



Apellidos, Nombre.....

Grid of 10 empty boxes for writing names.

- 1. Todos los alumnos deberán entregar esta hoja, grapada con las soluciones de los ejercicios 1 a 5.
2. El ejercicio 6 se entregará grapado aparte.
3. Entregar las respuestas en el orden propuesto.
4. Cualquier apartado puede ser resuelto sin haber resuelto los anteriores. Cualquier apartado o resultado del enunciado puede ser usado en otro aunque no haya sido resuelto.

1 (1'5 p.) Demostrar1 o refutar las siguientes igualdades entre lenguajes regulares:

- 1. L* - ε = L+
2. (a|b)*b = a*b(a*b)*

2 (2 p.)

- 1. Demostrar que L1 = {ambncn / m ≥ 3 ∧ n ≥ 0} no es regular.
2. Demostrar que L2 = {ambncp / m < 3 ∨ (m ≥ 3 ∧ n = p)} verifica el lema de bombeo de los lenguajes regulares.
3. Demostrar que L2 no es regular.

3 (1'5 p.)

- 1. Sea P un autómata a pila determinista. Sea L = LV(P) el lenguaje que P acepta por vaciado de pila. Demostrar que, si x, y ∈ L y x, y ≠ ε, entonces x no es un prefijo propio de y.
2. Deducir que {wwR / w ∈ Σ+} no es reconocible por vaciado de pila por ningún autómata a pila determinista (sea cual sea el alfabeto Σ).
3. Deducir que a+ no es reconocible por vaciado de pila por ningún autómata a pila determinista. ¿Es a+ reconocible por estado final por algún autómata a pila determinista? ¿Y a*?

4 (1'5 p.) Sea L un lenguaje. Se define L2 = {x ∈ L / |x| = 2} (conjunto de cadenas de L que tienen longitud par). Respóndase justificadamente a las siguientes preguntas. Las respuestas deben ser: SI (SIEMPRE), NO (NUNCA), PUEDE SERLO O NO SERLO.

- 1. Si L es independiente de contexto, ¿es L2 independiente de contexto? ¿Es L2 regular?
2. Si L es recursivo, ¿es L2 recursivo? ¿Es L2 recursivamente numerable?
3. Si L es recursivamente numerable no recursivo, ¿es L2 recursivamente numerable? ¿Es L2 recursivo?

1usando reconocedores finitos o equivalencias conocidas de expresiones regulares

- 5 (2 p.) Calcúlese la TASP para la siguiente gramática, especificando primeros y siguientes, en los espacios reservados para ello (completando las columnas que faltan).

$$G : \begin{cases} S \rightarrow AB \mid bDBE \\ A \rightarrow aSe \\ E \rightarrow e \\ B \rightarrow bBd \mid UV \\ U \rightarrow u \mid \varepsilon \\ V \rightarrow k \mid \varepsilon \\ D \rightarrow aD \mid k \end{cases}$$

Primeros	<i>S</i>	<i>A</i>	<i>E</i>	<i>B</i>	<i>U</i>	<i>V</i>	<i>D</i>	Siguientes	<i>S</i>	<i>A</i>	<i>E</i>	<i>B</i>	<i>U</i>	<i>V</i>	<i>D</i>

TASP	a	b	d	
<i>S</i>				
<i>A</i>				
<i>E</i>				
<i>B</i>				
<i>U</i>				
<i>V</i>				
<i>D</i>				

- 6 (1'5 p.) Construir un programa, utilizando Lex y Yacc, que reconozca y simule el comportamiento de un subconjunto de las estructuras condicionales simple y doble en lenguaje Pascal.

Las estructuras a reconocer son aquéllas en las que la condición está formada por dos números enteros y un operador relacional (>, >=, <, <=, =, <>) y las acciones, por una escritura en pantalla de un mensaje constante de hasta 100 caracteres (sólo el procedimiento `write`).

Por ejemplo:

Entrada	Salida esperada
if 8>9 then write (output, 'Hola') else write ('Adios');	Adios
if (5<=8) then if 1=1 then write ('Hola');	Hola