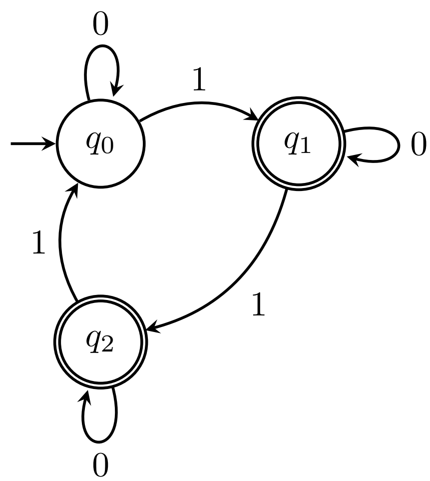
Modelos de Computación: Relación de problemas 1

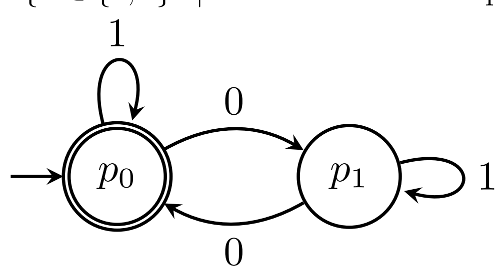
David Cabezas Berrido

Ejercicio 17. Autómata finito determinista que reconoce el lenguaje

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

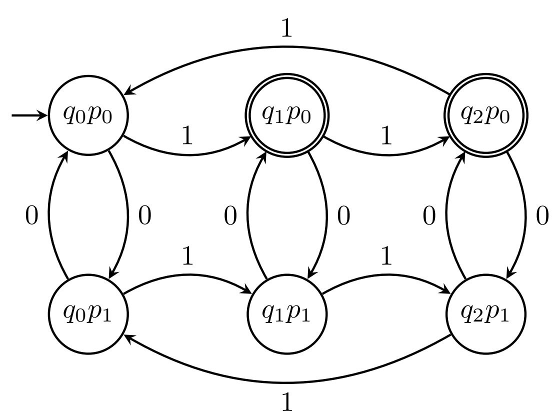


$$L_2 = \{u \in \{0,1\}^* \mid \text{el número de 0's es par}\}$$

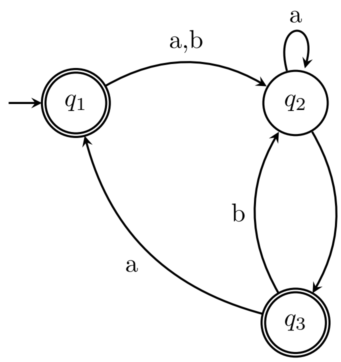


Ahora sólo tenemos que intersecar los dos autómatas para formar el deseado, el que reconoce el lenguaje

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



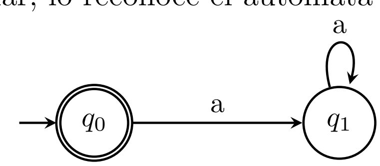
Ejercicio 22. Para hallar la expresión regular que representa el lenguaje aceptado por el autómata



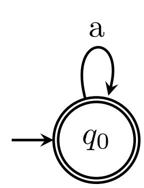
usaremos el algoritmo visto en clase (sólo mostraré un par de iteraciones)

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

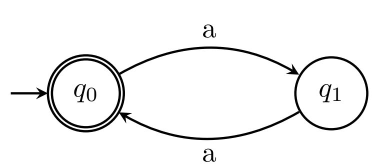
Ejercicio 23. Para probar que $B_n = \{a^k \mid \text{k es múltiplo de n}\} = \{a^{kn} \mid k \in \mathbb{N}\}$ es regular para todo n, construiremos un autómata finito determinista que lo reconozca.
$B_0 = \{\varepsilon\}$ es trivialmente regular, lo reconoce el autómata



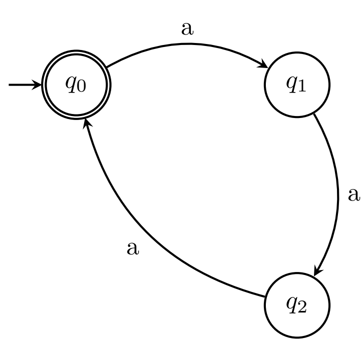
$B_1 = \{a\}^*$ es reconocido por



$B_2 =$ 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) $NOPREFIJO(L) = \{u \in L \mid \text{ningún prefijo propio de } u \text{ pertenece a } L\}$

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

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.

