

## ANÁLISIS LÉXICO Y SINTÁCTICO

### Examen final

<b>Curso/grupo:</b>	2º INF	<b>Curso académico:</b>	2010/2011
<b>Fecha examen:</b>	31/05/2010	<b>Duración examen:</b>	2h 30 minutos
<b>Convocatoria:</b>	Junio	<b>Material permitido:</b>	-
<b>Instrucciones:</b>	Las preguntas no requieren respuestas especialmente extensas, se valorará la claridad en las explicaciones y las respuestas concretas El examen se debe realizar en las hojas correspondientes.		

Nombre: \_\_\_\_\_ DNI: \_\_\_\_\_

**Pregunta 1** **(1 punto)**

Define los siguientes conceptos:

- a) Compilador (0.25 pts.)
- b) Token (0.25 pts.)
- c) Lexema (0.25 pts.)
- d) Intérprete (0.25 pts.)

**Pregunta 2** **(1.5 puntos)**

Dibuja una figura en la que se indiquen las distintas fases de en las que se divide el proceso de compilación y otros elementos importantes que influyan en el mismo. Describe qué funciones se realizan en cada una de las fases, sus inputs y outputs y en qué consisten los elementos nombrados.

**Pregunta 3** **(1.5 puntos)**

Una gramática recursiva a izquierdas (LEFT RECURSIVE) o que tenga dos producciones con un prefijo común (NOT LEFT FACTORED) no es LL(1). ¿Por qué? Indicad cómo se resuelven ambos problemas. Comentad cada punto por separado (0.75 pts. cada apartado).

**Pregunta 4** **(1.5 puntos)**

La siguiente gramática LL(1) reconoce una lista de números separados por coma:

$$\begin{aligned} L &\rightarrow \text{num } S ; \\ S &\rightarrow , \text{ num } S \mid \epsilon \end{aligned}$$

Obtened su tabla de parseo LL(1) (0.75 pts.). Indicad si se puede reconocer o no la siguiente cadena "num , num ; \$" (0.5 pts), si se puede, dibujad el árbol de parseo LL(1) (0.25 pts.).

**Pregunta 5** **(2 puntos)**

Dada la siguiente gramática LR(1):

- 1)  $S \rightarrow L ;$
- 2)  $L \rightarrow + L i$
- 3)  $L \rightarrow i$

Calcular la colección canónica LR(1), y el diagrama de transiciones asociado (DFA). (0.75 pts.)

A partir de estos elementos, obtener la tabla para el parser LR(1). (0.5 ptos.)

Con la tabla calculada, detalla el proceso de reconocimiento de la cadena "+ +iii;". (0.5 ptos.)

Detalla el proceso de reconocimiento de la cadena "+ +i". (0.25 ptos.)

### **Pregunta 7**

**(2.5 puntos)**

El planeta JSU, es un planeta en dos dimensiones, dividido mediante una cuadrícula 8x8 como si de un ajedrez fuese. Sus habitantes tienen un lenguaje muy particular, que pasamos a describir a continuación.

Se compone de 4 campos:

Tipo_de_mensaje	Origen	Destinatario	Mensaje
-----------------	--------	--------------	---------

Así, pasando a describir sus campos:

- Tipo\_de\_mensaje: Es obligatorio. Puede ser:

· 00 si es un mensaje de socorro, sólo se debe especificar origen.

· 01 es un mensaje normal, es obligatorio que se especifiquen todos los demás campos.

· 10, es un mensaje para el rey del planeta (no se especifica destinatario).

· 11, el mensaje se envía a todos los habitantes del planeta (no se especifica destinatario).

- Destinatario, Origen: Se componen de un dígito, una equis y otro dígito, marcando así a qué zona del planeta va dirigido el mensaje. Por ejemplo, son válidas estas direcciones: 1x1, 6x8, 8x3; pero no son válidas: x1, 2x, 3x9, ax2.

- Mensaje: El alfabeto comprende las consonantes del alfabeto en mayúsculas, las vocales en minúsculas y el espacio en blanco. Así, un mensaje se puede componer de palabras separadas por uno o varios espacios en blanco, pero siempre estará delimitado por comillas simples. Las palabras no pueden tener dos vocales consecutivas, o dos consonantes consecutivas.

Algunas posibles frases de distintas conversaciones serían:

00 1x1

01 6x6 3x2 'Hi'

01 3x2 6x6 'Hi aLiCe'

10 1x2 'Mi ReY'

11 4x4 'NoS VaMoS'

Cread un archivo .jj que permita reconocer (léxica y sintácticamente) cualquier frase del lenguaje descrito. Se valorará positivamente las soluciones más óptimas, y que el scanner y parser generados, sean restrictivos con las cadenas de entrada que no cumplan las reglas (es decir, que no haya falsos positivos).

En la siguiente página se incluye código en .jj que puede resultar útil. No hace falta copiar el código superior (options, PARSER\_BEGIN ... PARSER\_END, etc.) a no ser que consideréis necesario modificar algo.

```
/**
 * JavaCC template file created by SF JavaCC plugin_1.5.17+ wizard for
 JavaCC 1.5.0+
 */
options
{
    JDK_VERSION = "1.5";
    static = true;
}

PARSER_BEGIN(JSU_parser)
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;

public class JSU_parser {
    public static void main (String args [])
        throws ParseException {
        JSU_parser anLexSint = new JSU_parser (System.in);
        anLexSint.Input ();
        System.out.println("Analisis terminado:");
        System.out.println ("no se han hallado errores léxico-
sintácticos");
    }
}
PARSER_END(JSU_parser)

TOKEN_MGR_DECLS : {
}

SKIP :
{
}

TOKEN :
{
}

void Input () :
{}
{
}
```