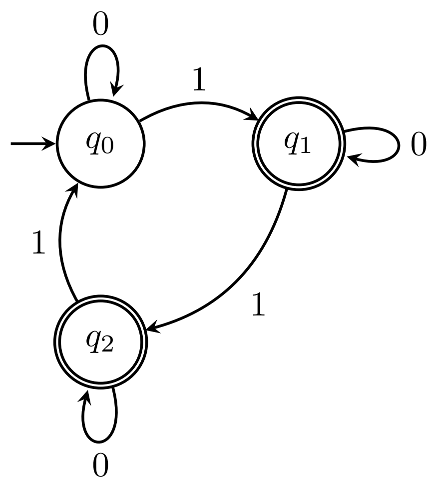
Modelos de Computación: Relación de problemas 1

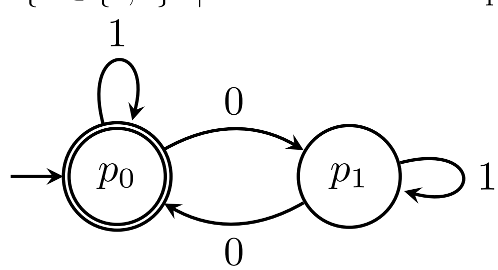
David Cabezas Berrido

Ejercicio 17. Autómata finito determinista que reconoce el lenguaje

$$L_1 = \{u \in \{0,1\}^* \mid \text{el número de 1's no es múltiplo de 3}\}$$

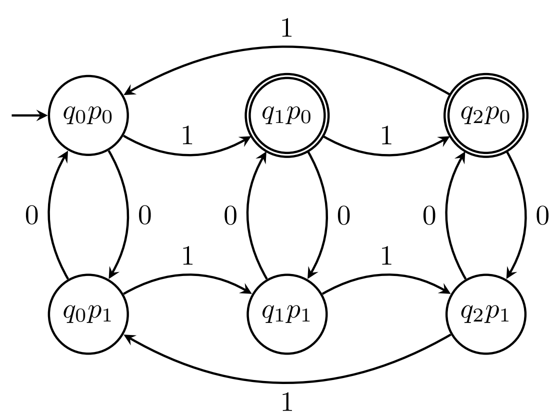


$$L_2 = \{u \in \{0,1\}^* \mid \text{el número de 0's es par}\}$$

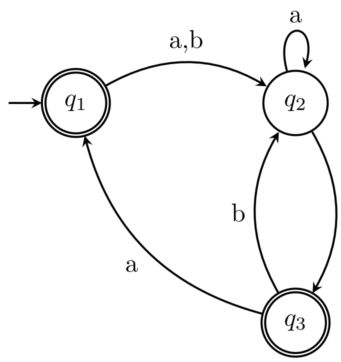


Ahora sólo tenemos que intersecar los dos autómatas para formar el deseado, el que reconoce el lenguaje

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



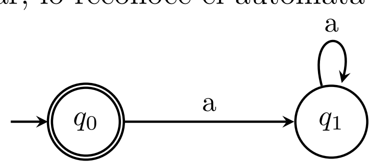
Ejercicio 22. Para hallar la expresión regular que representa el lenguaje aceptado por el autómata



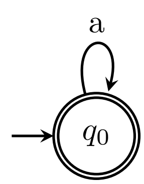
usaremos el algoritmo visto en clase (sólo mostraré un par de iteraciones)

$$
\begin{gathered}
r_{11}^3 + r_{13}^3 \\
r_{11}^2 + r_{13}^2(r_{33}^2)^* r_{31}^2 + r_{13}^2 + r_{13}^2(r_{33}^2)^* r_{33}^2 \\
\varepsilon + (a+b)a^*b(a(a+b)a^*b)^*a + (a+b)a^*b(a(a+b)a^*b)^* \\
\varepsilon + (a+b)a^*b(a(a+b)a^*b)^*(a+\varepsilon)
\end{gathered}
$$

