

Práctica 5

Domingo Gómez y Cristina Tîrnăucă

Materiales de la práctica:

- Generador de parsers bison <http://gnuwin32.sourceforge.net/> para windows
- Compilador g++

Objetivos de la práctica

- Dar los primeros pasos en el uso práctico de las gramáticas libres de contexto.
- Conocer el generador de analizadores sintácticos Bison.
- Manejar la memoria dinámica, punteros y estructuras.
- Empezar a utilizar las gramáticas de atributos.

Enunciado de la práctica

1. Dado el siguiente código:

```
%code requires{  
  
#include <iostream>  
using namespace std;  
  
int yylex(int *v){  
    char c[1];  
    *v = 0; // valor semantico para uso futuro  
    if (cin >> c[0])  
        return int(c[0]);  
    else return 0; // fin de datos  
}  
  
}  
  
%%
```

```

T : S;

S : N ;

N : D
  | N D ;

D : '0' | '1' | '2' | '3' | '4'
  | '5' | '6' | '7' | '8' | '9' ;

%%
using namespace yy;

void parser::error (const location_type& loc ,
                   const string& msg
                   ){
    cout << msg << "\n";
    exit (1);
}

int main() {
    parser p;
    p.parse ();
}

```

Modificar la gramática anterior para que acepte expresiones con sumas y restas. Para ello, añadir una variable OP que sea + o -.

2. Utilizando la siguiente gramática,

```

%code requires{
    // Archivo de ejemplo para probar el automata
    // Para compilarlo utilizar el siguiente comando
    // bison -L C++ -d -t -r itemsets -g practica3.y
#include <iostream>
using namespace std;

int yylex(int *v){
    char c[1];
    *v = 0; // valor semantico para uso futuro
    if (cin >> c[0])
        return int(c[0]);
    else return 0; // fin de datos
}
}

```

```

%%

S : S '(' S ')' | ;

%%

using namespace yy;
void parser::error (const location_type& loc ,
                  const string& msg
                  ){
    cout << msg << "\n";
    exit (1);
}

int main() {
    parser p;
    p.set_debug_level(1);
    p.parse();
}

```

Ampliar la gramática para que acepte expresiones con corchetes, de forma que siempre se cierren paréntesis y corchetes. Por ejemplo, esta entrada tiene que ser aceptada ([]), pero esta no ([]). Realizar varias computaciones con palabras, tanto aceptadas como no aceptadas.