

# E79 Procesadores de lenguaje

## Examen de teoría (2 de junio de 2001)

PREGUNTA 1

(6 PUNTOS)

Te has cansado de usar `metacomp` y has decidido crear tu propio metacompilador, `mimc`. Para no complicarte la vida, has decidido que `mimc` acepte gramáticas LR(1) y genere programas Python que implementen los correspondientes analizadores. Para esto, quieres aprovechar la biblioteca `metacompilador` que te has bajado de la red.

Un programa `mimc` es básicamente una colección de una o más reglas. Para escribir el par de reglas  $\langle A \rangle \rightarrow \langle A \rangle b$  y  $\langle A \rangle \rightarrow c$ , usamos la notación:

$$\{\langle A \rangle: \langle A \rangle b \mid c\}$$

Es decir, las **reglas** tienen la siguiente estructura: comienzan por una llave abierta (`{`); siguen con un no terminal y dos puntos (`:`); después, continúan con una o más partes derechas separadas por barras (`|`); y terminan con una llave cerrada (`}`).

La **parte derecha** es una secuencia de cero o más de los siguientes elementos:

- **Símbolos terminales**: representados por secuencias de letras mayúsculas y minúsculas.
- **Símbolos no terminales**: representados por secuencias de letras mayúsculas y minúsculas encerradas entre un símbolo “menor que” (`<`) y un símbolo “mayor que” (`>`).
- **Acciones semánticas**: secuencias de caracteres encerrados entre dos caracteres arroba (`@`). Las acciones no pueden contener ni el carácter nueva línea ni el carácter arroba.
- **Reglas anidadas**.

El anidamiento de reglas se utiliza para representar de una manera más compacta la gramática. Así, la regla  $\{\langle A \rangle: a \{\langle B \rangle: b \langle B \rangle \mid c\} d\}$  es equivalente al par de reglas  $\{\langle A \rangle: a \langle B \rangle d\}$  y  $\{\langle B \rangle: b \langle B \rangle \mid c\}$ . Las cadenas vacías se representan mediante partes derechas vacías ( $\langle A \rangle \rightarrow \lambda$  se escribe  $\{\langle A \rangle: \}$ ).

Las acciones semánticas contienen código Python que se ejecutará cuando la correspondiente parte de la regla sea reconocida.

Mezclados libremente entre las reglas y dentro de ellas, se pueden incluir **segmentos de código Python** que formarán parte del código finalmente generado y especificaciones de categorías léxicas. Los segmentos de código Python comienzan por una secuencia fin de línea, carácter de porcentaje (`\n%`) y terminan en una secuencia idéntica. Entre ellas, puede haber cualquier tipo de caracteres, incluyendo caracteres nueva línea, excepto una secuencia (`\n%`).

Las **especificaciones de categorías léxicas** son secuencias formadas por: un carácter abrir llave (`{`), un terminal o la palabra `None`, un carácter barra (`|`), un identificador de función Python o la palabra `None`, un segundo carácter barra (`|`), una expresión regular que no puede contener la llave cerrada y un carácter llave cerrada (`}`). Por ejemplo, los identificadores de un lenguaje de programación podrían ser: `{id|trata_id|[a-zA-Z][a-zA-Z_0-9]*}`.

En las especificaciones de categorías léxicas no pueden aparecer espacios salvo que formen parte de la expresión regular. En ningún caso pueden aparecer caracteres nueva línea en una especificación de categoría léxica.

Tanto los espacios como los tabuladores carecen de cualquier significado cuando no aparecen formando parte de acciones semánticas, código Python o especificaciones de categorías léxicas y sólo sirven para separar categorías léxicas. Los caracteres nueva línea sólo tienen relevancia cuando forman parte de segmentos de código Python. Se pueden incluir **comentarios** en cualquier lugar que no corresponda a código Python. Para ello se utiliza la sintaxis de C (el comentario comienza con `/*` y termina con `*/`).

La biblioteca `metacompilador` ofrece las siguientes clases:

- `mcAnalizadorLexico`: para construir analizadores léxicos. Para especificar las categorías, se utiliza el método `especifica` que tiene tres parámetros: una cadena representando una categoría léxica o el valor `None`; el nombre de la función que tratará los componentes léxicos de la categoría o `None`; y la expresión regular de la categoría.
- `mcAnalizadorSintactico`: para construir analizadores sintácticos. El constructor recibe como parámetro un analizador léxico *con todas las categorías ya especificadas*. Ofrece los métodos `regla`, `compila` y `analiza`. El primero sirve para especificar las reglas. Tiene dos parámetros: la parte izquierda, que ha de ser un no terminal; y la parte derecha, una lista con terminales, no terminales y acciones. El método `compila` hace que el objeto procese las reglas introducidas. El método `analiza` recibe un fichero y lo analiza.

