

## TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES

SEPTIEMBRE 2009

**Final**

---

NORMAS :

- La duración del examen es de 2 horas y 30 minutos.
  - **Todos los ejercicios se entregarán en hojas separadas.**
  - El examen tipo test cuenta hasta 2 puntos sobre la nota total.
  - Además de obtener un mínimo de 5 puntos en el global del examen, para poder aprobar es necesario obtener un mínimo de 1.3 puntos entre las dos primeras preguntas y un mínimo de 1.3 puntos entre las dos últimas.
1. (a) (0.9 puntos) Define una gramática regular (según la definición dada en clase) para el lenguaje compuesto de todas las palabras que cumplen todas las características siguientes:
- las palabras del lenguaje están formadas por las letras  $a$ ,  $b$  y  $c$
  - el número de  $a$ 's y de  $b$ 's es par
  - a la izquierda de cada  $a$  siempre hay una  $c$
  - la palabra vacía pertenece al lenguaje

**Solución:**

Una posible solución podría ser la siguiente:

$$\begin{aligned} G &= (\{S, A, B, C, D, E, F, G\}, \{a, b, c\}, S, P) \text{ con} \\ P &= \{S ::= aD|bB|cS|\lambda; \\ A &::= aE|bC|cA; \\ B &::= acC|bS|cB; \\ C &::= aG|bA|cC; \\ D &::= cA; \\ E &::= cS; \\ F &::= cC; \\ G &::= cB\} \end{aligned}$$

- (b) (0.7 puntos) Decide de que tipo de la jerarquía de Chomsky es la siguiente gramática y describe (en palabras) el lenguaje que define.

$$\begin{aligned} G &= (\{S, A, B, C\}, \{a, b, c\}, S, P) \text{ con} \\ P &= \{S ::= aAC|bS|cS|\lambda; \\ AC &::= abB|bAC|cAC; \\ bB &::= aS|bbB|cbB\} \end{aligned}$$

**Solución:**

Esta gramática es del tipo 0. El lenguaje que define esta compuesto por todas las cadenas que contienen los caracteres a, b, y c y, además, el número de a's es múltiplo de 3. La palabra vacía también pertenece al lenguaje.

2. (1.6 puntos) Demuestra si el lenguaje reconocido por el autómata finito determinista  $A_1$  es el mismo (o uno diferente) que el lenguaje reconocido por la expresión regular  $(bb)^*(a(aa)^*a + \lambda)$ . Todos los pasos para la resolución del ejercicio tienen que estar justificados utilizando algoritmos vistos en clase. No es obligatorio, pero sí conveniente, dar una pequeña explicación de qué se hace y porqué se hace.

$$A_1 = (\{a, b\}, \{q_0, q_1, q_2, q_3, q_4, q_5\}, f_1, q_0, \{q_0, q_4\})$$

$f_1$	a	b
$\rightarrow^* q_0$	$q_1$	$q_2$
$q_1$	$q_4$	$q_3$
$q_2$	$q_5$	$q_0$
$q_3$	$q_3$	$q_3$
$*q_4$	$q_1$	$q_3$
$q_5$	$q_5$	$q_3$

### Solución:

Vamos a minimizar el autómata para obtener el autómata finito determinista mínimo (y único).

$$Q/E_0 = \{\{q_0, q_4\}, \{q_1, q_2, q_3, q_5\}\}$$

$$Q/E_1 = \{\{q_0, q_4\}, \{q_1\}, \{q_2\}, \{q_3, q_5\}\}$$

$$Q/E_2 = \{\{q_0\}, \{q_4\}, \{q_1\}, \{q_2\}, \{q_3, q_5\}\}$$

$$Q/E_3 = \{\{q_0\}, \{q_4\}, \{q_1\}, \{q_2\}, \{q_3, q_5\}\} = Q/E$$

El autómata mínimo es

$f_1$	a	b
$\rightarrow^* A$	B	C
B	E	D
C	D	A
D	D	D
*E	B	D

Ahora, dada la expresión regular  $(bb)^*(a(aa)^*a + \lambda)$ , definimos el AFND correspondiente

$$A_2 = (\{a, b\}, \{A, B, C, D, E, F, G\}, f_2, S, \{G\})$$

$f_2$	a	b	$\lambda$
$\rightarrow A$		{B}	{C}
B		{C}	
C	{D}		{A,G}
D	{E}		{F}
E	{F}		
F	{G}		{D}
*G			

Ahora, obtendremos un autómata finito determinista. Para ello, partimos de la  $clausura_\lambda(A) = \{A, C, G\}$ .

