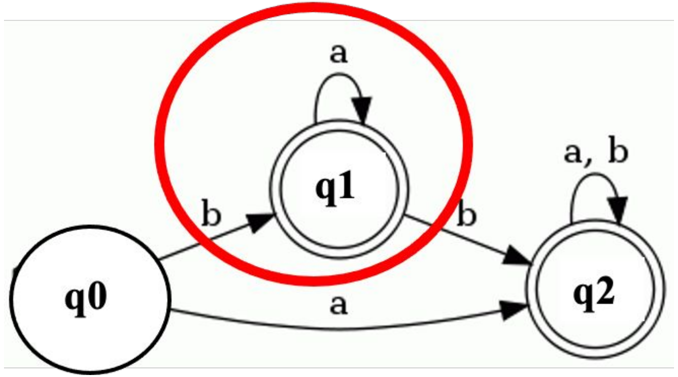
Minimización de autómatas finitos

Minimización AFD

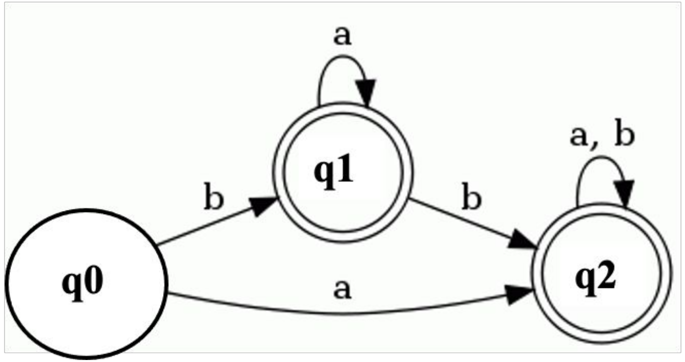
Nos encontramos con el inconveniente de que, dado un AFD M, podría tener algunos estados **redundantes**



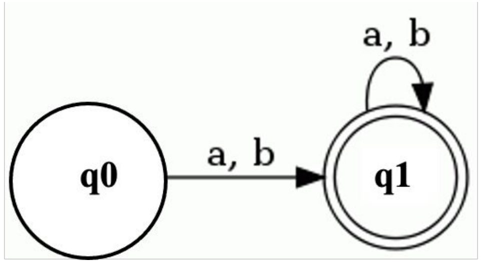
Como podemos ver, el estado q1 puede ser removido y M seguiría aceptando el mismo lenguaje

Minimización AFD

Vemos que ambos autómatas son “**equivalentes**”



AFD *minimizado*



Un autómata está minimizado cuando todos sus estados son necesarios

Repasito de relaciones y clases de equivalencias

Sea R una **relación** binaria en un conjunto U es es de **equivalencia** si

- R es **reflexiva** : $(x, x) \in R$, para cada $x \in U$
- R es **simétrica** : $(x, y) \in R$ implica que $(y, x) \in R$, para cada $x, y \in U$
- R es **transitiva** : $(x, y), (y, z) \in R$ implica que $(x, z) \in R$, para cada $x, y, z \in U$

Para cada $x \in U$ se define la **clase de equivalencia** x como: $[x] = \{ y, \in U$ tal que $(x, y) \in R \}$

Una clase de equivalencia R en U determina una partición y toda partición de U determina una relación de equivalencia

La partición tiene como elementos las clases de equivalencia que son disjuntas dos a dos y su union es U

- Para cualquier a_i, a_i no relacionados tenemos que $[a_i] \cup [a_j] = \emptyset$
- La unión de todos integra el total

Nos interesa estudiar propiedades de los AFDs para poder definir:

- **Equivalencias entre estados** (del AFD) que preserven el lenguaje que reconoce
- **Equivalencias entre autómatas** en relación al lenguaje que reconocen
- **Algoritmos** que nos permitan computar las relaciones de equivalencia entre estados y autómatas

