Convertir autómatas a regex

Dr. José Lázaro Martínez Rodríguez

Expresiones regulares

- Notación para especificar un lenguaje
  - Declarativo
  - Algo así como un lenguaje de programación.
    - Fundamental en algunos lenguajes como perl y aplicaciones como grep o lex
  - Capaz de describir lo mismo que un AFN
    - Los dos son realmente equivalentes, así que RE = NFA = DFA
  - Podemos definir un álgebra para las expresiones regulares

Algebra para lenguajes

- Anteriormente hemos hablado de estos operadores:
  - Unión
  - Concatenación
  - Cerradura (Estrella) de Kleene

Expresión regular

- R es una expresión regular si es
  1. a para alguna *a* en el alfabeto ∑, representando el lenguaje {a}
  2. ε, que representa el lenguaje {ε}
  3. **R1+R2** donde R1 y R2 son expresiones regulares, y + significa unión
     1. (a veces se usa |)
  4. **R1R2** donde R1 y R2 son expresiones regulares y esto significa concatenación
  5. R* donde R es una expresión regular y significa cierre
  6. (R) donde R es una expresión regular, entonces una R entre paréntesis es también una expresión regular

Precedencia: Los paréntesis tienen la mayor precedencia, seguidos de *, la concatenación y la unión.

Ejemplos de regex

- L(001) = {001}
- L(0+10*) = { 0, 1, 10, 100, 1000, 10000, … }
- L(0*10*) = {1, 01, 10, 010, 0010, …} es decir, {w | w tiene exactamente un solo 1}
- L(ΣΣ)* = {w | w es una cadena de longitud par}
- L((0(0+1))*) = { ε, 00, 01, 0000, 0001, 0100, 0101, …}
- L((0+ε)(1+ ε)) = {ε, 0, 1, 01}
- L(1Ø) = Ø ; al concatenar el conjunto vacío con cualquier conjunto se obtiene el conjunto vacío.
- Rε = R
- R+Ø = R

- Nótese que R+ε puede o no ser igual a R (estamos añadiendo ε al lenguaje)
- Nótese que RØ sólo será igual a R si el propio R es el conjunto vacío.

Ejercicio de regex

- Ejercicio: Escriba una expresión regular para el conjunto de cadenas que contiene un número par de 1's sobre $\sum = \{0,1\}$. Tratar cero 1's como un número par.

Equivalencia de AF y ER

- Los autómatas finitos y las expresiones regulares son equivalentes. Para demostrarlo:
  - Demostrar que podemos expresar un DFA como una RE equivalente
  - Demostrar que podemos expresar un RE como un ε-NFA. Dado que el ε-NFA se puede convertir en un DFA y el DFA en un NFA, entonces RE será equivalente a todos los autómatas que hemos descrito.

Convertir un AFD en un ER

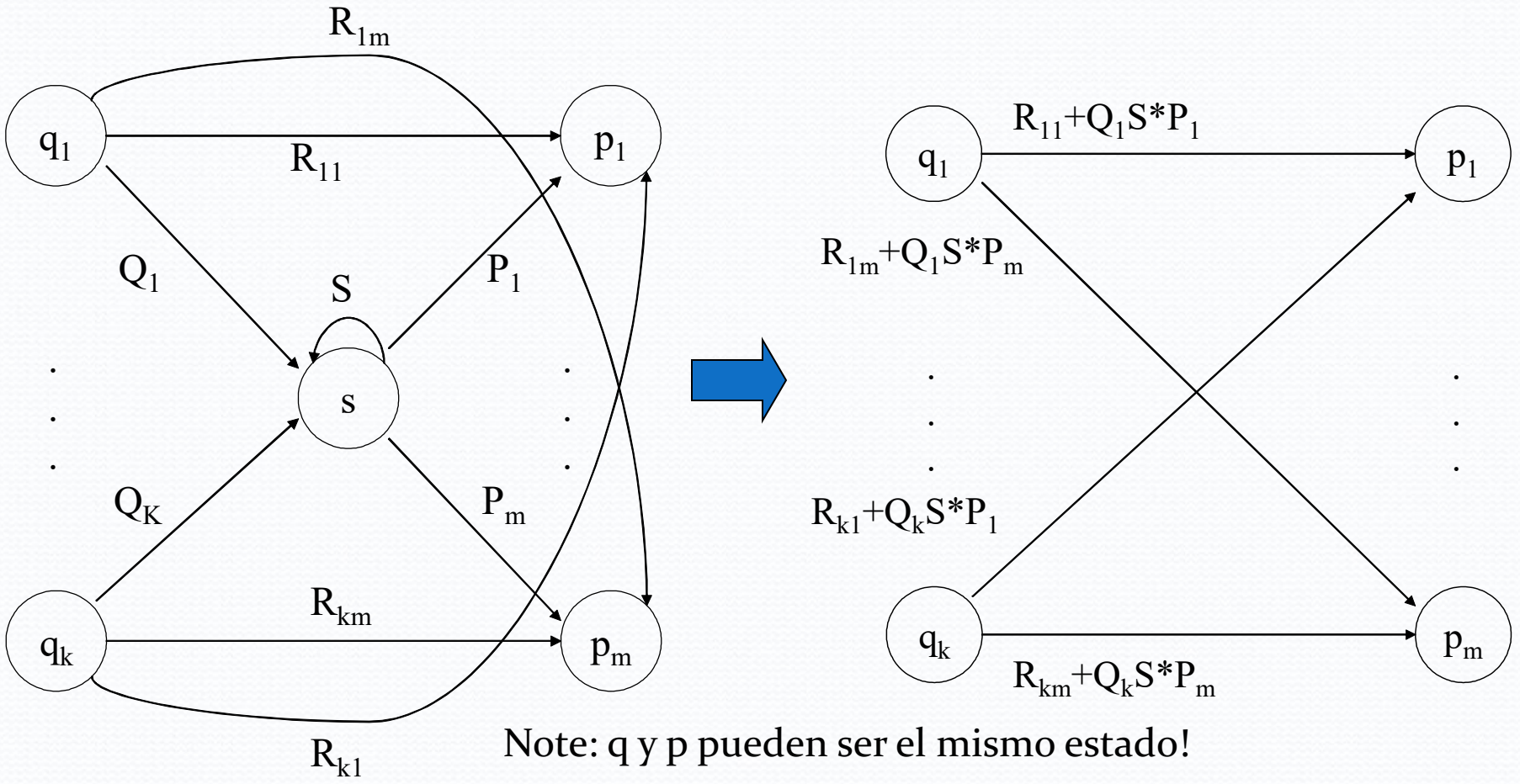
- Teorema: Si L=L(A) para algún AFD *A*, entonces existe una expresión regular R tal que L=L(R).
- Prueba:
  - Construir GNFA, NFA Generalizado
    - Nos saltaremos esto en la clase (por ahora)
  - Eliminación de estados
    - Veremos cómo hacer esto a continuación, más fácil que la construcción inductiva, no hay un número exponencial de expresiones

