Tema 4: Autómatas Finitos

Autómatas Finitos. Bibliografía

- M. Alfonseca, J. Sancho y M. Martínez. “Teoría de Lenguajes, Gramáticas y Autómatas”, R.A.E.C., Madrid, (1998).
- P. Isasi, P. Martínez y D. Borrajo. “Lenguajes, Gramáticas y Autómatas, un Enfoque Práctico”. Addison-Wesley, (1997).
- G. Fernández y F. Sáez. “Fundamentos de Informática (Lógica, Autómatas, Algoritmos y Lenguajes)”. Anaya, (1995).

Autómatas Finitos. Definiciones

- Los Autómatas Finitos son de dos tipos:
  - **Deterministas:**
    - cada combinación **(estado, símbolo de entrada)** produce un solo **(estado)**
  - **No Deterministas:**
    - cada combinación **(estado, símbolo de entrada)** produce varios **(estado1, estado 2, ..., estadoi)**
    - son posibles transiciones con λ

Autómatas Finitos Deterministas. Definición

- AF Deterministas, AFD's: se definen mediante una quíntupla

  (Σ, Q, f, q_0, F), donde:

  - **Σ: alfabeto de entrada**
  - **Q: conjunto de estados**, es conjunto finito no vacío, realmente un alfabeto para distinguir a los estados
  - **$f: Q \times \Sigma \rightarrow Q$**, **función de transición**
  - **$q_0 \in Q$, estado inicial**
  - **$F \subset Q$: conjunto de estados finales o de aceptación**

Autómatas Finitos Deterministas. Representación

- Se pueden representar mediante tablas de transición o diagramas de transición
  - Tablas de transición:
    - filas encabezadas por los estados ($q_i \in Q$)
    - columnas encabezadas por los símbolos de entrada ($e_i \in \Sigma$)

	e_1	e_2	...	e_n
→ q_1		$f(q_1, e_2)$		
...				
$*q_m$				

Autómatas Finitos Deterministas. Representación

- Se pueden representar mediante tablas de transición o diagramas de transición
  - **Diagramas de transición:**
    - nodos etiquetados por los estados ($q_i \in Q$)
    - arcos entre nodos q_i a q_j etiquetados con e_i si existe $f(q_i,e_i) = q_j$
    - q0 se señala con ➔
    - $q \in F$ se señala con ***** o doble círculo

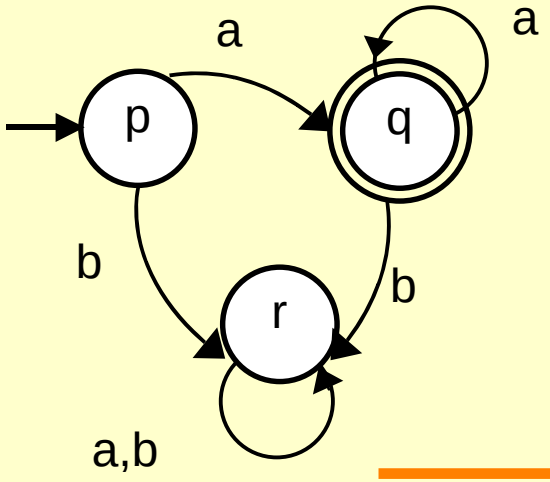
Autómatas Finitos Deterministas. Representación

- Ejemplo: Sea el $AFD_1 = (\{a,b\}, \{p,q,r\}, f, p, \{q\})$ donde f está definida por:

  f(p,a) = q f(p,b) = r
  f(q,a) = q f(q,b) = r
  f(r,a) = r f(r,b) = r

  escribir su tabla de transición y dibujar su diagrama de transición

	a	b
→ p	q	r
*q	q	r
r	r	r



Autómatas Finitos Deterministas. Conceptos Básicos

- **Configuración:** es un par ordenado de la forma (q,w) donde:
  - q: estado actual del AF
  - w: cadena que le queda por leer en ese instante, $w \in \Sigma^*$

  ➔ Configuración inicial: (q_0, t)
  - q_0: estado inicial
  - t: cadena de entrada a reconocer por el AFD $\in \Sigma^*$

  ➔ Configuración final: (q_i,λ)
  - q_i: estado final
  - λ: la cadena de entrada ha sido leída completamente
- **Movimiento:** es el tránsito entre dos configuraciones.

(q,aw) ⟶ **(q',w)**

F(q,a)=q'

Autómatas Finitos Deterministas. Conceptos Básicos

- **AFD como reconocedor de un Lenguaje:**
  - ➔ Cuando un AF transita desde q_0 a un estado final en varios movimientos ➔ se ha producido en **RECONOCIMIENTO** o **ACEPTACIÓN** de la cadena de entrada
  - ➔ Cuando un AF no es capaz de alcanzar un estado final, se dice que el AF **NO RECONOCE** la cadena de entrada y que ésta **NO PERTENECE** al lenguaje reconocido por el AF

Autómatas Finitos Deterministas. Conceptos Básicos

- **Extensión a palabra de la función de transición f:**

  ➔ Ampliar su definición a palabras de Σ^*

  $f: Q \times \Sigma^* \rightarrow Q$

  - a partir de f, que sólo considera palabras de longitud 1, hay que añadir:

    $f(q,\lambda) = q \quad \forall\, q \in Q$

    $f(q, ax) = f(f(q,a),x) \;\forall\, q \in Q,\ a \in \Sigma \quad y \quad x \in \Sigma^*$

