Instituto Politécnico Nacional
Escuela Superior de Cómputo

Teoría computacional

Clase 09: AFN, AFD y Construcción de Thompson

Solicitado: Ejercicios 07: Construcción de AFN´s con Thompson

M. en C. Edgardo Adrián Franco Martínez
http://computacion.cs.cinvestav.mx/~efranco
@efranco_escom
edfrancom@ipn.mx

Contenido

- Autómata finito
- Clasificación de los autómatas finitos
- Autómata finito no determinista (AFN)
- Autómata finito determinista (AFD)
- Teorema sobre la transformación de AFN en AFD
- Transformación de una expresión regular en un autómata finito
- Construcción de Thompson de un AFN a partir de una expresión regular
- Nomenclatura de Thompson
- Ejemplos
- Ejercicios 07: Construcción de AFN´s con Thompson

Autómata finito

- Un **autómata finito es un modelo matemático** de una máquina que acepta cadenas de un lenguaje definido sobre un alfabeto.

- Consiste en un ***conjunto finito de estados y un conjunto de transiciones entre esos estados***, que dependen de los símbolos de la cadena de entrada.

- El **autómata finito acepta** una **cadena *x*** si la secuencia de transiciones correspondientes a los símbolos de *x* conduce desde **el estado inicial a un estado final**.

Clasificación de los autómatas finitos

- La función **f: E*x Q→Q**, es en general no determinista. Así **en función de *f***, se hablará de ***autómatas finitos deterministas AFD*** *y* ***autómatas finitos no deterministas AFN***.

  - **Un autómata finito no determinista AFN** se caracteriza por la posibilidad de que dada una entrada ***e*** *en un estado* $\boldsymbol{q_i}$, *se pueda pasar a un estado* $\boldsymbol{q_j, q_k, \ldots, q_n}$ ***sin saber a ciencia cierta***, a cual de esos estados pasará. **Existiendo la misma probabilidad** de que pase a cualquiera de dichos estados.

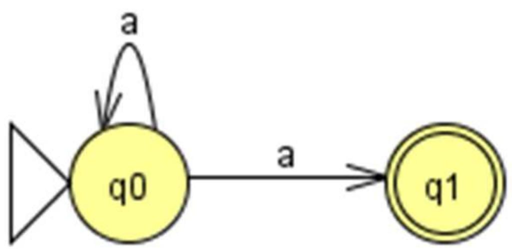
  - **Un autómata finito determinista AFD** es un caso particular de los autómatas finitos, en el que **la función de transición no presenta ninguna ambigüedad en las transiciones** de estados para una entrada dada.

Autómatas finitos no deterministas (AFN)

- La definición de autómata finito no determinista AFN o AFN coincide con la de autómata finito:

$$AFN=(E, Q, f, q1, F)$$

- Con la salvedad de que **f: E^*x Q→Q** es no determinista, i.e. es aquel que presenta cero, una o más transiciones por el mismo carácter del alfabeto.



