Instituto Politécnico Nacional

Escuela Superior de Cómputo

Compiladores

Profr. Edgardo Adrián Franco Martínez

09 Análisis léxico V

Autómatas finitos & Mapas conceptuales 01 y 02

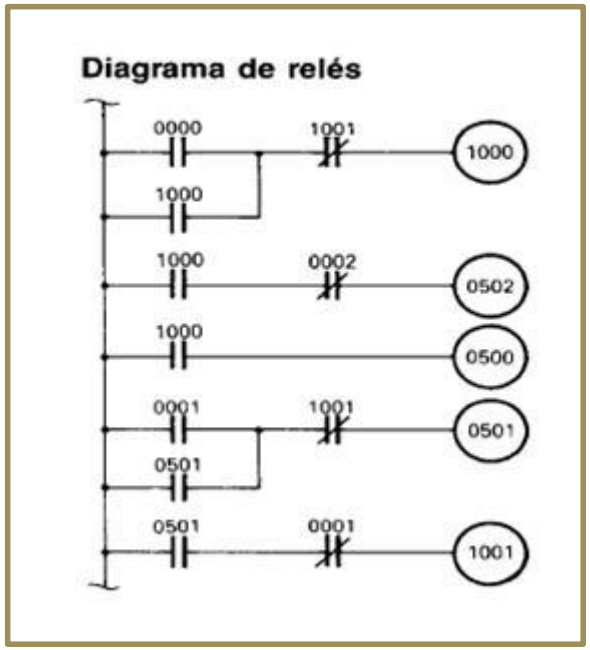
efranco.docencia@gmail.com
http://computacion.cs.cinvestav.mx/~efranco

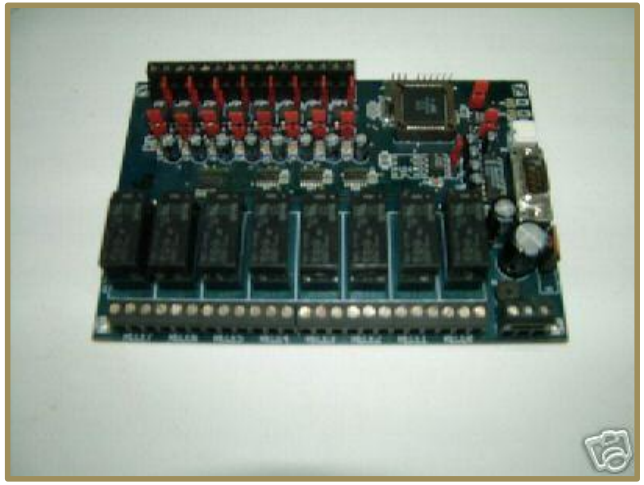
Contenido

- Autómata
- Definición formal de autómata
- Representación de un autómata
- Mediante tablas de transiciones
- Mediante diagramas de estados
- Autómata finito
- Definición formal de autómata finito
- Lenguaje reconocido por un autómata finito
- Ejemplos
- Mapas conceptuales 01 y 02

Autómatas

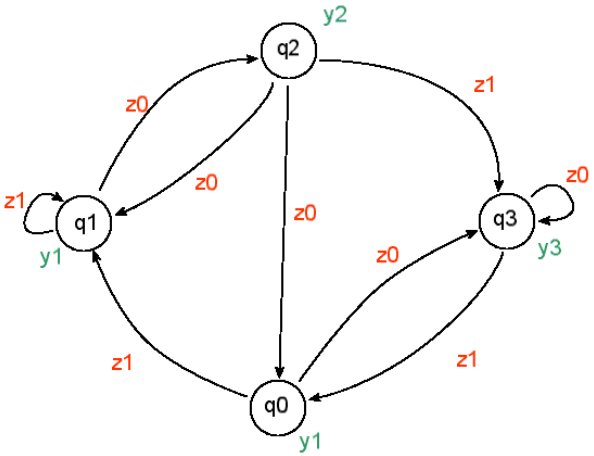
- En 1938 el matemático norteamericano Claude Elwood Shannon (1916-2001), estableció las bases de la aplicación de la lógica matemática para los circuitos combinatorios y secuenciales, construidos a bases de relés y más adelante con otros dispositivos de vacío y estado solido.