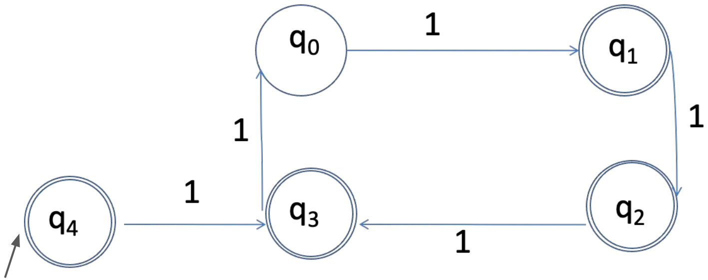
Autómata mínimo

Decimos que un autómata $A = (Q_A, \Sigma, \delta_A, q_{0_A}, F_A)$ es *mínimo* **si y sólo si** para todo $B = (Q_B, \Sigma, \delta_B, q_{0_B}, F_B)$ tal que L(A) = L(B), ocurre que $|Q_A| \leq |Q_B|$

Decimos que un estado q en $M = (Q, \Sigma, \delta, q_0, F)$ es *accesible* **si y sólo si** $(\exists w \in \Sigma^*)\,(\delta^*(q_0, w) = q)$

Decimos que un autómata M es *conectado* **si y sólo si** todos sus estados son accesibles

$$(\forall q \in Q)(\exists w \in \Sigma^*)\,(\delta^*(q_0, w) = q)$$



Estado **no** accesible

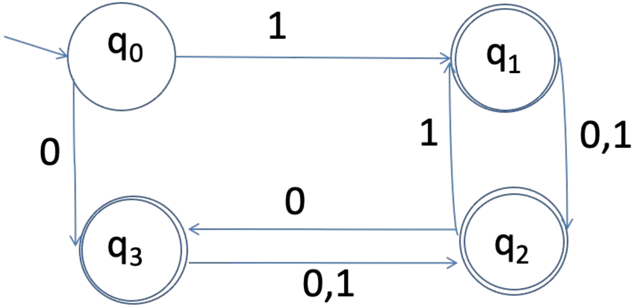
Equivalencia entre estados

Dos estados p y q son equivalentes **si y sólo si** son indistinguibles, en el sentido que todo string que lleve a un estado final a partir de p también lo hace a partir de q, y viceversa.

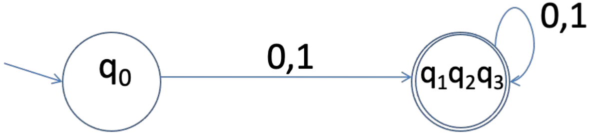
Dado $M = (Q, \Sigma, \delta, q_0, F)$ la relación de equivalencia entre estados

E_m sobre Q se define

$$(\forall s \in Q)(\forall t \in Q)(\, s\, E_M\, t \Leftrightarrow (\forall x \in \Sigma^*)(\delta^*(s,x) \in F \Leftrightarrow \delta^*(t,x) \in F))$$



Notar que q1, q2 y q3 son equivalentes



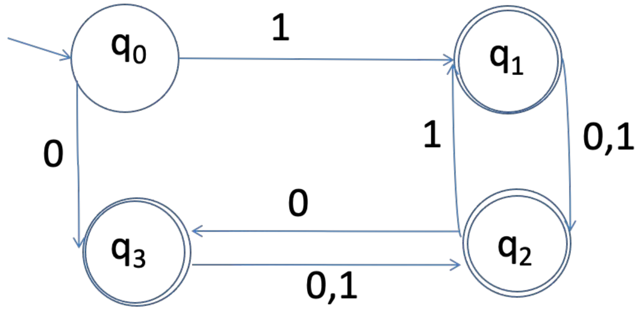
Equivalencia entre estados

El AFD N define la siguiente relación E_n como el conjunto de pares

{ (q0, q0), (q1, q1), (q1, q2), (q1, q3), (q2, q1), (q2, q2), (q2, q3), (q3, q1), (q3, q2), (q3, q3) }

Como clases de equivalencia:

$[q0]_{EN} = \{ q0 \}$ $[q1]_{EN} = [q2]_{EN} = [q3]_{EN} = \{ q1, q2, q3 \}$



La partición en Q entonces es Q/EN = **{ { q0 }, { q1, q2, q3 } }**

Y por lo tanto, el autómata mínimo que acepta L(N) tiene solo dos estados

Autómata reducido

Decimos que un autómata $M = (Q, \Sigma, \delta, q_0, F)$ es *reducido* si y sólo si $(\forall s, t \in Q)(s E_M t \Leftrightarrow s = t)$

Nota: En un autómata reducido, E_m debe ser la relación de identidad sobre Q, y en este caso cada clase de equivalencia contendrá sólo un elemento.

