3. Autómatas Finitos

Araceli Sanchis de Miguel
Agapito Ledezma Espino
José A. Iglesias Martínez
Beatriz García Jiménez
Juan Manuel Alonso Weber

Grado Ingeniería Informática
Teoría de Autómatas y Lenguajes Formales

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad
Carlos III de Madrid
www.uc3m.es

Objetivos

- Definir el concepto de Autómata Finito Determinista (AFD).
- Definir el concepto de Autómata Finito No Determinista (AFND).
- Establecer las equivalencias entre AFD.
- Convertir un AFND en un AFD.
- Minimizar AFD.
- Identificar el tipo de lenguaje aceptado por un AFND.

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

- Autómatas Finitos
  - Autómatas Finitos Deterministas (AFD)
    - AFD como reconocedores de lenguajes
    - Equivalencia y minimización de AFD
  - Autómatas Finitos No Deterministas (AFND)
- Equivalencia entre AFD y AFND

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

Autómatas Finitos

- 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, ..., estado i).
    - son posibles transiciones con λ

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

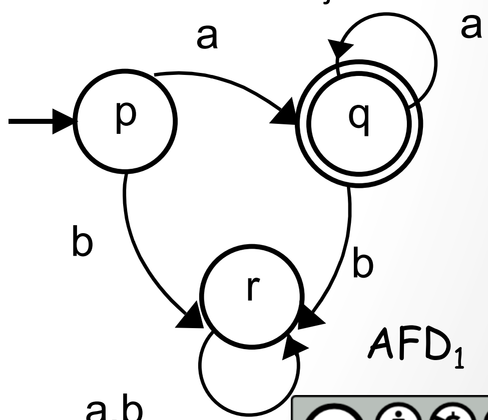
Universidad Carlos III de Madrid
www.uc3m.es

Autómatas Finitos. Representación

- Se pueden representar mediante:
  1. Diagramas de transición o
  2. Tablas de transición

1. Diagramas de transición:

- Nodos etiquetados por los estados (qi ∈ Conjunto de estados)
- Arcos entre nodos q_i a q_j etiquetados con e_i (e_i es un símbolo de entrada) si existe la transición de q_i, a q_j con e_i
- El estado inicial se señala con →
- El estado final se señala con * o doble círculo



AFD$_1$

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

Autómatas Finitos. Representación

2. Tablas de transición:

- Filas encabezadas por los estados ($q_i \in$ Conjunto de estados)
- Columnas encabezadas por los símbolos de entrada ($e_i \in$ alfabeto de entrada)

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

Símbolos de Entrada

Estados

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

- Autómatas Finitos
  - Autómatas Finitos Deterministas (AFD)
    - AFD como reconocedores de lenguajes
    - Equivalencia y minimización de AFD
  - Autómatas Finitos No Deterministas (AFND)
- Equivalencia entre AFD y AFND

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Autómatas Finitos Deterministas

- 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: $Q x \Sigma \rightarrow Q$, función de transición**
  - **$q_0 \in Q$, estado inicial**
  - **$F \subset Q$: conjunto de estados finales o de aceptación**

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

Autómatas Finitos Deterministas

- Ejemplo: El AFD_1 = ({0,1}, {p,q,r}, f, p, {q}), donde f está definida por:

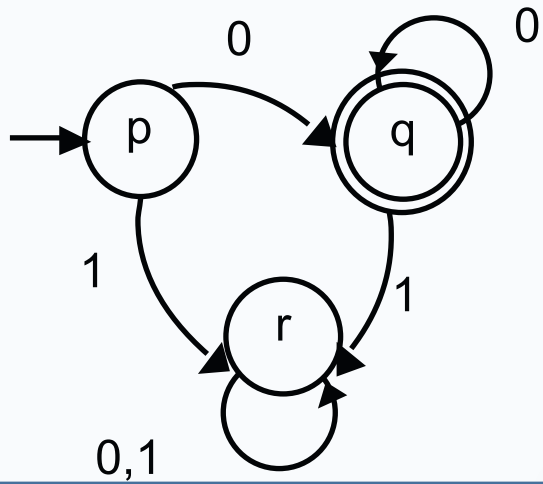
  f(p,0) = q   f(p,1) = r

  f(q,0) = q   f(q,1) = r

  f(r,0) = r   f(r,1) = r

Tiene la tabla de transición y el diagrama de estados siguientes:

	0	1
→p	q	r
*q	q	r
r	r	r



- Autómatas Finitos
  - Autómatas Finitos Deterministas (AFD)
    - AFD como reconocedores de lenguajes
    - Equivalencia y minimización de AFD
  - Autómatas Finitos No Deterministas (AFND)
- Equivalencia entre AFD y AFND

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD como reconocedores de Lenguajes

- Cuando un AF transita desde q_0 a un estado final en varios movimientos, se ha producido el 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

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. 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, $w \in \Sigma^*$

Universo del Discurso

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

$$(q,aw) \xrightarrow[f(q,a)=q']{} (q',w) \qquad a,w \in \Sigma^*$$

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Conceptos Básicos

Extensión a palabra de la función de transición f, f':

Es la ampliación de la definición de **f** a palabras de Σ^*, i.e. $w \in \Sigma^*$

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

  a partir de f, que sólo considera palabras de longitud 1,

  hay que añadir:

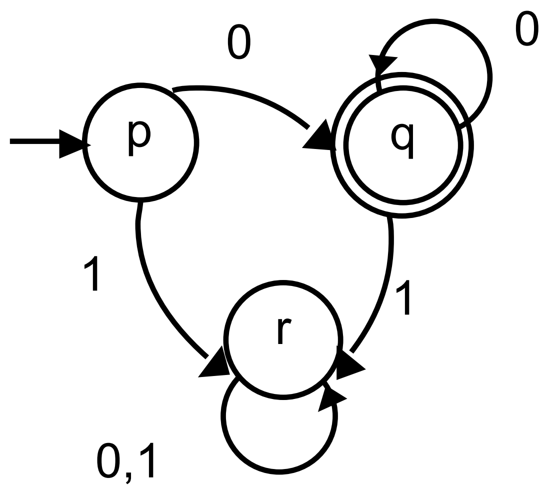
- $f'(q,\lambda) = q \quad \forall q \in Q$
- $f'(q, a \cdot x) = f'(f(q, a), x) \quad \forall q \in Q,\ \boxed{\mathbf{a \in \Sigma, x \in \Sigma^*}}$

