



5º Ingeniería Informática

E79 Procesadores de lenguaje

Examen de teoría (18 de diciembre 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.

Alias de palabras clave (2 puntos)

Mediante esta modificación, el programador puede modificar el conjunto de palabras reservadas del lenguaje introduciendo sinónimos para algunas de ellas. Para ello se emplea la secuencia `$alias<clave, sustituto>$`, donde *clave* y *sustituto* son identificadores que siguen las normas de *r-LaST*. Tras la sustitución, todas las apariciones de *sustituto*¹ serán equivalentes a apariciones de *clave*, que puede seguir siendo utilizada normalmente. Por ejemplo, el programa

```
$alias<do,haz>$
$alias<globals,globales>$ $alias<end,fin>$
$alias<while,mientras>$
Globales
  i: $alias<integer,entero>$
  entero
End
i:= 1;
Mientras i<= 10 Haz $alias<print,escribe>$
  Escribe(i<+>); nl;
Fin
```

escribiría los números del uno al diez.

Notas:

- No puede aparecer en la definición del alias ningún espacio, tabulador o fin de línea.
- Las definiciones de alias pueden aparecer en los mismos contextos que los comentarios y se aplican desde su aparición hasta el fin del programa, sin tener en cuenta las reglas de ámbito del resto de identificadores.
- Se pueden definir alias a otros alias definidos previamente, pero no se puede redefinir un alias.
- Si *clave* no es una palabra clave o un alias definido previamente, se produce un error.

Operador factorial (3 puntos)

Esta extensión crea un nuevo operador postfijo: el operador factorial, representado mediante `!`. La prioridad del nuevo operador es idéntica a la de los operadores de acceso a grupo y llamada a función. El operando sobre el que se calcula la factorial debe ser de tipo entero. Por ejemplo, el fragmento

```
a:= 5;
print ("Factorial de ",a," es ",a!); nl;
```

escribiría `Factorial de 5 es 120`.

Nota: no está definido el comportamiento ante valores negativos.

¹ Siguiendo las normas de *r-LaST*, no importa si se emplean mayúsculas o minúsculas.

PREGUNTA 2

(1,5 PUNTOS)

Sea la r_n la expresión regular $\overbrace{(a|ab) \dots (a|ab)}^n$. Se pide:

- Lenguaje representado por r_n .
- Número de estados del AFD construido a partir de r_n utilizando ítems; o bien, número de estados del AFN construido mediante el algoritmo de bloques.

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 fa de $\langle A \rangle$ contenga el número de aes al final de la cadena generada. Por ejemplo, dadas las cadenas **aaaaba**, **abaaaa** y **aaa**, los valores de fa serían 1, 4 y 3, respectivamente.

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

PREGUNTA 4

(2 PUNTOS)

Responde, justificadamente, a las siguientes preguntas:

- Sean G_1 y G_2 dos gramáticas LL(1) que tienen los mismos no terminales. Sean M_1, M_2 sus respectivas tablas de análisis. Decimos que las tablas son similares si para todo no terminal $\langle A \rangle$ y todo terminal a (o marca de fin de entrada, $\$$), la entrada $M_1[\langle A \rangle, a]$ no está vacía si y sólo si no lo está la entrada $M_2[\langle A \rangle, a]$. ¿Pueden ser distintos $L(G_1)$ y $L(G_2)$ si M_1 y M_2 son similares?
- Sea G la gramática siguiente:

$$\begin{aligned} \langle A \rangle &\rightarrow a \langle A \rangle a \\ \langle A \rangle &\rightarrow \lambda \end{aligned}$$

- ¿Cuál es el lenguaje generado por G ?
- ¿Puedes demostrar que G no es ambigua?
- ¿Es G SLR?, ¿y LR(1)?

Duración del examen: 4 horas
¡Buena suerte!