Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI3725 - Traductores e Interpretadores
Septiembre-Diciembre 2008

Examen Parcial I

(20 puntos)

Carnet:

Nombre:

1. Sea el alfabeto $\Sigma = \{a, b, c\}$.

   (a) **(1.5 puntos)** Escriba una expresión regular que denote el conjunto de palabras sobre Σ^* tales que *no* tengan c, pero tengan un número par de a y un número par de b.

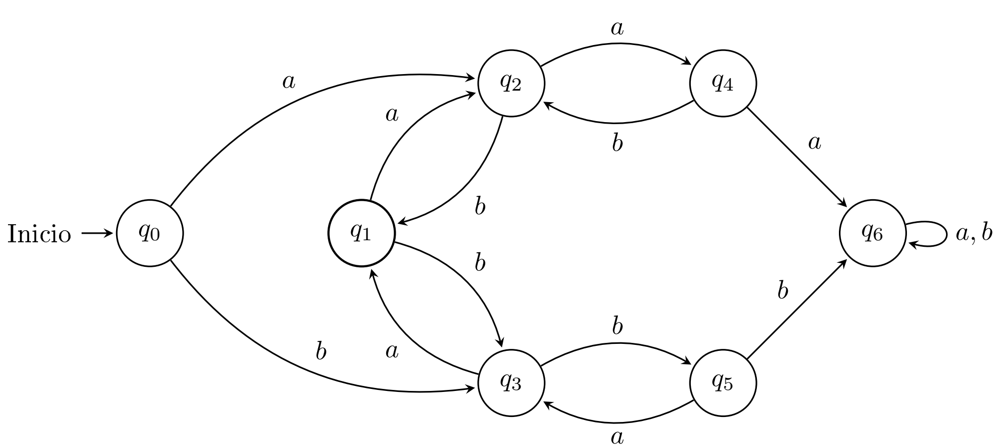
$$(aa + bb) * ((ab + ba)(aa + bb) * (ab + ba)(aa + bb)*)*$$

   (a) **(1.5 puntos)** Escriba una expresión regular que denote el conjunto de palabras sobre Σ^* tales que cuando contienen a o c, la cantidad de a y c sumadas *no* es múltiplo de tres.

$$(b * (a + c) * b * (a + c)b * (a + c)) * b * (a + c)b * (a + c + \lambda)b * +b*$$

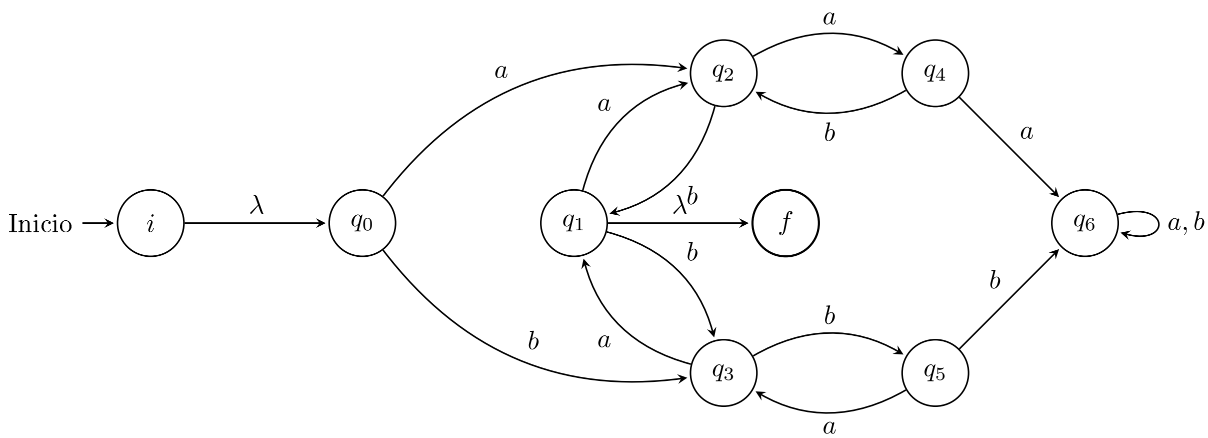
2. Sea el alfabeto $\Sigma = \{a, b\}$.

