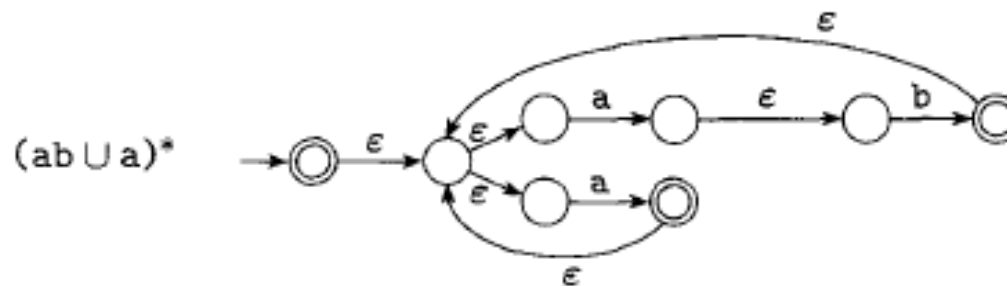
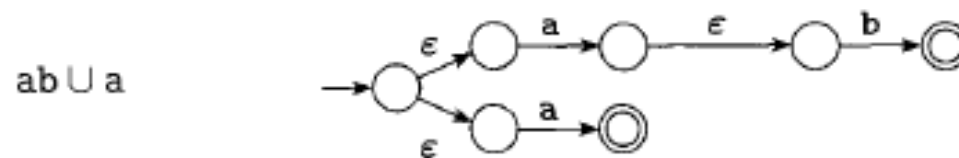
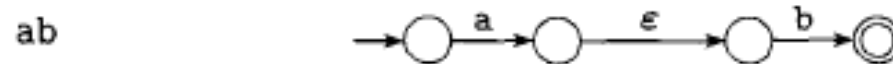


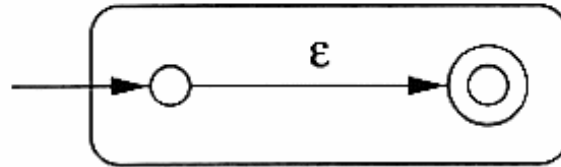
Equivalencia con AF

- Cualquier expresión regular puede ser convertida en un autómata finito que reconoce el lenguaje generado por ella.
- Un lenguaje es regular sí y sólo si es descrito por una expresión regular.

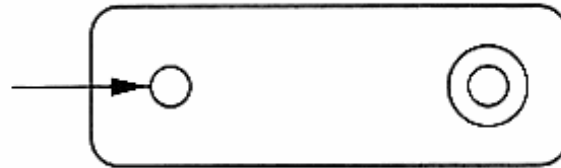
Ejemplo: Expresión $(ab \cup a)^*$



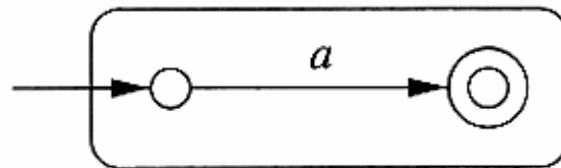
Bloques de construcción



(a)



(b)

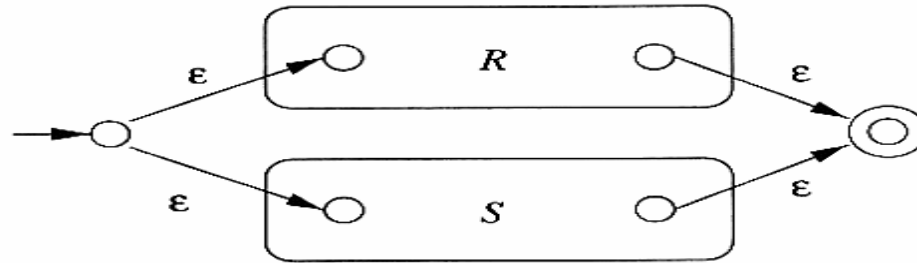


(c)

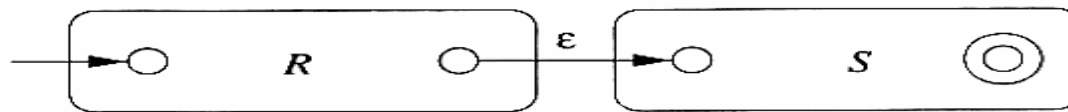
Bloques de construcción

- a) Para representar el lenguaje ε
 - b) Para representar el lenguaje \emptyset
 - c) Para representar la ER “a”
-

