Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI3725 - Traductores e Interpretadores
Septiembre-Diciembre 2006

Examen Parcial I

(20 puntos)

Carnet:

Nombre:

1. Para cada uno de los siguientes lenguajes regulares sobre el alfabeto $\{0, 1\}$, dé una expresión regular que lo denote:

   a) **(0.5 puntos)** El conjunto de frases que contienen la subfrase 01 y la subfrase 10.

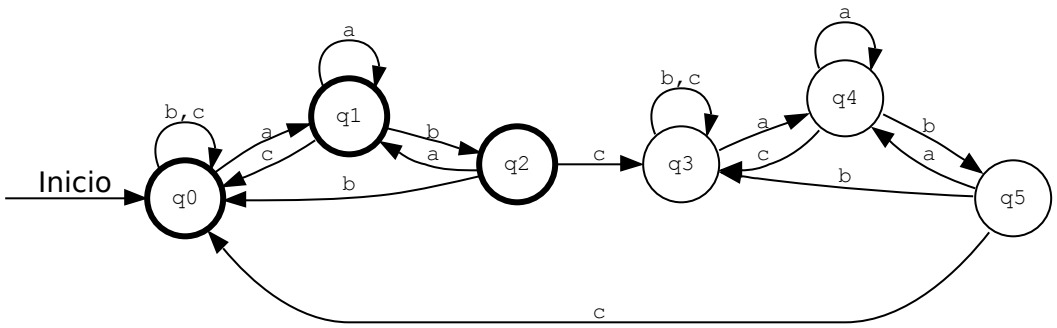
   $$(0+1)*01(0+1)*10(0+1)*+(0+1)*10(0+1)*01(0+1)*+101+010$$

   Nota: las subfrases 01 y 10 pueden aparecer en *cualquier* posición, por eso es necesario emplear la unión de ambas posibilidades, y además incluir los casos extremos.

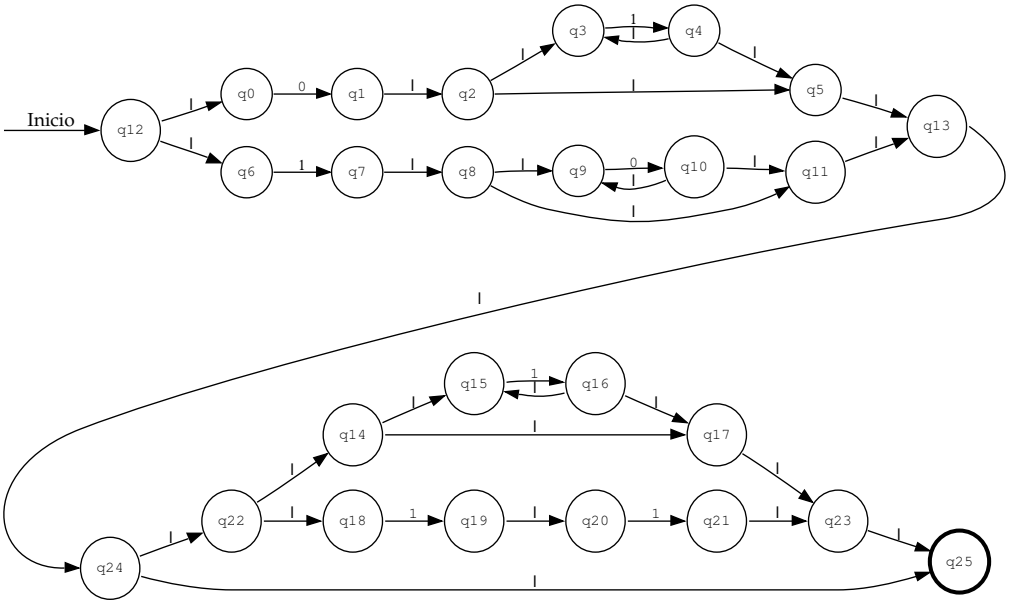
   b) **(0.5 puntos)** El conjunto de frases que contienen la subfrase 10 exactamente una vez.

   $$0*1*100*1*$$

2. **(3 puntos)** Construya un AFD que reconozca el lenguaje regular sobre el alfabeto $\{a, b, c\}$ de las frases que contienen un número **par** de veces la subfrase *abc*. Basta que dé la representación gráfica de su autómata en lugar de la 5-tupla correspondiente.

