



**Instituto Tecnológico de Costa Rica**

**Escuela de Ingeniería en Computación  
Compiladores e Interpretes**

**Grupo 40**

***Apuntes del 29 de marzo de 2017***

**Elaborado por:**

**Izcar Muñoz Torrez 2015069773**

**Profesor:**

**Dr. Francisco Torres Rojas**

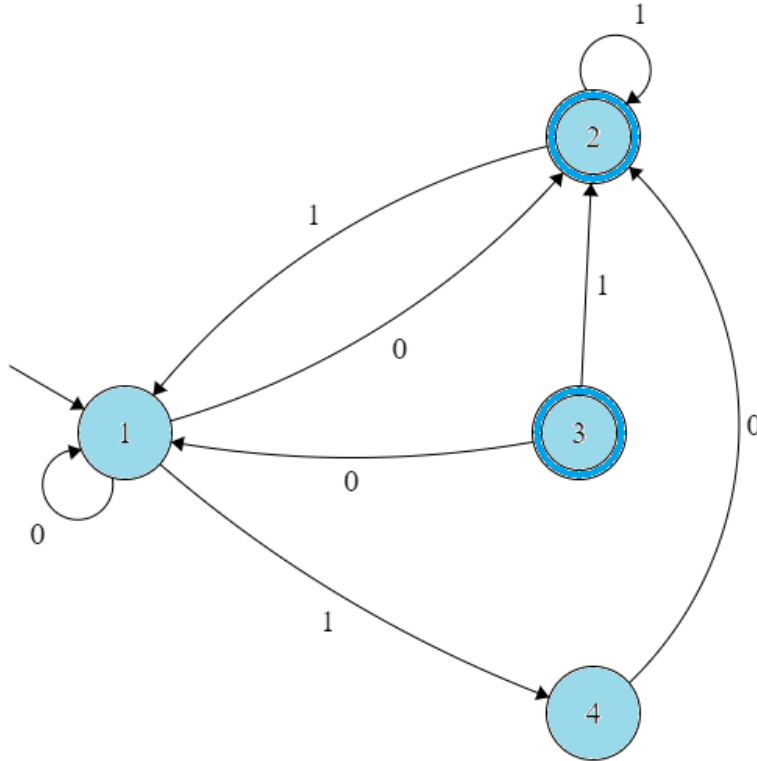
**I Semestre, 2017**

## Contents

Quiz #6 .....	3
Transiciones $\epsilon$ .....	4
Ejemplo 1 .....	4
Ejemplo 2 .....	5
Ejemplo 3 .....	5
Ejemplo 4 .....	6
Propiedades de Transiciones $\epsilon$ .....	6
Definición Formal .....	7
Teorema .....	7
Corolario.....	7
Cierre- $\epsilon$ .....	8
Ejemplo .....	8
Convirtiendo NFA- $\epsilon$ en DFA.....	9
Calculo de una Nueva Función de Transición .....	9
Estados de aceptación .....	10
Ejemplo 2 .....	11
Ejemplos 3.....	12
Lenguajes Regulares .....	13
Definición .....	13
Ejemplos de lenguajes regulares .....	13
Cierre en Conjuntos .....	14
Cierre en Unión de Lenguajes Regulares .....	14
Teorema .....	14
Ejemplo .....	14
Cierre en Complemento de Lenguajes Regulares.....	15
Teorema .....	15
Ejemplo .....	16
Cierre en Intersección de Lenguajes Regulares .....	16
Teorema .....	16
Recordatorios.....	17

## Quiz #6

1. Considere el siguiente NFA. Presente el DFA equivalente.



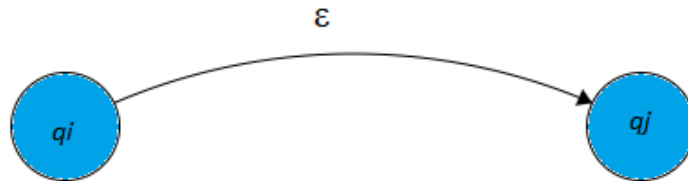
2. Diseñe un de DFA que reconozca el lenguaje sobre  $\Sigma = \{0,1\}$  de hileras donde en ninguna subhiler de tamaño 4 haya más 0's que 1's.



## Transiciones $\epsilon$

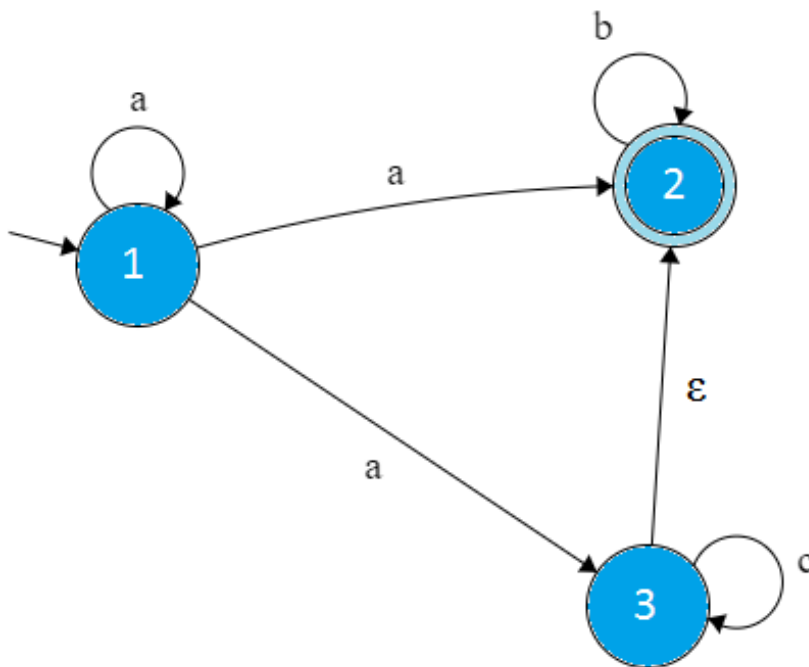
- Por el momento cada transición que hacemos nos consume un símbolo de la y se avanza al estado siguiente.
- Pero existen estos hermosos autómatas que nos hacen trasladarnos de un estado a otro sin consumir símbolos.