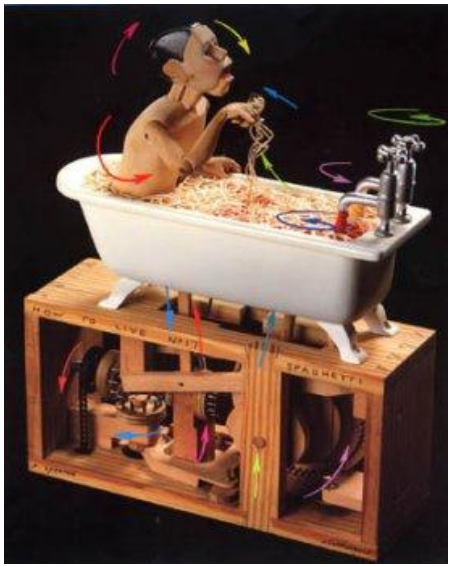




Autómatas

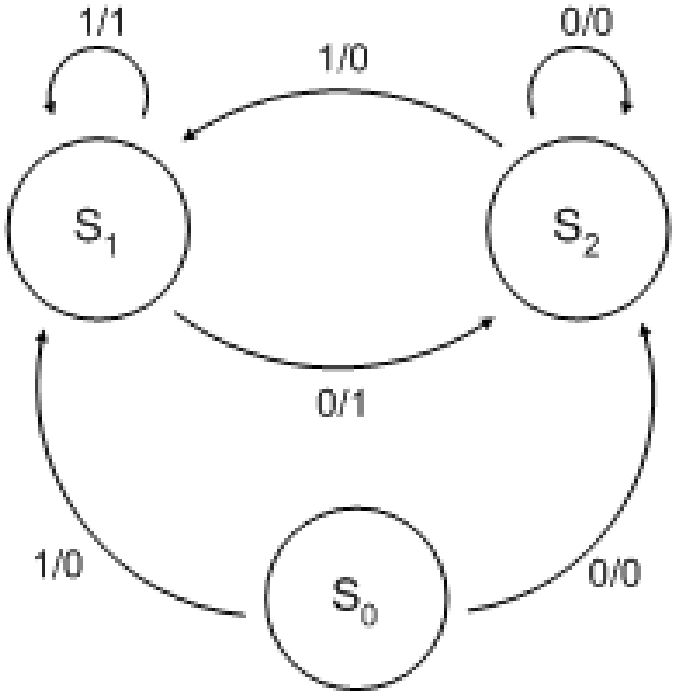
- Los autómatas son sistemas capaces de transmitir información. En sentido amplio, todo sistema que acepta señales de su entorno y, como resultado, cambia de estado y transmite otras señales al medio, puede considerarse un autómata.





Autómatas

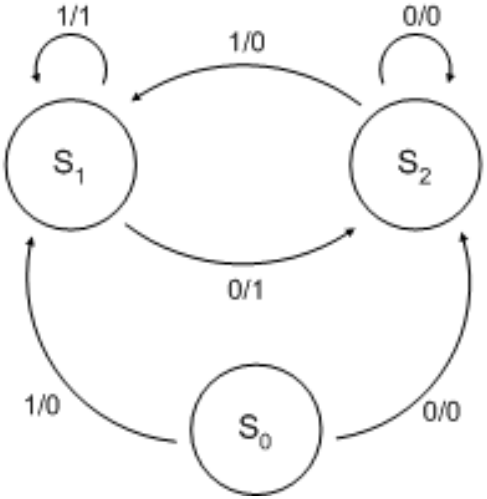
- Si la información se codifica en cadenas de símbolos, podemos con un autómata manipular cadenas de símbolos que se le presentan a su entrada, produciendo otras tiras o cadenas de símbolos a su salida.



Autómatas

- El autómata recibe los símbolos de entrada, uno detrás de otro, es decir secuencialmente.

- El símbolo de salida que en un instante determinado produce un autómata, no sólo depende del último símbolo recibido a la entrada, sino de toda la secuencia o cadena, que ha recibido hasta ese instante.



Autómata

- **El estado de un autómata** es toda la información necesaria en un momento dado, para poder deducir, dado un símbolo de entrada en ese momento, cual será el símbolo de salida, i.e. conocer el estado de un autómata, es lo mismo que conocer toda la **historia de símbolos de entrada**, así como el estado inicial, estado en que se encontraba el autómata al recibir el primero de los símbolos de entrada.

- El autómata tendrá un determinado **número de estados** (pudiendo ser infinitos), y se encontrará en uno u otro según sea la historia de símbolos que le han llegado.
  - Se define **configuración de un autómata** a su situación en un instante.
  - Se define **movimiento de un autómata** como el transito entre dos configuraciones.

Autómata

- Si un autómata se encuentra en un estado determinado, recibe un símbolo también determinado, producirá un símbolo de salida y efectuará un cambio o ***transición a otro estado*** (también puede quedarse en el mismo estado).

- Los elementos del lenguaje son sentencias, palabras, etc... i.e. **cadenas** formadas a partir de un alfabeto, que no es otra cosa que un conjunto finito de símbolos. Una cadena de símbolos pertenecerá al correspondiente lenguaje si tal cadena se ha formado obedeciendo esas reglas. Entonces un **autómata reconocedor de ese lenguaje, funciona de tal forma que** cuando reciba a su entrada una determinada cadena de símbolos <u>indica si dicha cadena pertenece o no al lenguaje</u>.

Definición formal de autómata

- Un autómata es una **quíntupla A = (E, S, Q, f, g)** donde :
  - **E = {conjunto de entradas o vocabulario de entrada}**
    - E es un conjunto finito, y sus elementos se llaman entradas o símbolos de entrada.
  - **S = {conjunto de salidas o vocabulario de salida}**
    - S es un conjunto finito, y sus elementos se llaman salidas o símbolos de salida.
  - **Q = {conjunto de estados}**
    - Q es el conjunto de estados posibles, puede ser finito o infinito
  - **f es la función de transición o función del estado siguiente, y para un par del conjunto *E × Q devuelve un estado perteneciente al conjunto Q, y es el producto cartesiano de E por Q.***
  - **g es la función de salida, y para un par del conjunto E × Q, devuelve un símbolo de salida del conjunto S.**

