2 Autómatas finitos y gramáticas regulares.

Autómata

- RAE
  - Instrumento o aparato que encierra dentro de sí el mecanismo que le imprime determinados movimientos.
- Algo autónomo que se comporta de determinada manera siempre.
- Un dispositivo que siempre hace lo mismo ante la misma situación.
- Tiene “estados” y “cambia” de estado en estado, por ejemplo, un elevador, las puertas automáticas (abierto-cerrado).
- Un autómata es un sistema que lo caracterizamos mediante estados y transiciones de estados (conjuntos finitos).

Autómata finito

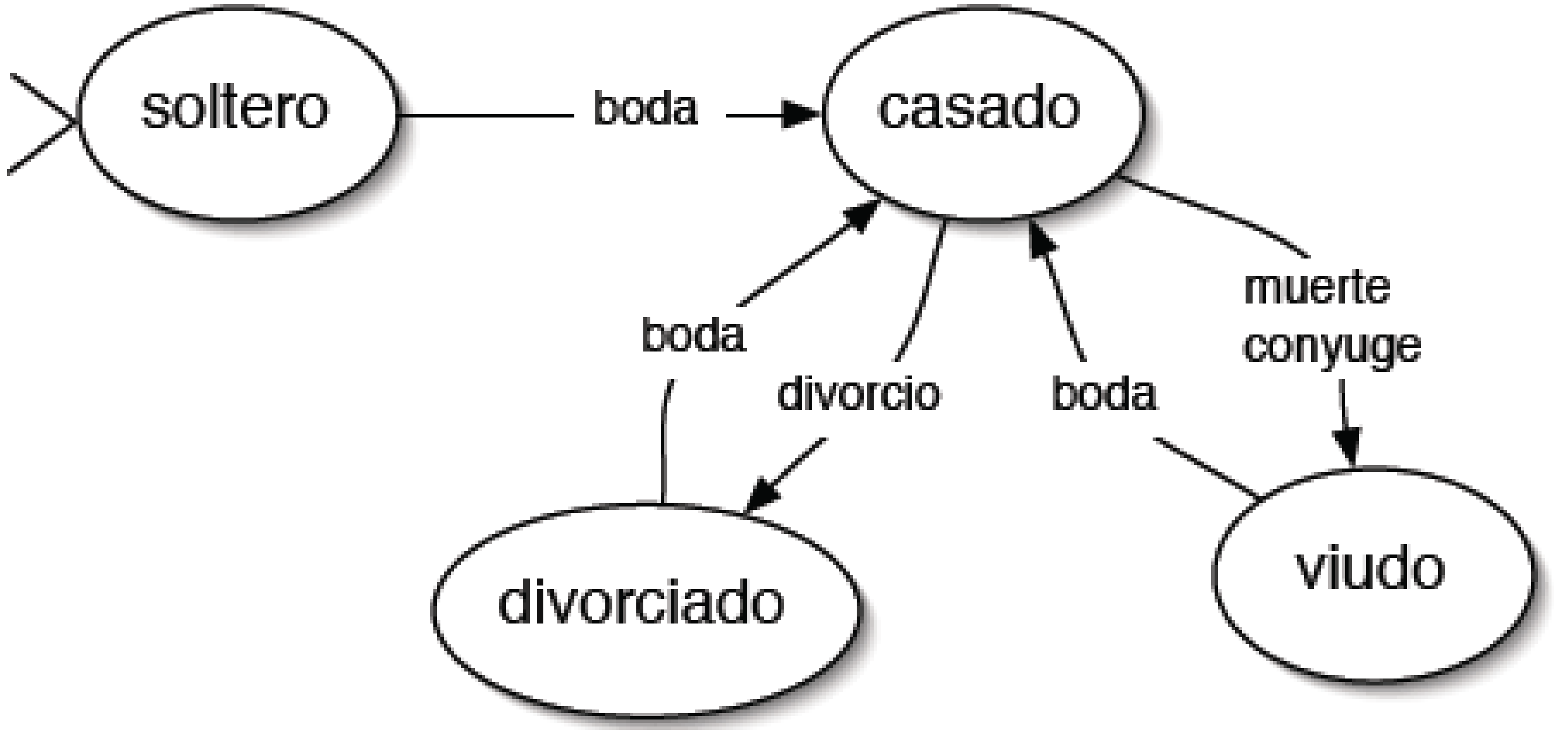
- Los AF son modelos matemáticos de un sistema que recibe entradas (estímulos) discretas y proporciona salidas (acciones) discretas.
- El sistema puede estar en cualquiera de un conjunto finito de estados.
- Un estado es una descripción instantánea del sistema, una especie de fotografía de la realidad congelada en un instante dado.

- Un cambio de estado se denomina transiciones, las cuales suceden de manera espontánea ó en respuesta a estímulos externos.
- Un sistema que consiste únicamente de un número finito de muchos estados y transiciones entre estados se denomina sistema de transición de estado finito ó simplemente sistema de estado finito.

Ejemplos, simples

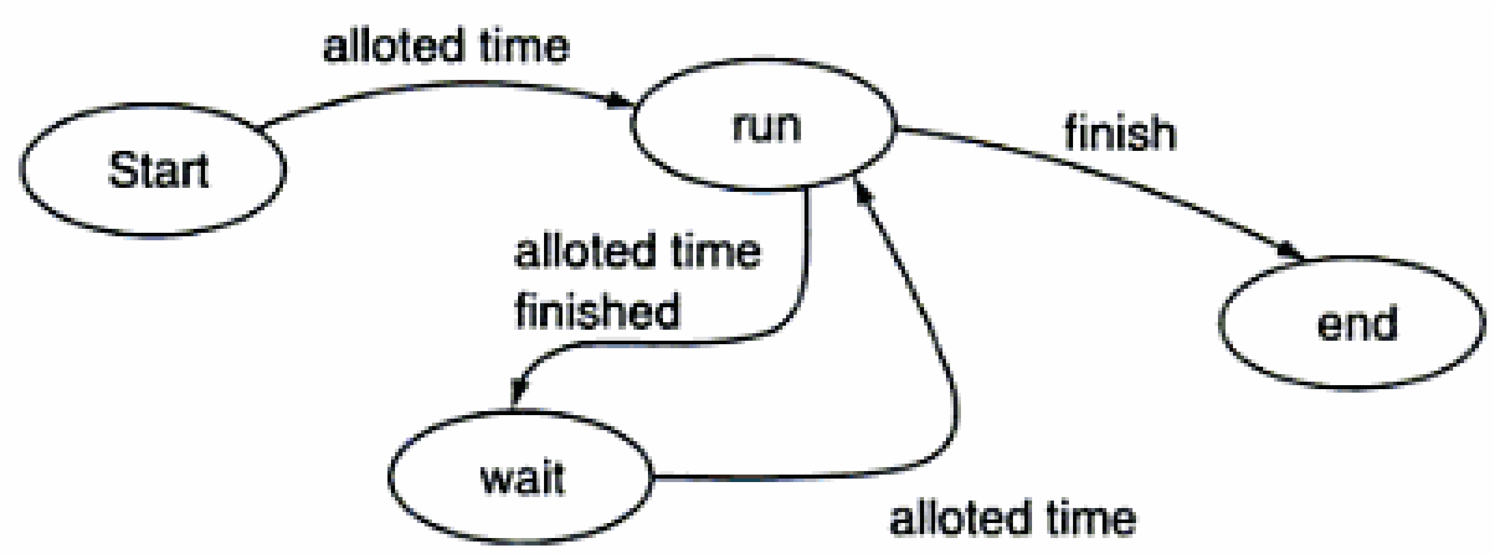
- Un estado es una situación en la que se permanece un cierto lapso de tiempo.
- Un ejemplo de la vida real es el de los “estados civiles” en que puede estar una persona: soltera, casada, viuda, divorciada, etc. De uno de estos estados se puede pasar a otro al ocurrir un evento o acción, que es el segundo concepto básico de la modelación discreta.
- Así, por ejemplo, del estado “soltero” se puede pasar al estado “casado” al ocurrir el evento “boda”.
- Similarmente, se puede pasar de “casado” a “divorciado” mediante el evento “divorcio”.

