Informe Trabajo Práctico 2

TP2 – Técnicas de Compilación

Docente: Maximiliano A. Eschoyez.

Alumnos: BOZZANO, Felipe Gabriel.
OLMOS BOSSA, Francisco Augusto.

Institución: Universidad Blas Pascal.

Carrera: Ingeniería Informática.

Asignatura: Técnicas de Compilación.

Fecha entrega: 20/06/2022

TP2 – Técnicas de Compilación
BOZZANO, Felipe
OLMOS, Francisco

Problemática

Resumen

El objetivo de este Trabajo Práctico es agregar al Trabajo Práctico 1 la tabla de símbolos y el reporte de errores. El programa a desarrollar para esta etapa tiene como objetivo tomar un archivo de código fuente en una versión reducida del lenguaje C, posiblemente con errores léxicos, sintácticos y semánticos, y generar como salida el árbol Sintáctico (ANTLR) y la Tabla de Símbolos y reporte de errores.

Consigna

Dado un archivo de entrada en lenguaje C, se debe generar como salida el Árbol Sintáctico (ANTLR) correcto y mostrar por consola o archivo el contenido de la Tabla de Símbolos de cada contexto. Esto es válido únicamente cuando el archivo de entrada es correcto.

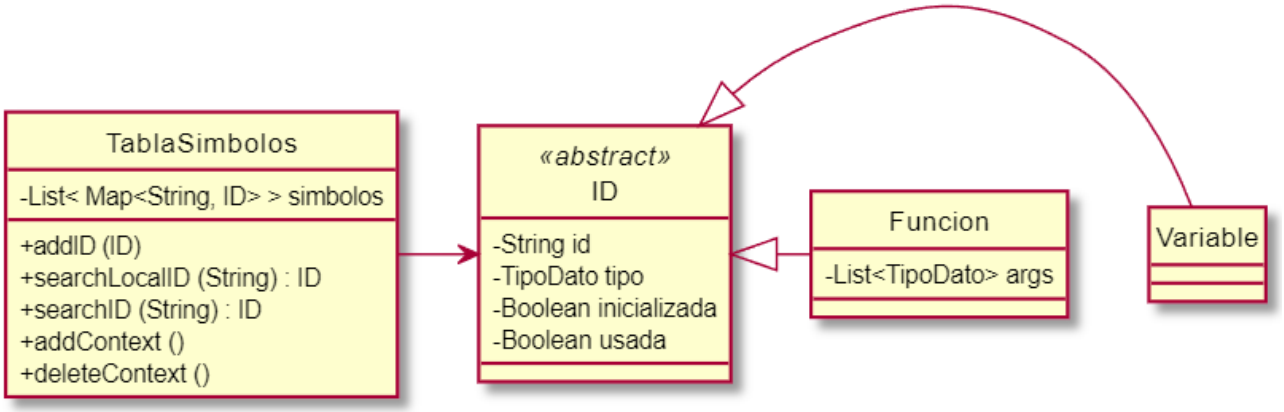
Para esta versión se deberá realizar un control de errores básicos y generar un reporte de lo encontrado para poder analizar archivos con posibles errores de codificación. El reporte de errores podrá realizarse por consola o archivo de texto. Los errores a detectar deben ser los siguientes:

- Errores sintácticos comunes:
  - Falta de un punto y coma,
  - Falta de apertura de paréntesis,
  - Formato incorrecto en lista de declaración de variables
- Errores semánticos comunes:
  - Doble declaración del mismo identificador,
  - Uso de un identificador no declarado,
  - Uso de un identificador sin inicializar,
  - Identificador declarado pero no usado,
  - Tipos de datos incompatibles.

Cada error debe reportarse si es sintáctico o semántico.

Solución

Partimos del TP-1, donde tuvimos que redefinir algunas reglas debido a que no podíamos tratarlas correctamente con los listener, y agregamos una tabla de símbolos para manejar contextos, variables y funciones del código.



Luego definimos un listener que se ejecuta a medida que recorremos el árbol sintáctico, el cual llena esta tabla de símbolos.

Por último, agregamos al listener un reporte de errores que se va dando a medida que se muestra por consola al mismo instante que se detectó el error.

Un ejemplo para comprender lo anteriormente mencionado es el siguiente:

A partir de la siguiente entrada:

```
int funcion2(double a, int c, int d){
  a = c + d;
  d = a;
  c = a;
  int a;
}

int main(double a, int b, int c){
  if(a<b){
    int hola;
    h = 5;
  }
  funcion2(a, b, c);
  int m;
  int l = m;
}
```

Obtenemos la siguiente salida por consola donde se pude ver que hay distintos errores semánticos.

```
INICIO DEL PARSEO
        INICIO DEL CONTEXTO GENERAL
                INICIO DEL CONTEXTO
                ERROR SEMANTICO: Doble declaración del mismo identificador -- ID: a
                a --- double a inicializada: true usada: true
                c --- int c inicializada: true usada: true
                d --- int d inicializada: true usada: true
                FIN DEL CONTEXTO
                INICIO DEL CONTEXTO
                INICIO DEL CONTEXTO
                ERROR SEMANTICO: Uso de un identificador no declarado -- ID: h
                ERROR SEMANTICO: Identificador declarado pero no usado
                        hola --- int hola inicializada: false usada: false
                FIN DEL CONTEXTO
                ERROR SEMANTICO: Uso de un identificador no inicializado -- ID: m
                a --- double a inicializada: true usada: true
                b --- int b inicializada: true usada: true
                c --- int c inicializada: true usada: true
                ERROR SEMANTICO: Identificador declarado pero no usado
                        l --- int l inicializada: true usada: false
                ERROR SEMANTICO: Identificador declarado pero no usado
                        m --- int m inicializada: false usada: false
                FIN DEL CONTEXTO
                funcion2 --- int funcion2 inicializada: false usada: true tipos args: [double , int , int ]
                ERROR SEMANTICO: Identificador declarado pero no usado
                        main --- int main inicializada: false usada: false tipos args: [double , int , int ]
        FIN DEL CONTEXTO GENERAL
FIN DEL PARSEO
```

.

Enlace a GitHub

Conclusión

A modo de conclusión, lo ideal es atomizar lo máximo posible las reglas sintácticas para luego facilitar el desarrollo del listener, porque sobre todo las reglas mas chicas son las que suelen llevar a cabo el reporte de errores.