

# Apuntes-Tema-3-MC.pdf



**LosCocos**



**Modelos de Computación**



**3º Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**  
**Universidad de Granada**

**¡HAZTE  
BILINGÜE!**

**958 261 159**

**615 834 365**

**academia-granada.es**

**CLASES DE INGLÉS**

**B1 B2**  
**C1** **BASIC  
English**  
(NIVEL PRINCIPIANTE)

**CLASES DE FRANCÉS**

**B1 B2**  
DELF DELF



**PUERTA  
REAL**

Academia de Enseñanza

B2  
FIRSTC1  
ADVANCED

Practica online tu examen de inglés

www.testandtrain.es

Código:

WUOT&amp;T

-5%  
D.T.O.

## TEMA 3: PROPIEDADES DE LOS CONJUNTOS REGULARES

### LEMA DE BOMBEO

Demostrar que un lenguaje es regular.

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

$$- |uv| \leq n$$

$$- |v| \geq 1$$

$$- (\forall i \geq 0) uv^i w \in L$$

Es condición NECESARIA

PERO NO SUFICIENTE

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

para aplicarlo:

1. Suponemos que sí es regular
2. Si fuera regular debería satisfacer el lema de bombeo, entonces debemos buscar una contradicción

Ej: Demostrar que  $L = \{p^i c p^{-i} \mid p \in \{0,1\}^*\}$  es regular

$$\begin{array}{l} p = 000 \\ p = 010 \end{array} \parallel u \in L; u = p c p^{-1} = 000 c 000$$

$$011 c 110$$

"Si es posible, construye un autómata finito que acepte las cadenas de este lenguaje. Si no es posible, demostrar por qué".

R: Para este lenguaje, aplicar el lema de Bombeo

(Además ver que no es un lenguaje regular debido a la relación sintáctica que existe  $(p, p^{-1})$ , por tanto no se puede construir un autómata finito).

**Ej**  $L = \{0^i 1^j : j \geq 0\}$

- Buscamos una cadena para encontrar la contradicción, que no se cumpla una de las 3 condiciones del lema
- $\overbrace{000\dots 0}^u \overbrace{1111}^u$ ;  $u = 0^k$ ,  $v = 0^l$ ,  $w = 0^{u-k-l} 1^u$  con  $l \geq 1$
- Con esto, la propiedad tercera no se cumple  
 $L = 2$ ,  $uv^2w = 0^k 0^{2l} 0^{u-k-l} 1^u = 0^{u+l} 1^u \notin L$

**Ej**  $L = \{uv \in \{0,1\}^* : v = -v\}$  // Palindromos

- Suponemos que el lenguaje es regular
- $z = 0^u 1^u 0^u$  (3)  $u + u \geq u$  Se cumple

Tenemos:

$$u = 0^k, v = 0^l, w = 0^{u-k-l}, \text{ con } l \geq 0$$

Si aumentamos el  $u$  de  $v$ , el resultado satisface las condiciones del lenguaje

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

**Ej**  $L = \{0^i 1^j 2^i 1^j : i, j \geq 0\}$

Suponemos lenguaje regular

-  $\exists u \in \{0,1,2\}^* z = 0^u 1^u 2^u 0^u \in L$

$|z| = 4u \geq u$  Se cumple

$$z = 0^k 0^l 0^{u-k-l} 1^u 2^u 0^u$$

Para  $i=0$

$$z = 0^k 0^{u-k-l} 1^u 2^u 1^u = 0^{u-l} 1^u 2^u 1^u \notin L$$



# Test&Train

Practica online  
tu examen de inglés  
[www.testandtrain.es](http://www.testandtrain.es)

**-5%**  
DTO.

Código:

**WUOT&T**



**B2**  
FIRST

**C1**  
ADVANCED

### 3ª PREGUNTA EXAMEN

$$L = \{ h p c p^{-1} / p \in \{0, 1\}^* \}$$

1º Suponemos lenguaje regular

2º  $\exists u \in \mathbb{N}$  tal que

$$z = 0^u \in 0^u = uvw$$

$$|z| = 2u + 1 \geq u$$

3.  $|uv| \leq u \rightarrow uv = 0^k; k \leq u$

luego  $w = 0^{u-k} \in 0^u$

4.  $|v| \geq 1 \rightarrow v = 0^l; l \geq 1$

luego  $u = 0^{k-l}$

5. CONTRADICCIÓN:  $\forall i \geq 0, uv^i w \in L$

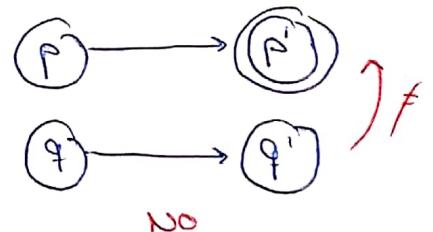
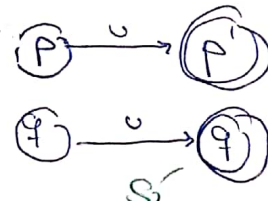
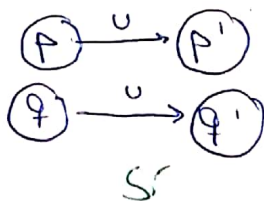
Para  $i=0$   $uv^0 w = 0^{k-l} 0^{u-k} \in 0^u = 0^{u-l} \in 0^u \in L$

