UNIVERSIDAD DE GRANADA

DECSAI

Modelos de Computación

Grado en Ingeniería Informática

Tema 3 – Propiedades de los conjuntos regulares

Manuel Pegalajar Cuéllar
manupc@ugr.es

Departamento de Ciencias de la Computación e Inteligencia Artificial
http://decsai.ugr.es

Objetivos del tema

- Entender el Lema de Bombeo y su aplicación directa a los autómatas regulares.
- Conocer las propiedades de los autómatas y los lenguajes regulares.
- Conocer algoritmos para comparación y transformaciones de autómatas regulares.
- Conocer algoritmos para la minimización de autómatas regulares.

Anotación sobre estas diapositivas:

El contenido de estas diapositivas es esquemático y representa un apoyo para las clases presenciales teóricas. No se considera un sustituto para apuntes de la asignatura.

Se recomienda al alumno completar estas diapositivas con notas/apuntes propios, tomados en clase y/o desde la bibliografía principal de la asignatura.

Propiedades de los conjuntos regulares

1. Lema de Bombeo
2. Operaciones con conjuntos regulares
3. Algoritmos para autómatas
4. Minimización de autómatas

El Lema de Bombeo

Sea L un conjunto regular, entonces existe un $n \in N$: $\forall z \in L$, si $|z|>=n$, entonces z se puede expresar de la forma $z=uvw$ donde

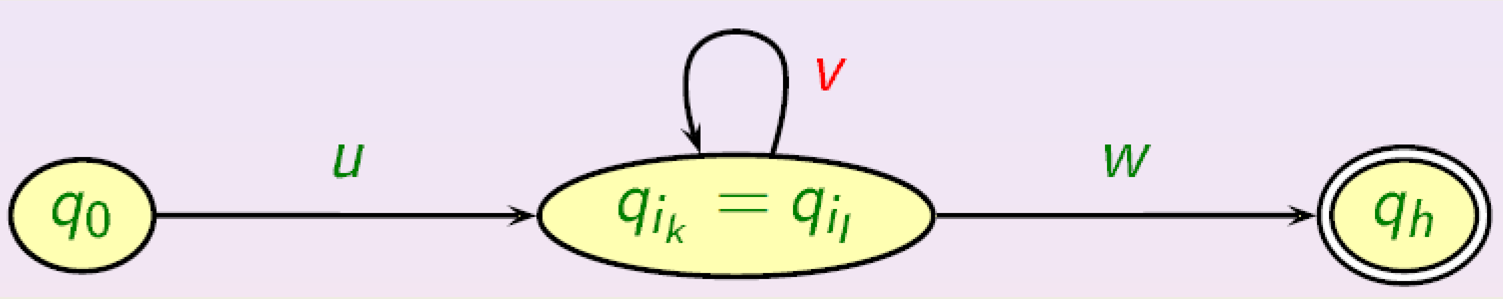
1. $|uv| <= n$
2. $|v|>=1$
3. $\forall\ i>=0\ uv^i w \in L$

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

¿Para qué sirve el Lema de Bombeo?

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

Un ejemplo



El Lema de Bombeo: Lenguajes no regulares

Un lenguaje no es regular si $\forall n \in N$, existe una palabra $z \in L$, con $|z| \geq n$, tal que para toda descomposición z=uvw. Si se verifica:

1. $|uv| <= n$
2. $|v|>=1$

Entonces tenemos que

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

Un ejemplo

Demostrar que $\{0^j 1^j : j \geq 0\}$ no es regular

$\forall n \in$ N, existe una palabra $z \in L$, con $|z| \geq n$, tal que para toda descomposición z=uvw. Si se verifica:

1. $|uv| <= n$
2. $|v| >= 1$

Entonces tenemos que

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

Un ejemplo

Demostrar que $\{0^j 1^j : j \geq 0\}$ no es regular

$\forall n \in N$, existe una palabra $z \in L$, con $|z| \geq n$,
$z = 0^n 1^n$

para toda descomposición z=uvw. Si se verifica:

1. $|uv| <= n$
2. $|v|>=1$

Entonces tenemos que $u=0^k$, $v=0^l$, $w=0^{n-k-l} 1^n$ con $l>=1$.

$\exists i \in N$, tal que $uv^iw \notin L$

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

Otro ejemplo

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

$\forall n \in \mathbb{N}$, existe una palabra $z \in L$, con $|z| \geq n$,
z= 0^{n^2}

para toda descomposición z=uvw. Si se verifica:

1. $|uv| <= n$
2. $|v|>=1$

Entonces tenemos que $u=0^k$, $v=0^l$,$w=0^{n^2-k-l}$ con l>=1.
$\exists i \in \mathbb{N}$, tal que $uv^iw \notin L$

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

Como $(n+1)^2-n^2= n^2+2n+1-n^2= 2n+1>n>=1$, entonces uv^2w no pertenece al lenguaje.

Otro ejemplo

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

$\forall n \in \mathrm{N}$, existe una palabra $z \in L$, con $|z| \geq n$,
z= $0^n 1^n 0^n$

para toda descomposición z=uvw. Si se verifica:

1. $|uv| <= n$
2. $|v|>=1$

Entonces tenemos que $u=0^k$, $v=0^l$, $w=0^{n-k-l}1^n0^n$, con l>=1.

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

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

Como $(n+1)^2-n^2= n^2+2n+1-n^2= 2n+1>n>=1$, entonces uv^2w no pertenece al lenguaje.

Consideraciones finales

¡CUIDADO!: El Lema de Bombeo es condición necesaria para lenguajes regulares, pero no suficiente.

Por ejemplo: El lenguaje $\{a^l b^j c^k : l=0 \text{ ó } j=k\}$ no es regular, pero satisface la condición del Lema de Bombeo.

UNIVERSIDAD DE GRANADA

Propiedades de los conjuntos regulares

1. Lema de Bombeo
2. Operaciones con conjuntos regulares
3. Algoritmos para autómatas
4. Minimización de autómatas

DECSAI

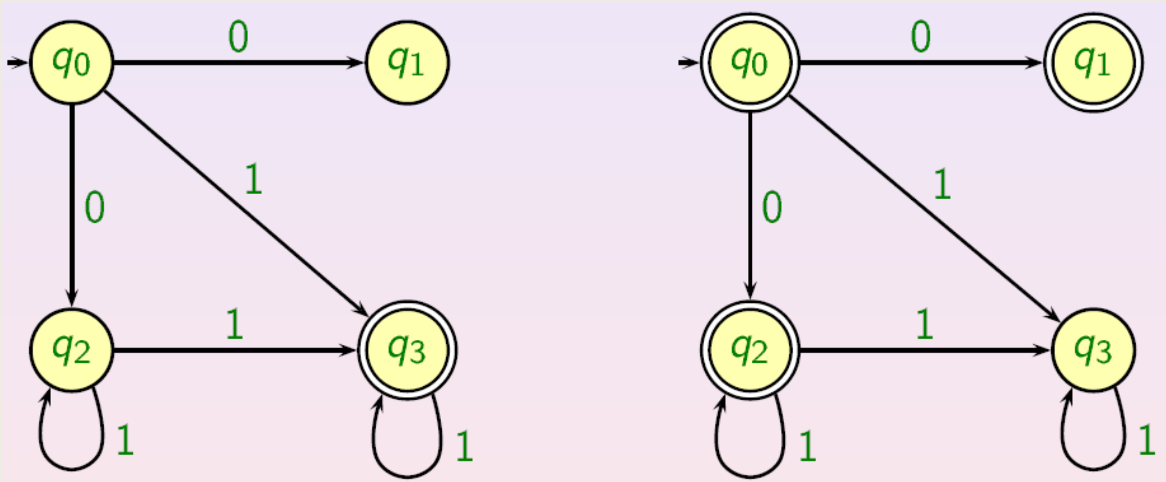
Operaciones básicas (I)

- **Unión**: Si L1 y L2 son conjuntos regulares, entonces L1$\cup$ L2 es regular
- **Concatenación**: Si L1 y L2 son regulares, entonces L1L2 es regular
- **Clausura de Kleene**: Si L es regular, entonces L* es regular
- **Complementario:** Si L$\subseteq$A* es un lenguaje regular entonces L'= A*-L es regular.

(Nota: Basta con considerar que si $M=(Q,A,\partial,q_0, F)$ es un ***autómata determinista****, $M'=(Q,A,\partial,q_0, Q\text{-}F)$ acepta el lenguaje complementario $A^*\text{-}L$)*

Complementario

- **¡OJO! El cálculo anterior del complementario no es válido para autómatas no deterministas.**



- *La cadena 011 es aceptada en ambos.*
- *La cadena 100 no es aceptada en ninguno.*

Operaciones básicas (II)

- **Intersección**: Si L1 y L2 son conjuntos regulares, entonces L1∩ L2 es regular.