Autómatas Finitos Deterministas. Conceptos Básicos

- **En el AFD_1, indicar el resultado de las siguientes expresiones:**

  $f(p,\lambda)$

  $f(p, a^n)$

  $f(p,bb)$

  $f(p,aabbaba)$

  $f(p,baa)$

Autómatas Finitos Deterministas. Conceptos Básicos

- **Lenguaje asociado a un AFD:**
  - Sea un AFD = (Σ, Q, f, q_0, F), se dice que una palabra **x** es **aceptada** o **reconocida** por el AFD si **$(q_0,x) \in F$**
  - Se llama lenguaje asociado a un AFD al conjunto de todas las palabras aceptadas por éste:

$$L = \{ x / x \in \Sigma^* \text{ AND } f(q0,x) \in F \}$$

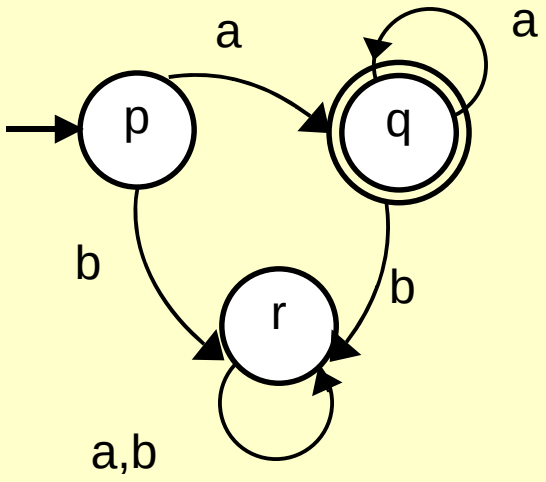
Si $F = \{\} \Rightarrow L = \phi$

$F = Q \Rightarrow L = \Sigma^*$

- Otra definición: $L = \{ x / x \in \Sigma^* \text{ AND } (q_0, x) \rightarrow (q,\lambda) \text{ AND } q \in F\}$

Autómatas Finitos Deterministas. Conceptos Básicos

- En el AFD_1, comprobar que $L(AFD_1) = \{a^n / n>0\}$. Comprobar que si se hace $F = \{r\}$, $L(AFD_1) = \{a^n bx / n \geq 0, x \in \Sigma^*\}$.

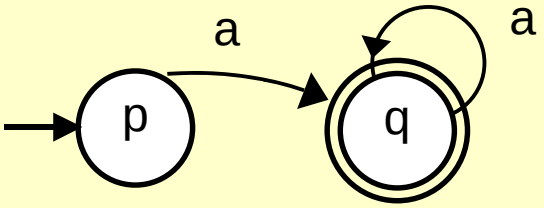


Autómatas Finitos Deterministas. Ejercicios

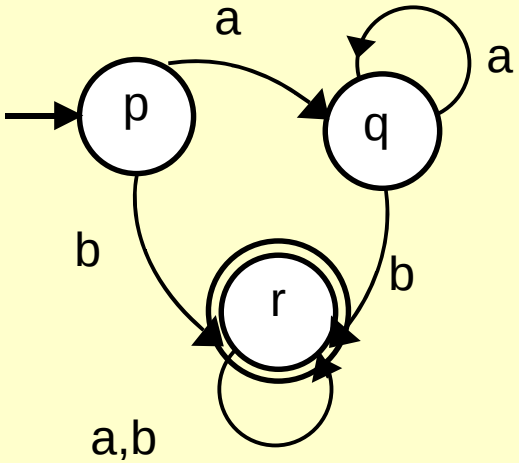
- En el AFD_1, comprobar que $L(AFD_1) = \{a^n / n>0\}$.

  Comprobar que si se hace F = {r},

  $L(AFD_1) = \{a^n bx \ / \ n \geq 0, x \in \Sigma^*\}$.



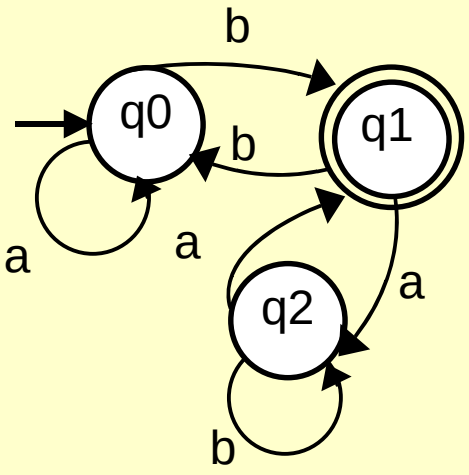
Desde p, con el número de a's que sea, pero al menos una, llego al estado final



Desde p, con una a llego al estado q y desde allí se pueden aceptar tantas a's como sean , luego con una b salto al estado final y allí puedo terminal o reconocer cualquier cadena de a´s y b´s. L= a*b (a*b*)*

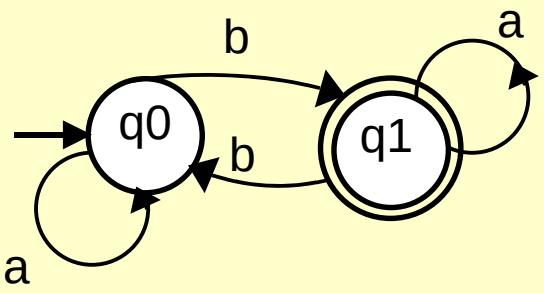
Autómatas Finitos Deterministas. Ejercicios

- En el $AFD_{2,}$, comprobar que $L(AFD_2)$ = {a*b ((b a*b)* (a b*a)*)*}. Es el L aceptado por el AFD_2:



Autómatas Finitos Deterministas. Ejercicios

- En el AFD_3, indicar cual es $L(AFD_3)$



Autómatas Finitos Deterministas. Conceptos Básicos

- **Estados accesibles y Autómatas conexos:**
  - ➔ Sea un AFD = (Σ, Q, f, q_0, F)

    el estado $p \in Q$ es **accesible** desde $q \in Q$ si $\exists\ x \in \Sigma^*$ ($f(q,x) = p$.

