UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

Centro Universitario UAEM Zumpango

Ingeniero en Computación

Autómatas y Lenguajes Formales
Unidad de Competencia II
Manejar la teoría de autómatas para conocer su relación con los lenguajes de programación

UAEM | Universidad Autónoma del Estado de México

M. en C. Rafael Rojas Hernández

rrojas.uaemex@gmail.com

Agosto 2016

Índice de la presentación

- Información general de la Unidad
- Estructura de la Unidad de Aprendizaje
- Unidad de Competencia II

Información general de la Unidad de Aprendizaje

Unidad de Aprendizaje

Autómatas y Lenguajes Formales

Propósito de la Unidad de Aprendizaje

Aplicará las estructuras algebraicas fundamentales en el manejo de estructuras de datos, así como los principios básicos para el diseño y la simplificación de circuitos lógicos, y el manejo de autómatas y su relación con los lenguajes de programación. Presentar una variedad de modelos formales de cómputo explicando su importancia desde puntos de vista teóricos y prácticos.

Estructura de la Unidad de Aprendizaje

1. Representar información a través del uso de grafos
2. **Manejar la teoría de autómatas para conocer su relación con los lenguajes de programación**
3. Conocer, utilizar y manipular expresiones regulares
4. Conocer, utilizar y diseñar gramáticas de libre contexto
5. Simplificar y normalizar gramáticas libres de contexto
6. Conocer la teoría básica de la Máquina de Turing

Objetivo de la Unidad de Competencia

Representar información a través del uso de autómatas finitos determinísticos, autómatas finitos no determinísticos, autómatas finitos no determinísticos con transiciones ϵ y equivalencias entre ellos.

Conocimientos

- Terminología básica.
- Aplicación en máquinas de estado finito.

Manejar la teoría de autómatas para conocer su relación con los lenguajes de programación

Habilidades

Comprender los elementos básicos en el uso de autómatas, para su conceptuación, análisis, a través del diseño de soluciones creativas y funcionales.

Actitudes y valores

Receptiva, analítica, propositiva, crítica y tolerancia.

- Definiciones básicas
- Autómatas finitos determinísticos
  - Autómatas finitos
  - Autómatas finitos deterministas
  - Tabla de transiciones
  - Lenguaje de un AFD
- Autómatas finitos no determinísticos
  - Autómatas finitos no determinísticos
  - Lenguaje de un AFN
- Autómatas finitos no determinísticos con transiciones ϵ
  - Autómatas finitos no determinísticos con transiciones ϵ
- Autómatas finitos no determinísticos con equivalencias entre ellos

Alfabeto

Un **alfabeto** es un conjunto finito no vacío de símbolos. Por convenio se utiliza el símbolo Σ para representar un alfabeto.

Entre los alfabetos más comunes se encuentran:

- $\Sigma = \{0, 1\}$, el alfabeto binario.
- $\Sigma = \{a, b, \ldots, z\}$, el conjunto de todas las letras minúsculas.
- El conjunto de todos los caracteres ASCII o el conjunto de todos los caracteres ASCII imprimibles.

Cadena

Una **cadena** (a veces llamada **palabra**) es una secuencia finita de símbolos partenecientes a un alfabeto.

01101 es una cadena del alfabeto binario $\Sigma = \{0, 1\}$. La cadena 111 es otra cadena de este alfabeto.

Cadena vacía

La **cadena vacía** es la cadena que contiene cero símbolos. Esta cadena, que se representa con ϵ, o también mediante λ, puede constituirse como cualquier alfabeto.

Longitud de una cadena

La **longitud** de una cadena es el número de posiciones de la cadena ocupadas por símbolos.

En la cadena 01101 hay solo dos símbolos, 0 y 1, pero hay cinco *posiciones* ocupadas por símbolos, y su longitud es 5.

- Sea $w = aeiou$, $|w| = 5$.
- Sea $u = 010101010101010101$, $|u| = 18$.
- Sea $v = anitalavalatina$, $|v| = 15$.

Potencias de un alfabeto

Se define a Σ* como el conjunto de cadenas de longitud k, tales que todos los dímbolos que las forman pertenecen a Σ.

