

Procesadores de lenguajes

DDS - Generadores de traductores dirigidos por la sintaxis

Objetivos del ejercicio

- Consolidación de conocimientos adquiridos sobre definiciones dirigidas por la sintaxis (DDS)

Actividades a realizar

1. Familiarización con CUP y DDSs.
2. Ejercicio de DDSs con CUP para la evaluación de expresiones lógicas.

Traductores DDS con CUP

Atributos y acciones semánticas con CUP

CUP permite la introducción de acciones semánticas en la gramática mediante la combinación '{:' para la apertura de las acciones y ':' para el cierre de las mismas.

Dentro de ellas se debe insertar código escrito en lenguaje Java. En este código se podrá hacer referencia a los símbolos leídos en la regla, empleando el identificador que asignemos a cada uno (éste estará detrás del símbolo separado de él por ':'). Estos identificadores deben ser tratados como instancias de la clase a la que pertenecen, la cual queda determinada por el nombre de la clase que aparece en la declaración del símbolo.

Ejemplo parcial de una gramática donde determinamos si un número es mayor que otro evaluando una expresión de la forma "número > número". A continuación mostramos una indicación por terminal mediante una acción semántica.

```
//...
terminal String numero;

non terminal Simbolo C;
//...
K ::= C:eval
    { :
      if (eval.valor)
        System.out.println("El primer numero es mayor.");
      else
        System.out.println("El primer numero NO es mayor.");
    : }
```

```

C ::= numero:n1 opmayor numero:n2
{
    int v1=Integer.parseInt(n1);
    int v2=Integer.parseInt(n2);

    Simbolo s=new Simbolo();
    s.tipo=Simbolo.T_BOOLEAN;
    s.valor=(v1 > v2);

    RESULT = s; // Volcamos en el símbolo C la información
}

```

Las **acciones semánticas** aparecen destacadas en amarillo. En azul se encuentran marcados los **atributos sintetizados**.

La clase Símbolo es una clase que representa cada uno de los símbolos de la gramática que leemos. Su composición es flexible y depende de las necesidades del analizador que queramos desarrollar. En general, suele contener atributos que almacenan el tipo (entero, float, boolean...), la clase de símbolo que es (método, variable, constante, clase...), el identificador... En este caso presentado aquí, la clase Símbolo tiene un atributo 'valor' que guarda el valor de las expresiones reconocidas. Este atributo no tendría sentido en analizadores que leyeran códigos fuente de programas, pero sí lo tiene cuando lo que se leen son expresiones que se deben evaluar (es decir, cuando no nos limitamos a analizar su corrección léxica, sintáctica y semántica).

Enunciado

Se presenta una gramática para reconocer expresiones lógicas, cuyo resultado será un valor de verdad; esto es, verdadero o falso. Las expresiones lógicas vendrán dadas en un fichero modificable por el alumno para poder efectuar las pruebas que estime necesario. El archivo podrá contener un número ilimitado de expresiones, todas ellas finalizadas por un ';'.

Se pide completar la definición dirigida por la sintaxis para lograr evaluar las expresiones, obteniendo así el resultado de las mismas. Gramática:

```

A ::= L ; A | L ;

L ::= L and J | L or J | L nand J | L nor | L xor J | L xnor J | J

J ::= ( L ) | true | false | not ( L ) | C

C ::= numero == numero | numero != numero | numero < numero
    | numero > numero | numero <= numero | numero >= numero

```