    Todo estado es accesible desde sí mismo pues $f(p,\lambda) = p$

    - Sea un AFD, $|Q| = n$, $\forall\ p, q \in Q$ p es accesible desde q sii $\exists\ x \in \Sigma^*$, $|x| < n$ / $f(p,x) = q$
    - Sea un AFD, $|Q| = n$, entonces $L_{AFD} \neq \phi$ Sii el AFD acepta al menos una palabra $x \in \Sigma^*$, $|x| < n$

Autómatas Finitos Deterministas. Ejercicios

- Hallar el AFD conexo equivalente al dado: AF= ({a,b}, {p,q,r,s}, p, f, {q,r,s}), donde f viene dada por la tabla:

	a	b
→ p	r	p
*q	r	p
*r	r	p
*s	s	s

Indicar, además el leguaje reconocido por ambos AFD's

Autómatas Finitos Deterministas. Ejercicios

- Sea el AFD1= ({a,b}, {q1,q2,q3,q4}, q1, {q3}), donde f viene dada por la tabla:

	a	b
→ q_1	q_2	q_4
q_2	q_2	q_3
*q_3	q_4	q_3
q_4	q_4	q_4

Indicar el leguaje que reconoce

- Sea el AFD2= ({0,1}, {q1,q2,q3,q4}, q1, {q2}), donde f viene dada por la tabla:

	0	1
→ q_1	q_4	q_2
*q_2	q_3	q_4
q_3	q_4	q_2
q_4	q_4	q_4

Indicar el leguaje que reconoce

Autómatas Finitos Deterministas. Ejercicios

- Sea el AFD3= ({a,b,c}, {q1,q2,q3,q4,q5}, q1, {q2,q4}), donde f viene dada por la tabla:

	a	b	c
→ q_1	q_2	q_3	q_5
* q_2	q_5	q_5	q_5
q_3	q_5	q_5	q_4
* q_4	q_5	q_5	q_5
q_5	q_5	q_5	q_5

Indicar el leguaje que reconoce

- Sea el AFD4= ({1,2,3}, {q1,q2,q3}, q1, {q2}), donde f viene dada por la tabla:

	1	2	3
→ q_1	q_1	q_1	q_2
* q_2	q_3	q_3	q_3
q_3	q_3	q_3	q_3

Indicar el leguaje que reconoce

Autómatas Finitos Deterministas. Equivalencia

- **Equivalencia y Minimización de AFD's**

  Un AFD es una Máquina Secuencial de Moore, así que mismos teoremas MS's:

  - Equivalencia de estados:

    pEq, $\forall$ p,q $\in$ Q, si $\forall$ x$\in \Sigma^* \Rightarrow$ f(p,x) $\in$ F$\Leftrightarrow$ f(q,x) $\in$ F
  - Equivalencia de orden "n"

    pE_nq, $\forall$ p,q $\in$ Q, si $\forall$ x$\in \Sigma^*$ / $|x| \leq n \Rightarrow$ f(p,x) $\in$ F$\Leftrightarrow$ f(q,x) $\in$ F
  - **pEq** $\Rightarrow$ **pE_nq,** $\forall$ n, p, q$\in$ Q
  - **$pE_nq \Rightarrow pE_kq$,** $\forall$ n > k
  - **$pE_{n+1}q \Leftrightarrow pE_nq$ AND f(p,a) E_nf(q,a)** $\forall$ a $\in \Sigma$
  - **Si $Q/E_n = Q/E_{n+1} \Rightarrow Q/E_n = Q/E_{n+i}$** $\forall$ i = 0, 1, **...**
  - **Si $Q/E_n = Q/E_{n+1} \Rightarrow Q/E_n = Q/E$ conjunto cociente**
  - **Si $|Q/E_0| = 1 \Rightarrow Q/E_0 = Q/E_1$**
  - **n = $|Q| > 1 \Rightarrow Q/E_{n-2} = Q/E_{n-1}$**

