Universidad Autónoma del Estado de México
Centro Universitario UAEM Texcoco

Departamento de Ciencias Aplicadas.

Ingeniería en Computación.

Autómatas y Lenguajes Formales.

Unidad de competencia III: “Conocer, utilizar y manipular expresiones regulares”

Presenta:
M. en C. C. J. Jair Vázquez Palma.

Autómatas y Lenguajes Formales

Objetivos de la Unidad de la Unidad de Aprendizaje

- Aplicar las estructuras algebraicas fundamentales en el manejo de estructuras de datos.
- Diseñar y simplificar circuitos lógicos.
- Manejo de autómatas y su relación con los lenguajes de programación.
- Representar en modelos formales de cómputo explicando su importancia desde puntos de vista teóricos y prácticos.

Unidad III: Conocer, utilizar y manipular expresiones regulares.

Contenido:

- Operaciones de concatenación y cerradura sobre conjunto de cadenas.
- Definición de expresiones regulares, autómatas finitos.

- Lenguajes y expresiones regulares.

Objetivo de la Unidad de Competencia III

- El discente adquirirá la de analizar y manipulación de expresiones regulares y se representación en autómatas finitos deterministas.

Conocer, utilizar y manipular expresiones regulares.

Lenguajes Regulares

•Los lenguajes regulares se llaman así porque sus palabras contienen “regularidades” o repeticiones de los mismos componentes, como por ejemplo en el lenguaje L1 siguiente:

$$L1 = \{ab, abab, ababab, abababab, . . .\}$$

•En este ejemplo se aprecia que las palabras de L1 son simplemente repeticiones de “ab” cualquier número de veces. Aquí la “regularidad” consiste en que las palabras contienen “ab” algún número de veces. Otro ejemplo más complicado sería el lenguaje L2:

$$L2 = \{abc, cc, abab, abccc, ababc, . . .\}$$

•La regularidad en L2 consiste en que sus palabras comienzan con repeticiones de “ab”, seguidas de repeticiones de “c”. Similarmente es posible definir muchos otros lenguajes basados en la idea de repetir esquemas simples. Esta es la idea básica para formar los lenguajes Regulares.

Lenguajes Regulares

•Adicionalmente a las repeticiones de esquemas simples, vamos a considerar que los lenguajes finitos son también regulares por definición. Por ejemplo, el lenguaje L3 = {*anita, lava, la, tina*}, es regular.

•Finalmente, al combinar lenguajes regulares uniéndolos o concatenándolos, también se obtiene un lenguaje regular.

•Por ejemplo, L1 U L3 = {anita, lava, la, tina, ab, abab, ababab, abababab, . . .} es regular. También es regular una concatenación como L3.L3 = {anitaanita, anitalava, anitala, anitatina, lavaanita, lavalava, lavala, lavatina, . . .}

Definición formal de Lenguajes Regulares

Definición.- Un lenguaje *L* es regular si y sólo si se cumple al menos una de las condiciones siguientes:

- *L* es finito;
- *L* es la unión o la concatenación de otros lenguajes regulares *R1* y *R2*, L = R1 U R2 o L = R1R2 respectivamente.
- *L* es la cerradura de Kleene de algún lenguaje regular, L = R*.

Esta definición nos permite construir expresiones en la notación de conjuntos que representan lenguajes regulares.

Expresiones regulares

•Ejemplo.- Sea el lenguaje *L* de palabras formadas por *a y b*, pero que empiezan con *a*, como *aab, ab, a, abaa*, etc. Probar que este lenguaje es regular, y dar una expresión de conjuntos que lo represente.

•Solución.- El alfabeto es = {*a, b*}. El lenguaje *L* puede ser visto como la concatenación de una a con cadenas cualesquiera de a y *b*; ahora bien, éstas últimas son los elementos de {*a, b*}, mientras que el lenguaje que sólo contiene la palabra a es {*a*}. Ambos lenguajes son regulares. Entonces su concatenación es {a}{a, b}*, que también es regular.