  Es inmediato, dado que $L1 \cap L2 = \overline{\overline{L1} \cup \overline{L2}}$

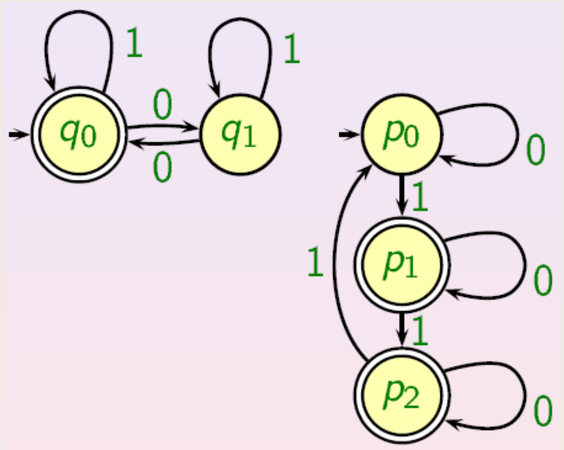
Existe también una demostración constructiva. Si $M_1=(Q_1,A,\partial_1,q_0^1, F_1)$ es un autómata finito determinístico que acepta L1, y $M_2=(Q_2,A,\partial_2,q_0^2, F_2)$ es un autómata que acepta L2, entonces

$$M=(Q_1 \times Q_2, A, \partial,(q_0^1, q_0^2),F_1\times F_2)$$ ***(autómata producto)***

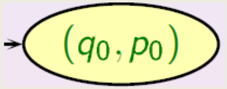
donde $\partial((q_i ,q_j),a)=(\partial_1(qi ,a), \partial_2(qj ,a))$, acepta el lenguaje L1∩L2

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.

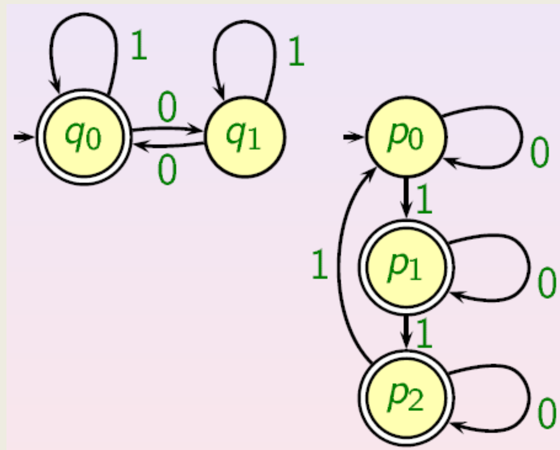


Resultado:

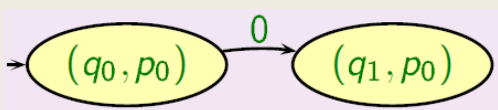


Ejemplo

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.

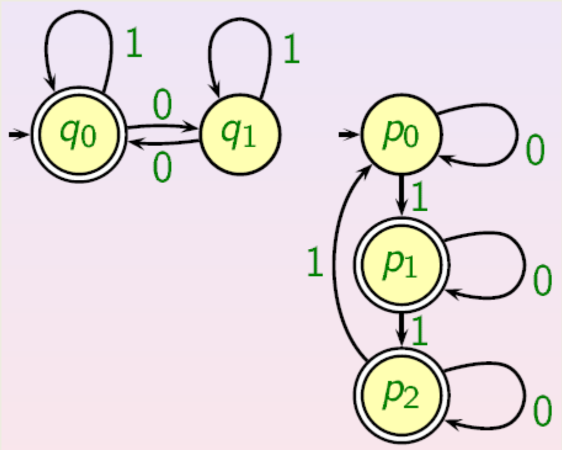


Resultado:

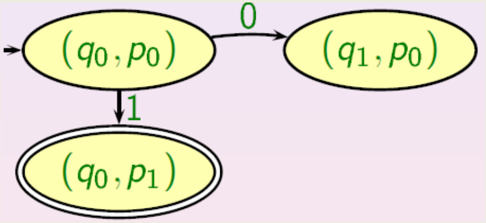


Ejemplo

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.

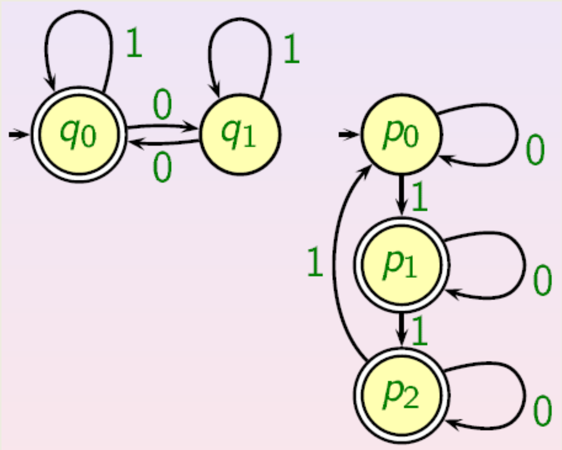


Resultado:

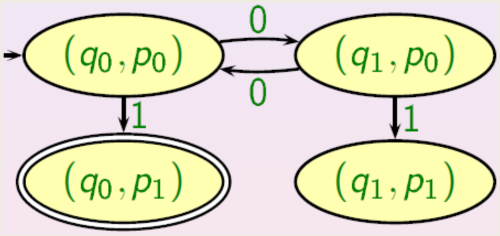


Ejemplo

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.

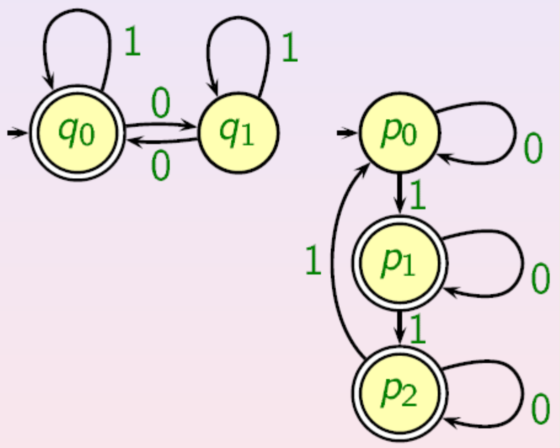


Resultado:

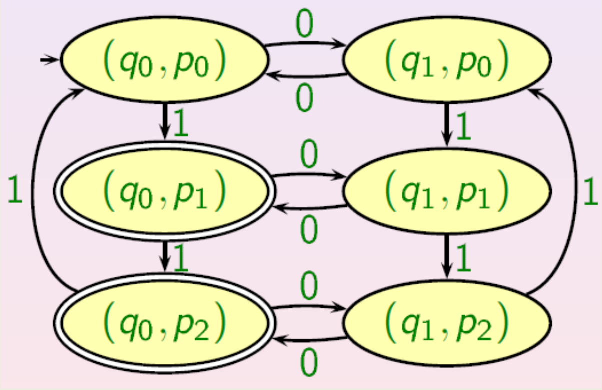


Ejemplo

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.



Resultado:

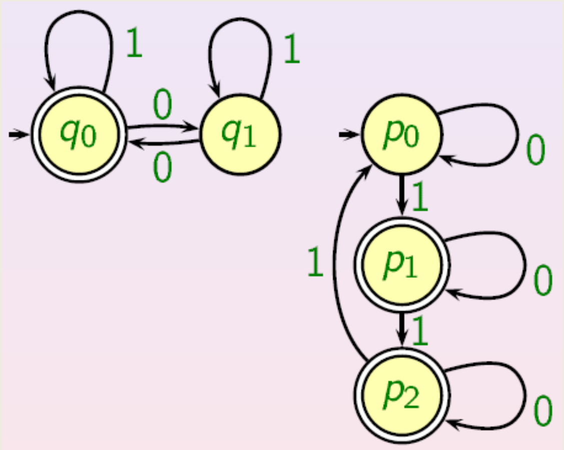


Unión de lenguajes con el autómata producto

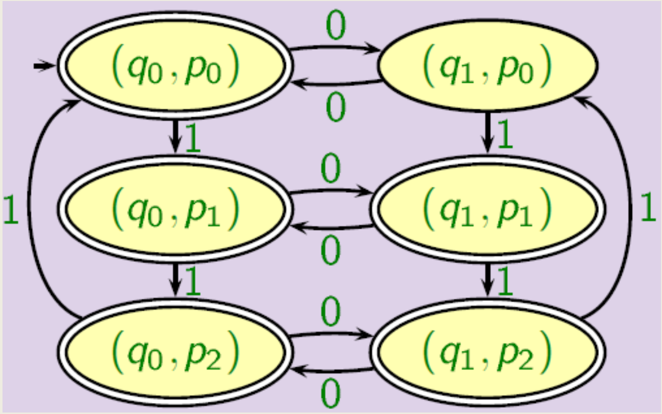
Para construir un autómata que acepte la unión de dos lenguajes, usando el autómata producto, basta con seguir los pasos similarmente a la intersección, y hacer finales aquellos estados donde al menos uno de los estados de los autómatas originales es final.