AFN Ejemplo

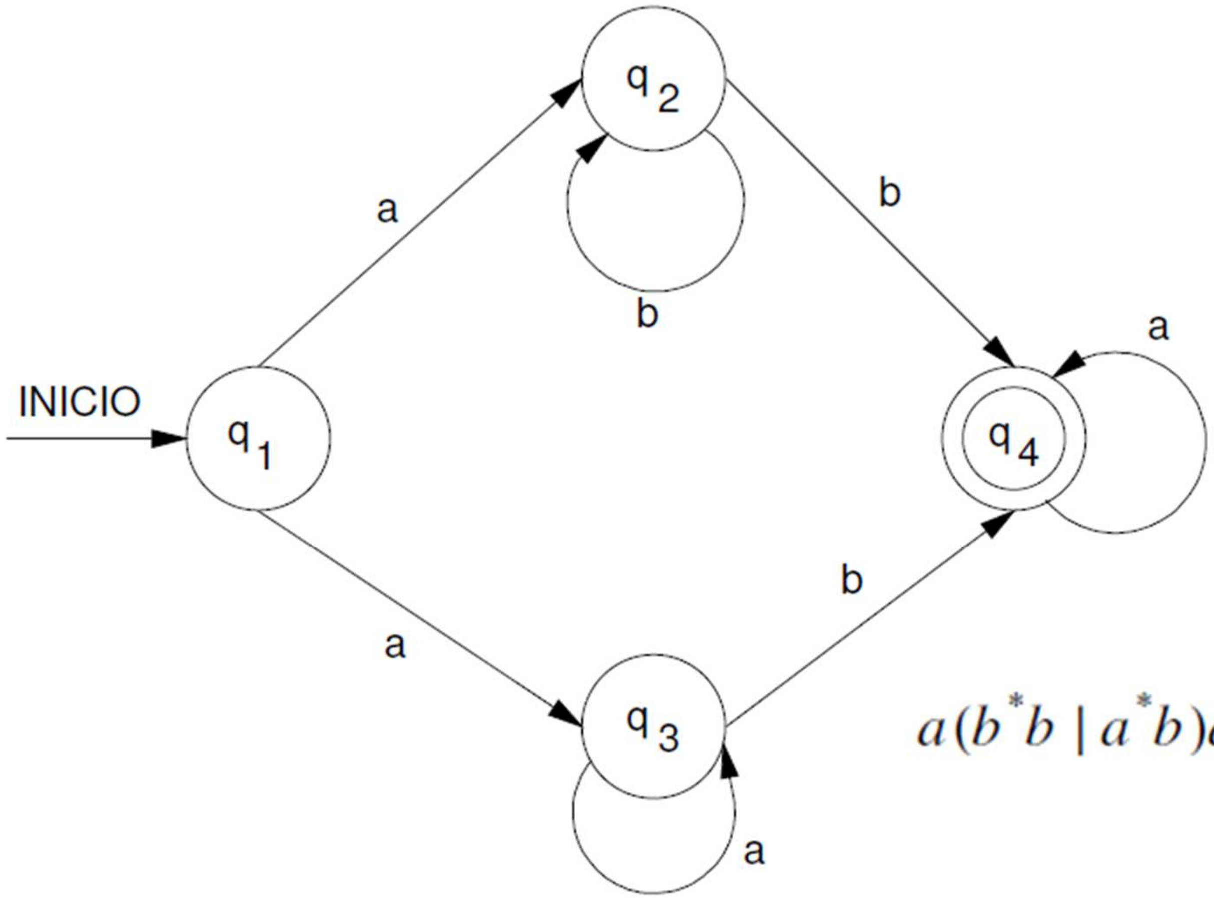
- ***Resolver:***

Sea el autómata finito no determinista $AFND = (E, Q, f, q_1, F)$ donde E=$\{a,b\}$, $Q = \{q_1, q_2, q_3, q_4\}$, $F = \{q_4\}$ y la función f viene dada por la siguiente tabla :

f	a	b
q_1	$\{q_2,q_3\}$	λ
q_2	λ	$\{q_2,q_4\}$
q_3	q_3	q_4
q_4	q_4	λ

Determinar el lenguaje que reconoce, y dar su expresión regular.

- ***Solución:***



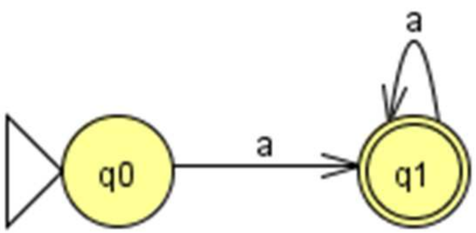
$$a(b^*b \mid a^*b)a^*$$

$$a(b^* \mid a^*)ba^*$$

Autómatas finitos deterministas (AFD)

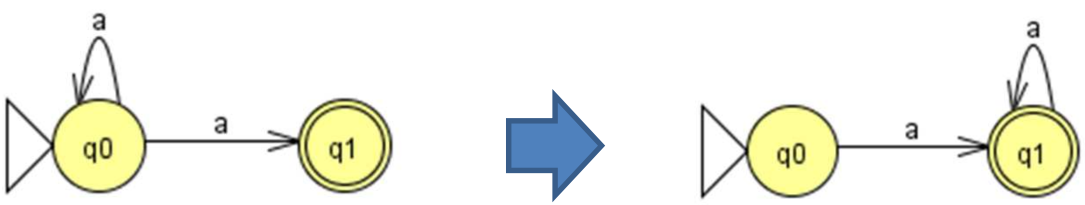
- Un autómata finito determinista **AFD** es un caso particular de los autómatas finitos, en el que **la función de transición no presenta ninguna ambigüedad** en las transiciones de estados para una entrada dada.

- Un autómata finito determinista es una quíntupla *AFD=(E, Q, f, q1, F)* donde la **función f: E^*x Q→Q** es **determinista.**



Teorema sobre la transformación de AFN en AFD

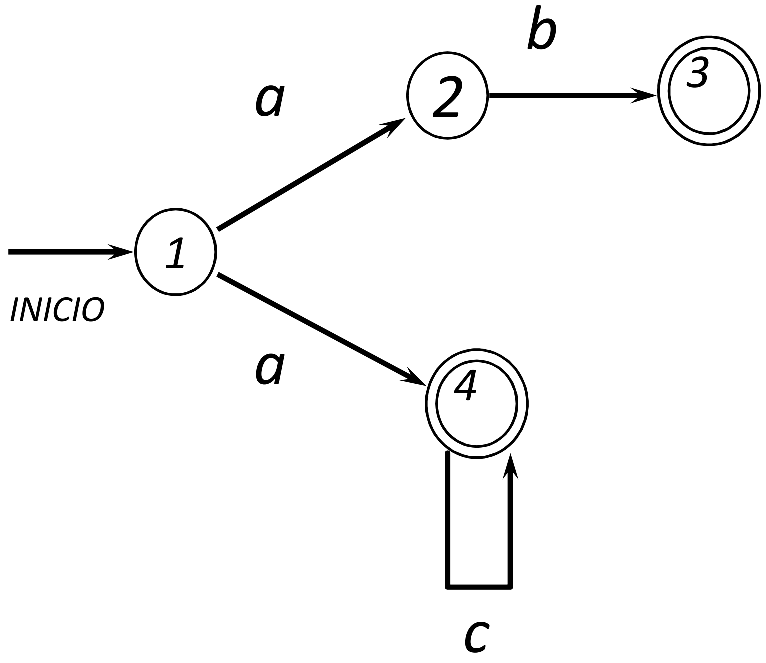
- "***Para todo autómata finito no determinista*** *AFN=(E, Q, f, q1,F)* ***se puede construir un autómata finito determinista*** *AFD=(E, Q', f', q'1, F')* ***tal que el lenguaje reconocido*** *por el autómata finito determinista AFD* ***coincida con*** *el lenguaje reconocido por el autómata finito no determinista AFN, es decir* ***L(AFD) = L(AFN)***".