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFD. Conceptos Básicos

- En el AFD_1 (de la figura), indicar el resultado de las siguientes expresiones:
  - f'(p,λ)
  - f'(p, 0^n)
  - f'(p,11)
  - f'(p,0011010)
  - f' (p,100)



A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. 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 $f'(q_0,x) \in F$
- Se llama **lenguaje asociado a un AFD** al conjunto de todas las palabras aceptadas por éste:

  $$\mathbf{L = \{ x \,/\, x \in \Sigma^* \text{ and } f'(q_0,x) \in F \}}$$

  - Si $F = \{\,\} = \emptyset \Rightarrow L = \emptyset$
  - Si $F = Q \Rightarrow L = \Sigma^*$

- Otra definición:

  $$\mathbf{L = \{ x \,/\, x \in \Sigma^* \text{ and } (q_0, x) \rightarrow (q,\lambda) \text{ and } q \in F\}}$$

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

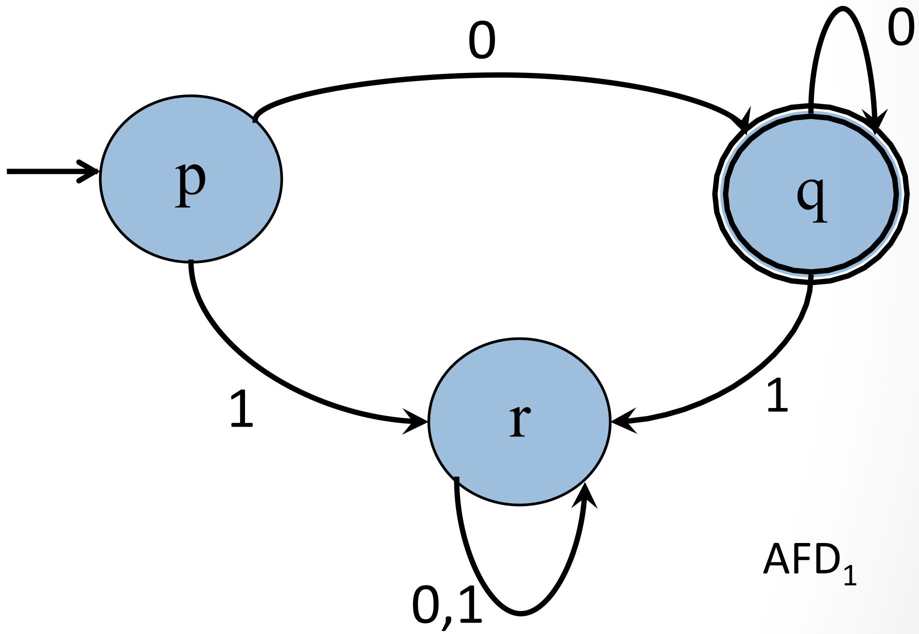
Universidad Carlos III de Madrid
www.uc3m.es

AFD. Conceptos Básicos

En el AFD_1

- Cuál es $L(AFD_1)$ = ¿?

- Y si se hace F = {r}, cuál es $L(AFD_1)$ = ¿?