   (a) **(3 puntos)** Construya un AFD que reconozca el lenguaje regular de las palabras de dos o más símbolos, que tengan la misma cantidad de a y b, y que en cada uno de sus prefijos, las cantidades de a y b difieran en a lo sumo dos. Basta que dé la representación gráfica de su autómata en lugar de la 5-tupla correspondiente.

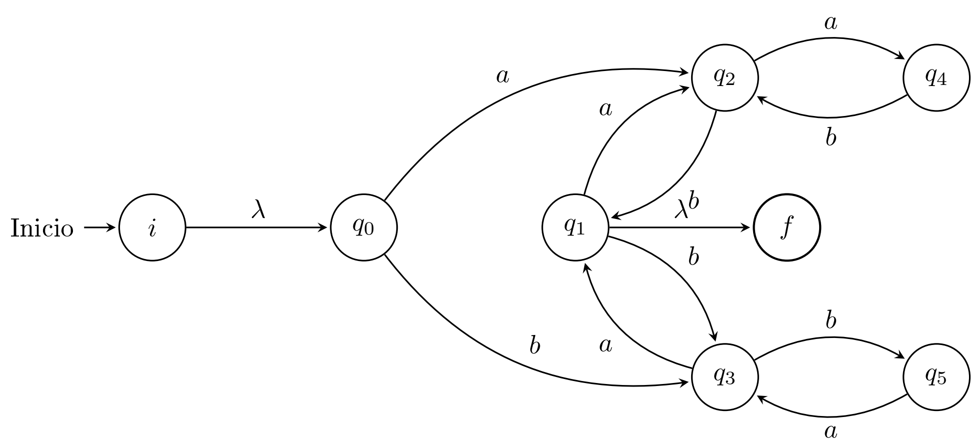


(b) **(3 puntos)** Calcule la expresión regular que denote el lenguaje reconocido por el AFD construido.

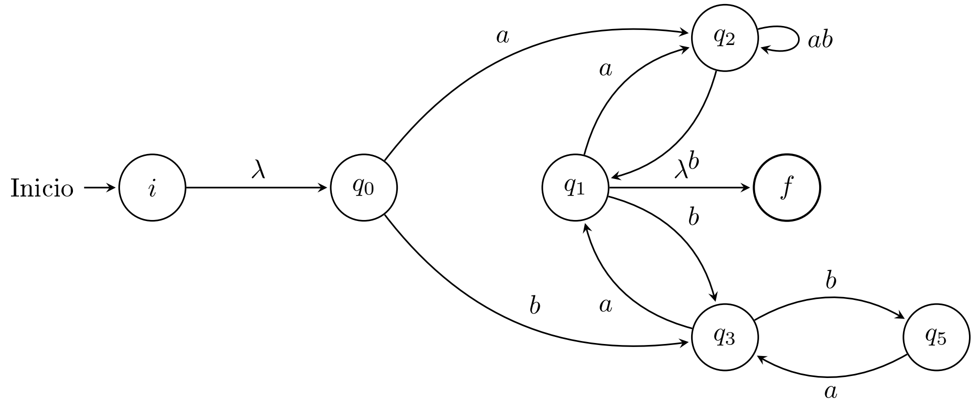
i. Agregamos un nuevo estado inicial i el cual conecta con el estado q_0 usando una λ-transición. Agregamos un nuevo estado final f y conectamos el estado $q_1$con f usando una λ-transición.



