



# I.T. Informática Gestión – Teoría de Autómatas y Lenguajes

## Examen segunda convocatoria. Parte Practica. 4 de septiembre de 2008.

Apellidos \_\_\_\_\_

Nombre \_\_\_\_\_ Grupo \_\_\_\_\_

--	--	--	--	--

DNI y Firma

<p>Resolver los problemas en una página diferente cada uno de ellos. Poner nombre y apellidos en todas las páginas. Las hojas del enunciado también deben entregarse.</p>	<p>Se valorará la presentación y la claridad en la exposición. No se calificarán las respuestas escritas a lápiz. La nota de cada ejercicio aparece en su enunciado. Resolver el ejercicio 3 en esta misma hoja.</p>
---	--

1. (1,25 ptos) Dado el siguiente fichero *y.output* obtenido por Yacc:

<pre> 0 \$accept : S \$end 1 S : AB 2 A : A b B 3   a 4 B : B a 5   b state 0     \$accept: . S \$end (0)     a shift 1     . error     S goto 2     A goto 3 state 1     A : a . (3)     . reduce 3 state 2     \$accept: S . \$end (0)     \$end accept         </pre>	<pre> state 3     S : A . B (1)     A : A . b B (2)     b shift 4     . error     B goto 5 state 4     A : A b . B (2)     B : b . (5)     b shift 6     \$end reduce 5     a reduce 5     B goto 7 state 5     S : A B . (1)     B : B . a (4)     a shift 8     \$end reduce 1         </pre>	<pre> state 6     B : b . (5)     . reduce 5 state 7     A : A b B . (2)     B : B . a (4)     a : shift 8     b reduce 2 state 8     B : B a . (4)     . reduce 4  4 terminals, 4 nonterminals 6 grammar rules, 9 states         </pre>
--	---	--

Se pide:

- 1) Obtener la tabla de análisis por desplazamiento reducción.
- 2) Simular el análisis para la cadena **abbba**, especificando los estados que se van almacenando en la pila y mostrar la derivación y el árbol de análisis obtenidos.

2. (0,75 pts) Calcúlese la TASP para la siguiente gramática, especificando primeros y siguientes, en los espacios reservados para ello (completando las columnas que falten).

$$G \left\{ \begin{array}{l} A \rightarrow B b \\ \quad \quad | C d \\ B \rightarrow a B \\ \quad \quad | \varepsilon \\ C \rightarrow c C \\ \quad \quad | \varepsilon \end{array} \right.$$

Primeros	A	B	C

Siguientes	A	B	C

TASP	a	b	c		
A					
B					
C					

3. (1,5 puntos) Dado el reconocedor finito determinista escrito en forma de tabla, con un cierto formato, se pide construir un programa (utilizando LEX y YACC) que determine si la entrada es correcta e informe del número de símbolos del alfabeto y del número de estados del autómata.

La tabla del RFD contendrá una línea de cabecera (con los símbolos del alfabeto de entrada separados por el carácter punto\_y\_coma ";") y una línea por cada estado.

Cada línea de estado contendrá en primer lugar el nombre o número de estado, entre paréntesis si es final y precedido por los caracteres "->" si es el inicial, y a continuación los estados siguientes para cada entrada separados por el carácter punto\_y\_coma ";". Para nombrar estados se usarán cadenas de letras minúsculas y dígitos.

Por ejemplo, dada la entrada:

```

a ; c
-> p ; q ; p
(r) ; q ; r
q ; p ; r

```

La salida debería ser

```

AFD correcto
Tiene 3 estados y 2 símbolos de entrada.

```