Modelos de Computación: Relación de problemas 3

David Cabezas Berrido

Ejercicio 27. Expresiones regulares para los lenguajes sobre $\{0,1\}$:

a) Palabras con número de símbolos 0 múltiplo de 3.

$$(1^*01^*01^*01^*)^*$$

b) Palabras que contienen como subcadena a 1100 ó a 00110.

$$(0+1)^*(1100+00110)(0+1)^*$$

c) Palabras en las que cada cero forma parte de una subcadena de 2 ceros y cada 1 forma parte de una subcadena de 3 unos.

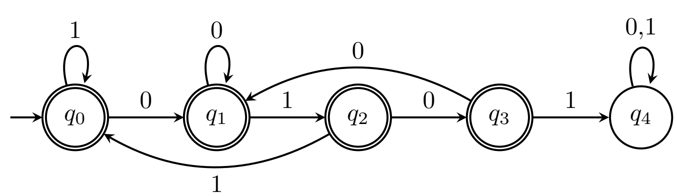
$$(00+111)^*$$

d) Palabras en las que el número de ocurrencias de la subcadena 011 es menor o igual que el de ocurrencias de la subcadena 110.

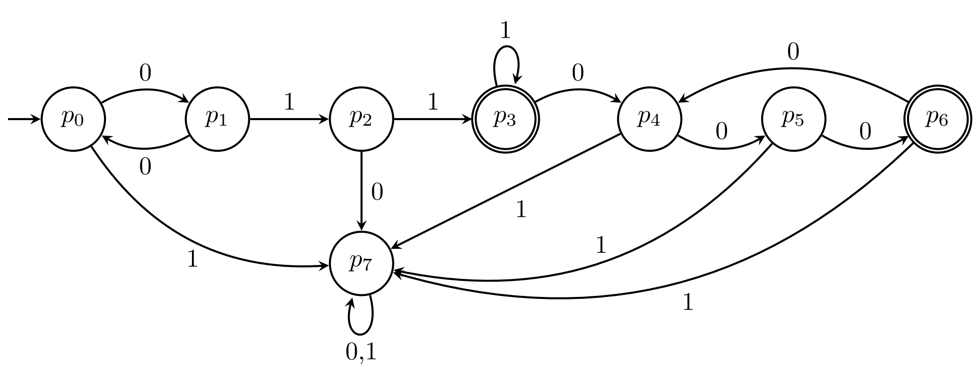
Este lenguaje no es regular, luego no existe una expresión regular para el mismo.

Ejercicio 29. Encontrar gramática tipo 3 o autómata finito que reconozca el lenguaje.

- $L_1 = \{u \in \{0,1\}^* : u \text{ no contiene la subcadena '0101'}\}$

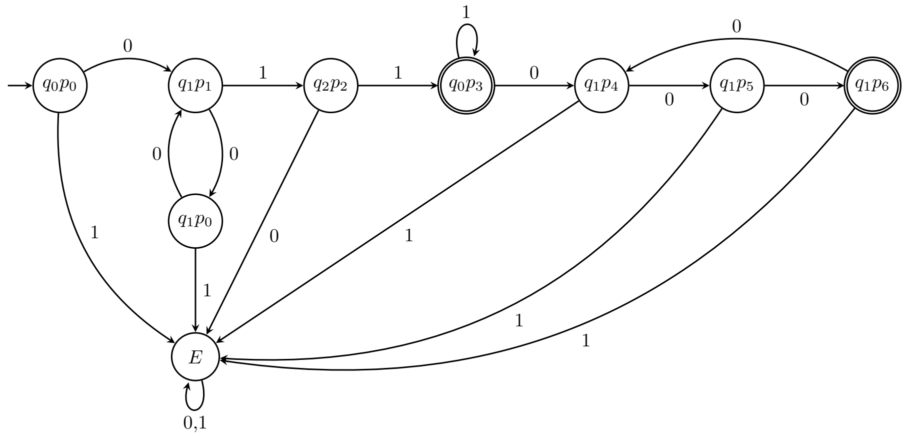


- $L_2 = \{0^i1^j0^k : i \geq 1,\ k \geq 0,\ i \text{ impar}, k \text{ múltiplo de 3 y } j \geq 2\}$