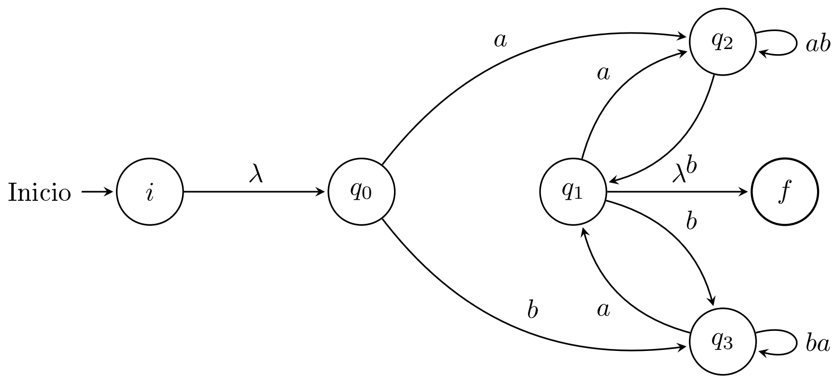
ii. Eliminamos el estado q_6 por tratarse de un sumidero.



iii. Seleccionamos el estado q_4 para remoción, preservando los caminos con los cuales contribuye. Desde el estado q_2 se puede llegar al estado $q_2$con la expresión ab, por lo tanto



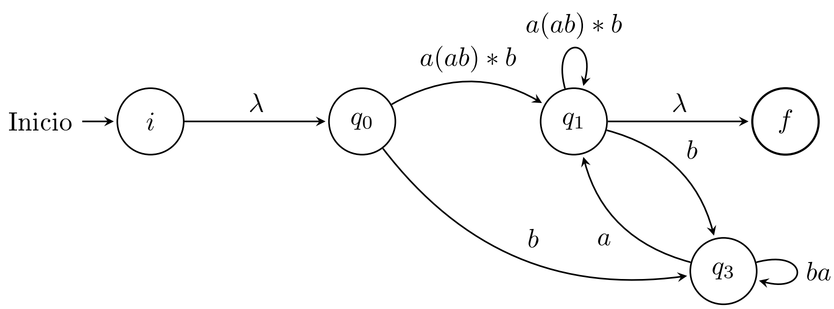
iv. Seleccionamos el estado q_5 para remoción, preservando los caminos con los cuales contribuye. Desde el estado q_3 se puede llegar al estado q_3 con la expresión ba, por lo tanto



v. Seleccionamos el estado q_2 para remoción, preservando los caminos con los cuales contribuye, a saber:

A. Desde el estado q_0 se puede llegar al estado q_1 con la expresión $a(ab)*b$.

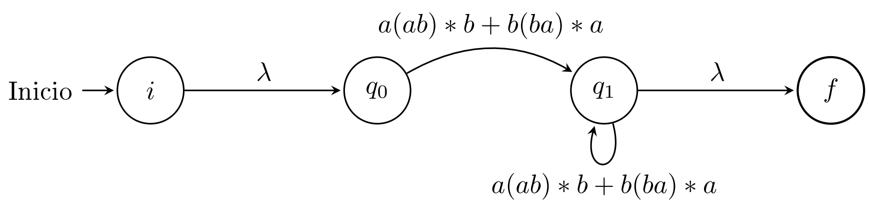
B. Desde el estado $q_1$se puede llegar al estado $q_1$con la expresión $a(ab)*b$.



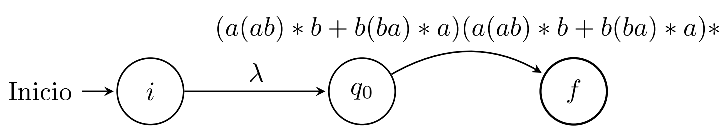
vi. Seleccionamos el estado q_3 para remoción, preservando los caminos con los cuales contribuye, a saber:

A. Desde el estado q_0 se puede llegar al estado q_1 con la expresión $b(ba)*a$.

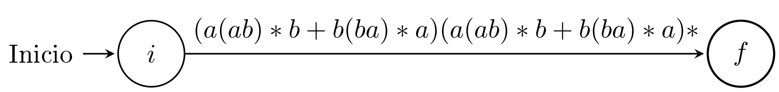
B. Desde el estado $q_1$se puede llegar al estado $q_1$con la expresión $b(ba)*b$.



vii. Seleccionamos el estado q_1 para remoción, preservando los caminos con los cuales contribuye. Desde el estado q_0 se puede llegar hasta el estado f con la expresión $(a(ab)*b+b(ba*)a)(a(ab)*b+b(ba)*a)*$.



viii. Seleccionamos el estado q_0 para remoción, preservando los caminos con los cuales contribuye. Desde el estado i se puede llegar hasta el estado f con la expresión $(a(ab)*b+b(ba*)a)(a(ab)*b+b(ba)*a)*$.