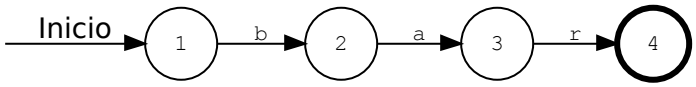


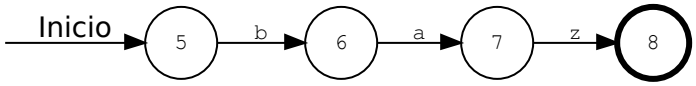
3. **(2 puntos)** Construya un AFN que reconozca el lenguaje regular sobre el alfabeto $\{0, 1\}$ denotado por la expresión regular $(01* + 10*)(1* + 11)*$

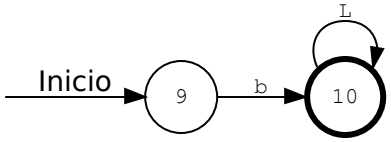


4. Sea el alfabeto que contiene a todas las letras minúsculas y que denotaremos $L = \{a\} \cup \{b\} \cup \{c\} \cup \ldots \cup \{z\}$.

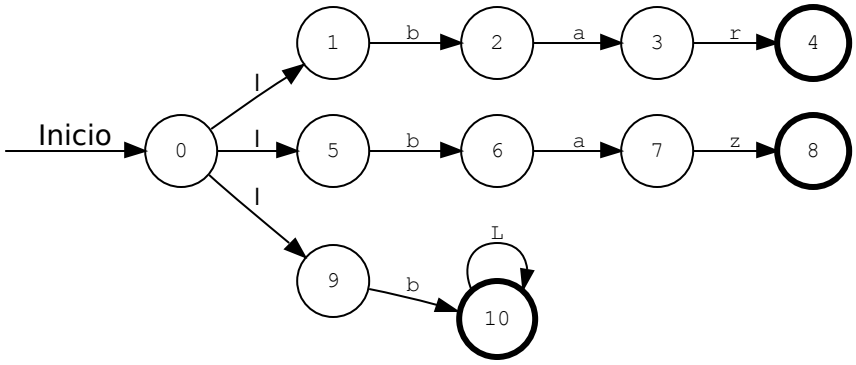
a) **(0.75 puntos)** Construya sendos autómatas finitos *determinísticos incompletos* que reconozcan las expresiones regulares *bar*, *baz* y cualquier secuencia de caracteres que comience por *b*. Basta que dé la representación gráfica de cada uno de los autómatas en lugar de la 5-tupla correspondiente.







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



El autómata creado es una máquina de Moore, pues dependiendo del estado final puede saberse cual lenguaje ha reconocido, en concreto

1) Si acepta en el estado **4** entonces $w \in \{bar\}$.
2) Si acepta en el estado **8** entonces $w \in \{baz\}$.
3) Si acepta en el estado **10** entonces $w \in \{b\}L^*$.

c) **(3 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. *Tip:* recuerde que L es un conjunto, aproveche la notación de conjuntos para simplificar su trabajo.