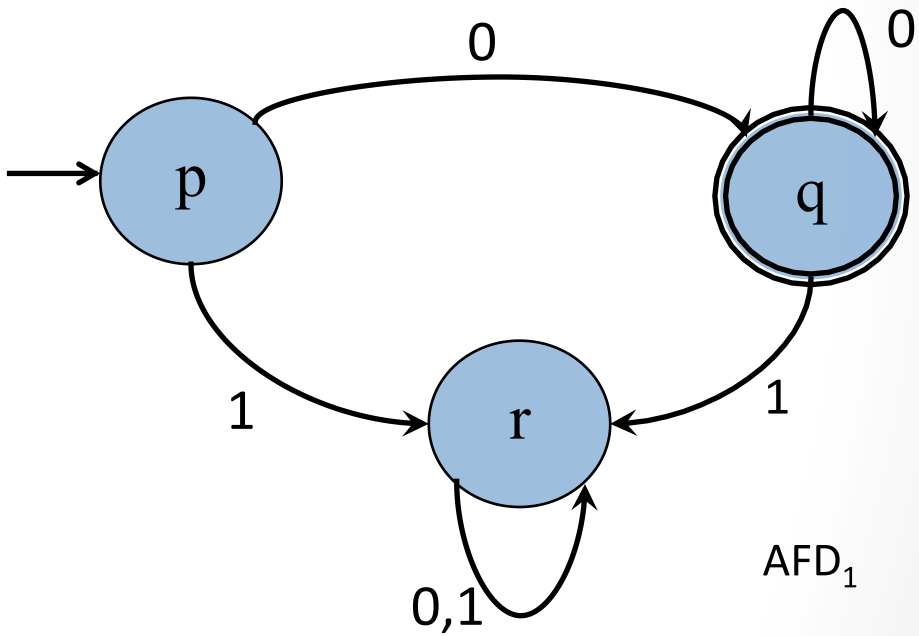
AFD_1

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Conceptos Básicos

En el AFD_1

- Cuál es $L(AFD_1)$ = **$\{0^n \ / \ n > 0\}$.**



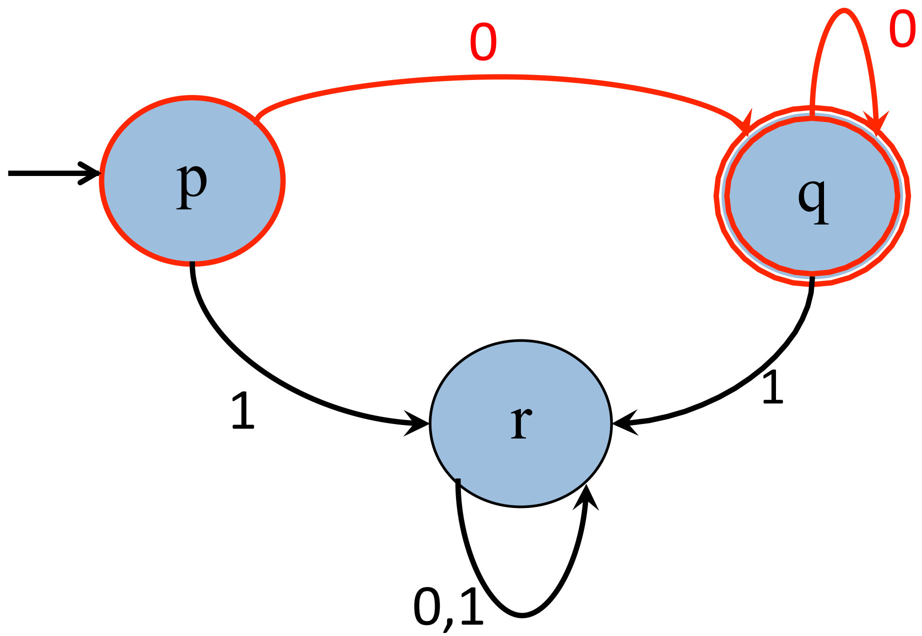
AFD_1

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Conceptos Básicos

En el AFD_1

- Cuál es $L(AFD_1)$ = **$\{0^n / n > 0\}$**. **Comprobación**



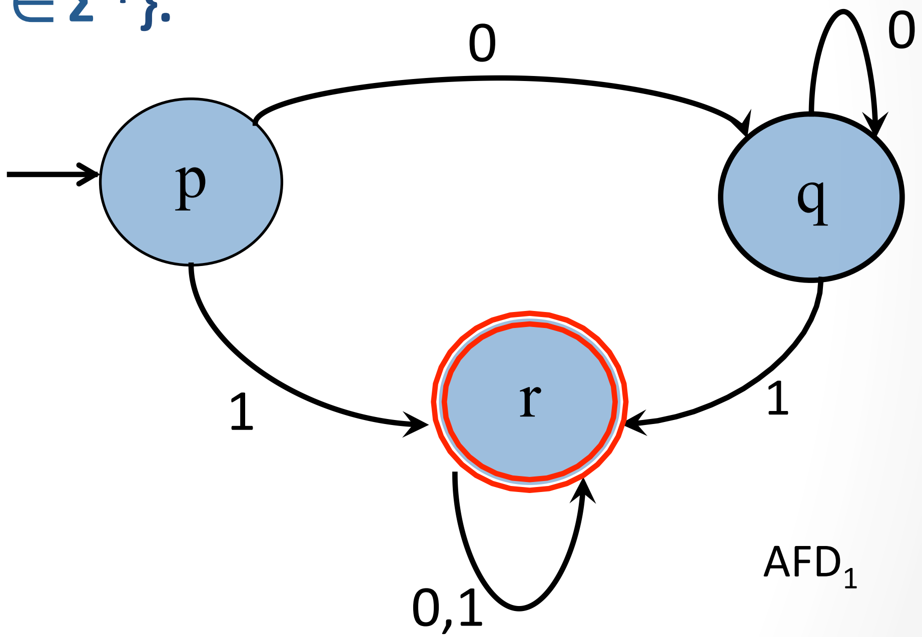
Desde p, con el número de 0's que sea, pero siempre al menos uno, se llega al estado final

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Conceptos Básicos

En el AFD_1

- Y si se hace F = {r},
  $L(AFD_1)$ = **{0^n1x / n ≥ 0, x ∈ Σ *}.**



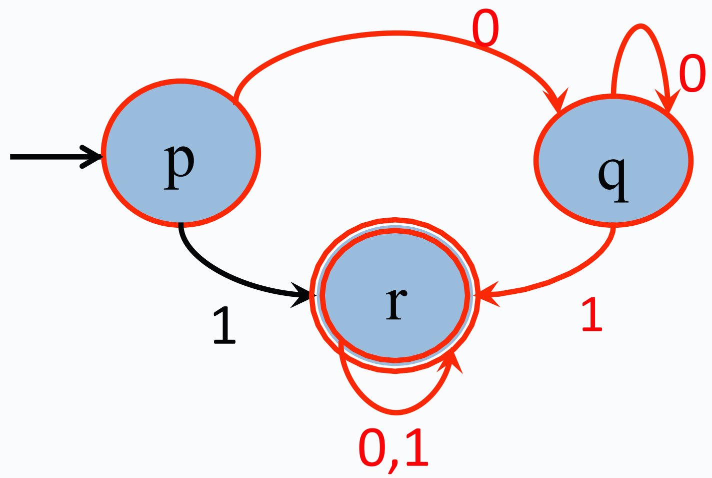
AFD_1

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Conceptos Básicos

En el AFD_1,

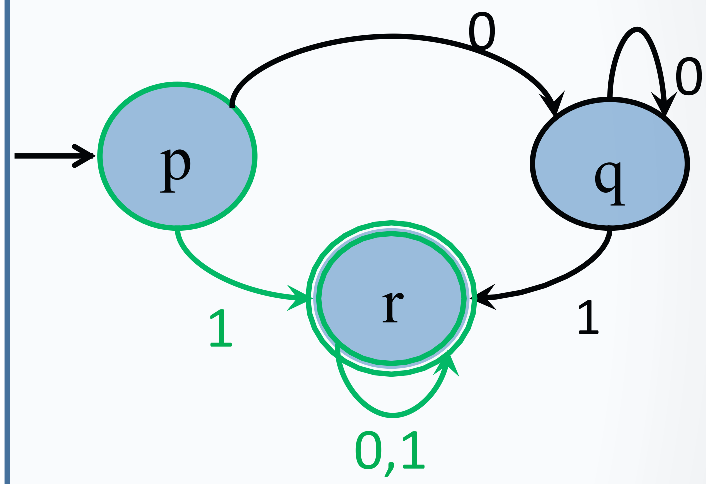
- comprobar que si se hace F = {r}, $L(AFD_1)$ = **$\{0^n 1x \;/\; n \geq 0, x \in \Sigma^*\}$.**



Desde p, con un “0” llego al estado q y desde allí se pueden aceptar tantos 0s como sean.

Luego con un 1 salto al estado final y allí puedo terminar o reconocer cualquier cadena de 0s y 1s.