Autómatas Finitos Deterministas. Equivalencia

- **Equivalencia y Minimización de AFD's**

Teorema: **pEq** $\Leftrightarrow$ **pE$_{n-2}$ q** $\quad |Q| = n > 1$

pEq Sii $\forall\ x \in \Sigma^*,\ |x| \leq n-2$

$f(p,x) \in F \Leftrightarrow f(q,x) \in F$

n-2 es el valor más pequeño que cumple este teorema

Autómatas Finitos Deterministas. Equivalencia

- Algoritmo para hallar Q/E en AFD's

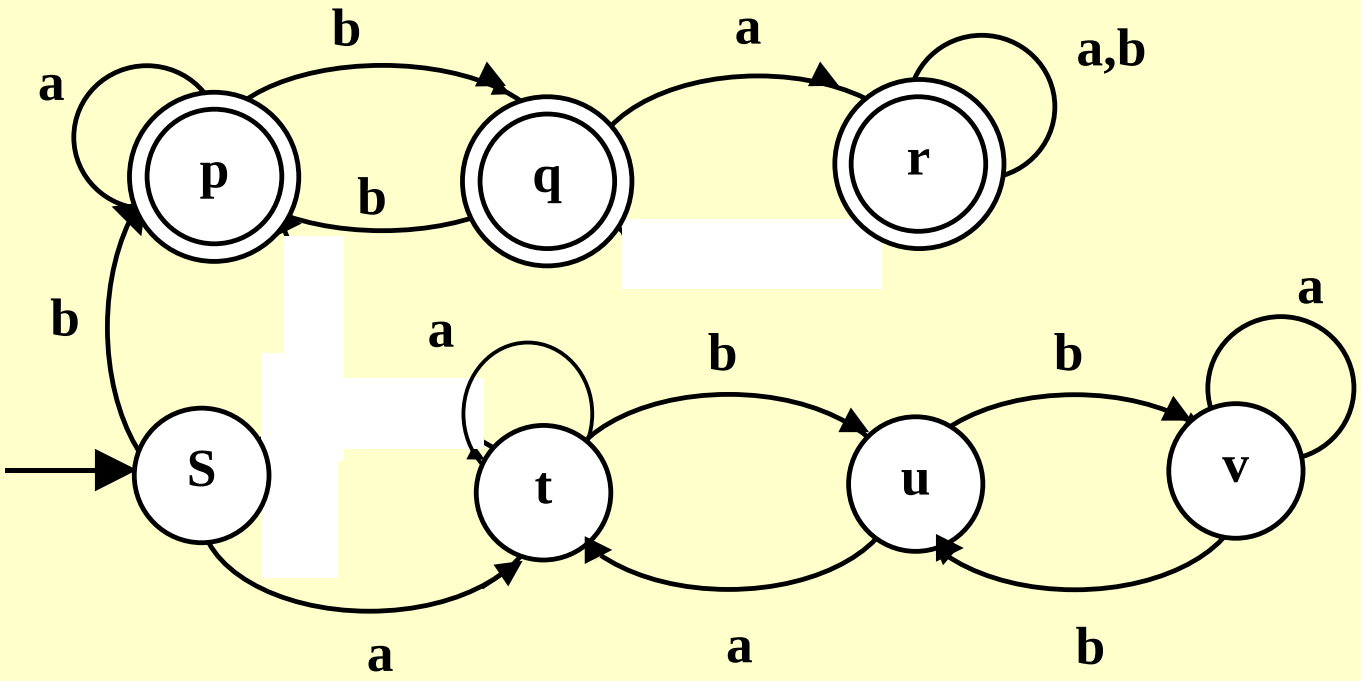
1. **$Q/E_0 = \{ F, \text{no } F\}$** 1ª división en función de si son o no estados finales.
2. **Q/E_{i+1}** : partiendo de **Q/E_i** = {C1,C2,...Cn}, se construye **Q/E_{i+1}**:

   p y q están en la misma clase si:

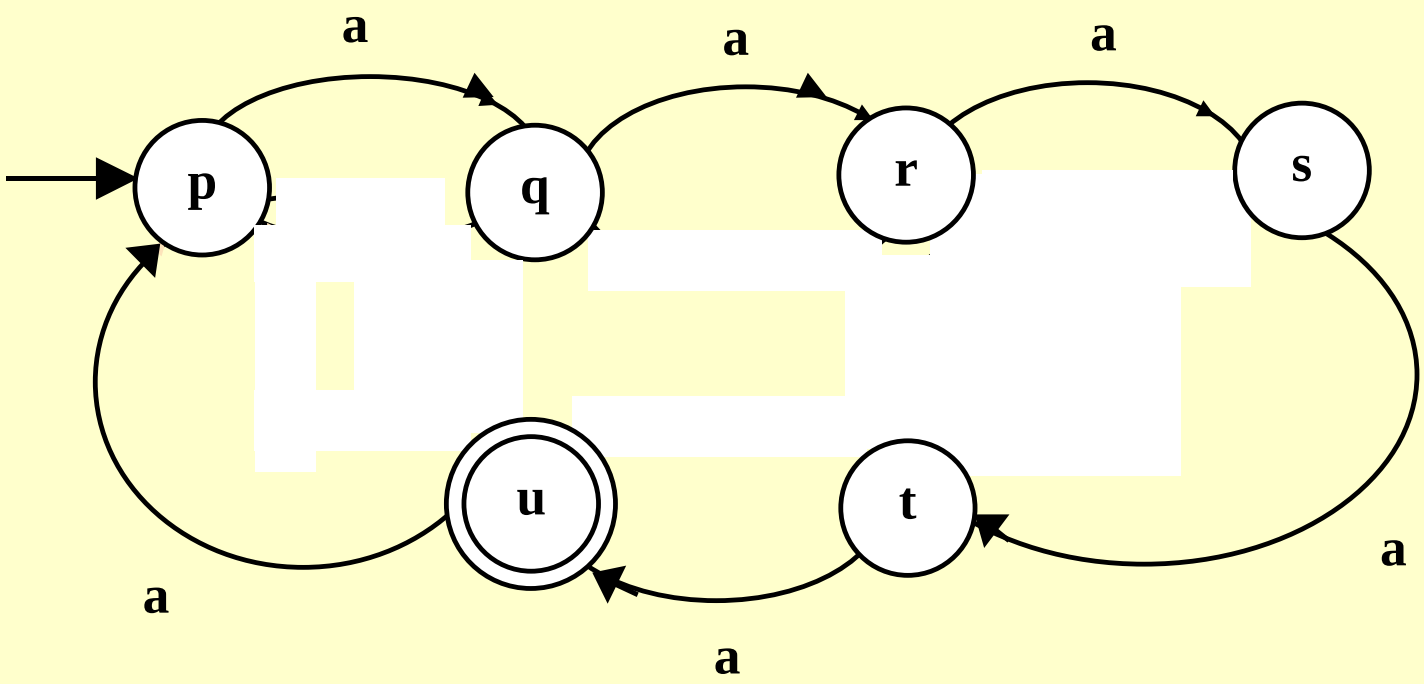
   $p, q \in Ck \in Q/Ei \; \forall \; a \in \Sigma \Rightarrow f(p,a)$ y $f(q,a) \in Cm \in Q/Ei$
3. Si **$Q/E_i = Q/E_{i+1}$** entonces **$Q/E_i = Q/E$**