Transformación de AFN en AFD

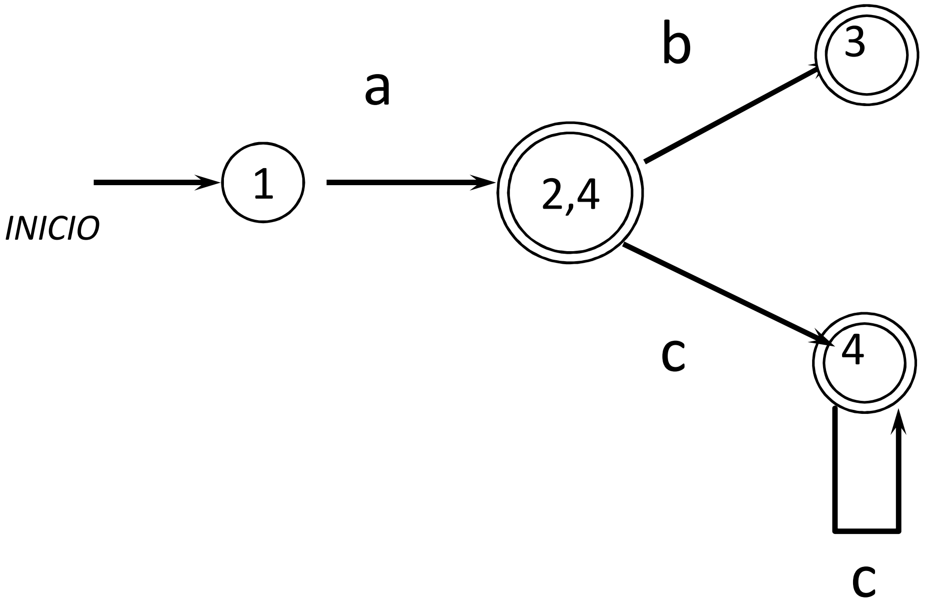
AFN

Expresión*: ab|ac**



AFD

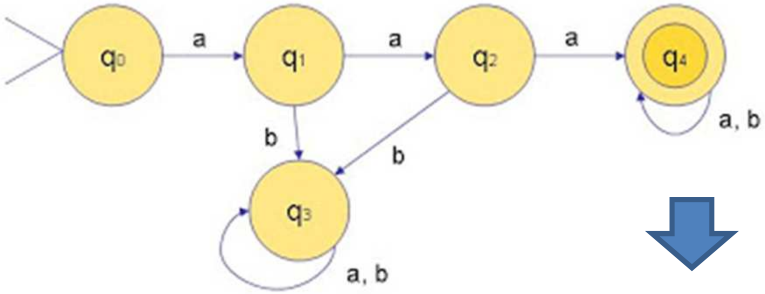
Expresión: $ab|ac^{*}$



Transformación de una expresión regular en un autómata finito

- **Dada una expresión regular existe un autómata finito** capaz de reconocer el lenguaje que ésta define.

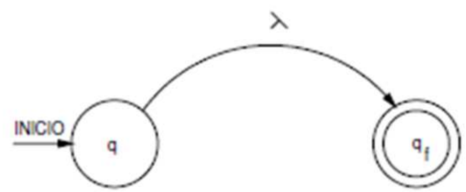
- Recíprocamente, **dado un autómata finito**, se puede expresar mediante una **expresión regular del lenguaje que reconoce.**



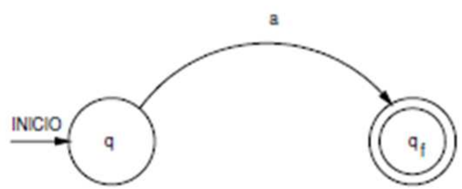
((a+b)(a(bba* + aa) + ba))(b*)

