3º curso / 1º cuatr.
Grado Ing. Inform.
Doble Grado Ing.
Inform. y Mat.

Modelos de Computación (MC)
Entrega de ejercicios.
Tema 2

Estudiante (nombre y apellidos): Miguel Lentisco Ballesteros
Grupo de prácticas: A1
Fecha de entrega: 13/10/2017

Ejercicios Tema 2

Ejercicio 1. Diseña un autómata finito determinista que reconozca el siguiente lenguaje:

$$L_3 = \{u \in \{0,1\}^* : \text{el número de 1's no es múltiplo de 3 y el número de 0's es par}\}$$

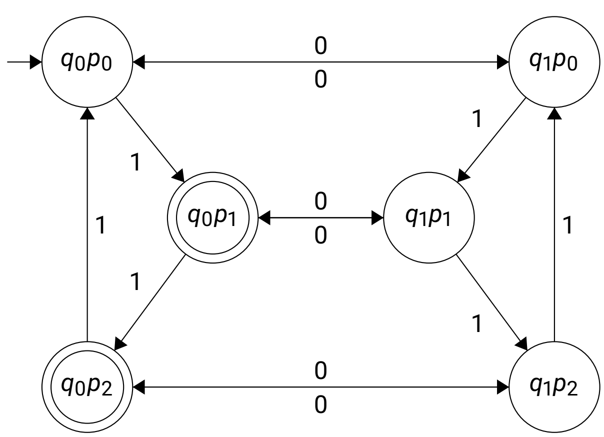
Respuesta. Vamos a tener en cuenta lo siguiente:

1. Para el 0 tendremos 2 estados por separado: la palabra tiene un número de 0's par; y la palabra tiene un número de 0's impar; q_o y q_1, respectivamente.

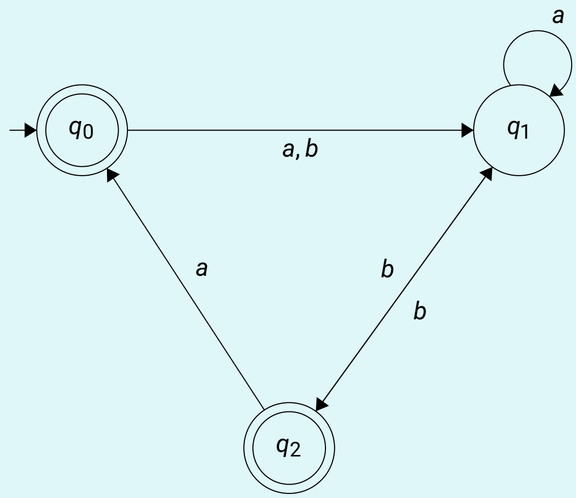
2. Para el 1 tendremos 3 estados: la palabra es múltiplo de 3, la palabra no es múltiplo de 3 (1) y la palabra no es múltiplo de 3 (2); p_0, p_1 y p_2 respectivamente.

Bien, como son estados independientes (0 de 1), si los juntamos tendremos $3 * 2 = 6$ estados diferentes, los cuales serían: $q_0p_0, q_1p_0, q_0p_1, q_1p_1, q_0p_2$ y q_1p_2. Entonces, si en un estado aparece q_1 o p_0 entonces *NO* es válido, dejándonos que los estados finales son q_0p_1 y q_0p_2.

El autómata determinista finito sería de la siguiente forma (empezamos en q_0p_0 puesto que ningun 0 es par (0) y ningún 1 es múltiplo de 3 (0 múltiplo de todos):



Ejercicio 2. Dar una expresión regular para el lenguaje aceptado por el siguiente autómata:



Respuesta. Comento que este ejercicio lo hice a ojo antes de saber que había un método para hacer esto, así que explicaré como he llegado a la respuesta. Tenemos varias posibilidades para aceptar palabras:

1. Solo $q_0 \equiv \epsilon$
2. $q_0 \to q_1 \to q_2$
3. $q_0 \to q_1 \to q_2 \to q_0$
4. $q_0 \to q_1 \to q_2 \to q_0 \to \cdots \to q_2 \equiv (3)^*2$
5. $q_0 \to q_1 \to q_2 \to q_0 \to \cdots \to q_0 \equiv (3)^*$

Está claro que si encontramos una expresión regular para 3, podemos poner $*$ para que se repita todas las veces que quiera y obtener 1 y 5 también. Ahora bien, la expresión para 2 debe ponerse al final de esta solución para 3, ya que después de dar muchas vueltas podemos acabar en q_2 y unirla con ϵ para el caso 1, obteniendo el caso 4.