   Si no, repetir el paso 2 partiendo de Q/E_{i+1}

Ejercicio 1 Hallar el AFD mínimo equivalente



Ejercicio 2 Hallar el AFD mínimo equivalente



Araceli Sanchis - D. de Informática
U. Carlos III de Madrid

Autómatas Finitos Deterministas. Equivalencia

- **Autómatas equivalentes**
  - **Estados equivalentes en AFD's distintos**:

    Sean 2 AFD's: (Σ,Q,f,q_0,F) y $(\Sigma',Q',f''q_0',F')$

    los estados p,q / $p \in Q$ y $q \in Q'$ son equivalentes (pEq)

    si: $f(p,x) \in F \Leftrightarrow f(q,x) \in F' \ \forall \ x \in \Sigma^*$
  - **Estados equivalentes en AFD's distintos**:

    Dos AFD's son equivalentes si reconocen el mismo lenguaje, es decir: Si $f(q_0,x) \in F \Leftrightarrow f(q_0',x) \in F' \ \forall \ x \in \Sigma^* \Rightarrow$ **2 AFD's son equivalentes si lo son sus estados iniciales:**

    $$q_0 \, E \, q_0'$$

Autómatas Finitos Deterministas. Equivalencia

- **Autómatas equivalentes, comprobación**

  1. **Suma directa de AFD's**
  2. **Teorema**
  3. **Algoritmo para comprobar la equivalencia de AFDs**

Autómatas Finitos Deterministas. Equivalencia

- **Autómatas equivalentes, comprobación**

  1. **Suma directa de AFD's:**

     Sean 2 AFD's:

     $A1 = (\Sigma, Q_1, f_1, q_{01}, F_1)$

     $A2 = (\Sigma', Q_2, f_2, q_{02}, F_2)$ } $Q_1 \cap Q_2 = \phi$

     Se llama suma directa de A1 y A2 al AF A:

     $A = A1 + A2 = (\Sigma, Q_1 \cup Q_2, f, q_0, F_1 \cup F_2)$

donde:

- q_0 es el estado inicial de uno de los AF's
- f: $f(p,a) = f1(p,a)$ si $p \in Q1$ } $\forall a \in \Sigma$
  $f(p,a) = f2(p,a)$ si $p \in Q2$

Autómatas Finitos Deterministas. Equivalencia

- **Autómatas equivalentes, comprobación**

  2. **Teorema:** sean A1, A2 / $Q_1 \cap Q_2 = \phi$, $|Q_1| = n_1$, $|Q_2| = n_2$

     A_1 E A_2 si q01 E q02 en $A = A_1 + A_2$ es decir, si A1 y A2

     aceptan las mismas palabras x / $|x| \leq n_1+n_2-2$

     además, n_1+n_2-2 es el valor mínimo que cumple el teorema

Autómatas Finitos Deterministas. Equivalencia

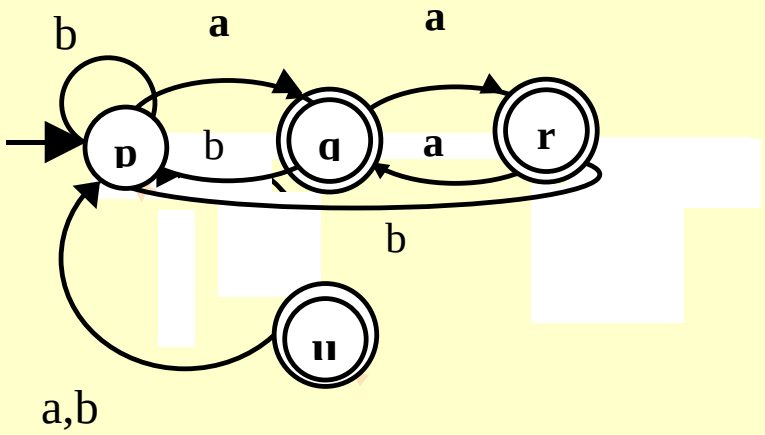
- **Autómatas equivalentes, comprobación**

  3. Algoritmo para comprobar la equivalencia de AFDs

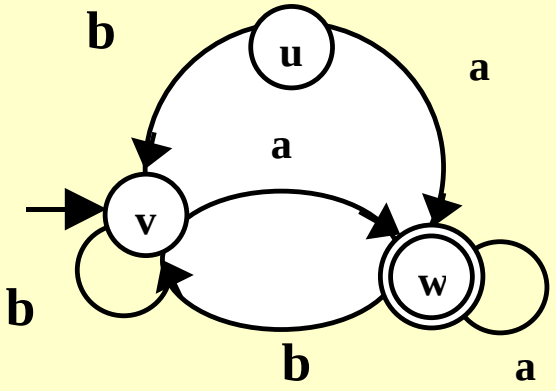
  1. Se hace la suma directa de los dos AFD's
  2. Se hace Q/E del AFD suma
  3. Si los dos estados iniciales están en la misma clase de equivalencia de Q/E $\Rightarrow$ los 2 AFD's son equivalentes

Autómatas Finitos Deterministas. Equivalencia

- Ejercicio:



A1



A2

AFDs. Isomorfismos

Sean A1 = $(\Sigma, Q_1, f_1, q_{01}, F_1)$ y A2 = $(\Sigma', Q_2, f_2, q_{02}, F_2)$ tales que $|Q_1| = |Q_2|$

Se dice que **A1 y A2 son isomorfos**, si existe una aplicación biyectiva **$i : Q_1 \rightarrow Q_2$** que cumple:

1 $i(q_{01}) = q_{02}$, es decir, los estados iniciales son correspondientes
2 $q \in F_1 \Leftrightarrow i(q) \in F_2$ es decir, los estados finales son correspondientes
3 $i(f_1(q,a)) = f_2(i(q),a)\ \forall\ a \in \Sigma\ \ q \in Q_1$

En definitiva, a cada estado le corresponde otro equivalente que solo se diferencia en el nombre de sus estados.