Ejemplos simples



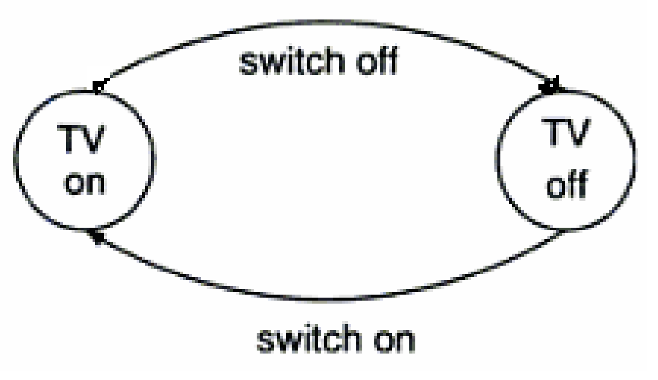
Ejemplo

- Estados de un proceso



Ejemplo

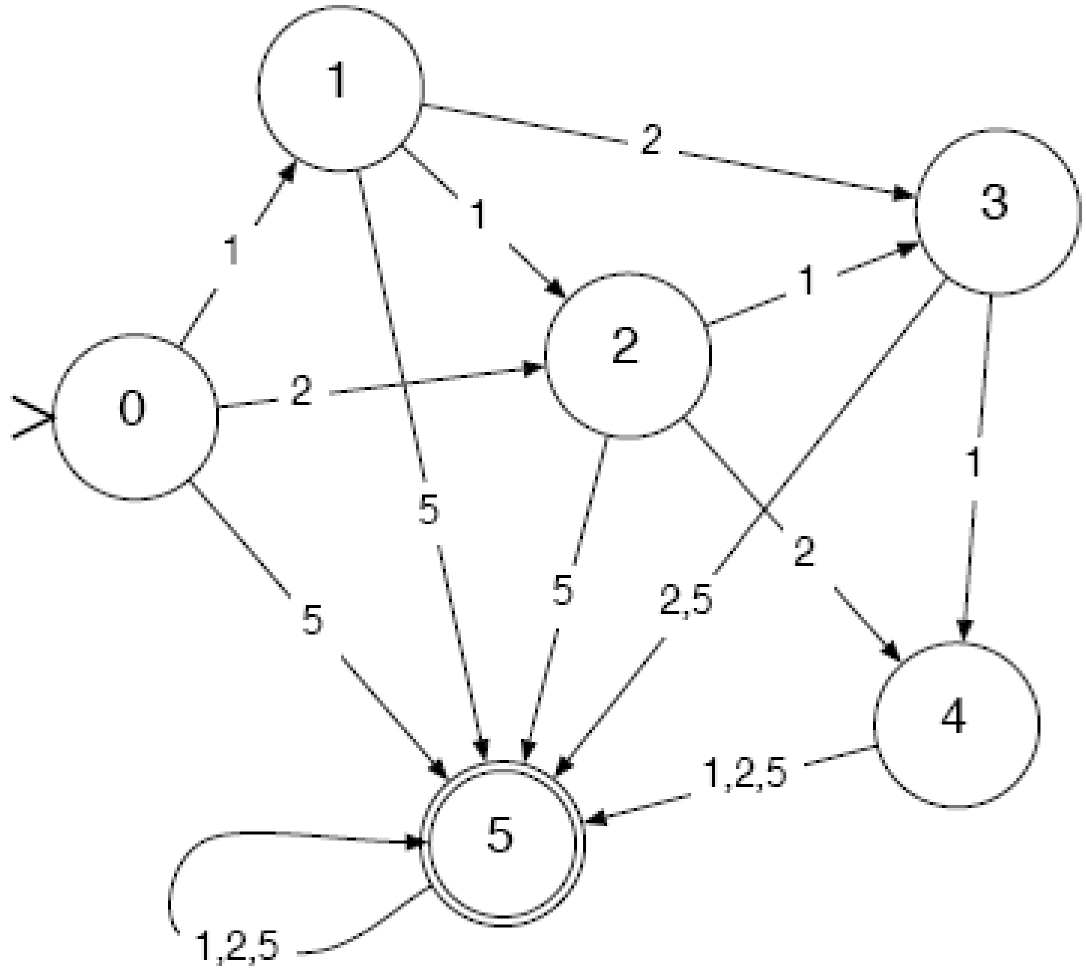
- Televisor



Ejercicio

- Máquina vendedora de bebidas:
  - Se quiere modelar el funcionamiento de una máquina automática vendedora de bebidas enlatadas. Dicha máquina acepta monedas de valor 1, 2 y 5, y el precio de cada lata es de 5. Vamos a considerar que el evento llamado “1” es la introducción de una moneda de valor 1 en la máquina, el evento “2” para la moneda de valor 2, etc.
  - Diseñar nuestro modelo:
    - Decidir cómo son los estados. Una idea sería que cada estado recordara lo que se lleva acumulado hasta el momento. El estado inicial, desde luego, recordaría que se lleva acumulado 0. El estado 4, lleva acumulado 4 pesos. El estado final o de aceptación (liberar bébiba) sería cuando la máquina obtiene 5 pesos.
    - Las siguientes secuencias son aceptables:
      - 1,1,1,1,1 = 5 pesos
      - 1,2,2 = 5 pesos
      - 1,2,2,5 >= 5 pesos
      - 5 = 5 pesos
      - 2,2,2 >= 5 pesos
      - 2,1,1 < 5 pesos, no se acepta (no libera bebida)

Solución



Ejercicio

- En la orilla de un río se encuentra un hombre, junto con un lobo, una oveja y paja. Hay un bote con la capacidad suficiente para llevar al hombre y a uno de los otros tres. El hombre con la paja, y demás compañeros deben cruzar el río, y el hombre puede llevar a uno sólo a la vez. Sin embargo, si el hombre deja solos al lobo y a la oveja en cualquier lado del río, con toda seguridad que el lobo se comerá a la oveja. Del mismo modo, si la oveja y la paja se quedan juntas, la oveja se comerá a la paja. ¿Es posible que se pueda cruzar el río sin que nadie sea comido?

- Diseño
  - Describiremos los estados mediante parejas de conjuntos separados por un guión, cada conjunto representa a quienes están en ésa orilla del río: Hombre (H), Oveja (O), Paja (P), Lobo (L).
  - El estado inicial del problema es la tupla
    - <H,L,O,P -- ø >   En la tupla en la parte derecha el conjunto vacío indica que ninguno de los pasajeros están en ese extremo del río.
    - Algunos estados son fatales y deben evitarse, por ejemplo: < L, O -- H, P>   "el lobo se come a la oveja"

- Las acciones que el hombre toma para atravesar el río son las transiciones de estados: puede cruzar solo en la barca (H) ó elegir como pasajero al lobo (L), a la oveja (O) ó a la paja (P). Por ejemplo,
  - < H, L, O, P – ø >   -> L ->   <O, P –- H, L > indica que el hombre tomó al lobo como acompañante para cruzar el río.
- La meta, el estado final ó estado de aceptación en el cual el problema está resuelto es cuando todos los pasajeros están en el otro extremo del río:
  - < ø – H, L, O, P >
- Para resolver el problema se tiene que elegir una secuencia de movimientos (atravesar el río) que a partir del estado inicial se alcance el estado final.