Expresión regular: $L_A = 0^+1\,(0+1)^*$



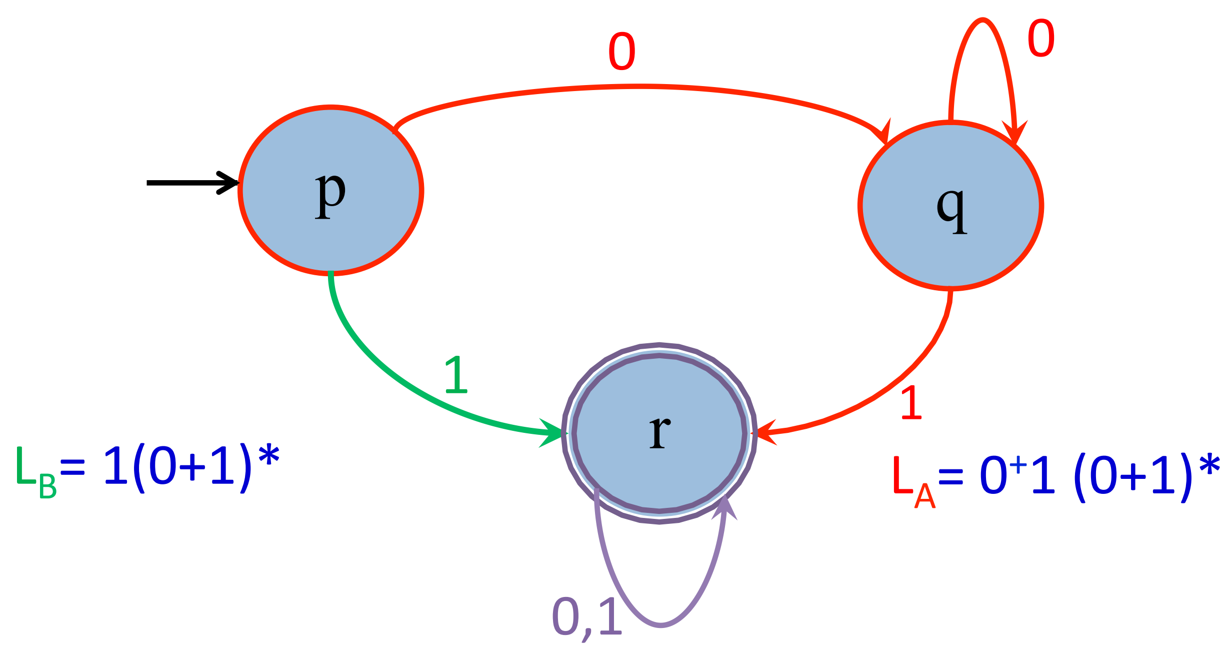
Expresión regular: $L_B = 1(0+1)^*$

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Conceptos Básicos

En el AFD_1,

- comprobar que si se hace F = {r}, $L(AFD_1)$ = **$\{0^n1x \ / \ n \geq 0, x \in \Sigma^*\}$.**



Expresión regular $L_A \cup L_B = 0^*1(0+1)^*$

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFD. 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 $\exists\, x \in \Sigma^*$ $f'(q,x) = p$. En otro caso se dice que INACCESIBLE.
  Todo estado es accesible desde sí mismo pues $f'(p,\lambda) = p$

Teoremas:

- teorema 3.2.2, libro 1 de la bibliografía.
  Sea un AFD, $|Q| = n$, $\forall\, p, q \in Q$ p es accesible desde q **sii** $\exists\, x \in \Sigma^*$, $|x| < n$ / $f'(p,x) = q$
- teorema 3.2.3, libro 1 de la bibliografía
  Sea un AFD, $|Q| = n$, entonces $L_{AFD} \neq \phi$ **sii** el AFD acepta al menos una palabra $x \in \Sigma^*$, $|x| < n$
  Nota: sii= "si y solo si"

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Conceptos Básicos

Estados accesibles y **Autómatas conexos:**

Sea un AFD = (Σ, Q, f, q_0, F). Diremos que el autómata es conexo si todos los estados de Q son accesibles desde q_0

Dado un autómata no conexo, podemos obtener a partir de él otro autómata equivalente conexo eliminando los estados inaccesibles desde el estado inicial. Los autómatas reconocen el mismo lenguaje.

<u>Eliminación de estados inaccesibles.</u>

- *¿Qué algoritmo, para ser implementado en un programa, se podría implementar para marcar los accesibles?*

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Ejercicios

- Hallar el AFD conexo equivalente al dado: AF= ({0,1}, {p,q,r,s}, p, f, {q,r,s}), donde f viene dada por la tabla.
  - Se eliminan todos los estados innacesibles y todos las transiciones (i.e. arcos) que salen desde dichos estados innacesibles.

	0	1
p	r	p
*q	r	p
*r	r	p
*s	s	s

- Indicar, además el leguaje reconocido por ambos AFD's (original y conexo).

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

- Autómatas Finitos
- Autómatas Finitos Deterministas (AFD)
- AFD como reconocedores de lenguajes
- **Equivalencia y minimización de AFD**
- Autómatas Finitos No Deterministas (AFND)
- Equivalencia entre AFD y AFND

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFD. Equivalencia y Minimización

- Es posible tener varios autómatas que reconozcan el mismo lenguaje.
- Para todo autómata se puede obtener un autómata equivalente (i.e. reconoce el mismo lenguaje) donde el número de estados del autómata sea el mínimo.
- **¿Por qué interesa obtener el mínimo?** **(Apartado 4.4 Libro 2 bibliografía)**

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Equivalencia y Minimización

¿Por qué interesa obtener el AFD mínimo? (Ap. 4.3 y 4.4 Libro 2 bibliograf)

- Se dispone de un descriptor del lenguaje (lenguaje regular): gramática tipo 3, AFD, AFND, expresión regular.
- Se plantean problemas de decisión:
  - ¿El lenguaje descrito es vacio?
  - ¿Existe una determinada cadena w en el lenguaje descrito?
  - ¿Dos descripciones de un lenguaje describen realmente el mismo lenguaje?
    - Nota: usualmente los lenguajes son infinitos, con lo que no es posible plantear la pregunta y recorrer el conjunto INFINITO de cadenas.