- Conectamos dos estados con una transición con etiqueta ' $\epsilon$ '.
- Cada transición  $\epsilon$  no consume símbolo.
- A estos los nombraremos **NFA-  $\epsilon$** .



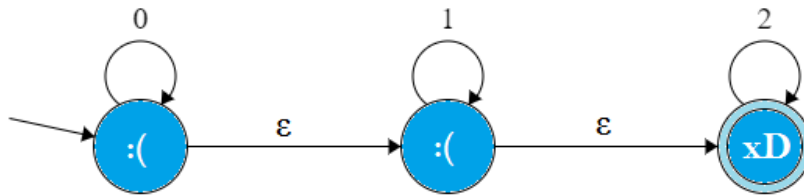
### Ejemplo 1

- Sea  $\mathcal{L}$  el lenguaje sobre  $\Sigma = \{a, b, c\}$  de hileras que empiecen con una serie de una o más "a", seguida de cero o más c y terminada en cero o más "b".
- Diseño NFA-  $\epsilon$  que reconoce a  $\mathcal{L}$ .



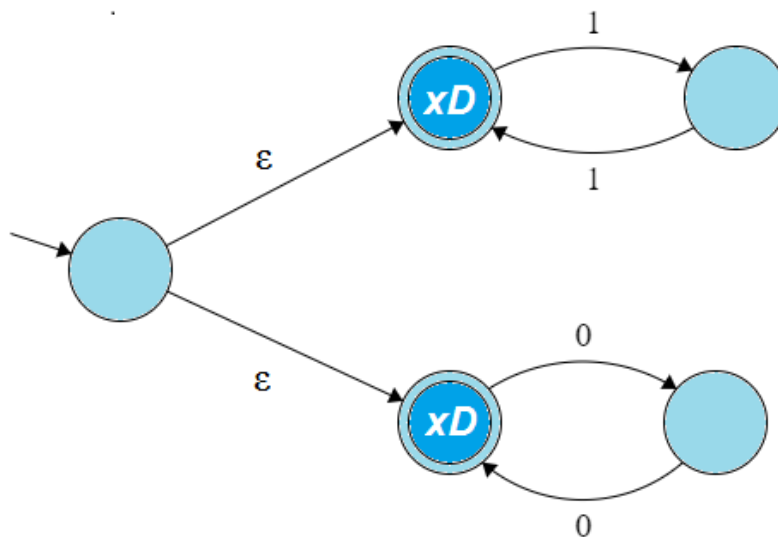
### Ejemplo 2

- Sea  $\mathcal{L}$  el lenguaje sobre  $\Sigma = \{0, 1, 2\}$  de hileras que empiecen con cualquier número de 0's, seguidos por cualquier número de 1's y terminada en cualquier número de 2's.
- Diseño NFA-  $\epsilon$  que reconoce a  $\mathcal{L}$ .



### Ejemplo 3

- Sea  $\mathcal{L}$  el lenguaje sobre  $\Sigma = \{0, 1\}$  de hileras de longitud par con un único tipo de símbolo.
- Diseño NFA-  $\epsilon$  que reconoce a  $\mathcal{L}$ .

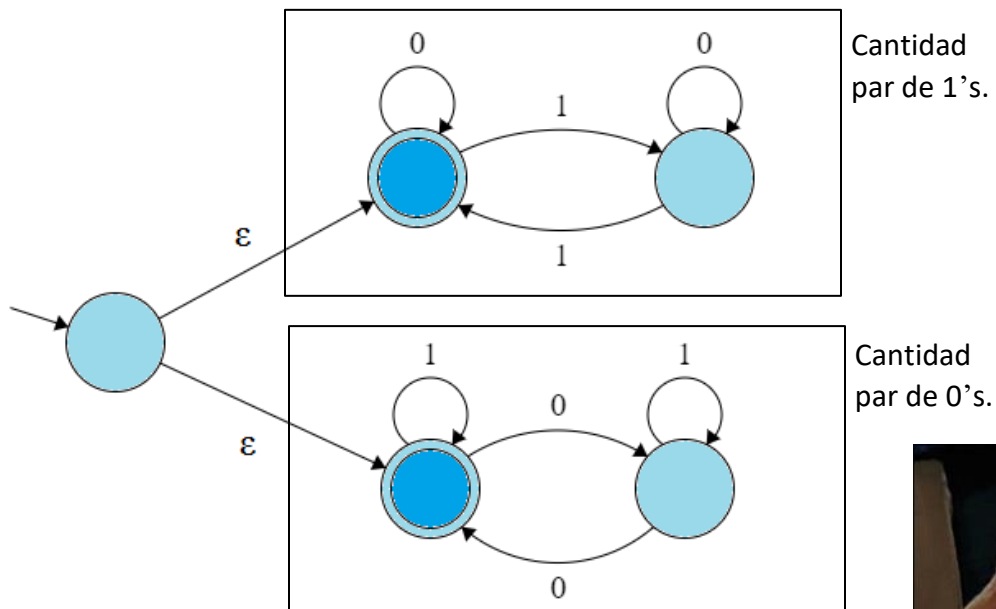


En este autómata, se puede decir que es el equivalente a un “OR”. Ya que va a ir al estado apropiado o al de conveniencia, y este será donde haya o este la posibilidad de aceptación para la hilera en evaluación.

