



5º Ingeniería Informática

E79 Procesadores de lenguaje

Examen de teoría (22 de mayo de 2004)

PREGUNTA 1

(5 PUNTOS)

A continuación, se presentan dos posibles extensiones del lenguaje *r-LaST*. Explica claramente qué modificaciones se tendrían que hacer en un compilador de *r-LaST* a Stan para que las aceptara. Las modificaciones son independientes entre sí; no hace falta que consideres sus posibles interacciones.

En tu descripción de las modificaciones, 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.

Escritura en minúsculas (2 puntos)

Esta modificación introduce la nueva palabra clave LCPRINT (por *lower case print*) que tiene un funcionamiento análogo al PRINT de *r-LaST*, excepto en el caso de las cadenas, que son escritas con todas las letras en minúsculas. La transformación en minúsculas afecta únicamente a la salida, no a la cadena original.

Así, el fragmento siguiente:

```
s:= LCPRINT("Hola Mundo\n");  
PRINT(s)
```

Escribiría por pantalla:

```
hola mundo  
Hola Mundo
```

Siendo *s* una variable de tipo cadena.

Nota: sólo hace falta transformar las letras mayúsculas del código ASCII (rango 65–90).

Cálculo de sumatorios (3 puntos)

Esta modificación introduce una nueva estructura de ejecución repetitiva que permite calcular sumatorios fácilmente. Para ello se utiliza la nueva palabra reservada SUM y la estructura

```
WHILE expresión1 SUM expresión2 END
```

La evaluación de esta estructura conlleva la evaluación de la *expresión*₂ mientras *expresión*₁ sea cierta. Como resultado se devuelve la suma de los valores obtenidos al evaluar la segunda expresión. Si *expresión*₁ es falsa la primera vez, el resultado es cero. Por ejemplo, el código siguiente:

```
i:= 0;  
PRINT (WHILE i:=i+1; i<= 5 SUM i END, ", ");  
PRINT (WHILE i<= 5 SUM i END)
```

Escribiría 15, 0.

Lógicamente, el tipo de la expresión es el mismo que el de *expresión*₂, que debe ser entero o real. Además, el tipo de *expresión*₁ debe ser lógico.

PREGUNTA 2

(3,5 PUNTOS)

Sea G la siguiente gramática para modelar polinomios:

$$\begin{aligned}\langle P \rangle &\rightarrow \langle P \rangle + \langle M \rangle \mid \langle M \rangle \\ \langle M \rangle &\rightarrow \text{ent } \langle F \rangle \mid \langle F \rangle \text{ent} \\ \langle F \rangle &\rightarrow \langle F \rangle \langle l \rangle \mid \langle l \rangle \\ \langle l \rangle &\rightarrow \text{id} \mid \text{id} \text{ent}\end{aligned}$$

donde **ent** representa literales enteros e **id**, identificadores. Se pide:

- Por 1,5 puntos:
 - Construye el autómata de prefijos viables para G utilizando ítems LR(0).
 - A partir del autómata, escribe la tabla de acciones LR(0).
 - Si la tabla anterior tiene conflictos, escribe la tabla de acciones SLR.
 - Si la tabla anterior tiene conflictos, indica cómo se podrían resolver modificando la tabla o la gramática.
- Por 1,5 puntos:
 - Añade a G^1 las acciones semánticas necesarias para que en el atributo *derivada* de $\langle P \rangle$ se obtenga la derivada del polinomio respecto a la variable x . Puedes elegir la representación que consideres apropiada (árbol, lista, cadena, ...). Utiliza los atributos que consideres necesarios, pero ninguna variable global. Ten en cuenta que la variable puede aparecer más de una vez en el monomio (ej, $2x^3yx$).
- Por 0,5 puntos:
 - Reescribe la gramática utilizando partes derechas regulares de modo que se obtenga una sola regla.
 - ¿Es regular el lenguaje generado por la gramática? Justifica la respuesta.

PREGUNTA 3

(2 PUNTOS)

Dadas las siguientes gramáticas para modelar el lenguaje formado por las cadenas de aes y bes alternadas que comienzan y terminan por a:

$$\begin{array}{lll} \langle S \rangle &\rightarrow \langle A \rangle (\langle B \rangle \langle S \rangle)^* & \langle S \rangle &\rightarrow ((\langle A \rangle \langle B \rangle)^* \langle S \rangle) \lambda & \langle S \rangle &\rightarrow ((\langle A \rangle \langle B \rangle \langle S \rangle)^* \\ \langle A \rangle &\rightarrow \mathbf{a} & \langle A \rangle &\rightarrow \mathbf{a} & \langle A \rangle &\rightarrow \mathbf{a} \\ \langle B \rangle &\rightarrow \mathbf{b} & \langle B \rangle &\rightarrow \mathbf{b} & \langle B \rangle &\rightarrow \mathbf{b} \end{array}$$

se pide, por cada gramática:

- Justificar si el lenguaje generado se corresponde o no con el esperado.
- Escribir la correspondiente tabla de análisis RLL(1).
- En caso de conflictos, comentar si se pueden resolver o no manipulando la tabla.

Duración del examen: 4 horas
¡Buena suerte!

¹Aunque hayas tenido que modificar G para resolver los conflictos, las acciones debes escribirlas sobre G , no sobre la versión modificada.