- Los algoritmos para responder a las dos primeras preguntas son sencillos. ¿Pero y para la última pregunta ?

  ¿Dos descripciones de un lenguaje describen realmente el mismo lenguaje?

  Consecuencia de esta comprobación: es necesario obtener el AFD mínimo equivalente

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Equivalencia y Minimización

Teoremas:

- Equivalencia de estados:

  p E q, donde $p,q \in Q$, si $\forall\ x \in \Sigma^*$ se verifica que

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

- Equivalencia de orden (o de longitud) "n"

  p E_n q, donde $p,q \in Q$, si $\forall\ x \in \Sigma^* \ / \ |x| \leq n$ se verfica que

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

E y E_n son relaciones de equivalencia.

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Equivalencia y Minimización

Equivalencia de estados – Casos particulares:

- **E_0** , x palabra $|x| \leq 0 \Rightarrow x = \lambda$ se verifica que

  p E_0 q, $\forall$ p,q $\in$ Q, si $\forall$ x$\in \Sigma^*$ / $|x| \leq 0$ se verfica que

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

  x es lamba

  $f'(p,x) = f'(p,\lambda) = p$ (por definición de f')

  $f(p, \lambda) \in F \Leftrightarrow f(q, \lambda) \in F$ -> $p \in F \Leftrightarrow q \in F$

  Todos los estados finales de son E_0 equivalentes.

  $\forall$ p,q $\in$ F se cumple que p E_0 q

  $\forall$ p,q $\in$ Q - F se cumple que p E_0 q

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Equivalencia y Minimización

Equivalencia de estados – Casos particulares:

- **E_1**, x palabra $|x| \leq 1$, ($x \in \Sigma$) se verifica que

  $p\ E_1\ q$, $\forall\ p,q \in Q$, si $\forall\ x \in \Sigma^* \ /\ |x| \leq 1$ se verifica que

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

  x es lamba o símbolo del alfabeto.

  $f'(p,x) = f'(p,a) = f(p,a)$ ó $f'(p,x) = f'(p,\lambda) = p$ (por definición de f')

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

  Partiendo de p y q con una sola transición se debe llegar a un estado final para ambos casos o uno no final para ambos casos.

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFD. Equivalencia y Minimización

- **Propiedades**

  Nota: en estas expresiones matemáticas, "n" *no* significa |Q|

  - Lema: $p\ E\ q \Rightarrow p\ E_n\ q,\ \forall\ n,\ p, q \in Q$
  - Lema: $p\ E_n\ q \Rightarrow p\ E_k\ q,\ \forall\ n > k$
  - Lema: $p\ E_{n+1}\ q \Leftrightarrow p\ E_n\ q$ and $f(p,a)\ E_n\ f(q,a)\ \forall\ a \in \Sigma$

- **Teorema**: $p\ E\ q \Leftrightarrow p\ E_{n-2}\ q$, donde $n = |Q| > 1$ **Aquí "n" *SÍ* significa |Q|**

  (Teorema 5.1 (pag 117 libro 4 bibliografia))

  $p\ E\ q$ sii $\forall\ x \in \Sigma^*,\ |x| = m \leq n-2$ se verifica que $f(p,x) \in F \Leftrightarrow f(q,x) \in F$

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

  (n-1 sí lo cumple, pero n-3 no se garantiza que se cumpla)

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Equivalencia y Minimización

"E" es una relación de equivalencia. ¿Qué significa Q/E?

- **Q/E es una partición de Q,**
- **$Q/E = \{C_1, C_2, \ldots, C_m\}$, donde $C_i \cap C_j = \emptyset$**
  - $p \; E \; q \Leftrightarrow (p,q \in C_i)$, por lo tanto

    $\forall \; x \in \Sigma^*$ se verifica que $f'(p,x) \in C_i \Leftrightarrow f'(q,x) \in C_i$

Nota: en libro 1 biblio, $p,q \in Ci$ se representa por $p = q = C_i$;

- Para la relación de orden n
  - E_n: $Q/E_n = \{C_1, C_2, \ldots, C_m\}$, C_i intersección $C_j = \emptyset$
  - $p \; E_n \; q \Leftrightarrow p,q \in C_i$;
  - por lo tanto $\forall \; x \in \Sigma^*$, $\underline{|x| \leq n}$ se verifica que $f'(p,x) \in C_i \Leftrightarrow f'(q,x) \in C_i$

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Equivalencia y Minimización

Propiedades. (Lemas)

- Lema: Si $Q/E_n = Q/E_{n+1} \Rightarrow Q/E_n = Q/E_{n+i}\ \forall\ i = 0, 1, \ldots$
- Lema: Si $Q/E_n = Q/E_{n+1} \Rightarrow Q/E_n = Q/E$ conjunto cociente
- Lema: Si $|Q/E_0| = 1 \Rightarrow Q/E_0 = Q/E_1$
- Lema: $n = |Q| > 1 \Rightarrow Q/E_{n-2} = Q/E_{n-1}$
- $p\ E_{n+1}\ q \Leftrightarrow (\ p\ E_n\ q$ and $f(p,a)\ E_n\ f(q,a)\ \forall\ a \in \Sigma\ )$

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Equivalencia y Minimización

Interpretación lemas anteriores:

El objetivo es obtener la partición Q/E, puesto que será el autómata mínimo, sin estados equivalentes .

- En cuanto se obtienen dos particiones consecutivas $Q/E_k = Q/_{Ek+1}$, se para.
- Para obtener Q/E, hay que empezar por Q/E_0, Q/E_1, etc.
- Para obtener Q/E, hay que obtener Q/E_{n-2} en el peor caso, ya que si se obtiene $Q/E_{n-k} = Q/E_{n-k+1}$, con k>=3, se habría obtenido ya Q/E.
- El lema $p\ E_{n+1}\ q \Leftrightarrow p\ E_n\ q$ and $f(p,a)\ E_n\ f(q,a)\ \forall\ a \in \Sigma$, permite es extender la equivalencia de orden n desde E_0 y E_1

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Equivalencia y Minimización

- **Teorema**:

$$pEq \Leftrightarrow pE_{n-2}q \text{ donde } |Q| = n > 1 \ (**)$$

Es decir, $p\,E\,q$ 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

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFD. Equivalencia y Minimización

Algoritmo formal para obtener Q/E:

1. **Q/E_0 = { F, no F}**

   1ª división en función de si son o no estados finales.

