

AUTÓMATAS Y GRAMÁTICAS

TRABAJO PRÁCTICO N° 1

CADENAS, LENGUAJES Y OPERACIONES CON CADENAS

AUTÓMATAS FINITOS DETERMINISTAS Y NO DETERMINISTAS, ACEPTACIÓN DE CADENAS

Contenido Conceptual

Cadenas, lenguajes y operaciones con los lenguajes.

Autómatas finitos deterministas y no deterministas, aceptación de cadenas.

Objetivos

- Identificar cadenas y lenguajes.
- Aplicar diferentes operaciones a cadenas y lenguajes.
- Identificar y diferenciar autómatas finitos deterministas y no deterministas.

PARTE A

Conceptos fundamentales: cadenas, lenguajes, operaciones con los lenguajes.

DEFINICIONES

Alfabeto: conjunto de símbolos finito y no vacío, por ejemplo, las letras y los caracteres.

Cadena: es una secuencia finita de símbolos seleccionados de algún alfabeto. La longitud de una cadena s , $|s|$, es el número de apariciones de símbolos en s .

Cadena vacía: Representa cero apariciones de símbolos. Esta cadena ϵ es una cadena que puede construirse en cualquier alfabeto.

Operaciones aplicadas a cadenas:

- **Concatenación:** de x e y , xy , es la cadena que resulta de agregar y a x .
- **Potencia:** cuando se considera la concatenación como un “producto”.

AUTÓMATAS Y GRAMÁTICAS

- **Prefijo:** una cadena que se obtiene eliminando cero o más símbolos desde la derecha otra cadena.
- **Sufijo:** una cadena que se forma suprimiendo cero o más símbolos desde la izquierda de otra cadena.
- **Subcadena:** una cadena que se forma suprimiendo un prefijo y un sufijo de otra cadena.

Lenguaje: conjunto de cadenas, todas ellas seleccionadas de un Σ^* , donde Σ es un determinado alfabeto se denomina lenguaje.

Operaciones con Lenguajes:

Sean A y B dos lenguajes se pueden definir las siguientes operaciones:

- **Unión:** $A \cup B = \{ x \mid x \text{ está en } A \text{ o } x \text{ está en } B \}$
- **Intersección:** $A \cap B = \{ x \mid x \text{ está en } A \text{ y } x \text{ está en } B \text{ simultáneamente} \}$
- **Concatenación:** $A.B = \{ w.x \mid w \text{ está en } A \text{ y } x \text{ está en } B \}$
- **Potencia:** Si $A = \{ ab \}$, $A^0 = \epsilon$, $A^1 = A = \{ ab \}$, $A^2 = A.A = \{ abab \}$, $A^3 = A.A.A = \{ ababab \}$
- **Cerradura de Kleene:** Es el lenguaje A compuesto por todas las cadenas sobre un alfabeto dado Σ . Si $\Sigma = \{ 1 \}$, $A^* = \{ \epsilon, 1, 11, 111, 1111, \dots \}$, A^* es infinito.

EJERCICIOS

Ejercicio 1

Dado el alfabeto $\Sigma = \{ a, b, c, d, 0, 1, 2, 3, 4 \}$, obtenga dos cadenas, x e y, del alfabeto dado, indique la longitud de cada cadena y aplique las siguientes operaciones: concatenación de x e y, las potencias x^0 , x^1 , y^2 , y^3 .

Ejemplo:

Si $x = abc123$ e $y = d4$, $|x| = 6$, $xy = abc123 d4$, $x^0 = \epsilon$, $y^3 = d4 d4 d4$

AUTÓMATAS Y GRAMÁTICAS

Ejercicio 2

Dados los siguientes lenguajes, A, el conjunto de letras y B, el conjunto de dígitos, realice las siguientes operaciones $A \cup B$, $A \cap B$, $A.B$, A^3 , B^2 , B^0 , A^* , $A(A \cup B)^*$.