Teorema: Un autómata finito determinístico *reducido* y *conectado* es *mínimo*

Autómata conectado

Dado un AFD $M=(Q,\Sigma,\delta,q_0,F)$ definimos un nuevo autómata $M^c=(Q^c,\Sigma,\delta^c,q_0^c,F^c)$ llamado *conectado* de la siguiente manera:

$$M^c=(Q^c,\Sigma,\delta^c,q_0^c,F^c)$$

$$Q^c=\{\mathrm{q}\in\mathrm{Q}\mid\exists\mathrm{x}\in\Sigma^*\mid\delta^*(\mathrm{q}_0,\mathrm{x})=\mathrm{q}\}$$ Q sin estados no alcanzables

$$q_0^c=\mathrm{q}_0$$

$$F^c=\mathrm{F}\cap Q^c=\{\mathrm{f}\in\mathrm{F}\mid\exists\mathrm{x}\in\Sigma^*\mid\delta^*(\mathrm{q}_0,\mathrm{x})=\mathrm{f}\}$$ estado finales alcanzables desde q0

δ^c se deriva de la restricción de δ a $Q^c\times\Sigma$,

$$(\forall\mathrm{a}\in\Sigma)(\forall\mathrm{q}\in Q^c)(\delta^c(\mathrm{q},\mathrm{a})=\delta(\mathrm{q},\mathrm{a}))$$ transiciones desde y hacia estados conectados

Autómata M módulo

Dado un AFD $M = (Q, \Sigma, \delta, q_0, F)$ definimos un nuevo autómata M/E_m

$$M/E_M = < Q_{EM}, \Sigma, \delta_{EM}, q_{0_{EM}}, F_{EM} >, \text{ donde}$$

$$Q_{EM} = \{[q]_{EM} \mid q \in Q\}$$

$$q_{0_{EM}} = [q_0]_{EM}$$

$$F_{EM} = \{[q]_{EM} \mid q \in F\} \text{ y}$$

δ_{EM} está definida por

$$(\forall a \in \Sigma)(\forall [q]_{EM} \in Q_{EM})(\delta_{EM}([q]_{EM}, a) = [\delta(q, a)]_{EM})$$

Teoremas!

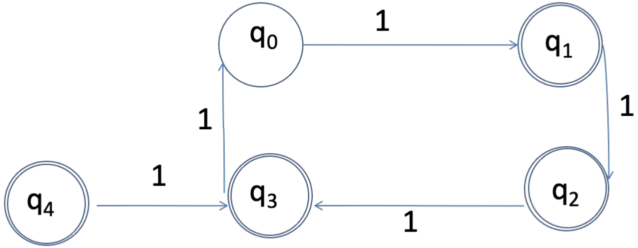
Teorema: Dado el AFD $M = (Q, \Sigma, \delta, q_0, F)$, M/$E_m$ es **reducido**

Teorema: Si $M = (Q, \Sigma, \delta, q_0, F)$ es **conectado**, entonces M/E_m es **conectado**

Teorema: Dado $M = (Q, \Sigma, \delta, q_0, F)$ entonces $\mathrm{L}(\mathrm{M}/\mathrm{E}_\mathrm{M}) = L(M)$

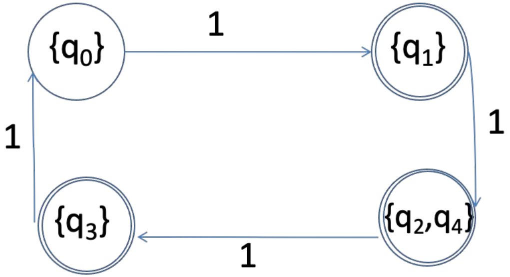
Corolario: Dado $M = (Q, \Sigma, \delta, q_0, F)$ entonces M/E_m es **mínimo**

Ejemplito



Vemos que B **no** es reducido ya que $\mathbf{E_b = \{ \{ q0 \}, \{ q3 \}, \{ q1 \}, \{ q2, q4 \} \}}$

Computamos B/E_b



1) Por Teoremas 1 y 2, B/E_b es **reducido** y **conectado**

2) Por Corolario, B/E_b es **mínimo**

Nota: Observar que $\{ q_0 \}$ se distingue de los demás estados con λ. $\{ q_0 \}$ se distingue de $\{ q_3 \}$ con 1 y de $\{ q_2, q_4 \}$ con 11. $\{ q_2, q_4 \}$ se distingue de $\{ q_3 \}$ con 1.