Transformación de una expresión regular en un autómata finito

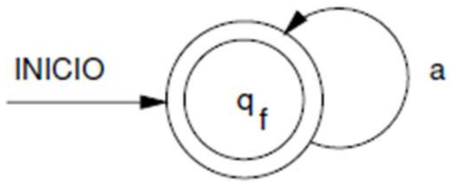
Expresión regular λ



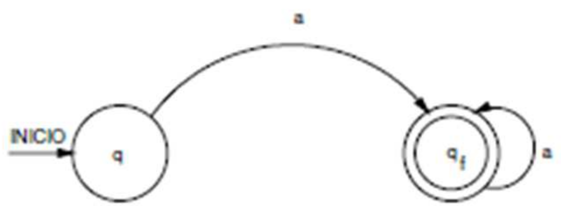
Expresión regular *a*



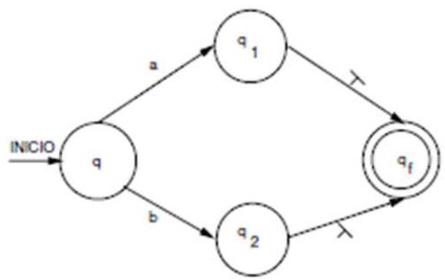
Expresión regular a^*

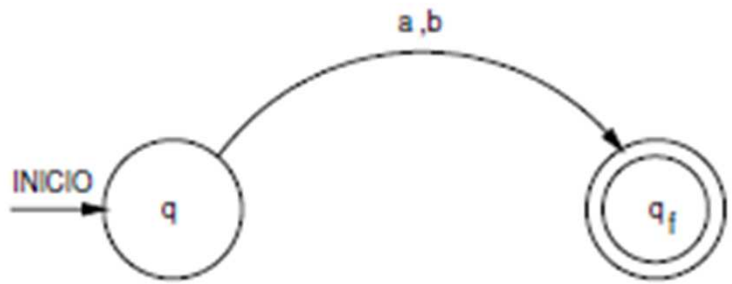


Expresión regular a^+

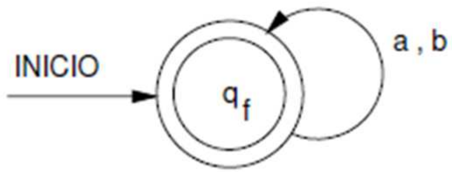


Expresión regular $a|b$

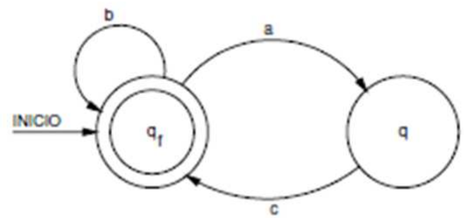




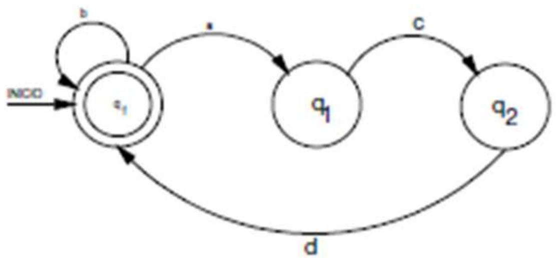
Expresión regular *(a|b)**



Expresión regular *(ac|b)**



Expresión regular *(acd|b)**

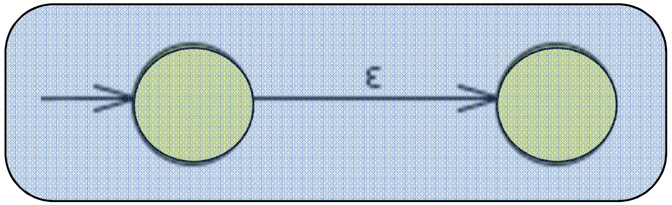


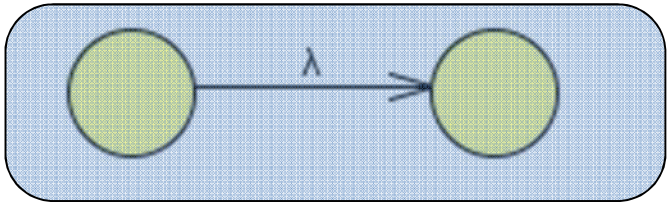
Construcción de Thompson de un AFN a partir de una expresión regular

- **La construcción de Thompson construye un AFN a partir de cualquier expresión regular**.
- La construcción de Thompson construye a partir de una expresión regular *r* un AFN que reconoce el lenguaje definido por *r*, esto se realiza con el objetivo de que en un algoritmo siguiente se pueda generar un AFD mínimo equivalente.
- Utiliza una notación estándar para generar el AFN

Nomenclatura de Thompson

- Para la representación de una cadena vacía se utiliza el símbolo ***ε*** o ***λ***

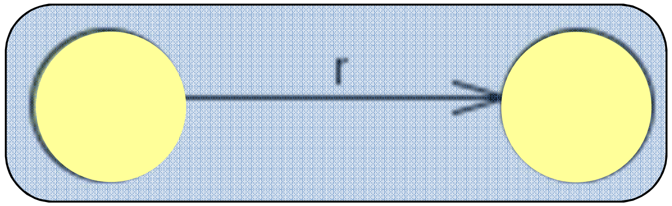




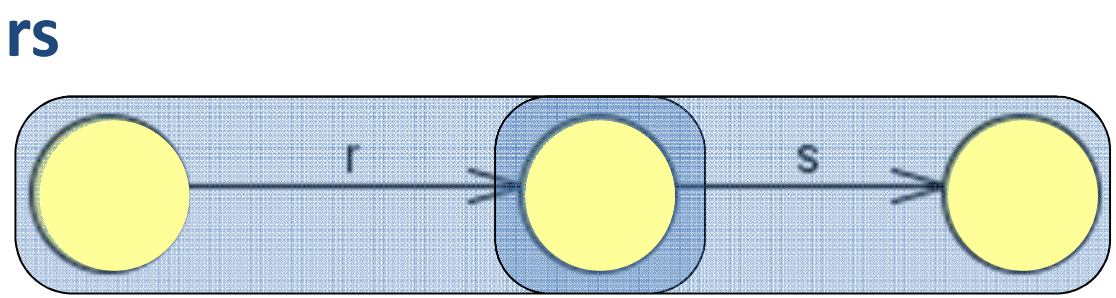
Cadena Vacía

- Para representar un símbolo, se utilizan dos estados y una transición para el movimiento con el símbolo.