2. **Q/E_{i+1}**

   partiendo de **Q/E_i** = $\{C_1, C_2, \ldots C_n\}$, se construye **Q/E_{i+1}**:

   p y q están en la misma clase si:

   $p, q \in C_k \in Q/E_i\ \forall\ a \in \Sigma \Rightarrow f(p,a)$ y $f(q,a) \in C_m \in Q/E_i$

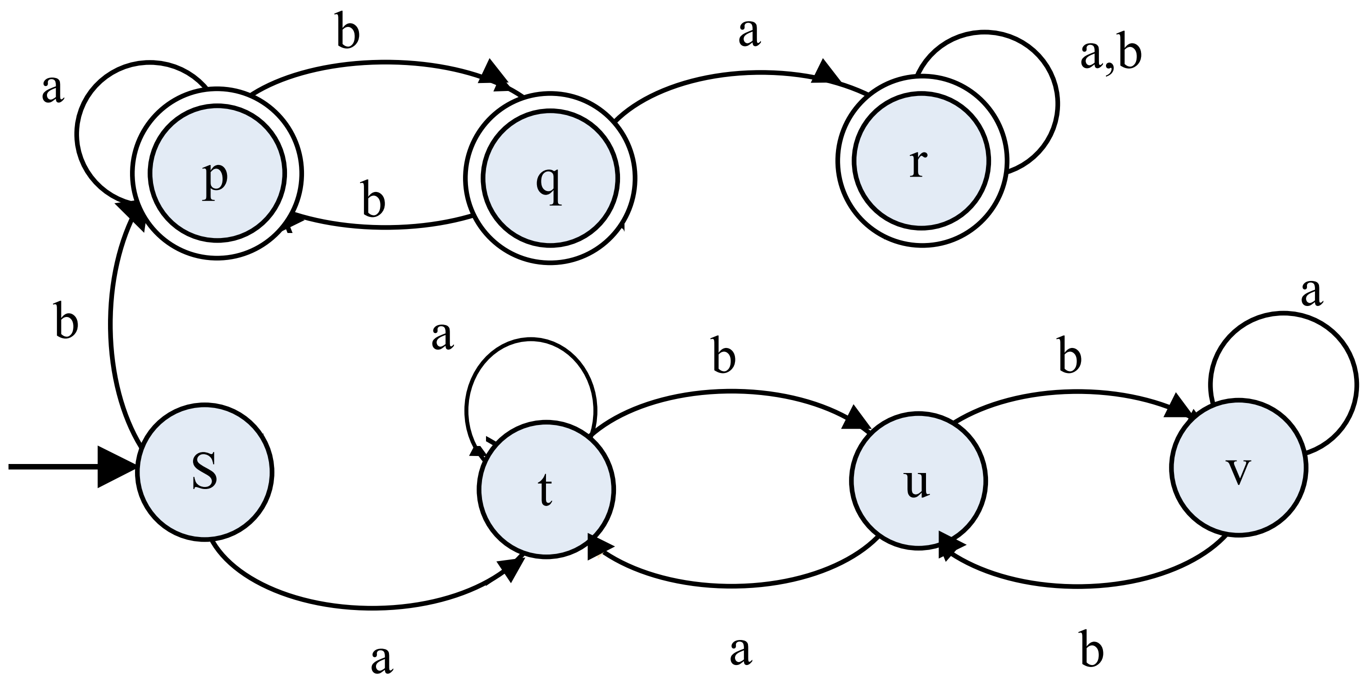
3. Si **$Q/E_i = Q/E_{i+1}$** entonces **$Q/E_i = Q/E$**

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

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Equivalencia y Minimización

- Ejercicio: Hallar el AFD mínimo equivalente



A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. 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 se verifica que $f(p,x) \in F \Leftrightarrow f''(q,x) \in F' \ \forall x \in \Sigma^*$

- **Estados equivalentes en AFD's distintos:**
  - Dos AFD's son equivalentes si reconocen el mismo lenguaje, es decir: Si $f(q_0,x) \in F \Leftrightarrow f(q_0',x) \in F' \ \forall x \in \Sigma^*$. Es decir:
  - **Dos AFD's son equivalentes si lo son sus estados iniciales: q_0 E q_0'**

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Equivalencia

¿Qué es la suma directa de 2 AFD's?

Sean 2 AFD's:

$$\left.\begin{array}{l} A1 = (\Sigma, Q_1, f_1, q_{01}, F_1) \\ A2 = (\Sigma', Q_2, f_2, q_{02}, F_2) \end{array}\right\}$$

$\boxed{\textbf{Donde } Q_1 \cap Q_2 = \phi}$

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 $p \in Q1$

$f(p,a) = f2(p,a)$ si $p \in Q2$

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Equivalencia

- **Teorema:** (el teorema (**) aplicado a la suma directa de dos autómatas):

  sean A1, A2 / $Q_1 \cap Q_2 = \phi$, $|Q_1| = n_1$, $|Q_2| = n_2$

  A_1 E A_2 si q_{01} E q_{02} en $A = A_1+A_2$

  Es decir, si A_1 y A_2 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

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFD. Equivalencia

Autómatas equivalentes, comprobación:

Algoritmo para comprobar la equivalencia de AFDs

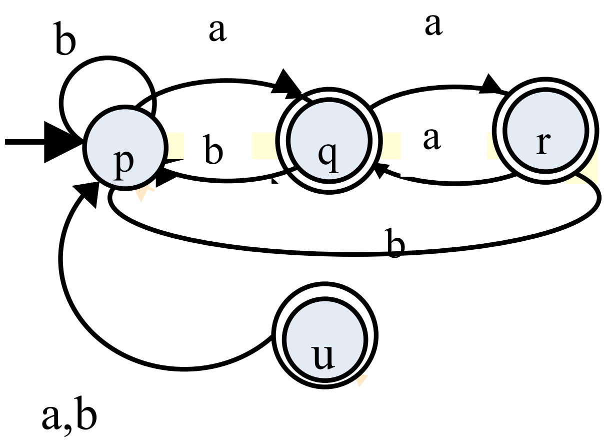
1. Se hace la suma directa de los dos AFD's
2. Se hace Q/E del AFD suma
3. Si los dos estados iniciales están en la misma clase de equivalencia de Q/E $\Rightarrow$ los 2 AFD's son equivalentes

