

Trabajo Práctico Nº 2

- Especificación de estados y transiciones
- Introducción a los Autómatas Finitos.
- Lenguajes regulares

NOTA: Los ejercicios marcados con * son de resolución y entrega optativa. Los ejercicios marcados con Δ tienen un nivel de dificultad similar a la del parcial.

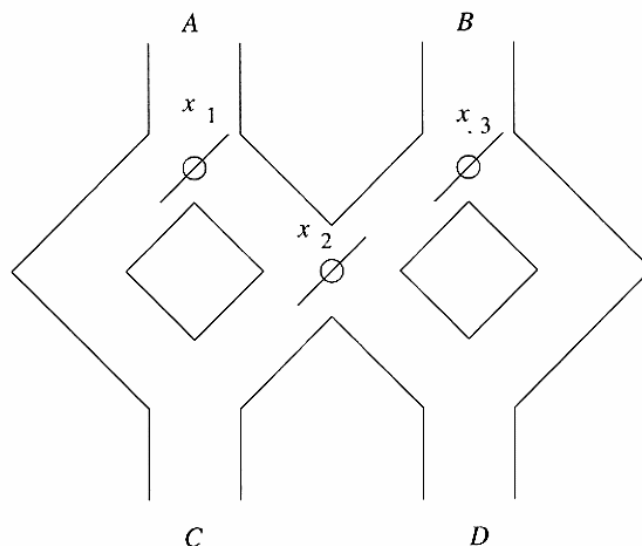
ENTREGA CÓDIGO FUENTE: Se debe adjuntar **UN UNICO** archivo .py en donde se implementan todos los ejercicios que requieren programación. La sección de código correspondiente a cada ejercicio debe destacarse con un comentario refiriendo al ejercicio que resuelve (si todo el código está en la sección main), o bien llamarse desde *main* (si cada ejercicio se resuelve por separado como una función). Se debe controlar

PREVIA ENTREGA del TP que el código es interpretado correctamente y no arroja errores de interpretación.

Ejercicios

1. Para los siguientes enunciados, grafique el diagrama de estados y transiciones correspondientes.

a. Δ **Parte 1:** Considere el juego que se muestra a continuación. Si dejamos caer una bolita en A o B, los niveladores x_1 , x_2 , x_3 hacen que la misma caiga a la izquierda o derecha. Cuando una bolita choca con un nivelador, hace que el mismo cambie de estado, de tal modo que la siguiente bolita que choque con él tomará la rama opuesta.



Modele el juego utilizando un autómata. Represente una bolita ingresando en A con 0 y en B con 1. Una secuencia de entrada es aceptada si la última bolita sale en D.

Parte 2: a) Empleando el lenguaje de programación Python y la librería automata-lib valide computacionalmente el diseño del autómata de la Parte 1.

b) Verifique computacionalmente el comportamiento del modelo diseñado en la Parte 1 probando un conjunto de cadenas de distintas longitudes generadas aleatoriamente por el programa.

* b. Plantee un diagrama de estados y transiciones que represente el proceso de asignación de puntos en un game de tenis, y al mismo tiempo, permita determinar cuál de los jugadores gana dicho game, y por qué puntaje.

c Δ. Imagine que usted está cursando una materia que como requisito de regularización incluye la aprobación de tres parciales, con la posibilidad de recuperar sólo dos de los tres en caso de no aprobarlos. Tenga en cuenta que los exámenes se aprueban con una nota igual o mayor a 4. Defina un diagrama de estados y transiciones que permita determinar si un alumno recursa o bien regulariza la materia (La situación normal es que regulariza) Tenga en cuenta que en el diagrama deben quedar especificadas claramente todas las instancias de evaluación, y qué recuperatorios debe realizar el alumno.

d. Δ Especifique el diagrama de transición de estados que represente el comportamiento de una máquina expendedora que vende café y gaseosas. El cliente debe ingresar primero el dinero a través de monedas, y luego realiza la elección del producto. Posteriormente, la máquina entrega el producto seleccionado o muestra un mensaje de error si el valor del producto excede el total de dinero ingresado. El costo del café chico es de \$1, el café mediano \$1,5 y el de las gaseosas \$2. La máquina acepta monedas de \$0,5 y \$1, sólo da vueltos con monedas de \$0,50. La máquina retorna automáticamente lo que se ingrese por encima de \$2. El sistema deberá devolver exactamente el dinero ingresado (si así lo desea el usuario a través de una operación de Cancelación) o el vuelto de lo que este ingresó, si el monto total supera el valor del producto entregado (20 pts.).

Tenga en cuenta lo siguiente:

- Suponga que la máquina dispone de una cantidad infinita de monedas de \$ 0,50 para dar vuelto.