- `mcTerm`: para representar terminales. El constructor recibe el nombre de la categoría léxica, *que debe haber sido especificada en el analizador léxico*.
- `mcNTerm`: para representar no terminales. El constructor recibe el nombre del no terminal.
- `mcAccion`: para representar acciones semánticas. Recibe una cadena con el código Python de la acción.

Un ejemplo de programa `mimc` sería el siguiente:

```

/* Eliminamos los espacios */
{None|None| [ \t\n]+}

/* Principal */
{<E>:
  {op|None|[+-]}
  <E> op {<T> : entero @T.r = entero.v@}
/* Insertamos código */
%
def suma(op, izq, der):
  if op == "+": return izq+der
  else: return izq-der
%
  @E.r = suma(op, E2.r, T.r)@}
{<E>: @E.r = T.r@ /* Vacía */}

/* Algunas categorías léxicas */
{entero|valor|[0-9]+}
{pa|None|\(\)}
{pc|None|\)\)}

/* Las expresiones entre paréntesis */
{<T> : pa <E> pc @T.r = E.r@}

/* Un poco más de código */
%
def valor(comp):
  comp.v= string.atoi(comp.lexema)

def main():
  f = open("expresiones")
  A = Analizador.analiza(f)

```

Una posible traducción a Python sería:

```

def suma(op, izq, der):
    if op == "+": return izq+der
    else: return izq-der

def valor(comp):
    comp.v= string.atoi(comp.lexema)

def main():
    f = open("expresiones")
    A = Analizador.analiza(f)

from metacompilador import mcAnalizadorLexico, mcAnalizadorSintactico,
    mcTerm, mcNTerm, mcAccion

alex= mcAnalizadorLexico()
alex.especifica(None,None,'[ \t\n]+')
alex.especifica('op',None,'[+-]')
alex.especifica('entero',valor,'[0-9]+')
alex.especifica('pa',None,'\(\(')
alex.especifica('pc',None,'\)\)')

Analizador= mcAnalizadorSintactico(alex)
Analizador.regla(mcNTerm('E'), [mcNTerm('E'), mcTerm('op'), mcNTerm('T'),
    mcAccion('E.r = suma(op, E2.r, T.r)')])
Analizador.regla(mcNTerm('E'), [mcNTerm('T'), mcAccion('E.r = T.r')])
Analizador.regla(mcNTerm('T'), [mcTerm('entero'), mcAccion('T.r = entero.v')])
Analizador.regla(mcNTerm('T'), [mcTerm('pa'), mcNTerm('E'), mcTerm('pc'),
    mcAccion('T.r = E.r')])
Analizador.compila()
main()

```

Se pide:

- Especificación del analizador léxico, incluyendo, al menos: categorías léxicas, sus expresiones regulares y las acciones correspondientes.
- Especificación del analizador sintáctico mediante una gramática incontextual.

- Transformación de la gramática anterior en una gramática LL(1), RLL(1), LR(0), SLR, LR(1) o LALR. Demostración de que la gramática transformada pertenece a alguna de estas familias.
- Relación (no implementación) de las comprobaciones semánticas que debe realizar el metacompilador.
- Gramática atribuida (que puede estar basada en la original o en la transformada) con las acciones semánticas para:
  - crear la traducción del programa mimc a Python; o
  - crear una representación intermedia del programa mimc. En este caso, debe explicarse cómo se realizaría el paso de la representación intermedia al programa Python.

PREGUNTA 2

(2 PUNTOS)

Estás preparando un nuevo método de análisis sintáctico para gramáticas incontextuales y has llegado a la conclusión de que necesitas saber, para una forma sentencial, cuál es el conjunto de terminales que pueden aparecer en segunda posición en las sentencias que se deriven de ella. Decides formalizar tu función segundos de la siguiente manera:

$$\text{segundos}(\alpha) = \{a \in \Sigma \mid \exists b \in \Sigma, \beta \in \Sigma^* : \alpha \xRightarrow{*} ba\beta\}$$

Escribe un algoritmo análogo al que se emplea para el cálculo de primeros. Puedes asumir que tu gramática no tiene recursividad por la izquierda y que dispones de las funciones:

- primeros( $\alpha$ ): devuelve los primeros de una forma sentencial  $\alpha$ , incluyendo la cadena vacía si  $\alpha \xRightarrow{*} \lambda$ .
- unitario( $\langle A \rangle$ ) devuelve cierto si  $\exists a \in \Sigma : \langle A \rangle \xRightarrow{*} a$ , falso en caso contrario.

