



Apellidos, Nombre..... Grupo:

Firma:

1 (14 p.) La gramática

$$\begin{array}{ll} S \rightarrow AR & A \rightarrow a \mid bAA \\ R \rightarrow aBR \mid bAR \mid \epsilon & B \rightarrow b \mid aBB \end{array}$$

genera el lenguaje $L = \{w \in (a|b)^* \mid |w|_a = |w|_b + 1\}$.

1. Constrúyase un Autómata con Pila, con un solo estado, cuyo lenguaje aceptado por vaciado de pila sea L

2. Veremos que L no cumple el lema de bombeo de los regulares:

sea N una pretendida constante para el lema. ¿Qué cadena elegirías para z ? _____

Si se descompusiera $z = uvw$ de forma que $|uv| \leq N$ y $|v| > 0$, ¿quiénes serían u , v y w ?

$u =$ _____ $v =$ _____ $w =$ _____

¿qué valor de i habría que elegir para que $uv^i w \notin L$ _____

¿Por qué razón $uv^i w \notin L$? _____

Suponiendo que el apartado 1 y el razonamiento anterior está completado ¿qué se puede concluir sobre el tipo de L en la jerarquía de Chomsky? _____

2 (12 p.) 1. Enunciar un algoritmo para eliminar la recursión **directa** por la derecha de una gramática i.c. sin reglas inútiles, ni simples, ni reglas "épsilon":

Sustituir $\begin{array}{l} A \rightarrow \alpha_1 A \mid \dots \mid \alpha_p A \\ \quad \mid \beta_1 \mid \dots \mid \beta_q \\ \text{donde las } \beta_i \text{ no acaban por } A \end{array}$ por

2. Eliminar la recursión por la derecha de la gramática

$$\begin{array}{l} S \rightarrow aBB \mid a \\ B \rightarrow SB \mid BS \mid b \end{array}$$

3 (14 p.) Considérense los tres lenguajes siguientes, sobre el alfabeto $\{a, b\}$:

- $L_1 = (b|ba)^*$
- $L_2 = (bb^*a)^*b^*$
- $L_3 = (bb^*a)^*a(a|b)^*$

Mostrar, justificadamente, la relación que hay entre ellos.

- 4 (6 p.) Supongamos que un lenguaje verifica que $L^2 \subseteq L$. Demuestra que, entonces, para todo $n > 1$ se tiene que $L^n \subseteq L$.

Un ejemplo en el que $L^2 \subseteq L$: _____

Un ejemplo en el que $L^2 \not\subseteq L$: _____

- 5 (14 p.) 1. Para la máquina de Turing sobre el alfabeto binario, estado inicial q_1 , estados $\{q_1, q_2, q_3\}$, estado final q_2 y función de transición:

$$f(q_1, 0) = (q_2, 0, \rightarrow) ; f(q_1, 1) = (q_3, 1, \leftarrow) ; f(q_3, \hbar) = (q_1, \hbar, \rightarrow)$$

El lenguaje de parada es $L_P =$ _____

El lenguaje reconocido es $L_R =$ _____

Poniendo $X_1 = 0$, $X_2 = 1$, $X_3 = \hbar$, $D_1 = \leftarrow$, $D_2 = \rightarrow$

una codificación de la máquina es _____

2. Considérese el lenguaje formado por todas las codificaciones de máquinas de Turing cuyo lenguaje reconocido es no vacío:

$$L_{nv} = \{ \langle M \rangle \in (0|1)^* / L_R(M) \neq \emptyset \}$$

Supondremos que toda cadena de $(0|1)^*$ es una máquina de Turing: si no corresponde a una codificación, se entiende que se trata de una (la) máquina sin transiciones. Por ejemplo, 10101101011 y la codificación de la máquina del apartado anterior están en L_{nv} , pero ni 1010101011 ni 10 están en L_{nv} .

Describir un algoritmo reconocedor que justifique que L_{nv} es recursivamente numerable (se dispone de la Máquina de Turing Universal).

3. Dada un máquina de Turing M y una cadena w de entrada para M , se construye otra máquina de Turing M' con la siguiente "lógica":

```

var x: string
begin
  leer x (en la cinta de entrada)
  borrar x
  simular la operación de M sobre w
    (haciendo uso de la M.T. Universal)
  escribir SI (si M se ha parado sobre w)
end

```

¿Cuál es el lenguaje reconocido por M' ?

6 (15 p.) Calcular la TASP de la siguiente gramática, especificando los PRIMEROS y SIGUIENTES necesarios:

- | | | |
|------------------------------|------------------------------|---------------------------|
| (1) $S \rightarrow aSc$ | (5) $B \rightarrow \epsilon$ | (9) $D \rightarrow ABd$ |
| (2) $\quad \quad DEb$ | (6) $\quad \quad b$ | (10) $\quad \quad eCdD$ |
| (3) $A \rightarrow \epsilon$ | (7) $C \rightarrow BA$ | (11) $E \rightarrow AcBe$ |
| (4) $\quad \quad a$ | (8) $\quad \quad cSB$ | (12) $\quad \quad dC$ |

Pr.								

Sg.								

TASP				

7 (5 p.) Escribir órdenes grep que calculen cuántas líneas de la entrada:

1. contienen un punto _____
2. contienen dos puntos consecutivos _____
3. contienen exactamente dos puntos no necesariamente consecutivos _____

(Por ejemplo, para la entrada siguiente las salidas deben ser, respectivamente, 7, 4 y 3)

```
-rw-r--r-- 1 pepito users 0 may 30 19:34 aaa...
-rw-r--r-- 1 pepito users 0 may 30 19:35 aaa..
-rw-r--r-- 1 pepito users 0 may 30 19:36 aaabbb
-rw-r--r-- 1 pepito users 0 may 30 19:37 aaa..wq
-rw-r--r-- 1 pepito users 0 may 30 19:38 aa.txt
-rw-r--r-- 1 pepito users 0 may 30 19:39 ab..t.a
-rw-r--r-- 1 pepito users 0 may 30 19:40 abc.p.s
-rw-r--r-- 1 pepito users 0 may 30 19:41 ab.txt
```

8 (5 p.) ¿Qué es y.output?

9 (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 siguientes tareas (un programa o una combinación de fuente Lex y fuente Yacc para cada apartado):

1. partiendo de un programa Pascal correcto (compilable sin errores), devuelve solamente los comentarios
2. partiendo de un programa correcto, que no tiene comentarios, ni definiciones de registros (`record ... end`), ni sentencias `case`, debe mostrar solamente una línea por cada subprograma, constituida por el número de línea en que comienza la cabecera del mismo y los números de línea en que se encuentran las palabras reservadas BEGIN y END que lo enmarcan, más una línea con la misma información referida al programa principal.

EJEMPLO:

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

SALIDA:

```
3: (4-4) FUNCION
2: (5-8) PROCEDIMIENTO
1: (10-22) PRINCIPAL
```