$$f : E \times Q \rightarrow Q$$

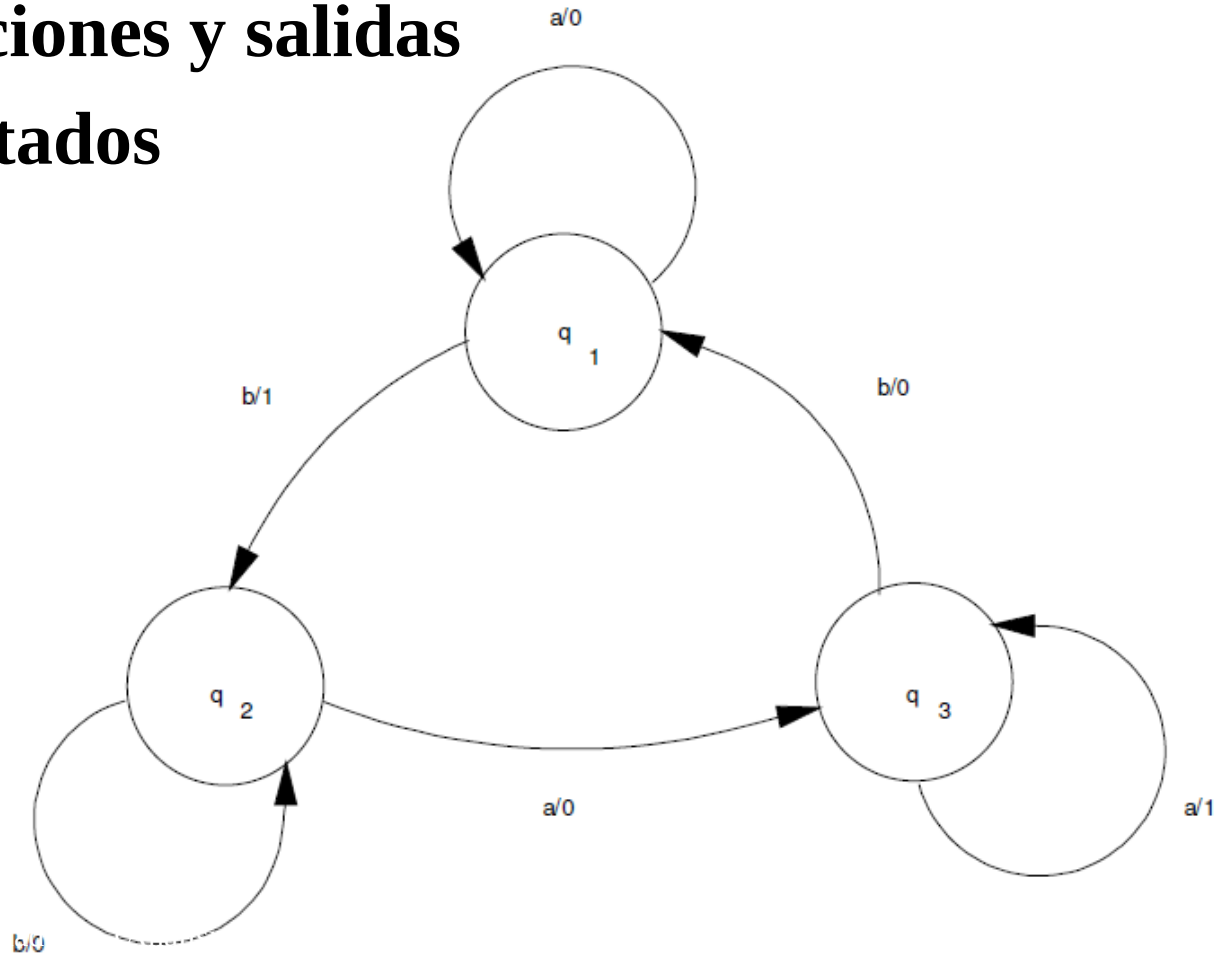
$$g : E \times Q \rightarrow S$$

Representación de un autómata

- Los autómatas se pueden representar mediante :
  - **Tablas de transiciones y salidas**
  - **Diagramas de estados**

f	*a*	*b*
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_3	q_1

g	*a*	*b*
q_1	*0*	*1*
q_2	*0*	*0*
q_3	*1*	*0*



Representación de un autómata

(Mediante tablas de transiciones y salidas)

- Las funciones f y g pueden representarse mediante una tabla, con tantas filas como estados y tantas columnas como entradas. Así por ejemplo se puede representar el autómata A = (E, S, Q, f, g) donde E = {*a,b*}, *S = {0,1}, Q = {q1, q2, q3} y las funciones f y g se* pueden representar por :

f	*a*	*b*
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_3	q_1

g	*a*	*b*
q_1	*0*	*1*
q_2	*0*	*0*
q_3	*1*	*0*

Así se tiene que $f(a, q_1) = q_1$; $g(a, q_1) = 0$; o también $f(a, q_2) = q_3$; y $g(a, q_3) = 1$.

Representación de un autómata

(Mediante tablas de transiciones)

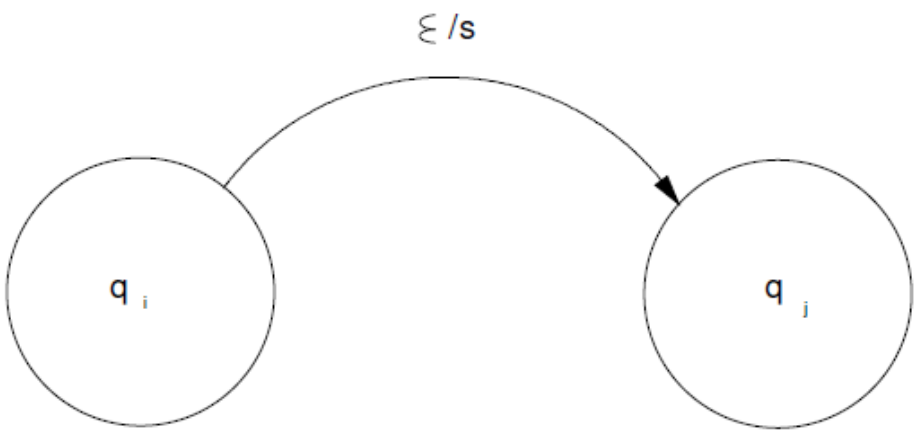
- Ambas funciones también se pueden representar en una misma tabla de la siguiente forma :

f / g	*a*	*b*
q_1	$q_1/0$	$q_2/1$
q_2	$q_3/0$	$q_2/0$
q_3	$q_3/1$	$q_1/0$

Representación de un autómata

(Mediante diagramas de estados)