Expresiones regulares

- El término alfabeto denota cualquier conjunto finito de símbolos. Los símbolos se toman de un alfabeto finito. Los códigos ASCII y EBCDIC son dos ejemplos de alfabetos de computador. Una cadena es una secuencia finita de símbolos tomados de ese alfabeto. Un lenguaje es un conjunto de cadenas de un alfabeto fijo.
- Las expresiones regulares representan patrones de cadenas de caracteres. Un patrón es una regla que describe el conjunto de lexemas que pueden representar a un determinado componente léxico (Token) en los programas fuente.

Expresiones regulares

- En la Tabla se muestran ejemplos de componentes léxicos con su lexema y la descripción informal del patrón, es decir la regla que describe el conjunto de lexemas. El patrón para el componente léxico **Const** de la Tabla1 es simplemente la cadena sencilla ***Const*** que deletrea la palabra reservada.

COMPONENTE LÉXICO	LEXEMAS DE EJEMPLO	DESCRIPCIÓN INFORMAL DEL PATRÓN
If	if	if
Relación	<, <=, =, <>, >, >=	< o <=o =o <> o > o >=
Id	pi, cuenta, resultado	letra seguida de letras y dígitos
Num	3.1416, 0, 2.2E6	cualquier constante numérica
Const	const	const

Expresiones regulares

- Una expresión regular *r* se encuentra completamente definida mediante el conjunto de cadenas con las que concuerda.

- Este conjunto se denomina lenguaje generado por la expresión regular y se escribe como L(r), aquí lenguaje se utiliza para definir conjunto de cadenas. Una expresión regular *r* también contendrá caracteres del alfabeto; en una expresión regular todos los símbolos indican patrones.

- También una expresión regular *r* puede contener caracteres especiales llamados metacaracteres y por lo general no pueden ser caracteres legales en el alfabeto.

Expresiones regulares

•Una expresión regular básica es un carácter simple del alfabeto. Dado cualquier carácter *a* del alfabeto Σ, indicamos que la expresión regular **a** corresponde al carácter *a* escribiendo

$$L(\mathbf{a})= \{a\}.$$

•Existen dos símbolos que se necesitan en situaciones especiales: la cadena vacía y el conjunto vacío. La cadena vacía no contiene ningún carácter; se utiliza el símbolo ***ξ*** para denotar la cadena vacía y el metacaracter ***ξ*** estableciendo que

$$L(\boldsymbol{\xi})=\{\boldsymbol{\xi}\}.$$

- El conjunto vacío corresponde a la ausencia de cadenas, es decir cuyo lenguaje sea el conjunto vacío, el cual se escribe como { }. Se emplea el símbolo ϕ y se escribe $L(\phi)=\{\ \}$. La diferencia entre { } y {ξ} es que el conjunto { } no contiene ninguna cadena, mientras que el conjunto {ξ} contiene la cadena simple que no se compone de ningún carácter.

- **Definición**.- El significado de una ER es una función L : ER → 2^{Σ^*} (esto es, una función que toma como entrada una expresión regular y entrega como salida un lenguaje), definida de la manera siguiente:

L("Φ") = ; (el conjunto vacío).

L("Λ") = {***ξ***}.

L("δ") = {δ}, δ ∈ Σ.

L("("R"•"S")") = L(R)L(S),R, S ∈ ER

L("("R"+"S")") = L(R) U L(S),R, S ∈ ER

L("("R")*") = L(R)*,R ∈ ER

- Para calcular el significado de una ER en particular, se aplica a ella la función *L*. Las ecuaciones dadas arriba se aplican repetidamente, hasta que el símbolo *L* desaparezca.

- Ejemplo.- El significado de la ER "(((a + b))* • a)" se calcula de la manera siguiente:

L("(((a + b))* • a)") = L("((a + b))*")L("a") -usando 4,

= L("(a + b)")*{a} -por 6 y 3,

= (L("a") U L("b"))*{a} -aplicando 5,

= ({a} U {b}){a} = {a, b}*{a} -usando 3 y simplificando.

