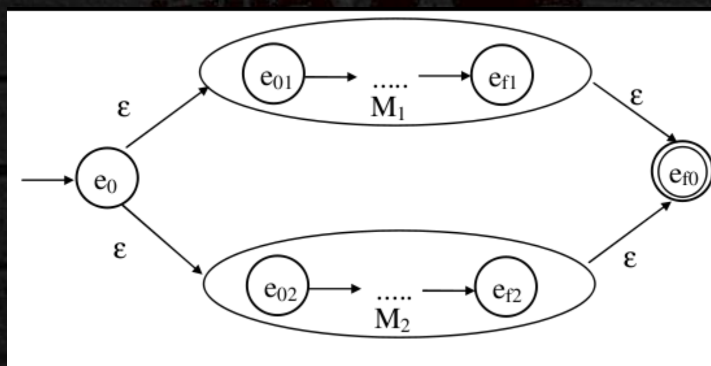
ESCOM

Transiciones épsilon y Construcción de Thompson



ESCOM

Contenido

- Autómatas finitos con transiciones-ε
- Notación formal para un AFN-ε
- Transformación de una expresión regular en un autómata finito
- Construcción de Thompson
- Nomenclatura de Thompson

Autómatas finitos con transiciones-ε

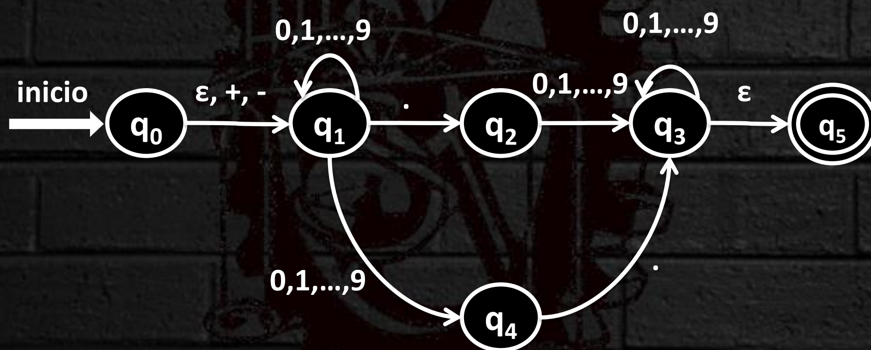
- Otra extensión de los autómatas finitos es la que permite transiciones para **ε**, la cadena vacía. Así, un AFN puede hacer una transición espontáneamente, sin recibir un símbolo de entrada.
- Esta nueva capacidad no expande la clase de lenguajes que los autómatas finitos pueden aceptar, pero proporciona algunas facilidades de programación.
- Las transiciones-ε están estrechamente relacionadas con las expresiones regulares resultan útiles para demostrar la equivalencia entre las clases de lenguajes aceptados por los autómatas finitos y las expresiones regulares

Autómatas finitos con transiciones-ε

- A los AFN con transiciones-ε los denominaremos AFN-ε
- Los autómatas aceptarán aquellas secuencias de etiquetas que siguen caminos desde el estado inicial a un estado de aceptación. Sin embargo, cada ε que se encuentra a lo largo de un camino es “invisible”; es decir, no contribuye a la cadena que se forma a lo largo del camino.

Ejemplo 1: AFN- ε

- Un AFN- ε que acepta números decimales.



Notación formal para un AFN-ε

- Podemos representar un AFN-ε del mismo modo que representaríamos un AFN con una excepción: la función de transición tiene que incluir la información sobre las transiciones para ε.

Ejemplo 1: AFN- ε

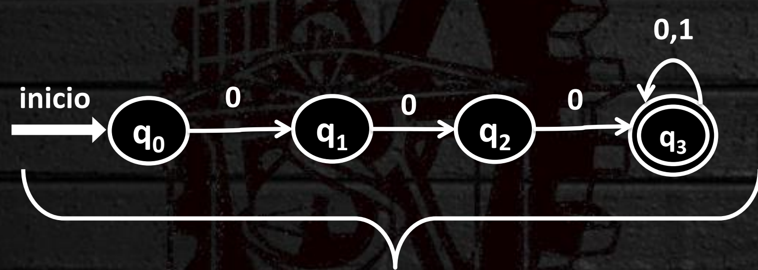
Δ	ε	+,-	.	0,1,…,9
->q_0	q_1	q_1	Ø	Ø
q_1	Ø	Ø	q_2	{q_1, q_4}
q_2	Ø	Ø	Ø	q_3
q_3	q_5	Ø	Ø	q_3
q_4	Ø	Ø	q_3	Ø
*q_5	Ø	Ø	Ø	Ø