Ejemplo de unión de lenguajes con autómata producto

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.



Resultado:



Propiedades de los conjuntos regulares

1. Lema de Bombeo
2. Operaciones con conjuntos regulares
3. Algoritmos para autómatas
4. Minimización de autómatas

Homomorfismo

Si A y B son alfabetos y f: $A^* \rightarrow B^*$ un homomorfismo entre ellos, entonces $L \subseteq A^*$ es un lenguaje regular, $f(L)=\{u \in B^*: \exists v \in L$ *verificando* $f(v)=u\}$ 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 sustituir 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, 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

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

Homomorfismo inverso

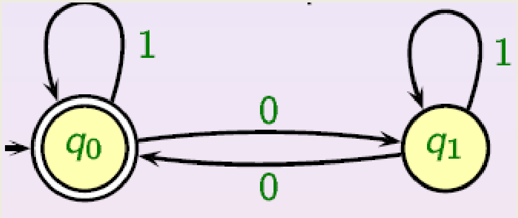
Si A y B son alfabetos y f: $A^* \rightarrow B^*$ un homomorfismo entre ellos, entonces $L \subseteq B^*$ es un conjunto regular, también lo es
$f^{-1}(L)=\{u \in A^* : f(u) \in L\}$

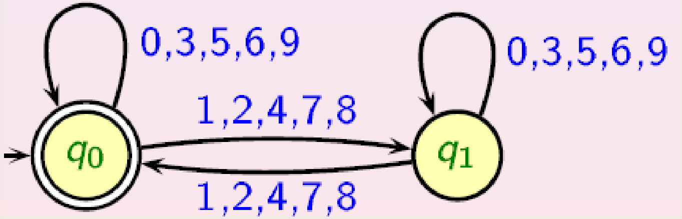
Supongamos que $M=(Q,A,\partial,q_0, F)$ que acepta el lenguaje L. Entonces el autómata $M'=(Q,A,\partial',q_0, F)$ donde $\partial'(q,a)= \partial''(q,f(a))$ Acepta el lenguaje $f^{-1}(L)$.

Ejemplo

En el lenguaje L= palabras con número par de 0's, y 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





Algoritmos: Lenguaje vacío

Para saber si un **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.

Algoritmos: Lenguaje finito o infinito

Para saber si un **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 estados finales.

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

Se comprueba posteriormente si en el grafo resultante quedan ciclos

Algoritmos: Igualdad/equivalencia

Para saber si dados dos autómatas finitos M1 y M2 comprobar si aceptan el mismo lenguaje:

Basta con construir el autómata dado por:

$$(L(M_1) - L(M_2)) \cup (L(M_2) - L(M_1)) =$$
$$(L(M_1) \cap \overline{L(M_2)}) \cup (L(M_2) \cap \overline{L(M_1)})$$

Después se comprueba si este autómata da el lenguaje vacío.

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

Propiedades de los conjuntos regulares

1. Lema de Bombeo
2. Operaciones con conjuntos regulares
3. Algoritmos para autómatas
4. Minimización de autómatas

DECSAI

Definición: Autómata minimal

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

- Estados inaccesibles
- Estados indistinguibles

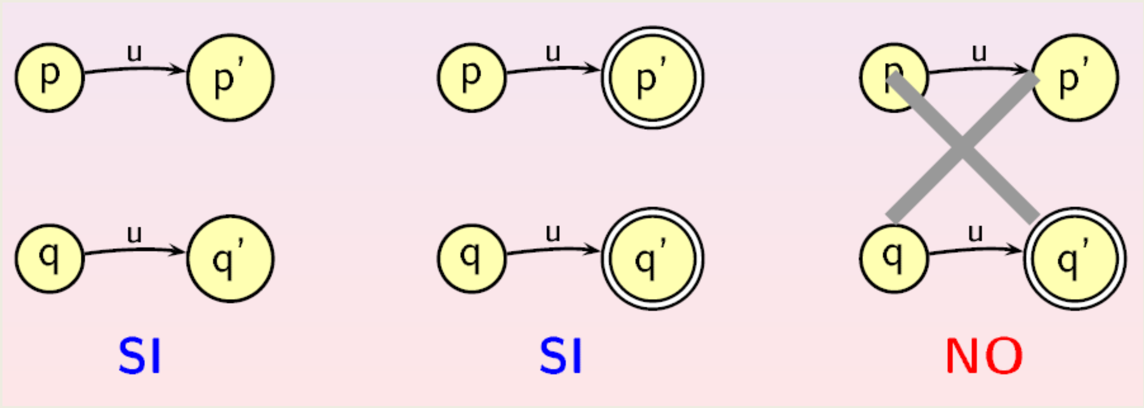
Dado un autómata M calcularemos el autómata minimal que acepta el mismo lenguaje.

Concepto: Estados indistinguibles

Si $M=(Q,A,\partial,q_0,F)$ 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

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

Ejemplos



Autómata minimal

Un autómata sin estados innacesibles es minimal si y solo si no tiene estados indistinguibles

Un autómata no es minimal $\Leftrightarrow$ tiene estados indistinguibles.

Identificación de estados indistinguibles

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 $n \in N$

Identificación de estados indistinguibles

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

Si para un n las parejas distinguibles a nivel n son las mismas que las distinguibles a nivel n+1 entonces para todo m≥n, las parejas distinguibles a nivel m son las mismas que las distinguibles a nivel n.

Teorema

Si $M=(Q,A,\partial,q_0,F)$ es un AFD y $q_1, q_2 \in Q$ son una pareja de estados indistinguibles distintos, entonces el AFD $M'=(Q',A,\partial',q_0',F')$, donde

- $Q' = Q \backslash \{q_1\}$
- $\delta'(p,a) = \delta(p,a)$ si $\delta(p,a) != q_1$ y $\delta'(q,a)=q_2$ si $\delta(p,a)=q_1$
- $q_0' = q_0$ si $q_0 != q_1$ y $q_0' = q_2$ si si $q_0 = q_1$
- $F' = F \backslash \{q_1\}$

Acepta el mismo lenguaje que M.

Teorema

Si $M=(Q,A,\partial,q_0,F)$ es un AFD y $q_1, q_2 \in Q$ son una pareja de estados indistinguibles distintos, entonces el AFD $M'=(Q',A,\partial',q_0',F')$, donde

- $Q'= Q\setminus\{q_1\}$
- $\delta'(p,a)= \delta(p,a)$ si $\delta(p,a)!=q_1$ y $\delta'(q,a)=q_2$ si $\delta(p,a)=q_1$
- $q_0'=q_0$ si $q_0!=q_1$ y $q_0'=q_2$ si si $q_0=q_1$
- $F'=F\setminus\{q_1\}$

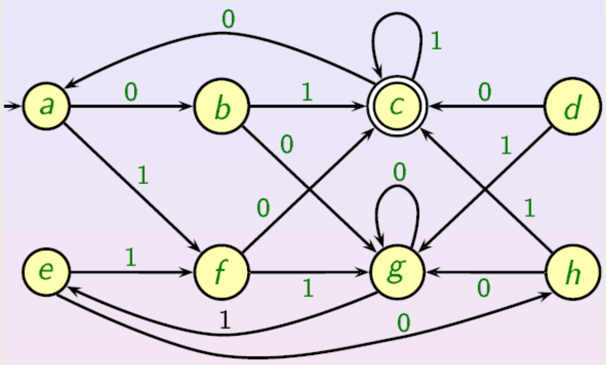
Acepta el mismo lenguaje que M.

