Universidad de Valladolid

Departamento de Informática

Teoría de autómatas y lenguajes formales . 2º I.T.Informática. Gestión.

Examen de segunda convocatoria. 7 de septiembre de 2009

Apellidos, Nombre.. **Grupo:** ....

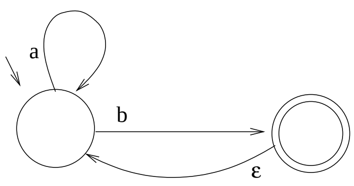
Notas para la resolución del examen: Responder a los ejercicios 3, 4 y 5 en la misma página del enunciado. Responder y entregar por separado los ejercicios 6 y 7.

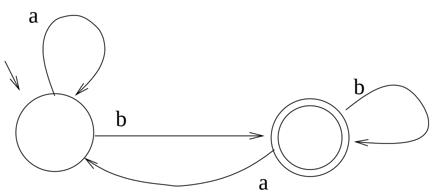
$\boxed{1}$ **(2 p.)** Sea L_1 el lenguaje dado por la expresión $(a^*b)^+$ y L_2 el reconocido por el autómata

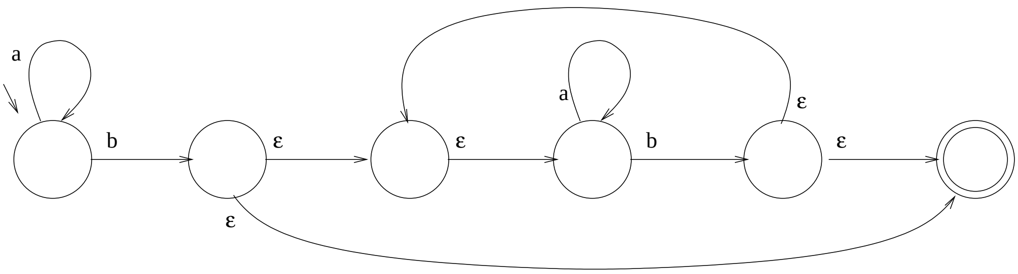
	a	b
$\to 1$	2	1
(2)		1

Obtenga razonadamente un reconocedor finito determinista mínimo cuyo lenguaje reconocido sea $L_1 \cup L_2$

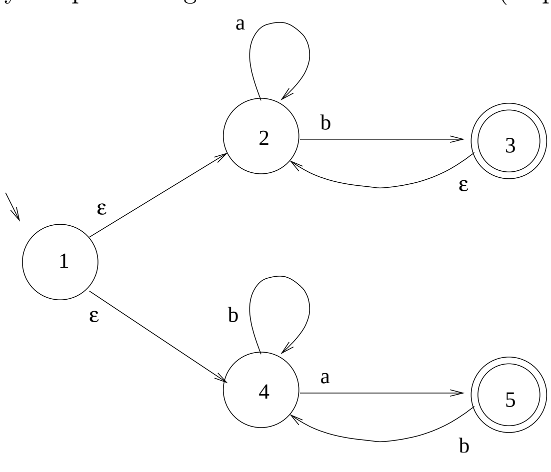
Solución: Se construye un reconocedor para el lenguaje L_1 (no necesariamente determinista). Por ejemplo cualquiera de los siguientes: (el segundo de ellos es determinista y mínimo)







Para L_2 ya se tiene un reconocedor. Se construye un reconocedor para la unión como muestra la figura siguiente, y se aplica el algoritmo de determinación: (empleamos aquí el primero de los anteriores):



	a	b	ϵ
$\to 1$			$2,4$
2	2	3	
(3)			2
4	5	4	
(5)		4	

	a	b	
$\to A$	B	C	$\underline{1}, 2, 4$
(B)	D	C	$\underline{2}, \underline{5}$
(C)	B	C	$\underline{3}, \underline{4}, 2$
D	D	E	$\underline{2}$
(E)	D	E	$\underline{3}, 2$