AFD a ER

- Elimina los estados del autómata y sustituye las aristas por expresiones regulares que incluyen el comportamiento de los estados eliminados.
- Finalmente llegamos a la situación con sólo un nodo inicial y final, y esto es fácil de expresar como una ER

Eliminación del Estado

- Considere la figura siguiente, que muestra un estado genérico s a punto de ser eliminado. Las etiquetas de todas las aristas son expresiones regulares.
- Para eliminar s, debemos hacer etiquetas desde cada q_i a p_1 hasta p_m que incluyan los caminos que podríamos haber hecho a través de s.



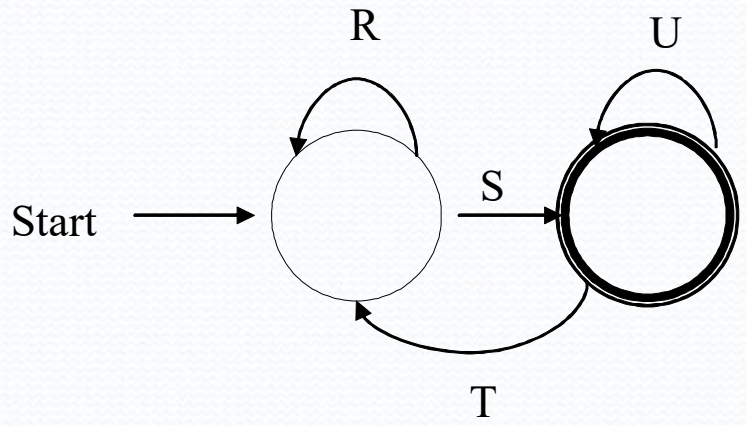
Note: q y p pueden ser el mismo estado!

DFA a RE a través de la eliminación del Estado (1)

1. Empezando por los estados intermedios y pasando por los estados de aceptación, aplique el proceso de eliminación de estados para producir un autómata equivalente con etiquetas de expresiones regulares en las aristas.
   - El resultado será un autómata de uno o dos estados con un estado inicial y un estado de aceptación.

DFA a RE a través de la eliminación del Estado (2)

2. Si los dos estados son diferentes, tendremos un autómata con el siguiente aspecto :



-Notar ciclos
-Notar concatenaciones
-Notar Unión

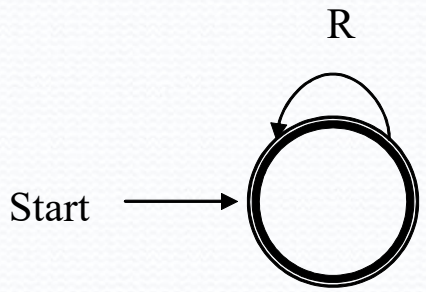
Podemos describir este autómata como :

(R*SU*T)*SU*

Todos los caminos posibles hasta llegar al estado de aceptación

DFA a RE a través de la eliminación del Estado (3)

3. Si el estado inicial también es un estado de aceptación, entonces también debemos realizar una eliminación de estados del autómata original que elimine todos los estados menos el estado inicial. Esto deja lo siguiente :