- Los diagramas de Moore son otra forma de representar las funciones de transición y salida de un autómata.
- El diagrama de Moore es **un grafo orientado** en el que cada nodo corresponde a un estado; y si $f(\varepsilon, q_i) = q_j$ y $g(\varepsilon, q_i) = s$ existe un arco dirigido del nodo q_i al correspondiente q_j, sobre el que se pone la etiqueta ε / s.

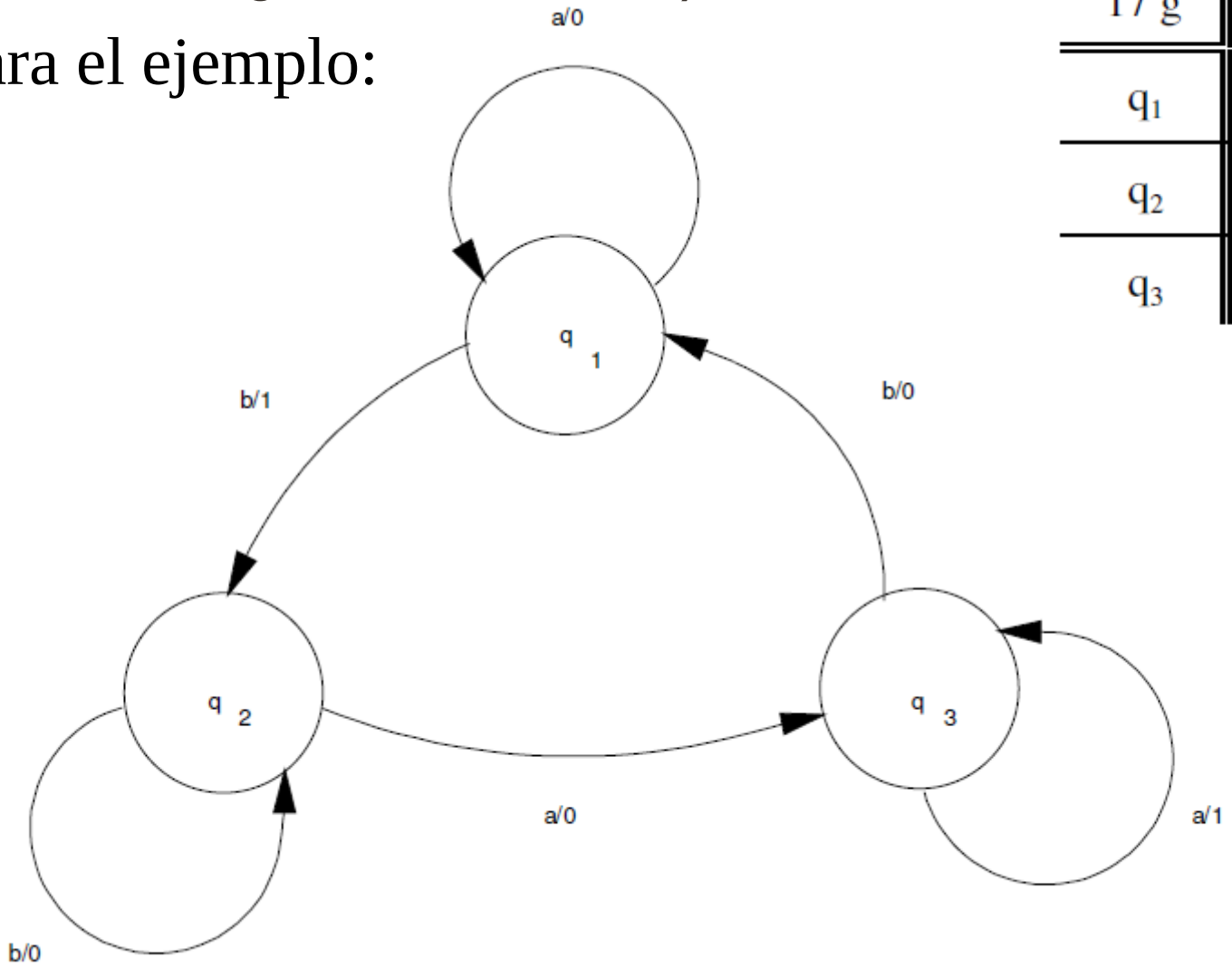


Representación de un autómata

(Mediante diagramas de Moore)

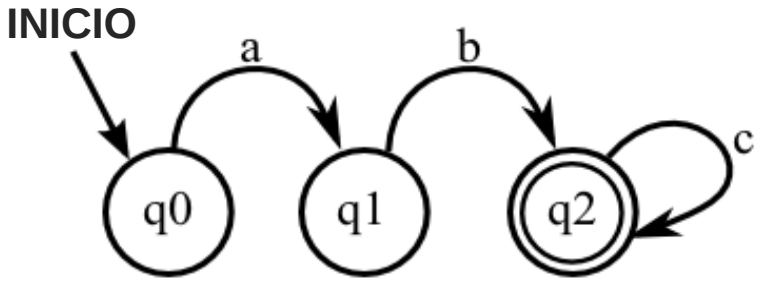
- Para el ejemplo:

f / g	a	b
q_1	$q_1/0$	$q_2/1$
q_2	$q_3/0$	$q_2/0$
q_3	$q_3/1$	$q_1/0$



Autómata finito

- Un **autómata finito** o **máquina de estado finito** es un modelo matemático, aplicado a la teoría de lenguajes opera como una máquina que recibe una cadena constituida por símbolos de un alfabeto y determina si esa cadena pertenece al lenguaje regular que el autómata reconoce, i.e. logra reconocer palabras de un lenguaje descrito por una **expresión regular**.



