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

Examen de convocatoria ordinaria, 4 de junio de 2007

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

Firma:

---

1 **(12 p.)** Dado el reconocedor finito

	0	1
$\rightarrow p$	p	q
(q)	q	p

1. Demostraremos que el lenguaje aceptado es $L = \{w \in (0|1)^* \ / \ |w|_1 \text{ es impar } \}$
   Se demostrará por inducción sobre la longitud de w, n, que

   $$f^*(p, w) = q \text{ si y solamente si } w \text{ tiene un número impar de unos}$$

   B *Base: Si $n = 0$, entonces $w = \epsilon$; $f^*(p, w) = p$ y w tiene 0 unos, así que el enunciado es cierto.*
   H *Suponemos que el enunciado es cierto para $n - 1$ con $n \geq 1$*
   P *Sea w de longitud n. Hay dos casos: $w = x1$ o $w = x0$ donde x es el prefijo de w de longitud $n - 1$.*

   - $w = x1$: *Si w tiene un número par de unos, entonces x tiene un número impar, por* [ ] *. Luego $f^*(p, x) = q$ por* [ ] *. Por lo tanto $f^*(p, w) = p$ por* [ ] *.*
     Si w tiene un número impar de unos, entonces x tiene un número par, por [ ] *. Luego $f^*(p, x) = p$ por* [ ] *. Por lo tanto $f^*(p, w) = q$ por* [ ]
     Así pues está probado el enunciado para este caso.
   - $w = x0$:

     _____

     _____

     _____

     _____

   Se pide completar los huecos de la demostración, justificando por qué cada paso es cierto. Los argumentos deben elegirse entre B (base de la inducción) , H (hipótesis de inducción) y :
   F la definición de f^* (la extensión de la función de transición a $Q \times \Sigma_E^*$)
   Q la propiedad $f^*(q, xy) = f^*(f^*(q, x), y))$
   R hechos conocidos sobre enteros o cadenas (por ejemplo: "la suma de un par y un impar es impar", "si $x = yz$ el número de aes de x es la suma del número de aes de y más el de z" , etc.)

2. Renombrando los estados $1 = p$ y $2 = q$ ¿Cuáles son las expresiones regulares siguientes?

   $R_{11}^0 =$ _____ $R_{22}^1 =$ _____ $R_{12}^2 =$ _____

3. Como la constante del lema de bombeo podría ser 2, toda cadena $z \in L$ de longitud mayor o igual que 2 admite una descomposición uvw con $|uv| \leq 2$ de forma que v es no nula y bombeable. Descríbase quién podría ser v en cada uno de los casos siguientes y justifíquese por qué $uv^iw \in L \ \ \forall i \geq 0$:

   - $z = 00z_1$ _____
   - $z = 10z_1$ _____
   - $z = 11z_1$ _____

2 **(10 p.)** 1. Hallar un RF determinista equivalente (especificando el significado de cada estado) a

	0	1	ϵ
$\rightarrow 1$	5		3
(2)		3	
3			4
4	2	4	
5		3	

2. Hallar un RF determinista mínimo equivalente al anterior o probar que ya era mínimo:

3 **(12 p.)** Sea G la gramática

$$\begin{array}{lcl} S & \rightarrow & bPP \mid TRU \\ P & \rightarrow & Q \mid a \mid \epsilon \\ Q & \rightarrow & PS \end{array} \qquad \begin{array}{lcl} R & \rightarrow & Ra \mid UR \\ U & \rightarrow & QP \mid \epsilon \\ T & \rightarrow & aRb \mid PP \end{array}$$

1. Los símbolos terminables de G son: _____
2. Los símbolos accesibles de G son: _____
3. Los símbolos útiles de G son: _____
4. Una gramática equivalente sin símbolos inútiles es por lo tanto:

5. Si al resultado anterior se le eliminan reglas épsilon se obtiene:

6. Una gramática equivalente a G sin reglas épsilon, ni reglas simples ni símbolos inútiles es:

4 **(6 p.)** Para diseñar un AP cuyo lenguaje aceptado por vaciado de pila sea

$\{w \in (a|b)^* \ / \$ si x es prefijo de w entonces $|x|_a \leq |x|_b\}$

se comienza poniendo las transiciones (A es el símbolo inicial de pila)

$$(r1) \quad f(q_1, b, A) = \{(q_1, BA)\} \qquad (r2) \quad f(q_1, b, B) = \{(q_1, BB)\} \qquad (r3) \quad f(q_1, a, B) = \{(q_1, \epsilon)\}$$

1. Explica el significado de estas transiciones

2. Completa la definición del AP

3. Propón una gramática que genere el lenguaje

5 **(6 p.)** Se tiene un máquina de Turing M que genera un lenguaje L. Se sabe que L no es recursivo, y que las primeras cadenas generadas son (en ese orden) aa, a, aba, a, ab, $a^{47}b$, b^3ab. Se construye otra máquina de Turing (M'), usando ésta, de la siguiente manera:

```
var long : integer;
begin(* M' *)
     long := 2; writeln ('aa');
     repeat
            hacer que M genere la siguiente x
            if M está parada then exit (parar)
            if length(x) >long then
               begin writeln (x); long := length(x) end
     until (1=0)
end (* M' *)
```

Evidentemente, esta máquina muestra un subconjunto de cadenas de L.