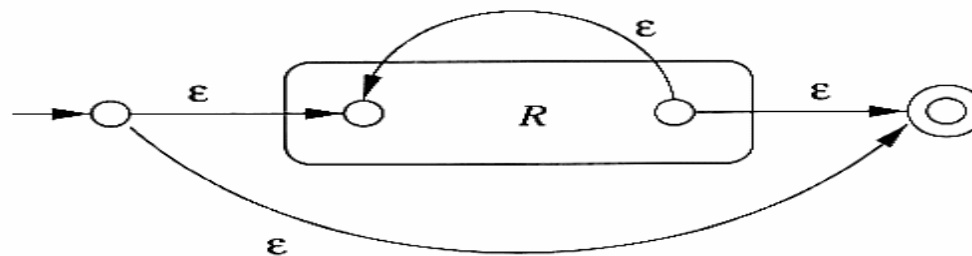
Bloques II



(a)



(b)



(c)

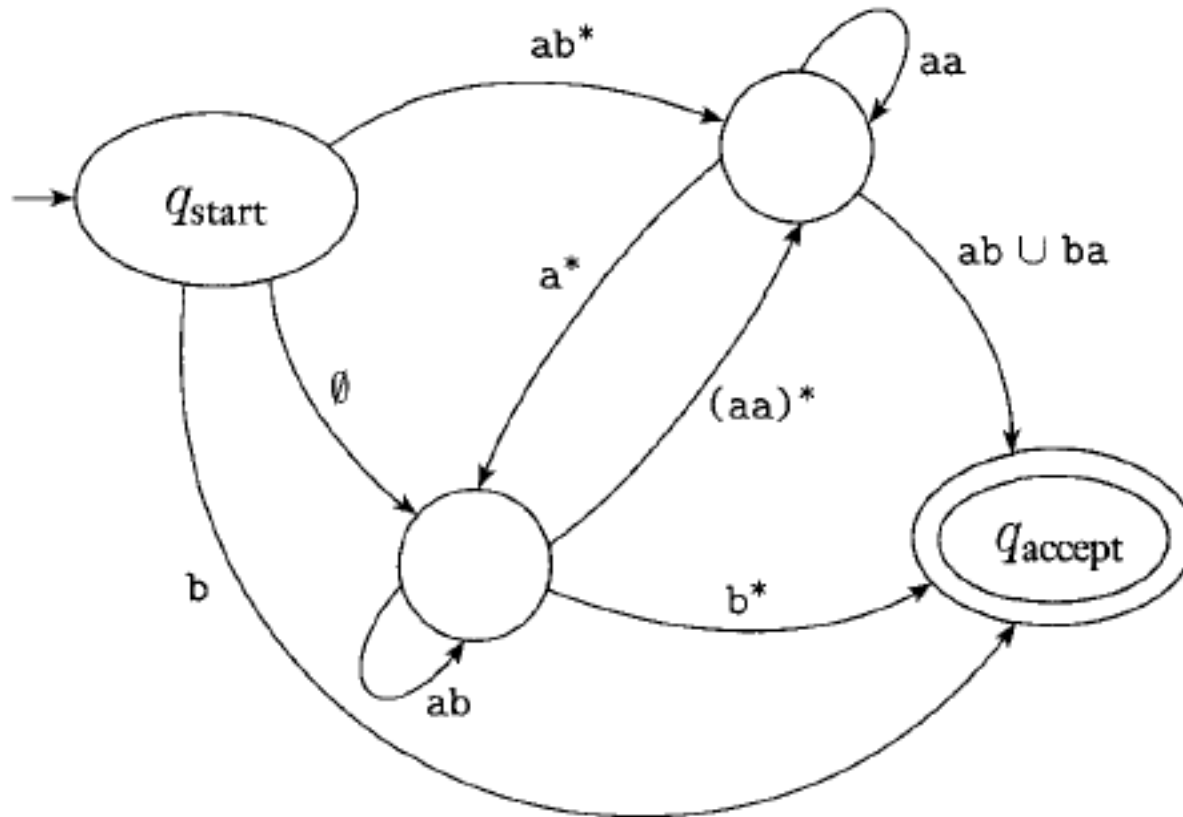
GNFA

- Si el lenguaje A es regular, puede ser descripto por una expresión regular.
- Si el lenguaje A es regular, puede ser representado por un AF.
- Utilizamos un nuevo tipo de autómata denominado Autómata Finito Nodeterminístico Generalizado

GNFA

- Son autómatas no determinísticos en los cuales las transiciones pueden estar etiquetadas con expresiones regulares, y no solamente con miembros del alfabeto.
- El GNFA lee BLOQUES de símbolos de la entrada (no uno a la vez)
- Los bloques de símbolos constituyen expresiones regulares, que producen transiciones entre los estados.