Procedimiento de minimización de autómatas

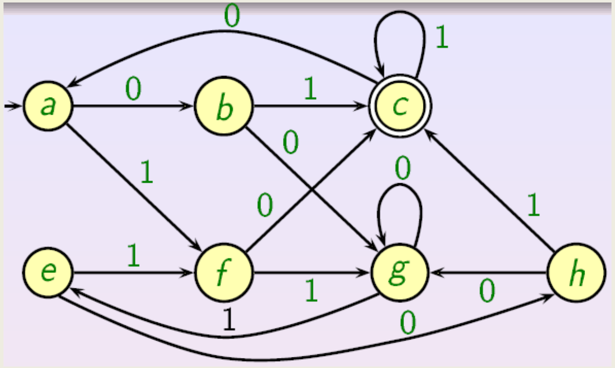
Pareja de estados → variable booleana + lista de parejas de estados

1. Eliminar estados inaccesibles.
2. Para cada pareja de estados accesibles {*qi*,*qj*}
   2.1. Si uno de ellos es final y el otro no, hacer la variable booleana asociada igual a true
3. Para cada pareja de estados accesibles {*qi*,*qj*}
   3.1 Para cada símbolo *a* del alfabeto de entrada
   - Calcular los estados *qk* y *ql* a los que evoluciona el autómata desde *qi* y *qj* leyendo *a*
   - Si $qk \neq ql$ entonces, Si la {*qk*,*ql*} está marcada entonces se marca la pareja {*qi*,*qj*} y recursivamente todas las parejas en la lista asociada.
   - Si la pareja {*qk*,*ql*} no está marcada, se añade la pareja {*qi*,*qj*} a la lista asociada a la pareja {*qk*,*ql*}

Ejemplo: Minimización del autómata

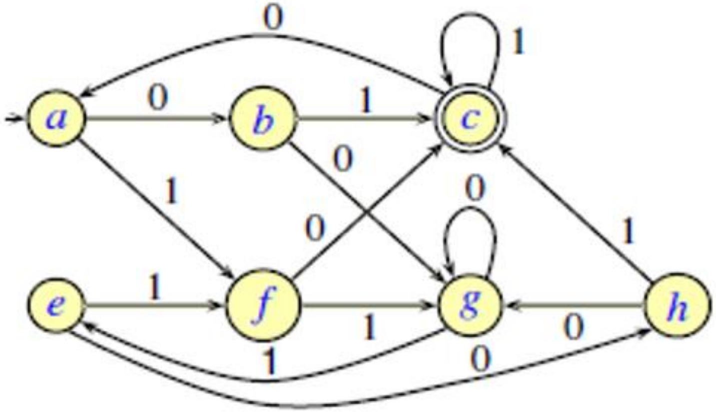


El estado d es innaccesible y se elimina



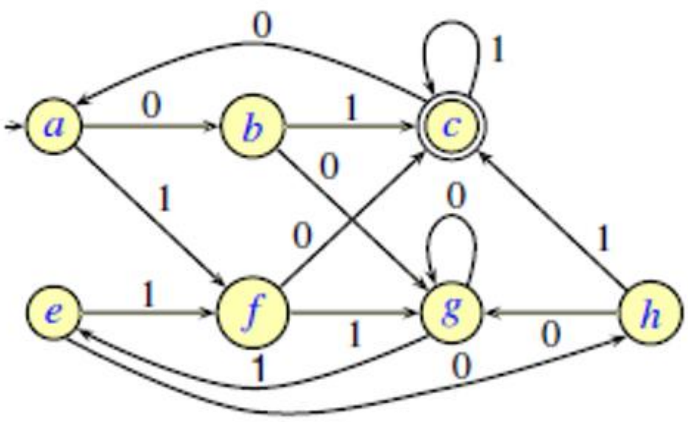
Ejemplo: Minimización del autómata

b						
c						
e						
f						
g						
h						
	a	b	c	e	f	g



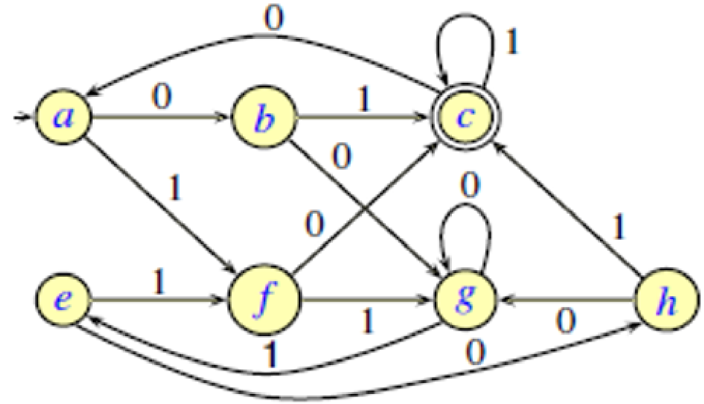
Ejemplo: Minimización del autómata (paso 2)

	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X			
g			X			
h			X			



Ejemplo: Minimización del autómata (paso 3)

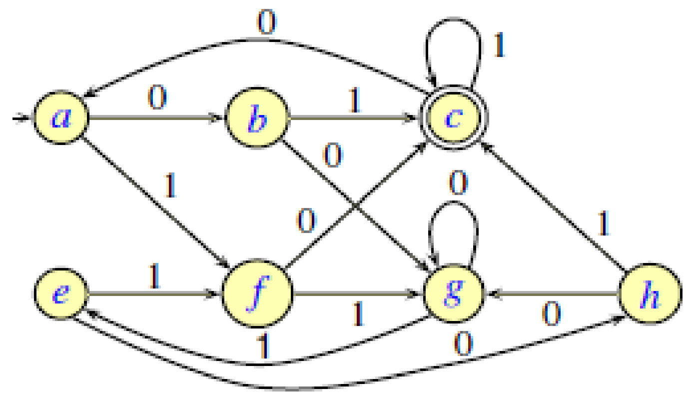
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X			
g		•	X			
h			X			



	0	1
h	g	
a	b	

Ejemplo: Minimización del autómata (paso 3)

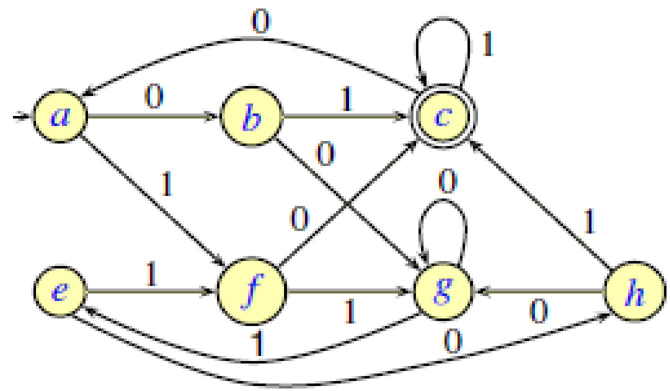
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X			
g		(h,a)	X			
h			X			



	0	1
h		c
a		f

Ejemplo: Minimización del autómata (paso 3)

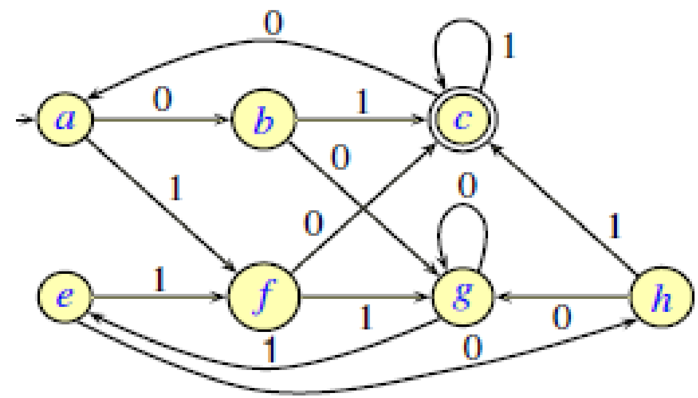
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X			
g		(h,a)	X			
h	X		X			



	0	1
h	g	
b	g	

Ejemplo: Minimización del autómata (paso 3)

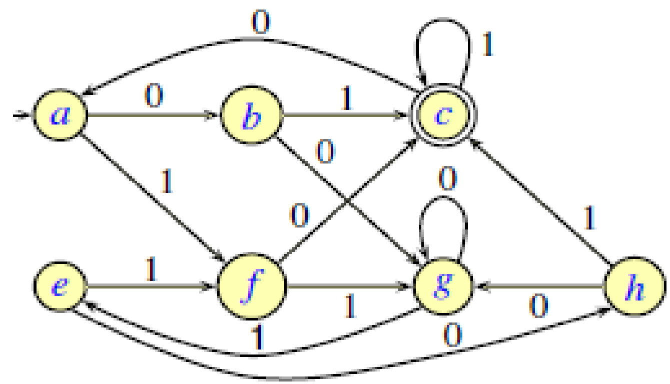
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X			
g		(h,a)	X			
h	X		X			



	0	1
h		c
b		c

Ejemplo: Minimización del autómata (paso 3)

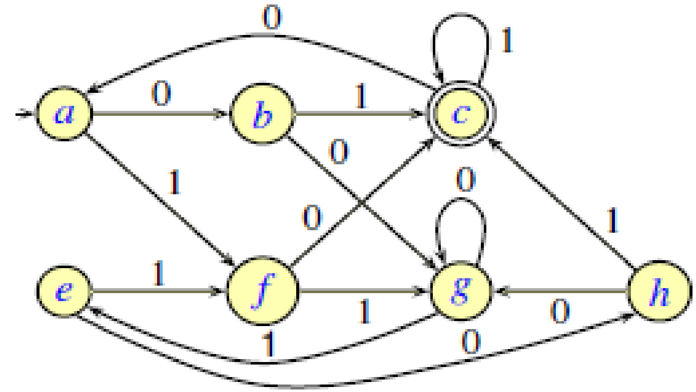
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X			
g		(h,a)	X			
h	X		X			●