*abc**

Autómata finito

- Los **autómatas finitos reconocen los lenguajes regulares**, *o de tipo 3* y se pueden representar intuitivamente por una cinta y una cabeza de lectura.

•La *cinta de entrada, sólo contiene símbolos de un determinado alfabeto, y se mueve* en una sólo dirección.

•El *control de estados, determina el funcionamiento del autómata.*

•Una sentencia de un lenguaje determinado, colocada en la cinta, y leída por el autómata finito, es reconocida por éste, si el control de estados llega a un estado final.

Definición formal de autómata finito

Se puede definir como una quíntupla $AF = (E, Q, f, q_1, F)$ donde :

E = {conjunto finito de símbolos de entrada, que constituye el vocabulario}

Q = {conjunto finito de estados}

$f{:}E^* \times Q \rightarrow Q$ es la *función de transición*

$q_1 \in Q$, es el *estado inicial*

$F \subset Q$ es el *conjunto de estados finales*

Definición formal de autómata finito

Se entiende por *configuración* de un autómata finito, a un par de la forma (q, W) donde q, es el estado actual, y W la cadena que queda por leer en ese instante. Según la definición anterior, se puede afirmar que la *configuración inicial* de un autómata finito es el par (q_1, t) siendo t la sentencia o cadena de entrada a reconocer. La *configuración final* se representa por el par (q_i, λ) donde $q_i \in F$, y λ indica que no queda nada por entrar de la cinta.

Un *movimiento* de un autómata finito, puede definirse como el tránsito entre dos configuraciones, y se representa por $(q, aW) \rightarrow (q', W)$ y se debe de cumplir que $f(q,a)=q'$.

Lenguaje reconocido por un autómata finito

- Cuando un autómata transita a una configuración final partiendo de la configuración inicial, en varios movimientos, se dice que se ha producido aceptación o *reconocimiento de la cadena de entrada. Es decir que dicha cadena, pertenece al lenguaje reconocido por* el autómata.

- Por el contrario, *cuando el autómata finito no es capaz de llegar a un estado final*, se dice que el autómata no reconoce dicha cadena y que por tanto no pertenece al lenguaje.

$$L(AF) = \{t/t \in E^*, \quad (q_1, t) \rightarrow (q_i, \lambda), \quad q_i \in F\}$$

Lenguaje reconocido por un autómata finito

- Para toda gramática regular "G3", existe un autómata finito "AF", tal que el lenguaje reconocido por el autómata finito es igual al lenguaje generado por la gramática.

$$L(AF) = L(G_3)$$

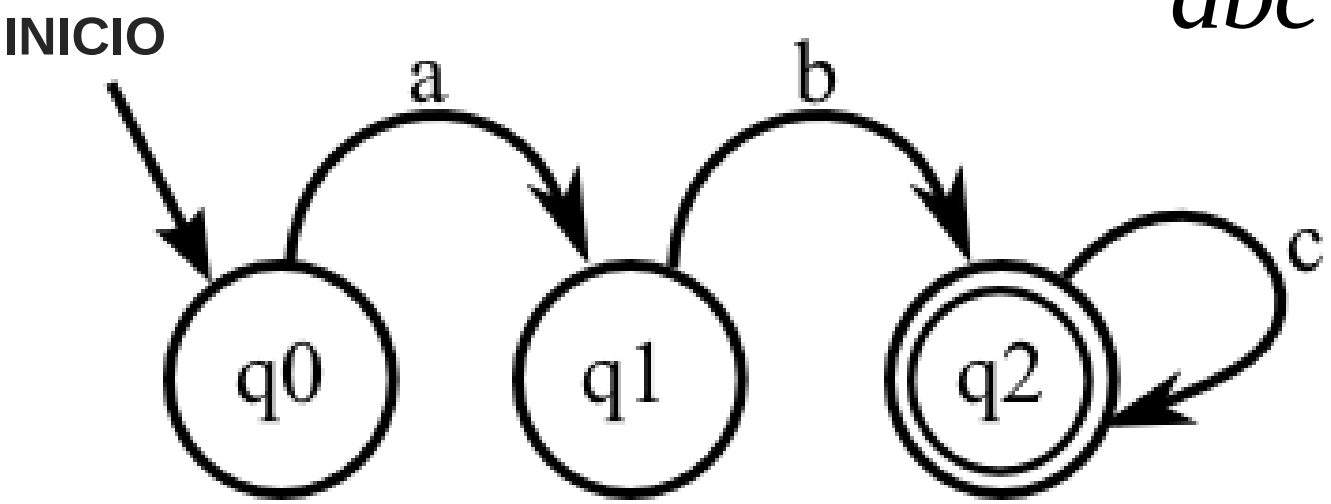
- La forma habitual de representar los autómatas finitos es mediante un grafo o diagrama de estados, donde los nodos son los estados y las ramas están marcadas con los símbolos del alfabeto de entrada. Las ramas se construyen según la función de transición, así debe de cumplir .