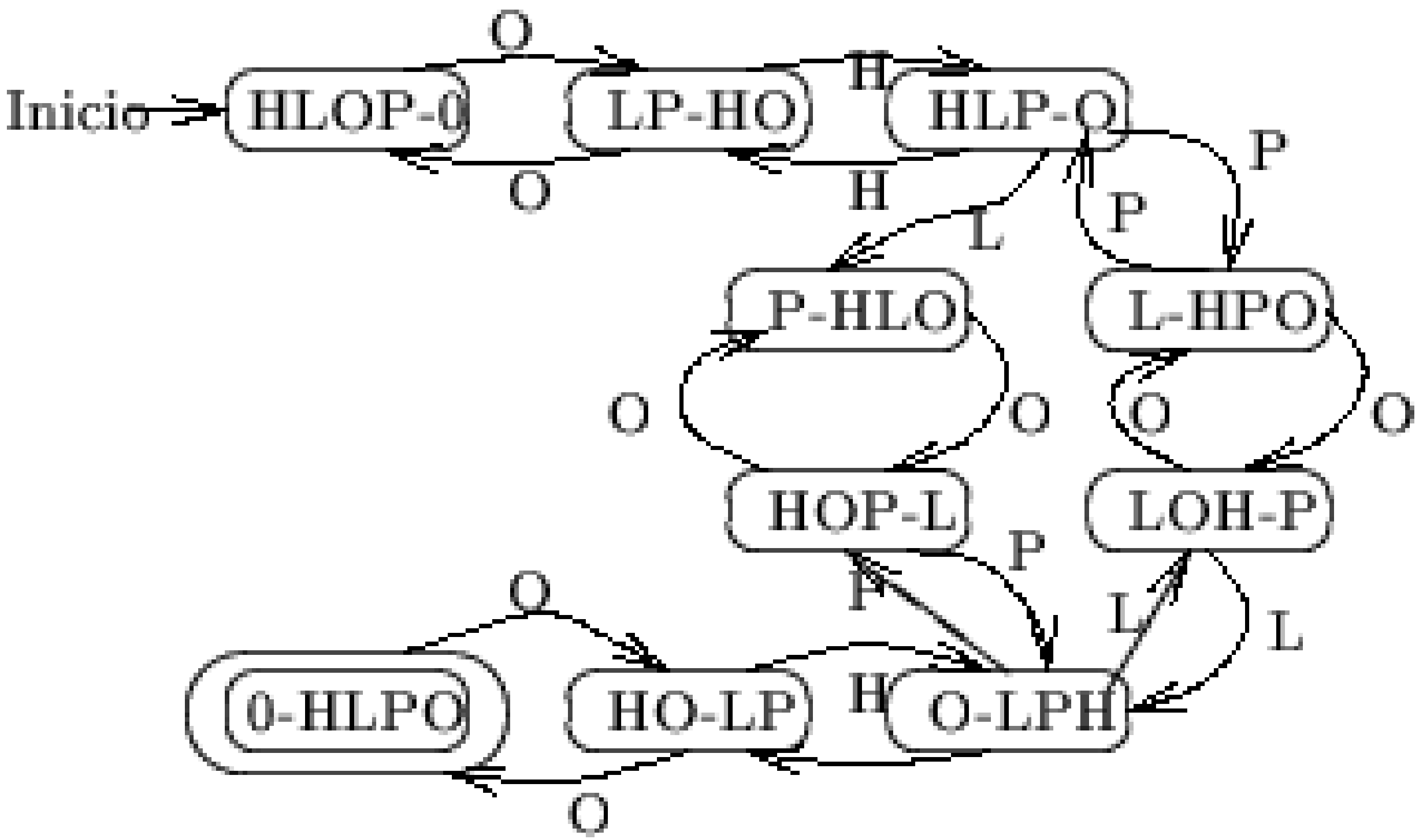
Secuencia solución al problema

$$< H, L, O, P - \emptyset > \xrightarrow{O} < L, P - H, O >$$
$$< L, P - H, O > \xrightarrow{H} < H, L, P - O >$$
$$< H, L, P - O > \xrightarrow{P} < L - O, H, P >$$
$$< L - O, H, P > \xrightarrow{O} < H, L, O - P >$$
$$< H, L, O - P > \xrightarrow{L} < O - H, L, P >$$
$$< O - H, L, P > \xrightarrow{H} < H, O - L, P >$$
$$< H, O - L, P > \xrightarrow{O} < \emptyset - H, L, P >$$

- Lo cual se indica:

$$< H, L, O, P - \emptyset > \xrightarrow{*} < \emptyset - H, L, O, P >$$

Diagrama de transiciones para el problema de cruzar el río

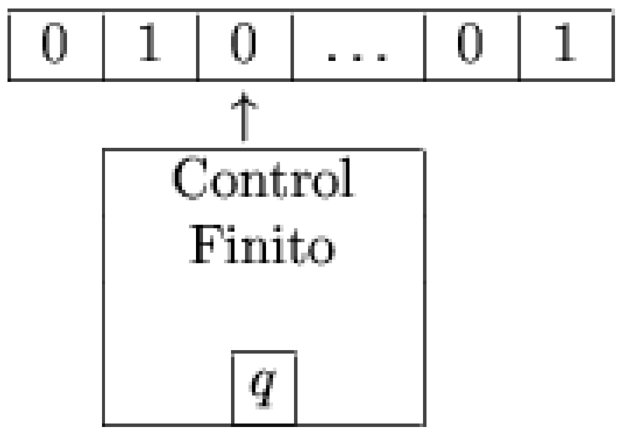


Definición AFD

- Un autómata finito determinístico M es una estructura representada por la tupla
  - $M = < Q, \sum, \delta, q0, F >$
- Q es un conjunto finito de estados: {q0, q1, q2, …, qn}
- ∑ es un conjunto finito de símbolos: alfabeto
- δ es la función de transición que indica el siguiente estado a partir del estado actual y leyendo un símbolo de entrada:
  - $\delta : Q \times \sum \dashrightarrow Q$
  - Para cada símbolo de entrada, existe a lo más una transición que sale del estado actual y lleva a otro estado (posiblemente el mismo).

- $q0$ es el estado inicial en el cual el AF inicia su ejecución.
- F es un subconjunto de Q, llamado conjunto de estados finales.

- Un autómata finito lo podemos ver como una caja negra de control, que va leyendo símbolos de una cadena escrita en una cinta, Existe una cabeza de lectura que en cada momento esá situada en una casilla de la cinta. Inicialmente, esta se sitúa en la casilla de más a la izquierda.
- El autómata en cada momento está en uno de los estado de Q. Inicialmente se encuentra en *q*0.
- En cada paso, el autómata lee un símbolo y según el estado en que se encuentre, cambia de estado y pasa a leer el siguiente símbolo. Así sucesivamente hasta que termine de leer todos los
- símbolos de la cadena. Si en ese momento la máquina está en un estado final, se dice que el autómata acepta la cadena. Si no está en un estado final, la rechaza.



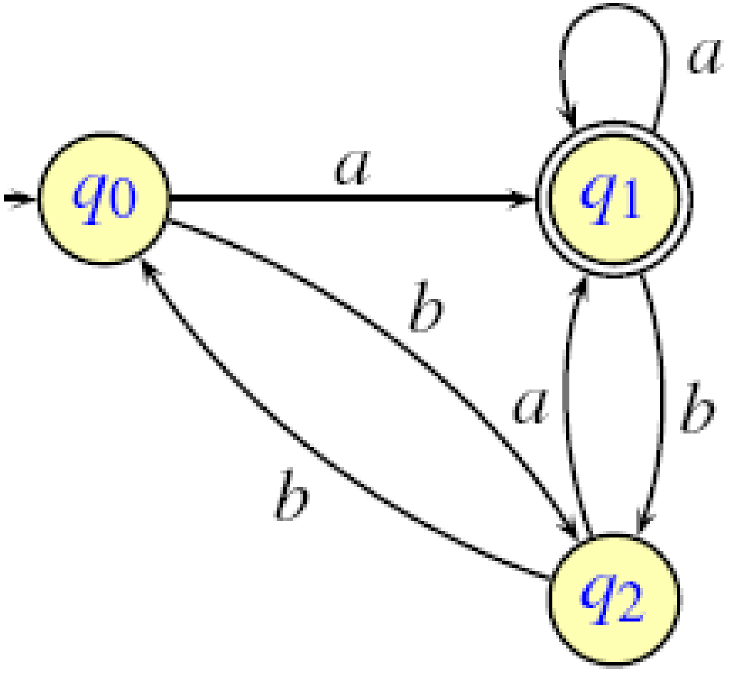
- El diagrama de transición de un Autómata de Estado Finito es un grafo en el que los vértices representan los distintos estados y los arcos las transiciones entre los estados.
- Cada arco va etiquetado con el símbolo que corresponde a dicha transición.
- El estado inicial y los finales vienen señalados de forma especial (por ejemplo, con un flecha el estado inicial y con un doble círculo los finales).