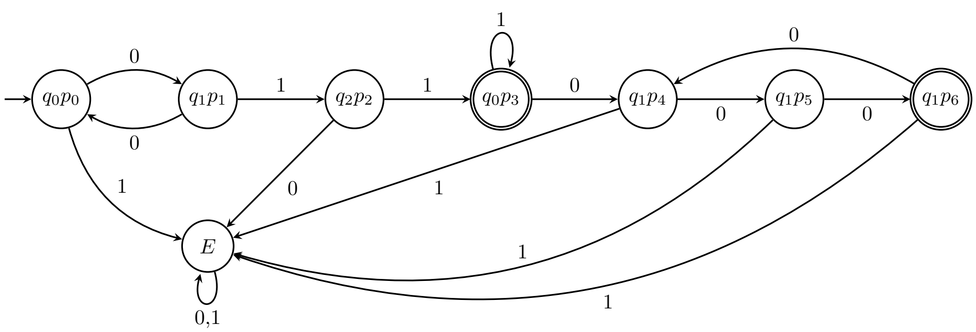


Diseñar el AFD minimal que reconoce $(L_2 \cap L_1)$.

El autómata sin minimizar (sólo agrupando los estados de error) es el siguiente.



Tiene sólo dos estados indistinguibles: q_0p_0 y q_1p_0, luego el autómata minimal es el que resulta de agrupar ambos estados.



Podemos apreciar que este autómata es isomorfo al que hemos diseñado para L_2, lo cual ocurre porque $L_2 \subset L_1$ (es claro que las palabras de L_2 no tienen a '0101' como subcadena).

Ejercicio 45. $A = \{0, 1, +, =\}$, probar que el lenguaje

$$ADD = \{x = y + z \mid x, y, z \text{ son números en binario y la suma es correcta}\}$$

no es regular.

Probaré que ADD no satisface el lema de bombeo. Sea $n \in \mathbb{N}$, la palabra $1^n = 0 + 1^n$ está en el lenguaje y tiene longitud $2n+3 \geq n$. Tomamos una descomposición cualquiera de la forma uvw con $|uv| = l \leq n$ y $|v| \geq 1$, por lo tanto $v = 1^k$ para algún $k \geq 1$.

La palabra uv^2w será $1^{l-k}1^{2k}1^{n-l} = 0 + 1^n$, que simplificando queda $1^{n+k} = 0 + 1^n$. Como $k \geq 1$, el valor númerico de 1^{n+k} será estrictamente mayor que el de 1^n, por lo tanto la palabra uv^2w no está en ADD (la suma no es correcta).

Resumiendo, dado un valor de n cualquiera, podemos encontrar en el lenguaje una palabra z de longitud mayor que n tal que para toda descomposición de z de la forma uvw, la palabra uv^2w no está en el lenguaje. Luego ADD no es regular.

Ejercicio 46*.* La mezcla perfecta de dos lenguajes L_1 y L_2 sobre un alfabeto A se define como

$$\{w \mid w = a_1b_1a_2b_2\dots a_kb_k \text{ donde } a_1\dots a_k \in L_1,\ b_1\dots b_k \in L_2,\ a_i, b_i \in A\}$$

Demostrar que si L_1 y L_2 son regulares, su mezcla perfecta lo es.