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.**

ESCOM

Ejemplo 1



000(0|1)*

ESCOM

Construcción de Thompson

- **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***

Nomenclatura de Thompson

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

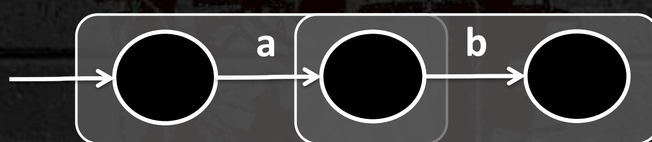
a

ESCOM

Nomenclatura de Thompson

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

ab

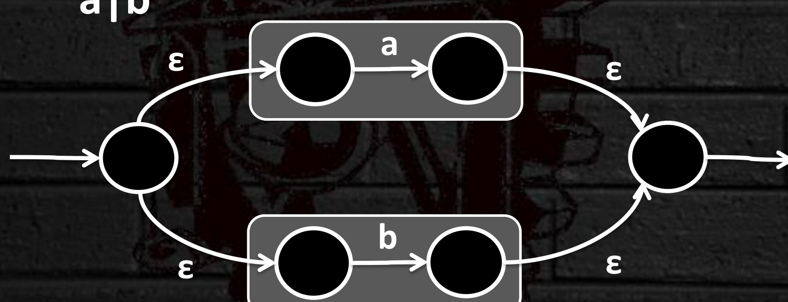


ESCOM

Nomenclatura de Thompson

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

a|b

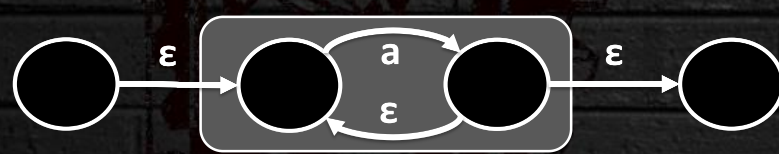


ESCOMM

Nomenclatura de Thompson

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

a^+

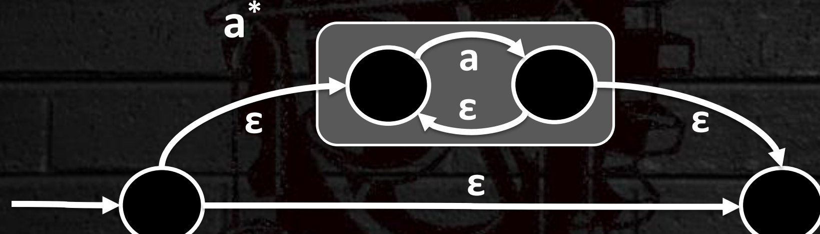


ESCOM

Nomenclatura de Thompson

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

a^*



ESCOM

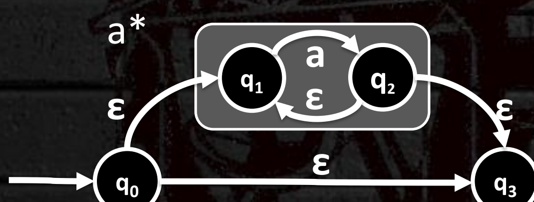
Ejemplo 1: Thompson

- Construir el diagrama del AFN que representa la ER **a*b**.

ESCOM

Ejemplo 1: Thompson

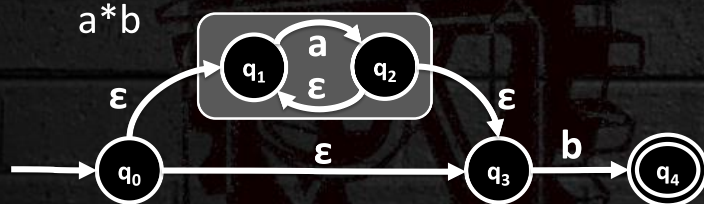
- Construir el diagrama del AFN que representa la ER **a*b**.