### Ejemplo 4

- Sea  $\mathcal{L}$  el lenguaje sobre  $\Sigma = \{0, 1\}$  de hileras que tengan un número par de 0's o un número par de 1's.
- Diseño NFA-  $\epsilon$  que reconoce a  $\mathcal{L}$ .

Este ejemplo es similar al anterior, con el cambio que ahora se pide que reconozca un numero par de 1's o 0's. Para este tipo de lenguaje  $\epsilon$  es ideal. Ya que se pueden crear dos máquinas distintas, una donde se reconozca solo la cantidad par de 1's y otra donde se reconozca la cantidad par de 0's y luego las conectamos con majestuosas **transiciones  $\epsilon$** .



### Propiedades de Transiciones $\epsilon$

- Mientras haya transiciones  $\epsilon$  disponibles podemos trasladarnos entre estados sin consumir símbolos.
- Se puede seguir trasladándose, aunque se haya terminado la hilera, obviamente siempre y cuando este una transición  $\epsilon$  disponible.
- No es obligatorio usarlas :v.
- Naturaleza **no determinística**.
- Para este tipo de autómatas el nombre oficial es **NFA-  $\epsilon$** .

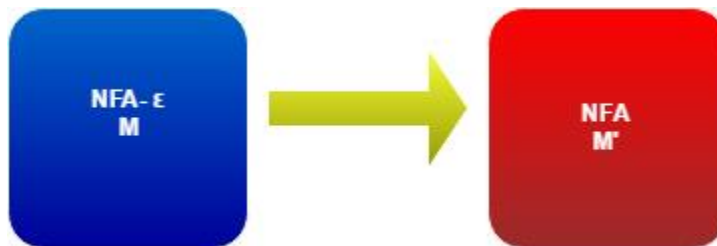
## Definición Formal

- Un Autómata No Determinístico de Estados Finitos con transiciones  $\epsilon$  (**NFA-  $\epsilon$** ) es un quinteto  $M = (Q, \Sigma, \delta, q_0, F)$ , donde:
  - $Q$  es un conjunto finito de **estados**.
  - $\Sigma$  es un **alfabeto**.
  - $\delta: Q \times (\Sigma \cup \{\epsilon\}) \rightarrow P(Q)$
  - $q_0 \in Q$  es el **estado inicial**.
  - $F \subseteq Q$  conjunto de **estados de aceptación**.



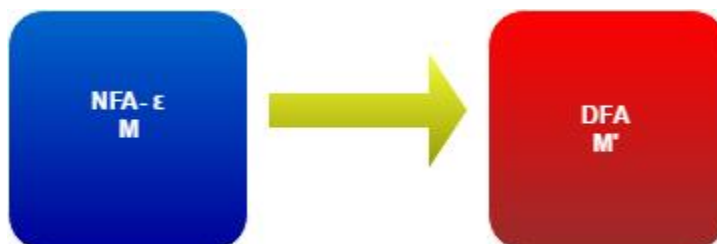
## Teorema

- Sea  $M = (Q, \Sigma, \delta, q_0, F)$  un **NFA-  $\epsilon$** . Siempre existe un NFA  $M' = (P(Q), \Sigma, \delta', Q_0, F')$  que reconoce exactamente el mismo lenguaje que  $M$ .
- Todo **NFA-  $\epsilon$**  tiene un NFA sin movimientos  $\epsilon$  **que** reconoce exactamente el mismo lenguaje.



## Corolario

- Sea  $M = (Q, \Sigma, \delta, q_0, F)$  un **NFA-  $\epsilon$** . Siempre existe un DFA  $M' = (P(Q), \Sigma, \delta', Q_0, F')$  que reconoce exactamente el mismo lenguaje que  $M$ .
- Todo **NFA-  $\epsilon$**  tiene un DFA equivalente que reconoce exactamente el mismo lenguaje.



## Cierre- $\epsilon$

- Es una función que recibe un estado de  $Q$  y nos retorna un subconjunto de  $Q$ .
- El Cierre- $\epsilon$  de un estado  $p$  es el conjunto de todos los estados alcanzables desde  $p$  sin consumir **ningún símbolo** de la entrada.
- Obviamente solo usando transiciones  $\epsilon$ .
- Implícitamente hay una transición  $\epsilon$  de  $p$  con sí mismo.
- Para los próximos cálculos, cada vez que necesitemos usar  $p$ , usaremos **Cierre- $\epsilon(p)$** , hasta para calcular el **Cierre- $\epsilon$** .