Con objeto de hacer la notación menos pesada, vamos a simplificar las ER de la manera siguiente:

- Omitiremos las comillas “ ”.
- Se eliminan los paréntesis innecesarios. Se supone una precedencia de operadores en el orden siguiente: primero “*”, luego “•” y finalmente “+”. Además se supone que los operadores “•” y “+” son asociativos.
- Eventualmente omitiremos el operador “•”, suponiendo que éste se encuentra implícito entre dos sobrexpresiones contiguas.

Operaciones sobre expresiones regulares

•Existen tres operaciones básicas en las expresiones regulares:

1) selección entre alternativas, la cual se indica mediante el metacaracter | (barra vertical);
2) concatenación que se indica mediante yuxtaposición (sin un metacaracter), y
3) repetición o cerradura la cual se indica mediante el metacaracter *

•*Selección entre alternativas.* Si *r* y *s* son expresiones regulares, entonces *r*|*s* es una expresión regular que define cualquier cadena que concuerda con *r* o con *s*. En términos de lenguajes, el lenguaje de r|s es la unión de los lenguajes de r y s, o L(r|s)=L(r) U L(s). Como ejemplo tenemos:

L(a|b|c|d| e)=L(a) U L(b) U L(c) U L(d) U L(e) = {a,b,c,d,e}

Operaciones sobre expresiones regulares

- *Concatenación.* La concatenación es el conjunto de cadenas formado al concatenar todas las cadenas de cada una de ellas. La concatenación de dos expresiones regulares *r* y *s* se escribe como *rs*, y corresponde a cualquier cadena que sea la concatenación de dos cadenas con la primera de ellas correspondiendo a *r* y la segunda a *s*.

- Por ejemplo si tenemos S_1={aa,b} y S_2={a,bb} entonces S_1S_2={aaa,aabb,ba,bbb}

Operaciones sobre expresiones regulares

- *Repetición.* La repetición de una cadena regular se denomina también en ocasiones como cerradura de Kleene, y se escribe ***r****, donde r es una expresión regular.

- La expresión regular ***r****corresponde a cualquier concatenación finita de cadenas, cada una de las cuales corresponde a ***r***.

- Por ejemplo, **a*** corresponde a las cadenas ***ξ***, a, aa, aaa, aaaaa, ... (concuerda con e porque e es la concatenación de ninguna cadena concordante con **a**).

Operaciones sobre expresiones regulares

- En términos matemáticos la cerradura de Kleene se define como:

$$S^* = \bigcup_{n=0}^{\infty} S^n$$

- Donde $S^n = S \ldots S$ es la concatenación de S por n veces ($S^0 = \{\xi\}$)
- Por lo tanto se tiene que:

$$S^* = \{\xi\} \cup S \cup SS \cup SSS \cup \ldots$$

- En una expresión regular la operación de repetición se define como sigue:

$$L(r^*) = L(r)^*$$

Operaciones sobre expresiones regulares

- Como ejemplo la expresión regular (a|bb)* corresponde a cualquiera de las cadenas siguientes: ***ξ***, a, bb, abb, bba, bbbb, aaa, aabb, abba, abbbb, bbaa, ... y así sucesivamente.

EJEMPLOS:

- (0 + 1)*1 = {***1, 01, 001, 1111,00001,000011111000011, …..*** }.
- (ab U cb)*d* ={***ξ, d, ab, cb, abd, cdd, abababddddd, cbcbdddd,…***}

Operaciones sobre expresiones regulares

- *Una o más repeticiones.* Dada una expresión regular r, la repetición de r se describe utilizando la operación de cerradura r^+. Esto permite que r se repita 1 o más veces. Una situación típica es la necesidad de una o más repeticiones en lugar de ninguna, lo que garantiza que aparece por lo menos una cadena correspondiente a r y no permite la cadena vacía ***ξ***.

- Por ejemplo si deseáramos definir números binarios se define como:

$$(0|1)^+$$

Lo cual corresponde a 0, 1, 01, 10, 11, 100, 101,... sin incluir el conjunto ***ξ***. Donde el signo + indica una o más repeticiones.