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

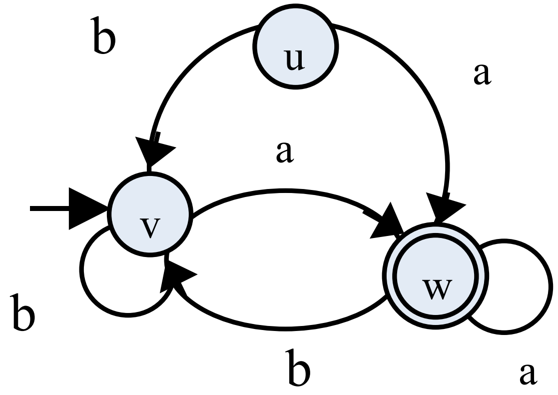
Universidad Carlos III de Madrid
www.uc3m.es

AFD. Equivalencia

- **Ejercicio:** Comprobar que los autómatas A1 y A2 son equivalentes.



A1



A2

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFD. Equivalencia

- Sean dos autómatas:
  - $A1 = (\Sigma, Q_1, f_1, q_{01}, F_1)$ 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.**

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFD. Minimización

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

1. Partir del AFD conexo, i.e. 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 $\exists q \in C_i$, $p \in C_j$ / $f(q,a) = p$

$q_0' = C_0$ 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 $q \in C)\}$

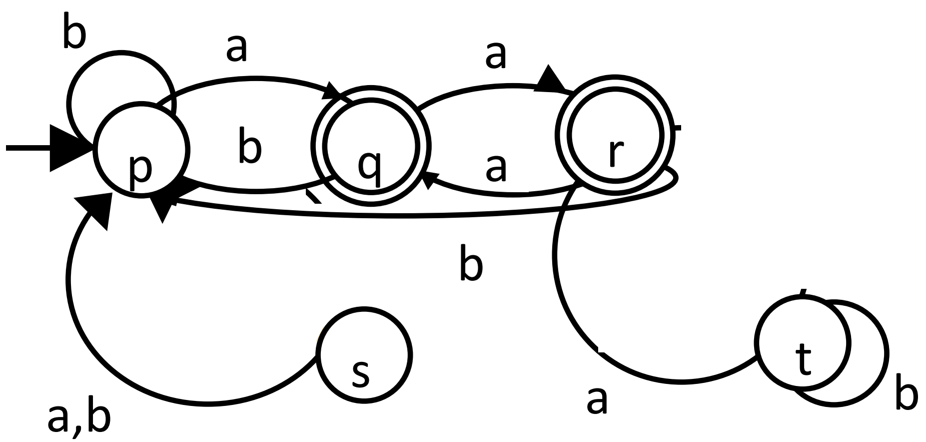
COROLARIO:

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

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD. Ejercicio

Hallar el AFD mínimo equivalente al dado:



A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

- Autómatas Finitos
- Autómatas Finitos Deterministas (AFD)
- AFD como reconocedores de lenguajes
- Equivalencia y minimización de AFD
- **Autómatas Finitos No Deterministas (AFND)**
- Equivalencia entre AFD y AFND

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Autómatas Finitos No Deterministas

Definiciones de AFND :

1. AFND = (Σ, Q, f, q_0, F), donde
   - $f: Q \times (\Sigma \cup \lambda\ \}\rightarrow Q$ es No determinista,

     es decir, por ejemplo: $f(p,a) = \{q,r\}$ y $f(p,\lambda) = \{q,r\}$

2. AFND = $(\Sigma, Q, f, q_0, F, T)$, donde
   - $f : Q \times \Sigma \rightarrow P(Q)$: 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 $f(p, \lambda)=q$

     <u>Nota: "T" es la definición formal de la transición λ</u>

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

Autómatas Finitos No Deterministas

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) = {}

La tabla de transiciones es

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

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFNDs. Función de Transición extendida a palabras

- Se define a partir de f, una función de transición f'', que actúa sobre palabras de Σ^*;

  f' es la función de transición sobre palabras.

- Es una aplicación: $f'': Q \times \Sigma^* \rightarrow P(Q)$. Donde :

1. $f''(q,\lambda) = \{p \;/\; qT^*p \;\; \forall q \in Q\}$ (T* se define más adelante)

   donde se cumple que $q \in f'(q,\lambda)$

2. sea $x = a_1a_2a_3...a_n$, $n>0$

   $f''(q,x) = \{p$ / p es accesible desde q por medio de la palabra $\underbrace{\lambda^*a_1 \lambda^*a_2 \lambda^*a_3 \lambda^*... \lambda^*a_n \lambda^*}_{\text{es idéntica a x}} \;\forall q \in Q\}$

<u>Lectura recomendada: Apartado 3.3.4 del primer libro de la bibliografía básica</u>

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFNDs. Función de Transición extendida a palabras

Calculo de T*

Sea AFND = $(\Sigma, Q, f, q_0, F, T)$.

- Para calcular f' es necesario extender las transiciones con una λ a λ^*, es decir calcular T* del AFND= $(\Sigma, Q, f, q_0, F, \underline{\boldsymbol{T}})$
- Para ello existe el método formal de las matrices booleanas, o el método de la matriz de pares (estado, estado).

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFNDs. Función de Transición extendida a palabras

Calculo de T*. Método de la matriz de pares de estados

- Se construye una matriz con tantas filas como estados.
- En la 1ª columna se coloca el par correspondiente al estado en cuestión, es decir, por ej. (p,p) puesto que cada estado es accesible desde si mismo.
- 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 λ (q,r) y se añade la transición λ (r,s), habrá que añadir asimismo, la transición (q,s).
- Cuando no sea posible añadir ningún par más, se habrá terminado T*

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFNDs. Función de Transición extendida a palabras

Calculo de T*. Ejemplo 1:

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

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

$$T^* = \begin{pmatrix} (p,p) & & \\ (q,q) & (q,s) & \boxed{(q,r)} \\ (r,r) & (r,s) & \\ \boxed{(s,s)} & (s,r) & \end{pmatrix}$$

Producen (s,s), que ya estaba

Producen (q,r)

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFNDs. Función de Transición extendida a palabras

Calculo de T*. Ejemplo 2:

- 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

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFNDs. Función de Transición extendida a palabras

Calculo de T*. Ejemplo 3:

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

→

	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

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFND. 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.