ya que  $l \geq 1$

### MINIMIZACIÓN DE AUTOMATAS

1º Eliminamos los estados inaccesibles

2º Detectar estados indistinguibles



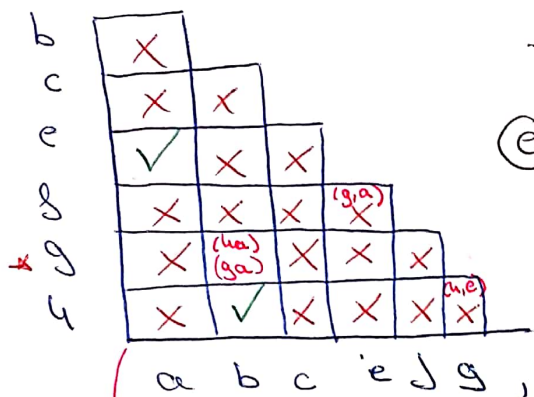
Para el algoritmo montamos la matriz triangular

- abajo todos los elementos menos el último
- arriba todos los elementos menos el primero

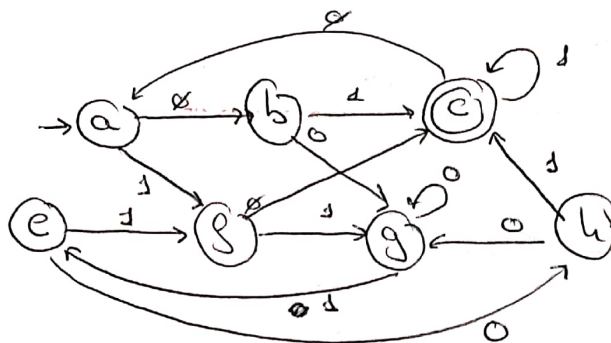
El algoritmo marca las parejas distinguibles a true.



Ejemplo



→ bool esDistinguible d? → FALSE, al principio son todas indistinguibles



Empezamos por abajo, de izq a derecha →

- ① Marcamos todas las  $\alpha$ s de  $C$  porque es el único estado final.  
la cadena que lo hace distinguible es  $\epsilon$ .

$\varepsilon$  desde  $u \rightarrow u$  (NO Final)

E desde  $c \longrightarrow c$  (final)

- ② Varios ( $h, a / h, b / h, c / h, g / h, g / g, a \dots$ )

 $(h, a)$ 

	0	1
h	(g c) (FINAL)	
a	(c b) (NO FINAL)	

DIVISIBLE  $h \xrightarrow{1} e \xrightarrow{0} 9$

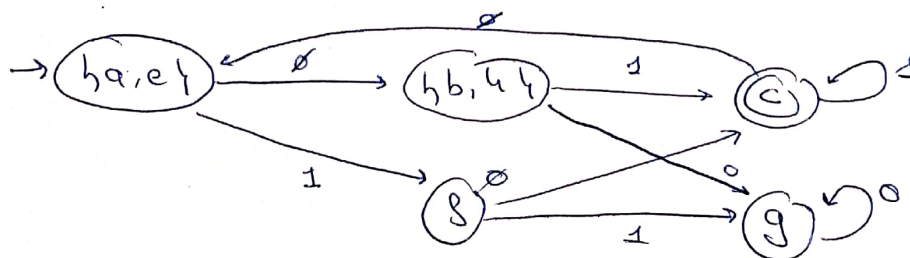
→ Si alcanzo una casilla distinguible, la marco como distinguible.

	0	1
u	g	b
g	b	g

→ Si no tengo info,  
marco de donde vengo

- \* Como se que ha es distinguible, la marca, como es distinguible y luego desde  $g, a$ , la marca como distinguible

- ③ Para los autómata acíclicos, agrupamos los estados indistinguibles

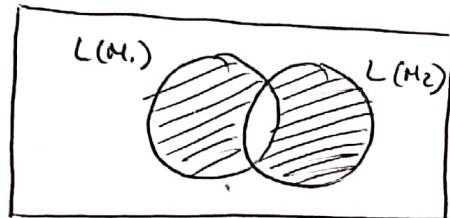


## AUTOMATAS EQUIVALENTES

Son automatas que aceptan el mismo lenguaje.

$$\text{Si } (L(M_1) - L(M_2)) \cup (L(M_2) - L(M_1)) = \emptyset$$

es porque  $L(M_1) = L(M_2) \Rightarrow$  SON EQUIVALENTES



Hacemos automata producto y comprobamos si el lenguaje aceptado es vacío

→ Si es vacío, los automatas son EQUIVALENTES

- Un lenguaje es vacío cuando eliminamos los estados inaccesibles (mediante un recorrido por el grafo a partir del estado inicial) y comprobamos que no quedan ESTADOS FINALES
- Dos estados son COMPATIBLES si sus estados son ambos finales o no finales