Dos AFDs isomorfos también son equivalentes y reconocen el mismo lenguaje

AFDs. Minimización

Sea el AFD, $A = (\Sigma, Q, f, q_0, F)$:

1. Partir del AFD conexo: eliminar estados inaccesibles desde el estado inicial
2. Construir Q/E del autómata conexo
3. El AFD mínimo, salvo isomorfismos, es:

   $A' = (\Sigma, Q', f', q_0', F')$

donde:

$Q' = Q/E$

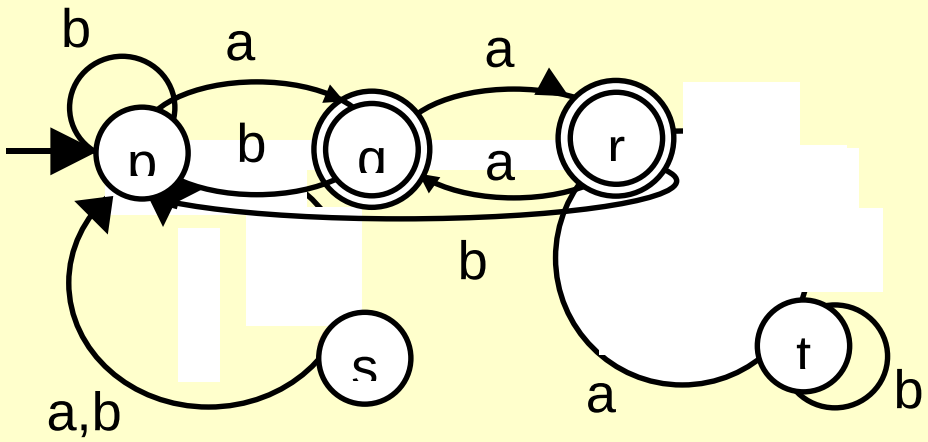
f' se construye: $f'(C_i, a) = C_j$ si $\exists q \in C_i$, $p \in C_j$ / $f(q,a) = p$

$q_0' = C0$ si $q0 \in C_0$, $C_0 \in Q/E$

$F' =$ {C / C contiene al menos un estado de F($\exists$ un $q \in F$ tal que $q \in C$)}

COROLARIO: 2 AFD's son equivalentes si sus AF mínimos respectivos son isomorfos.

Ejercicio 4: Hallar el AFD mínimo equivalente al dado:



AF asociado a una G3 (I)

Sea la G3 lineal izquierda: $G = (\Sigma_T, \Sigma_N, S, P)$

Se construye a partir de ella el AF: $A = (\Sigma_T, \Sigma_N \cup \{p,q\}, f, p, \{S\})$

donde: $p, q \notin$ ni a Σ_T, Σ_N

f se define:

1) $f(U,t) = V$ si $V ::= U\ t \in P$

2) $f(p,t) = V$ si $V ::= t \in P$

3) $f(U,t) = q \quad \forall t \in \Sigma_T / V ::= U\ t \notin P$

4) $f(p,t) = q \quad \forall t \in \Sigma_T / V ::= t \notin P$

5) $f(q,t) = q \quad \forall t \in \Sigma_T$

AF asociado a una G3 (II)

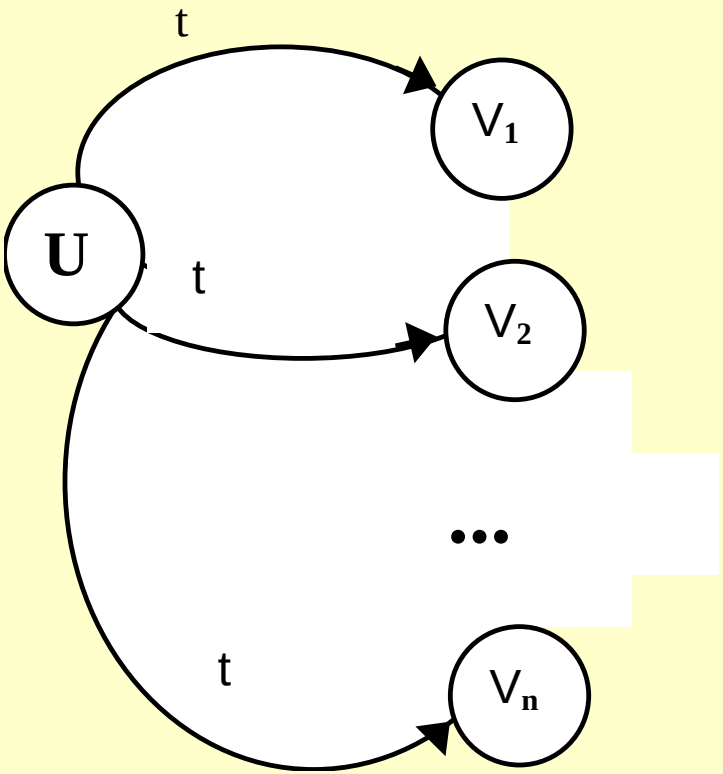
- Esta definición no asegura un AF determinista, ya que puede ocurrir:

  V1 ::= U t

  V2 ::= U t

  ...

  V3 ::= Ut



AF asociado a una G3 (III)

Ejercicio 5

Sea la G3 lineal izquierda:

G= ({0,1}, {S,U,V}, S, P)

donde P = {S ::= U0 / V1

U ::= S1 / 1

V ::= S0 /0}

Hallar el AFD mínimo que reconozca el lenguaje generado por dicha G.

Para ello: 1) Hallar el AF (Determinista en este caso)

2) Minimizarlo.