r



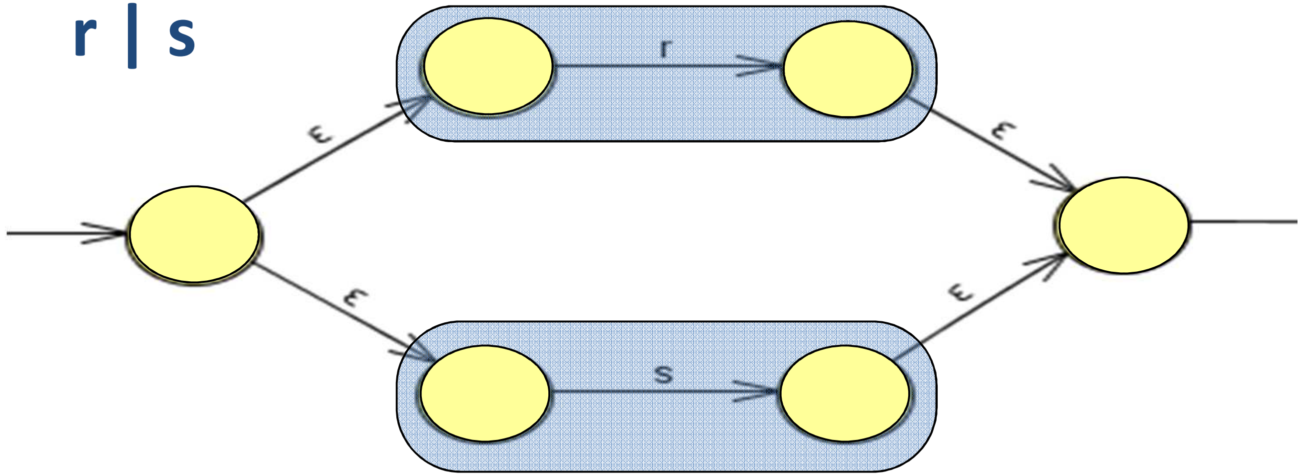
- Para la **concatenación** de dos símbolos únicamente se unen



Concatenación de símbolo

- Para la elección de **alternativas**, crear transiciones ε para la unión de las transiciones.

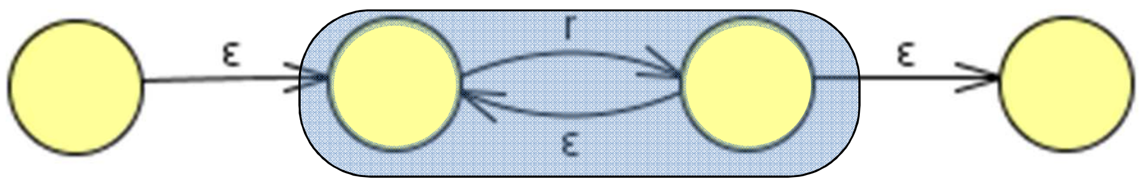
r | s



Elección de alternativas

- Para la **cerradura positiva**, se agregan transiciones ε para retornar al estado previo, permitiendo agregar 1 o mas veces el símbolo

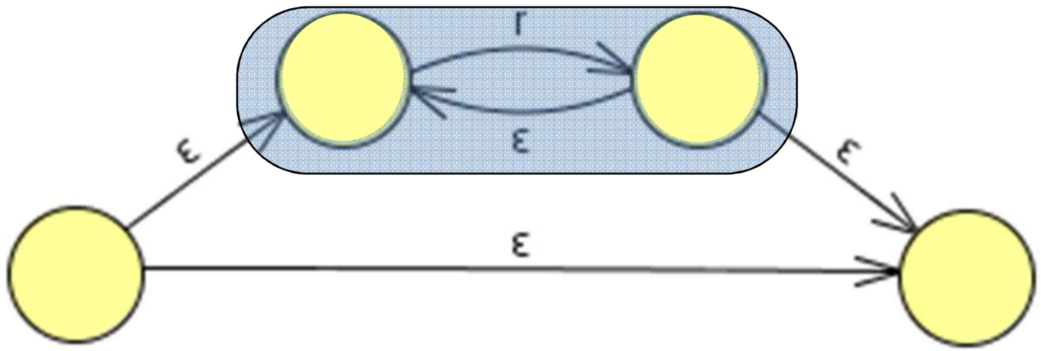
r +



Cerradura positiva

- Para la **cerradura de Kleene**, se agregan transiciones ε para retornar a estado previo. Y otra transición ε para saltar la transición con r.

