



Práctica 4:

ANÁLISIS SINTÁCTICO: INTERPRETE DE C++

OBJETIVO

- Utilización del Prolog para el análisis sintáctico. Uso de las gramáticas de cláusulas definidas (DCG).
- Prolog como intérprete de otros lenguajes.

PLANTEAMIENTO DEL PROBLEMA

En esta práctica se debe realizar un intérprete del lenguaje C++ simplificado. Para ello se deberá escribir una gramática que reconozca un subconjunto del C++ y lo convierta a términos de Prolog. Posteriormente, estos términos serán interpretados y “ejecutados” por el propio Prolog.

El C++ que vamos a utilizar no admite ningún tipo de declaración: ni variables, ni tipos, ni funciones (excepto el main).

Las únicas instrucciones permitidas son:

- Asignación
- Bucle **while**
- **cin** >>
- **cout** <<

Donde **cin** únicamente admite una variable y **cout** únicamente permite mostrar una expresión numérica.

Además se admiten las construcciones:

- **int main(void) { ... }** Programa principal
- **{ ... }** Bloque de instrucciones

Las expresiones numéricas únicamente admiten constantes numéricas de 1 cifra, variables y los operadores: +, -, *, /.

Las expresiones lógicas sólo admiten los operadores: <, <=, >, >=, ==, !=.

Los identificadores sólo pueden ser de 1 letra minúscula.

En el siguiente ejemplo tenemos un programa que calcula el factorial de un número:

```
int main(void)
{
    f=1;
    i=1;
    cin >> n;

    while (i<=n)
    {
        f=f*i;
        i=i+1;
    }
    cout << f;
}
```



La gramática de C++ deberá reconocer este programa y transformarlo en una lista de términos de Prolog que representarán las instrucciones. El programa traducido quedará de la siguiente forma:

```
[asigna(f, 1) , asigna(i, 1) , lee(n) , while(i =< n, [asigna(f, f * i) ,  
asigna(i, i + 1)]) , escribe(n)]
```

Una vez esté el programa traducido, se le pasará al predicado *interprete/1*, cuya función es ejecutar cada una de las instrucciones.

En la página web de la asignatura se encuentra el fichero *utilpr4.pl*, que contiene algunos predicados útiles para la práctica y parte de la gramática a realizar, y el fichero *f.cpp*, que contiene el programa de ejemplo.

OPCIONAL

Modificar la gramática y el *interprete* para que también admita la construcción **for** de C++. El primer campo del **for** únicamente podrá ser una asignación, el segundo una expresión booleana y el tercero una asignación. Para incrementar la variable de control utilizaremos una asignación normal en vez del operador ++.

Reescribir el programa de ejemplo para que utilice un bucle **for** en lugar del **while** y comprobar que funciona correctamente.

PUNTUACIÓN DE LA PRÁCTICA

El apartado obligatorio es un 80 % de la nota y el apartado opcional un 20 %.

DURACIÓN DE LA PRÁCTICA:

1 Sesión.