Ejemplo

- *Supongamos el autómata* $M = (Q, A, q0, \delta, F)$ *donde*
- $Q = \{q0, q1, q2\}$
- $A = \{a, b\}$
- *La función de transición* δ *está definida por las siguientes igualdades:*
  - $\delta(q0, a) = q1$ $\delta(q0, b) = q2$
  - $\delta(q1, a) = q1$ $\delta(q1, b) = q2$
  - $\delta(q2, a) = q1$ $\delta(q2, b) = q0$
- $F = \{q1\}$

Ejemplo

- Diagrama de transición



Cómputo

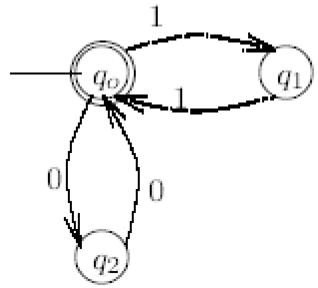
- Un cómputo es la descripción de cambios de estado que sufre un autómata ante una cadena de entrada dada.
- Un cómputo siempre termina puesto que la cadena es finita y siempre es posible saber si está o no el autómata en un estado de aceptación.

Grafo de transición

- Dado un AF, su grafo dirigido de transiciones se construye de la siguiente forma:
  - Los vértices del grafo corresponden a los estados del AF.
  - Si existe una transición del estado q al estado p ante la entrada a entonces existe un arco del nodo q al nodo p etiquetado con el símbolo a.
  - En el grafo, el estado inicial q0 se indica con una flecha (arco) que llega, sin ninguna etiqueta.
  - Los estados finales de aceptación se indican doble circulo.

Ejemplo

- Sea M1 = <Q, ∑, δ, q0, {q0}> un AFD donde Q={q0, q1, q2}, ∑={0,1} y δ definida en la tabla.



δ	0	1
F:$\rightarrow q_o$	q_2	q_1
q_1		q_o
q_2	q_o	

- Sea la cadena de entrada w="1100", mostramos a continuación la secuencia de transiciones (cómputo) que realiza M1:

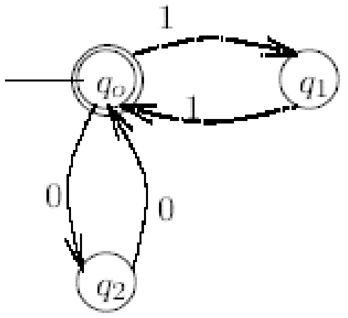
$$\rightarrow q_o \xrightarrow{1} q_1 \xrightarrow{1} q_o \xrightarrow{0} q_2 \xrightarrow{0} q_o$$

- Lo cual también se describe con la tabla:

1	1	0	0	
q_o	q_1	q_o	q_2	q_o

Ejemplo

- Sea u= “11100”, entonces M1 rechaza a u.



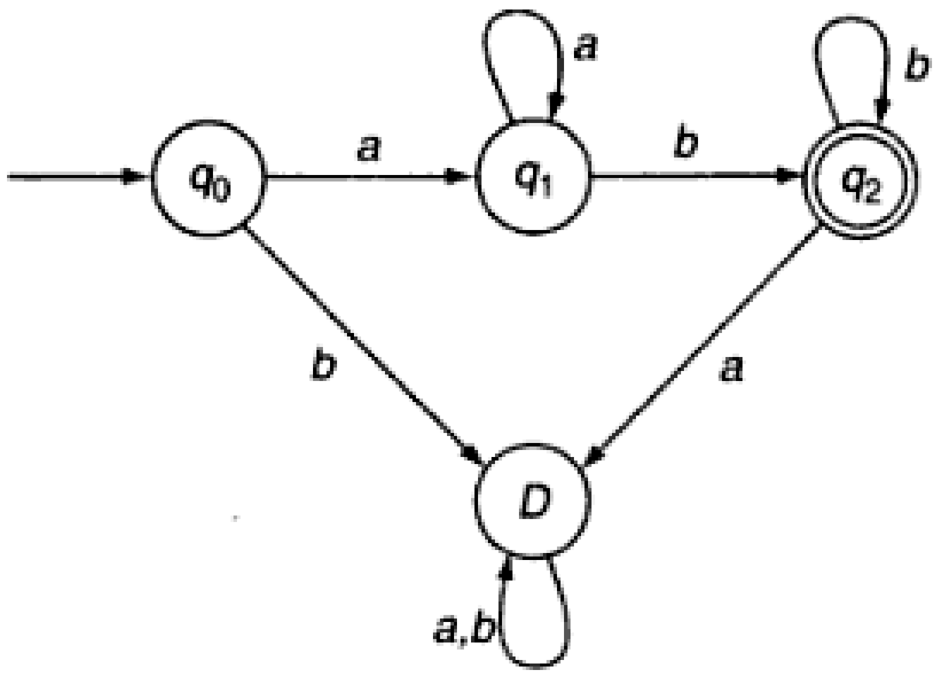
$$\rightarrow q_o \xrightarrow{1} q_1 \xrightarrow{1} q_o \xrightarrow{1} q_1 \xrightarrow{0} indefinido$$

1	1	1	0	0
q_o	q_1	q_o	q_1	$indef$

- Un AFD acepta una cadena de entrada w si y sólo si la secuencia de transiciones (ruta) que corresponden a cada uno de los símbolos de w (leídos en orden de izquierda a derecha) llevan del estado inicial a un estado final.

Ejemplo

$$\{a^n b^m \mid n, m \geq 1\}$$



Función extendida $\hat{\delta}$

- Para un AF M definimos la función extendida δ ^: Q X Σ^* -> Q a partir de la función δ :Q X Σ -> Q, de tal forma que δ ^(q,w) representa el único estado p tal que existe una ruta en el grafo de transiciones de M que parte de q y llega a p etiquetada por w.
- Definimos a δ ^ inductivamente

1. δ ^(q, ε) = q, (caso base)
2. δ ^(q, wa) = δ (δ ^(q,w),a) para toda cadena w en Σ^*, y símbolo a en Σ.

- El caso base indica que el AF no cambia de estado a menos que lea un símbolo de entrada, mientras que el paso inductivo indica cómo se debe calcular el siguiente estado ante una secuencia no vacía de símbolos de entrada: primero se encuentra el estado p=δ^(q,w) después de leer w y entonces se evalúa δ^(p,a). Note que δ y δ^ coinciden en cadenas de longitud 1:

$$
\begin{aligned}
\hat{\delta}(q,a) &= \hat{\delta}(q,\epsilon a) && \text{dado que ``}a\text{'' } = \epsilon a \\
&= \delta\big(\hat{\delta}(q,\epsilon)\big), a) && \text{paso inductivo de la def.} \\
&= \delta(q,a).
\end{aligned}
$$

Lenguaje regular

- Formalmente, una cadena w es aceptada por un AFD M=<Q,∑, δ, q0,F > si y sólo si