Si $\Sigma = \{0, 1\}$, entonces:

- $\Sigma^1 = \{0, 1\}$.
- $\Sigma^2 = \{00, 01, 10, 11\}$.
- $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$.

Por convenio, el conjunto de todas las cadenas de una alfabeto Σ se representa como Σ^*, $\Sigma^0 = \{\lambda\}$, y el conjunto de cadenas no vacías se representa como Σ^+, sea cual sea el alfabeto Σ.

Concatenación de cadenas

Sean x e y cadenas, la **concatenación** de x e y, denotada por xy, es la cadena que se forma al realizar una copia de x seguida de una copia de y.

- Si $u = lacasa$, $v = roja$ son cadenas; la concatenación $uv = lacasaroja$.
- Si $u = lacasa$, $v = roja$ son cadenas; la concatenación $vu = rojalacasa$.
- Si $u = linea$, $v = circulo$, $w = rectangulo$ son cadenas; la concatenación $wuv = rectangulolineacirculo$.

Reversa de una cadena

La *reversa* de una cadena se obtiene escribiendo los símbolos en orden reverso, si w es una cadena $w = a_1 a_2 \dots a_n$ su reversa es:

$$w^R = a_n \dots a_2 a_1$$

- Sea la cadena $u = pais$, la reversa de la cadena es $u^R = siap$.
- Sea la cadena $u = natas$, la reversa de la cadena es $u^R = satan$.
- Sea la cadena $u = salas$, la reversa de la cadena es $u^R = salas$.

Subcadena

Cualquier cadena de caracteres consecutivos en una cadena w se dice que es una *subcadena* de w si:

$$w = vu$$

donde las subcadenas v y u se dice que son *prefijo* y *sufijo* de w respectivamente.

- La cadena $w = temario$, tiene como subcadenas a $u = tema$ y $v = rio$, y son el prefijo y sufijo respectivamente.
- La cadena $w = camino$, tiene como subcadenas a $u = ca$ y $v = mino$, y son el prefijo y sufijo respectivamente.
- La cadena $w = descapotable$ tiene como subcadenas a $u_0 = des$, $u_1 = capo$ y $u_2 = table$.

Lenguaje

Se llama *lenguaje* a un conjunto de cadenas, todas ellas elegidas de algún Σ^*, donde Σ es un alfabeto. Si Σ es un alfabeto, y $L \subseteq \Sigma^*$, entonces L es un *lenguaje de* Σ.

- El lenguaje formado por todas las cadenas compuestas por n ceros seguidos por n unos, para cualquier $n \geq 0$: $\{\lambda, 01, 0011, 000111, \ldots$
- El conjunto de todas las cadenas de ceros y unos con igual número de ambos símbolos: $\{\lambda, 01, 10, 0011, 0101, 1001, \ldots\}$.
- El conjunto de los números binarios que representan un número primo:
  $\{10, 11, 101, 111, 1011, \ldots\}$.

Operaciones con Lenguajes:

Unión	$L_3 = L_1 \cup L_2$
Intersección	$L_3 = L_1 \cap L_2$
Diferencia	$L_3 = L_1 - L_2$
Complemento	$\overline{L} = \Sigma^* - L$
Inversa de un lenguaje	$L^R = \{w^R, w \in L\}$
Concatenación	$L_1 L_2 = \{xy, x \in L_1, y \in L_2\}$

Si $L = \{a^n b^n, n \geq 0\}$ entonces:

$$\begin{aligned} L^2 &= \{a^n b^n a^m b^m, m \geq 0, m \geq 0\} \\ L^R &= \{b^n a^n, n \geq 0\} \end{aligned}$$

Gramática

Una *grámatica* G se define como una cuádrupla

$$G(V, T, S, P)$$

donde:

$V =$ conjuno finito de objetos llamados *variables*.

$T =$ conjuno finito de objetos llamados *símbolos terminales*.

$S \in V$ es un símbolo especial llamado la *variable inicial*.

$P =$ conjunto finito de precedencias.

Gramática

Además, V y T son no vacíos y disjuntos, se asume que todas las reglas son de la forma:

$$x \to y \exists x \in V, y \in (V \cup T)^*$$

La operación $x \to y$ indica que x produce y.