Ejemplo:

- AA es el conjunto de cadenas que consta de una letra seguida de otra letra.
 - A^2 es el conjunto de todas las cadenas de dos letras.
 - B^* es el conjunto de todas las cadenas de números, incluyendo ε , la cadena vacía.
 - $D(L \cup D)^*$ es el conjunto de todas las cadenas de dígitos y letras que comienzan con un dígito.
-

PARTE B

Conceptos fundamentales: autómatas finitos deterministas y no deterministas, aceptación de cadenas.

DEFINICIONES

Diagrama de transiciones representa las acciones que tienen lugar cuando el analizador léxico es llamado por el analizador sintáctico para obtener el siguiente componente léxico.

Construcción del diagrama de transiciones:

- Las *posiciones* en un diagrama de transición se representan con un círculo y se llaman *estados*.
- Los *estados* se conectan mediante flechas, llamadas *aristas*.
- Las *aristas* tienen etiquetas que indican los caracteres de entrada.
- El *estado de inicio* es el estado inicial del diagrama de transición.
- El *estado de aceptación* es el estado en el cual se ha encontrado un token y se indica con un círculo doble.

AUTÓMATAS Y GRAMÁTICAS

Autómatas finitos: o máquinas de estados finitos, son una forma matemática para describir clases particulares de algoritmos (o "máquinas"). Se pueden utilizar para describir el proceso de reconocimiento de patrones en cadenas de entrada, y de este modo construir analizadores léxicos. Se representan a través de **diagramas de transiciones**.

Autómatas finitos determinista AFD: Se limitan a aceptar o no una determinada cadena recibida en la entrada. La salida solo tendrá dos valores posibles aceptar o no aceptar la entrada. Deben tener una transición para cada estado y carácter. Aquellas transiciones que dan errores como resultado, se dejan fuera del diagrama para el AFD. La definición de un AFD no especifica lo que ocurre cuando se presenta un error. No especifica la acción que tomará un programa al alcanzar un estado de aceptación.

Autómatas finitos no determinista AFN: Pueden tener ninguna, una o más transiciones de un estado sobre el mismo símbolo de entrada. Pueden tener transiciones vacías o transiciones ϵ .

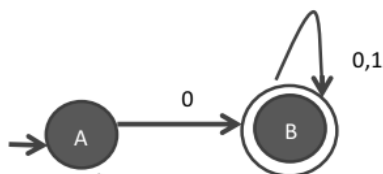
EJERCICIOS

Ejercicio 1:

Obtenga el AFD y la tabla de transición.

Ejemplo:

Dado el siguiente lenguaje definido en el alfabeto $\Sigma = \{0,1\}$. El AFD del conjunto de cadenas que inician en "0" y la tabla de transición son:



Estados	Alfabeto	
	0	1
A	B	
B	B	B

- Del lenguaje definido en el alfabeto $\Sigma = \{0,1\}$, para el conjunto de cadenas que terminan en "1".
- Del lenguaje definido en el alfabeto $\Sigma = \{0,1\}$, para el conjunto de cadenas que contienen a la subcadena "01".

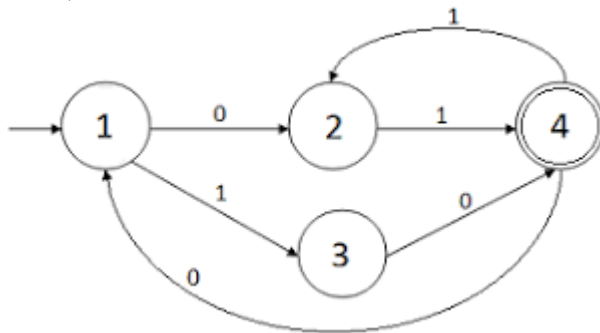
AUTÓMATAS Y GRAMÁTICAS

- c) Del lenguaje definido en el alfabeto $\Sigma=\{a,b,c\}$, para el conjunto de cadenas que inician con la subcadena “ac”.