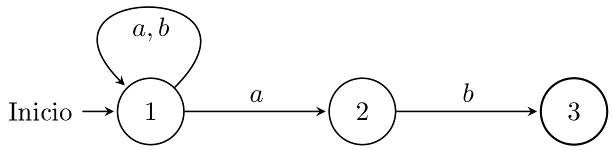
Por tanto, la expresión regular asociada al DFA es

$$(a(ab)*b+b(ba*)a)(a(ab)*b+b(ba)*a)*$$

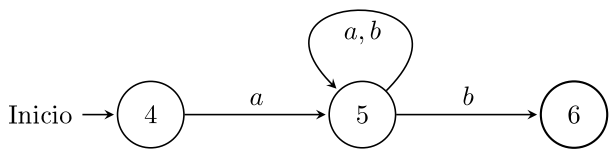
3. Sea el alfabeto $\Sigma = \{a, b\}$. Considere los siguientes NFA :

(a) **(0.75 puntos)** Construya sendos autómatas finitos *no-determinísticos,* posiblemente utilizando $\lambda-$transiciones, que reconozcan las expresiones regulares correspondientes a los conjuntos:

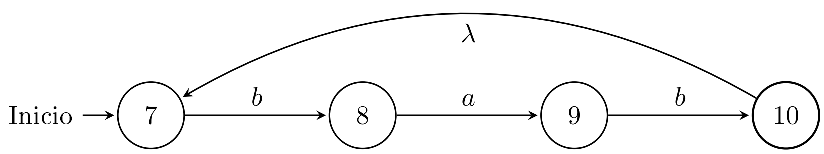
i. Las palabras que terminan en ab.



ii. Las palabras que comienzan por a y terminan con b.

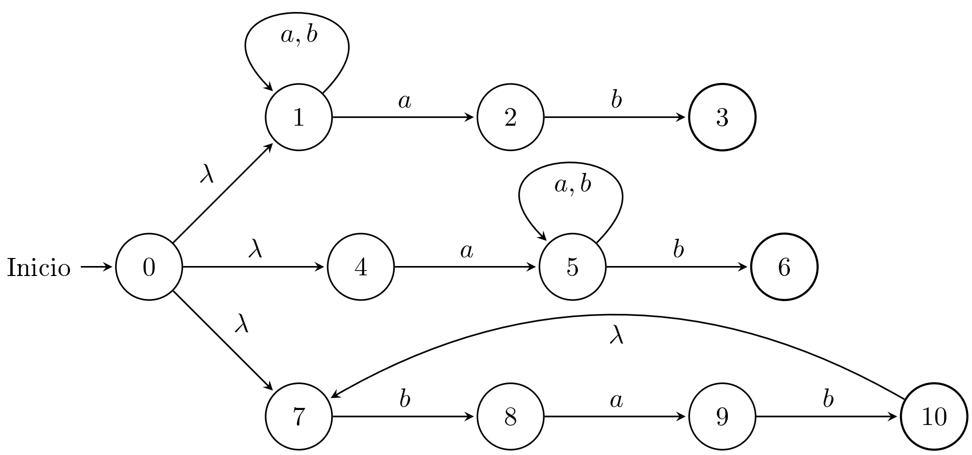


iii. Las palabras constituidas por una o más repeticiones de la cadena bab.



Basta la representación gráfica de cada uno de los autómatas en lugar de la 5-tupla correspondiente.

(b) **(0.25 puntos)** Combine los tres autómatas para crear un AFN-λ que reconozca la unión de los tres lenguajes anteriores, de manera que al procesar alguna cadena de entrada y reconocerla, se pueda saber a cuál de los tres lenguajes pertenece.