Derivación

Si $w_1 \to w_2 \to \ldots w_n$ se dice que w_1 *deriva* w_n y se escribe $w_1 \xrightarrow{*} w_n$.

Lenguaje

Sea $G = (V, T, S, P,)$ una gramática, entonces el conjunto $L(G) = \{w \in T_* : S \xrightarrow{*} w\}$ es el lenguaje generado por G.

Para mostrar que un lenguaje dado es generado por una gramática G debe mostrarse (i) que cada $w \in L$ puede ser derivada de S usando G y (ii) que cada cadena derivada este en L.

Ejemplo: Sea $G = (\{S\}, \{a, b\}, S, P)$, con P dada por:

$$\begin{array}{lcl} S & \rightarrow & aSb \\ S & \rightarrow & \lambda \end{array}$$

entonces: $S \stackrel{*}{\rightarrow} aabb$.
– $aabb$ es una sentencia del lenguaje generado por G.
– $aaSbb$ es una forma de sentencias.

Ejemplo: Encontrar una gramática $L(G)$ que genere

$$L = \{a^n b^{n+1}, n \geq 0\}$$

$$G = (V, T, S, P)$$

donde:

$$\begin{aligned} V &= \{A, S\} \\ S &= A \\ T &= \{a, b\} \\ P &= A \rightarrow Sb \\ & \quad S \rightarrow aSb \\ & \quad S \rightarrow \lambda \end{aligned}$$

Gramáticas equivalentes

Se dice que dos gramáticas G_1 y G_2 son equivalentes si generan el mismo lenguaje, esto es, si:

$$L(G_1) = L(G_2)$$

Ejemplo: Considere la gramática $G_1(\{A, S\}, \{a, b\}, S, P_1\}$, P_1 con las producciones:

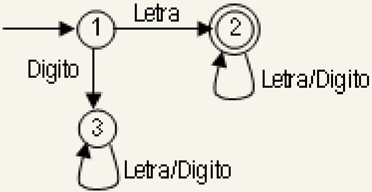
$$\begin{aligned} S &\rightarrow aAb|\lambda \\ A &\rightarrow aAb|\lambda \end{aligned}$$

Esta gramática es equivalente a la gramática $G_2(\{S\}, \{a, b\}, S, P)$ con producciones:

$$\begin{aligned} S &\rightarrow aSb \\ S &\rightarrow \lambda \end{aligned}$$

Dado que ambas producen el lenguaje $L = \{a^n b^n : n \geq 0\}$.

Un autómata finito tiene un conjunto de estados, y su "control" se mueve de estado en estado, en respuesta a "entradas" externas. Los autómatas finitos se dividen en diversas clases, dependiendo de si su control es "determinista" (lo que significa que el autómata no puede estar en más de un estado simultáneamente) o "no determinista" (lo que significa que puede estar en varios estados al mismo tiempo).



También un autómata puede ser considerado como un modelo abstracto de una *computadora digital*, con las siguientes características:

- La unidad de control que puede estar en algún número finito de estados internos y puede cambiar de estado de una manera definida.
- Se asume que opera en marcas de tiempo discreto.
- En un momento dado, la unidad de control esta en algún estado interno y el mecanismo de entrada analiza un símbolo partículas de la entrada.
- El estado interno de la unidad de control en el siguiente instánte de tiempo esta determinado por el *siguiente estado* o *función de transición*.

Computadora Analógica

Representan los números mediante una cantidad física, es decir, asignan valores numéricos por medio de la medición física de una propiedad real, como la longitud de un objeto, el ángulo entre dos líneas o la cantidad de voltaje que pasa a través de un punto en un circuito eléctrico.

Las computadoras analógicas obtienen todos sus datos a partir de alguna forma de medición. La presición de los datos usados en una computadora analógica está íntimamente ligada a la precisión con que puede medirse.

Computadora Digital

Representa los datos o unidades separadas. La forma más simple de computadora digital es contar con los dedos. Cada dedo representa una unidad del artículo que se esta contando. A diferencia de la computadora analógica, esta limitada por la precisión de las mediciones que pueden realizarse, la computadora digital puede representar correctamente los datos con tantas precisiones y números que se requieran.

Las computadoras analógicas miden, mientras que las computadoras digitales cuentan.