Operaciones sobre expresiones regulares

- *Cualquier carácter.* Un metacaracter típico que se utiliza para expresar concordancia de cualquier carácter es el punto "."
- Por ejemplo con el metacaracter punto se puede escribir una expresión regular para todas las cadenas que contengan al menos una b como se muestra a continuación:

.*b.*

- *Cualquier carácter que no esté en un conjunto dado.* El metacaracter tilde (~) representa la operación negación lógica la cual excluye un carácter simple del conjunto de caracteres por generar. Por ejemplo podríamos escribir una expresión regular para un carácter en el alfabeto que no sea a, ni b, ni c, como sigue:

~(a|b|c)

Operaciones sobre expresiones regulares

•*Un intervalo de caracteres.* Para escribir un intervalo de caracteres como el de todas las minúsculas o de todos los dígitos se emplean corchetes y un guión.

•Por ejemplo [a-z] representa las letras minúsculas, [0-9] los dígitos. También se pueden incluir intervalos múltiples, de manera que [a-zA-Z] representa todas las letras minúsculas y mayúsculas.

Expresiones regulares y su relación con los AF.

AUTOMATAS FINITOS

- Al describir una máquina de estaos finitos en particular, debemos incluir las informaciones que varían de un autómata a otro; es decir, no tiene sentido incluir descripciones generales aplicables a todo autómata. Estas informaciones son exactamente las que aparecen en un diagrama de estados y transiciones, como los que hemos presentado antes.

Expresiones regulares y su relación con los AF.

AUTOMATAS FINITOS

Definición.- Una maquina de estados finitos M es un quíntuplo (K, Σ, S, F,), donde:

- K es un conjunto de identificadores (símbolos) de estados;
- Σ es el alfabeto de entrada;
- $S \in K$ es el estado inicial;
- $F \subseteq K$ es un conjunto de estados finales;
- $\delta : K \times \Sigma \rightarrow K$ es la función de transición, que a partir de un estado y un símbolo del alfabeto obtiene un nuevo estado.

Expresiones regulares y su relación con los AF.

- **EXPRESIONES REGULARES**

  - **Alfabeto:** denota cualquier conjunto finito de símbolos.
  - **Símbolo:** Representación distinguible de cualquier información.
  - **Cadena:** es una secuencia finita de símbolos tomados del alfabeto.
  - **Lenguaje:** es un conjunto de cadenas de un alfabeto fijo.
  - **Expresiones Regulares:** representan patrones de cadenas de caracteres.
  - **Patrón:** es una regla que describe el conjunto de lexemas que pueden representar a un determinado componente léxico (Token) en los programas.

Expresiones regulares y su relación con los AF.

Ejercicios:

1) **Dar 3 ejemplos de lenguajes basados en el alfabeto {a,b,c,}**

2) **Calcular la concatenacion del lenguaje {**ξ, aba**}, {a,bb, ξ}**

3) **Obtener {a,bb}* (dar los primeros 10 elementos)**

4) **Sea L1 = {ca,ma}, L2{nta,sa} y L3{dor,sola}**

Verificar L1.L2 = L2.L1 ➔ Propiedad Conmutativa

Verificar (L1.L2) L3 = L1(L2.L3) ➔ Propiedad Asociativa

Expresiones regulares y su relación con los AF.

EXPRESIONES REGULARES

Ejemplo:

Sea Σ1 = {a, b, c, ….z}

Sea Σ2 = {0, 1, 2, …9}

$L^1 = \{x | x \in \Sigma^*_1$ y tiene longitud 1}

$L^2 = \{x | x \in \Sigma^*_2$ y tiene longitud 2}

Expresiones regulares y su relación con los AF.

