



# Ingeniería Informática

## Procesadores de lenguaje

Examen de teoría (28 de mayo de 2005)

PREGUNTA 1

(5 PUNTOS)

A continuación, se presentan dos posibles extensiones para los lenguajes *Sé* y *Sé*-. Explica claramente qué modificaciones se tendrían que hacer en un compilador de estos lenguajes a Stan o Rossi para que las aceptara. Las extensiones son independientes entre sí; no hace falta que consideres sus posibles interacciones.

En tu descripción, procura ser claro, escueto y preciso. En particular, no es necesario que describas partes del compilador que no estén afectadas por las modificaciones. Puedes optar por descripciones algorítmicas o en lenguaje natural para lograr una mayor sencillez en la explicación. También puede facilitarte la exposición una estructura que siga las distintas etapas del compilador.

**Explicita cualquier asunción que hagas acerca del compilador o del enunciado propuesto.**

### Operadores mínimo y máximo (2 puntos)

Esta modificación introduce dos operadores binarios,  $<?$  y  $>?$ , que devuelven, respectivamente, el mínimo y el máximo de sus operandos. Ambos tienen menos prioridad que  $==$  y  $!=$  y más que  $\&\&$ . Al utilizarlos, se garantiza que se evalúa cada operando exactamente una vez.

Un ejemplo de uso sería el fragmento:

```
printf("El menor es %d y el mayor %d.", a<?b, a>?b);
```

El tipo del resultado es el más general de los tipos de sus operandos y estos deben tener tipo elemental<sup>1</sup>.

Muestra qué código generarías para la expresión:

```
f = i<?c;
```

donde  $f$  es una variable de tipo real,  $i$  de tipo entero y  $c$  de tipo carácter. Todas son variables globales y puedes asignarles las direcciones que quieras.

### Máximos de secuencias (3 puntos)

Mediante esta extensión, el programador puede calcular fácilmente el máximo de una serie de elementos. Para ello, se introduce la nueva palabra reservada `max`. Al utilizarla, se sigue la sintaxis:

```
max( $e1$ ;  $e2$ ;  $e3$ )
```

La ejecución de esta expresión se realiza de manera similar al bucle `for`. En primer lugar se evalúa  $e1$  (que hace el papel de inicializador), después, mientras  $e2$  sea cierta, se evalúa  $e3$  y se devuelve como resultado de la expresión el máximo de estas evaluaciones de  $e3$ . Por ejemplo, el fragmento<sup>2</sup>

```
printf("El doble del máximo es: %f", 2*max(i=0; i<n; v[i++]));
```

escribiría del doble del máximo de los  $n$  primeros elementos del vector  $v$ .

Las tres expresiones deben ser de tipos elementales y el tipo del resultado es el de  $e3$ .

Muestra qué código se generaría para el ejemplo anterior, asumiendo que  $i$  y  $n$  son variables globales de tipo entero y  $v$  un vector global de tipo real. Puedes asignarles las direcciones que quieras.

PREGUNTA 2

(2 PUNTOS)

Suponiendo que el alfabeto sobre el que se definen las expresiones regulares es el de los dígitos, escribe expresiones regulares para los siguientes lenguajes:

- Cadenas de dígitos que no empiecen por 22.
- Cadenas de dígitos que no empiecen ni terminen por 22.

<sup>1</sup>Durante el examen se aclaró que son asociativos por la izquierda.

<sup>2</sup>En la versión original, faltaba el incremento de  $i$ .

- Cadenas de dígitos que no contengan la subcadena 22.
- Cadenas de dígitos que contengan exactamente una vez la subcadena 22.

## PREGUNTA 3

(1,5 PUNTOS)

Podemos saber si un número es múltiplo de once sumando los dígitos que ocupan posiciones pares y los que ocupan posiciones impares. Si la diferencia entre ambas sumas es múltiplo de once, también lo era el número original.

Sea  $G$  la siguiente gramática:

$$\begin{aligned} \langle S \rangle &\rightarrow \langle N \rangle \\ \langle N \rangle &\rightarrow \langle \text{Dig} \rangle \langle \text{Dig} \rangle \langle A \rangle \langle \text{Dig} \rangle \langle A \rangle \mid \langle N \rangle \langle \text{Dig} \rangle \langle A \rangle \\ \langle A \rangle &\rightarrow \langle A \rangle \langle A \rangle \mid \langle \text{Dig} \rangle \\ \langle \text{Dig} \rangle &\rightarrow \text{dígito} \end{aligned}$$

Supongamos que la primera regla tiene asociada la acción

$$\langle S \rangle \rightarrow \langle N \rangle \{ \langle S \rangle . m11 := (\langle N \rangle . sp - \langle N \rangle . si) \% 11 = 0 \}$$

donde los atributos  $si$  y  $sp$  almacenan las sumas de los dígitos en posición par e impar. Añade, al final de cada una de las restantes reglas, las acciones semánticas necesarias para que se calcule el valor de estos atributos.

Puedes utilizar los atributos adicionales que consideres necesarios, pero ninguna variable global. Además, los atributos que añadas deben ser de tipo entero o lógico.

## PREGUNTA 4

(1,5 PUNTOS)

Definimos la *inversa* de una gramática  $G = (N, \Sigma, P, \langle S \rangle)$  como la gramática  $G^{-1} = (N, \Sigma, P^{-1}, \langle S \rangle)$  donde  $P^{-1}$  es

$$P^{-1} = \{ \langle A \rangle \rightarrow \alpha^{-1} \mid \langle A \rangle \rightarrow \alpha \in P \} \quad (1)$$

y  $\alpha^{-1}$  representa la cadena  $\alpha$  escrita en orden inverso.

Demuestra la verdad o falsedad de las siguientes afirmaciones:

- Si  $G$  es LL(1), entonces  $G^{-1}$  es LL(1).
- Si  $G$  es LL(1), entonces  $G^{-1}$  no es LL(1).
- Si  $G$  es SLR, entonces  $G^{-1}$  es SLR.
- Si  $G$  es SLR, entonces  $G^{-1}$  no es SLR.

Duración del examen: 4 horas

¡Buena suerte!