Autómata Finito Determinista (AFD)

Un **AFD** esta definido por la quíntupla:

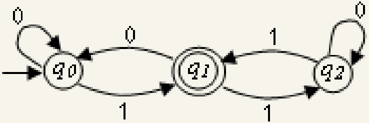
$$A = (Q, \Sigma, \delta, q_0, F)$$

donde: $Q =$ un conjunto finito de estados internos.
$\Sigma =$ un conjunto finito de símbolos llamado el alfabeto de entrada.
$\delta = Q \times \Sigma \rightarrow Q$ es una función total llamada *función de transición*.
$q_0 \in Q$ es el estado inicial.

Grafo de transición de un AFD

Si $M = (Q, \Sigma, \delta, q_0, F)$ es un AFD, entonces su **grafo de transición** G_M tiene exactamente $|Q|$ vértices, cada uno etiquetado con diferente $q_i \in Q$. Cada regla de transición $\delta(q_i, a) = q_j$ en el grafo, tiene una arista (q_i, q_j) etiquetando a. El vértice asociado con q_0 es llamado vértice inicial, mientras aquellos etiquetados con $q_i \in F$ son los vértices finales.

El siguiente grafo:



Representa al AFD $M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$, donde δ esta dada por:

$\delta(q_0, 0) = q_0 \quad \delta(q_0, 1) = q_1$
$\delta(q_1, 0) = q_0 \quad \delta(q_1, 1) = q_2$
$\delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_1$

Tabla de transiciones de un AFD

Una **tabla de transiciones** es una representación tabular convencional de una función como δ, que recibe dos argumentos y devuelve un valor. Las filas de la tabla corresponden a los estados, y las columnas a las entradas. El valor correspondiente a la fila del estado q y a la columna de la entrada a es el estado $\delta(q, a)$.

Función de transición extendida

Sí $\delta(q_0, a) = q_1$ y $\delta(q_1, b) = q_2$, entonces $\delta^*(q_0, ab) = q_2$.

Ejemplo: Diseñe un AFD que acepte el lenguaje:

$$L = \{w | w \text{ tiene un número par de ceros y unos}\}$$

El AFD es:

$$A = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_0\})$$

Con transiciones:

$\delta(q_0, 0) = q_2 \quad \delta(q_0, 1) = q_1$
$\delta(q_1, 0) = q_3 \quad \delta(q_1, 1) = q_0$
$\delta(q_2, 0) = q_0 \quad \delta(q_2, 1) = q_3$
$\delta(q_3, 0) = q_1 \quad \delta(q_3, 1) = q_2$

Tabla de transiciones:

	0	1
$* \rightarrow q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

Lenguaje de un AFD

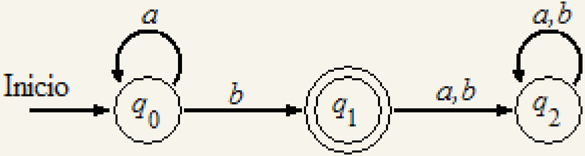
El lenguaje aceptado por un AFD $A = (Q, \Sigma, \delta, q_0, F)$ es el conjunto de todas las cadenas sobre Σ aceptadas por A. Esto es:

$$L(A) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F\}$$

Un AFD procesará cualquier cadena en Σ^* y la aceptara o no. La no aceptación significa que el AFD se detiene en un estado no final.

$$\overline{L}(A) = \{w \in \Sigma^* : \delta^*(q_0, w) \notin F\}\}$$

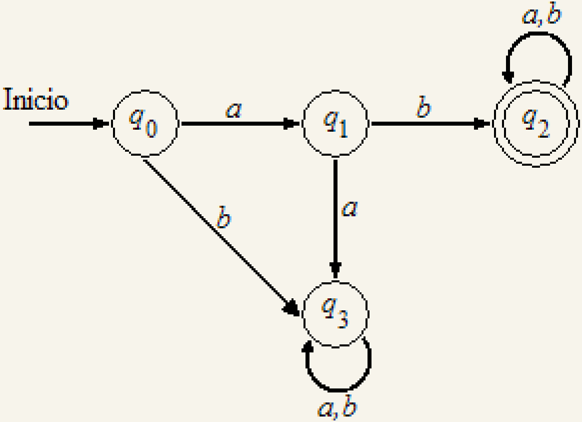
Dado el AFD de la figura, ¿qué lenguaje acepta?.