r *

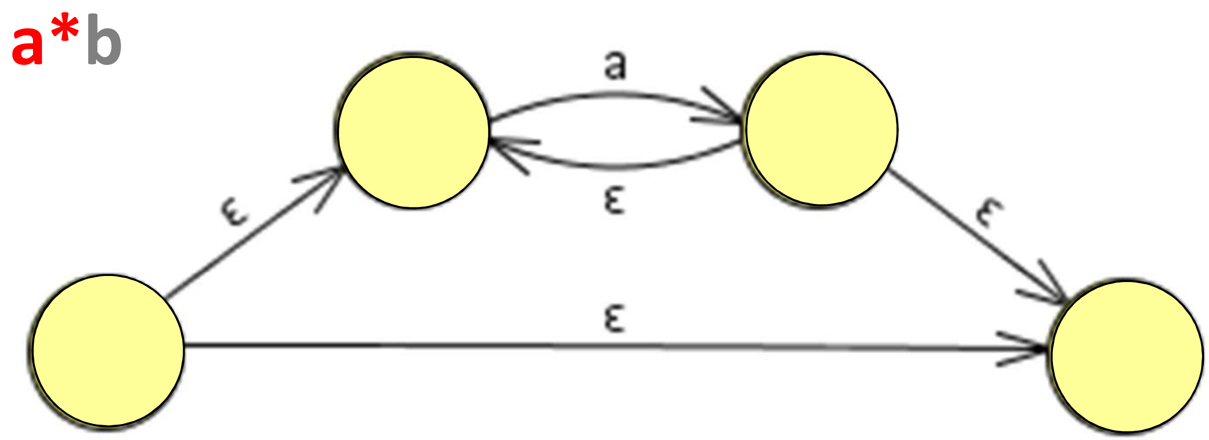


Cerradura de Kleene

Ejemplo 01 Método de Thompson

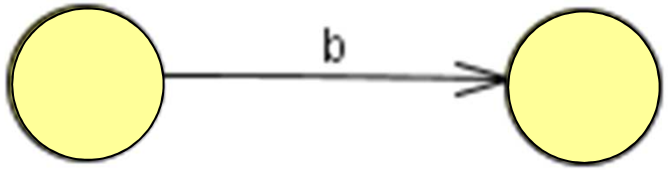
- Diagrama del AFN que representa la ER **a*b**.

1. Parte de la cerradura de Kleene.



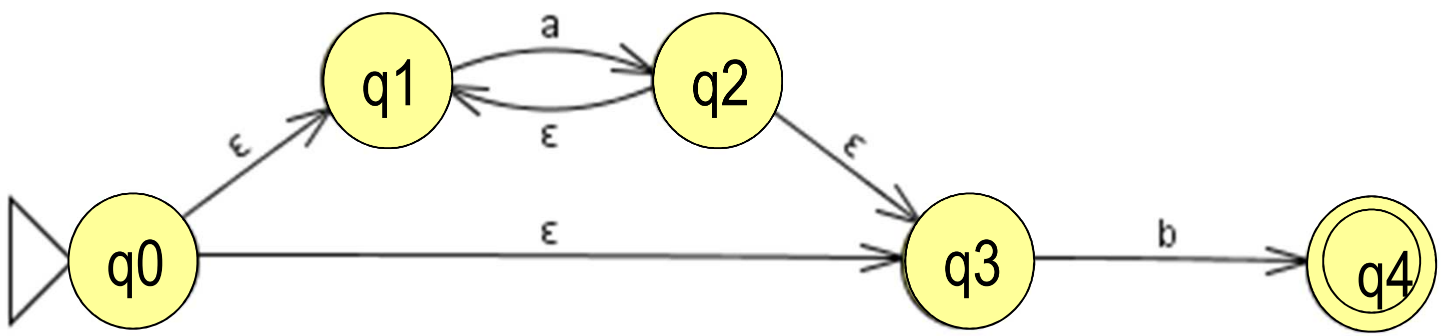
2. Para continuar se generan la concatenación del símbolo **b**

a*b



3. Para finalizar se numeran los estados y se indica el estado inicial y final

a*b



4. Formalizando

- AFN=(E,Q,f,q0,F)
- E={a,b}
- Q={q0,q1,q2,q3,q4}
- F={q4}
- f: E*x Q→Q

L(a*b)=L(AFN)

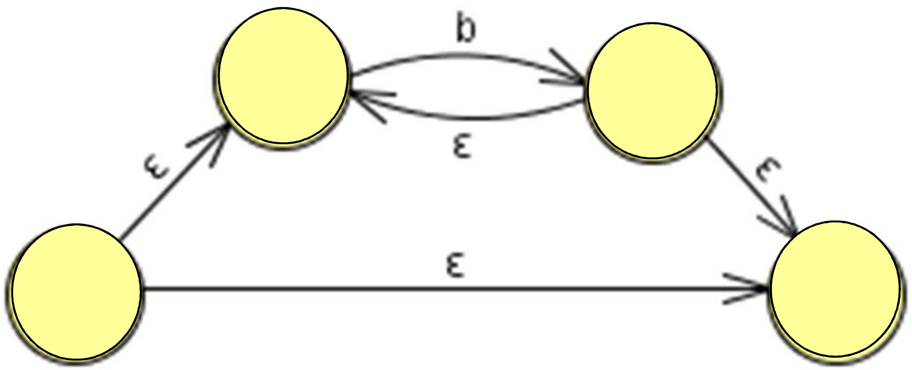
f	a	b	ε
q0	-	-	{q1, q3}
q1	q2	-	-
q2	-	-	{q1, q3}
q3	-	q4	-
q4	-	-	-