Formalmente:

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

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFND. Lenguaje aceptado por un AFND

- Al ser un AFND, desde q_o puede haber más de un camino para la palabra "***x*** ", y "***x***" es aceptada sólo con que uno de los caminos lleve a un estado final.

- Además:

  $\lambda \in$ L AFND si :

  - $q_o \in F$ ó
  - **∃** un estado final, $q \in F$, tal que está en

    relación T* con q_o (q_o T* q)

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

- Autómatas Finitos
- Autómatas Finitos Deterministas (AFD)
- AFD como reconocedores de lenguajes
- Equivalencia y minimización de AFD
- Autómatas Finitos No Deterministas (AFND)
- Equivalencia entre AFD y AFND

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

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 $A = (\Sigma, Q, f, q_0, F, T)$.
  - Se define a partir de A el AFD B, donde:

    $B = (\Sigma, Q', f\hat{}, q_o', F')$, tal que:

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

    $q_0' = f'(q_o, \lambda)$ (f' extensión a palabra de f, i.e. todos los estados que tengan relación T* con q_0).

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

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

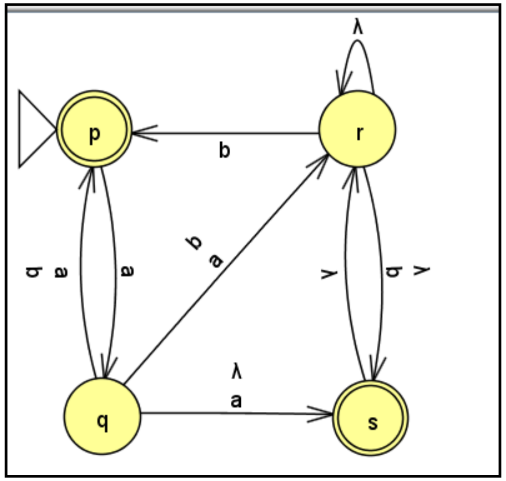
A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFD equivalente a un AFND. Ejemplo

- Obtener el AFD correspondiente al siguiente AFND

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



- Pasos:
  1. Eliminar transiciones λ
     a) Determinar λ* (el cierre de las transiciones λ, T*)
     b) Obtener la tabla sin transiciones λ
  2. Aplicar algoritmo de creación de nuevos estados que pertenecen a P(Q), añadiendo su transiciones.

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD equivalente a un AFND. Ejemplo

1. Eliminar transiciones λ
   a) Determinar λ* (el cierre de las transiciones λ) a partir de la tabla de transiciones.

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

↓

	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	***s***,r

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFD equivalente a un AFND. Ejemplo

1. Eliminar transiciones λ
   a) Determinar λ* (el cierre de las transiciones λ)

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

   b) Obtener la tabla sin transiciones λ
   (transiciones con entrada λ*a λ*, para cada elemento, a, del alfabeto Σ)

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

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad Carlos III de Madrid
www.uc3m.es

AFD equivalente a un AFND. Ejemplo

2. Aplicar algoritmo de creación de nuevos estados que pertenecen a P(Q), añadiendo su transiciones.

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

	a	b
→*p	{q,r,s}	Ø
~~q~~	~~p,r,s~~	~~p,r,s~~
~~r~~	~~Ø~~	~~p,r,s~~
~~*s~~	~~Ø~~	~~p,r,s~~
{q,r,s}	**{p,r,s}U Ø U Ø**	**{p,r,s}U {p,r,s}U{p,r,s}**

↓

	λ*aλ*	λ*bλ*
→*p	{q,r,s}	Ø
~~q~~	~~p,r,s~~	~~p,r,s~~
~~r~~	~~Ø~~	~~p,r,s~~
~~*s~~	~~Ø~~	~~p,r,s~~
{q,r,s}	{p,r,s}	{p,r,s}

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

AFD equivalente a un AFND. Ejemplo

2. Aplicar algoritmo de creación de nuevos estados que pertenecen a P(Q), añadiendo su transiciones.

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

	λ*aλ*	λ*bλ*
→*p	{q,r,s}	Ø
~~q~~	~~p,r,s~~	~~p,r,s~~
~~r~~	~~Ø~~	~~p,r,s~~
~~*s~~	~~Ø~~	~~p,r,s~~
{q,r,s}	{p,r,s}	{p,r,s}

	λ*aλ*	λ*bλ*
→*p	{q,r,s}	Ø
~~q~~	~~{p,r,s}~~	~~{p,r,s}~~
~~r~~	~~Ø~~	~~{p,r,s}~~
~~*s~~	~~Ø~~	~~{p,r,s}~~
{q,r,s}	{p,r,s}	{p,r,s}

	λ*aλ*	λ*bλ*
→*p	{q,r,s}	Ø
~~q~~	~~p,r,s~~	~~p,r,s~~
~~r~~	~~Ø~~	~~p,r,s~~
~~*s~~	~~Ø~~	~~p,r,s~~
{q,r,s}	{p,r,s}	{p,r,s}
{p,r,s}	**{q,r,s} U Ø U Ø**	**Ø U {p,r,s} U {p,r,s}**

	λ*aλ*	λ*bλ*
→*p	{q,r,s}	Ø
{q,r,s}	{p,r,s}	{p,r,s}
{p,r,s}	**{q,r,s}**	**{p,r,s}**

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

3. Autómatas Finitos

Araceli Sanchis de Miguel
Agapito Ledezma Espino
José A. Iglesias Martínez
Beatriz García Jiménez
Juan Manuel Alonso Weber

Grado Ingeniería Informática
Teoría de Autómatas y Lenguajes Formales

A. Sanchis, A. Ledezma, J.A. Iglesias, B. García, J. M.Alonso

Universidad
Carlos III de Madrid
www.uc3m.es