delta', q_0', F')$ aceptan el mismo lenguaje y $u, v \in A^*$ son tales que $\delta^*(q_0, u) = \delta^*(q_0, v)$, entonces $q_i' = \delta'^*(q_0', u)$ y $q_j' = \delta'^*(q_0', v)$ son indistinguibles en M'.

$uz \in L(M') \Leftrightarrow vz \in L(M')$.
Esto es equivalente a:

$$\delta'^*(q_0', uz) \in F' \Leftrightarrow \delta'^*(q_0', uz) \in F'$$

Como $\delta'^*(q_0', uz) = \delta'^*(\delta'^*(q_0', u), z) = \delta'^*(q_i', z)$ y $\delta'^*(q_0', vz) = \delta'^*(\delta'^*(q_0', v), z) = \delta'^*(q_j', z)$, tenemos

$$\delta'^*(q_i', z) \in F' \Leftrightarrow \delta'^*(q_j', z) \in F'$$

y q_i' y q_j' son indistinguibles en M'.

Propiedad

Un autómata sin estados inaccesibles es **minimal** si y solo si no tiene una pareja de estados distintos indistinguibles.

Demostraremos que un autómata no es minimal $\Leftrightarrow$ tiene estados indistinguibles.

$\Leftarrow$ Si q_1 y q_2 son estados indistinguibles distintos de M, entonces el autómata M_m que agrupa los estados indistinguibles acepta el mismo lenguaje y tiene menos estados, ya que ahora, al menos, q_1 y q_2 se han juntado en un solo estado.
Por tanto, M no sería minimal.

Propiedad

Un autómata sin estados inaccesibles es **minimal** si y solo si no tiene estados indistinguibles.

Demostraremos que un autómata no es minimal $\Leftrightarrow$ tiene estados indistinguibles.

$\Rightarrow$ Si M no es minimal existe otro autómata M' con menos estados que acepta el mismo lenguaje.
Es fácil comprobar que exiten dos palabras $u, v \in A^*$ tales que al leerlas en M se llega a dos estados distintos p y q; y al leerlas en M' se llega al mismo estado p'.
Se puede comprobar que p y q tienen que ser indistinguibles, ya que M y M' aceptan el mismo lenguaje.

Unicidad del Autómata Minimal

Teorema de Unicidad

Si $M = (Q, A, \delta, q_0, F)$ y $M' = (Q', A, \delta', q_0', F')$ son dos autómatas minimales que aceptan el mismo lenguaje, entonces son isomorfos, es decir existe una aplicación biyectiva $f : Q \to Q'$, tal que

- $f(q_0) = q_0'$
- Si $\delta(q, a) = p$, entonces $\delta'(f(q), a) = f(p)$.
- $f(F) = F'$

Unicidad del Autómata Minimal (Demostración)

La demostración se basa en lo siguiente: para cualquier estado $q \in Q$, este estado es accesible. Por lo tanto existe una palabra $u_q \in A^*$ tal que $\delta^*(q_0, u_q) = q$. Entonces hacemos $f(q) = \delta'^*(q'_0, u_q)$.
Esta aplicación no depende de la palabra $u_q \in A^*$ tal que $\delta^*(q_0, u_q) = q$, ya que no puede ser que $\delta^*(q_0, u_q) = q = \delta^*(q_0, v_q)$ y que $\delta'^*(q'_0, u_q) \neq \delta'^*(q'_0, v_q)$, ya que entonces por un resultado anterior $\delta'^*(q'_0, u_q)$ y $\delta'^*(q'_0, v_q)$ serían indistiguibles, lo que contradice que sea minimal.

Demostración (Cont.)

La aplicación es inyectiva, ya que si $f(q) = \delta'^*(q'_0, u_q) = \delta'^*(q'_0, u_p) = f(p)$, entonces como M y M' aceptan el mismo lenguaje, p y q son indistinguibles y como M es minimal, son iguales.
La aplicación es biyectiva, ya que el número de estados es el mismo y la aplicación es inyectiva (el número de estados es también finito).