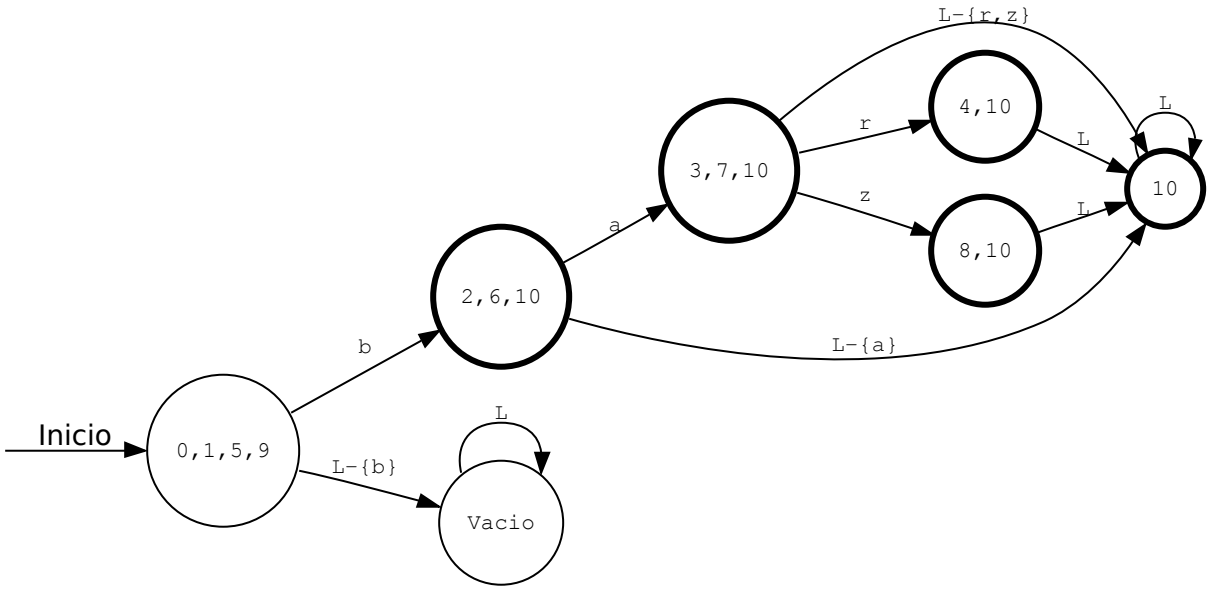
Primero es necesario calcular la λ-clausura para cada uno de los estados del AFN-λ, que por simple inspección puede verse

$$\begin{aligned} \lambda - clausura(0) &= \{0,1,5,9\} \\ \lambda - clausura(q) &= \{q\},\ \forall q \neq 0 \end{aligned}$$

Calculamos entonces la tabla con la Función de Transición Extendida. Nótese el uso de notación de conjuntos para abreviar el trabajo

	a	b	r	z	$L - \{a,b,r,z\}$
0	Ø	$\{2,6,10\}$	Ø	Ø	Ø
1	Ø	$\{2\}$	Ø	Ø	Ø
2	$\{3\}$	Ø	Ø	Ø	Ø
3	Ø	Ø	$\{4\}$	Ø	Ø
4	Ø	Ø	Ø	Ø	Ø
5	Ø	$\{6\}$	Ø	Ø	Ø
6	$\{7\}$	Ø	Ø	Ø	Ø
7	Ø	Ø	Ø	$\{8\}$	Ø
8	Ø	Ø	Ø	Ø	Ø
9	Ø	$\{10\}$	Ø	Ø	Ø
10	$\{10\}$	$\{10\}$	$\{10\}$	$\{10\}$	$\{10\}$

El AFD resultante queda



d) **(1 punto)** Indique a cuál de los lenguajes originales corresponde cada estado final del AFD.

1) Si el DFA acepta en el estado $\{4, 10\}$, entonces está aceptando el lenguaje $\{bar\}$ pues en el NFA-λ original el estado 4 era el de aceptación para dicho lenguaje, y en el DFA solamente el estado $\{4, 10\}$ contiene al 4.
2) Si el DFA acepta en el estado $\{8, 10\}$, entonces está aceptando el lenguaje $\{baz\}$ pues en el NFA-λ original el estado 8 era el de aceptación para dicho lenguaje, y en el DFA solamente el estado $\{8, 10\}$ contiene al 8.
3) Si el DFA acepta en los estados $\{2, 6, 10\}$, $\{3, 7, 10\}$ o $\{10\}$, entonces está aceptando el lenguaje $bL*$ pues en el NFA-λ original el estado 10 era el de aceptación para dicho lenguaje y los estados $\{4, 10\}$ y $\{8, 10\}$ establecen los otros dos lenguajes de interés.

5. Sea el AFD definido por la 5-tupla

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

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

a) **(3 puntos)** Construya el AFD *mínimo* equivalente. Muestre y justifique la construcción de los $\equiv_i$ necesarios.