	0	1
h	g	
e	h	

Ejemplo: Minimización del autómata (paso 3)

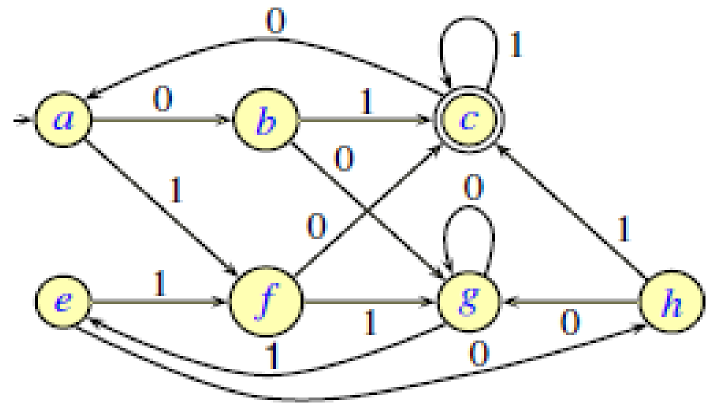
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X			
g		(h,a)	X			
h	X		X			(h,e)



	0	1
h		c
e		f

Ejemplo: Minimización del autómata (paso 3)

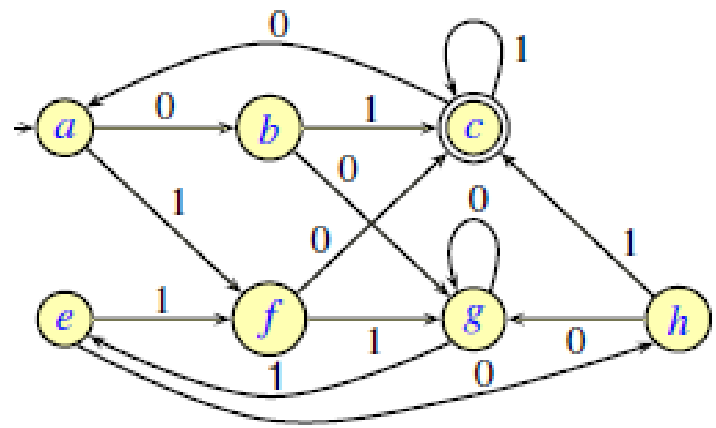
b						
c	X	X				
e			X			
f			X			
g		(h,a)	X			
h	X		X	X		(h,e)
	a	b	c	e	f	g



	0	1
h	g	
f	c	

Ejemplo: Minimización del autómata (paso 3)

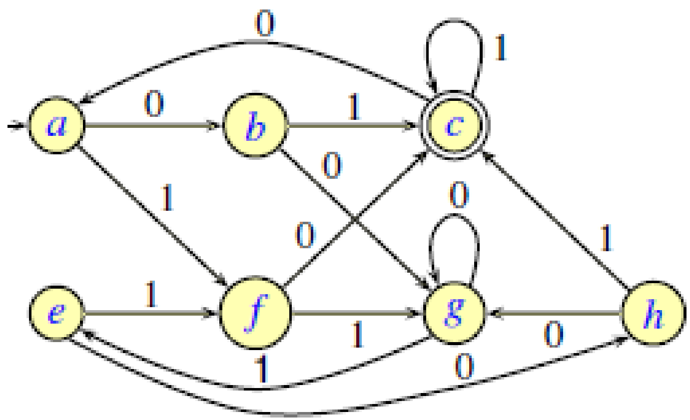
b						
c	X	X				
e			X			
f			X			
g		(h,a)	X			
h	X		X	X	X	(h,e)
	a	b	c	e	f	g



	0	1
h	g	
g	g	

Ejemplo: Minimización del autómata (paso 3)

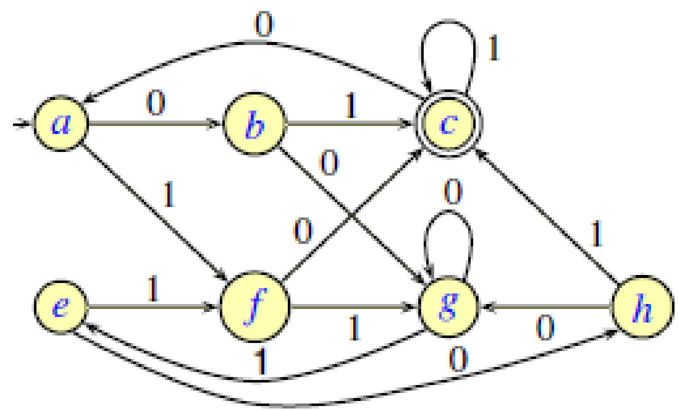
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X			
g		(h,a)	X			
h	X		X	X	X	(h,e)



	0	1
h		c
g		e

Ejemplo: Minimización del autómata (paso 3)

	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X			
g		(h,a)	X			
h	X		X	X	X	X

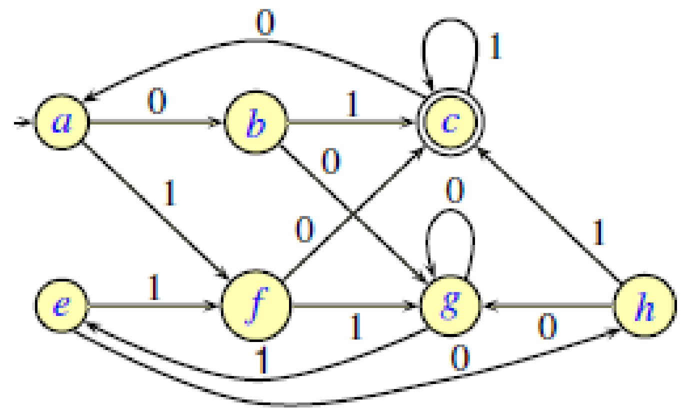


(h,e)

	0	1

Ejemplo: Minimización del autómata (paso 3)

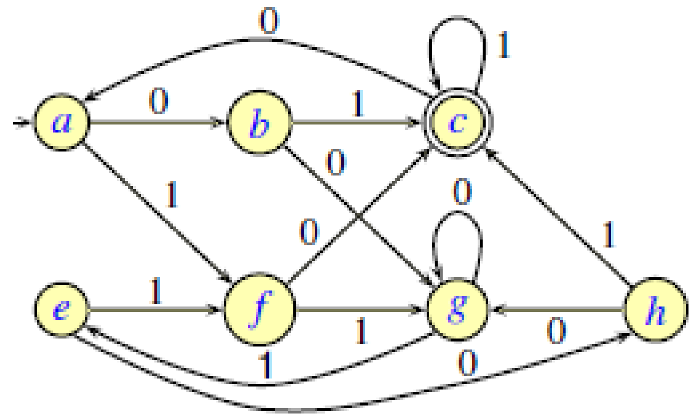
b						
c	X	X				
e			X			
f			X			
g		(h,a)	X			
h	X		X	X	X	X
	a	b	c	e	f	g



	0	1
g	g	
a	b	

Ejemplo: Minimización del autómata (paso 3)

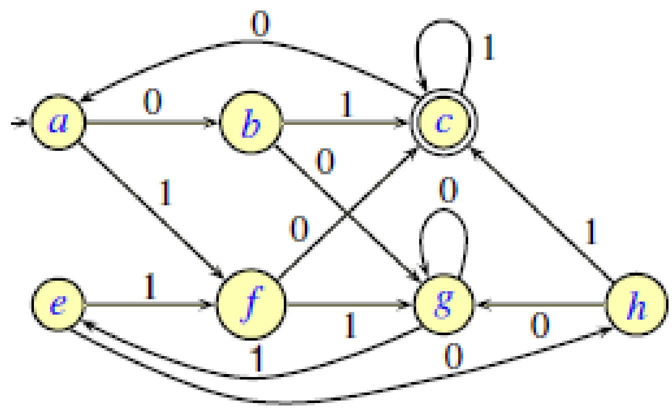
b						
c	X	X				
e			X			
f			X	•		
g		(h,a) (g,a)	X			
h	X		X	X	X	X
	a	b	c	e	f	g



	0	1
g		e
a		f

Ejemplo: Minimización del autómata (paso 3)