- El diagrama planteado debe especificar claramente las operaciones realizadas y los resultados de las mismas. No se considerarán correctos nombres de estados y transiciones genéricos como “devolver \$1”, “devolver dinero restante”, “ingreso de dinero”, “devolver 2 monedas” etc. Recuerde que, por ejemplo, la devolución de \$1 implica dos operaciones distintas de devolución de monedas de \$0,50 que se producen automáticamente. En ese sentido, la cantidad de operaciones efectuadas en cada caso debe ser consignada claramente a partir de los estados y transiciones especificados.

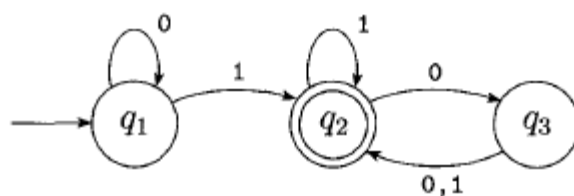
2. Para los autómatas 1, 2, 3, 4,5 y 6 establezca:

- a. La definición matemática del mismo.

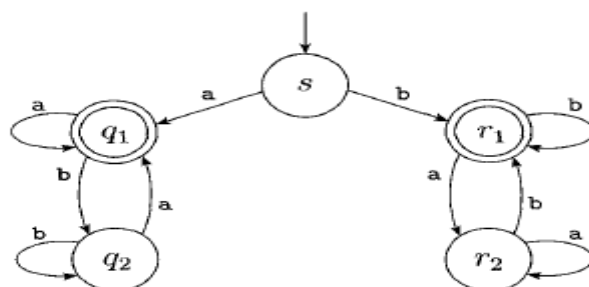
- b. ¿Qué lenguaje regular reconoce? De 5 ejemplos de Strings pertenecientes al lenguaje.

3. Explique con sus palabras, para el autómata 3, que operación realiza el mismo, y cómo la lleva a cabo.

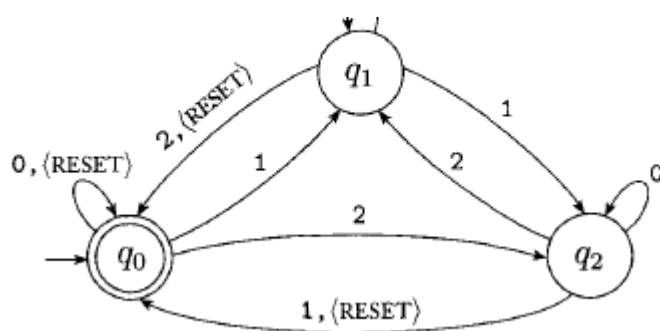
Autómata 1



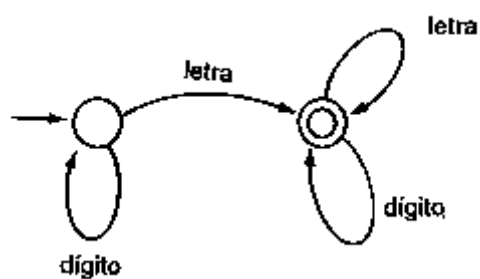
Autómata 2



Autómata 3 Δ



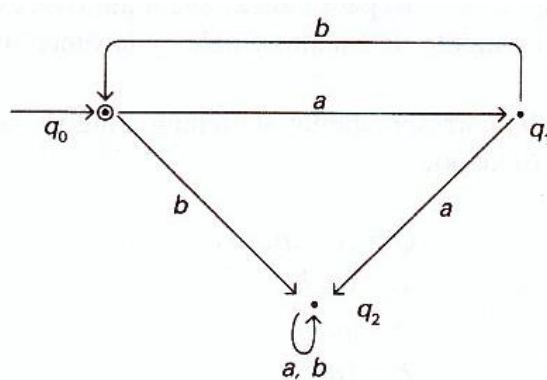
Autómata 4 *



Autómata 5 *



Autómata 6



* 4. Diseñe el autómata que representa de manera abstracta un switch on/off.

5. **Parte 1:** Diseñe y defina formalmente autómatas finitos que reconozcan los siguientes patrones en cadenas que se ingresan.