$$f(q_1, a) \rightarrow q_2$$

Lenguaje reconocido por un autómata finito

- *Los nodos que representan los estados finales, suelen marcarse con un doble círculo, y el estado inicial también se marca con una flecha, encima de la cual se coloca la palabra INICIO.*

*abc**



Ejemplo 01

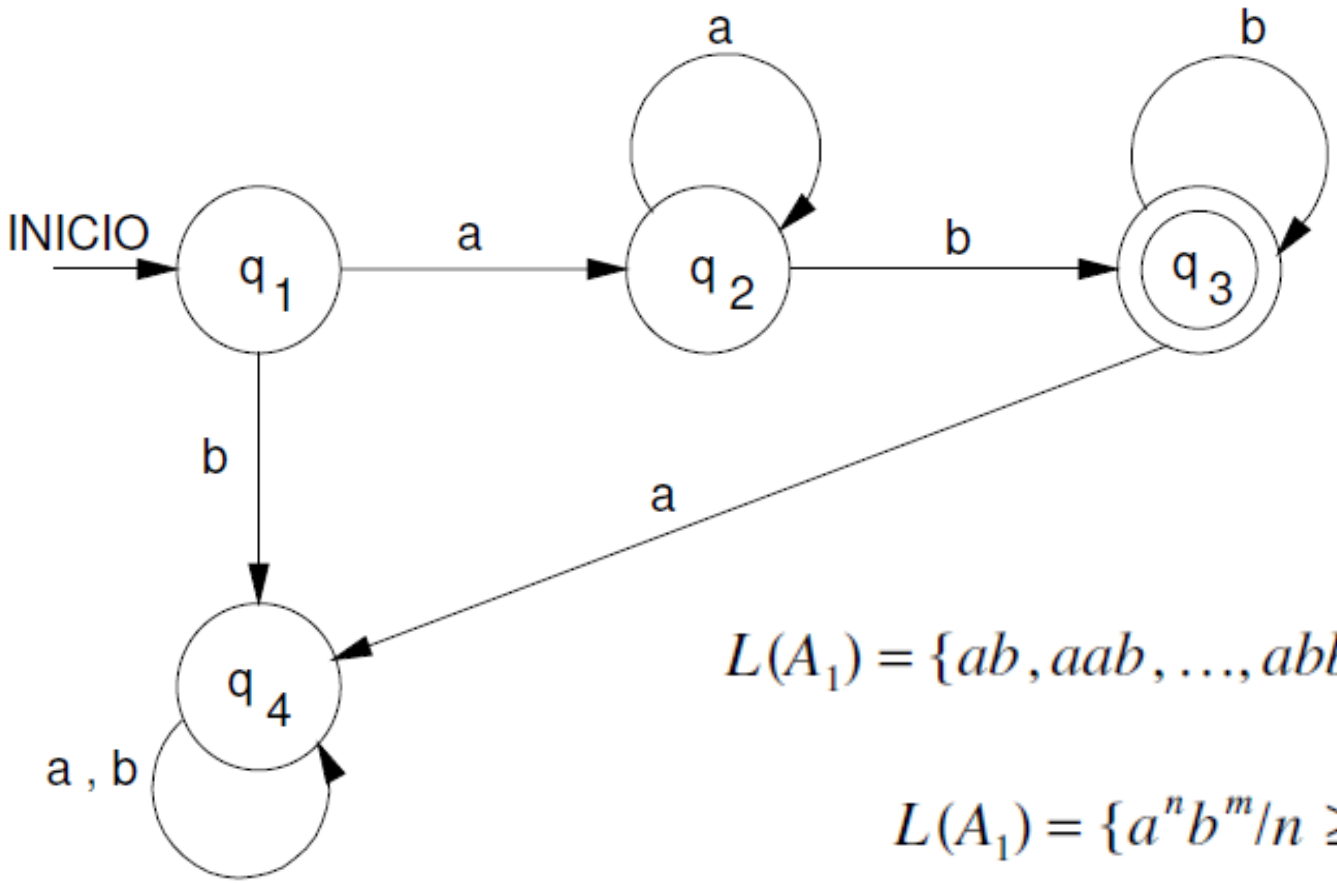
Sea el autómata finito $A_1 = (E, Q, f, q_1, F)$ donde $E = \{a, b\} \cup \{\lambda\}$; $Q = \{q_1, q_2, q_3, q_4\}$ y la función f viene dada por la tabla siguiente y el conjunto de estados finales es $F=\{q_3\}$

f	a	b
q_1	q_2	q_4
q_2	q_2	q_3
q_3	q_4	q_3
q_4	q_4	q_4

Ejemplo 01: Solución

- **Solución:** Se construye el diagrama de Moore, colocando en primer lugar todos los estados dentro de círculos, marcando con doble círculo el estado final. El estado inicial se indica con una flecha que lo señala con la palabra INICIO encima. Para construir las ramas, nos situamos en el primer estado de la tabla de transiciones y se observa que f(q1,a)=q2, entonces se traza una flecha entre q1 y q2, apuntando a q2, y se coloca encima de la flecha el símbolo del vocabulario de entrada a. De igual forma se recorre la tabla de transiciones para cada estado y entrada completándose el diagrama de Moore.

Ejemplo 01: Lenguaje reconocido por un autómata finito



$$L(A_1) = \{ab, aab, \ldots, abbb, \ldots, aabb, \ldots\}$$

$$L(A_1) = \{a^n b^m / n \geq 1 m \geq 1\}$$

La expresión regular que denota el lenguaje es a^+b^+ o también aa^*bb^*.

