



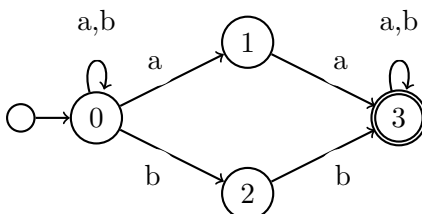
Nombre y Apellido:

Examen Final

1. Autómatas de Estado Finito (AEF) y Lenguajes formales Regulares

Los autómatas finitos son una herramienta que permite poner en término de estados y transiciones sistemas que de otra manera serían muy difíciles de sintetizar. Una de sus principales aplicaciones reside en que tienen una correspondencia uno a uno con los lenguajes formales regulares.

Por ejemplo,



El autómata presentado está relacionado con el lenguaje de todas las palabras que se pueden formar con el alfabeto $\{a,b\}$ que contengan como subcadena 'aa' o 'bb'. Para formar todas las palabras del lenguaje asociado al autómata es necesario

- partir del estado inicial. El nodo inicial se identifica porque es apuntado por una transición o flecha sin etiqueta. En el ejemplo, 0.
- recorrer un camino cuyas transiciones estén etiquetadas con los caracteres de la palabra.
- terminar el recorrido en un nodo de aceptación. Los nodos de aceptación se identifican por su trazo doble. En el ejemplo 3 es el único estado de aceptación.

Algunas palabras válidas del lenguaje son:

- 'abaab' al recorrer el camino de nodos 000133
- 'aa' al recorrer el camino de nodos 013
- 'abbaaa' al recorrer el camino de nodos 0023333

pero 'bab' no es una palabra aceptada por el autómata ya que no existe camino desde el nodo inicial a un nodo de aceptación cuyas transiciones estén etiquetadas con 'b', 'a', 'b'.

1.1. Autómatas de Estado Finito No Determinista - AEFND

El autómata del ejemplo tiene la particularidad de que desde el nodo 0 es posible transicionar con etiqueta a al nodo 0 pero también es posible transicionar con etiqueta a al nodo 1. A este tipo de autómatas se los llama **no deterministas** porque existe más de una transición saliente del mismo nodo con la misma etiqueta.

2. Representación de un autómata

Una forma de representar de manera compacta un autómata consiste en hacer una tabla cuyas filas y columnas sean el conjunto de nodos y en cada componente (i,j) de la tabla se almacenen las etiquetas de las transiciones de origen i y destino j . A esta forma de representación se la llama Matriz de Adyacencia.

La matriz de adyacencia del autómata ejemplo es la siguiente.

	0	1	2	3
0	a,b	a	b	-
1	-	-	-	a
2	-	-	-	b
3	-	-	-	a,b

La primera fila indica que existen transiciones de 0 a 0 etiquetadas con a y b respectivamente, que 0 transiciona a 1 con etiqueta a y que también lo hace a 2 con etiqueta b .

3. Objetivos

El objetivo de este final es que se elijan una o varias ED adecuadas para representar un autómata de estados finito de alfabeto $\{a,b\}$. Además, se deberá poder determinar de manera eficiente cuáles palabras son aceptadas por el autómata, o lo que es lo mismo decir, cuáles palabras pertenecen al lenguaje inducido por el autómata.

3.1. Detalles de la implementación

En concreto se pide que

- Se elijan una o varias ED para representar un AEF de alfabeto $\{a,b\}$ y se indique en forma de comentario una justificación acotada de la elección.
- Se defina una función que dado un AEF, un nodo origen A , un nodo destino B y una etiqueta x , permita agregar una transición de A a B etiquetada con x al AEF.
- Se defina una función que dada una palabra y un AEF determine si la palabra es aceptada por el autómata.

3.2. Formato de entrada

A continuación se encuentra el archivo de entrada asociado al autómata ejemplo.

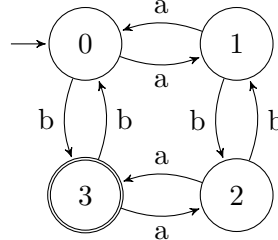
```
4
a,b;a;b;
;;;a
;;;b
;;;a,b
3
```

La primera línea indica la cantidad de nodos (4) de nombres 0,1,2 y 3. Desde la línea 2 hasta la 5 inclusive se encuentra la matriz de adyacencia del AEF en donde las casillas están separadas por ';' y las etiquetas que son transición del mismo par (origen,destino) están separadas por ','. Por último, el único nodo de aceptación es el 3, último nodo representado en la matriz de adyacencia.

Importante: se asume que el nodo inicial es el primero de los nodos del autómata. En la matriz de adyacencia su información está en la primera fila y primera columna.

4. Otros ejemplos

- Autómata que induce el lenguaje de todas palabras formadas por el alfabeto $\{a,b\}$ con **número impar de 'b'** y **número par de 'a'**.



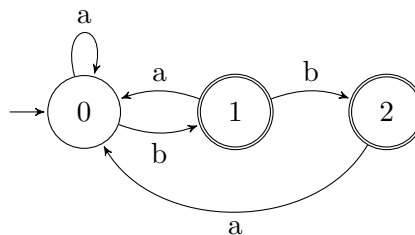
Entrada asociada

```

4
;a;;b
a;;b;
;b;;a
b;;a;
3
  
```

El autómata acepta las palabras 'b', 'bbb', 'babab' y 'baa' pero no acepta 'ba', 'a' y 'abab'.

- Autómata que induce el lenguaje de todas palabras formadas por el alfabeto $\{a,b\}$ con **a lo sumo 2 'b' consecutivas** y que **no terminen en 'a'**.



Entrada asociada

```

3
a;b;
a;;b
a;;
1
2
  
```

El autómata acepta las palabras 'abaab', 'abbab' y 'b' pero no acepta 'abbb' y 'abba'.

5. Evaluación

Para la evaluación se tomarán en cuenta los siguientes elementos:

- Estructuras de datos usadas.
- Algoritmos propuestos y Eficiencia.
- Calidad del código entregado.