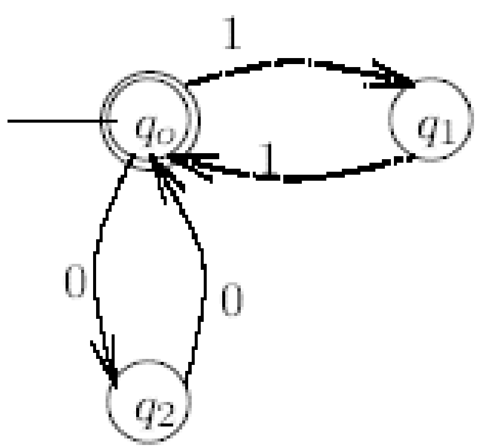
$$\hat{\delta}(q_0, w) \in F$$

- El lenguaje aceptado por M lo denotamos por L(M) y es el conjunto de todas las cadenas aceptadas por M

$$L(M) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$$

Ejemplo

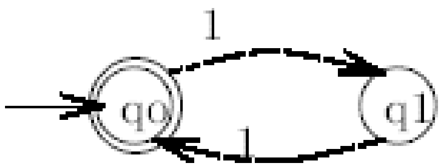
- Secuencia de evaluaciones δ^ para la cadena u=11100



$$
\begin{aligned}
\hat{\delta}(q_o, u) &= \delta(\hat{\delta}(q_o, 1110), 0) \\
&= \delta(\delta(\hat{\delta}(q_o, 111), 0), 0) \\
&= \delta(\delta(\delta(\hat{\delta}(q_o, 11), 1), 0), 0) \\
&= \delta(\delta(\delta(\delta(\hat{\delta}(q_o, 1), 1), 0), 0) \\
&= \delta(\delta(\delta(\delta(\delta(\hat{\delta}(q_o, \epsilon), 1), 1), 1), 0), 0) \\
&= \delta(\delta(\delta(\delta(\delta(q_o, 1), 1), 1), 0), 0) \\
&= \delta(\delta(\delta(\delta(q_1, 1), 1), 0), 0) \\
&= \delta(\delta(\delta(q_o, 1), 0), 0) \\
&= \delta(\underbrace{\delta(q_1, 0)}_{indefinida}, 0)
\end{aligned}
$$

Ejemplo

- ¿Cuál es el lenguaje que acepta el AF M1’?
- Primero identifica la propiedad que cumplen las cadenas aceptadas por M1’



δ_1	0	1
F:$\rightarrow q_o$	q_2	q_1
q_1		q_o

- Si ordenamos las cadenas aceptadas por M1’ tenemos:

$L(M_1')$	$2n$	n
ϵ	0	0
11	2	1
1111	4	2
$\vdots$	$\vdots$	$\vdots$

Demostración

- Formalmente, demostraremos que M1' acepta cadenas formadas por secuencias pares de 1's (cadenas de longitud par de 1's). Aplicamos inducción sobre la longitud de las cadenas que acepta M1'.

1. Caso base:

$$|w| = 0 \ (\ w = \epsilon \ )$$

$$\epsilon \text{ en } L(M_1')$$

$$\hat{\delta}(qo, \epsilon) = qo \in \{q_0\} = F$$

2. Paso inductivo

- Paso Inductivo
  - H.I.: Suponer para cualquier m, la cadena w cumple |w|=2m (i.e., es de longitud par) está formada por ‘1’s y pertenece a L(M1’).
  - Por demostrar que se cumple para cadenas de longitud sucesor(m) aceptadas por M1’.
    - Sea v=wx, tal que |v|=2(m+1) lo cual significa que x=11 puesto que es la única forma de llegar a un estado de aceptación.
    - Claramente $|v| = |w| + |x| = 2m + 2 = 2(m+1)$

Ejercicios

- Para cada uno de los siguientes AFD indique quienes son cada uno de sus componentes, encuentre el lenguaje que acepta y muestre si las cadenas w y v son aceptadas o rechazadas por el AFD

(a) $w = 01101$ y $v = 1110101$

δ	0	1
$\rightarrow q_o$	q_2	q_1
q_1	q_3	q_o
F:q_2	q_0	q_3
q_3	q_1	q_2

(b) $w = 001100$ y $v = 1010101$

δ	0	1
F:$\rightarrow q_o$	q_2	q_1
q_1	q_o	
q_2	q_o	

Ejercicio

- Considera el conjunto

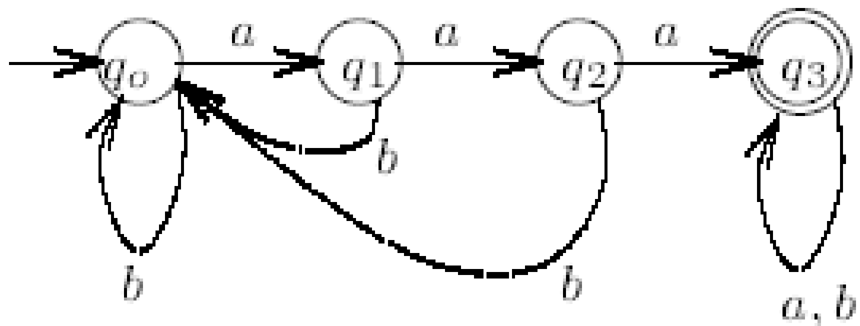
  L={waaav | w, v en {a,b}* }
- Construye su AFD

Cadenas:

- La cadena abaaabab debe ser aceptada
- La cadena babbabba debe ser rechazada

Solución

L={waaav | w, v en {a,b}* }



δ	a	b
$\rightarrow q_o$	q_1	q_0
q_1	q_2	q_0
q_2	q_3	q_0
F:q_3	q_3	q_3

Ejercicios

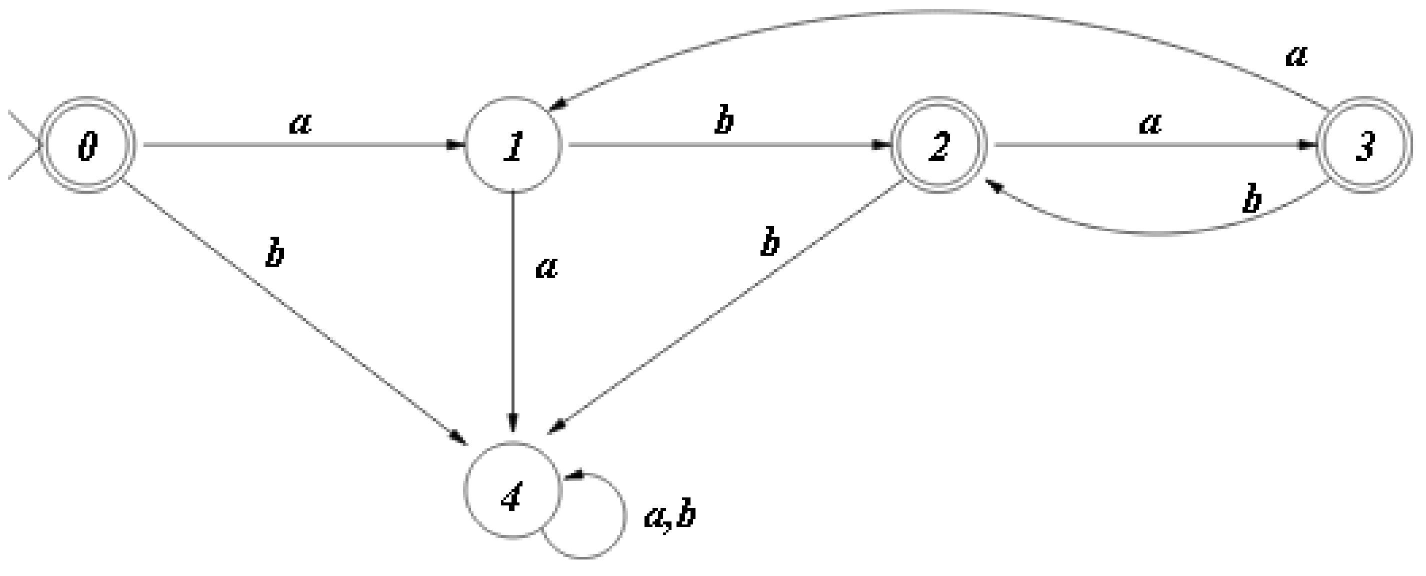
1. Obtenga el AFD que acepte los siguientes lenguajes sobre el alfabeto {0,1}
   1. El conjunto de todas las cadenas que terminen en “00”
   2. El conjunto de todas las cadenas con tres 0’s consecutivos.