(c) **(4 puntos)** Convierta el AFN-λ construido en un AFD. Presente el procedimiento detallado de cálculo de las λ-clausuras y la construcción de la función de transición extendida.

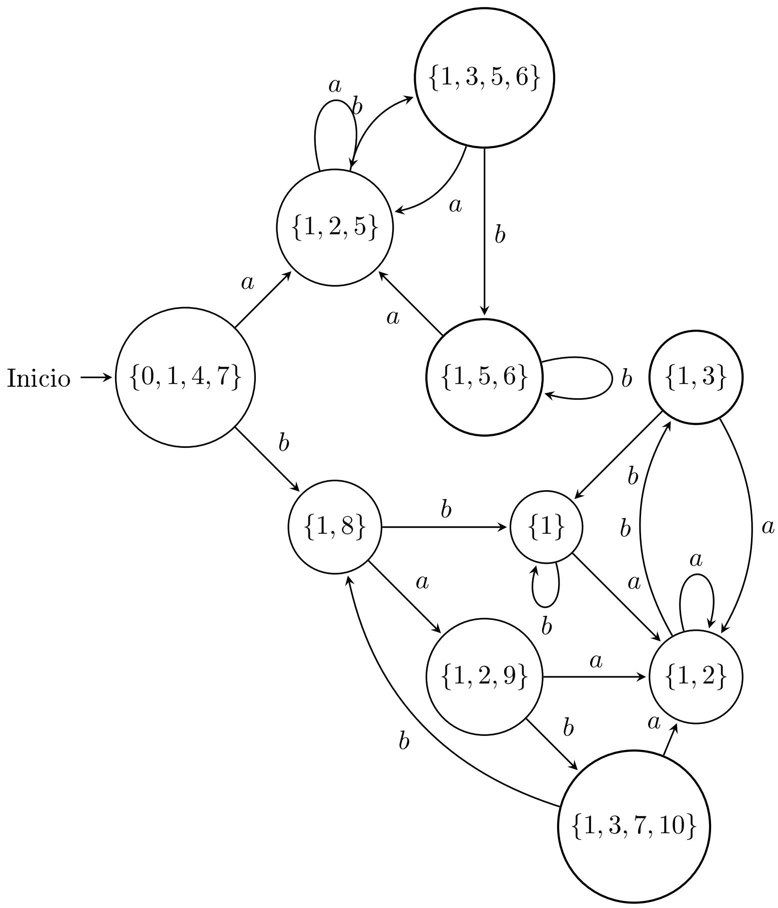
Primero es necesario calcular la λ-clausura para cada uno de los estados del λ-AFN, que por simple inspección puede verse

$$\begin{aligned} \lambda - clausura(0) &= \{0, 1, 4, 7\} \\ \lambda - clausura(i) &= \{i\}, 1 \leq i \leq 9 \\ \lambda - clausura(10) &= \{7, 10\} \end{aligned}$$

Calculamos entonces la tabla con la Función de Transición Extendida.

t	a	b
0	$\{1,2,5\}$	$\{1,8\}$
1	$\{1,2\}$	$\{1\}$
2	$\emptyset$	$\{3\}$
3	$\emptyset$	$\emptyset$
4	$\{5\}$	$\emptyset$
5	$\{5\}$	$\{5,6\}$
6	$\emptyset$	$\emptyset$
7	$\emptyset$	$\{8\}$
8	$\{9\}$	$\emptyset$
9	$\emptyset$	$\{7,10\}$
10	$\emptyset$	$\{8\}$

Construimos el AFD según el algoritmo explicado en clase, partiendo del nuevo estado inicial construido a partir de $\lambda - clausura(0)$.