$f_2$	a	b
$\rightarrow^* \{A, C, G\}$	{D,F}	{B}
{D,F}	{E,G}	$\emptyset$
{B}	$\emptyset$	{A,C,G}
*{E,G}	{D,F}	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$

Renombramos los estados

$f_2$	a	b
$\rightarrow^* q_0$	$q_1$	$q_2$
$q_1$	$q_3$	$q_4$
$q_2$	$q_4$	$q_0$
* $q_3$	$q_1$	$q_4$
$q_4$	$q_4$	$q_4$

Vamos a minimizar el autómata para obtener el autómata finito determinista mínimo (y único).

$$Q/E_0 = \{\{q_0, q_3\}, \{q_1, q_2, q_4\}\}$$

$$Q/E_1 = \{\{q_0, q_3\}, \{q_1\}, \{q_2\}, \{q_4\}\}$$

$$Q/E_2 = \{\{q_0\}, \{q_3\}, \{q_1\}, \{q_2\}, \{q_4\}\}$$

$$Q/E_3 = \{\{q_0\}, \{q_3\}, \{q_1\}, \{q_2\}, \{q_4\}\} = Q/E$$

El autómata ya es mínimo. Como podemos comprobar, los autómatas mínimos  $A_1$  y  $A_2$  son isomorfos. Por lo tanto, hemos demostrado que la gramática y la expresión regular dados en el enunciado definen el mismo lenguaje.

3. (1.6 puntos) Demuestre que el siguiente lenguaje NO es regular:

$$L = \{a^{2^i} \mid i \geq 1\}$$

**Solución:**

Cada lenguaje regular cumple el lema de bombeo. Demostraremos que este lenguaje no cumple el lema de bombeo, por lo que no puede ser regular. Un lenguaje cumple el lema si todas sus palabras  $x$  que son más grandes o iguales que una constante  $n$ , se pueden descomponer de forma  $x = u.v.w$  y tal que se cumple lo siguiente:

1.  $|u.v| \leq n$
2.  $|v| > 0$
3. para todo  $i$ :  $u.v^i.w$  es una palabra que pertenece también al lenguaje.

Eligimos la palabra  $x = a^{2^n}$ . Sea cual sea la constante  $n$ , obviamente  $x$  es más larga que  $n$ . Por tanto, si  $L$  fuese regular,  $x$  debe poder descomponerse como descrito arriba. Sin embargo, veremos que no hay ninguna descomposición de  $x$  que cumple los criterios.

Analizamos: Supongamos que tenemos una descomposición de  $x = u.v.w = a^k.a^p.a^q$  que cumple los dos primeros criterios, es decir,  $k + p \leq n$  y  $p > 0$ . Entonces, para ser una descomposición válida se tendría que verificar que la palabra  $z = u.v^2.w = a^k.a^{2p}.a^q = a^{2^n + p}$  pertenece a  $L$  (según la tercera condición). Demostraremos por la longitud de la palabra  $z$ , que eso es imposible.

Primero, es obvio que  $|z| > |x|$ , ya que  $p > 0$ . Por tanto,  $z$  no puede ser igual a ninguna palabra de  $L$  que tiene una longitud menor o igual a  $x$ .

Segundo, miramos si  $z$  puede ser igual a alguna palabra de  $L$  con longitud mayor a  $x$ . La siguiente palabra en  $L$ , después de  $x$ , en términos de longitud es  $y = a^{2^{(n+1)}}$ . La longitud de  $y$  es  $|y| = 2^{(n+1)} = 2^n \cdot 2 = 2^n + 2^n$ . Por otro lado, la longitud de  $z$  es  $|z| = 2^n + p$ . Dado que  $2^n < n$  y por la condición del lema  $n \geq p$ , se cumple que  $|z| < |y|$ . Es decir,  $z$  no puede ser igual a ninguna palabra en  $L$  que es igual o más grande que  $y$ . Dado que en  $L$  no existe ninguna palabra más grande que  $x$  y más pequeña que  $y$ , se concluye que  $z$  no pertenece a  $L$ . Por tanto, no es posible descomponer la palabra  $x$  según el lema de bombeo. Por tanto, el lenguaje no cumple el lema de bombeo y no puede ser regular.

4. (1.6 puntos) **Definición:** Sea  $w = xy$ . Se dice que  $x$  es **cabeza** de  $w$ .

Por ejemplo, la palabra  $abcd$  tiene las cabezas  $\lambda, a, ab, abc, abcd$ .

---

Dado el siguiente lenguaje sobre el alfabeto  $\{a, b, c, d\}$ .

$$L = \{w \mid n_{ab}(w) = n_{cd}(w) \text{ y } n_{ab}(x) \geq n_{cd}(x) \text{ para cualquier cabeza } x \text{ de } w\}.$$

Ejemplos:

$abcd \in L$	$abcacd \in L$	$abcdabc \notin L$
$cdab \notin L$	$abcdcdab \notin L$	$aaabbccd \in L$
$aabcbabcbacd \in L$		

Diseñe un autómata a pila que reconozca el lenguaje  $L$ . Su solución no debe tener más de cinco estados.