Ejercicio

- Obtenga el AF que representa a la expresión (ab | aba)*

Ejercicio, solución

- Obtenga el AF que representa a la expresión (ab | aba)*



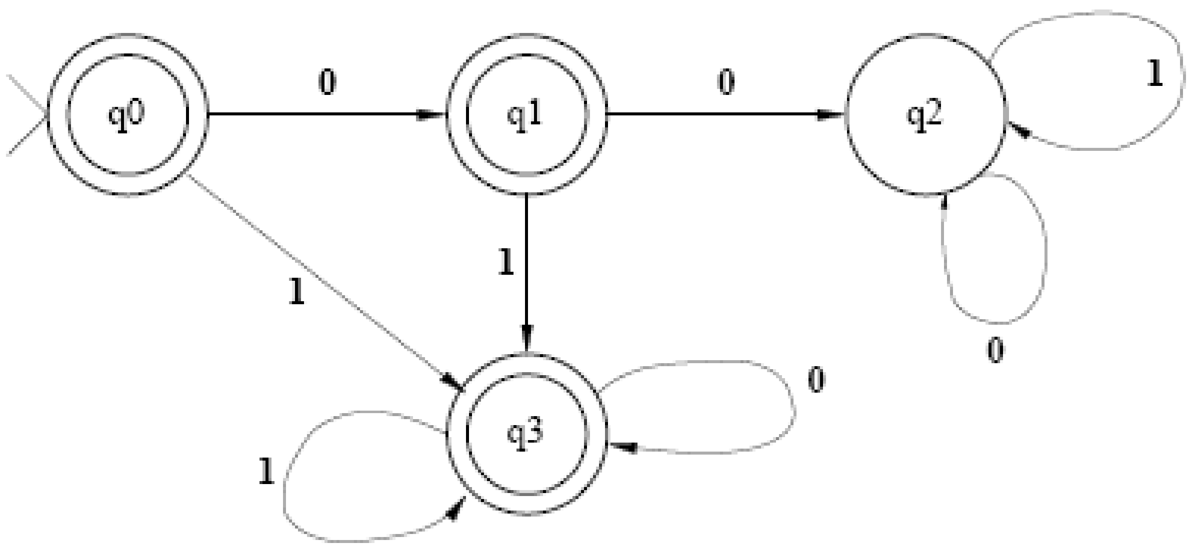
Ejercicio

- Diseñar un AFD que acepte exactamente el lenguaje en el alfabeto {0, 1} en el que las palabras no comienzan con 00.
- Condiciones:

Estado	Condición
q0	No se han recibido caracteres
q1	Se ha recibido un cero al inicio
q2	Se han recibido dos ceros iniciales
q3	Se recibió algo que no son dos ceros iniciales

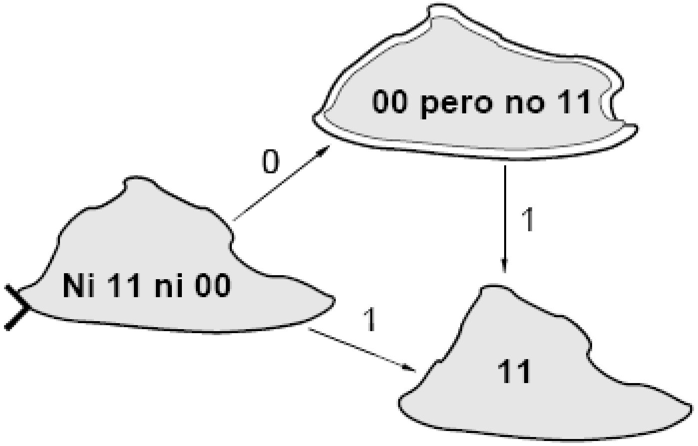
Solución

- Diseñar un AFD que acepte exactamente el lenguaje en el alfabeto {0, 1} en el que las palabras no comienzan con 00.



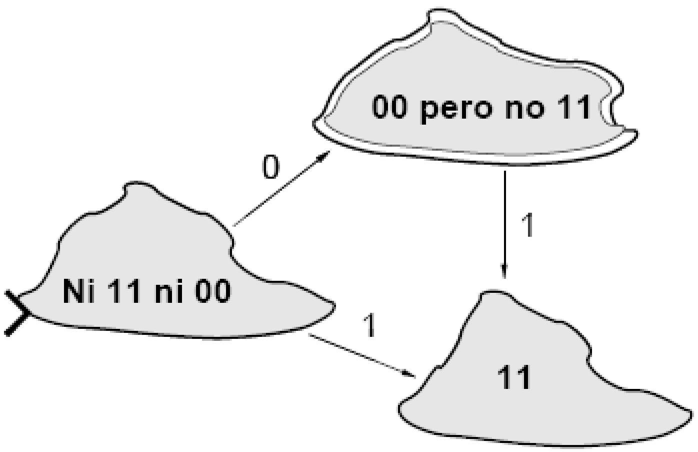
Ejercicio

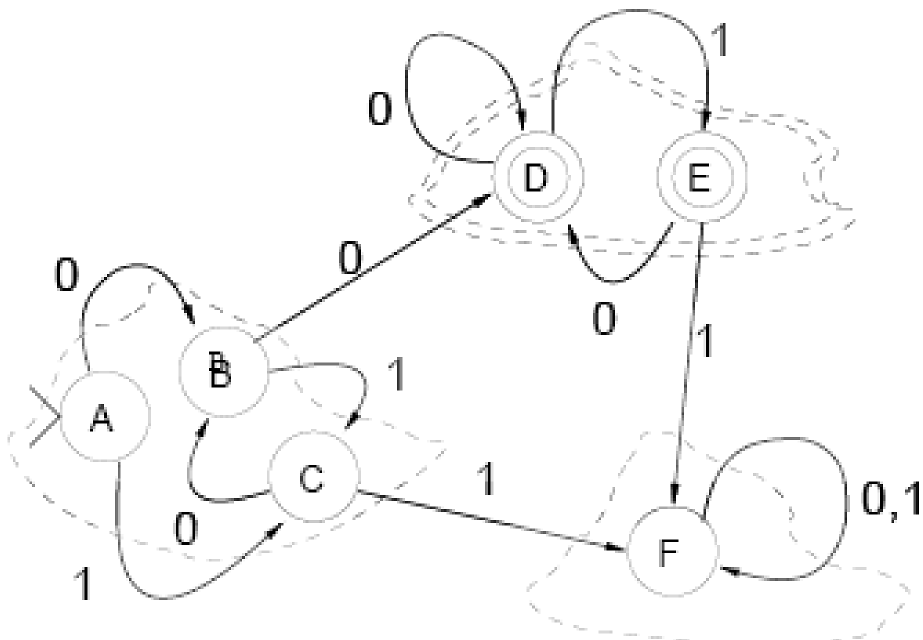
- Diseñar un AFD que acepte las palabras del lenguaje en {0, 1} donde las palabras no contienen la subcadena 11 pero sí 00.
- Situaciones:
  - Las letras consumidas hasta el momento no contienen ni 00 ni 11.
  - Contienen 00 pero no 11
  - Contienen 11.



Solución

- Diseñar un AFD que acepte las palabras del lenguaje en {0, 1} donde las palabras no contienen la subcadena 11 pero sí 00.



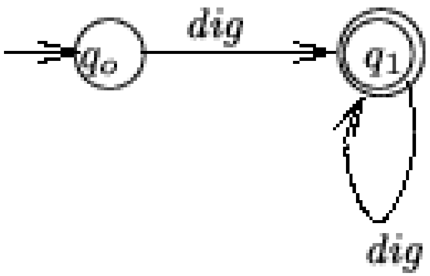


Ejercicio

- Construye un AF que reconozca los números enteros positivos.

Solución

- Construye un AF que reconozca los números enteros positivos.