Ahora debe aplicarse el algoritmo de determinación, o bien calcular cadenas que distingan cada par de estados para comprobar que el obtenido es mínimo:

$$\mathcal{P}_0 = \{\{A, D\}, \{B, C, E\}\}$$
$$\mathcal{P}_1 = \{\{A\}, \{D\}, \{B, E\}, \{C\}\}$$
$$\mathcal{P}_2 = \{\{A\}, \{D\}, \{B\}, \{E\}, \{C\}\}$$

B	ϵ			
C	ϵ	a		
D	a	ϵ	ϵ	
E	ϵ	ba	a	ϵ
	A	B	C	D

NOTA: **no** es necesario calcular una expresión regular para L_2, que es $b^*a(bb^*a)^* = (b|ab)^*a$ y **no es** $(b^*a)^+$ (compruébese con la cadena aa, por ejemplo).

$\boxed{2}$ **(4 p.)** Denotemos por w^R al reflejado de una cadena w y por L^R al reflejado del lenguaje L. Se definen

$$\Diamond L := LL^R$$

y

$$\bowtie L := \{ww^R \ / \ w \in L\}$$

Evidentemente $\bowtie L \subseteq \Diamond L$ para cualquier lenguaje.

a. (0'5 p.) Calcule $\Diamond$ y $\bowtie$ para $L_3 = \{abb, b\}$ y $L_4 = ab^*$

Solución:

$\Diamond L_3 = \{abbbba, abbb, bbba, bb\}$ $\qquad \bowtie L_3 = \{abbbba, bb\}$

$\Diamond L_4 = ab^*b^*a = ab^*a$ $\qquad \bowtie L_4 = \{ab^nb^na \ / \ n \geq 0\} = a(bb)^*a$

b. (0'75 p.) Demuestre que $L_5 = \{a^nb^{n+m}c^m \ / \ n, m > 0\}$ es independiente de contexto.

Solución: La gramática independiente de contexto siguiente genera L_5:

$S \to AC \qquad A \to aAb \mid ab \qquad C \to bCc \mid bc$

c. (1 p.) Justifique por qué, si L es regular, entonces $\bowtie L$ es independiente de contexto.

(INDICACIÓN: construya un autómata con pila para $\bowtie L$ a partir de un reconocedor para L)

Solución: Si L es regular, existe un reconocedor finito determinista para L.

Las cadenas de $\bowtie L$ deben tener una primera parte de L y una segunda parte que sea la reflejada de la primera. Se puede utilizar el reconocedor para caracterizar las cadenas de L e ir almacenando los caracteres leídos una pila, de forma que cuando el reconocedor llegue a un estado final, el autómata a pila que se construye pase a un estado en el que se intente vaciar la pila leyendo el resto de los caracteres de entrada. Así la pila sólo se podrá vaciar si se encuentra en la entrada la reflejada de primera parte de la cadena.

Sea el reconocedor $R = (\Sigma, Q, q_1, f, F)$. El autómata a pila que se construye será:

$P = (\Sigma, Q \cup r_v, q_1, \Gamma = \Sigma \cup Z, Z, f_P, \emptyset)$ en el que la función de transición se define del siguiente modo:

$f_P(q_1, a, Z) := (f(q_1, a), a), \ \forall a \in \Sigma :$ desde el estado inicial, con el estado se imita el comportamiento del RFD, y se sustituye el símbolo inicial de la pila por el símbolo de entrada

$f_P(q, a, b) := (f(q, a), ab), \ \forall a, b \in \Sigma, q \in Q :$ se imita el comportamiento del RFD con los estados, y se apila el símbolo de entrada

Se añade (r_v, a) a $f_P(q_f, \epsilon, a), \ \forall a \in \Sigma \cup \{Z\}, q_f \in F :$ siempre que se llegue a un estado de aceptación del RFD, el autómata con pila puede pasar al estado de vaciado de pila, sin leer entrada y sin tocar el contenido de la pila