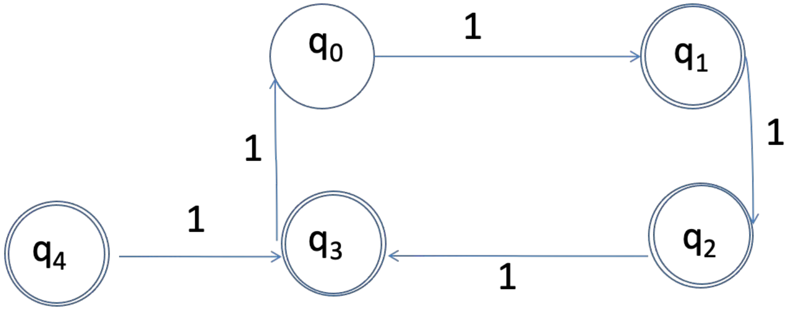
Algunas definiciones

Dado un autómata finito $A = \langle Q, \Sigma, \delta, q0, F\rangle$ y un entero i, definimos la i-ésima relación parcial de equivalencia de estados sobre A, una relación entre los estados de A denotada por EiA, de siguiente modo $\mathbf{(\forall s,t \in Q)(sE_{iA}t \Leftrightarrow (\forall x \in \Sigma^* \mid |x| \leq i)(\delta^*(s,x) \in F \Leftrightarrow \delta^*(t,x) \in F))}$

- E_{iA} relaciona estados que no pueden distinguirse con strings de longitud i o menor; esto contrasta con la definición de E_A, que relaciona estados que no pueden distinguirse con strings de cualquier longitud

- E0A denota un criterio relativamente débil, que es fortalecido con las sucesivas relaciones E_{iA}. Estas relaciones culminan en la relación que buscamos, E_A

$$(\forall s,t\in Q)(sE_{iB}t\Leftrightarrow(\forall x\in\Sigma^* \,||x|\leq i)(\delta^*(s,x)\in F\Leftrightarrow\delta^*(t,x)\in F))$$



Para i= 0 : x = λ, E_{0B}={ { q_0 } , { q_1, q_2, q_3, q_4 } }

En E_{1B}, λ diferencia q_0 de los otros estados, pero 1 distingue q_3 de q_1, q_2 y q_4 pues $\delta(q_3, 1) \notin F$, pero $\delta(q_i , 1) \in F$ para i = 1, 2, y 4.

Así, E_{1B}={ { q_0 }, { q_3 }, { q_1, q_2, q_4 } }

Similarmente, $\delta(q_1, 11) \in F$ pero $\delta(q_2, 11) \notin F$ y $\delta(q_4, 11) \notin F$. E_{2B}= { { q_0 }, { q_3 }, { q_1 },{ q_2, q_4 } }

Fácilmente se comprueba que $E_{2B} = E_{3B} = E_{4B} = E_{5B} = \cdots$, y así $E_B = E_{2B}$

Lema 3.2

- Dado un autómata finito $A = \langle Q,\Sigma, \delta, q0,F\rangle$ y un entero m, E_A es un refinamiento de E_{mA}, y así $(\forall s, t \in Q)(sE_At \Rightarrow sE_{mA}t)$. Esto es, $E_A \subseteq E_{mA}$
- **Demo**: Sea $s, t \in Q$. Entonces $sE_At \Rightarrow$ (por definición de E_A) $(\forall x \in \Sigma^*)(\delta^*(s, x) \in F \Leftrightarrow \delta^*(t, x) \in F) \Rightarrow$ (obviamente, para x "más cortos") $(\forall x \in \Sigma^* \,|\, |x| \leq m)(\delta^*(s,x) \in F \Leftrightarrow \delta^*(t,x) \in F) \Rightarrow$ (por definición of E_{mA}) $sE_{mA}t$