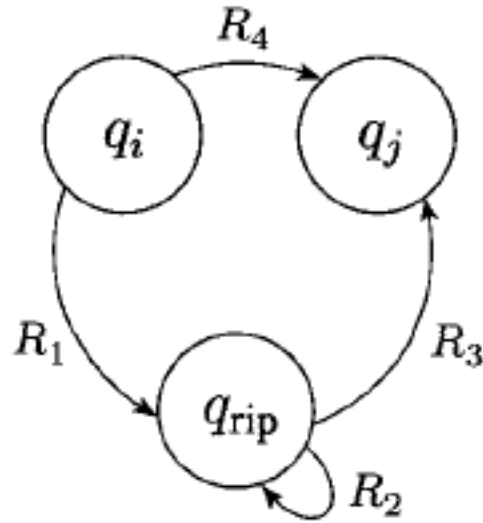
Ejemplo GNFA



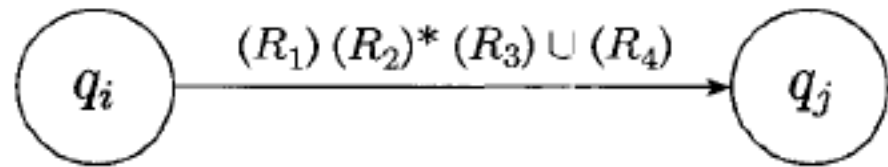
Particularidades

- El E1 posee transiciones hacia todos los demás estados, pero ninguna hacia él.
- El EF posee transiciones que provienen desde todos los demás estados, pero ninguna sale desde él a los demás.
- Existe un solo EF.
- Todos los demás estados pueden presentar transiciones entre ellos y hacia sí mismos.

Reduciendo la cantidad de estados



before



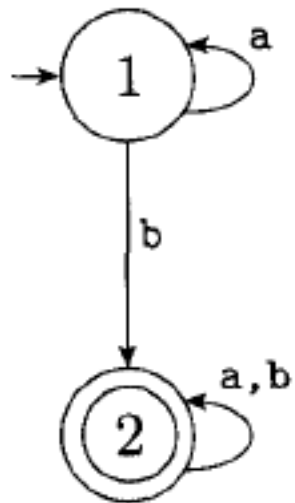
after

GNFA

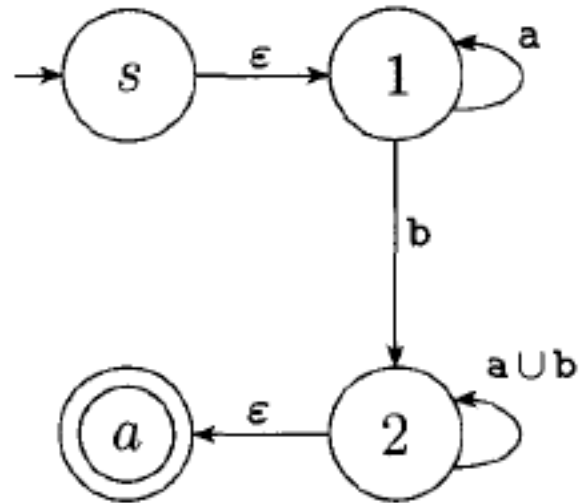
A *generalized nondeterministic finite automaton* is a 5-tuple, $(Q, \Sigma, \delta, q_{\text{start}}, q_{\text{accept}})$, where

1. Q is the finite set of states,
2. Σ is the input alphabet,
3. $\delta: (Q - \{q_{\text{accept}}\}) \times (Q - \{q_{\text{start}}\}) \longrightarrow \mathcal{R}$ is the transition function,
4. q_{start} is the start state, and
5. q_{accept} is the accept state.

Conversión Paso 1

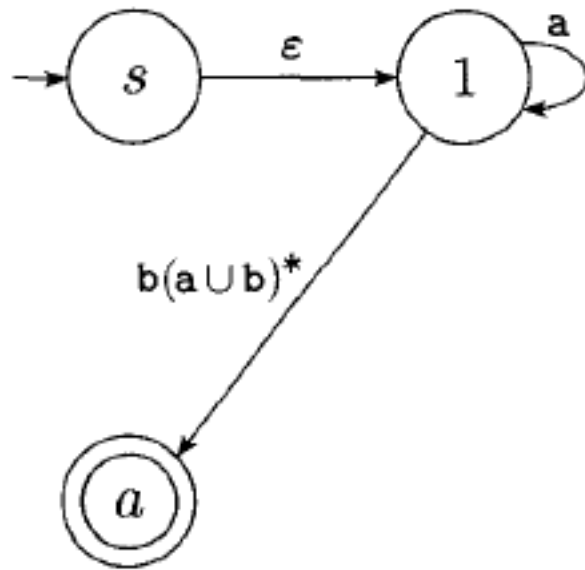


(a)