25

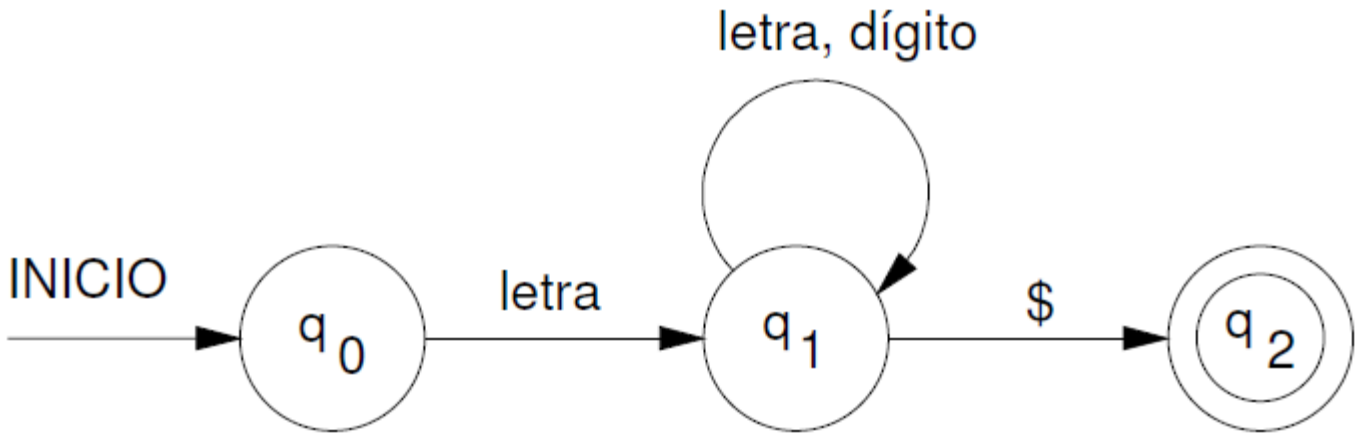
Ejemplo 02

- Los identificadores de C son cadenas de letras, dígitos y guiones bajos.

*letra_*	->	[A-Za- z_]
dígito	->	[0-9]
id	->	*letra_* (*letra_* | *dígito*)*

Ejemplo 02: Solución

- **Solución :** Este ejemplo es inverso al anterior, pues se da un lenguaje y se pide el autómata que lo reconoce. En primer lugar se construye un diagrama de Moore, de tal forma que a partir del estado inicial, después de leer una letra, acepte letras o dígitos de forma variable, y cuando encuentre un carácter diferente de letra o dígito alcance el estado final. El diagrama de Moore es el que se muestra en la figura.



Ejemplo 02: Solución

- $ representa a el fin de la cadena

- El autómata finito se deduce del diagrama de estaos y es el siguiente:

$$AF = (E = \{a, b, \ldots, z, 0, 1, \ldots 9, \$\}, Q = \{q_0, q_1, q_2\}, f, q_0, F = \{q_2\})$$

- donde f se define por :

f	<letra>	<dígito>	$
q_0	q_1	-	-
q_1	q_1	q_1	q_2
q_2	-	-	-

Mapas Conceptuales 01 y 02

- La fecha máxima de entrega de los mapas es el domingo 27 de Febrero de 2011 a las 23:59:59 hrs.

  - *Mapa conceptual 01:* De las clases 02 y 03
  - *Mapa conceptual 02:* De las clases 05 a la 09
  - Si deberá de incluir encabezados en cada mapa.
  - Recomendación usar CmapTools
  - Enviar separados con el nombre:
    - *MapaConceptual0x_NombreAlumno.pdf*