Ejemplo 1: Thompson

- Construir el diagrama del AFN que representa la ER **a*b**.

a*b



Ejemplo 1: Thompson

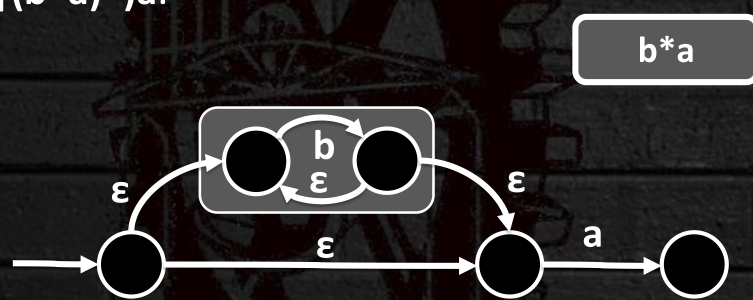
Formalizando:

- AFN=(Q,Σ,Δ,q_0,F)
- Σ ={a,b}
- Q={q_0,q_1,q_2,q_3,q_4}
- F={q_4}

Δ	ε	a	b
->q_0	{q_1, q_3}	Ø	Ø
q_1	Ø	q_2	Ø
q_2	{q_1, q_3}	Ø	Ø
q_3	Ø	Ø	q_4
*q_4	Ø	Ø	Ø