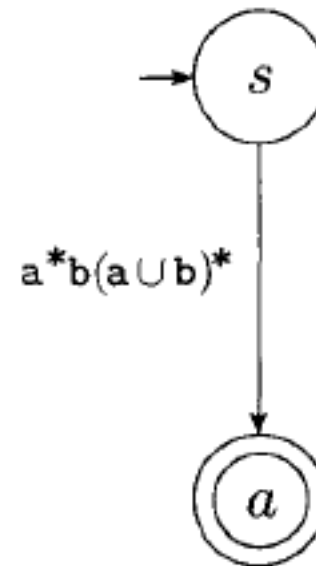


(b)

Conversión Paso 2



(c)



(d)

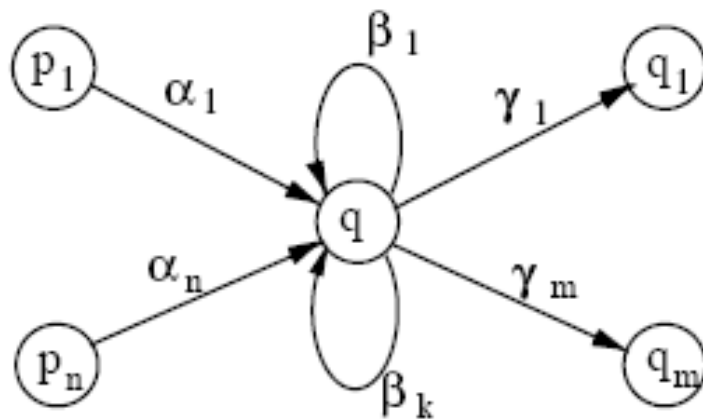
Reduciendo la cantidad de estados

- Eliminar nodos intermedios en la GT:
- Se llama nodo intermedio a aquel que se encuentra en una trayectoria entre el estado inicial y el final. El procedimiento de eliminación de nodos intermedios es directo.
- La idea es que al suprimir el nodo en cuestión, no se alteren las cadenas que hay que consumir para pasar de uno a otro de los nodos vecinos.

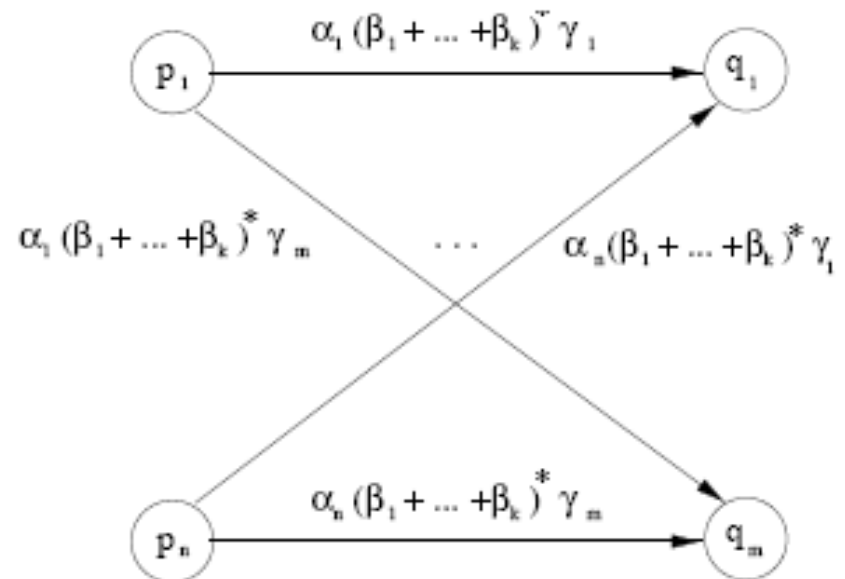
Reduciendo la cantidad de estados

- En otras palabras, al suprimir dicho nodo, se deben reemplazar las transiciones que antes tomaban ese nodo como punto intermedio para ir de un nodo vecino a otro, por otras transiciones que vayan del nodo vecino origen al nodo vecino destino, pero ahora sin pasar por el nodo eliminado.
-

Reduciendo la cantidad de estados



(a) Nodo a eliminar



(b) GT sin el nodo eliminado