3) Hallar L(G) y L (AF) y ver si coinciden

4) repetir el ejercicio eliminando el axioma inducido

AF no Deterministas, AFND's

- Definiciones (son equivalentes):

1) AFND = (Σ, Q, f, q_0, F), donde

   f: Q x Σ^* $\rightarrow$ Q es No determinista, es decir, por ejemplo:

   f(p,a) = {q,r} y f(p,λ) = {q,r}

2) AFND = $(\Sigma, Q, f, q_0, F, T)$, donde:

   Σ,Q, q_0,F : idem que en AFD

   f : Q x Σ $\rightarrow$ *P*(Q): conjunto de las partes de Q

   T : Relación definida sobre pares de elementos de Q:

   pTq = (p,q) $\in$T si está definida la transición f(p, λ)=q

AF no Deterministas, AFND's

- Ejemplo: Sea el AFND siguiente:

A = ({a,b}, {p,q,r,s}, f,p, {p,s}, T= {(q,s), (r,r), (r,s), (s,r)}) donde f:

f(p,a) = {q} f(p,b) = {}

f(q,a) = {p,r,s} f(q,b) = {p,r}

f(r,a) = {} f(r,b) = {p,s}

f(s,a) = {} f(s,b) = {}

cuya tabla de transiciones es:

	a	**b**	**λ**
→*p	q		
q	p,r,s	p,r	s
r		p,s	r,s
*s			r

Extensión a palabra de f en AFND's

- Se define a partir de f una función de transición f'' que actúa sobre palabras de Σ^*
- Es una aplicación: f'': $Q \times \Sigma^* \rightarrow P(Q)$

Considerando:

1) $f''(q,\lambda) = \{p / qT^*p \ \forall q \in Q\}$

2) sea x = a1a2a3...an n>0

$f''(q,x) = \{p / p$ es accesible desde q por medio de la palabra $\lambda^*a1\ \lambda^*a2\ \lambda^*a3\ \lambda^*...\ \lambda^*an\ \lambda^*, \forall q \in Q\}$

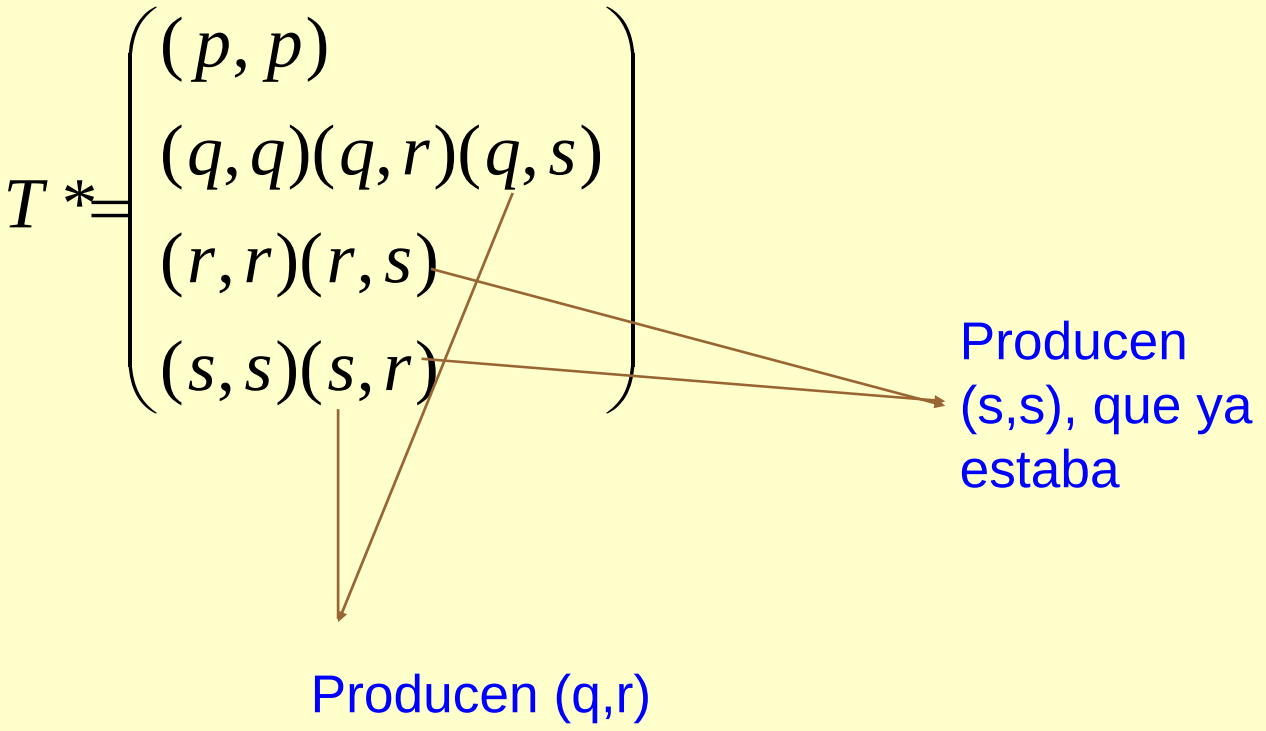
es idéntica a x

Calculo de T*

- Para calcular f' es necesario extender las transiciones con una λ a λ*.
- Para ello existe el método formal de las matrices booleanas, o el método de la matriz de pares (estado, estado). Emplearemos esta última:
  1. Se construye una matriz con tantas filas como estados.
  2. En la 1ª columna se coloca el par correspondiente al estado en cuestión, es decir, pej (p,p) puesto que cada estado es accesible desde si mismo.
  3. En las columnas siguientes se añaden las transiciones λ definidas en el AFND, considerando si el hecho de añadirlas permite extender alguna transición más. Pej. Si existe la transición (q,r) y se añade la transición (r,s), habrá que añadir asimismo, la transición (q,s).
  4. Cuando no sea posible añadir ningún par más, se habrá terminado T*

Cálculo de T*. Ejemplo (I)

- Sea el AFND: A, definido anteriormente, donde T= {(q,s), (r,r), (r,s), (s,r)} se trata de calcular T*



Calculo de T*. Ejemplo (II)

- Se extiende la tabla de transición anterior para contener T*, insertando una nueva columna correspondiente a λ*

	a	**b**	λ	**λ***
→*p	q			p
q	p,r,s	p,r	s	q,s,r
r		p,s	r,s	r,s
*s			r	r,s

Calculo de T*. Ejemplo (III)

- Y ahora se calcula la tabla de transición correspondiente a f', cambiando las transiciones con a por λ*aλ* y las de b por λ*bλ*.