Ejemplo 02 Método de Thompson

A partir de la ER **(b|(b*a)*)a**

1. Parte de la cerradura de Kleene que se encuentra dentro de paréntesis.

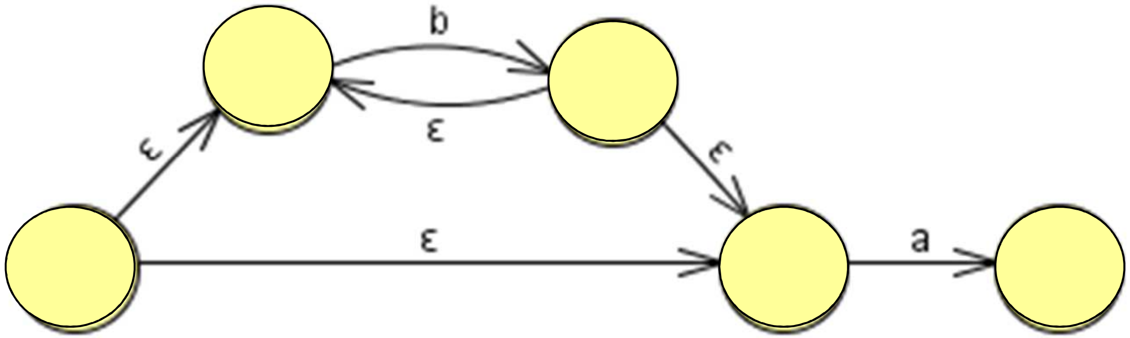
(b|(b*a)*)a



A partir de la ER **(b|(b*a)*)a**

2. Completamos dicho paréntesis concatenando el símbolo a

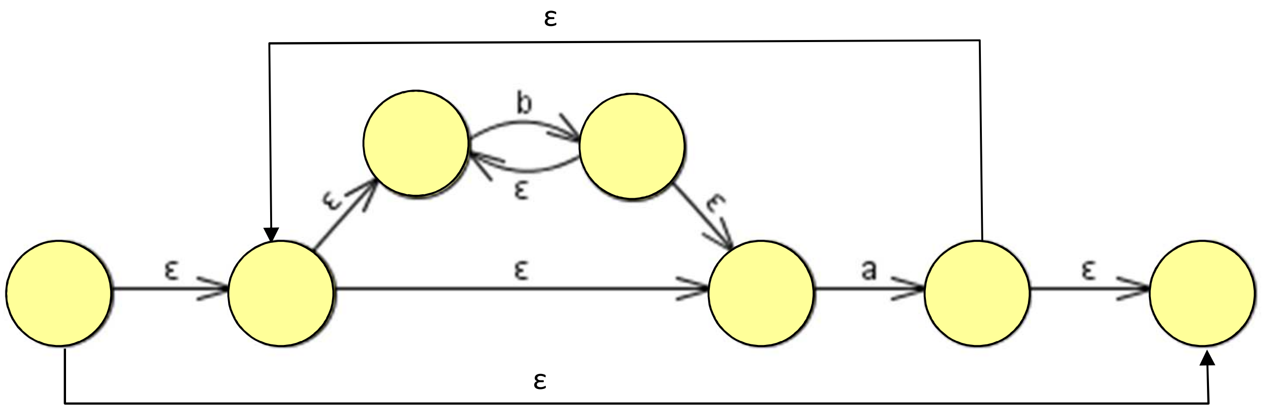
(b|(**b*a**)*)a



A partir de la ER **(b|(b*a)*)a**

3. Aplicar la Cerradura de Kleene al paréntesis

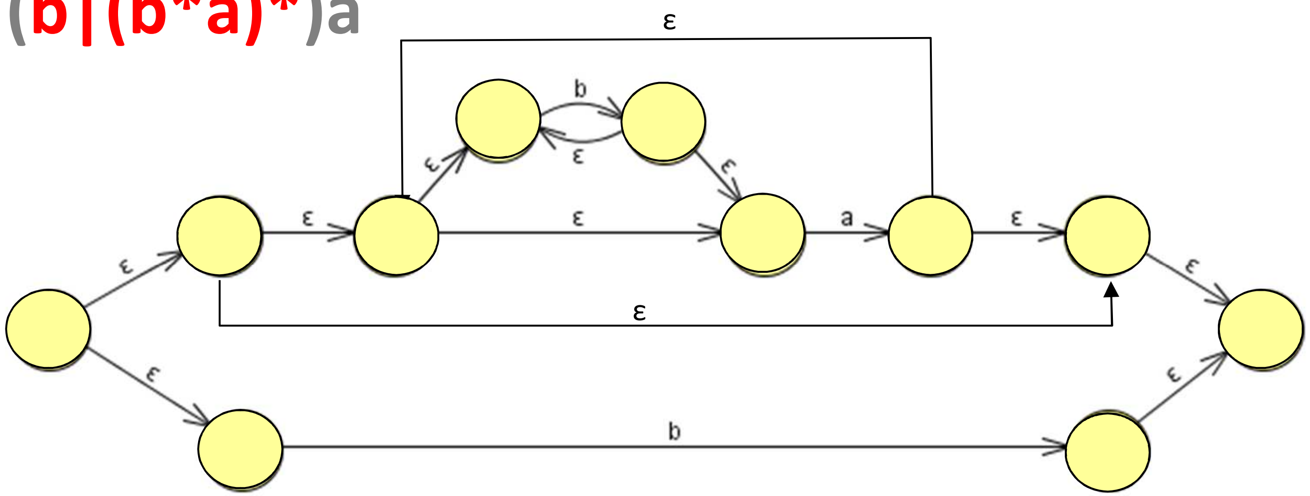
(b|**(b*a)***)a



A partir de la ER **(b|(b*a)*)a**

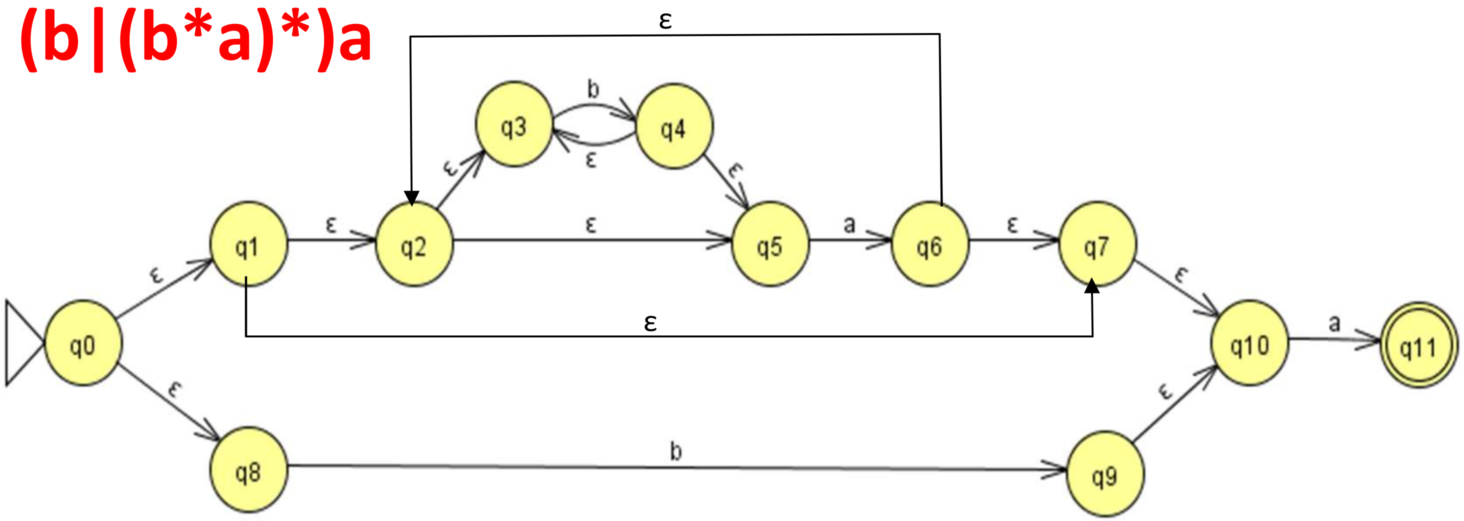
4. La elección de alternativas del b y el diagrama anterior.

(b|(b*a)*)a



5. Concatenamos el último símbolo, enumerando e indicando el estado inicial y el final

(b|(b*a)*)a



6. Formalizando

L((b|(b*a)*)a)=L(AFN)

AFN=(E,Q,f,q0,F)

E={a,b}

Q={q0,q1,q2,q3,q4, q5,q6,q7,q8,q9,q10,q11}

F={q11}

f: E*x Q→Q

f	A	b	ε
q0	-	-	{q1, q8}
q1	-	-	{q2, q7}
q2	-	-	{q3, q5}
q3	-	q4	-
q4	-	-	q5
q5	q6	-	-
q6	-	-	{q7, q2}
q7	-	-	q10
q8	-	q9	-
q9	-	-	q10
q10	q11	-	-
Q11	-	-	-

Ejercicios 07: Construcción de AFN´s con Thompson

- Construir grafica y formalmente los autómatas para las siguientes expresiones regulares a través de la nomenclatura de Thompson.

1. **(abc)***
2. **(b|bc)$^{+}$**
3. **letra_(letra_|digito)***
4. **(a|b)*abb**
5. **[(b|b*a)*]a**
6. **(a*|b^{+})** $^{+}$

Se entregarán antes del día **Lunes 30 de Septiembre de 2013 (23:59:59 hora limite).*

**Sugerencia utilizar Jflap para el dibujo y simulación de los autómatas*

**Incluir la redacción de cada ejercicio*

**Portada y encabezados de pagina*