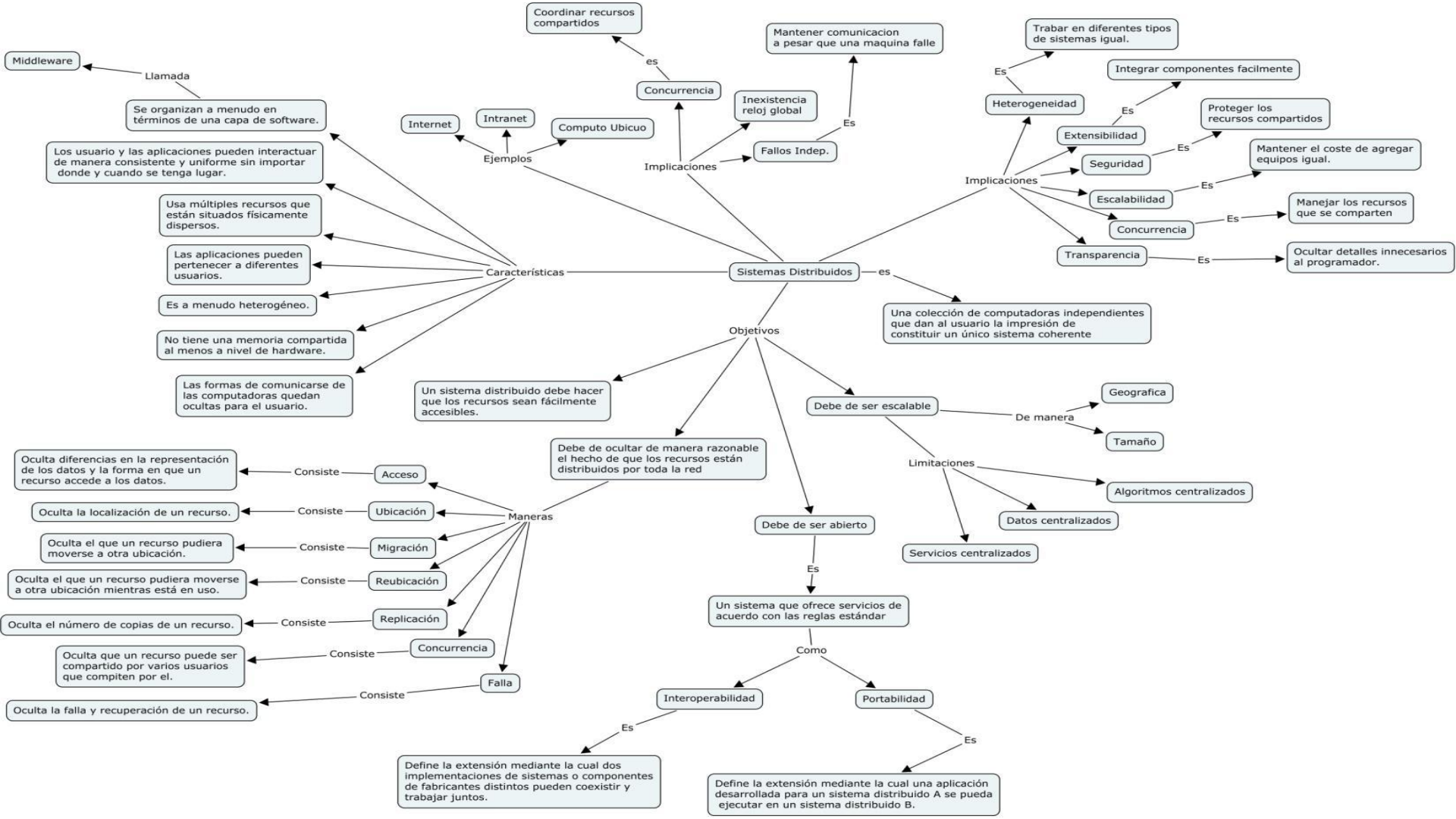
Características de un mapa conceptual

- **Jerarquización:** Se refiere a la ordenación de los conceptos más generales e inclusivos en la parte superior o central y mediante una diferenciación progresiva, están incluidos los conceptos más específicos.

- **Impacto visual:** Debe considerar la limpieza, espacios, claridad, ortografía para reducir confusiones y amontonamientos, por ello es conveniente dibujarlos varias veces ya que el primer mapa que se construye tiene siempre, casi con toda seguridad algún defecto. También se recomienda usar óvalos ya que son más agradables a la vista que los triángulos y los cuadrados.

- **Simplificación:** Se refiere a la selección de los conceptos más importantes, haciendo una diferenciación del contenido y localizando la información central de la que no lo es para una mejor comprensión y elaboración de un contenido. Los conceptos, al ir relacionándose por medio de las palabras enlace, se van almacenando en la mente de modo organizado y jerárquico de manera que serán más fácilmente comprendidos.

Ejemplo


Sistemas Distribuidos
es
Una colección de computadoras independientes que dan al usuario la impresión de constituir un único sistema coherente
Características
Llamada
Middleware
Se organizan a menudo en términos de una capa de software.
Los usuario y las aplicaciones pueden interactuar de manera consistente y uniforme sin importar donde y cuando se tenga lugar.
Usa múltiples recursos que están situados físicamente dispersos.
Las aplicaciones pueden pertenecer a diferentes usuarios.
Es a menudo heterogéneo.
No tiene una memoria compartida al menos a nivel de hardware.
Las formas de comunicarse de las computadoras quedan ocultas para el usuario.
Ejemplos
Internet
Intranet
Computo Ubicuo
Implicaciones
es
Concurrencia
Coordinar recursos compartidos
Inexistencia reloj global
Fallos Indep.
Es
Mantener comunicacion a pesar que una maquina falle
Implicaciones
Es
Heterogeneidad
Trabar en diferentes tipos de sistemas igual.
Es
Extensibilidad
Integrar componentes facilmente
Es
Seguridad
Proteger los recursos compartidos
Es
Escalabilidad
Mantener el coste de agregar equipos igual.
Es
Concurrencia
Manejar los recursos que se comparten
Es
Transparencia
Ocultar detalles innecesarios al programador.
Objetivos
Un sistema distribuido debe hacer que los recursos sean fácilmente accesibles.
Debe de ocultar de manera razonable el hecho de que los recursos están distribuidos por toda la red
Maneras
Consiste
Acceso
Oculta diferencias en la representación de los datos y la forma en que un recurso accede a los datos.
Consiste
Ubicación
Oculta la localización de un recurso.
Consiste
Migración
Oculta el que un recurso pudiera moverse a otra ubicación.
Consiste
Reubicación
Oculta el que un recurso pudiera moverse a otra ubicación mientras está en uso.
Consiste
Replicación
Oculta el número de copias de un recurso.
Consiste
Concurrencia
Oculta que un recurso puede ser compartido por varios usuarios que compiten por el.
Consiste
Falla
Oculta la falla y recuperación de un recurso.
Debe de ser escalable
De manera
Geografica
Tamaño
Limitaciones
Algoritmos centralizados
Datos centralizados
Servicios centralizados
Debe de ser abierto
Es
Un sistema que ofrece servicios de acuerdo con las reglas estándar
Como
Interoperabilidad
Es
Define la extensión mediante la cual dos implementaciones de sistemas o componentes de fabricantes distintos pueden coexistir y trabajar juntos.
Portabilidad
Es
Define la extensión mediante la cual una aplicación desarrollada para un sistema distribuido A se pueda ejecutar en un sistema distribuido B.