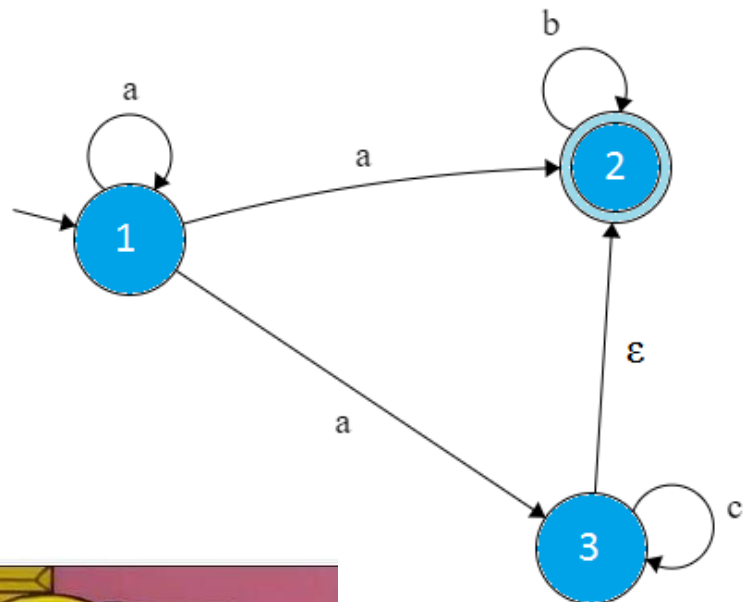


## Ejemplo

- “Conjunto de todos los estados alcanzables desde  $p$  sin consumir **ningún** símbolo de la entrada”.

Este es el primer paso que vamos a hacer para ir convirtiéndolo en DFA.

Estado	Cierre- $\epsilon$
1	{1}
2	{3}
3	{2, 3}





## Convirtiendo NFA- $\epsilon$ en DFA

- El alfabeto  $\Sigma$  es el mismo en las dos máquinas.
- Los estados  $M'$  son subconjuntos  $Q$  de  $M$ .
- Hay que calcular la nueva función de transición  $\delta'$ .
- $F'$  es un conjunto de subconjuntos del conjunto  $Q$  de  $M$ .
- Hay que calcular el **Cierre-  $\epsilon$**  de todos los estados de  $Q$ .

## Calculo de una Nueva Función de Transición

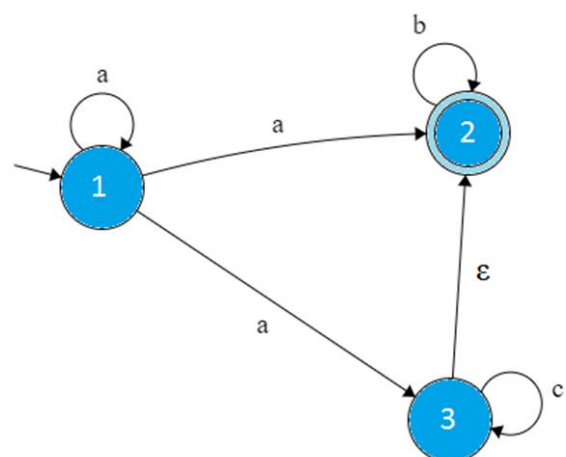
1. El estado inicial de  $M'$  es **Cierre-  $\epsilon$**  ( $q_0$ ). Esta es la primera fila de la tabla asociada a  $\delta'$ .
2. En la tabla asociada a la función de transición  $\delta'$  en la columna de cada símbolo de  $\Sigma$  la unión de todos los **Cierre-  $\epsilon$**  de todos los estados a los que se puede llegar desde los elementos de **Cierre-  $\epsilon$**  de  $q_0$  con dicho símbolo (el conjunto vacío es una opción válida).
3. Tome el primero de estos subconjuntos y póngalo con una nueva fila de la tabla de  $\delta'$ .
4. Llene las columnas con la unión de todos los **Cierre-  $\epsilon$**  de todos los estados a los que se puede llegar desde todos los elementos del conjunto asociado a la fila actual con el símbolo correspondiente a la columna actual (puede ser el conjunto vacío).
5. Repita mientras queden subconjuntos pendientes.



## Ejemplo de Función de Transición

Para este ejemplo utilizaremos el ejemplo de [Cierre-  \$\epsilon\$](#)

Estado	Cierre- $\epsilon$
1	{1}
2	{3}
3	{2, 3}



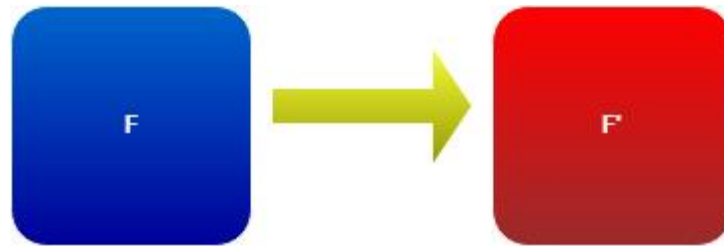
- Tabla de función de transición.

	a	B	c
{1}	{1, 2, 3}	$\emptyset$	$\emptyset$
{1, 2, 3}	{1, 2, 3}	{2}	{2, 3}
{2}	$\emptyset$	{2}	$\emptyset$
{2, 3}	$\emptyset$	{2}	{2, 3}