Consideraremos $\equiv_1$ y $\equiv_2$ las clases de equivalencia en A^* determinadas por L_1 y L_2 respectivamente. Si $u, v \in A^*$:

$$u \equiv_1 v \text{ cuando } \forall z \in A^*,\ uz \in L_1 \Leftrightarrow vz \in L_1$$

$$u \equiv_2 v \text{ cuando } \forall z \in A^*,\ uz \in L_2 \Leftrightarrow vz \in L_2$$

Consideremos también los autómatas minimales que reconocen a L_1 y a L_2, los conjuntos de estados serán los conjuntos cocientes que definen estas relaciones.

$$M_1 = (Q_1 = \frac{A^*}{\equiv_1}, A, [\varepsilon]_1, \delta_1, F_1 = \{[u]_1 : u \in L_1\})$$
$$M_2 = (Q_2 = \frac{A^*}{\equiv_2}, A, [\varepsilon]_2, \delta_2, F_2 = \{[u]_2 : u \in L_2\})$$

donde

$$\delta_1([u]_1, a) = [ua]_1 \quad \forall u \in A^*, a \in A$$
$$\delta_2([u]_2, a) = [ua]_2 \quad \forall u \in A^*, a \in A$$

Definimos el autómata

$$M = (Q_1 \times Q_2 \times \{0,1\}, A, ([\varepsilon]_1, [\varepsilon]_2, 0), \delta, F_1 \times F_2 \times \{0\})$$

donde

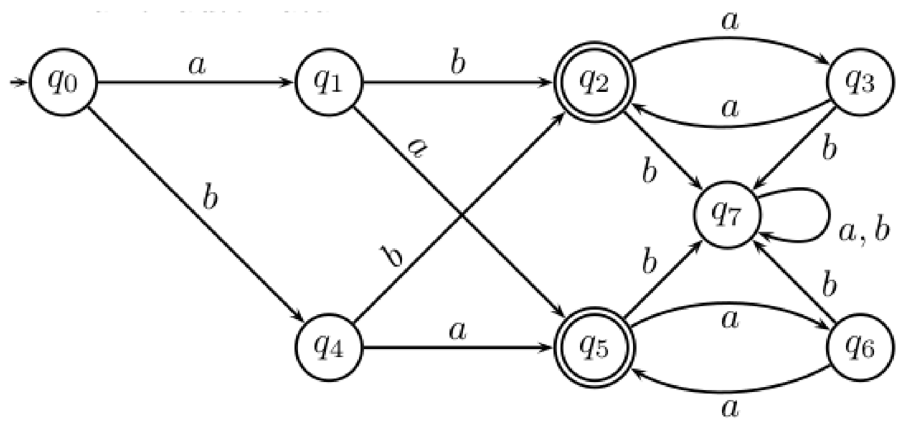
$$\delta(([u]_1, [v]_2, 0), a) = (\delta_1([u]_1, a), [v]_2, 0) = ([ua]_1, [v]_2, 1) \quad \forall u, v \in A^*, a \in A$$

$$\delta(([u]_1, [v]_2, 1), a) = ([u]_1, \delta_2([v]_2, a), 0) = ([u]_1, [va]_2, 0) \quad \forall u, v \in A^*, a \in A$$

Una palabra $w = a_1a_2\dots{}_{2k} \in A^*$ lleva al estado $([u]_1, [v]_2, 0)$ donde $u = a_1a_3\dots a_{2k-1}$ y $v = a_2a_4\dots a_{2k}$, por lo que $w \in L(M) \Leftrightarrow ([u]_1, [v]_2, 0) \in F_1 \times F_2 \times \{0\} \Leftrightarrow u \in L_1$ y $v \in L_2$. Luego M acepta una palabra de longitud par si y sólo si pertenece a la mezcla perfecta de L_1 y L_2.
Una palabra de longitud impar llevará a un estado de $Q_1 \times Q_2 \times \{1\}$, luego M no aceptará palabras de longitud impar, puesto que estas nunca estarán en la mezcla perfecta de L_1 y L_2.

Resumiendo, hemos hallado un AFD que reconoce la mezcla perfecta de L_1 y L_2, luego es regular.

Ejercicio 47. Minimizar el autómata.



No hay estados inaccesibles.
Usando el algoritmo visto en clase, he obtenido que las parejas de estados indistinguibles son (q_1, q_4), (q_2, q_5) y (q_3, q_6).
El autómata minimal será el resultante de agrupar cada una de las parejas.