- a. La cadena contiene 1101 $\Sigma = \{0,1\}$
- b. La cadena contiene 00100 $\Sigma = \{0,1\}$
- c. Δ La cadena contiene aazc $\Sigma = \{a,z,c\}$
- d. * La cadena termina con 00 $\Sigma = \{0,1\}$
- e. * La cadena empieza con abc $\Sigma = \{a,b,c\}$
- f. Δ La cadena empieza con 01 y termina con 101 $\Sigma = \{0,1\}$
- g. La cadena no contiene aba $\Sigma = \{a,b\}$
- h. La cadena tiene un uno en la décima posición $\Sigma = \{0,1\}$
- i. La cadena empieza o termina con 01 $\Sigma = \{0,1\}$
- j. La cadena contiene un número par de ceros $\Sigma = \{0,1\}$
- k. * Δ El conjunto de las cadenas cuyo número de ceros sea divisible por 2 y el de unos divisible por 3 $\Sigma = \{0,1\}$
- l. Δ Las cadenas en las cuáles no hay ningún par de ceros consecutivos en cualquier posición a la derecha, después de un par de unos consecutivos $\Sigma = \{0,1\}$
- m. Δ Las cadenas que contienen a lo sumo un par de unos consecutivos $\Sigma = \{0,1\}$
- n. Las cadenas del lenguaje que tienen un número impar de ocurrencias de la subcadena ab. $\Sigma = \{a,b\}$

Parte 2: a) Empleando el lenguaje de programación Python y la librería automata-lib, valide computacionalmente el diseño de los autómatas de la Parte 1.

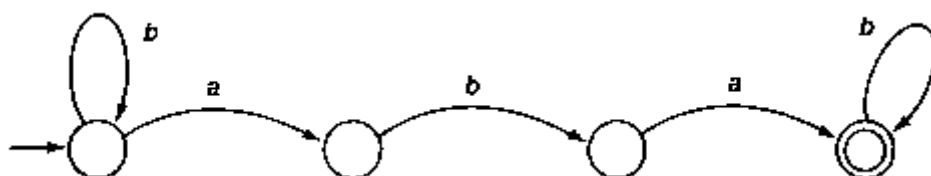
b) Verifique computacionalmente el comportamiento de los autómatas diseñados en la Parte 1 probando cadenas de distintas longitudes generadas aleatoriamente por el programa.

c) Verifique computacionalmente que los autómatas implementados en a) son los autómatas mínimos.

6. Δ Diseñar un autómata que reconozca el conjunto de todos los Strings que comiencen con 1 tales que, interpretándolos como números enteros binarios, sean múltiplos de 5. $\Sigma = \{0,1\}$.

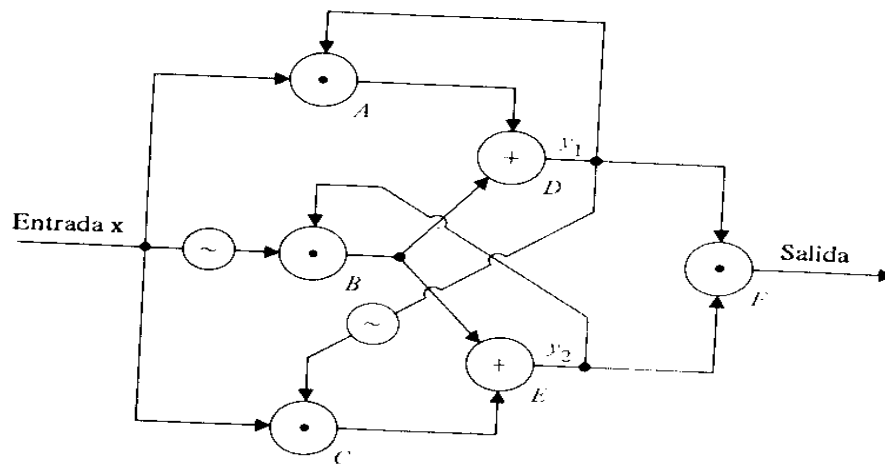
SUGERENCIA: No buscar patrones en las cadenas para identificar múltiplos porque no existen.

7. Modifique el siguiente diagrama de transiciones, para que esté completamente definido como AFD y acepte las mismas cadenas que antes.



* 8. Identifique tres sistemas que encuentre en la vida real, y diseñe un autómata que represente el funcionamiento de los mismos. Especifique el diagrama de transición de estados para cada uno.

9. **Δ Parte A:** Plantee un AFD cuyo comportamiento corresponda al circuito que se muestra en la figura, donde los estados finales correspondan a una salida=1 (La entrada x corresponde a una cadena formada por 0's y 1's). El círculo con un punto interno representa una compuerta And, y el círculo con un + en su interior representa una compuerta Or. El círculo con ~ representa un inversor. Suponga que inicialmente el valor en $y_1=y_2=0$. Asuma que el circuito proporciona la salida correspondiente, y los valores se propagan antes de realizarse la lectura del siguiente carácter.



Parte B: a) Empleando el lenguaje de programación Python y la librería automata-lib, valide computacionalmente el diseño del autómata de la Parte 1.

b) Verifique computacionalmente el comportamiento del autómata diseñado en la Parte 1 probando cadenas de distintas longitudes generadas aleatoriamente por el programa.

c) Verifique computacionalmente que el autómata implementado en a) es el autómata mínimo.