PREGUNTA 3

(2 PUNTOS)

Indica, para cada uno de los siguientes pares de estados, si la gramática correspondiente tiene un conflicto SLR, si no podemos saberlo o si los estados son incorrectos. Justifica cada respuesta:

<p>1)</p> <table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle</math></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c</math></td></tr> <tr><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \langle A \rangle \langle B \rangle c \cdot</math></td><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \cdot ab</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot \langle C \rangle</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot d</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle C \rangle \rightarrow \cdot a</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle C \rangle \rightarrow \cdot b</math></td></tr> </table>	$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$	$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$	$\langle B \rangle \rightarrow \langle A \rangle \langle B \rangle c \cdot$	$\langle A \rangle \rightarrow \cdot ab$		$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$		$\langle B \rangle \rightarrow \cdot \langle C \rangle$		$\langle B \rangle \rightarrow \cdot d$		$\langle C \rangle \rightarrow \cdot a$		$\langle C \rangle \rightarrow \cdot b$	<p>2)</p> <table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle</math></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c</math></td></tr> <tr><td style="padding: 2px;"><math>\langle A \rangle \rightarrow ab \cdot</math></td><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \cdot ab</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot \langle C \rangle</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot d</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle C \rangle \rightarrow \cdot a</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle C \rangle \rightarrow \cdot \langle D \rangle</math></td></tr> </table>	$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$	$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$	$\langle A \rangle \rightarrow ab \cdot$	$\langle A \rangle \rightarrow \cdot ab$		$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$		$\langle B \rangle \rightarrow \cdot \langle C \rangle$		$\langle B \rangle \rightarrow \cdot d$		$\langle C \rangle \rightarrow \cdot a$		$\langle C \rangle \rightarrow \cdot \langle D \rangle$
$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$	$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$																												
$\langle B \rangle \rightarrow \langle A \rangle \langle B \rangle c \cdot$	$\langle A \rangle \rightarrow \cdot ab$																												
	$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$																												
	$\langle B \rangle \rightarrow \cdot \langle C \rangle$																												
	$\langle B \rangle \rightarrow \cdot d$																												
	$\langle C \rangle \rightarrow \cdot a$																												
	$\langle C \rangle \rightarrow \cdot b$																												
$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$	$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$																												
$\langle A \rangle \rightarrow ab \cdot$	$\langle A \rangle \rightarrow \cdot ab$																												
	$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$																												
	$\langle B \rangle \rightarrow \cdot \langle C \rangle$																												
	$\langle B \rangle \rightarrow \cdot d$																												
	$\langle C \rangle \rightarrow \cdot a$																												
	$\langle C \rangle \rightarrow \cdot \langle D \rangle$																												
<p>3)</p> <table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle</math></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c</math></td></tr> <tr><td style="padding: 2px;"><math>\langle A \rangle \rightarrow ab \cdot</math></td><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \cdot ab</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot \langle C \rangle</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot d</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle C \rangle \rightarrow \cdot a</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle C \rangle \rightarrow \cdot b</math></td></tr> </table>	$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$	$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$	$\langle A \rangle \rightarrow ab \cdot$	$\langle A \rangle \rightarrow \cdot ab$		$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$		$\langle B \rangle \rightarrow \cdot \langle C \rangle$		$\langle B \rangle \rightarrow \cdot d$		$\langle C \rangle \rightarrow \cdot a$		$\langle C \rangle \rightarrow \cdot b$	<p>4)</p> <table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle</math></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c</math></td></tr> <tr><td style="padding: 2px;"><math>\langle A \rangle \rightarrow ab \cdot</math></td><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \cdot ab</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot \langle C \rangle</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle B \rangle \rightarrow \cdot d</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle C \rangle \rightarrow \cdot a</math></td></tr> <tr><td style="padding: 2px;"></td><td style="padding: 2px;"><math>\langle C \rangle \rightarrow \cdot</math></td></tr> </table>	$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$	$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$	$\langle A \rangle \rightarrow ab \cdot$	$\langle A \rangle \rightarrow \cdot ab$		$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$		$\langle B \rangle \rightarrow \cdot \langle C \rangle$		$\langle B \rangle \rightarrow \cdot d$		$\langle C \rangle \rightarrow \cdot a$		$\langle C \rangle \rightarrow \cdot$
$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$	$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$																												
$\langle A \rangle \rightarrow ab \cdot$	$\langle A \rangle \rightarrow \cdot ab$																												
	$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$																												
	$\langle B \rangle \rightarrow \cdot \langle C \rangle$																												
	$\langle B \rangle \rightarrow \cdot d$																												
	$\langle C \rangle \rightarrow \cdot a$																												
	$\langle C \rangle \rightarrow \cdot b$																												
$\langle A \rangle \rightarrow \langle B \rangle \cdot c \langle D \rangle$	$\langle B \rangle \rightarrow \cdot \langle A \rangle \langle B \rangle c$																												
$\langle A \rangle \rightarrow ab \cdot$	$\langle A \rangle \rightarrow \cdot ab$																												
	$\langle A \rangle \rightarrow \cdot \langle B \rangle c \langle D \rangle$																												
	$\langle B \rangle \rightarrow \cdot \langle C \rangle$																												
	$\langle B \rangle \rightarrow \cdot d$																												
	$\langle C \rangle \rightarrow \cdot a$																												
	$\langle C \rangle \rightarrow \cdot$																												

Tiempo estimado: 4 horas.