



Ingeniería Informática

Procesadores de lenguaje

Examen de teoría (24 de junio de 2006)

PREGUNTA 1

(5 PUNTOS)

Explica cómo debería modificarse `minicomp` (en su versión para Stan o para Rossi, pero sin ninguna de las extensiones planteadas en la página de la asignatura) de modo que aceptara las extensiones que se presentan a continuación. Las extensiones son independientes entre sí; no hace falta que consideres sus posibles interacciones.

En tu descripción, procura ser claro, escueto y preciso. 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 enunciado propuesto.

Array asociativo (3 puntos)

Con esta extensión se introduce un nuevo tipo de datos: el vector asociativo. La especificación del nuevo tipo tiene la forma:

```
asociativo [ índices ] de tipoBase
```

donde `asociativo` es una nueva palabra reservada, `índices` es una lista no vacía de literales enteros distintos entre sí y separados por comas y `tipoBase` es cualquier tipo del lenguaje (incluido el nuevo tipo asociativo, pero no el tipo lógico).

Una variable de tipo asociativo representa una secuencia de tantos objetos del tipo base como índices haya en la lista. El acceso a los elementos se realiza mediante una expresión sintácticamente idéntica a la de acceso a los elementos de un vector, cambiando su interpretación de modo que al primer elemento se accede mediante el valor del primer índice y así sucesivamente. Por ejemplo, tras la declaración

```
v: asociativo [3,2,4] de cadena;
```

las expresiones `v[3]` y `v[2]` harían referencia al primer y segundo elementos de `v`, respectivamente. Si el valor de la expresión entre corchetes no se corresponde con ninguno de los índices de la declaración, se hace referencia al último elemento del vector asociativo. En nuestro ejemplo, esto significa que tanto `v[4]` como `v[3*8]` hacen referencia al tercer elemento de `v`.

Además de las modificaciones, muestra qué código se generaría para la secuencia

```
v[3] := "ejemplo";  
escribe v[1+2];
```

suponiendo la declaración anterior. Asigna a `v` y el literal de cadena las direcciones que consideres convenientes.

Operador sufijo (2 puntos)

Esta extensión introduce un nuevo operador, el operador sufijo. Es un operador binario que se representa mediante `::` y tiene como operando izquierdo una expresión de tipo cadena y como operando derecho una expresión de tipo entero. El resultado de la evaluación del operador es el sufijo del resultado de la expresión de cadena con longitud igual al resultado de la expresión entera. Por ejemplo, si en `v[2]` tuviéramos la cadena `hola mundo`, la sentencia

```
escribe v[2]::(2+3);
```

escribiría la cadena `mundo`. Si la longitud del sufijo es mayor que la de la cadena, el resultado está indefinido. Este operador es el de mayor prioridad entre los operadores binarios.

Además de las modificaciones, muestra qué código se generaría para el ejemplo. Asigna a `v` la dirección que consideres conveniente.

PREGUNTA 2

(1,5 PUNTOS)

Sea r_n la expresión regular $\mathbf{ab|aabb|...|a...ab...b}$. ¿Cuántos estados tiene el autómata construido para r_n mediante el algoritmo explicado en clase?

Si el autómata resultante es mínimo, demuéstralo; en caso contrario, di cuántos estados tiene el autómata mínimo.

PREGUNTA 3

(1,5 PUNTOS)

Diseña un método para calcular los últimos de un no terminal $\langle A \rangle$, definidos como

$$\text{últimos}(\langle A \rangle) = \{a \in \Sigma \mid \exists \alpha \in \Sigma^* : \langle A \rangle \xRightarrow{*} \alpha a\}.$$

Es decir, $\text{últimos}(\langle A \rangle)$ es el conjunto de terminales que pueden ser el último símbolo de las cadenas generadas por $\langle A \rangle$.

Escribe la traza de la aplicación de tu método en el cálculo de los últimos de $\langle A \rangle$ en la gramática siguiente:

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

PREGUNTA 4

(2 PUNTOS)

¿Puede un estado del autómata de prefijos viables de una gramática SLR contener simultáneamente los ítems $\langle A \rangle \rightarrow \cdot \langle A \rangle \langle A \rangle$, $\langle A \rangle \rightarrow \langle A \rangle \cdot \langle A \rangle$ y $\langle A \rangle \rightarrow \langle A \rangle \langle A \rangle \cdot$? ¿Y los ítems $\langle B \rangle \rightarrow \cdot \langle A \rangle \langle A \rangle$, $\langle B \rangle \rightarrow \langle A \rangle \cdot \langle A \rangle$ y $\langle B \rangle \rightarrow \langle A \rangle \langle A \rangle \cdot$?

Para cada pregunta, si es imposible, demuéstralo; si es posible, escribe una gramática en la que se cumpla.

Duración del examen: 4 horas

¡Buena suerte!