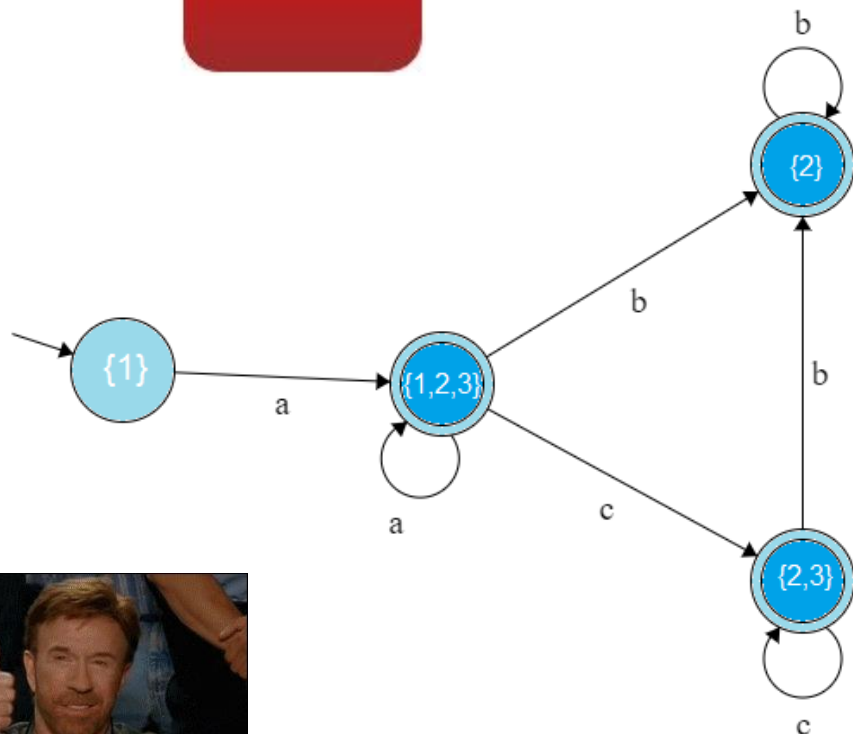
### Estados de aceptación

- Cualquier estado en  $M'$  que contenga al menos un elemento que sea miembro de  $F$  es miembro de  $F'$ .

Por lo tanto en el conjunto que se encuentre un '2' será un estado de aceptación en el autómata anterior.

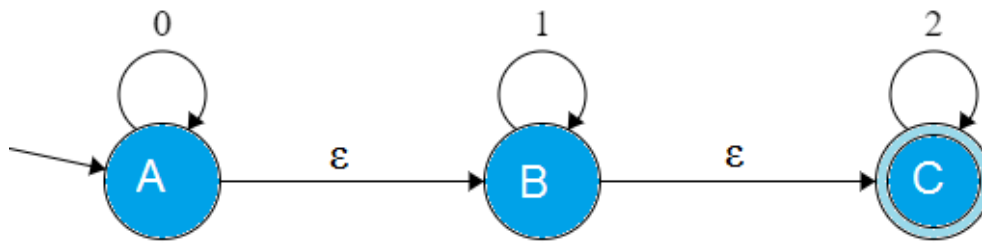


	a	b	c
{1}	{1, 2, 3}	$\emptyset$	$\emptyset$
{1, 2, 3}	{1, 2, 3}	{2}	{2, 3}
{2}	$\emptyset$	{2}	$\emptyset$
{2, 3}	$\emptyset$	{2}	{2, 3}



## Ejemplo 2

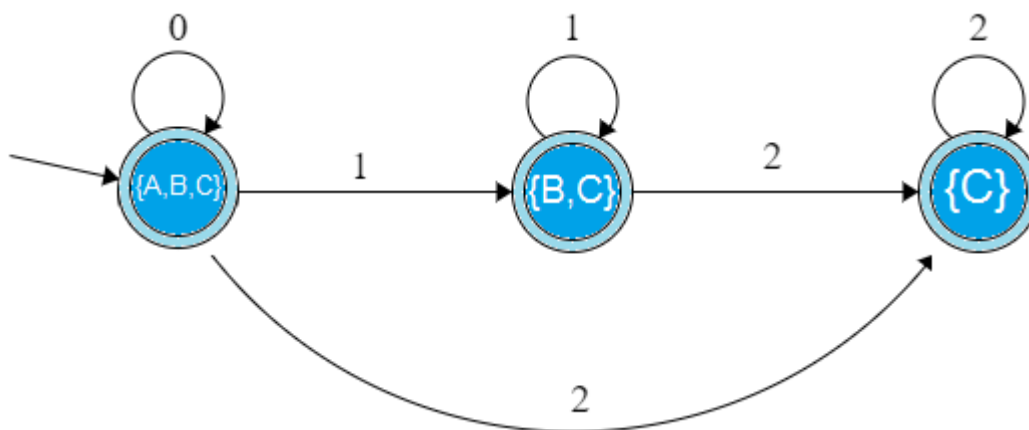
- Encuentre el DFA correspondiente al siguiente NFA-  $\epsilon$ .