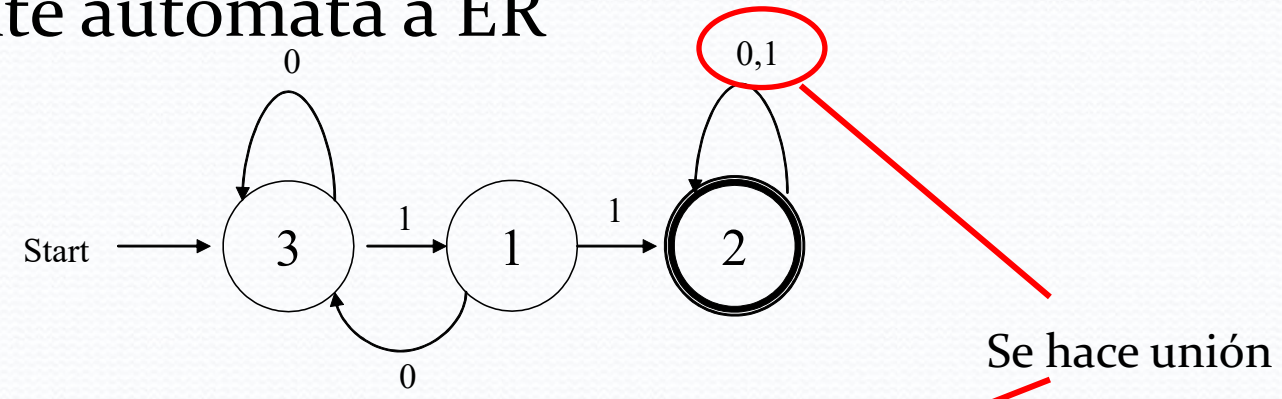
Podemos describir este autómata simplemente como R*

DFA a RE a través de la eliminación del Estado (4)

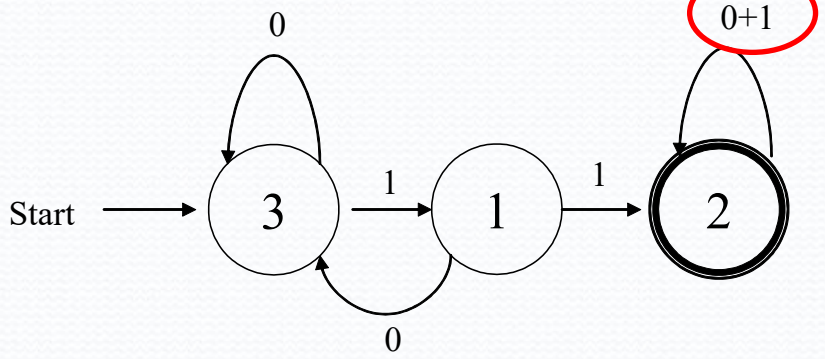
4. Si hay n estados de aceptación, debemos repetir los pasos anteriores para cada estado de aceptación para obtener n expresiones regulares diferentes, R_1, R_2, … R_n.
   - En cada repetición convertimos cualquier otro estado aceptable en no aceptable.
   - La expresión regular deseada para el autómata es entonces la unión de cada una de las n expresiones regulares: $R1 \cup R2 \ldots \cup R_N$

AFD→ER Ejemplo

- Convertir el siguiente autómata a ER

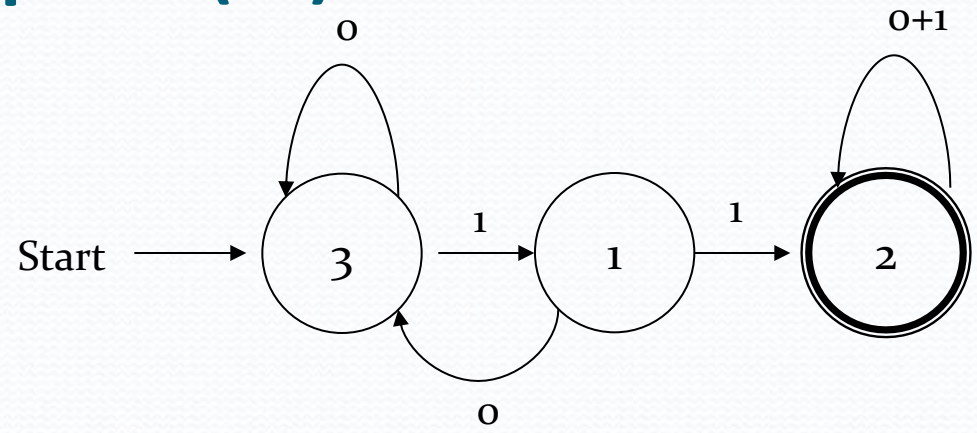


- Primero convertir aristas a ER's:



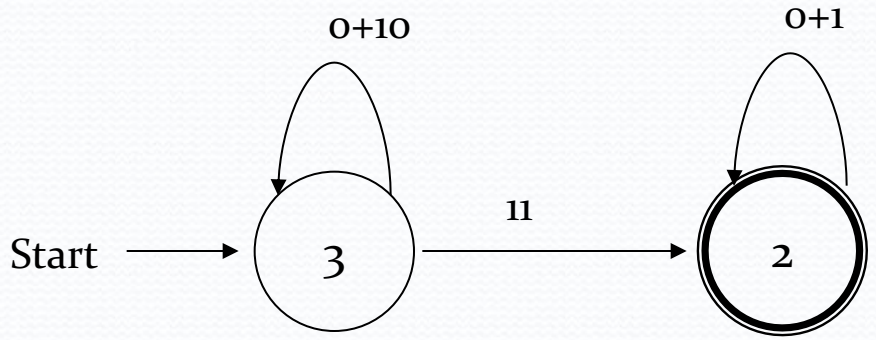
AFD→ER Ejemplo (2)

- Eliminar Estado 1:



- Entonces:

Notar arista de 3→3

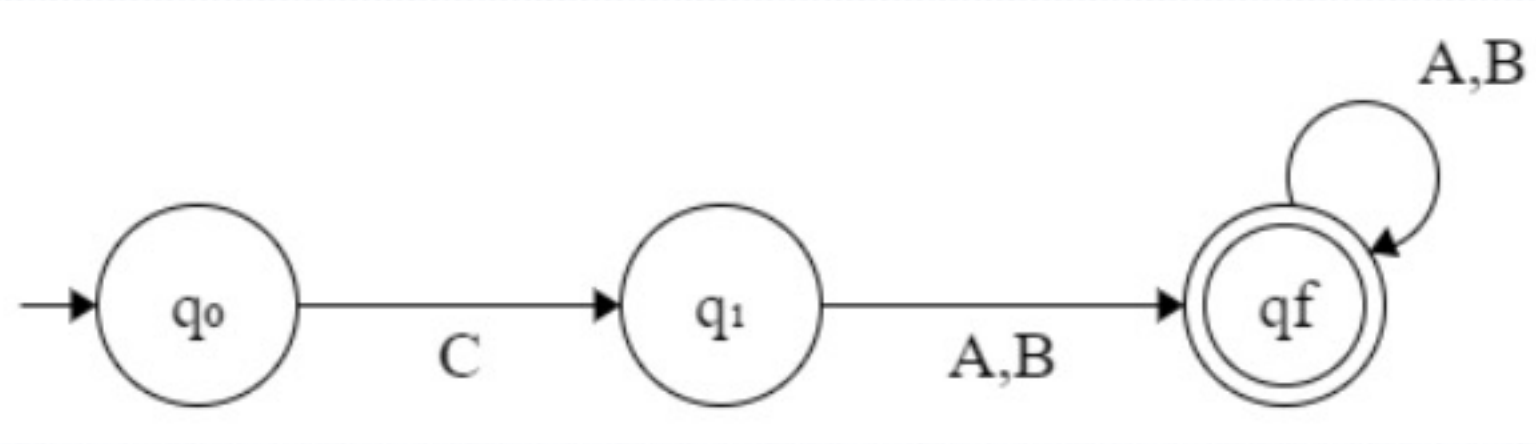


Respuesta: (0+10)*11(0+1)*

(0+10)*11(0+1)*

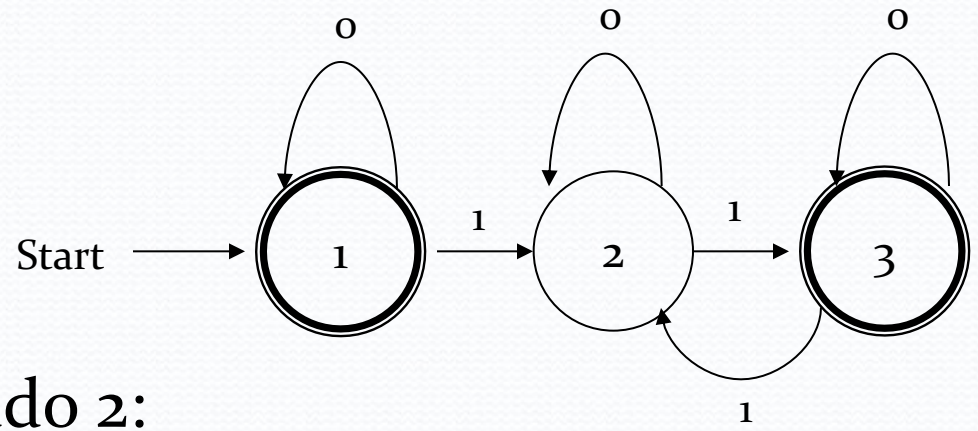
Práctica

- ¿Cómo sería este caso?

