

Procesadores de Lenguajes - Tema 7. Análisis semántico

Relación de ejercicios

BLOQUE 1

Ejercicio 1. Considera el siguiente esquema de traducción incompleto:

$$\begin{aligned} S &\rightarrow A \textbf{ fin } \{ \text{imprimir } S.val \} \\ A &\rightarrow \textbf{ abre } A \textbf{ cierra} \\ A &\rightarrow \textbf{ lit} \end{aligned}$$

y que el analizador léxico utilizado dota a los componentes de la categoría **lit** de un atributo *val* donde se almacena el valor del literal. Además, observa que toda secuencia de categorías léxicas aceptada por la gramática implícita en el esquema de traducción incluye un único literal **lit**.

Se debe completar el esquema con nuevas acciones semánticas para conseguir que conforme se realiza un análisis sintáctico ascendente lo que muestre la acción imprimir *S.val* sea el valor del literal presente en la secuencia de componentes utilizada.

Ejercicio 2. Considera la siguiente gramática:

$$\begin{aligned} Lis &\rightarrow Ini \textbf{ elem } Mas \textbf{ Fin} \\ Mas &\rightarrow \textbf{ sep elem } Mas \mid \lambda \\ Ini &\rightarrow \textbf{ apar } \mid \textbf{ acor} \\ Fin &\rightarrow \textbf{ cpar } \mid \textbf{ ccor} \end{aligned}$$

en la que los tokens **apar** y **cpar** corresponden a paréntesis (de apertura y cierre, respectivamente), **acor** y **ccor** corresponden a corchetes (de apertura y cierre, respectivamente) y **sep** corresponde al separador coma. Como se puede observar, esta gramática permite utilizar un delimitador de cierre de tipo distinto al utilizado en la apertura (abrir un corchete y cerrar un paréntesis, por ejemplo). Se desea controlar semánticamente esta circunstancia.

Por tanto, se deben añadir acciones semánticas a la gramática de forma que el esquema de traducción resultante puede calcular, mientras lleva a cabo un análisis sintáctico ascendente de su entrada, un atributo *Lis.ok* que indique si se ha utilizado el mismo tipo de limitador de apertura y cierre. Se deben tener en cuenta las siguientes restricciones:

- Todos los atributos usados deben ser de tipo entero.
- Sólo se permiten acciones semánticas en el extremo derecho de la parte derecha de las producciones.

Ejercicio 3. Considera la siguiente gramática, que genera un lenguaje similar al del ejercicio anterior:

$$\begin{aligned}\text{Lis} &\rightarrow \text{Ini elem Mas} \\ \text{Mas} &\rightarrow \text{sep elem Mas} \mid \text{Fin} \\ \text{Ini} &\rightarrow \text{apar} \mid \text{acor} \mid \text{alla} \\ \text{Fin} &\rightarrow \text{cpar} \mid \text{ccor} \mid \text{clla}\end{aligned}$$

El nuevo lenguaje utiliza llaves como delimitadores para apertura y cierre (mediante los tokens **alla** y **clla**) y la estructura de la gramática es ligeramente distinta.

Se deben añadir acciones semánticas a la gramática de forma que, mientras lleva a cabo un análisis sintáctico ascendente de la entrada, se llame a una función especial `trata_error` si se ha utilizado un delimitador de cierre de distinto tipo al de apertura. La llamada debe producirse desde acciones de la producción $\text{Mas} \rightarrow \text{Fin}$. Se debe tener en cuenta la siguiente restricción:

- Todos los atributos deben ser de tipo entero.

BLOQUE 2

Los ejercicios de este bloque se deben resolver teniendo en cuenta las siguientes restricciones:

- El esquema de traducción resultante debe ser aplicable mientras se lleva a cabo un análisis sintáctico ascendente de la entrada.
- Los únicos tipos permitidos para los atributos de los símbolos no terminales son entero y lógico.

