



5º Ingeniería Informática

E79 Procesadores de lenguaje

Examen de teoría (21 de junio 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.

Asignación entre grupos (2 puntos)

Con esta modificación se extiende la asignación de modo que se permite que parte derecha e izquierda tengan tipo grupal. El resto de características de la asignación no cambia; en particular, deberá coincidir el tipo entre ambas partes y, como efecto secundario, la parte izquierda recibe una copia de la parte derecha.

Por ejemplo, si *g1* y *g2* son variables del mismo tipo grupal con un campo entero *x*, el fragmento

```
g1.x:= 1; g2.x:= 2;
g1:= g2;
print (g1.x, " ", g2.x); nl;
(g1:= g2).x:= 3;
print (g1.x, " ", g2.x); nl;
```

escribe por pantalla 2 2 y 2 3.

Notas: Dado que estamos empleando el lenguaje *r-LaST*, no hace falta considerar la posibilidad de que los grupos correspondan a objetos locales a funciones. No se permite que la talla del código generado crezca exponencialmente con la del código fuente.

Búsqueda binaria en grupos (3 puntos)

Con esta extensión, se añade un nuevo operador, el *?*, que permite realizar búsquedas binarias de un elemento en la parte vectorial de los grupos, que se supone ordenada. Este operador es asociativo por la derecha y tiene una prioridad intermedia entre los niveles 3 (más unario, cambio de signo y negación) y 4 (multiplicación, división, etc.). La parte izquierda del operador es una expresión de tipo grupal y la parte derecha una expresión de tipo entero, que también debe ser el tipo base de la parte vectorial del grupo. El resultado es un entero que corresponde bien a la posición del vector donde se encuentra el elemento buscado, bien a la posición del menor de los números mayores que ese elemento. Si todos los elementos de la parte vectorial fueran inferiores al elemento buscado, se devolvería la talla de la parte vectorial más uno.

Por ejemplo, el siguiente fragmento, donde *g* es del tipo `{[3]: integer}`,

```
g[1]:= 1; g[2]:= 3; g[3]:= 5;
print (0, g ? 0); nl;
print (3, g ? 3); nl;
print (7, g ? 7); nl;
```

escribiría por pantalla:

```
0, 1
3, 2
7, 4
```

Notas: Está indefinido el comportamiento ante vectores desordenados. Dado que estamos empleando el lenguaje *r-LaST*, no hace falta considerar la posibilidad de que los grupos correspondan a objetos locales a funciones. No se permite que la talla del código generado crezca exponencialmente con la del código fuente.

PREGUNTA 2

(1,5 PUNTOS)

El método explicado en el tema de análisis léxico para generar autómatas a partir de expresiones regulares utilizando ítems, ¿genera autómatas mínimos? Justifica la respuesta.

PREGUNTA 3

(1,5 PUNTOS)

Sea G la siguiente gramática:

$$\begin{aligned}\langle A \rangle &\rightarrow \langle B \rangle \langle C \rangle | a \\ \langle B \rangle &\rightarrow \langle A \rangle a \langle B \rangle | \langle C \rangle a \\ \langle C \rangle &\rightarrow a \langle C \rangle \langle C \rangle | a b a \langle C \rangle | a\end{aligned}$$

Añade a G las reglas semánticas necesarias para que el atributo ia de $\langle A \rangle$ contenga el número de a 's al inicio de la cadena generada. Por ejemplo, dadas las cadenas $aaaaba$, $abaaaa$ y aaa , los valores de ia serían 4, 1 y 3, respectivamente.

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

(2 PUNTOS)

Sea G la siguiente gramática:

$$\begin{aligned}\langle S \rangle &\rightarrow a \langle B \rangle \langle S \rangle | b \langle A \rangle \langle S \rangle | \lambda \\ \langle A \rangle &\rightarrow a | b \langle A \rangle \langle A \rangle \\ \langle B \rangle &\rightarrow b | a \langle B \rangle \langle B \rangle\end{aligned}$$

Se pide:

- Construir el autómata de prefijos viables y la tabla de acciones SLR.
- Si la tabla no tiene conflictos, mostrar cómo se analizaría y qué árbol de derivación tendría la cadena $aabbba$.
- Si la tabla tiene conflictos, explicar si se deben a que la gramática es ambigua o a que necesitaríamos un método de análisis más potente.

Duración del examen: 4 horas

¡Buena suerte!