$f_P(r_v, a, a) := (r_v, \epsilon), \ \forall a \in \Sigma :$ en el estado de vaciado de pila, se saca un símbolo de la pila si coincide con el símbolo de entrada

$f_P(r_v, \epsilon, Z) := (r_v, \epsilon) :$ si la cadena vacía está en L, q_1 es final y hay que vaciar la pila para reconocer ϵ como cadena de $\bowtie L$

Con esta construcción, se tendrá $LV(P)$, el lenguaje aceptado por vaciado de pila de P es $\bowtie (L)$

d. (0'75 p.) Si L es independiente de contexto ¿es siempre $\bowtie L$ independiente de contexto?

Justifique la respuesta.

Solución: No siempre.

Por ejemplo, L_4 es independiente de contexto y $\bowtie L_4$ también, mientras que L_5 es independiente de contexto pero $\bowtie L_5$ no lo es:

$\bowtie L_5 = \{a^nb^{n+m}c^mc^mb^{n+m}a^n \ / \ n, m \geq 0\}$. Para ver que no es independiente de contexto, basta comprobar qeu no cumple el lema de bombeo (versión "fuerte") correspondiente: para cualquier entero positivo N, la cadena $z = a^Nb^{2N}a^N$ está en $\bowtie L_5$ y tiene longitud mayor que N. Cualquier descomposición de z en $xyuvw$ en la que $0 < |yv|$ y $|yuv| \leq N$ y en la que se puedan "bombear" simultáneamente y y v permaneciendo en el lenguaje, exige que tanto y como v estén compuestas por un único símbolo, ya que de otro modo el bombeo alternaría aes y bes más veces. Por lo tanto

- o bien y, v están en el prefijo formado por aes
- o bien y está en el prefijo de aes y v en la subcadena de bes
- o bien y está en la subcadena de bes y v en el sufijo de aes
- bien ambas están en el sufijo de aes

En cualquiera de los casos, la repetición de y y v "desequilibraría" los números de aes y bes, que deben permanecer iguales, y con el mismo número de aes en los extremos.

e. (0'5 p.) Justifique por qué, si L es recursivo, entonces $\bowtie L$ es recursivo.

Solución: Se trata de justificar la existencia de un algoritmo caracterizador para $\bowtie L$ sabiendo que existe uno para L.

Se puede construir comprobando en primer lugar que la cadena de entrada consta de dos mitades, siendo la segunda la reflejada de la primera, y borrando la segunda mitad al tiempo que se hace la comprobación. (En términos de máquina de Turing, se realiza un proceso de marcado del primer carácter, desplazamiento al último, y borrado de éste si coincide, retroceso hasta el primero no marcado, y repetición de este proceso hasta que no queden símbolos sin marcar. Si se ha conseguido, ya se sabe que la cadena era de la forma requerida, y se recorre la parte marcada recuperando su contenido original, que deberá ser entonces la primera mitad). Si no se ha conseguido, se para sin aceptar.

Si se ha conseguido, a continuación se aplica el algoritmo caracterizador de L a la primera mitad, devolviendo su resultado.

f. (0'5 p.) Justifique por qué, si L es recursivamente numerable, entonces $\bowtie L$ es recursivamente numerable.

Solución: Se trata, por ejemplo, de construir un algoritmo generador de $\bowtie L$ a partir de un algoritmo generador de L.

Simplemente se aplica el generador de L, y, cada vez que éste genere una cadena, se escribe en la salida la concatenación de ésta con su reflejada. (En términos de máquinas de Turing, la cadena que escribe la generadora de L debe escribirse además en una cinta o zona intermedia y luego releer esta zona en orden inverso para concatenarlo a continuación en la cinta de salida).