Lema 3.3

- Dado un autómata finito $A = \langle Q,\Sigma, \delta, q0,F\rangle$, E_{0A} tiene dos clases de equivalencia, F y $Q - F$ (excepto que F o $Q - F$ sean vacías, en cuyo caso solo hay una clase equivalencia, Q)
- **Demo:** Inmediata por la definición de E_{0A}

Lema 3.9

- Dado un AFD $A = \langle Q, \Sigma, \delta, q_0, F\rangle, (\exists m \in N \mid E_{mA} = E_{m+1A}) \Rightarrow (\forall k \in N)(E_{m+kA} = E_{mA})$
- **Demo:** Por inducción sobre k

Lema 3.10

- Dado un autómata finito $A = \langle Q,\Sigma,\delta, q0,F\rangle (\exists m \in N \mid E_{mA} = E_{m+1A}) \Rightarrow E_{mA} = E_A$
-

Algoritmos de Minimización de Autómatas Finitos

La tarea de *minimizar* un AFD, entonces, consiste en transformar automáticamente un AFD en otro AFD con un número **mínimo** de estados

- Se conocen varios algoritmos y variantes

- EL cálculo de E_A los sucesivos E_{mA} para un autómata A y sucesivos m= 0,1,... es un algoritmo

- Veremos un algoritmo relacionado muy simple y **NO** muy eficiente

Algoritmo de Minimización

Sea un AFD $A = (Q, \Sigma, \delta, q_0, F)$ tenemos que dos estados p y q son **distintos** si

- $p \in F$ y $q \notin F$ o viceversa, o bien

- Para algún $a \in \Sigma$, $\delta(p, a) \neq \delta(q, a)$

NOTA: Usando esta definición inductiva, podemos calcular cuáles estados son distinto

Algoritmo de Minimización (Técnica)

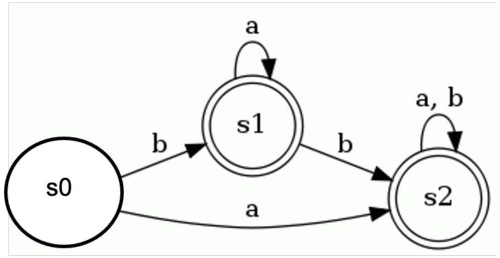
1. Creamos una tabla triangular llamada **DISTINTO** (inicialmente en blanco)

2. Para **cada par** de estados p, q **si** p es un estado final y q no (o viceversa), **entonces** DISTINTO(p, q) = ε

3. Iteramos hasta que no haya cambios en DISTINTO:
    a. Para cada par de estados (p, q) y cada símbolo a, **si** DISTINTO(p, q) está en blanco **y** DISTINTO(δ(p, α), δ(q,α)) no está en blanco, **entonces** DISTINTO(p, q) = a

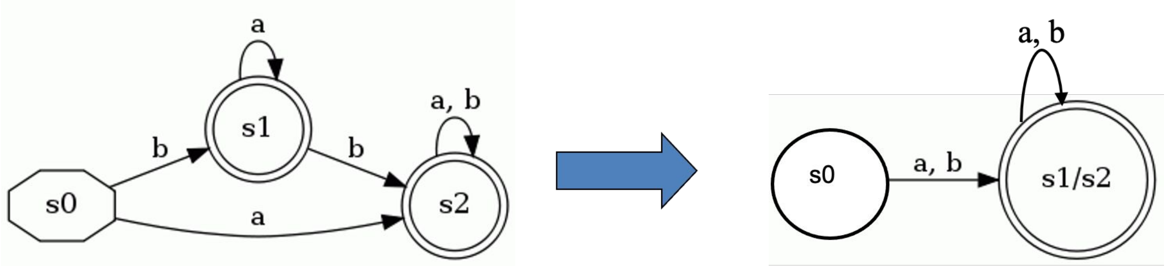
Ejemplito :)

s0			
s1	ε		
s2	ε		
	s0	s1	s2



Notar que DISTINTO(s1, s2) está vacío, entonces s1 y s2 **son estados equivalentes**