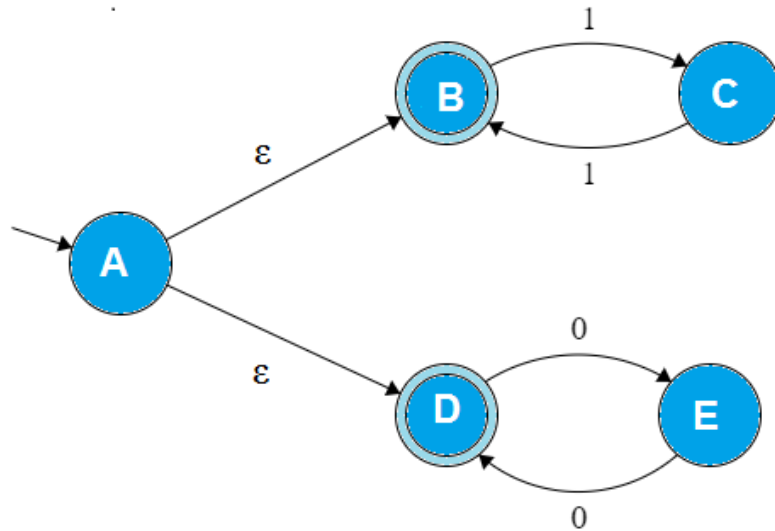
Estado	Cierre- $\epsilon$
A	{A, B, C}
B	{B, C}
C	{C}

- Luego hacemos la tabla de función de transición.

	0	1	2
{A, B, C}	{A, B, C}	{B, C}	{C}
{B, C}	$\emptyset$	{B, C}	{C}
{C}	$\emptyset$	$\emptyset$	{C}



### Ejemplos 3



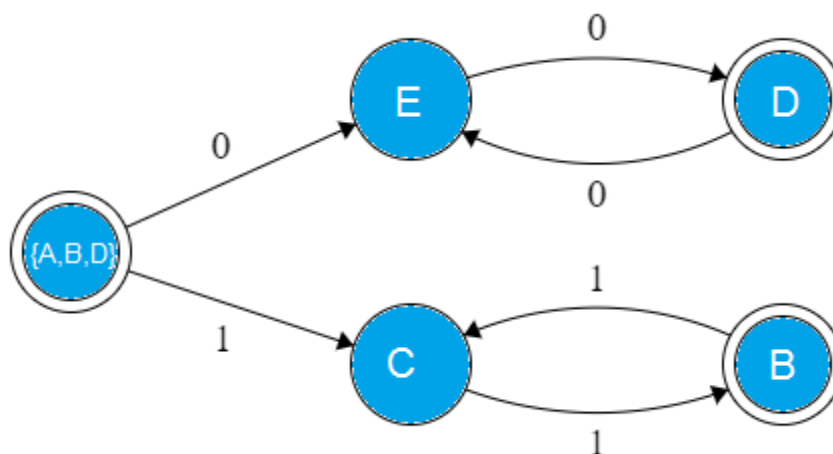
Estado	Cierre- $\epsilon$
A	{A, B, D}
B	{B}
C	{C}
D	{D}
E	{E}

- Tabla de función de transición.

	0	1
{A, B, D}	{E}	{C}
{E}	{D}	$\emptyset$
{C}	$\emptyset$	{B}
{D}	{E}	$\emptyset$
{B}	$\emptyset$	{C}



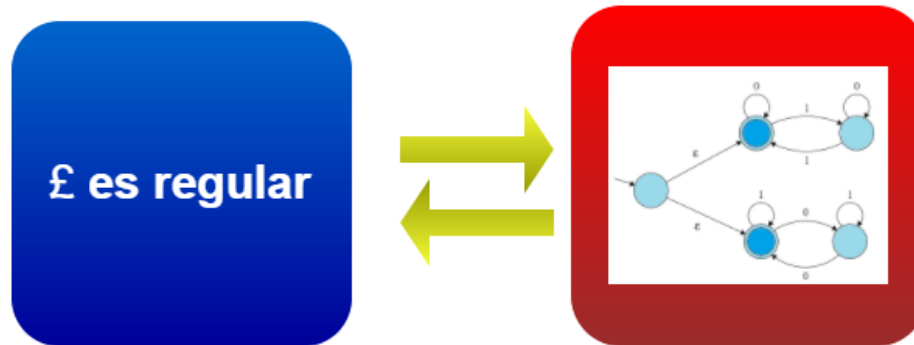
- autómata resultante



## Lenguajes Regulares

### Definición

- Un lenguaje  $\mathcal{L}$  es **regular** si y sólo si existe un **DFA** que reconozca que lo reconozca exactamente (acepta todas las hileras  $\mathcal{L}$  y rechaza cualquier hilera que no sea miembro de  $\mathcal{L}$ ).
- Entonces, si nos dicen que un lenguaje es regular significa que existe un **DFA** que lo reconozca, y viceversa.

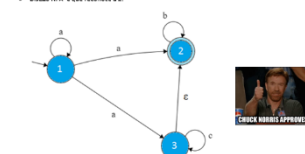


- Sabemos que Todo NFA o NFA-  $\epsilon$  tiene un **DFA** equivalente por el [teorema](#) que vimos anteriormente. Por lo tanto **un lenguaje es regular si tiene un NFA o NFA-  $\epsilon$  que lo reconozca.**
- ¿Cómo se demuestra que un lenguaje es regular?  
Diseñando un DFA que lo reconozca.
- ¿Como se demuestra que un lenguaje no es regular?  
Demostrando que no existe ningún DFA que lo reconozca.
  - **PUMPING LEMMA**

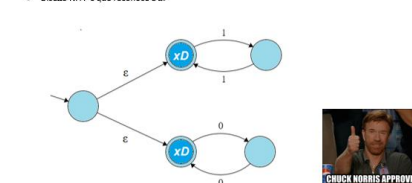
### Ejemplos de lenguajes regulares

- Todos los lenguajes que hemos detectado hasta el momento son regulares(ya que se hicieron DFA's que los reconocieran).