Operación	Descripción	
L1 U L2	Conjunto de las letras a-z y los numero de 0-9.	{a,b,….z, 0,1,….9}
L1.L2	Conjunto de las cadenas que tienen una letra seguida de un numero.	{a0, a1, …. ba, b1….. }
L^*_1	Conjunto de todas las cadenas de letras incluyendo ξ	{ξ, a,b,…aa,.. Bbb…}
L^+_2	Conjunto de todas las cadenas de 1 o mas números.	{0,01, … 001, … 300…}
$L_1(L_1$ u $L_2)$	Conjunto de todas las cadenas que inician con una letra seguido de un numero o una letra	{a1, ab, ar, r1……}
$L_1(L_1$ u $L_2)$ *	Conjunto de todas las cadenas que inician con una letra seguida de una cadena de letras y números repetidos las veces que sean incluyendo el vacio.	{ξ, aaaa1111, b2222….}

Expresiones regulares y su relación con los AF.

Relación entre Lenguajes, Gramáticas y Autómatas

•Lenguajes: Conjuntos de palabras.

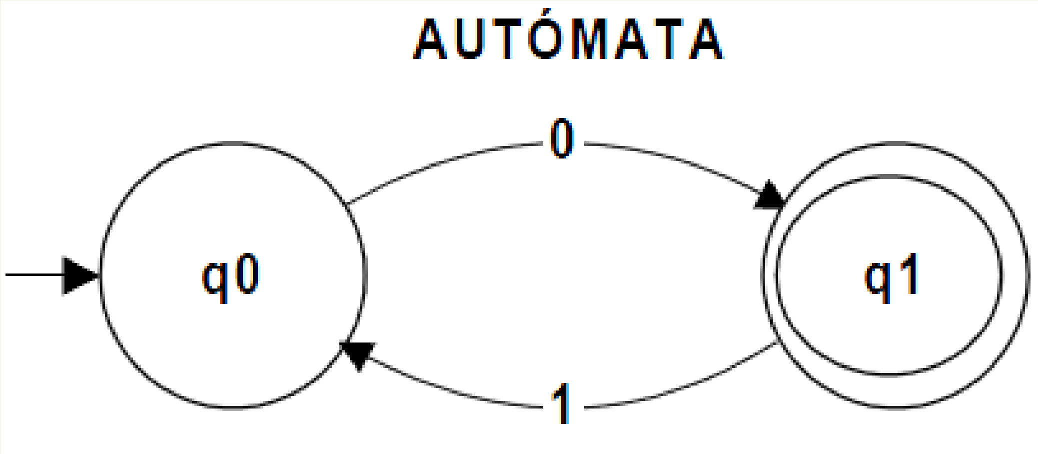
Ej: *L = {0, 010, 01010, ...}*

•Gramáticas: Herramientas para la generación de Lenguajes.

Ej: $S \rightarrow 01S \rightarrow 0101S \rightarrow 01010$

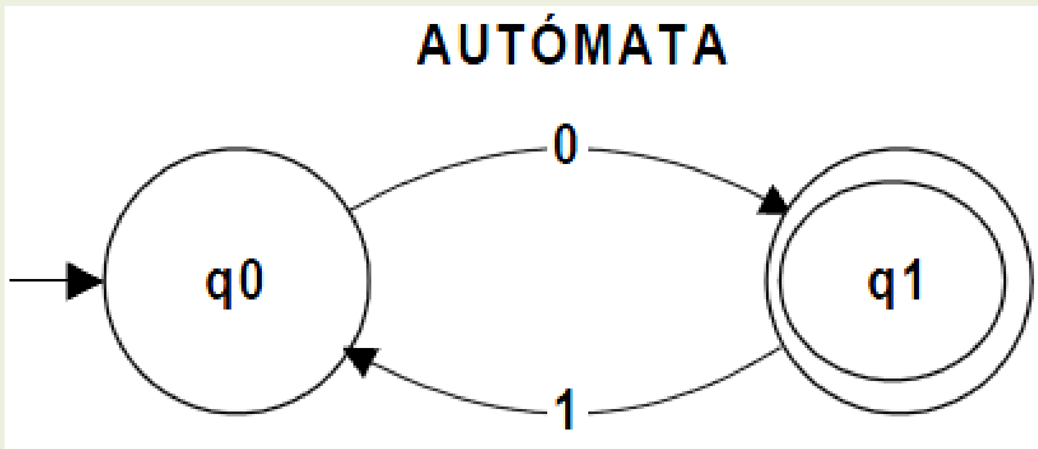
•Autómatas: Herramientas para el reconocimiento de Lenguajes.

$$\text{Ej: } q0 \rightarrow_0 q1 \rightarrow^1 q0 \rightarrow^0 q1 \rightarrow^1 q0 \rightarrow^0 q1$$



Expresiones regulares y su relación con los AF.