δ	dig
$\rightarrow q_o$	q_1
$q_f : q_1$	q_1

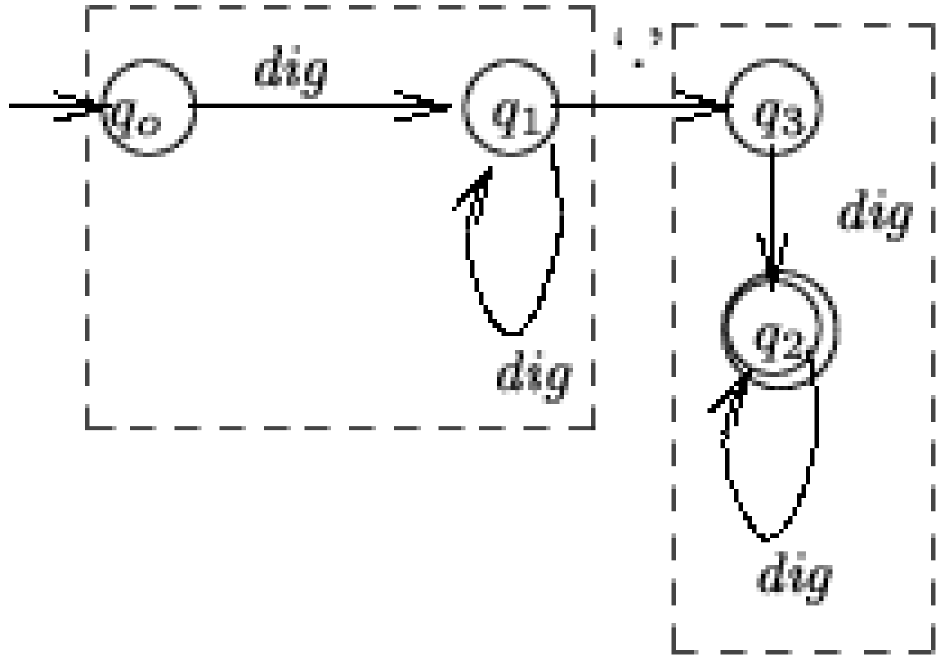
Composición de AFD

- Para la construcción de un nuevo AFD podemos utilizar (como si fueran módulos) otros AFDs.
- Debemos asegurar que la composición entre estos módulos produzca un AFD, es decir, a partir de lenguajes regulares se obtiene otro lenguaje regular.

Ejercicio

- Construir un AFD que acepte el lenguaje de las cadenas que representan números reales de acuerdo a la siguiente descripción: Un número real tiene una parte entera, a continuación un '.' decimal y una parte decimal.
- Ejemplos: "097.5432", "134.566"

Solución



- En el ejemplo anterior se observa claramente la presencia de la operación de concatenación entre AFD.
- El lenguaje M se obtiene concatenando los lenguajes que aceptan respectivamente cado uno de los AFDs vistos como submódulos.

Propiedades

- Si dos lenguajes A y B son regulares, entonces
  - $A \cap B$
  - $A \cup B$
  - A^c
- son regulares.

Construcción del producto

- Sean A, B regulares y sean M_1 y M_2 AFDs tales que $L(M_1)=A$ y $L(M_2)=B$, donde
  - $M_1 = < Q_1, \sum, \delta_1, q_{01}, F_1>$,
  - $M_2 = < Q_2, \sum, \delta_2, q_{02}, F_2>$,
- Para demostrar que $A \cap B$ es regular construimos un AFD M_3 tal que $L(M_3)= A \cap B$.
- Ante una cadena de entrada w, M_3 simula el comportamiento de M_1 y M_2 simultáneamente: M_3 acepta a w si y sólo si ambos aceptan w:
  - $M_3 = < Q_3, \sum, \delta_3, q_{03}, F_3>$,

- $M_3 = < Q_3, \Sigma, \delta_3, q_{03}, F_3>$,
- Donde:

$$Q_3 = Q_1 \times Q_2 = \{(p,q) \mid p \in Q_1 \ y \ q \in Q_2\}$$

$$F_3 = F_1 \times F_2 = \{(p,q) \mid p \in F_1 \ y \ q \in F_2\}$$

$$q_{03} = (q_{01}, q_{02})$$

- La función de transición δ_3 se define como:
  - $\delta_3((p,q), a) = (\delta_1(p,a), \delta_2(q,a))$

- Y su extensión inductiva para cadenas es

$$\hat{\delta}_3((p,q),\varepsilon) = (p,q)$$

$$\hat{\delta}_3((p,q),xa) = \delta_3(\hat{\delta}_3((p,q),x),a)$$

- Lema: Para toda $w \in \Sigma^*$

$$\hat{\delta}_3((p,q),w) = (\hat{\delta}_1(p,w),\hat{\delta}_2(q,w))$$

- Demostramos a continuación que:

$$L(M_3) = L(M_1) \cap L(M_2)$$

Para toda $x \in \Sigma^*$:

$$\begin{aligned}
x \quad & \in L(M_3) \\
& \Longleftrightarrow \hat{\hat{\delta}}_3(q_{o3}, x) \in F \\
& \Longleftrightarrow \hat{\hat{\delta}}_3\big((q_{o1}, q_{o2}), x\big) \in F_1 \times F_2 \\
& \Longleftrightarrow \big(\hat{\hat{\delta}}_1(q_{o1}, x), \hat{\hat{\delta}}_2(q_{o2}, x)\big) \in F_1 \times F_2 \\
& \Longleftrightarrow \hat{\hat{\delta}}_1(q_{o1}, x) \in F_1 \text{ y } \hat{\hat{\delta}}_2(q_{o2}, x) \in F_2 \\
& \Longleftrightarrow x \in L(M_1) \text{ y } x \in L(M_2) \\
& \Longleftrightarrow x \in L(M_1) \cap L(M_2)
\end{aligned}$$

Construcción del complemento

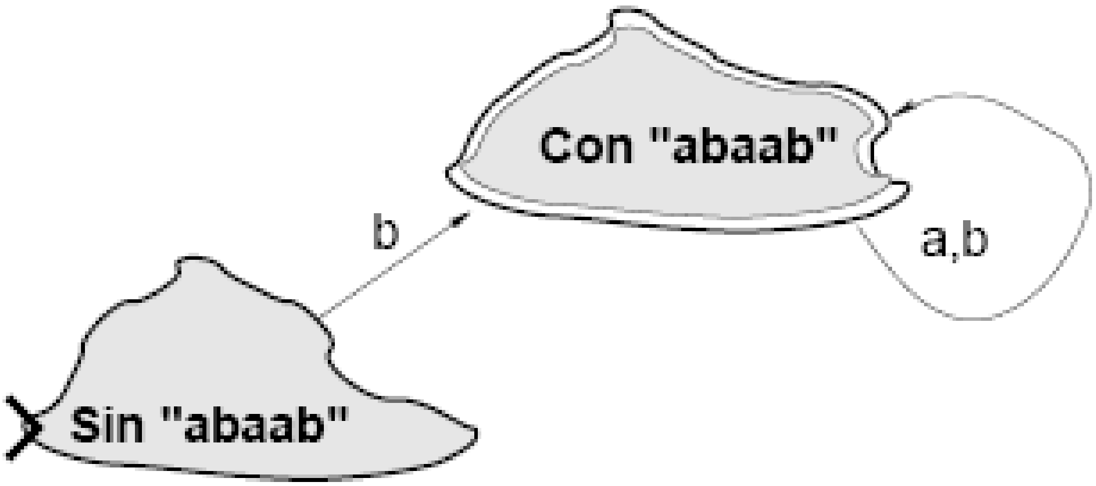
- Para demostrar que A^c es regular si A lo es, sea M un AFD que acepta A y tomamos a M' simplemente intercambiando estados de aceptación por estados de rechazo