Aclaración: En ambos apartados no he tenido en cuenta que la palabra vacía no es un prefijo propio. Procedo a arreglar ambas pruebas:
En los autómatas finitos deterministas que he hallado, si la palabra vacía está en L, se debe eliminar el estado inicial q_0 del conjunto de estados finales y añadir una transición nula de q_0 a otro nuevo estado final r. Ahora tengo autómatas no deterministas con transiciones nulas que verdaderamente reconocen los lenguajes $NOPREFIJO(L)$ y $NOEXTENSION(L)$, luego son regulares.

Ejercicio 25. $L \subseteq A^*$ define una relación $\equiv$ en A^*.

$$u \equiv v \text{ si y sólo si } \forall z \in A^* \text{ se tiene } (uz \in L \Leftrightarrow vz \in L)$$

a) Probemos que $\equiv$ es una relación de equivalencia

Reflexiva: $(uz \in L \Leftrightarrow uz \in L)\ \forall z \in A^*$, luego $u \equiv u\ \forall u \in A^*$.

Simétrica: $(uz \in L \Leftrightarrow vz \in L) \Leftrightarrow (vz \in L \Leftrightarrow uz \in L)$, luego $u \equiv v \Leftrightarrow v \equiv u$.

Transitiva: Si $(uz \in L \Leftrightarrow vz \in L)$ y $(vz \in L \Leftrightarrow wz \in L)$ claramente tenemos $(uz \in L \Leftrightarrow wz \in L)$, luego $u \equiv v$ y $v \equiv w$ implica $u \equiv w$.

b) Calcularemos las clases de equivalencia de $L = \{a^i b^i | i \geq 0\}$.

$C =$ Clase de las palabras que no son prefijo de ninguna palabra de L:

- Palabras de la forma $A^*bA^*aA^*$.
- Palabras de la forma $a^i b^j$ donde $j > i \geq 0$.

C_{ij} $(i \geq j \geq 0) =$ Clase de equivalencia de la palabra $a^i b^j$ (formada únicamente por dicha palabra).

c) Calcularemos las clases de equivalencia de $L = \{a^i b^j | i, j \geq 0\}$.

$C =$ Clase de las palabras que no son prefijo de ninguna palabra de L, son de la forma $A^*bA^*aA^*$.

$C_a =$ Clase de equivalencia de las palabras de la forma a^i con $i \geq 0$.

$C_b =$ Clase de equivalencia de las palabras de la forma $a^i b^j$ con $i \geq 0,\ j > 0$.

d) Probaré que L es aceptado por un autómata finito determinista si y sólo si el número de clases de equivalencia es finito.

Si L es aceptado por un autómata finito determinista, debe existir un autómata minimal que lo acepte (obviamente finito), en el apartado siguiente pruebo que palabras de distinta clase de equivalencia llevan a estados distintos, luego el número de clases de equivalencia debe ser finito.

Si el número de clases de equivalencia de L es finito ($C_0, \ldots, C_{n-1}$, podemos suponer $\varepsilon \in C_0$), creamos un autómata que lo reconozca: $M = (\{q_0, \ldots, q_{n-1}\}, A, q_0, \delta, F)$

$\delta(q_i, a) =$ q_j donde C_j es la clase de equivalencia de $u_i a$, siendo u_i cualquier palabra de C_i

$F =$ $\{q_i \in Q | u \in C_i \implies u \in L\}$

e) La relación que existe entre el número de clases de equivalencia y el autómata finito minimal que acepta L es que el número de clases de equivalencia es el número de estados del autómata.

La idea es que si u y v son dos palabras de la misma clase de equivalencia, $q_u = \delta(q_0, u)$ y $q_v = \delta(q_0, v)$ deben ser el mismo estado, ya que $\forall z \in A^*$ tenemos que $\delta(q_u, z) \in F \Leftrightarrow \delta(q_v, z) \in F$ y si $q_u \neq q_v$ se podrían unificar, contradiciendo así la propiedad de ser minimal.

Recíprocamente, si u y v no pertenecen a la misma clase de equivalencia q_u y q_v deberán ser diferentes, ya que existe un $z \in A^*$ cumpliendo $\delta(q_u, z) \in F$ y $\delta(q_v, z) \notin F$ o viceversa.