Unicidad del Autómata Minimal (Cont.)

- Como podemos elegir $u_{q_0} = \varepsilon$, tenemos que $f(q_0) = \delta'^*(q'_0, \varepsilon) = q'_0$.
- Supongamos que $\delta(q,a) = p$, y sea u_q la palabra tal que $\delta^*(q_0, u_q) = q$, está claro que $\delta^*(q_0, u_q a) = \delta(q,a) = p$. Luego, $f(q) = \delta'^*(q'_0, u_q)$ y $f(p) = \delta'^*(q'_0, u_q a) = \delta'(\delta'^*(q'_0, u_q), a) = \delta'(f(q), a)$,
- $f(F) = F'$ ya que ambos autómatas aceptan el mismo lenguaje. Si por ejemplo $q \in F$ y $f(q) \notin F$, entonces u_q es aceptada por M y no por M'.

Dos estados p, q son distinguibles de nivel n si y solo si existe una palabra $u \in A^*$ de longitud menor o igual que n tal que en el conjunto $\{\delta^*(p,u), \delta^*(q,u)\}$ hay un estado final y otro no final.

Una pareja de estados es distinguible si y solo si es distinguible a nivel n para algún $n \in \mathbb{N}$.

Las parejas distinguibles a nivel 0 son las formadas por un estado final y otro no final.

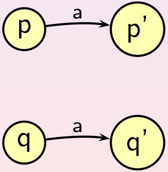
Las parejas $\{p, q\}$ distinguibles a nivel $n+1$ son las que son distinguibles a nivel n más aquellas tales que existe un $a \in A$ tal que $\{\delta(p,a), \delta(q,a)\}$ es distinguible a nivel n.

Cálculo de Parejas de Estados Indistinguible

Procedimiento básico

El conjunto de parejas de estados distinguibles, $\mathcal{D}$ se pueden calcular, introduciendo las parejas de estados distinguibles a nivel 0, y después ejecutando un algoritmo que cada vez que garantice:

- Si p, q son dos estados y la pareja $\{\delta(p,a), \delta(q,a)\}$ está en $\mathcal{D}$, entonce la pareja $\{p,q\}$ también se introduce en $\mathcal{D}$
- Aparte de las iniciales, solo se introducen parejas en $\mathcal{D}$ si ocurre lo anterior.



Si p', q' son distinguibles ($p', q' \in \mathcal{D}$) hay que hacer distinguibles a p, q ($p, q \in \mathcal{D}$)

Método alternativo

Pero es más efectivo calcular las parejas de nivel 0 y para toda pareja $\{p, q\}$ no etiquetada como distinguible, y todo $a \in A$, se calcula $\{\delta(p, a), \delta(q, a)\}$. Si esta es una pareja distinguible se pone $\{p, q\}$ como distinguible, y si no lo es se añade $\{p, q\}$ a una lista de parejas asociados a la pareja $\{\delta(p, a), \delta(q, a)\}$, para poner $\{p, q\}$ como distinguible, si alguna vez $\{\delta(p, a), \delta(q, a)\}$ resulta ser una pareja distinguible.

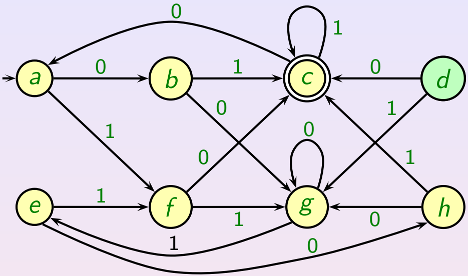
Identificación de Estados Indistinguibles

Pareja de estados $\rightarrow$ variable booleana (indistinguile) inicialmente false + lista de parejas de estados

1. Eliminar estados inaccesibles.
2. Para cada pareja de estados accesibles $\{q_i, q_j\}$
   3. Si uno de ellos es final y el otro no, hacer la variable booleana asociada igual a true.
4. Para cada pareja de estados accesibles $\{q_i, q_j\}$
   5. Para cada símbolo a del alfabeto de entrada
      6. Calcular los estados q_k y q_l a los que evoluciona el autómata desde q_i y q_j leyendo a
      7. Si $q_k \neq q_l$ entonces
         8. Si la $\{q_k, q_l\}$ está marcada entonces se marca la pareja $\{q_i, q_j\}$ y recursivamente todas las parejas en la lista asociada.
      9. Si la pareja $\{q_k, q_l\}$ no está marcada, se añade la pareja $\{q_i, q_j\}$ a la lista asociada a la pareja $\{q_k, q_l\}$

Es inmediato demostrar por inducción que este procedimiento encuentra todas las parejas de estados indistiguibles.