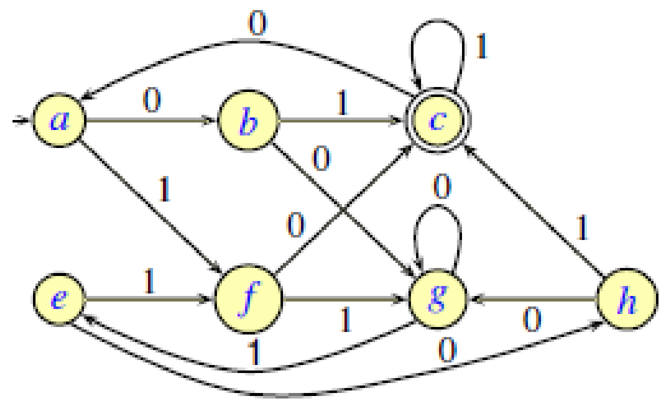
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X	(g,a)		
g		(h,a) (g,a)	X			
h	X		X	X	X	X



	0	1
g	g	
b	g	

Ejemplo: Minimización del autómata (paso 3)

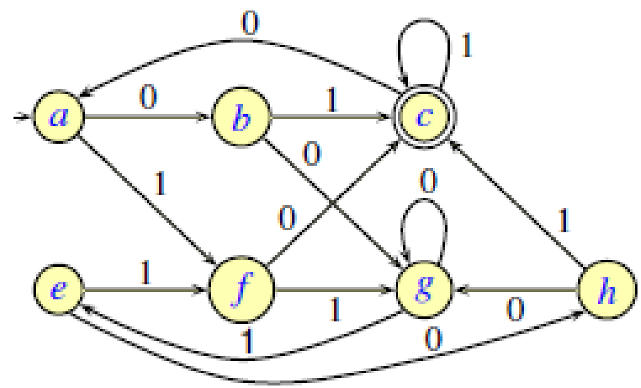
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X	(g,a)		
g		(h,a) (g,a)	X			
h	X		X	X	X	X



	0	1
g		e
b		c

Ejemplo: Minimización del autómata (paso 3)

	a	b	c	e	f	g
b						
c	X	X				
e			X			
f			X	(g,a)		
g		X	X			
h	X		X	X	X	X

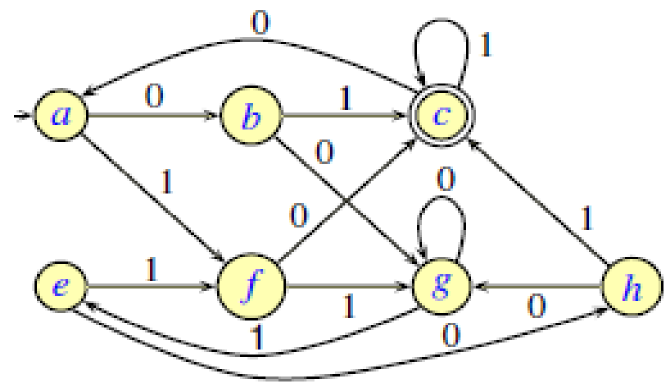


(h,a) (g,a)

	0	1

Ejemplo: Minimización del autómata (paso 3)

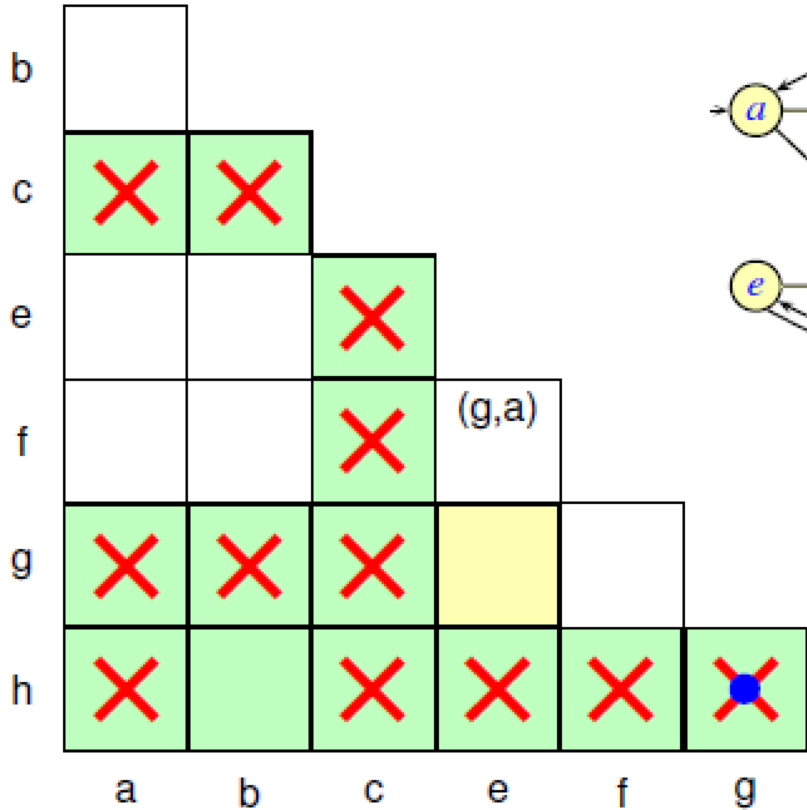
b						
c	X	X				
e			X			
f			X	(g,a)		
g		X	X			
h	X		X	X	X	X
	a	b	c	e	f	g

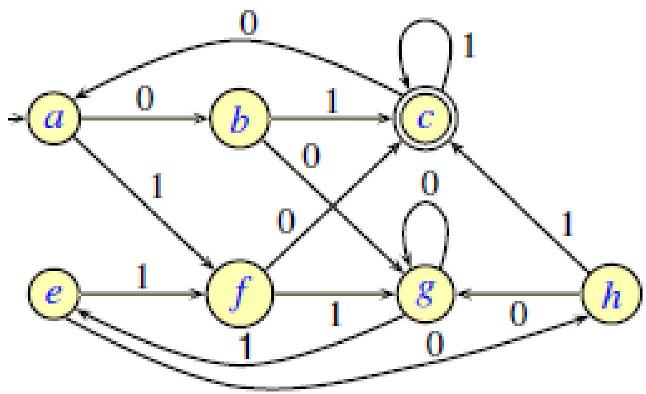


(g,a)

	0	1

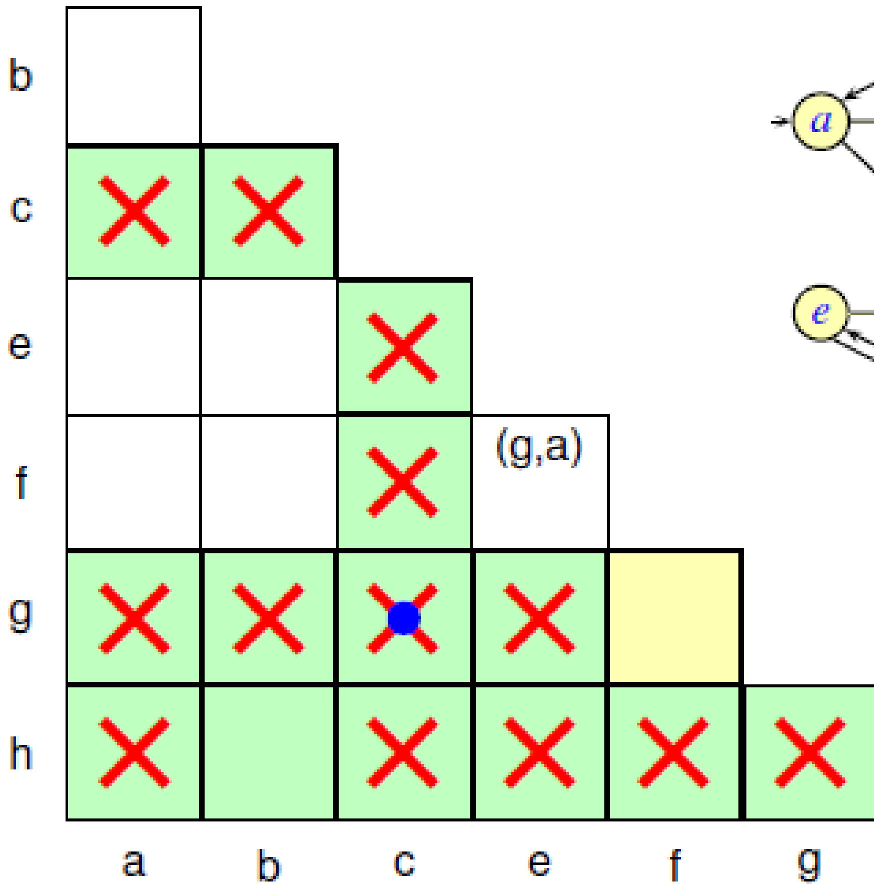
Ejemplo: Minimización del autómata (paso 3)