3 **(0.5 p.)**

a. Para cualesquiera lenguajes L_1, L_2, L_3 es cierta una de las dos contenciones siguientes. ¿Cuál?

- $L_1(L_2 \cap L_3) \subseteq L_1L_2 \cap L_1L_3$
- $L_1L_2 \cap L_1L_3 \subseteq L_1(L_2 \cap L_3)$

Solución: La primera afirmación es cierta.

b. Dé un ejemplo de tres lenguajes no vacíos L_1, L_2 y L_3 tales que $L_1(L_2 \cap L_3) \neq L_1L_2 \cap L_1L_3$

Algunas soluciones posibles:

L_1	L_2	L_3	$L_2 \cap L_3$	$L_1(L_2 \cap L_3)$	L_1L_2	L_1L_3	$L_1L_2 \cap L_1L_3$
$a\|ab$	c	bc	$\emptyset$	$\emptyset$	$ac\|abc$	$abc\|abbc$	abc
$\epsilon\|a$	a	ϵ	$\emptyset$	$\emptyset$	$a\|aa$	$\epsilon\|a$	a
a^*	a^+	ϵ	$\emptyset$	$\emptyset$	a^+	a^*	a^+
a^*	$(aa)^*$	$a(aa)^*$	$\emptyset$	$\emptyset$	a^*	$a(aa)^*$	$a(aa)^*$

4 **(1 p.)** Construya la Tabla de Análisis Sintáctico Predictivo para la gramática siguiente, una vez eliminados los símbolos y reglas inútiles, y especificando los primeros y siguientes de cada auxiliar.

$$
\begin{array}{lllllllllll}
S & \rightarrow & ABC \mid AP \mid QP & A & \rightarrow & C \mid bA & B & \rightarrow & aAS \mid b & C \rightarrow dC \mid \epsilon \\
P & \rightarrow & RA & R & \rightarrow & RP \mid P & Q & \rightarrow & aQ \mid \epsilon &
\end{array}
$$

Gramática sin reglas inútiles:

Solución:

$S \rightarrow ABC$ (P y R son no terminables
$A \rightarrow C \mid bA$ y Q resulta inaccesible
$B \rightarrow aAS \mid b$ depués de eliminarlos)
$C \rightarrow dC \mid \epsilon$

Primeros	S	A	B	C			
	b	b	a	d			
	d	d	b	ϵ			
	a	ϵ					

Siguientes	S	A	B	C			
	$\$$	a	d	$\$$			
	d	b	$\$$	d			
		d		a			
				b			

TASP	a	b	d	$\$$
S	$S \rightarrow ABC$	$S \rightarrow ABC$	$S \rightarrow ABC$	
A	$A \rightarrow C$	$A \rightarrow bA$, $A \rightarrow C$	$A \rightarrow C$	
B	$B \rightarrow aAS$	$B \rightarrow b$		
C	$C \rightarrow \epsilon$	$C \rightarrow \epsilon$	$C \rightarrow dC$, $C \rightarrow \epsilon$	$C \rightarrow \epsilon$

5 **(0'5 p.)** Determine los pivotes de las formas sentenciales siguientes respecto a la gramática

$$N \rightarrow 11 \mid 1001 \mid N0 \mid NN$$

- $NN11$: pivotes:
  Solución:
  NN ($\underline{NN}11$) con la regla $N \rightarrow NN$ (de la derivación m.d. $N \Rightarrow NN \Rightarrow N11 \Rightarrow NN11$)
  y
  11 ($NN\underline{11}$) con la regla $N \rightarrow 11$ (de la derivación m.d. $N \Rightarrow NN \Rightarrow NNN \Rightarrow NN11$)

- 1111 : pivotes:
  Solución:
  la primera subcadena 11 ($\underline{11}11$) con la regla $N \rightarrow 11$ (de la derivación m.d. $N \Rightarrow NN \Rightarrow N11 \Rightarrow 1111$)
  (pivote único)