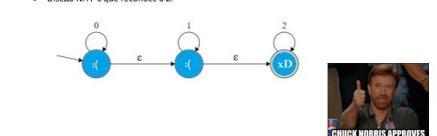
Ejemplo 1



Ejemplo 3



Ejemplo 2



## Cierre en Conjuntos

- Un conjunto  $\mathcal{X}$  puede ser “**cerrado con respecto a una operación  $\varphi$** ”.
- Significa que siempre que apliquemos  $\varphi$  operación sobre elementos de  $\mathcal{X}$  obtenemos algo que también es elemento de  $\mathcal{X}$ .
- Números naturales son cerrados respecto a suma o multiplicación.
- Números naturales **no son** cerrados respecto división o resta.

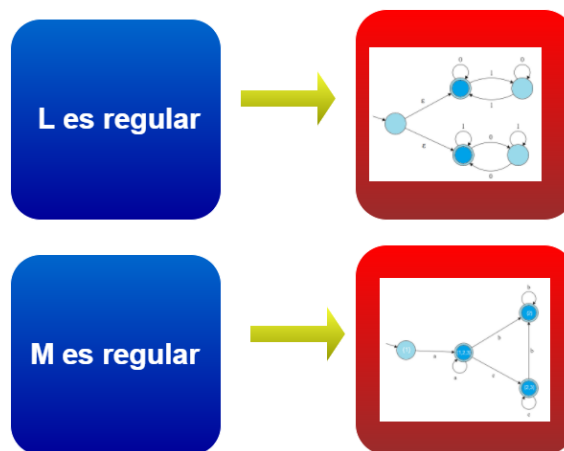
## Cierre en Unión de Lenguajes Regulares

### Teorema

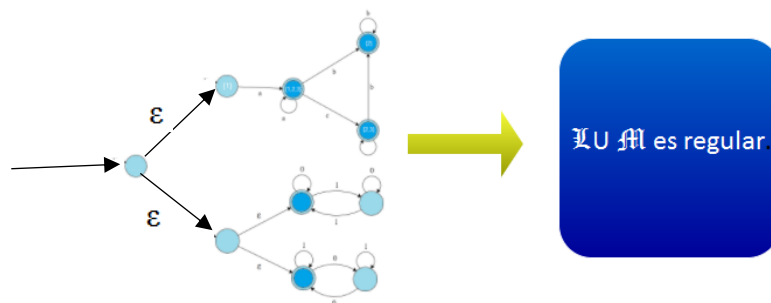
- Si  $\mathcal{L}$  y  $\mathcal{M}$  son lenguajes regulares entonces  $\mathcal{L} \cup \mathcal{M}$  también es regular.

Los lenguajes regulares son cerrados respecto a la unión de lenguajes.

- Demostración (Idea General)  
Si  $\mathcal{L} \cup \mathcal{M}$  fuera regular, existiría un NFA que lo reconociera.  
Premisas:



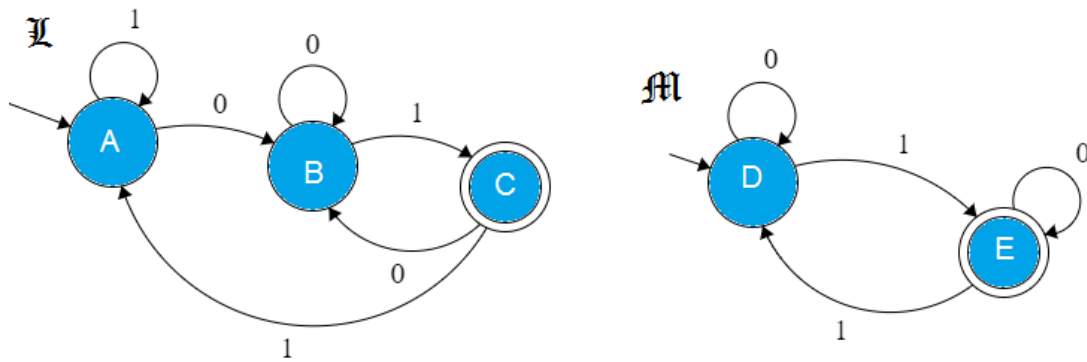
- Demostración :v.



