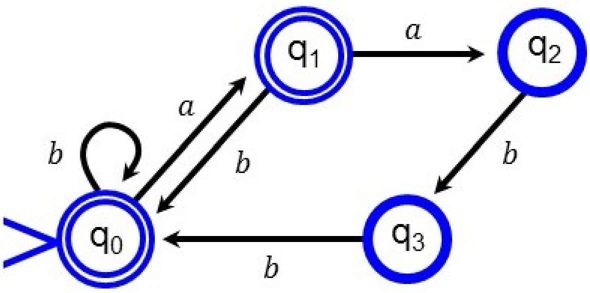
Traductores de Lenguajes de Programación – Bloque I **28 Mayo 2020**
Duración del examen: **1 hora 10 minutos** – Ponderación del examen: 50% de la nota total Ejercicio 1 a) : 2 puntos Ejercicio 2 b) : 1,5 puntos Ejercicio 1 b) : 2 puntos Ejercicio 3 : 3 puntos Ejercicio 2 a) : 1,5 puntos **Tiempo**. Ejercicios 1 y 2 : 40 minutos Ejercicio 3 : 30 minutos Fecha prevista de publicación de notas: 10 de Junio, en el Moodle normal. Revisión: El día y horario de revisión se publicará junto con las notas.

Ejercicio 1. Sea el alfabeto $\Sigma = \{a, b\}$. Hallar autómatas finitos deterministas para los siguientes lenguajes:

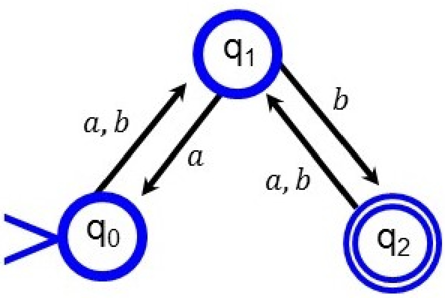
a) $L_1 = \{x \in \Sigma^* :$ toda aparición de aa en x está seguida inmediatamente por $bb\}$.

b) $L_2 = \{x \in \Sigma^* : x$ termina en $b, |x| \geq 2$ y $|x|$ es par$\}$.

a)



b)



Ejercicio 2. a) Sea Σ un alfabeto cualquiera y sean $L_1, L_2 \subset \Sigma^*$ dos lenguajes. Se sabe que L_1 es un lenguaje regular y que los lenguajes concatenados $L_1 \cdot L_2$ y $L_2 \cdot L_1$ también son regulares. ¿Es L_2 necesariamente un lenguaje regular? Justificar la respuesta.

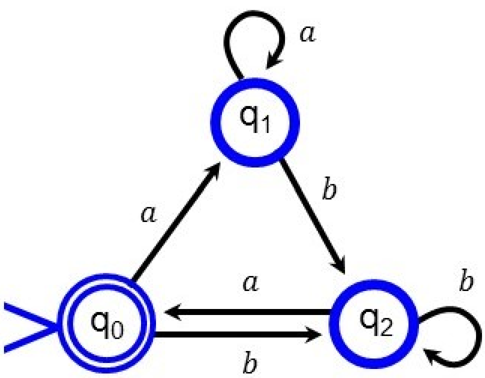
b) Demostrar que el lenguaje $L = \{a^n b^m : n > m \geq 1\}$ no es regular.

a) No necesariamente. Por ejemplo, los lenguajes $L_1 = a^*$ y $L_2 = \{a^{n^2} : n \geq 0\}$ verifican el enunciado, y L_2 no es un lenguaje regular. Nótese que en este caso $L_1 \cdot L_2 = L_2 \cdot L_1 = a^*$.

b) Sea p como en el lema de bombeo y sea $w = a^{p+1}b^p = xyz \in L$, entonces $|w| = 2p + 1 \geq p$. La parte xy sólo puede contener símbolos a, ya que $|xy| \leq p$. Por otra parte, al ser $|y| \geq 1$, la palabra $w_0 = xy^0z$ puede escribirse como $w_0 = a^{p+1-i}b^p$ para algún $1 \leq i \leq p$. Pero $p + 1 - i \leq p$, y por tanto $w_0 \notin L$. Esto significa que L no es regular.

Ejercicio 3. Hallar el autómata finito determinista mínimo que acepta el lenguaje generado por la expresión regular $(a^*(ba)^*bb^*a)^*$. Justificar razonadamente por qué el autómata obtenido es mínimo.

El autómata mínimo es:



Es obvio que el autómata mínimo no puede tener un estado. Tampoco es posible aceptar el subconjunto $(a^*ba)^*$ del lenguaje con sólo dos estados, ya que el estado inicial debe ser aceptador. En consecuencia, el autómata dado en la figura es mínimo, por la unicidad de los autómatas mínimos.