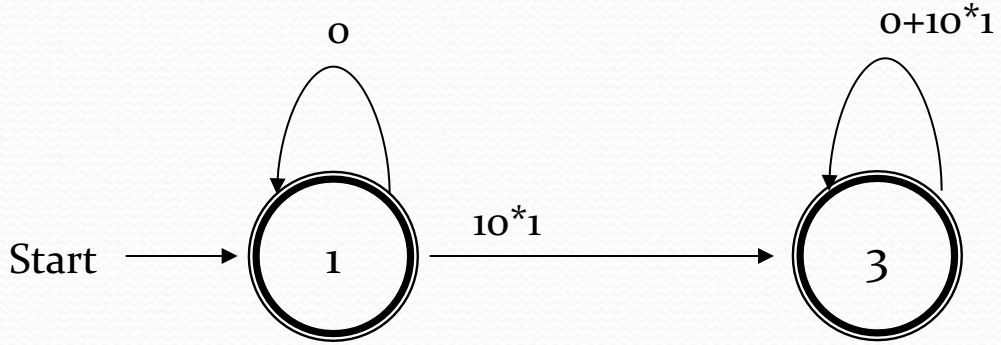


Segundo ejemplo

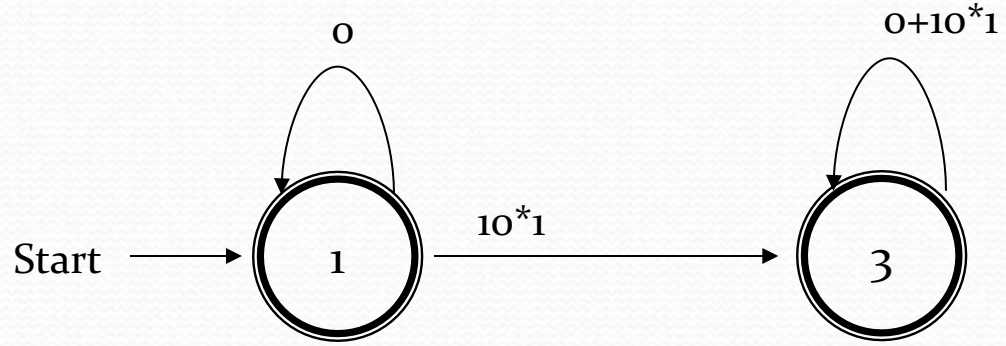
- Automata que acepta número par de 1's



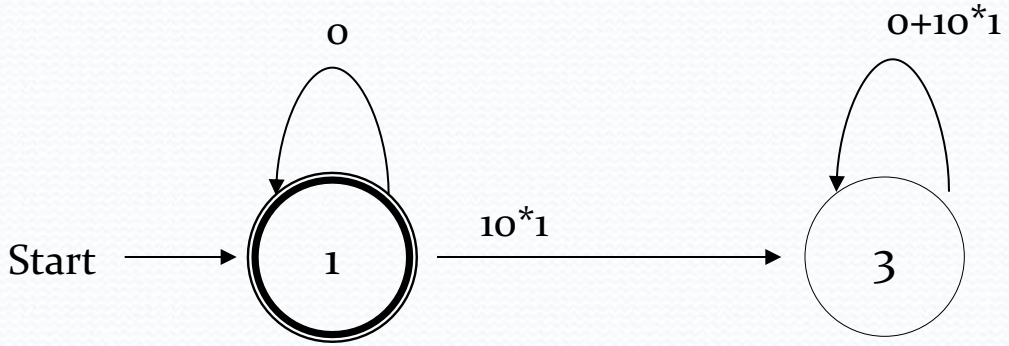
- Eliminar estado 2:



Segundo ejemplo (2)

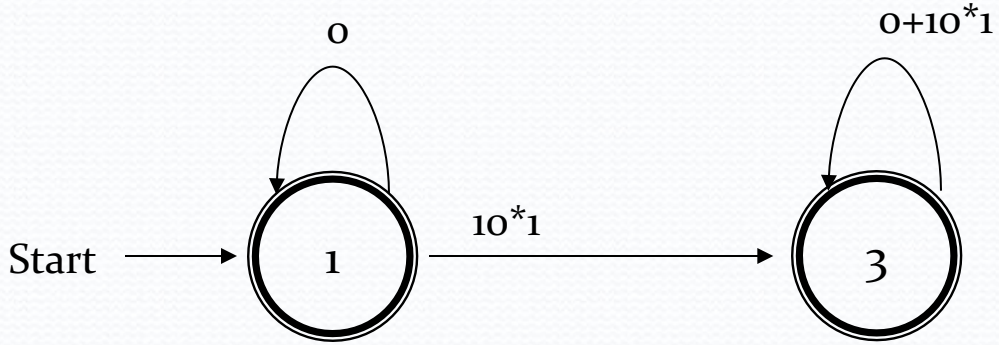


- Dos estados de aceptación, apague primero el estado 3

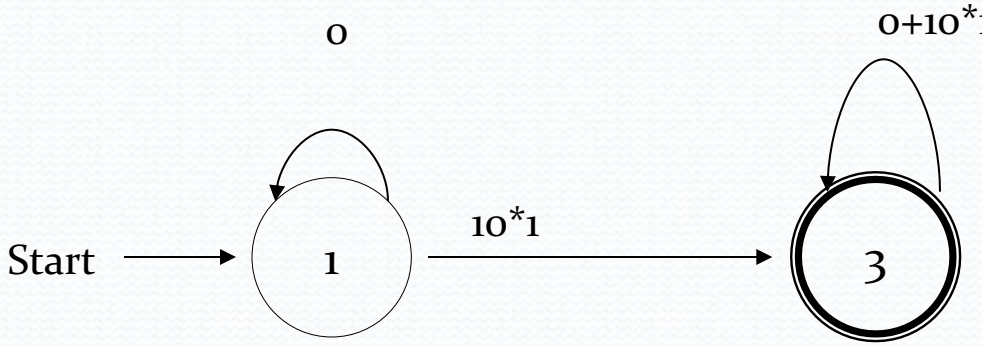


Esto es sólo 0*; puede ignorar ir al estado 3 ya que "moriríamos"

Segundo ejemplo (3)



- Apagar el estado 1 en segundo lugar:

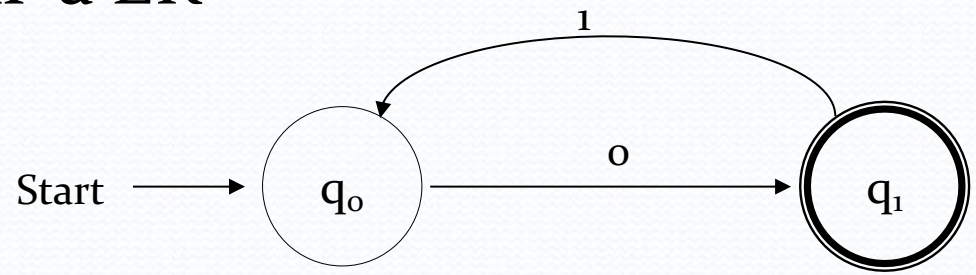


Esto es solo 0*10*1(0+10*1)*

Combinar de la diap. anterior para obtener
0* + 0*10*1(0+10*1)*

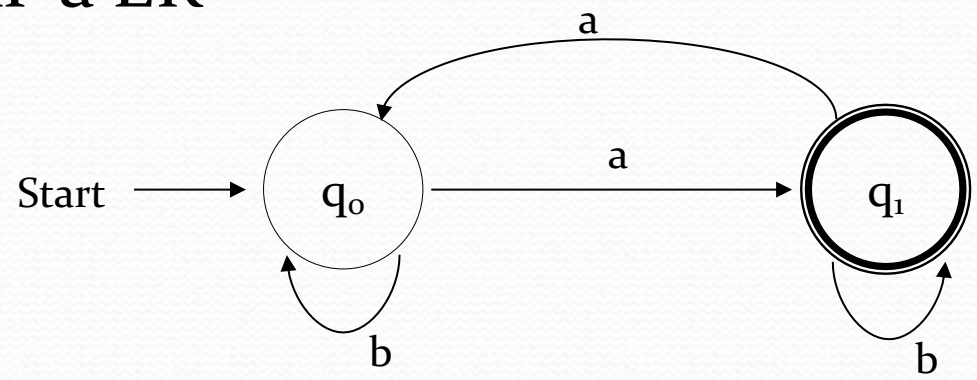
Tercer ejemplo

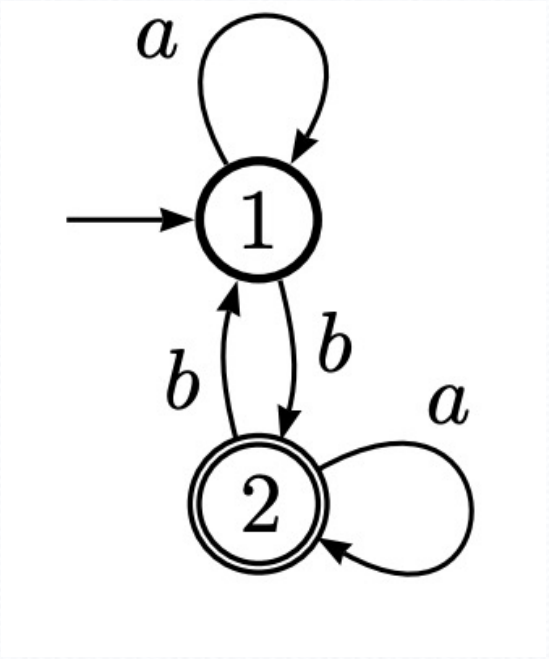
- Convertir AF a ER



Ejemplo

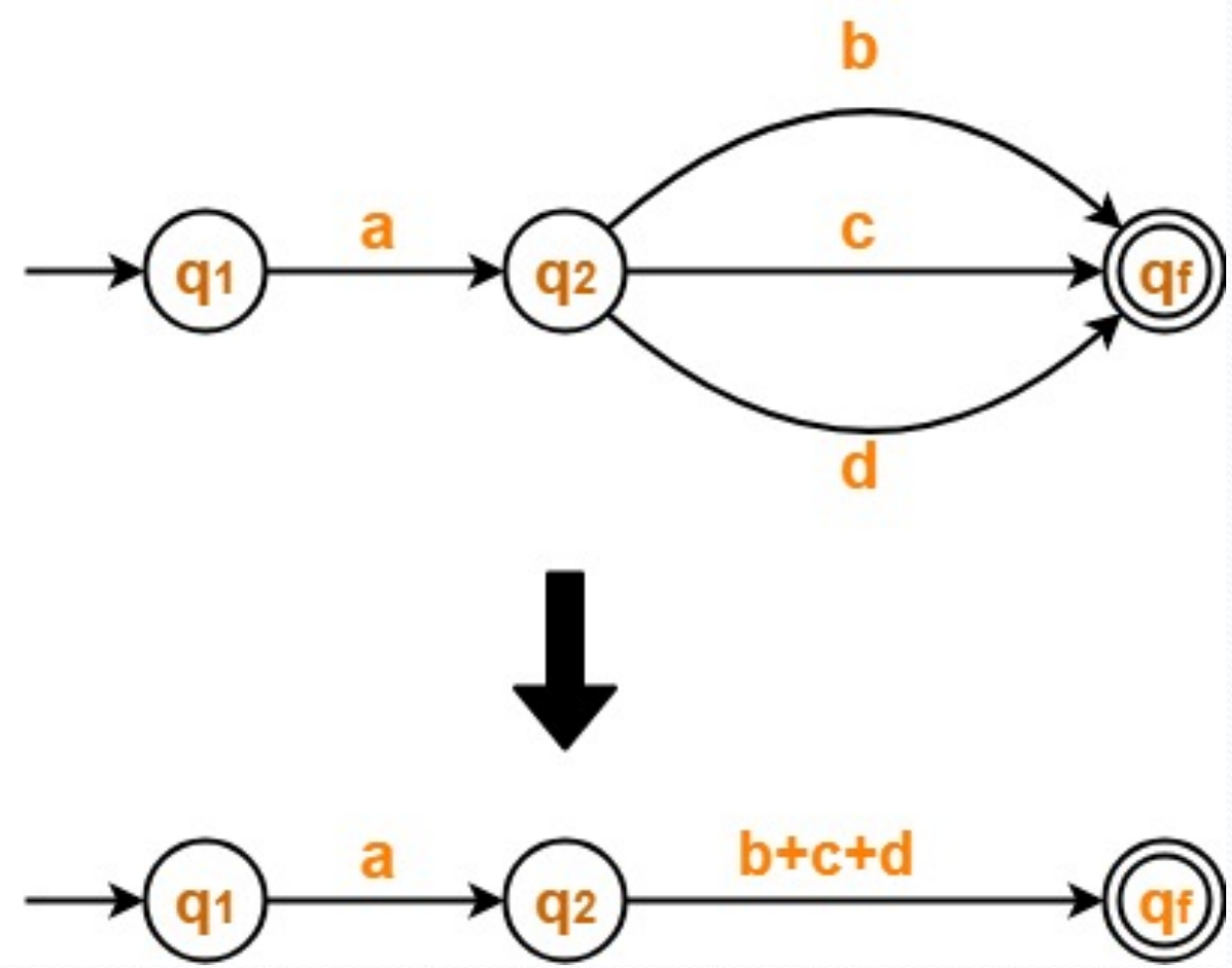