Por definición, la clase de equivalencia $\equiv_0$ tiene dos conjuntos, el de estados iniciales y el de estados finales, por tanto

$$\equiv_0 \quad = \quad \{\{0, 2, 4\}, \{1, 3, 5\}\}$$

Para calcular $\equiv_1$ consideramos:

1) Los estados 0 y 2 **no** son equivalentes puesto que $\delta(0, b) \not\equiv_0 \delta(2, b)$.
2) Los estados 2 y 4 **si** son equivalentes puesto que $\delta(2, a) \equiv_0 \delta(4, a)$ y $\delta(2, b) \equiv_0 \delta(4, b)$.
3) Si los estados 2 y 4 son equivalentes, y los estados 0 y 2 no son equivalentes, entonces 0 y 4 tampoco son equivalentes.
4) Los estados 1 y 3 **si** son equivalentes puesto que $\delta(1, a) \equiv_0 \delta(3, a)$ y $\delta(1, b) \equiv_0 \delta(3, b)$.
5) Los estados 1 y 5 **no** son equivalentes puesto que $\delta(1, a) \not\equiv_0 \delta(5, a)$.
6) Si los estados 1 y 3 son equivalentes, y los estados 1 y 5 no son equivalentes, entonces 3 y 5 tampoco son equivalentes.

en consecuencia

$$\equiv_1 \quad = \quad \{\{0\}, \{2, 4\}, \{1, 3\}, \{5\}\}$$

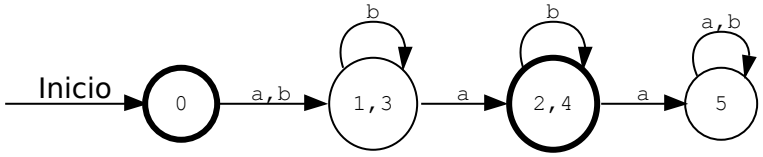
Para calcular $\equiv_2$ consideramos:

1) El estado 0 ya no puede ser equivalente con otro.
2) Los estados 2 y 4 **si** son equivalentes puesto que $\delta(2, a) \equiv_1 \delta(4, a)$ y $\delta(2, b) \equiv_1 \delta(4, b)$.
3) Los estados 1 y 3 **si** son equivalentes puesto que $\delta(1, a) \equiv_2 \delta(3, a)$ y $\delta(1, b) \equiv_2 \delta(3, b)$.
4) El estado 5 ya no puede ser equivalente con otro.

en consecuencia

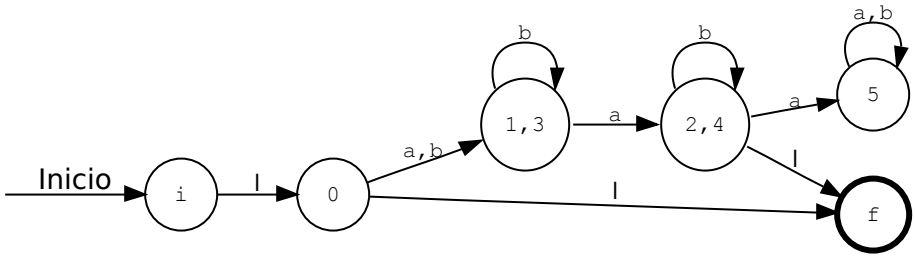
$$\equiv_2 \quad = \quad \{\{0\}, \{2, 4\}, \{1, 3\}, \{5\}\}$$

Como $\equiv_1 = \equiv_2$ hemos llegado al punto fijo de las clases de equivalencias. Cada una de las clases de equivalencia representará uno de los estados. Así, el AFD mínimo resultante será

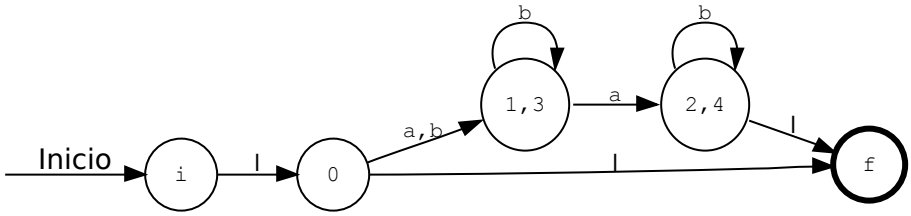


b) **(3 puntos)** Calcule la expresión regular que denota el lenguaje reconocido por el AFD *mínimo* construido.

1) Agregamos un nuevo estado inicial i el cual se conecta al estado 0 usando una λ-transición. Agregamos un nuevo estado final f. Conectamos los estados 0 y $2,4$ con f usando sendas λ-transiciones.



2) Notamos que el estado 5 no participa en *ningún* camino que conduzca desde el estado inicial i hasta el estado final f por tanto no contribuye a la expresión regular y en consecuencia podemos eliminarlo, incluyendo las transiciones que comiezan o terminan en él.

