

EJERCICIOS sobre GRAMATICAS de ATRIBUTOS

1. Construir una gramática de atributos que genere el conjunto de todas las constantes enteras correspondientes a los valores menores que 2^{31} con la siguiente forma:

<sec-de-digitos-octales> (8)

Ejemplo:

La constante 32 (8) es correcta y tiene el valor 26.

Se exige además que las constantes no tengan ceros a la izquierda.

2. Mediante una gramática de atributos, describir el lenguaje de los números reales con las siguientes restricciones:

- No debe tener ceros a la izquierda la parte entera
- No escribir el cero de la parte entera cuando el valor de la parte decimal sea distinto de cero
- No escribir la parte decimal si su valor es cero.

Ejemplos:

Son incorrectos: 095.76 , 0.45 , 23.0

Son correctos: 95.76 , .45 , 23 , 105.7 , 0 , 4.50

3.

a) Describir en BNF la declaración de los arrays de enteros. Por ejemplo:

array tabla [1..5, 3..8]

representa la declaración de un array de nombre tabla, de 2 dimensiones de rangos 1 a 5 y 3 a 8 respectivamente.

Añadir los atributos necesarios para comprobar las siguientes condiciones de contexto:

- Se considerará un error un rango del tipo x..y con $x > y$
- El número total de elementos del array no puede exceder de 100.

b) Suponiendo que estas declaraciones se han añadido al lenguaje Pam, hay que comprobar que toda variable array que aparezca en alguna instrucción de programa tiene que haber sido declarada con la dimensión correspondiente. Añadir los atributos necesarios para dicha comprobación.

4. Un ejemplo de declaración de una variable de tipo “registro” puede ser el siguiente:

```
var libro : record
    autor, titulo, codigo : string;
    año_publicacion : integer
end
```

a) Definir una sintaxis no ambigua en BNF para dichas declaraciones.

b) Extender dicha definición mediante una Gramática de Atributos, de forma que el atributo **Decs** guarde la información necesaria, asegurando además la condición de que todos los nombres de los campos del registro sean distintos.

5. Suponer un lenguaje de programación con instrucciones condicionales del tipo

```
if (<exp.logica>) <etiqueta>1, <etiqueta>2
```

que indican un salto a la instrucción etiquetada con **<etiqueta>₁** (respect. **<etiqueta>₂**) si la **<exp.logica>** es cierta (respect. falsa).

Partiendo de la sintaxis BNF siguiente, describe mediante una gramática de atributos las condiciones que creas necesarias para verificar la corrección de los saltos a instrucciones.

```
<programa> ::= <cuerpo>
<cuerpo> ::= <sentencia> | <cuerpo> <sentencia>
<sentencia> ::= <sent. ejecutable> | <etiqueta> <sent. ejecutable>
<sent. ejecutable> ::= <sent. if> | <sent. asign> | .....
<sent. if> ::= if (<exp.logica>) <etiqueta>, <etiqueta>
.....
```

6. Dado el siguiente fragmento de gramática BNF, extenderlo a una G.Atributos para verificar la concordancia de tipos en las expresiones.

```
<programa> ::= <declaraciones> <cuerpo>
<declaraciones> ::= <declaracion> | <declaracion> <declaraciones>
<declaracion> ::= <lista-var> : <tipo>;
<lista-var> ::= <var> | <lista-var> , <var>
<tipo> ::= entero | string
<cuerpo> ::= <sentencia> | <cuerpo> <sentencia>
<sentencia> ::= .... | <sent. asign> | .....
<sent. asign> ::= <var> := <expresion> ;
<expresion> ::= <expresion> <op> <termino> | <termino>
<termino> ::= <var> | <constante> | (<expresion>)
<op> ::= + | - | * | div | concat
```