¿Es este lenguaje recursivamente numerable?. (Si/No/Puede serlo o no ... justifíquese).

¿Es este lenguaje recursivo?. (Si/No/Puede serlo o no ... justifíquese).

6 **(6 p.)** Sean L_1, L_2 y L_3 tres lenguajes

Probar que "Si $L_2 = L_3$ entonces $L_1\,L_2 = L_1\,L_3$"

Probar que el recíproco ("Si $L_1\,L_2 = L_1\,L_3$" entonces $L_2 = L_3$) no es cierto

7 **(8 p.)** Para la máquina de Turing sobre el alfabeto binario, estado inicial q_1, estado final q_2 y función de transición $f(q_1, 0) = (q_3, 0, \rightarrow)$; $f(q_1, 1) = (q_2, 1, \leftarrow)$; $f(q_3, 1) = (q_1, 1, \leftarrow)$

el lenguaje de parada es $L_P =$ _____

el lenguaje reconocido es $L_R =$ _____

una codificación de la máquina es _____

8 **(15 p.)** Calcular la TASP de la siguiente gramática, especificando los PRIMEROS y SIGUIENTES necesarios:

(1)	S	$\rightarrow$	bSc	(5)	Q	$\rightarrow$	ϵ	(9)	T	$\rightarrow$	PQe
(2)		$\mid$	TVb	(6)		$\mid$	a	(10)		$\mid$	$dReT$
(3)	P	$\rightarrow$	ϵ	(7)	R	$\rightarrow$	PQ	(11)	V	$\rightarrow$	$PcQd$
(4)		$\mid$	b	(8)		$\mid$	cSP	(12)		$\mid$	eR

Pr.								

Sg.						

TASP				

9 **(5 p.)** En el directorio por defecto se encuentran los siguientes ficheros: (todos de tipo texto)

1	aaa..	3	aaa..wq	5	ab	7	abc.txt	9	p.p	11	uno
2	aaabbb	4	aa.txt	6	abc.p.s	8	1.p	10	a.c	12	1a.pp

- La orden `ls -l | egrep -w "a*.*"` mostrará (*elíjase la opción correcta y complétese*):
  1. las líneas de `ls -l` correspondientes a los ficheros de número:

     _____
  2. las líneas de `ls -l` correspondientes a ficheros con determinado contenido:

     _____
  3. otra cosa:

     _____
- ¿Qué orden (similar a la anterior) mostraría lo mismo pero de los ficheros cuyo nombre sea *carácter punto carácter* (ej. 8, 9, 10)?

  _____

10 **(5 p.)** ¿Qué es `y.tab.h`?

11 **(15 p.)** *(Se exige una calificación mínima de 7 puntos para considerar la nota de la práctica)*

Elaborar programas fuente en Lex y Yacc para realizar las siguiente tarea: se parte de un programa Pascal correcto (compilado sin errores).

1. patiendo de un programa Pascal correcto (compilado sin errores), devuelve el mismo programa del que se han eliminado los comentarios
2. partiendo de un programa correcto, que no tiene comentarios, ni definiciones de registros (`record` ... `end`), ni sentencias `case`, -pero que puede tener subprogramas–, debe mostrar solamente una línea por cada par `begin` ... `end` del *programa principal* consituída por los números de línea en los que se encuentran estas palabras. En la línea del par que enmarca el programa principal pondrá `PRINCIPAL`.
   `repeat` ... `until` debe considerarse como un par `begin` ... `end`.
   EJEMPLO:

```
(* 1*) program pp
(* 2*) procedure q
(* 3*) procedure r
(* 4*) begin      end
(* 5*) begin
(* 6*) begin      end
(* 7*) begin      end
(* 8*) end
(* 9*)
(*10*) begin                                    SALIDA:
(*11*) begin
(*12*) begin
(*13*) end                                      (12-13)
(*14*) begin                                    (14-15)
(*15*) end                                      (11-16)
(*16*) end                                      (19-19)
(*17*) begin                                    (18-20)
(*18*) repeat                                   (17-21)
(*19*) begin end                                (10-22) PRINCIPAL
(*20*) until ....
(*21*) end
(*22*) end
```