(d) **(1 punto)** Indique a cuál de los lenguajes originales corresponde cada estado final del AFD. En caso de ambigüedad, se prefieren las palabras del conjunto (1) antes que las palabras del conjunto (2), y se prefieren las palabras del conjunto (3) antes que las palabras del conjunto (1).

i. Si el AFD acepta en el estado $\{1,3\}$ entonces está aceptando una palabra que corresponde al conjunto (1), i.e. palabras que terminan en ab, pues en el λ-AFN original el estado 3 era el de aceptación para dicho lenguaje.

ii. Si el AFD acepta en el estado $\{1,5,6\}$ entonces está aceptando una palabra que corresponde al conjunto (2), i.e. palabras que comienzan por a y terminan por b, pues en el λ-AFN original el estado 6 era el de aceptación para dicho lenguaje.

iii. Si el AFD acepta en el estado $\{1, 3, 5, 6\}$ entonces está aceptando una palabra que corresponde al conjunto (1), i.e. palabras que terminan en ab, pues en el λ-AFN original el estado 3 era el de aceptación para el conjunto (1) y el estado 6 era el de aceptación para el conjunto (2), sin embargo la precedencia establecida en el enunciado de la pregunta indica que ante esta ambigüedad debemos preferir al conjunto (1).

iv. Si el AFD acepta en el estado $\{1, 3, 7, 10\}$ entonces está aceptando una palabra que corresponde al conjunto (3), i.e. palabras constituidas por una o más repeticiones de la cadena bab, pues en el λ-AFN original el estado 3 era el de aceptación para el conjunto (1) y el estado 10 era el de aceptación para el conjunto (3), sin embargo la precedencia establecida en el enunciado de la pregunta indica que ante esta ambigüedad debemos preferir al conjunto (3).

4. **(3 puntos)** Construya el AFD mínimo equivalente, mostrando y justificando la construcción de los $\equiv_i$ necesarios, para el AFD definido por la 5-tupla

$$M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{a, b\}, \delta, q_0, \{q_1, q_3, q_5, q_6\})$$

δ	a	b
q_0	q_1	q_3
q_1	q_2	q_4
q_2	q_5	q_5
q_3	q_4	q_2
q_4	q_5	q_5
q_5	q_6	q_5
q_6	q_6	q_6

Por definición, la clase de equivalencia $\equiv_0$ tiene dos conjuntos, el de estados finales y el de estados no finales, por tanto

$$\equiv_0 \quad = \quad \{\{q_0, q_2, q_4\}, \{q_1, q_3, q_5, q_6\}\}$$

Para calcular $\equiv_1$ consideramos:

(a) Los estados q_0 y q_2 **si** son equivalentes puesto que $\delta(q_0, a) \equiv_0 \delta(q_2, a)$ y $\delta(q_0, b) \equiv_0 \delta(q_2, b)$.

(b) Los estados q_0 y q_4 **si** son equivalentes puesto que $\delta(q_0, a) \equiv_0 \delta(q_4, a)$ y $\delta(q_0, b) \equiv_0 \delta(q_4, b)$.

(c) Los estados q_1 y q_3 **si** son equivalentes puesto que $\delta(q_1, a) \equiv_0 \delta(q_3, a)$ y $\delta(q_1, b) \equiv_0 \delta(q_3, b)$.

(d) Los estados q_1 y q_5 **no** son equivalentes puesto que $\delta(q_1, a) \not\equiv_0 \delta(q_5, a)$.

(e) Los estados q_1 y q_6 **no** son equivalentes puesto que $\delta(q_1, a) \not\equiv_0 \delta(q_6, a)$.

(f) Los estados q_5 y q_6 **si** son equivalentes puesto que $\delta(q_5, a) \equiv_0 \delta(q_6, a)$ y $\delta(q_5, b) \equiv_0 \delta(q_6, b)$.

en consecuencia

$$\equiv_1 \quad = \quad \{\{q_0, q_2, q_4\}, \{q_1, q_3\}, \{q_5, q_6\}\}$$

Para calcular $\equiv_2$consideramos:

(a) Los estados q_0 y q_2 **no** son equivalentes puesto que $\delta(q_0, a) \not\equiv_1 \delta(q_2, a)$.

(b) Los estados q_0 y q_4 **no** son equivalentes puesto que $\delta(q_0, a) \not\equiv_1 \delta(q_4, a)$.