- Convertir AF a ER





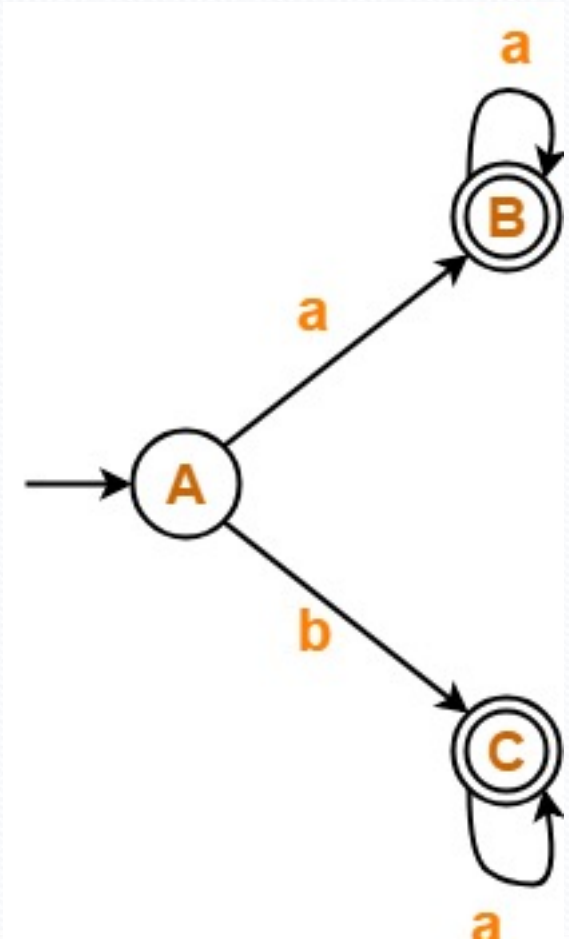
Ejemplo

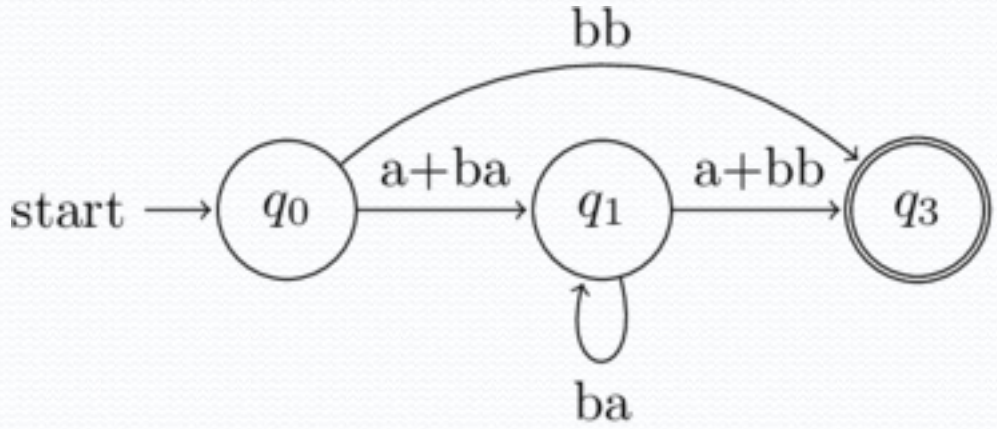
- Otro



Ejemplo

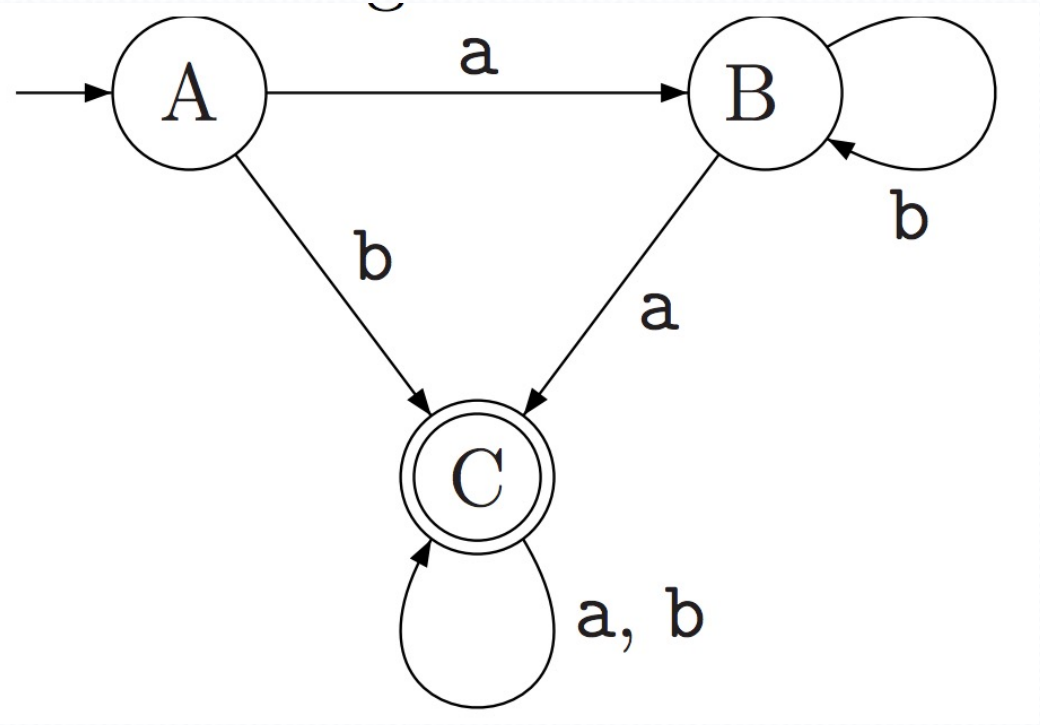
- Mas





Ejemplo

- Convertir AF a ER

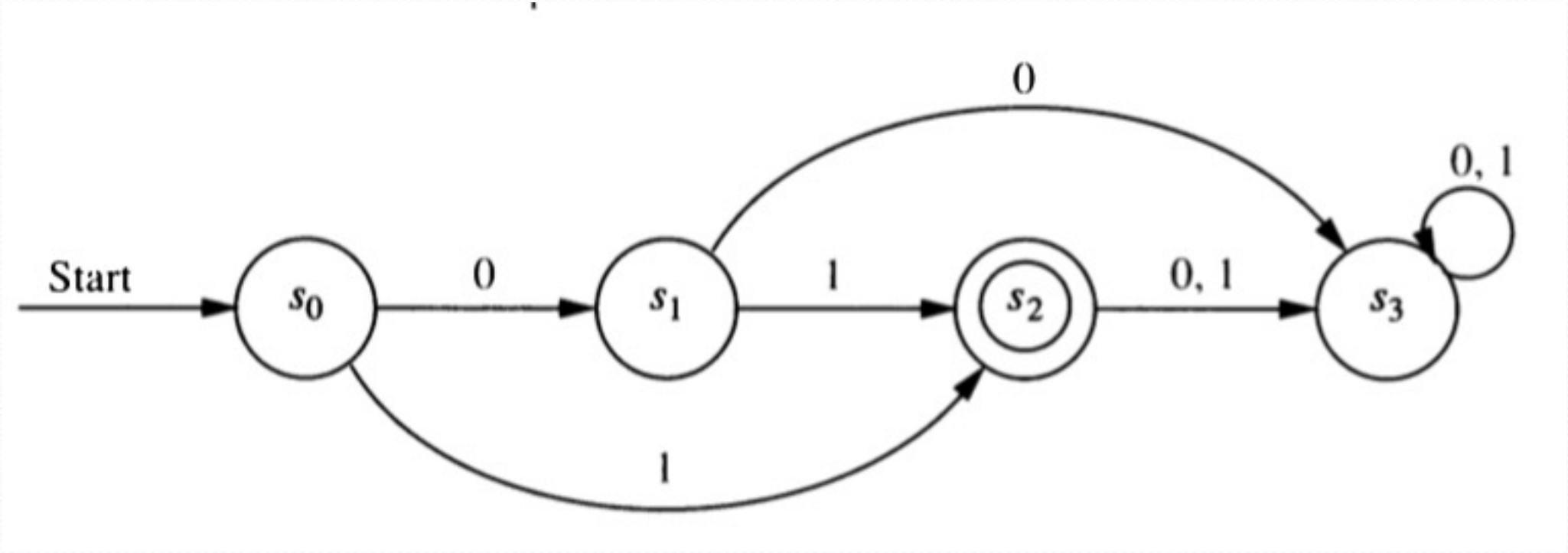


¿Cuál es la tabla de transiciones?

¿Qué cadenas acepta?

¿Qué regex representa este autómata?

Ejercicio 1



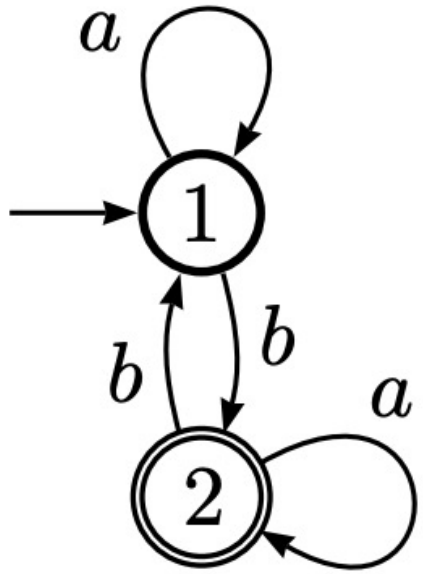
¿Cuál es la tabla de transiciones?

¿Qué cadenas acepta?

¿Qué regex representa este autómata?

Ejercicio 2

- Convertir AF a ER



¿Cuál es la tabla de transiciones?

¿Qué cadenas acepta?

¿Qué regex representa este autómata?