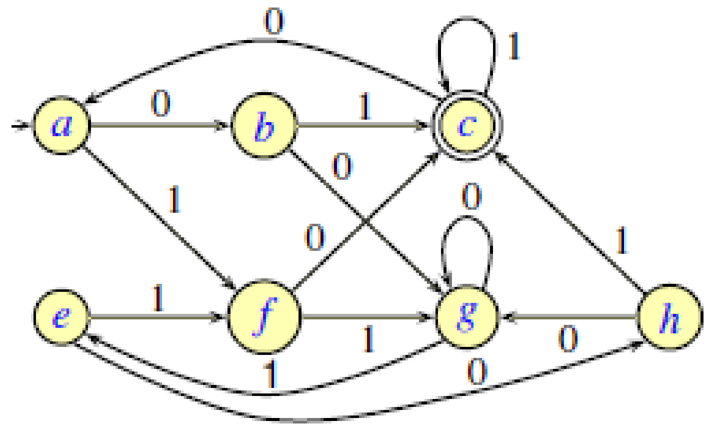




	0	1
g	g	
e	h	

Ejemplo: Minimización del autómata (paso 3)

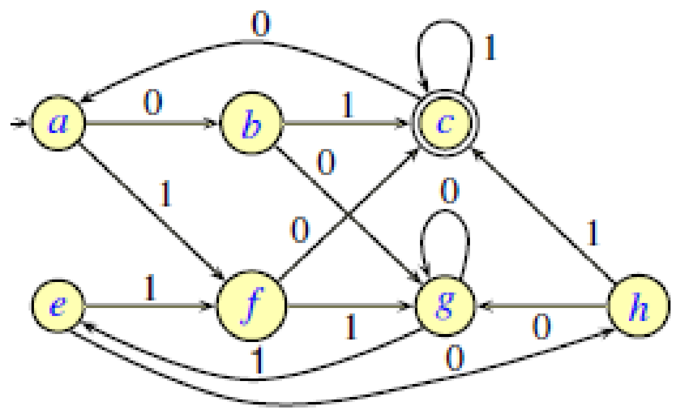




	0	1
g	g	
f	c	

Ejemplo: Minimización del autómata (paso 3)

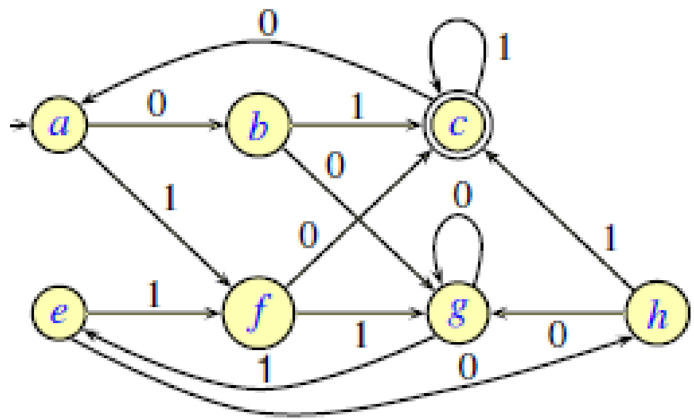
b						
c	X	X				
e			X			
f			X	(g,a)		
g	X	X	X	X	X	
h	X		X	X	X	X
	a	b	c	e	f	g



	0	1
f	c	
a	b	

Ejemplo: Minimización del autómata (paso 3)

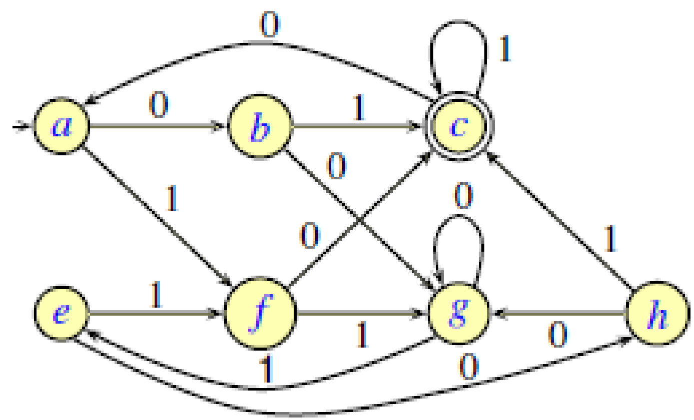
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f	X		X	(g,a)		
g	X	X	X	X	X	
h	X		X	X	X	X



	0	1
f	c	
b	g	

Ejemplo: Minimización del autómata (paso 3)

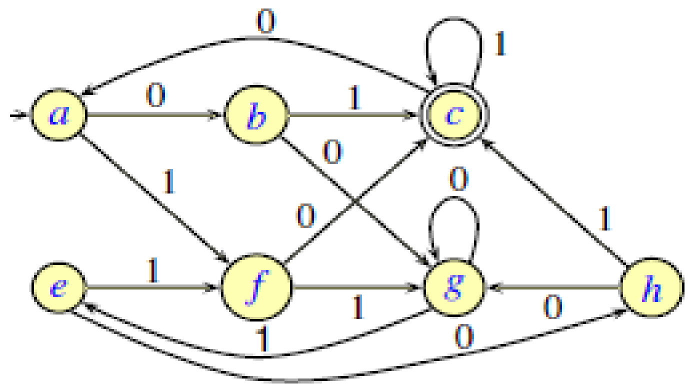
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f	X	X	X	(g,a)		
g	X	X	X	X	X	
h	X		X	X	X	X



	0	1
f	c	
e	h	

Ejemplo: Minimización del autómata (paso 3)

b						
c	X	X				
e			X			
f	X	X	X	X		
g	X	X	X	X	X	
h	X		X	X	X	X
	a	b	c	e	f	g

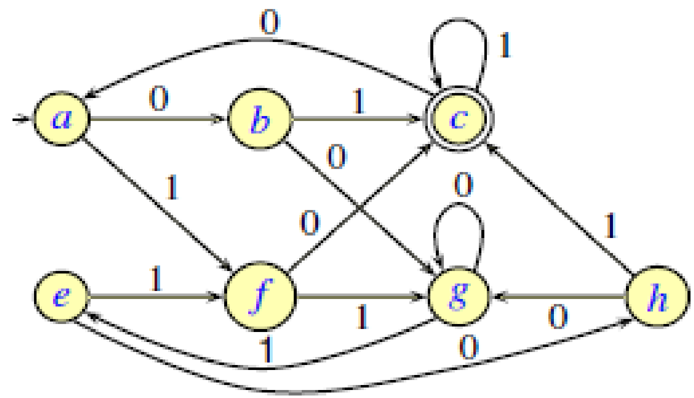


(g,a)

	0	1

Ejemplo: Minimización del autómata (paso 3)

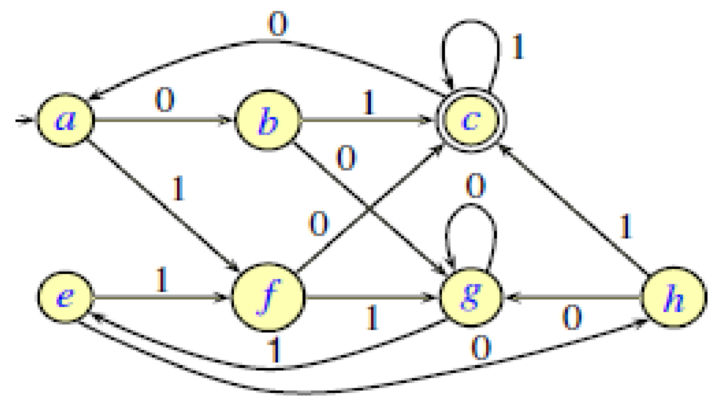
	a	b	c	e	f	g
b						
c	X	X				
e			X			
f	X	X	X	X		
g	X	X	X	X	X	
h	X	•	X	X	X	X