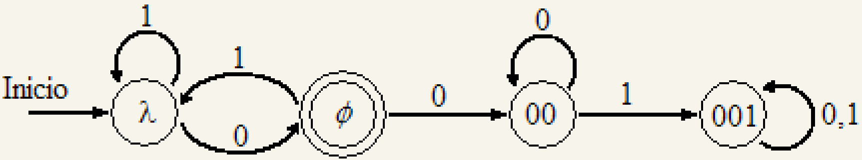
$$L = \{a^n b : n \geq 0\}$$

Ejemplo

Encuentre un AFD que reconozca el conjunto de todas las cadenas sobre $\Sigma = \{a, b\}$ que inicien con el prefijo ab.



Encuentre un AFD que acepte todas las cadenas sobre $\{0, 1\}$, excepto aquellas que contengan la subcadena 001.

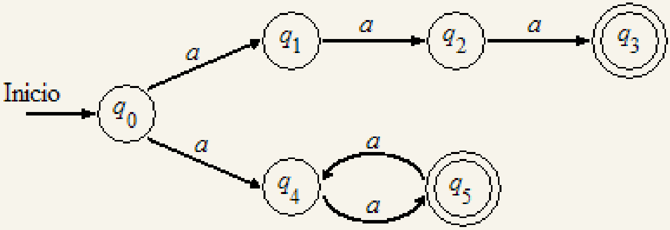


Autómata Finito No Determinista (AFN)

Un AFN esta definido por la quíntupla:

$$A = (Q, \Sigma, \delta, q_0, F)$$

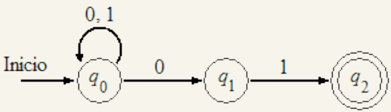
donde: Q, Σ, q_0 y F se definen igual que para un AFD, pero $\delta : Q \times (\Sigma \cup \{\lambda\}) \to 2^Q$.



El AFN de la figura puede especificarse formalmente de la siguiente manera:

$$(\{q_1, q_2, q_3\}, \{0, 1\}, \sigma, q_0, \{q_2\})$$

La función de transición δ esta dada por la tabla.



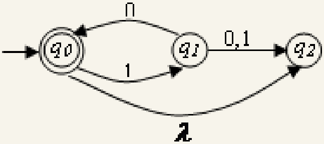
	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	ϕ	$\{q_2\}$
$*q_2$	ϕ	ϕ

Lenguaje de un AFN

El lenguaje aceptado por un AFN $A = (Q, \Sigma, \delta, q_0, F)$ esta definido como el conjunto de todas las cadenas en el siguiente sentido.
Si entre dos vértices v_i y v_j existe algún camino etiquetado w, entonces debe haber algún camino etiquetado w de longitud no más que $\# + (1 + \#)|w|$ donde $\#$ es el número de aristas λ en el grafo.
Formalmente:

$$L(A) = \{w \in \Sigma^* : \delta^*(q_i, w) \cap F \neq \phi\}$$

¿Cuál es el lenguaje aceptado por el autómata de la figura?.



$L = \{(01)^n : n \geq 0\}$
¿Qué pasa con la cadena 110?
$\sigma(q_2, 0)$ es una configuración muerta.
Se puede decir que $\sigma^*(q, 110) = \phi$.

AFN-ϵ

Un AFN-λ esta definido por la quíntupla:

$$A = (Q, \Sigma, \delta, q_0, F)$$

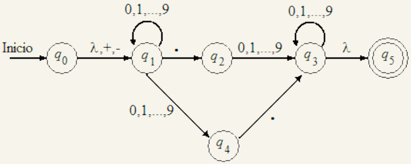
donde, Q, Σ, q_0 y F se definen igual que para un AFN y δ es ahora una función con dos argumentos:

1. Un estado perteneciente a Q.
2. Un miembro de $\Sigma \cup \{\lambda\}$, es decir, un símbolo de entrada o el símbolo λ. Se exige que λ, el símbolo que representa la cadena vacía, no forme parte del alfabeto Σ, para que no produzca confusiones.