Ej: $q0 \rightarrow_0 q1 \rightarrow^1 q0 \rightarrow^0 q1 \rightarrow^1 q0 \rightarrow^0 q1$

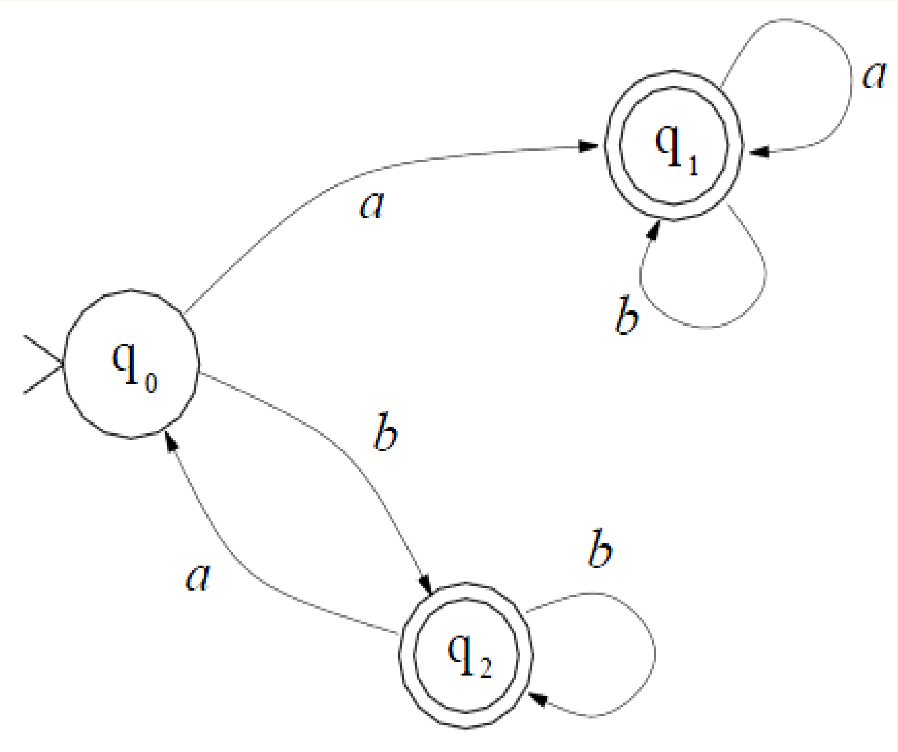


Este autómata reconoce:

L={0, 010, 01010, ...}

2.- AUTOMATAS FINITOS DETERMINISTAS

Ejemplo.- El autómata finito determinista de la figura puede ser expresado formalmente como:



Notación formal:

K = {q0, q1, q2}

Σ = {a, b}

δ = {((q0, a), q1), ((q0, b), q2),
((q1, a), q1), ((q1, b), q1),
((q2, a), q0), ((q2, b), q2)}

F = {q1, q2}

S = {q0}

Expresiones regulares y su relación con los AF.

AUTOMATAS FINITOS DETERMINISTAS

La función de transición ***δ*** puede ser expresada mediante una tabla como la siguiente, para este ejemplo:

$\delta = \{((q0, a), q1), ((q0, b), q2),$
$((q1, a), q1), ((q1, b), q1),$
$((q2, a), q0), ((q2, b), q2)\}$

q	σ	$\delta(q, \sigma)$
q_0	a	q_1
q_0	b	q_2
q_1	a	q_1
q_1	b	q_1
q_2	a	q_0
q_2	b	q_2

Expresiones regulares y su relación con los AF.