(c) Los estados q_2 y q_4 **si** son equivalentes puesto que $\delta(q_2, a) \equiv_1 \delta(q_4, a)$ y $\delta(q_2, b) \equiv_1 \delta(q_4, b)$.

(d) Los estados q_1 y q_3 **si** son equivalentes puesto que $\delta(q_1, a) \equiv_1 \delta(q_3, a)$ y $\delta(q_1, b) \equiv_1 \delta(q_3, b)$.

(e) Los estados q_5 y q_6 **si** son equivalentes puesto que $\delta(q_5, a) \equiv_1 \delta(q_6, a)$ y $\delta(q_5, b) \equiv_1 \delta(q_6, b)$.

En consecuencia

$$\equiv_2 \quad = \quad \{\{q_0\}, \{q_2, q_4\}, \{q_1, q_3\}, \{q_5, q_6\}\}$$

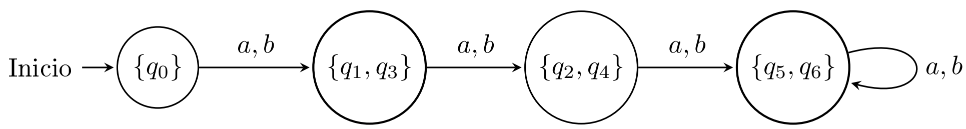
Para calcular $\equiv_3$ consideramos

(a) El estado q_0 continúa en su propia clase de equivalencia.
(b) Los estados q_2 y q_4 **si** son equivalentes puesto que $\delta(q_2, a) \equiv_2 \delta(q_4, a)$ y $\delta(q_2, b) \equiv_2 \delta(q_4, b)$.
(c) Los estados q_1 y q_3 **si** son equivalentes puesto que $\delta(q_1, a) \equiv_2 \delta(q_3, a)$ y $\delta(q_1, b) \equiv_2 \delta(q_3, b)$.
(d) Los estados q_5 y q_6 **si** son equivalentes puesto que $\delta(q_5, a) \equiv_2 \delta(q_6, a)$ y $\delta(q_5, b) \equiv_2 \delta(q_6, b)$.

En consecuencia

$$\equiv_2 \quad = \quad \{\{q_0\}, \{q_2, q_4\}, \{q_1, q_3\}, \{q_5, q_6\}\}$$

Como $\equiv_2=\equiv_3$ hemos llegado al punto fijo de las clases de equivalencia. Cada una de las clases de equivalencia representará uno de los estados. Así, el AFD mínimo resultante será:



5. **(3 puntos)** Sea el alfabeto $\Sigma = \{a, b\}$. Considere el lenguaje $L = \{w \mid w \in \Sigma^* \wedge |w|_a = |w|_b\}$. Use el Lema de Bombeo para Lenguajes Regulares para demostrar que L no es regular.

Asumo que L es Lenguaje Regular, entonces existe $M = (Q, \Sigma, \delta, q_0, F)$ con $|Q| = k$ que acepta palabras de L. Por el Lema de Bombeo para Lenguajes Regulares sabemos que $\forall z \in L$ con $|z| \geq k$ siempre se puede escribir $z = uvw$ tal que $|uv| \leq k, |v| > 0$ y luego $\forall i \geq 0$ se cumple $uv^iw \in L$.

Consideremos la palabra $w = a^kb^k \in L$, entonces cualquier partición de w que cumpla con las condiciones del Lema de Bombeo debe tener necesariamente $uv = a^j$ con $1 \leq j \leq k$, más aún $v = a^p$ con $p \geq 1$. Así para *cualquier* partición que escojamos, si se bombea v dos o más veces, el segmento de a va a aumentar su longitud mientras el segmento de b permanece intacto, con lo que la cantidad de a en la palabra bombeada será mayor a la cantidad de b y en consecuencia $uv^iw \notin L$ contradiciendo el Lema de Bombeo. Esa contradicción es consecuencia de haber asumido que L era en efecto Lenguaje Regular, por tanto no puede serlo.