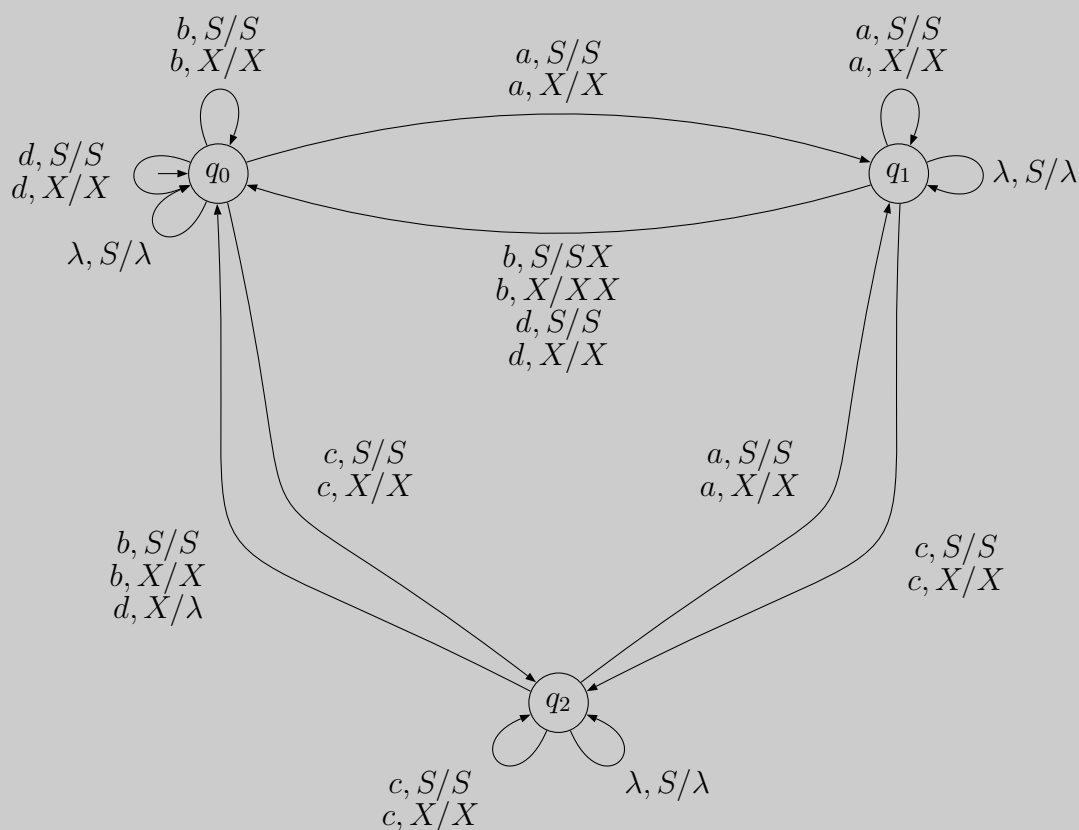
Describe su funcionamiento.

### Solución:

La solución consiste en un autómata a pila con tres estados que reconoce por vaciado de pila:

$$AP = (\{a, b, c, d\}, \{S, X\}, \{q_0, q_1, q_2\}, q_0, S, f)$$

El autómata a pila  $AP$  funciona de la siguiente manera: Para cada  $ab$  se apila un símbolo  $X$  en la pila (estados  $q_0$  y  $q_1$ , y para cada  $cd$  se elimina dicho símbolo  $X$  de la pila (estados  $q_0$  y  $q_2$ ). Nunca se podría quitar un  $X$  de la pila que no apilado anteriormente, de esta forma está asegurado en cada momento que haya más  $ab$  leídos que  $cd$  (segunda condición).



En el estado  $p_0$  no se ha leído ningún símbolo  $a$  ni  $c$  que son los comienzos de las cadenas  $ab$  y  $cd$ . Si  $AP$  lee un símbolo  $a$ , cambia al estado  $q_1$  atento por si viene un símbolo  $b$  para apilar la marca  $X$ .

$AP$  se mantiene en el estado  $q_1$  mientras llegan más  $a$ . Con un  $c$  cambia al estado  $q_2$  y en el caso de un  $d$  vuelve al principio sin apilar nada (falsa alarma, después del  $a$  no viene ningún símbolo  $b$ ).

Lo mismo pasa en el estado  $q_2$ ,  $AP$  ha consumido el símbolo  $c$  y se encuentra a la espera de un símbolo  $d$  para quitar una marca  $X$  de la pila. Se mantiene en  $q_2$  mientras sigue consumiendo símbolos  $c$ , porque todavía podría llegar un  $d$ . Si no viene ningún  $d$  otra vez es una falsa alarma y  $AP$  cambia al estado  $q_1$  al leer un símbolo  $a$  (comienzo de una cadena  $ab$ ) o al estado inicial  $q_0$  al consumir un  $b$ .