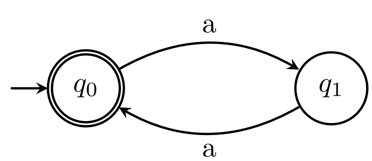
Ejercicio 23. Para probar que $B_n = \{a^k \mid \text{k es múltiplo de n}\} = \{a^{kn} \mid k \in \mathbb{N}\}$ es regular para todo n, construiremos un autómata finito determinista que lo reconozca.
$B_0 = \{\varepsilon\}$ es trivialmente regular, lo reconoce el autómata



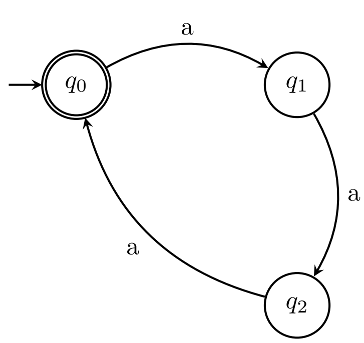
$B_1 = \{a\}^*$ es reconocido por



$B_2 =$ palabras sobre $\{a\}^*$ con número par de a's es reconocido por



B_3 por



De esta forma, $M_n = (\{q_0, \ldots, q_{n-1}\}, \{a\}, q_0, \delta_n, \{q_0\})$ con
$\delta_n(q_i, a) = q_{(i+1)\,\%n}$ $\forall i = 0, \ldots, n-1$ es un autómata finito determinista que reconoce el lenguaje B_n $\forall n \geq 2$.

También puede razonarse que $B_n = (a_1 a_2 \ldots a_n)*$ donde $a_i = a$ $\forall i$.

Ejercicio 24. Dado un lenguaje regular L, probaremos que los siguientes lenguajes son regulares

a)$NOPREFIJO(L) = \{u \in L \mid$ ningún prefijo propio de u pertenece a $L\}$

Buscamos eliminar las palabras del lenguaje que tienen prefijos propios que pertenecen al lenguaje, podemos conseguir esto con tan sólo hacer inaccesibles algunos de los estados finales del autómata.
Supongamos que $M = (Q, A, q_0, \delta, F)$ es un autómata finito determinista que reconoce a L. Construimos $M' = (Q \cup \{p\}, A, q_0, \delta', F)$ y definimos δ' de la siguiente forma

$$\begin{aligned}
\delta'(f,a) =& p \quad \forall f \in F,\ \forall a \in A \\
\delta'(p,a) =& p \quad \forall a \in A \\
\delta'(q,a) =& \delta(q,a) \quad \text{en el resto de casos}
\end{aligned}$$

Si u es aceptada por M', significa que $\delta'(q_0, u) \in F$, además no hemos pasado por ningún estado final en medio (de haberlo hecho acabaría ciclando en p), por lo que siempre hemos aplicado la definición del resto de casos, esto significa que ningún prefijo propio de u es aceptado por M' y que todas las palabras aceptadas por M' son de L.

Resumiendo, hemos encontrado un autómata finito determinista que reconoce $NOPREFIJO(L)$, luego es regular.

b)$NOEXTENSION(L) = \{u \in L \mid u$ no es prefijo propio de ninguna palabra de $L\}$

Si un estado es final pero leyendo más símbolos se puede llegar a otro (o el mismo) estado final, se deben eliminar del lenguaje las palabras que lleven a ese estado. Esto lo conseguimos eliminando estados finales.
Supongamos que $M = (Q, A, q_0, \delta, F)$ es un autómata finito determinista que reconoce a L. Construimos $M' = (Q, A, q_0, \delta, F')$ donde

$$F' = \{q \in F \mid \nexists u \in A^*,\ \nexists f \in F : (q,u) \vdash^* (f, \varepsilon)\}$$

De esta forma, M' sólo acepta palabra de L. Además, de toda palabra u aceptada por M', sabemos que no existe otra palabra $v \neq \varepsilon$ tal que $uv \in L$ justo como queríamos.

Resumiendo, hemos encontrado un autómata finito determinista que reconoce $NOEXTENSION(L)$, luego es regular.