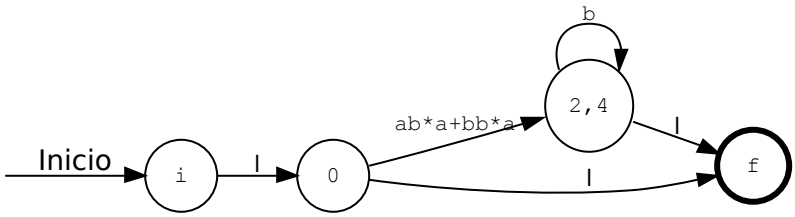


3) Seleccionamos el estado $1,3$ para remoción, preservando los caminos con los cuales contribuye, a saber:

   a' Desde el estado 0 se puede llegar hasta $2,4$ con la expresión $ab*a$.

   b' Desde el estado 0 se puede llegar hasta $2,4$ con la expresión $bb*a$.

   En consecuencia, al remover el estado $1,3$ debe agregarse una transición desde el estado 0 hasta el estado $2,4$ con la expresión $ab*a + bb*a$.

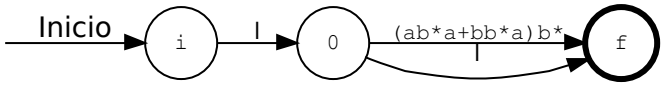


4) Seleccionamos el estado $2,4$ para remoción, preservando los caminos con los cuales contribuye, a saber:

   a' Desde el estado 0 se puede llegar hasta f con la expresión $(ab*a+bb*a)b*$.

   b' Desde el estado 0 **ya existe** otro camino para llegar hasta f con la expresión λ el cual **se mantiene**.

   En consecuencia, al remover el estado $2,4$ debe agregarse una transición desde el estado 0 hasta el estado f con la expresión $(ab*a+bb*a)b*$.

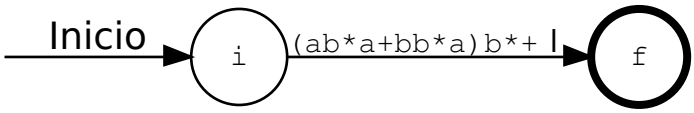


5) Seleccionamos el estado 0 para remoción, preservando los caminos con los cuales contribuye, a saber:

   a' Desde el estado i se puede llegar hasta f con la expresión $(ab*a+bb*a)b*$.

   b' Desde el estado i se puede llegar hasta f con la expresión λ.

   En consecuencia, al remover el estado 0 debe agregarse una transición desde el estado i hasta el estado f con la expresión $(ab*a+bb*a)b*+\lambda$.



6) Puesto que entre los estados i y f sólo queda una transición, la expresión regular asociada a dicha transición es, en efecto, la expresión regular que denota el lenguaje regular reconocido por el autómata original, la cual podemos simplificar aplicando la distributividad de la concatenación sobre la unión

$$\begin{aligned}(ab*a+bb*a)b* &+ \lambda\\ (a+b)b*ab* &+ \lambda\end{aligned}$$

Nota: el orden de remoción de los estados podría haber sido cualquiera, lo importante es justificar cada paso (incluso la remoción del estado que no contribuye, pero explicando por qué). Así mismo, cualquiera de las dos expresiones finales son válidas con o sin simplificación, dado que no se exigió la simpificación de la expresión final.

6. **(3 puntos)** Use el Lema de Bombeo sobre Lenguajes Regulares para demostrar que el conjunto $\{a^n b^m | n < m\}$ no es regular.

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

   Consideremos la palabra $w = a^k b^{k+1} \in L$, entonces cualquier partición de w que cumpla con el Lema de Bombeo debe tener $uv = a^j$ con $1 \leq j \leq k$. Siendo así, necesariamente$v = a^p$, con $1 < p \leq k$ y entonces w tiene la forma

   $$a^{k-p} a^p b^{k+1}$$

   y luego al bombear v una vez tenemos

   $$uv^2 w = a^{k-p} a^p a^p b^{k+1} = a^k a^p b^{k+1} = a^{k+p} b^{k+1}$$

   y como $p > 1$ tenemos $k+p > k+1$ con lo que $uv^2 w \notin L$ contradiciendo el Lema de Bombeo. Esa contradicción es consecuencia de haber asumido que L era en efecto Lenguaje Regular, por tanto no puede serlo.