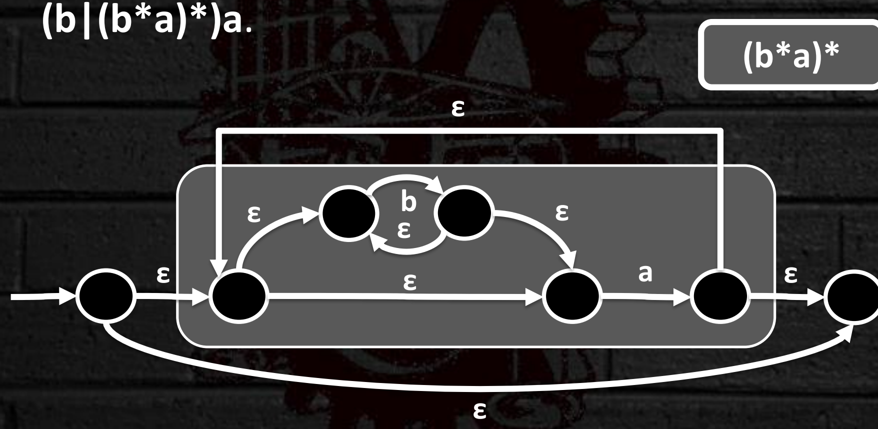
Ejemplo 2: Thompson

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



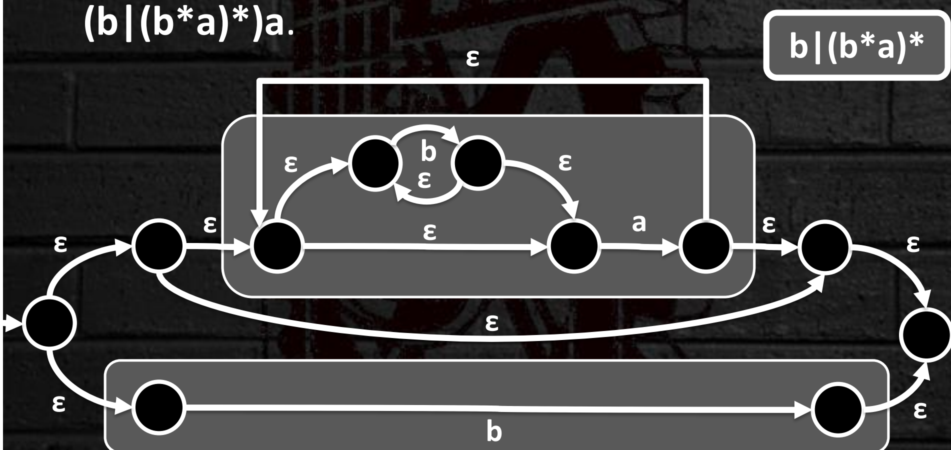
Ejemplo 2: Thompson

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



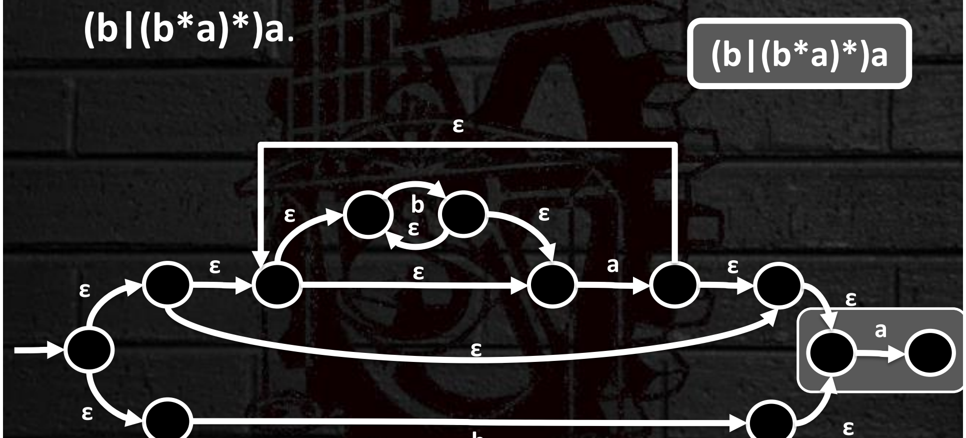
Ejemplo 2: Thompson

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



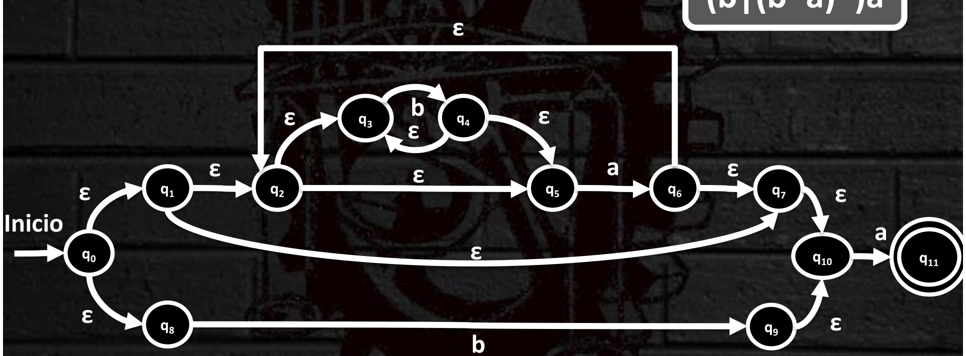
Ejemplo 2: Thompson

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



Ejemplo 2: Thompson

- Finalmente enumerando los estados e indicando el estado inicial y el final



Ejemplo 2: Thompson

Formalizando:

- AFN=(Q,Σ,Δ,q_0,F)
- Σ ={a,b}
- Q=$\{q_0,q_1,q_2,q_3,q_4,q_5,q_6,q_7,q_8,q_9,q_{10},q_{11}\}$
- F=$\{q_{11}\}$

Δ	ε	a	b
->q_0	$\{q_1, q_8\}$	Ø	Ø
q_1	{q2, q7}	Ø	Ø
q_2	$\{q_3, q_5\}$	Ø	Ø
q_3	Ø	Ø	q_4
q_4	q_5	Ø	Ø
q_5	Ø	q_6	Ø
q_6	$\{q_7, q_2\}$	Ø	Ø
q_7	q_{10}	Ø	Ø
q_8	Ø	Ø	q_9
q_9	q_{10}	Ø	Ø
q_{10}	Ø	q_{11}	Ø
*q_{11}	Ø	Ø	Ø

ESCOMN

Ejercicios

- Construir el **diagrama y formalizar** los autómatas para las siguientes expresiones regulares a través de la nomenclatura de Thompson.

1. **(a|b|c)*b***
2. **(a|b)***
3. **(a*b*c*)***
4. **$(bc)^{+}|(ab)^{*}$**
5. **((b|b*a)*)a**
6. **$(a^{*}|b^{+})^{+}$**