



Apellidos, Nombre.....

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|

1. Todos los alumnos deberán entregar esta hoja, grapada con las soluciones de los ejercicios 1 a 6.
2. Se entregarán las respuestas EN EL ORDEN PROPUESTO. La respuesta al ejercicio 7 se entregará SEPARADA DEL RESTO.

1 (1'5 p.) Obtener un reconocedor finito determinista en forma mínima, una expresión regular y una gramática de tipo 3 para el conjunto de cadenas de ceros y unos que comienzan por 00, pero en los que la subcadena 00 no aparece en ningún otro lugar (en particular, 00 está en el lenguaje, pero 000 no).

2 (1'5 p.) Sobre el alfabeto $\{a, b, c\}$ se considera el lenguaje $L = \{a^i b^j c^k / i \neq j \vee j \neq k\}$.

1. Demostrar que L es independiente del contexto.
2. Demostrar que su complementario no lo es.

3 (0'5 p.) Obtener una gramática en forma normal de Greibach equivalente a

$$\begin{aligned}
 A_1 &\rightarrow A_2 A_2 \\
 A_2 &\rightarrow A_3 A_3 \\
 &\dots \\
 A_{p-1} &\rightarrow A_p A_p \\
 A_p &\rightarrow A_1 A_1 \mid a
 \end{aligned}$$

4 (2'5 p.) Considérese la gramática siguiente:

$$\begin{aligned}
 S &\rightarrow aDR \mid dAR \\
 R &\rightarrow S \mid \epsilon \\
 D &\rightarrow d \mid aDD \\
 A &\rightarrow a \mid dAA
 \end{aligned}$$

que genera $L(G) = \{x \in (a|d)^+ / |x|_a = |x|_d\}$

1. Obtener la TASP para G . Simular el comportamiento de un analizador sintáctico predictivo basado en G para la cadena $adda$
2. Simular el comportamiento de un analizador sintáctico por desplazamiento-reducción basado en G para la cadena $adda$
3. Obtener una gramática no ambigua para el lenguaje $\{x \in (a|b|d)^* / |x|_a + |x|_b = |x|_d\}$

5 (1 p.) 1. Sea L un lenguaje recursivo. Probar que existe una máquina de Turing generadora de L que produce las cadenas de L en orden creciente de longitud.

2. Sea L un lenguaje recursivamente numerable para el que existe una máquina de Turing que genera las cadenas de L en orden creciente de longitud. Probar que, entonces, L es en realidad recursivo.

6 (1 p.) 1. ¿Qué significa que un problema es NP-completo?

2. ¿Qué significa que un problema es P?
3. ¿Es $P=NP$?

tellano, que se describen a continuación, así como el analizador léxico necesario, usando LEX.

Las oraciones constarán de un *sujeto*, un *verbo*, y un *objeto directo* opcional, en ese orden. La entrada consiste en una sucesión de letras, blancos, tabuladores y un retorno de carro para finalizar la oración.

Además, el programa resultante deberá mostrar un mensaje de oración correcta si existe concordancia entre sujeto y verbo (de persona y número) y el posible sintagma nominal concuerda (en género y número) , y un mensaje de oración incorrecta en caso contrario.

El sujeto sólo podrá estar formado por un pronombre personal (*yo, tú, él, ella, nosotros, -as, vosotros, -as, ellos, -as*).

Sólo se considerarán verbos las palabras que terminen en los sufijos correspondientes al futuro simple de indicativo de la primera conjugación (*aré, arás . . .*).

Un objeto directo será un sintagma nominal formado por un determinante (*un -a -os -as, el, los, la -s*) y un nombre.

Se considerará nombre cualquier secuencia de letras que, no siendo determinante ni verbo, termine en *a, as, o, u os*.