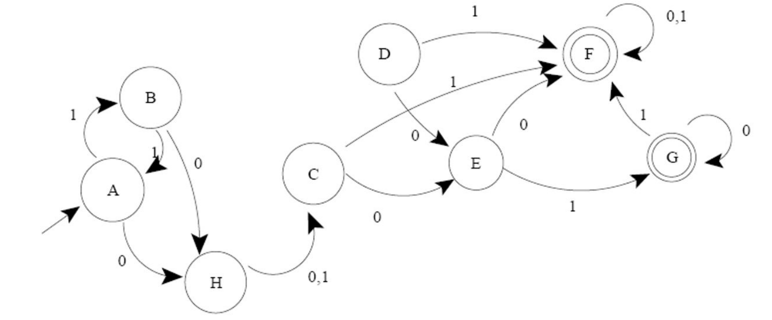
s0			
s1	ε		
s2	ε		
	s0	s1	s2



Ejemplo complejo

Paso 1)

	a	b	c	d	e	f	g
b							
c							
d							
e							
f	ϵ	ϵ	ϵ	ϵ	ϵ		
g	ϵ	ϵ	ϵ	ϵ	ϵ		
h						ϵ	ϵ



Paso 2) "Para cada par de estados (p,q) y cada símbolo α Si DISTINTO(p,q) es blanco y DISTINTO(δ(p,α), δ(q,α)) es no blanco DISTINTO(p,q) = α"

It 1)

	a	b	c	d	e	f	g
b							
c	1	1					
d	1	1					
e	0	0	0	0			
f	ϵ	ϵ	ϵ	ϵ	ϵ		
g	ϵ	ϵ	ϵ	ϵ	ϵ		
h			1	1	0	ϵ	ϵ

it 2)

	a	b	c	d	e	f	g
b							
c	1	1					
d	1	1					
e	0	0	0	0			
f	ϵ	ϵ	ϵ	ϵ	ϵ		
g	ϵ	ϵ	ϵ	ϵ	ϵ		
h	1	1	1	1	0	ϵ	ϵ

it 3)

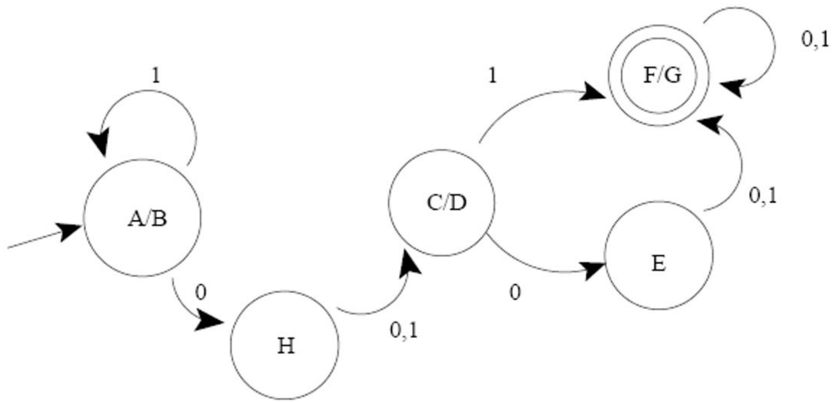
	a	b	c	d	e	f	g
b							
c	1	1					
d	1	1					
e	0	0	0	0			
f	ϵ	ϵ	ϵ	ϵ	ϵ		
g	ϵ	ϵ	ϵ	ϵ	ϵ		
h	1	1	1	1	0	ϵ	ϵ

(No produce cambios)

Y los casilleros en blanco **son pares de estados equivalentes**

Ejemplo complejo

Combinamos los estados **equivalentes** { (a, b), (d, c), (g, f) } para obtener el AFD **minimizado**



Conclusión

- Minimización es un proceso fácilmente comprensible, y es útil en muchas áreas
  - Algoritmos muy similares son usados **para hacer optimización de compiladores**. Por ejemplo para eliminar cálculos duplicados

- El algoritmo descrito es **O(n^2)**
  - John Hopcroft describe otro algoritmo más complejo que es **O(n log n)**