El AFN-*epsilon* de la figura se representa formalmente así:

$$E = (\{q_0, q_1, \ldots, q_5\}, \{., +, -, 0, 1, \ldots, 9\}, \delta, q_0, \{q_5\})$$

donde δ viene definido por la tabla de transiciones.

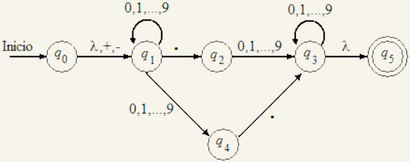


	λ	$+, -$	$.$	$0, 1, \ldots, 9$
q_0	$\{q_1\}$	$\{q_1\}$	ϕ	ϕ
q_1	ϕ	ϕ	$\{q_2\}$	$\{q_1, q_4\}$
q_2	ϕ	ϕ	ϕ	$\{q_3\}$
q_3	$\{q_5\}$	ϕ	ϕ	$\{q_3\}$
q_4	ϕ	ϕ	$\{q_3\}$	ϕ
q_5	ϕ	ϕ	ϕ	ϕ

Clausuras respecto de λ

Con el estado q en $CLAUS_\lambda(q)$, si el estado p está en $CLAUS_\lambda(q)$ y existe una trasición desde el estado p al estado r con la etiqueta λ, entonces r está en $CLAUS_\lambda(q)$, es decir, si δ es la función de transición del AFN-λ y p está en $CLAUS_\lambda(q)$, entonces $CLAUS_\lambda(q)$ también contiene todos los estados de $\delta(p, \lambda)$.

Para el autómata de la figura, la clausura respecto de λ de cada estado contiene únicamente dicho estado, con dos excepciones: $CLAUS_\lambda(q_0) = \{q_0, q_1\}$ y $CLAUS_\lambda(q_3) = \{q_3, q_5\}$.



	λ	$+,-$	$.$	$0,1,\ldots,9$
q_0	$\{q_1\}$	$\{q_1\}$	ϕ	ϕ
q_1	ϕ	ϕ	$\{q_2\}$	$\{q_1, q_4\}$
q_2	ϕ	ϕ	ϕ	$\{q_3\}$
q_3	$\{q_5\}$	ϕ	ϕ	$\{q_3\}$
q_4	ϕ	ϕ	$\{q_3\}$	ϕ
q_5	ϕ	ϕ	ϕ	ϕ

Eliminación de transiciones λ

Dado un AFN-λ, se puede encontrar un AFD D que acepte el mismo lenguaje que E. La única diferencia es que hay que agregar las transiciones λ de E, lo que se hace mediante el mecanismo de la clausura de λ.

Sea $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$. El AFD equivalente se define así:

1. Q_D es el conjunto de los subconjuntos de Q_E.
2. $q_D = CLAUS_\lambda(q_0)$.
3. $F_D = \{S | S$ pertenece a Q_D, y $S \cap F_E \neq \phi\}$.
4. $\delta_D(S, a)$ se calcula $\forall a \in \Sigma$ y $\forall S \in Q_D$:
   - 4.1 Sea $S = \{p_1, p_2, \ldots, p_k\}$.
   - 4.2 Se calcula $\bigcup_{i=1}^{k} \delta(p_i, a)$; sea este conjunto $\{r_1, r_2, \ldots, r_m\}$.
   - 4.3 Entonces, $\delta_D(S, a) = \bigcup_{i=1}^{m} CLAUS_\lambda(r_j)$.

Autómas equivalentes

Dos autómatas A_1 y A_2 se dice que son equivalentes si:

$$L(A_1) = L(A_2)$$

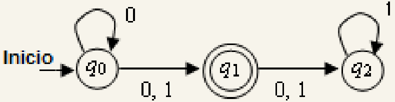
es decir, si ambos aceptan el mismo lenguaje.