Finalmente, hay que añadir una transición que elimina el símbolo inicial  $S$  de la pila a todos los estados, ya que se desconoce el último símbolo de la entrada, y por tanto, el estado en que se encuentra  $AP$  después de consumir toda la palabra.

5. (1.6 puntos) Dada la siguiente máquina de Turing

$$MT = (\{1, B\}, \{1\}, B, Q, q_0, f).$$

En la cinta se encuentra la siguiente información (y con la cabeza lectora de  $MT$  encima del primer 1):

$$\dots BBB \underbrace{1B1B1B1B1 \dots 1B1B1B1}_{n \text{ símbolos } 1} BBB \dots$$

Esta máquina de Turing tiene que transformar dicha información en el siguiente contenido de la cinta (con la cabeza lectora encima del primer 1):

$$\dots BBB \underbrace{1111 \dots 1111}_{n \text{ símbolos } 1} BBB \dots$$

donde la cantidad de símbolos 1 es exactamente la misma que en la configuración inicial.

Define el conjunto finito de estados  $Q$  con sus correspondientes transiciones  $f$  de tal forma que  $MT$  resuelva el problema (no está permitido modificar los componentes dados de  $MT$ ).

Describe su funcionamiento.

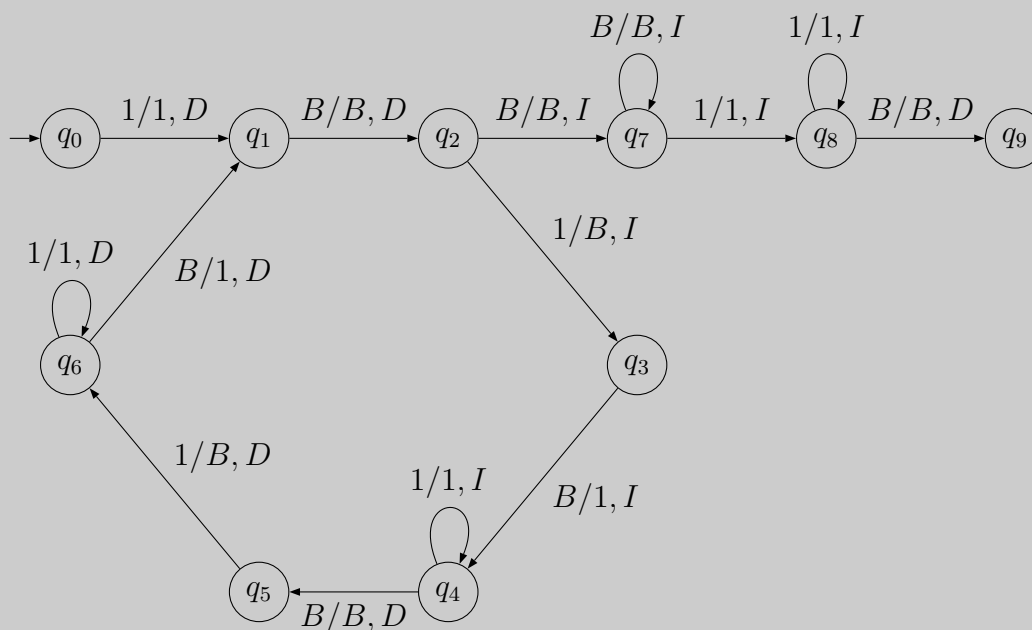
### Solución:

$MT$  tiene que ejecutar el siguiente algoritmo:

1. Ir dos símbolos hacia la derecha ( $q_0 - q_2$ )
2. Reemplazar el 1 por el símbolo blanco e ir dos símbolos hacia la izquierda ( $q_2 - q_3$ )
3. Escribir el 1 e ir hacia la izquierda ( $q_3 - q_4$ )

4. Ir hacia la izquierda, saltando todos los 1 ( $q_4$ )
5. Al principio de la palabra, eliminar el 1, reemplazándolo por el símbolo blanco ( $q_4 - q_6$ )
6. Ir hacia la derecha, saltando todos los 1 ( $q_6$ )
7. Escribir el símbolo 1 ( $q_6 - q_1$ )
8. Empezar desde el paso 1.

Si el segundo símbolo después del último 1 es un símbolo blanco, se ha terminado. Lo único que falta, es mover la cabeza lectora al principio de la palabra. En el estado ( $q_7$ ) retrocede la final de la palabra, en  $q_8$  la principio hasta el símbolo blanco que se encuentra a la izquierda de la palabra. En la transición de ( $q_8$ ) a ( $q_9$ ) se sitúa la cabeza lectora al primer símbolo de la palabra.



Por tanto:

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$$

con las transiciones  $f$  según el grafo.