



Apellidos, Nombre.....

--	--	--	--	--	--	--	--	--	--

1. Todos los alumnos deberán entregar esta hoja, grapada con las soluciones de los ejercicios 1,2, 3 y 4.
2. Los ejercicios 5, 6 y 7 se entregarán grapados aparte.
3. Comenzar cada problema en una nueva hoja y en el orden propuesto.

**1** (2'2 p.) Considérese la máquina de Mealy dada por la tabla

	0	1
$q_1$	$q_2/\bullet$	$q_3/\bullet$
$q_2$	$q_1/0$	$q_1/2$
$q_3$	$q_1/1$	$q_1/3$

1. Calcular la salida desde  $q_1$  para las entradas 011011, 0110111, 01101110 y 01101111.
2. Hallar una máquina de Moore mínima equivalente.
3. ¿Cómo se podría utilizar esta máquina para hacer un cambio de base 2 a base 4? (Indicación: intérpretese la entrada y la salida en orden inverso)

**2** (4 p.) 1. Sea  $G_1$  la gramática

$$G_1 : \begin{cases} S \rightarrow aS \mid aA \mid a \\ A \rightarrow aB \mid bS \\ B \rightarrow aB \mid bB \\ C \rightarrow aA \mid bC \end{cases}$$

Obtener  $\{x \in L(G_1) \mid |x| \leq 4\}$

2. Calcular un reconocedor finito determinista mínimo para  $L(G_1)$ . Aplicar el algoritmo de análisis para obtener una expresión de  $L(G_1)$ .
3. Calcular  $(a|ab)^*a \Delta L(G_1)$  (diferencia simétrica).
4. Obtener una gramática ( $G_2$ ) equivalente a  $G_1$  a partir del reconocedor determinista.
5. Sea  $G$  una gramática de tipo 3 por la derecha. ( $A_i \rightarrow aA_j \dots$ ) ¿Cómo son los árboles de derivación para  $G$ ? ¿Cómo son las formas sentenciales? ¿Cómo pueden localizarse los pivotes?
6. ¿Es  $G_1$  LALR(1)? (Indicación: considerar el apartado anterior y los análisis para las cadenas de  $L(G_1)$  de longitud  $\leq 4$ . ¿Cuáles son los posibles contenidos de la pila en un análisis con éxito?)
7. Considérese ahora la gramática también equivalente

$$G_3 : \begin{cases} S \rightarrow aS \mid aA \mid a \\ A \rightarrow aA \mid bS \end{cases}$$

¿Qué puede decirse sobre la ambigüedad de  $G_1$ ,  $G_2$  y  $G_3$ ?

8. ¿Qué lenguajes de tipo 3 son inherentemente ambiguos?

**3** (1 p.) Sean  $L_1 = \{x \in (a|b)^* / |x|_a = |x|_b\}$  y  $L_2 = \{a^n b^n / n \geq 0\}$

1. Demostrar que  $L_2$  no es regular.
2. Demostrar que  $L_2$  verifica el lema de bombeo de los lenguajes independientes del contexto.
3. Demostrar que  $L_1$  verifica el lema de bombeo de los lenguajes regulares.
4. Demostrar que  $L_1$  no es regular.

**4** (0'8 p.) 1. Dado un lenguaje  $L$  recursivamente numerable no recursivo, y un programa  $P_L$  diseñado para reconocer  $L$  ¿qué podemos garantizar sobre el comportamiento de  $P_L$ ?

Respuestas posibles:

- a)  $P_L$  no se parará nunca.
  - b) Ante cualquier posible cadena de entrada  $x$ ,  $P_L$  se parará (respondiendo afirmativa o negativamente).
  - c) Ante todas las cadenas de  $L$ ,  $P_L$  se parará respondiendo afirmativamente; pero es posible que para alguna cadena del complementario de  $L$ ,  $P_L$  no se pare.
  - d) Ante todas las cadenas de  $L$ ,  $P_L$  se parará respondiendo afirmativamente; pero con seguridad hay por lo menos una cadena del complementario de  $L$  para la que  $P_L$  no se para.
2. Supongamos que la empresa "Ultratemporal S.A." ha realizado un programa *superP* que, tomando como entrada un programa arbitrario  $P$  y una cualquiera de sus entradas  $x$ , determina si  $P$  se parará o no con tal entrada  $x$ . ¿Qué pasa entonces con el programa  $P_L$  del apartado anterior?
    - a) Que no podría existir, porque ... (completar)
    - b) Nada. No tiene nada que ver.
    - c) ¡Ah! pero ¿existe un lenguaje recursivamente numerable no recursivo?
    - d) Que no sería entrada válida para *superP*, porque ... (completar)

**5** (0'75 p.) Escribir en una línea la orden que conseguiría:

1. Mostrar los números de líneas vacías de un texto, con **grep**
2. Eliminar las líneas vacías de un fichero, con **sed**
3. Eliminar las líneas vacías de un fichero, con **awk**

**6** (1'25 p.) Usando la herramienta Yacc, construir un analizador sintáctico para las expresiones de sumas de vectores de números reales de hasta 10 elementos.

Los elementos de cada vector estarán separados por espacios en blanco y encerrados entre corchetes.

El analizador léxico necesario para resolver el problema se construirá utilizando la herramienta Lex.

El programa resultante deberá devolver el resultado de realizar las operaciones algebraicas indicadas en la expresión de la entrada sólo en el caso de que ésta sea correcta.

Por ejemplo:

Entrada ( [1.0 0 -1 1] + [1 0.1 -1 0] ) + [1 4.3 5 7.2]

Salida esperada [3.0 4.4 3 8.2]

**7** (0 p. pero de obligada contestación, breve, para considerar la práctica) Para los alumnos que hayan utilizado Yacc: ¿Qué tipo de estructura o estructuras has utilizado para almacenar los datos de salida en la práctica?

Para los alumnos que sólo hayan entregado la solución parcial en Lex: ¿De qué forma has eliminado los comentarios del programa de entrada?