Sea L un lenguaje aceptado por un AFN $A_N(Q_N, \Sigma, \delta_N, q_0, F_N)$. Entonces existe un AFD $A_D(Q_D, \Sigma, \delta_D, q_0, F_D)$ tal que:

$$L(A_N) = L(A_D)$$

1. Crear un grafo G_D con vértices $\{q_0\}$. Identificar este vértice como el estado inicial.
2. Repetir los siguientes pasos hasta que no haya más aristas pérdidas.
   2.1 Tomar cualquier vértice $\{q_i, q_j, \ldots, q_k\}$ de G_D que no tenga aristas de salida para alguna $a \in \Sigma$.
   2.2 Calcular $\delta^*(q_i, a), \delta^*(q_j, a), \ldots, \delta^*(q_k, a)$.
   2.3 Formar la unión de todos estos δ^*, creando el conjunto $\{q_l, q_m, \ldots, q_n\}$.
   2.4 Crear un vértice para G_D etiquetado $\{q_l, q_m, \ldots, q_n\}$ si es que éste no existe.
   2.5 Agregar una arista desde $\{q_i, q_j, \ldots, q_k\}$ hasta $\{q_l, q_m, \ldots, q_n\}$ y etiquetarlo con m.
3. Cada estado de G_D cuya etiqueta contiene cualquier $q_f \in F_N$ es identificado como un vértice final.
4. Si A_N acepta λ, el vértice $\{q_0\}$ en G_D se hace también vértice final.

Convertir el AFN de la figura a su AFD equivalente

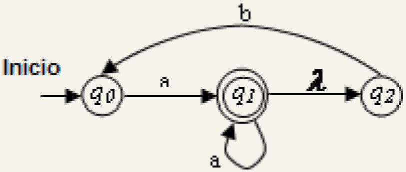


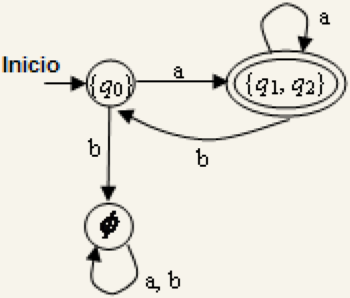
$\delta^*(\{q_0\},0) = \{q_0, q_1\}$
$\delta^*(\{q_0, q_1\},0) = \{q_0 q_1, q_2\}$
$\delta^*(\{q_0, q_1, q_2\},0) = \{q_0, q_1, q_2\}$
$\delta^*(\{q_1, q_2\},0) = \{q_2\}$
$\delta^*(\{q_1\},0) = \{q_2\}$
$\delta^*(\{q_2\},0) = \phi$

$\delta^*(\{q_0\},1) = \{q_1\}$
$\delta^*(\{q_0, q_1\},1) = \{q_1 q_2\}$
$\delta^*(\{q_0, q_1, q_2\},1) = \{q_1, q_2\}$
$\delta^*(\{q_1, q_2\},1) = \{q_2\}$
$\delta^*(\{q_1\},1) = \{q_2\}$
$\delta^*(\{q_2\},1) = \{q_2\}$

$$S = \{q_0\}$$

Convertir el AFN de la figura a su AFD equivalente





Bibliografía

- Hopcrof John E., Ullman Jeffrey D; ''Introducción a la teoría de autómatas, lenguajes y computación'', CECSA, México 2000.
- Ayres Frank Jr., ''Álgebra Moderna'', Mc Graw Hill, México 1991.
- Barco Gómez C., Barco Gómez G., Aristazábal Botero W., ''Matemática Digital'', Mc Graw Hill, Colombia. 1998.
- Grossman W.- Jerrold, "Discrete Mathematics an Introduction to Concepts, Methods and Applications", Mac Millan Publishing Co., USA, 1992.
- Jonhnsonbaugh R., ''Matemáticas Discretas'', Grupo Editorial Iberoamérica, México, 1988.
- Kelley Dean; ''Teoría de Autómatas y Lenguajes Formales'', Prentice Hall, España, 1995.

Bibliografía

- Kolman Bernard, Busby Robert C., ''Estructuras de Matemáticas Discretas para la Computación'', Prentice Hall, México, 1997.
- Manno Morris, ''Lógica Digital'', Prentice Hall.
- Ross Kenneth A., et.al., ''Matemáticas Discretas'', Prentice Hall, México, 1990.
- Harrison, ''Introduction to witching and automata theory''; Mc Graw Hill,
- D.Kosen, ''Automata and Computability'', Springer Verlag, 1997.
- Brookshear ''Theory of Computation; Formal Languages, Automata and Complexity'', Cummings, 1989.