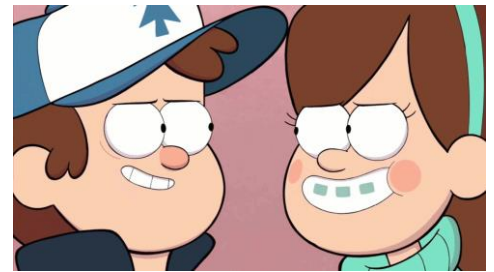
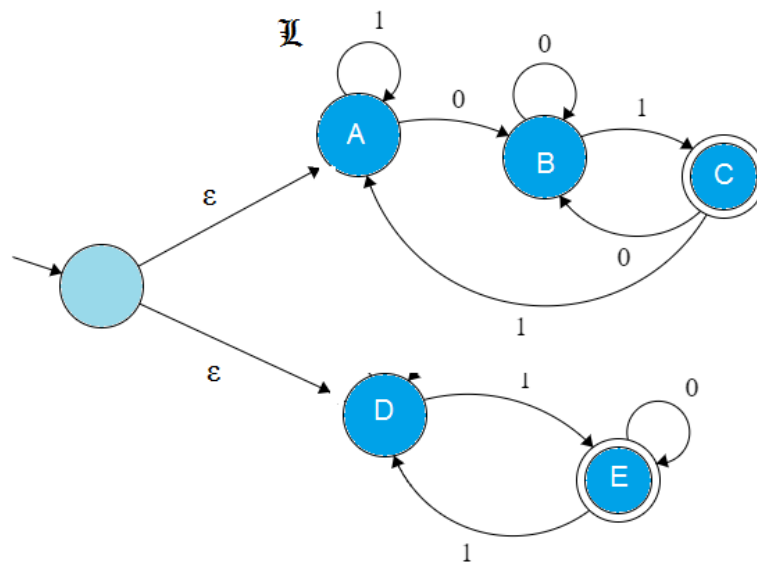
### Ejemplo

- $\mathcal{L}$  = lenguaje sobre  $\Sigma = \{0, 1\}$  de hileras que terminan en 01.
- $\mathcal{M}$  = lenguaje sobre  $\Sigma = \{0, 1\}$  de hileras con un número impar de 1's.

- $\mathcal{L} \cup \mathcal{M}$  es regular.



- Ahora con las transiciones  $-\epsilon$ .

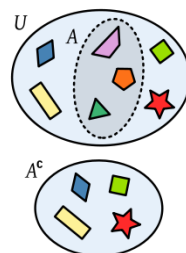


## Cierre en Complemento de Lenguajes Regulares

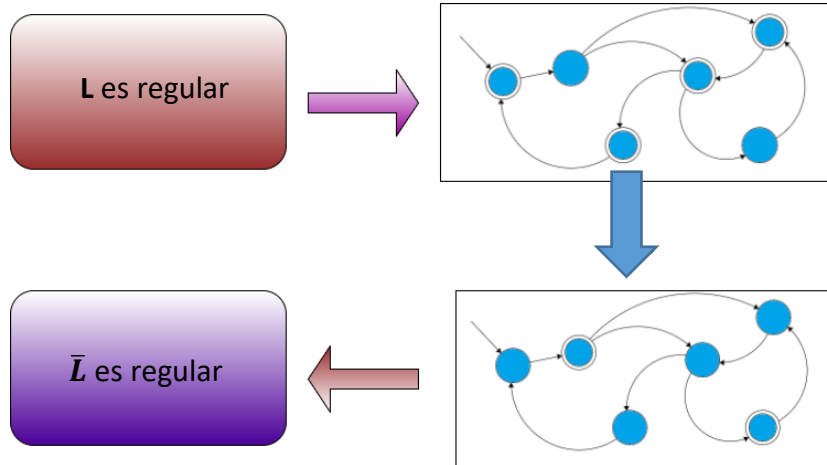
### Teorema

- Si  $L$  es un lenguaje regular, entonces  $\bar{L}$  también es regular.

Los lenguajes regulares son **cerrados respecto** al complemento de lenguajes.

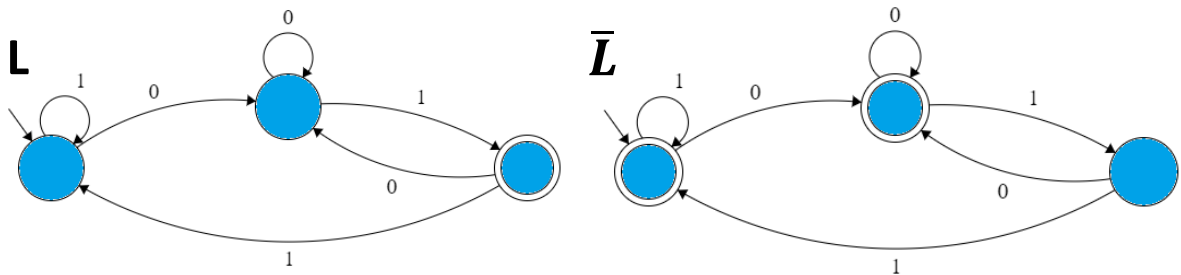


- Demostración (Idea General)



Ejemplo

- $L$  = lenguaje sobre  $\Sigma = \{0, 1\}$  de hileras que terminan en 01.
- $\bar{L}$  es regular.



## Cierre en Intersección de Lenguajes Regulares

### Teorema

- Si  $L$  y  $M$  son lenguajes regulares entonces  $L \cap M$  también es regular.

Los lenguajes regulares son cerrados respecto a la intersección de lenguajes.

- Demostración

- $L \cap M = \overline{\overline{L \cap M}}$  (Propiedad del complemento de lenguajes)
- $\overline{\overline{L \cap M}} = \overline{\bar{L} \cup \bar{M}}$  (De Morgan)
- $L$  y  $M$  son lenguajes regulares.
- Los lenguajes regulares son cerrados respecto a **complemento** y respecto a **unión**.





## Recordatorios

- Tarea de expresiones regulares, los mismos ejercicios que se dejaron para lo autómatas hay que hacerlos para expresiones regulares :v.
- Examen el Viernes 7 de abril de 2017, se puede llevar un **foro**, de una pagina, no se permita impresiones.
- Revisión de proyecto#1, Miércoles 5 de abril de 2017.