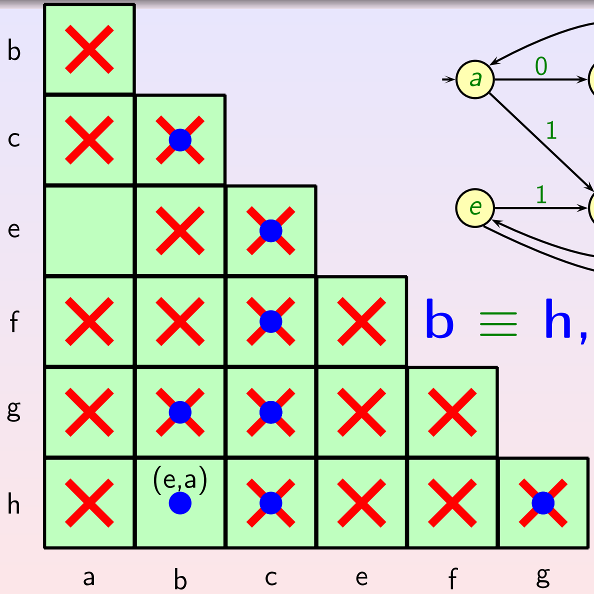
Ejemplo

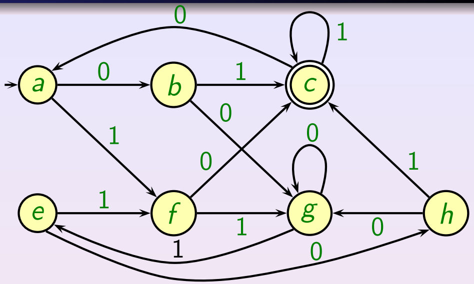


El estado d es inaccesible

y se elimina

Ejemplo





$b \equiv h, \quad a \equiv e$

	0	1

Construcción del Autómata Minimal

El autómata minimal se construye identificando los estados indistinguibles.

Si el autómata original es $M = (Q, A, \delta, q_0, F)$,

R es la relación de equivalencia de indistinguibilidad entre estados

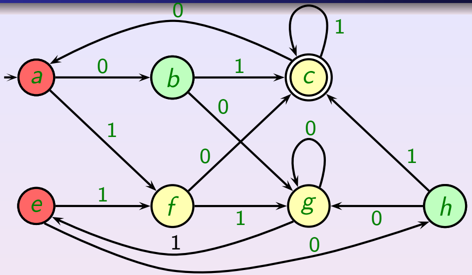
$[q]$ la clase de equivalencia asociada al estado q,

El nuevo autómata, $M_m = (Q_m, A, \delta_m, q_0^m, F_m)$ tiene los siguientes elementos,

- $Q_m = \{[q] : q \text{ es accesible desde } q_0\}$
- $F_m = \{[q] : q \in F\}$
- $\delta_m([q], a) = [\delta(q, a)]$
- $q_0^m = [q_0]$

Este autómata es minimal ya que no tiene parejas de estados indistinguibles: Si $[p]$ y $[q]$ son indistinguibles como $\delta_m^*([p], u) = [\delta^*(p, u)]$ y $\delta_m^*([q], u) = [\delta^*(q, u)]$, entonces p y q son indistinguibles en M y $[p] = [q]$, por lo que M_m no tiene parejas de estados indistiguibles distintos.

Ejemplo: Autómata Minimal



b $\equiv$ h

a $\equiv$ e