Entonces veamos lo siguiente:

- Caso 2: $(a + b)a^*b(ba^*b)^*$
- Caso 3,5: $((a + b)a^*b(ba^*b)^*a)^*$

El resultado final para todos los casos es:

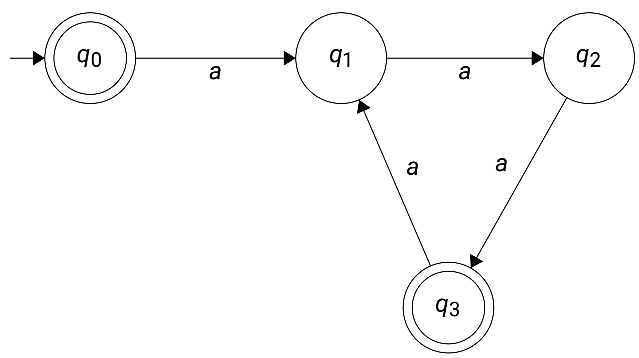
$$((a + b)a^*b(ba^*b)^*a)^*((a + b)a^*b(ba^*b)^* + \epsilon)$$

Ejercicio 3. Sea $B_n = \{a^k$: k es múltiplo de n$\}$ demostrar que B_n es regular para todo n.

Respuesta. La demostración es sencilla, vemos que si un k es múltiplo de n podemos decir que $k = mn$, $m \in \mathbb{N}$, luego podemos reescribir $B_n = \{a^{kn} : \forall k \in \mathbb{N}\} = \{a^0 = \epsilon, a^n, a^{2n} = a^n a^n, \cdots\}$.

Entonces, podemos construir un autómata que detecte la palabra a mediante los estados $q_0, q_1 \cdots q_n$ con estado inicial q_0 y estado final (para la palabra vacía); y el otro estado final el q_n. De cada estado a otro estado leemos a y de q_n volvemos a q_1.

Por ejemplo para n = 3:



Está claro que podemos formar una expresión regular de la forma: $(aaa)^*$ y en general será de la expresión:

$$(a^n)^* = (a \cdots a)^*$$

Luego efectivamente B_n es regular $\forall n \in \mathbb{N}$.

Ejercicio 4. Decimos que u es un prefijo de v si existe w tal que $uw = v$. Decimos que u es un prefijo propio de v si además $u \neq v$ y $u \neq \epsilon$. Demostrar que si L es regular, también lo son los lenguajes:

1. $NOPREFIJO(L) = \{u \in L$: ningún prefijo propio de u pertence a L$\}$
2. $NOEXTENSION(L) = \{u \in L$: u no es un prefijo propio de ninguna palabra de L$\}$

Respuesta.

1. $PREFIJO(L) = \{u \in A^*$: algún prefijo propio de u pertenece a L$\}$, es regular por ser concatenación de lenguajes regulares $PREFIJO(L) = LA^+$. Como los lenguajes regulares son cerrados para la intersección y el complementario, $NOPREFIJO(L)$ es regular porque: $NOPREFIJO(L) = \overline{PREFIJO(L)} \cap L$
2. Montamos un autómata que reconozca los prefijos de L, que sean los estados finales, entonces tenemos que el complementario este autómata (el complementario es regular) forma todas las palabras que no son prefijos propios de ninguna palabra de L, que es lo que queríamos demostrar (y es regular).