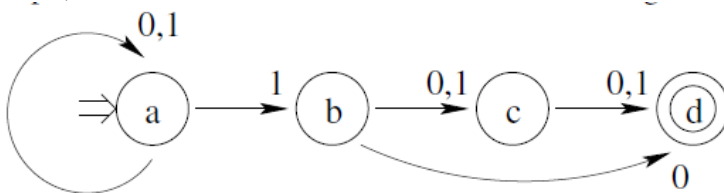
Ejercicio 2:

Indique en cada caso si es un AFD (autómata finito determinista) o un AFN (autómata finito no determinista).

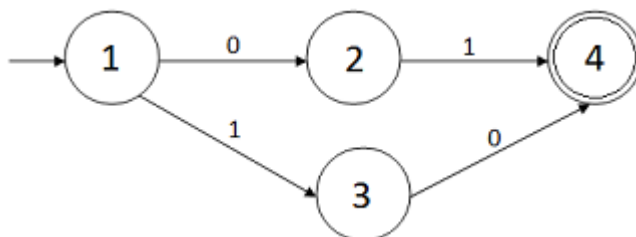
a)



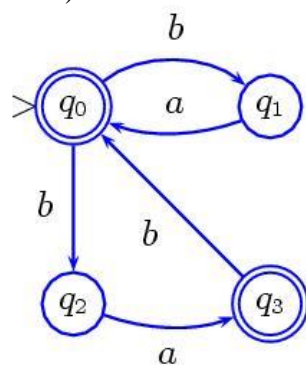
b)



c)

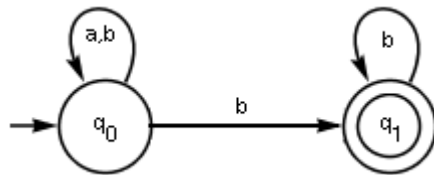


d)



AUTÓMATAS Y GRAMÁTICAS

e)



Ejercicio 3:

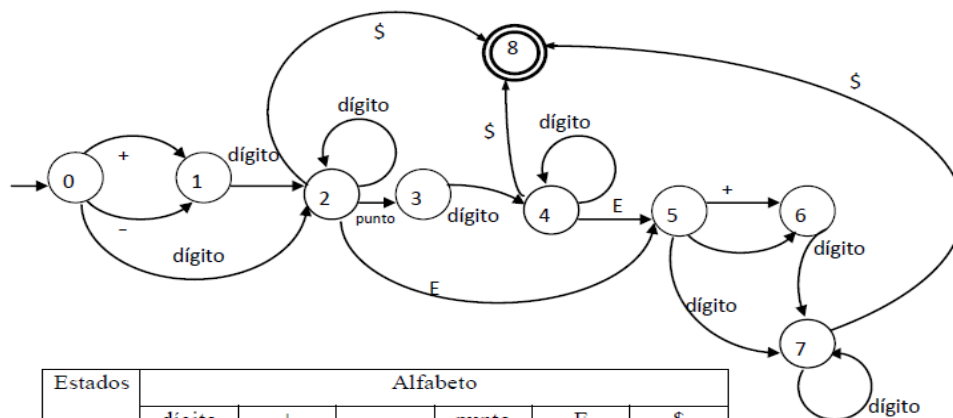
Implementar mediante el lenguaje Python un autómata que reconozca números con signo o sin signo y en formato exponencial, usando una matriz de transición. Para implementar la matriz de transición en Python, se pueden utilizar listas.

tabla = [[2,1,1,',',','],[2,',',',',',',','],...,['acepta','acepta','acepta','acepta','acepta','acepta']]

Ejemplo de implementación en C:

```

estado =0;
while (estado != 8 && estado != error )
{
    ch = siguiente carácter de entrada;
    estado = T[estado][ch];
}
if (estado ==8) aceptar();
else error();
  
```



Estados	Alfabeto					
	dígito	+	-	punto	E	\$
0	2	1	1			
1	2					
2	2			3	5	8
3	4					
4	4				5	8
5	7	6	6			
6	7					
7	7					8
8	acepta					