AUTOMATAS FINITOS DETERMINISTAS

Palabras aceptadas

Decimos que un AFD reconoce o acepta una palabra si

se cumplen las siguientes condiciones:

1. Se consumen todos los caracteres de dicha palabra de entrada, siguiendo las transiciones y pasando en consecuencia de un estado a otro;

2. Al terminarse la palabra, el estado al que llega es uno de los estados finales del autómata (los que tienen doble círculo en los diagramas, o que son parte del conjunto F en la representación formal).

AUTOMATAS FINITOS DETERMINISTAS

Palabras aceptadas

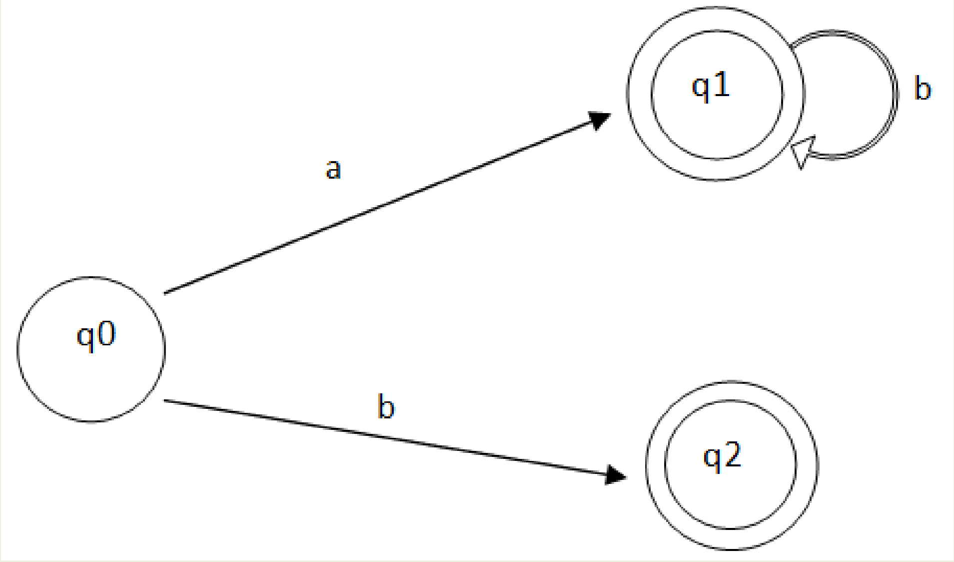
El concepto de ***lenguaje aceptado*** es una simple extensión de aquel de palabra aceptada:

Definición.- El lenguaje aceptado por una máquina ***M*** es el conjunto de palabras aceptadas por dicha máquina.

Expresiones regulares y su relación con los AF.

AUTOMATAS FINITOS DETERMINISTAS

Notación formal:



K = ?

Σ = ?

δ = ?

F = ?

S = ?

Cual es la cadena aceptada por el autómata…?

AUTOMATAS FINITOS NO DETERMINISTAS CON TRANSICIONES ξ Y EQUIVALENCIAS ENTRE ELLOS – Implementación en Java

Requerimientos del programa en java:
Jcreator
Manejo java swing.
Manejo de archivos txt. (el archivo contiene la cadena a analizar).
Mensajes manejados con el JOptionPane.

Programas a entregar.
1) Expresión regular. Por ejemplo: $L=\{a^+ b^+\}$
2) Palabra reservada. Por ejemplo: if, for, else, while, elseif, etc.

Ambos programas se entregaran en laboratorio de forma presencial e individual, el diseño del autómata será en inmediato así como la tabla de transiciones.

BIBLIOGRAFÍA

1.- Aho V. Alfred 2008, “Compiladores; principios, técnicas y herramientas” 2da Edición, Pearson Addison Wesley.

2.- Hopcroft E. Jhon 2008, “Teoría de autómatas lenguajes y computación” 3ra Edición, Ed. Pearson Addison Wesley.

3.- Brena R. 2003 “Autómatas y lenguajes”, Tecnológico de Monterrey.