Ejercicio 5. Si $L \subset A*$, define la relación $\equiv$ en A^* como sigue: si $u, v \in A^*$, entonces $u \equiv v$ si y solo si para toda $z \in A^*$, tenemos que ($xz \in L \iff yz \in L$).

1. Demostrar que $\equiv$ es una relación de equivalencia.
2. Calcular las clases de equivalencia de $L = \{a^i b^i : i \geq 0\}$
3. Calcular las clases de equivalencia de $L = \{a^i b^j : i, j \geq 0\}$
4. Demostrar que L es aceptado por un autómata finito determinístico si y solo si el número de clases de equivalencia es finito.
5. ¿Qué relación existe entre el número de clases de equivalencia y el autómata finito minimal que acepta L?

Respuesta.

1. Tenemos que ver que $\equiv$ cumple las propiedades de reflexión, simetría y transitividad:
   - Reflexiva: Si se cumple que $\forall z \in A^*$, $xz \in L$ obviamente $xz \in L \iff xz \in L$ luego $x \equiv x$.
   - Simétrica: Si $x \equiv y$ se cumple $\forall z \in A^*$, $xz \in L \iff yz \in L$, y por tanto $yz \in L \iff xz \in L$, luego $y \equiv x$.
   - Transitiva: Si $x \equiv y$ e $y \equiv z$ entonces tenemos que $\forall a \in A^*$, $xa \in L \iff ya \in L \iff za \in L$, en particular $xa \in L \iff za \in L$, y por tanto $x \equiv z$.

2. Sabemos que por definición si $u \notin Cab(L) \Rightarrow \forall z \in A^* : uz \notin L$ por tanto, cualquiera de las palabras contenidas en $A \setminus Cab(L)$ forman una clase de equivalencia (si empieza por b, o hay más b que a).

   Ahora veamos que $Cab(L) = \{a^i b^j : i \geq j \geq 0\}$, luego si dos palabras de $Cab(L)$ están relacionadas es que le añado la misma terminación, por tanto tenemos que añadir el mismo número de b a las dos palabras, y para obtener una palabra de L a partir de $Cab(L)$ tengo que añadir b^{i-j} al final de ambas. Por tanto tenemos que $\forall u, v \in Cab(L)$:

   $$u = a^i b^j \equiv v = a^n b^m \iff i - j = n - m$$

   Concluyendo así que si ponemos $b^{i-j} = b^n, \forall n \in \mathbb{N}$ para cada n tenemos una clase de equivalencia.

3. Por lo mismo de arriba, una clase de equivalencia sería $A^* \setminus Cab(L)$
   Ahora bien, si pongo cualquier secuencias de a, obviamente puedo poner a continuación lo que me de la gana y seguiría estando en L (si pusiera otra cosa de L entonces no estaría en L), luego este conjunto $a^n : n \geq 0$ forman una clase de equivalencia.

   De la misma forma, cualquier palabra de L, que puede ser cualquier número de a seguido de cualquier número de b, podemos ponerle la misma secuencia de b, que como no hay ninguna restricción, pues sigue estando en L, por tanto todo L forma una clase de equivalencia.

   Efectivamente, las palabras que quedaban: $b^j a^i$ con $i, j \geq 1$ forman parte de $A^* \setminus Cab(L)$ luego ya tenemos cubiertas todas las posibilidades.

4. Si L es aceptado por un AFD, entonces como $\forall z \in A^*$ se cumple $az \in L \equiv yz \in L$ entonces para cualquiera z tenemos que la transición al añadir el mismo sufijo deben llegar al mismo estado (dando por supuesto que u y v deben llegar al mismo estado). Por tanto el número de clases es igual al número de estados (finito).

   De la misma manera, construimos un autómata que los estados finales sean los estados a los que se llegan al añadir el mismo sufijo a todas las clases de equivalencia, por tanto haciendo un AFD que admite L.

5. Obviamente de arriba, vemos que el autómata finito minimal que acepta L tendrá tantos estados como clases de equivalencia, podríamos reducir las clases de equivalencia al número mínimo de estados finitos del autómata finito minimal que acepta L.