$$\hat{\delta}'(q_0, x) \in F' \iff \hat{\delta}(q_0, x) \notin F$$

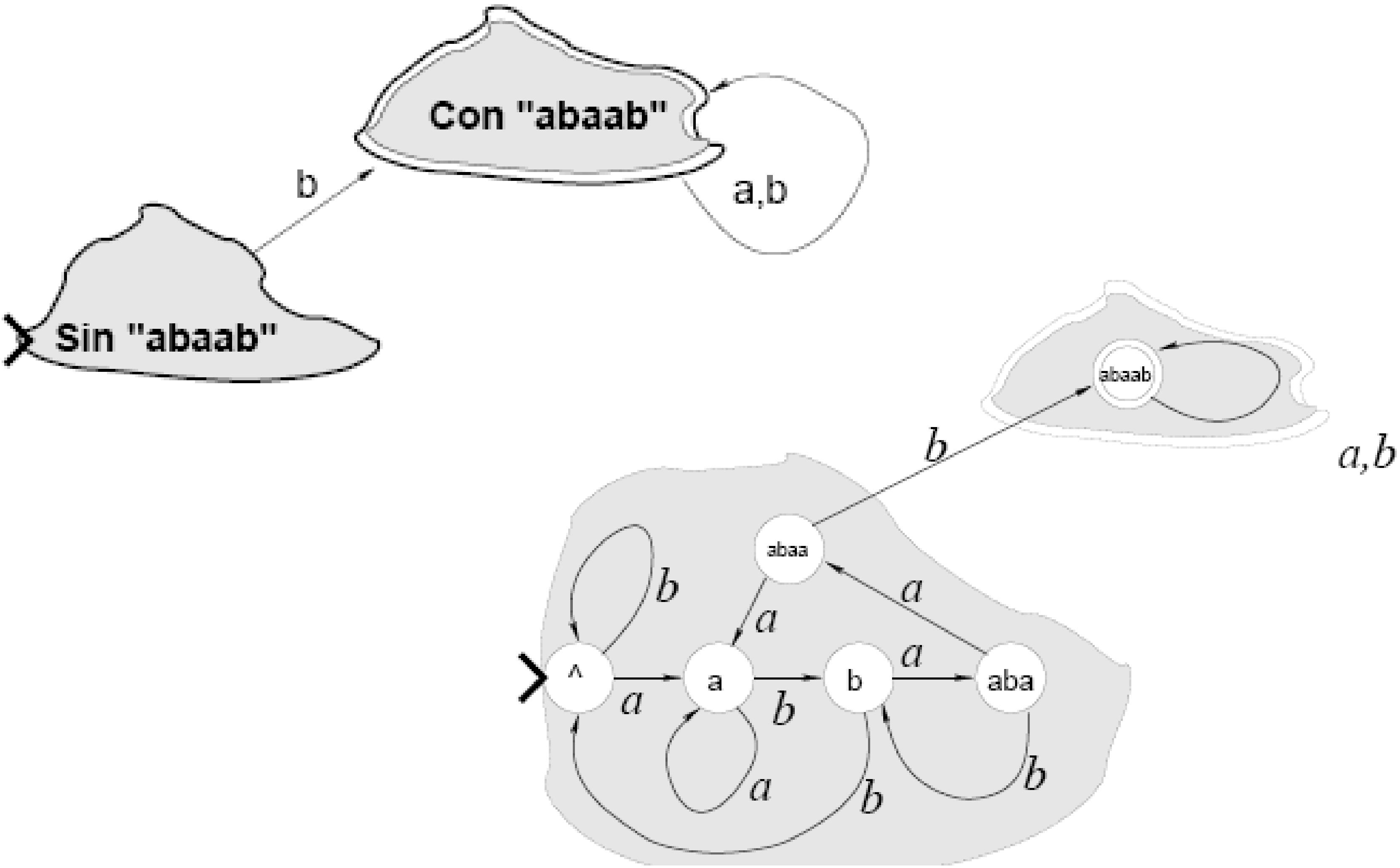
- M' aceptará exactamente lo que M rechaza.

Ejercicio, diseño de un AF por complemento

- Obtener un AF para el lenguaje en {a, b} de las palabras que no contienen la cadena “abaab”.

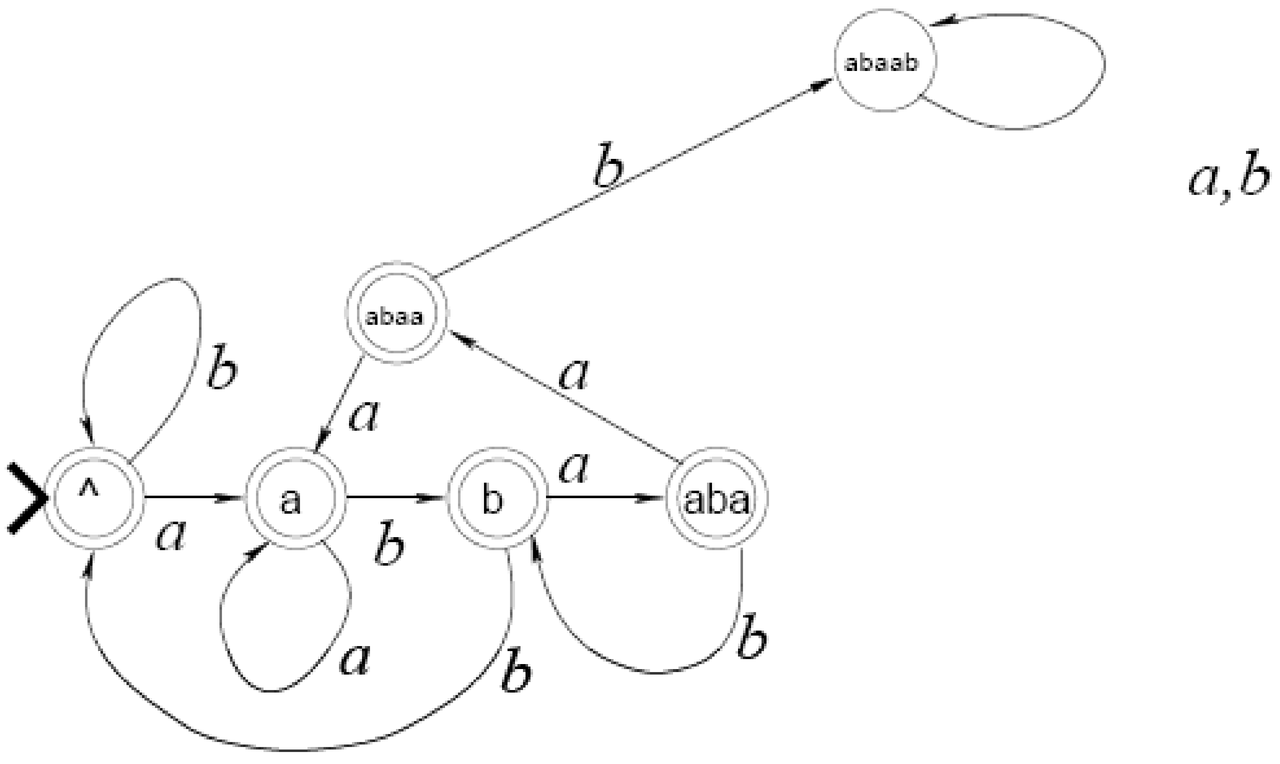


Solución



Ejercicio

- Obtener un AF para el lenguaje en {a, b} de las palabras que no contienen la cadena "abaab".



Construcción de la unión

- Aplicando las leyes de De Morgan que Au B es regular, si A y B lo son:

$$A \cup B = \,\tilde{}\,(\,\tilde{}\,A \cap \,\tilde{}\,B)$$

- Lo cual ofrece un AFD que se parece al AFD del producto pero los estados de aceptación son

$$\begin{aligned} F_3 &= \{(p,q) \mid p \in F_1 \text{ ó } q \in F_2\} \\ &= (F_1 \times Q_2) \cup (Q_1 \times F_2) \end{aligned}$$

- Intuitivamente, la unión de dos lenguajes nos permite describir el AFD M_3 correspondiente mediante una alternativa: en un estado dado, el AFD elige seguir una transición y finalmente una ruta que lleva a un estado de aceptación de M_1 ó de M_2.

- ¿Qué ocurre si tal elección debe tomarse leyendo en ambos casos el mismo símbolo de entrada actual? Esto significaría que ante el mismo símbolo existe una transición que lleva a estados distintos, lo cual no está definido en un AFD.

- Esto conduce al **no determinismo**.

Ejercicios

- Obtenga el AFD que acepte los siguientes lenguajes sobre el alfabeto ∑={0,1}:
  - El conjunto de todas las cadenas tales que el décimo símbolo (contado de izquierda a derecha) sea un ’1’.
  - El conjunto de todas las cadenas tales que todo bloque de cinco símbolos consecutivos contenga al menos dos ‘0’s.
  - El conjunto de todas las cadenas con número impar de ‘0’s y número par de ‘1’s.