	λ*aλ*	**λ*bλ***
→*p	q,r,s	Φ
q	p,r,s	p,r,s
r	Φ	p,r,s
*s	Φ	p,r,s

Lenguaje aceptado por un AFND

- Una palabra **x $\in\Sigma^*$ es aceptada por un AFND** si:
  - f' (q0,x) y F tienen al menos un elemento común, es decir, que f'(q0,x) contiene al menos un estado final.
  - El conjunto de todas las palabras aceptadas por un AFND es el lenguaje aceptado por ese AFND.

$$L_{AFND} = \{x \,/\, x \in \Sigma^* \text{ y } \exists\, qo \rightarrow F\} = \{x \,/\, x \in \Sigma^* \text{ y } f'(qo,x) \cap F \neq \Phi\}$$

  - Al ser un AFND puede haber más de un camino, x es aceptada sólo con que uno de los caminos lleve a un estado final.
  - $\lambda \in L_{AFND}$ si :

    1 qo $\in$ F

    2 $\exists$ un estado final, q $\in$ F, tal que está en relación T* con qo:

    qo T* q

AFD equivalente a un AFND

- Dado un AFND siempre es posible encontrar un AFD que reconozca el mismo lenguaje:
  - ➔ El conjunto de los L_{AFND} = al conjunto de los L_{AFD}.
  - ➔ Un AFND no es más potente que un AFD, sino que un AFD es un caso particular de AFND.
- Paso de AFND a AFD:
  - ➔ Sea el AFND A = (Σ, Q,f,qo,F,T). Se define a partir de él B, que es un AFD:

    B = (Σ, Q',f',q'o,F'), tal que:

    Q'= P(Q) conjunto de las partes de Q que incluye a Q y a Φ.

    q0' = f'(qo,λ) (todos los estados que tengan relación T* con q0).

    F' = {C / C $\in$ Q' y $\exists$ q $\in$ C / q $\in$ F}

    f'(C,a) = {C' /C' = $\bigcup_{q \in C} f'(q,a)$ }

AFD equivalente a un AFND. Casos

- **Caso 1: AFND sin transiciones λ**
  - ➔ **Dado el AFND descrito por la tabla siguiente, hallar el AFD equivalente.**

	a	**b**
→p	q	
q	r,q	
*r		r

Comprobar que el AFD acepta el mismo lenguaje.

AFD equivalente a un AFND. Casos

- **Caso 2: AFND con transiciones λ**

  Dado el AFND descrito por la tabla siguiente, hallar el AFD equivalente.

	a	**b**	**c**	**λ**
→ p	p	q		q
q	q	p,r		r
r			s	p
*s	s			

AF asociado a una G3

- Se ha visto el procedimiento para obtener el AF que aceptaba el lenguaje descrito por una G3 LI, sin embargo, ese procedimiento no siempre conduce a un AFD, lo habitual es:

  G3 → AFND → AFD

- Ejercicio 1: Sea la G LI: G= ({0,1}, {S,U}, S,{S ::= U0, U ::= U0 /S1 /0})

  Hallar el AFD correspondiente.

- Ejercicio 2: Sea la G LI: G= ({0,1}, {S,U}, S,{S ::= U0 / λ, U ::= U0 /S1 /0})

  Hallar el AFD correspondiente.

AF asociado a una G3 (cuando es LD)

1 **De AF → G3:**

Sea el AF, A = (Σ, Q, qo, f, F), existe una G3 LD tal que

L(G3LD) = L(A)

Se construye:

$\Sigma_T = \Sigma$

$\Sigma_N = Q$

S = qo

P= {f(q,e) =q' → **si q' no es estado final** → q::= eq'

q' ∈ F y f(q,e) = q' → q::= e (+ q::= eq')}

q0 ∈ F → q0 ::= λ

si f(q, λ) = q' → q ::= q'}

AF asociado a una G3 (cuando es LD)

1 **De AF → G3: Ejemplo**

Sea el AF descrito por la siguiente tabla, hallar la G3 LD que genera el lenguaje por ella descrito. Comprobar que los lenguajes son iguales

	0	**1**
→A	A	C
B	A	C
C	C	B

AF asociado a una G3 (cuando es LD)

2 **De G3 $\rightarrow$ AF:**

Sea la G3 LD, G = (Σ_T, Σ_N S, P), existe un AF, A, tal que L(G3LD) = L(A)

Se construye:

$\Sigma = \Sigma_T$

$Q = \Sigma_N \cup \{F\}$ $F \notin \Sigma_N$

qo = S

F = {F}

f: $A ::= aB \rightarrow f(A,a) = B$

$A::= a \rightarrow f(A,a) = F$

$S ::= \lambda \rightarrow f(S, \lambda) = F$

AF asociado a una G3 (cuando es LD)

2 **De G3 → AF : Ejemplo**

Sea la G3 LD hallar el AF correspondiente.

G = ({d,c}, {A,S,T}, A, {A ::= cS, S::= d/cS/dT, T::= dT/d})