

1. - (2 ptos.) Escribir una gramática LL(1) que sea capaz de reconocer una lista de distribución de e-mails. Los datos que nos llegan tienen la siguiente estructura:

```
To alum1@uv.es ; < Pepe > alguien@yahoo.com
Subject Texto para todos
Text
Aquí puedo escribir lo que quiera....
```

Las palabras reservadas son: To, Subject, Text y como símbolos especiales tenemos <, > y ; Como condiciones adicionales se impone que: el campo To no puede estar vacío pero sí los campos Subject y Text. Si hay más de un destinatario de correo, irá separado del anterior por un ; y si existe un alias para el destinatario, este irá entre < > seguido de la dirección de correo electrónico (aunque no tiene por qué existir).

2. -(4 ptos.) Sea la siguiente gramática un subconjunto muy reducido del lenguaje de programación PASCAL:

Programa → TKN_PROGRAM TKN_ID ; Variables TKN_BEGIN Operacion ListaOperaciones TKN_END

Variables → TKN_VAR TKN_ID ListaVars : tipo ; | ε

ListaVars → , TKN_ID ListaVars | ε

Tipo → TKN_INTEGER | TKN_REAL

ListaOperaciones → Operacion ListaOperaciones | ε

Operacion → TKN_ID := ExpresionArit ;

ExpresionArit → ExpresionArit + ExpresionArit | ExpresionArit * ExpresionArit |
TKN_ID | TKN_NUM | - TKN_NUM

Que se corresponde con el siguiente ejemplo:

```
program nombre_del_programa;
var a,b,c:integer;
begin
b:=-5; c:=9;
a:=2*c*3+b;
end
```

Se pide:

- Escribir el analizador sintáctico usando BISON. (El operador + tiene menor precedencia que el * y ambos son asociativos a la izquierda).
- Diseñar una estructura para la tabla de símbolos, decidiendo la estructura a usar y definiendo (no implementando) las funciones necesarias para:
 - buscar en la tabla un determinado nombre,
 - crear una nueva entrada para ese nombre si no hay ninguna y,
 - devolver en cualquier caso un apuntador al registro correspondiente a dicho nombre.
- Llama a la función "insertar nuevo elemento en la tabla de símbolos" desde las producciones correspondientes del BISON y haz las siguientes comprobaciones semánticas:
 - En la declaración de variables no pueden existir dos identificadores iguales.
 - Dentro del bloque "begin...end" no pueden haber identificadores que no hayan sido definidos previamente en la declaración de variables.

3. -(4 ptos.) Supongamos que tenemos la gramática de PASCAL del ejercicio anterior y queremos ahora implementarla con PCCTS. Se pide:

- Compactar la gramática y escribir las producciones en PCCTS.
- Indicar qué es necesario hacer o incluir en PCCTS para generar el árbol de forma automática y genéralo para las producciones del apartado a).
- Mostrar cuál es la salida que obtenemos al introducir el ejemplo anterior de programa.
- Crear el código intermedio para la producción asignación (de forma simplificada).