



Apellidos, Nombre..... **Grupo:**

Notas para la resolución del examen: Responder a los ejercicios 5 a 8 en la misma página del enunciado. Responder y entregar por separado los ejercicios 9 y 10.

1 (1 p.) Construya una máquina de Mealy que obtenga el cociente entero entre 3 de un entero binario. Por ejemplo, para la entrada 101000101(325) la salida debe ser 001101100(108)

Indicación: son situaciones distintas que la cadena de entrada represente un múltiplo de 3, un múltiplo de 3 + 1, o un múltiplo de 3 + 2. Añadir un 0 a un número binario supone multiplicarlo por 2, mientras que añadir un 1 supone multiplicarlo por 2 y sumarle 1; en ambos casos el resto y el cociente se modifican en consecuencia.

2 (2 p.) 1. Obtenga un reconocedor finito determinista mínimo equivalente al dado por la tabla siguiente.

	0	1
→ p	{p, q}	{p}
q	{r, s}	{t}
r	{p, r}	{t}
(s)		
(t)		

2. Obtenga una gramática regular por la izquierda para el lenguaje
3. Obtenga una gramática regular por la derecha para el lenguaje
4. Justifique si el lenguaje reconocido es o no el de las cadenas binarias cuyo penúltimo carácter es un 0

3 (1'5 p.) Considere el conjunto de cadenas no vacías de paréntesis y corchetes equilibrados, entendiendo que los paréntesis se equilibran con paréntesis y los corchetes con corchetes. (Por ejemplo, ([[[]]])() está equilibrada, pero [[]] no lo está). Recodifique los paréntesis (y) como a y b respectivamente.

1. Diseñe una gramática que genere este lenguaje
2. Diseñe un autómata con pila determinista que lo reconozca
3. Demuestre que no es regular.

4 (1 p.) Probar las siguientes afirmaciones

1. La intersección de dos lenguajes recursivos es recursiva.
2. La intersección de dos lenguajes recursivamente numerables es recursivamente numerable.
3. Si L_r es recursivo y L_n es recursivamente numerable, entonces $L_n - L_r$ es recursivamente numerable.
4. Si L_r es recursivo y L_n es recursivamente numerable, entonces $L_r - L_n$ no tiene por qué ser recursivamente numerable, es decir, puede serlo o no serlo.

5 (0.5 p.) Dé un ejemplo de tres lenguajes no vacíos L_1 , L_2 y L_3 tales que $L_2 \neq L_3$ pero $L_1L_2 = L_1L_3$

$L_2 =$

$L_1 =$

$L_1L_2 =$

$L_3 =$

6 (1 p.) Construya la Tabla de Análisis Sintáctico Predictivo para la gramática siguiente, una vez eliminados los símbolos y reglas inútiles, y especificando los primeros y siguientes de cada auxiliar.

$S \rightarrow DA \mid PA \mid PQR$ $B \rightarrow BA \mid A$ $P \rightarrow R \mid cP$ $R \rightarrow bR \mid \epsilon$
 $A \rightarrow BP$ $D \rightarrow aD \mid \epsilon$ $Q \rightarrow aPS \mid c$

Gramática sin reglas inútiles:

Primeros									Siguietes								
----------	--	--	--	--	--	--	--	--	-----------	--	--	--	--	--	--	--	--

TASP	a	
S		

7 (0.5 p.) Determine los pivotes de las cadenas siguientes respecto a la gramática $S \rightarrow aSbS \mid bSaS \mid \epsilon$ (Nota: como la gramática es ambigua, es posible que alguna cadena tenga más de un pivote).

- $aabb$: pivotes:
- $abab$: pivotes:

8 (0.5 p.) La gramática

$$R \rightarrow R + R \mid R \bullet R \mid R * \mid (R) \mid \mathbf{0} \mid \mathbf{1} \mid \mathbf{a} \mid \mathbf{b} \mid \mathbf{c}$$

genera las expresiones regulares sobre el alfabeto $\{ \mathbf{a}, \mathbf{b}, \mathbf{c} \}$ pero de forma ambigua (el terminal $\mathbf{1}$ representa a la expresión regular ϵ y el terminal $\mathbf{0}$ a la expresión regular \emptyset). La concatenación se expresa siempre con \bullet , el asterisco no se pone como superíndice y la unión se representa con el símbolo $+$, de forma que lo que típicamente se escribe $(a|b)^*abb|\epsilon$ debe escribirse ahora como $(a + b) * \bullet a \bullet b \bullet b + 1$).

Obtégase una gramática no ambigua para este lenguaje que refleje las precedencias y asociatividades habituales.



Universidad de Valladolid

Departamento de Informática

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

Examen de primera convocatoria. 18 de junio de 2009

Apellidos, Nombre.....

Notas para la resolución del examen: Entregar estos ejercicios separados de los anteriores.

--	--

- 9 (1'25 p.) Construya un programa, utilizando las herramientas Lex y Yacc, que obtenga, para una expresión regular dada sobre el alfabeto { a, b, c } el número de estados y el número de arcos ϵ que generaría el método de Thompson al construir el autómata finito (no determinista) a partir de dicha expresión. (Ver ejercicio 8).

Por ejemplo, para la expresión regular $(a + b) * \bullet a \bullet b \bullet b$ debería obtenerse que tiene 11 estados y 8 arcos ϵ .

- 10 (0.75 p.) Se desea 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 para resolver el problema se construiría utilizando la herramienta Lex. El programa debería 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]

Se pide **solamente** determinar una gramática y tipos semánticos necesarios para poder resolver este problema y especificar en cuál de los códigos fuente (Lex o Yacc) deberían estar definidos.