	0	1
e	h	
a	b	

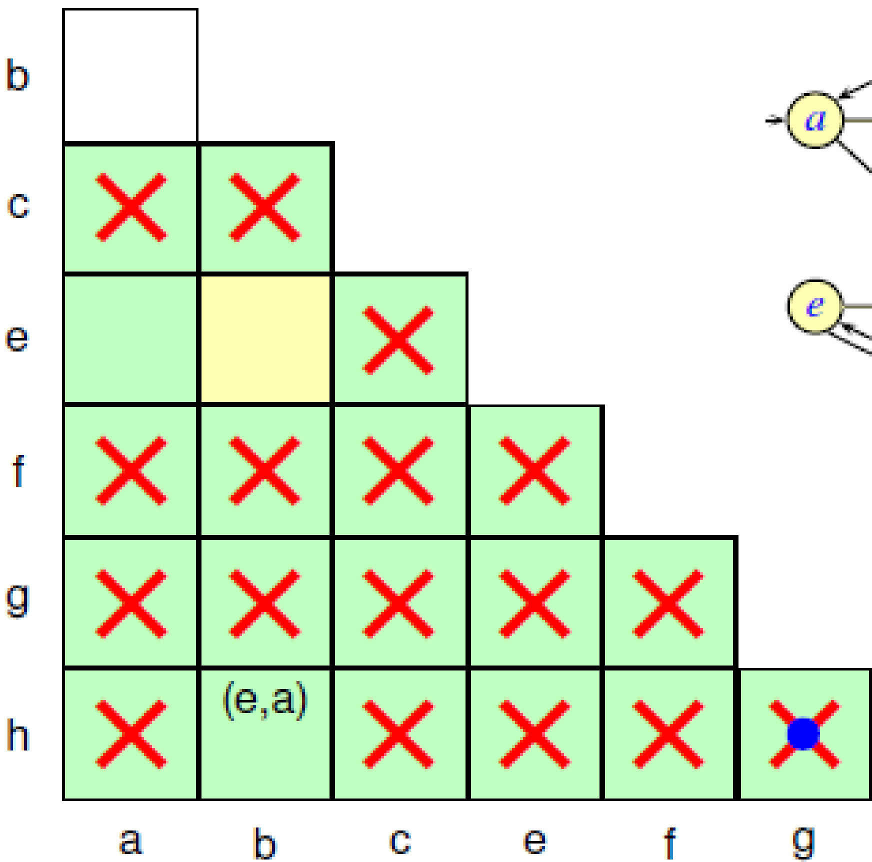
Ejemplo: Minimización del autómata (paso 3)

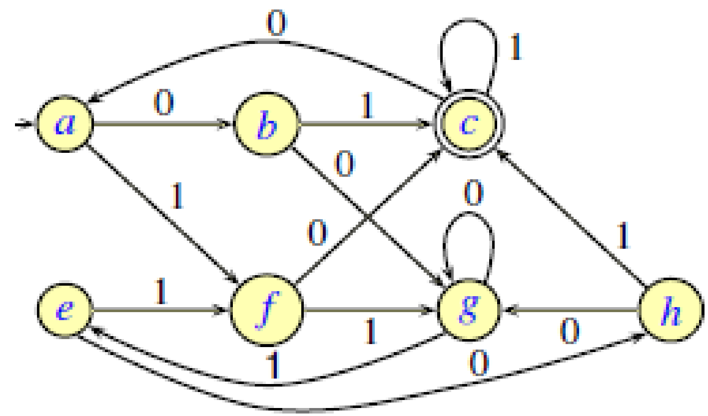
b						
c	X	X				
e			X			
f	X	X	X	X		
g	X	X	X	X	X	
h	X	(e,a)	X	X	X	X
	a	b	c	e	f	g



	0	1
e		f
a		f

Ejemplo: Minimización del autómata (paso 3)

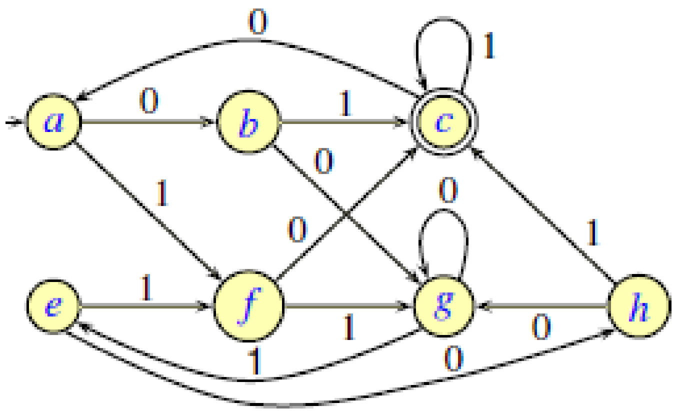




	0	1
e	h	
b	g	

Ejemplo: Minimización del autómata (paso 3)

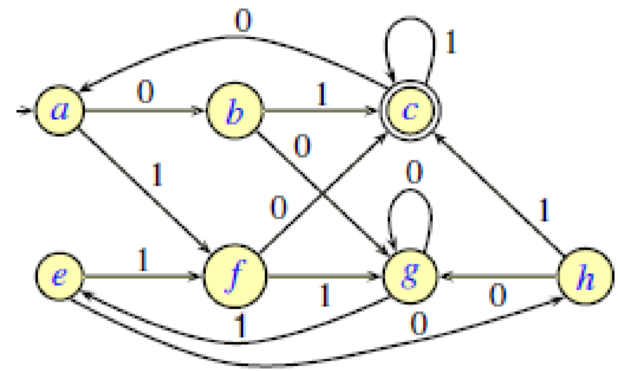
	a	b	c	e	f	g
b						
c	X	X				
e		X	X			
f	X	X	X	X		
g	X	X	X	X	X	
h	X	(e,a)	X	X	X	X



	0	1
b	g	
a	b	

Ejemplo: Minimización del autómata (paso 3)

b	X					
c	X	X				
e		X	X			
f	X	X	X	X		
g	X	X	X	X	X	
h	X	(e,a)	X	X	X	X
	a	b	c	e	f	g



$\mathbf{b \equiv h}$, $\mathbf{a \equiv e}$

	0	1

Construcción del autómata minimal

El autómata minimal se construye identificando los estados indistinguibles.

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

R es la relación de equivalencia de indistiguibilidad entre estados

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

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

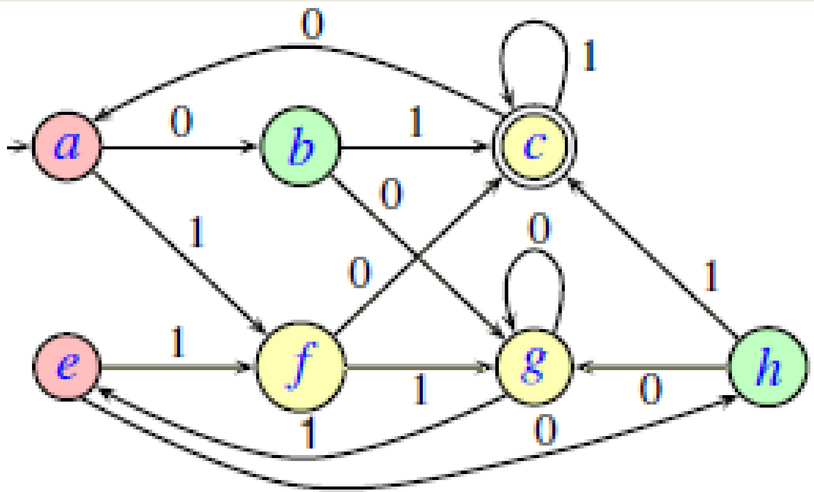
$Q_m=\{[q]: q \text{ es accesible desde } q_0\}$

$F_m =\{[q]: q \in F\}$

$\partial_m ([q],a)=[\partial(q,a)]$

$q_0^m=[q_0]$

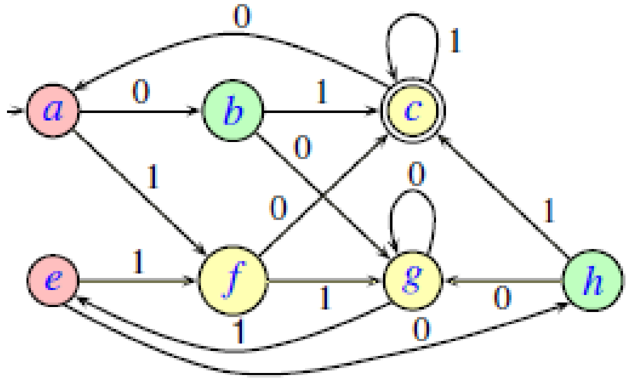
Ejemplo: Construcción del autómata



$$\mathbf{b} \equiv \mathbf{h},$$

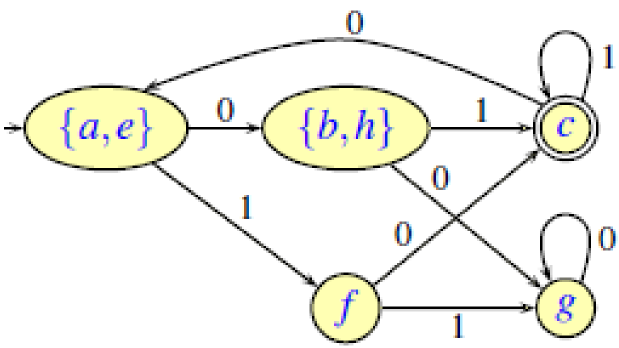
$$\mathbf{a} \equiv \mathbf{e}$$

Ejemplo: Construcción del autómata



b ≡ h,

a ≡ e



UNIVERSIDAD DE GRANADA

DECSAI

Modelos de Computación

Grado en Ingeniería Informática

Tema 3 – Propiedades de los conjuntos regulares

Manuel Pegalajar Cuéllar
manupc@ugr.es

Departamento de Ciencias de la Computación e Inteligencia Artificial
http://decsai.ugr.es