Ejercicio 4. Se deben añadir acciones semánticas a cada una de las siguientes gramáticas para que en un atributo $S.na$ se calcule el número de letras **a** o **A** presentes en la entrada.

A. Una gramática de cadenas de letras **a**:

$$\begin{aligned}S &\rightarrow X \\ X &\rightarrow \mathbf{a} X \mid \lambda\end{aligned}$$

B. Una gramática de cadenas de letras **a**, **b** y **c**:

$$\begin{aligned}S &\rightarrow \mathbf{a} X \mathbf{b} \\ X &\rightarrow \mathbf{a} X X \mid \mathbf{b} \mathbf{a} X \mathbf{a} \mid \mathbf{c} Y \\ Y &\rightarrow \mathbf{c} Y \mid \mathbf{a}\end{aligned}$$

C. Una gramática en la que cada uno de los tokens asociados a sus tres símbolos terminales, **may** (una letra mayúscula cualquiera), **min** (una minúscula) y **dig** (un dígito), cuenta con un atributo *lex* donde se guarda el correspondiente lexema:

$$\begin{aligned} S &\rightarrow \mathbf{may} \ X \mid X \mid \mathbf{dig} \ Y \\ X &\rightarrow X \ \mathbf{min} \mid \lambda \\ Y &\rightarrow \mathbf{dig} \ Y \mid \lambda \end{aligned}$$

Ejercicio 5. Considera la siguiente gramática:

$$\begin{aligned} S &\rightarrow \mathbf{apar} \ A \ \mathbf{cpar} \\ A &\rightarrow \mathbf{apar} \ A \ A \ \mathbf{cpar} \\ A &\rightarrow \mathbf{x} \end{aligned}$$

en la que los tokens **apar** y **cpar** corresponden, respectivamente, a paréntesis de apertura y cierre.

El *nivel de profundidad* de un punto de la entrada puede definirse como el número de paréntesis que, habiendo sido abiertos con anterioridad a ese punto, aún no han sido cerrados.

A partir de la gramática anterior se deben construir dos esquemas de traducción distintos:

A. En el primero se deben añadir acciones semánticas, exclusivamente en el extremo derecho de la parte derecha de cada producción, para que se calcule en un atributo entero *S.mnp* el máximo nivel de profundidad alcanzado en la entrada.

B. En este caso, se deben añadir las acciones semánticas para que, cada vez que se analice un token **x**, se muestre en pantalla el nivel de profundidad en que se ha encontrado. Se necesitará para ello algún atributo heredado.

Ejercicio 6. Considera la siguiente gramática, que también tiene paréntesis **apar** y **cpar**:

$$\begin{aligned} S &\rightarrow A \\ A &\rightarrow \mathbf{apar} \ \mathbf{apar} \ A \ \mathbf{cpar} \ A \ \mathbf{cpar} \mid \mathbf{x} \ A \mid \mathbf{y} \ \mathbf{apar} \ \mathbf{x} \ A \ \mathbf{cpar} \mid \lambda \end{aligned}$$

Se deben añadir las acciones semánticas necesarias para calcular en el símbolo *S.next* el número de tokens **x** que aparecen a nivel de profundidad 3.

BLOQUE 3

Ejercicio 7. Sea L_1 el lenguaje de todas las cadenas formadas por cero o más letras minúsculas y que no tienen tres o más de tres letras b seguidas en su interior. Por ejemplo, las siguientes cadenas pertenecerían a L_1 : xyz, bb, bebebe, baobab, λ . No pertenecerían a L_1 estas otras cadenas: Gato, abbbba, xxxbbb, salu2.

Asumiremos una especificación léxica en la que se han definido dos únicas categorías, ambas emitidas por el analizador léxico y sin atributos: **be**, para la letra b minúscula, y **otra**, para cualquier letra minúscula distinta de la b. Asumiremos, además, la siguiente gramática:

$$\begin{aligned}C &\rightarrow M M^* \mid \lambda \\ M &\rightarrow \mathbf{be} \mid \mathbf{otra}\end{aligned}$$

Se deben añadir acciones semánticas a la gramática anterior de modo que el esquema de traducción resultante, ante cadenas aceptadas por la gramática y que no pertenezcan a L_1 , reaccione llamando a una función `trata_error` que se encargará de generar un mensaje adecuado y abortar el análisis. En el esquema de traducción debes utilizar atributos, y si lo crees necesario, también puedes usar variables locales.

