Universidad de Valladolid
Departamento de Informática

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

Examen extraordinario. 6 de septiembre de 2000.

Apellidos, Nombre..

1. Todos los alumnos deberán entregar esta hoja, grapada con las soluciones de los ejercicios 1 a 4.
2. Se entregarán las respuestas EN EL ORDEN PROPUESTO. La respuesta al ejercicio 5 se entregará SEPARADA DEL RESTO.

---

1 **(1'5 p.)** Se considera el reconocedor finito no determinista dado por $\Sigma = \{a, b\}$, $Q = \{q_1, q_2\}$, $F = \{q_1\}$ y $f(q_1, a) = f(q_2, b) = \{q_1, q_2\}$, $f(q_1, b) = \{q_2\}$, $f(q_2, a) = \emptyset$

Se pide:

1. una expresión regular para el lenguaje reconocido, obtenida directamente del RFN.
2. un RFD en forma mínima equivalente
3. una expresión regular equivalente a la primera, obtenida a partir de este RFD, y una descripción literal de dicho lenguaje.

2 **(4 p.)** Sobre el alfabeto $\{a, b, c\}$ se considera el lenguaje L_P formado por los palíndromos sobre $\{a, b, c\}$ que contienen exactamente una c, es decir,

$L_P = \{x \in (a|b|c)^*/x = x^R \wedge |x|_c = 1\} = \{wcw^R/w \in (a|b)^*\}$

1. Demostrar que L_P no es regular.
2. Probar que $L_N = \{x \in (a|b|c)^*/x \neq x^R \wedge |x|_c = 1\}$ no es regular, pero verifica el lema de bombeo de los lenguajes regulares.
3. (En este apartado se probará que L_N es independiente de contexto). Supóngase conocido que $P \to aPa|bPb|c$ genera L_P. Sea G la gramática

$$\begin{array}{lllllll} S & \to & aSa & \mid bSb & \mid aSb & \mid bSa & \mid aPb & \mid bPa \\ P & \to & & & aPa & \mid bPb & \mid c & \end{array}$$

¿Quién es $L(G)$? Justificarlo (nota: no es L_N). Hallar una gramática independiente de contexto para L_N.

4. Justificar la inambigüedad de G.
5. Después de factorizar a la izquierda G, hallar la TASP para la gramática resultante, especificando los conjuntos *primeros* y *siguientes*.
6. Realizar la simulación de un análisis ascendente por desplazamiento-reducción basado en G para la cadena $aabbcbbba$. Justificar por qué G puede ser LALR(1).

3 **(1'5 p.)** 1. 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.

2. Descríbase cómo podría construirse una máquina de Turing generadora de $\{a^{n^2}/n \geq 0\}$. ¿Es éste un lenguaje recursivo? (Indicación: $(n+1)^2 = n^2 + 2n + 1$).

4 **(1 p.)** 1. Describe brevemente un problema concreto que sea NP-completo.

2. ¿Qué significa que un problema es P?
3. ¿Qué debería encontrarse para demostrar que P=NP? ¿Y para demostrar que P no es NP?

de cadenas de caracteres o sumas de números enteros que aparezcan en una línea finalizada en ; y devuelva, en el primer caso, la suma del número de caracteres de cada cadena y en el segundo, la concatenación de los números que forman la expresión. Por ejemplo:

Entrada	**Salida**
`hueso + perro + casa;`	`14`
`examen;`	`6`
`42 + 143;`	`42143`