Ejercicio 8. Sea un lenguaje definido mediante dos niveles, léxico y sintáctico, dados por la especificación siguiente:

Categoría léxica	Expresión regular	Atributos
a	[Aa]	--
be	[Bb]	--
ce	[Cc]	--
apar	(--
cpar)	--
lit	[0 - 9]*	<i>valor</i>

y la siguiente gramática:

$$\begin{aligned}S &\rightarrow L N \\ L &\rightarrow \mathbf{a a a L} \mid \mathbf{be a a L} \mid \mathbf{ce a} \\ N &\rightarrow \mathbf{apar lit cpar}\end{aligned}$$

Se deben añadir acciones semánticas a la gramática anterior para que el esquema de traducción resultante, mientras se lleva a cabo un análisis sintáctico de la entrada, pueda hacer los cálculos necesarios para acabar obteniendo en un atributo `S.ok` un valor lógico que identifique si el valor del literal entero entre paréntesis coincide exactamente con el número de letras a (en mayúscula o minúscula) de la cadena de caracteres analizada. Por ejemplo, debería obtenerse un valor lógico *verdadero* para casos como: `aAAca(4)` o `BaaBaaCA(5)`; sin embargo, para casos como: `ca(007)`, `aAAca(3)` o `baaAAbaaCA(11)`, el resultado del cálculo debería ser *falso*. Se debe tener en cuenta la siguiente restricción:

- Sólo se permiten atributos de tipo entero o lógico.

Ejercicio 9. Para poder gestionar los tipos de las expresiones modeladas mediante la siguiente gramática:

$$\begin{aligned} E &\rightarrow X \ M \\ X &\rightarrow \mathbf{litV} \\ X &\rightarrow \mathbf{litW} \\ X &\rightarrow \mathbf{apar} \ E \ \mathbf{cpar} \\ M &\rightarrow O \ E \\ M &\rightarrow \lambda \\ O &\rightarrow \mathbf{opA} \\ O &\rightarrow \mathbf{opB} \end{aligned}$$

habrá que tener en cuenta lo siguiente:

- Los operadores binarios **opA** y **opB** son asociativos por la derecha.
- Hay dos tipos, V y W , que cuentan con sus respectivos literales, **litV** y **litW**.
- Habrá que manejar también un tipo especial, *ERROR*, que será el asociado a las expresiones con algún tipo de error de tipos.
- El operador **opA** exige que sus dos operandos sean del mismo tipo y, entonces, ese será el tipo del resultado; en otro caso, se producirá un error de tipos.
- El operador **opB** produce un resultado de tipo W independientemente de si sus operandos son de tipo V o W ; sólo presentará error de tipos si alguno de sus operandos lo presenta.
- Encerrar una expresión entre paréntesis (componentes **apar** y **cpar**) no afecta a su tipo.

Por tanto, observa que cada una de las tres expresiones que se muestran a continuación como ejemplo tiene un tipo distinto:

- La expresión **litV opA litV** es de tipo V .
- La expresión **litW opA litV opB litV** es de tipo W .
- La expresión **apar litW opA litV cpar opB litV** es de tipo *ERROR*.

Debes añadir acciones semánticas a la gramática anterior para que el esquema de traducción resultante pueda calcular, mientras se lleva a cabo un análisis sintáctico de una expresión, un atributo $E.t$ que indique el tipo de esa expresión (V , W o *ERROR*) en el nodo raíz del correspondiente árbol de análisis. Se debe tener en cuenta la siguiente restricción:

- Sólo se permiten acciones semánticas en el extremo derecho de la parte derecha de cada producción.