

編譯器

Project 3

資工三 404410016 宋慶濤

壹、描述

這次延續了上一次的進度，在已有的 Parser 上加上了 Type Checker，讓專案能作語意分析(Semantic Analysis)。此外，也修改了先前完成的部份，把一些發現的 bug 都修好了，費時許久才完成。

貳、支援型態

這次作業主要是透過一個 hashmap 去紀錄使用的變數有沒有宣告、型態為何、賦值動作有沒有合法(等號左右型態是否相等)等等。支援型態如下：

attr_type & attr_number	
int	1
float	2
char	3
void	4
short	5
longlong	6
double	7
struct	8
doesn't exist	-1
error detected	-2

參、程式架構圖

- Program Body
 - Include Header Files /* 引入標頭檔 */
 - Sub-functions declarations (definitions) /* 子函式的宣告、定義 */
 - Codelines (Legal Statments) /* 任何合法的程式程式碼 */
 - vardeclarations // 變數的宣告
 - varassignments // 變數的賦值
 - callfunctions // 函數呼叫
 - expressions // 任何加減乘除、比較運算皆包含於此
 - ifconditions // if、if-else、if-elseif-else 敘述
 - forloops // for 迴圈
 - whileloops // while 迴圈
 - continues // loop 的 continue
 - breaks // loop 的 break
 - Return Value /* 函數回傳值 */
 - Main function
 - Codelines (Legal Statments) /* 任何合法的程式程式碼 */
 - vardeclarations // 變數的宣告
 - varassignments // 變數的賦值
 - callfunctions // 函數呼叫
 - expressions // 任何加減乘除、比較運算皆包含於此
 - ifconditions // if、if-else、if-elseif-else 敘述
 - forloops // for 迴圈
 - whileloops // while 迴圈
 - continues // loop 的 continue
 - breaks // loop 的 break
 - Return Value /* 函數回傳值 */
 - Sub-functions declarations (definitions)
 - 與上面相同，會上下兩邊都出現是因為自訂函數能夠在 main 函數之前宣告或者定義，也能夠在 main 函數之後宣告或者定義。

肆、Demo

請在作業資料夾目錄下先下達以下指令，來編譯這次的 program：

>> make compile

```
ecl@ecl-1700-linux:~/antlrPractice/semantic-analyzer$ make compile
java -cp antlr-3.5.2-complete.jar org.antlr.Tool myChecker.g
javac -cp antlr-3.5.2-complete.jar:. myChecker test.java
Note: ./myCheckerParser.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
ecl@ecl-1700-linux:~/antlrPractice/semantic-analyzer$
```

接下來可以透過下面指令透過預先寫好的測試程式來執行這次的 program：

>> make (test1 || test2 || test3)

由於關於 Syntax 的部份，在上一次作業都測過了(有沒有分號啊、函數有沒有回傳值、宣告型態存不存在等等)。這次的三個範例程式測試重點如下

- Test Program 1：測試重複宣告變數

```
1 // 測試重複宣告變數
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 int main()
6 {
7     // 宣告和初始化現在不能同時做
8     // 在一行宣告多餘兩個變數只抓的到頭尾的變數，最好還是一行一行宣告...
9     int x, y;
10    x = 10;
11    y = 20;
12    // 重複宣告變數
13    int x;
14    int y;
15    int z;
16    /* do something here */
17    return 0;
18 }
19
```

執行結果：

```
ecl@ecl-1700-linux:~/antlrPractice/semantic-analyzer$ make test1
java -cp antlr-3.5.2-complete.jar:. myChecker test test1.c
Type Error: 13: redeclared identifier x
Type Error: 14: redeclared identifier y
ecl@ecl-1700-linux:~/antlrPractice/semantic-analyzer$
```

- Test Program 2 : 測試等號左右兩邊型態不同的賦值行為

```
1 // 測試等號左右兩邊型態不同的賦值行為
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 int main()
6 {
7     int x;
8     x = 0.666
9     return 0;
10 }
```

執行結果：

```
ecl@ecl-1700-linux:~/antlrPractice/semantic-analyzer$ make test2
java -cp antlr-3.5.2-complete.jar:. myChecker test test2.c
Type Error: 8: types of the operands in the expressions are different
ecl@ecl-1700-linux:~/antlrPractice/semantic-analyzer$
```

- Test Program 3 : 測試使用沒有宣告的變數

```
1 // 測試使用沒有宣告的變數
2 #include <stdio.h>
3 #include <stdlib.h>
4 int main()
5 {
6     int x;
7     x = 10;
8     y = x;
9     return 0;
10 }
```

執行結果：

```
ecl@ecl-1700-linux:~/antlrPractice/semantic-analyzer$ make test3
java -cp antlr-3.5.2-complete.jar:. myChecker test test3.c
Type Error: 8: undeclared identifier y
ecl@ecl-1700-linux:~/antlrPractice/semantic-analyzer$
```

伍、待改善項目與其他應注意事項

- 目前在同一行宣告超過兩個變數的時候會有 bug，建議還是先一行宣告一個
- 一行只能夠做一個 assignment。不可以寫 `a = 10, b = 20, c = 30;`
- 宣告目前就只能宣告變數，沒辦法同時初始化。不可以寫 `int a = 10`
- 目前如果在 main 前面定義 function，裡面用到的 local variable 會被紀錄在 hash 之中，但是回到 main function 的時候，如果有使用相同名稱的變數，就會噴出錯誤，告訴你你重複宣告了，這點我還不知道怎麼改善。

陸、結語

以上就是關於這次 project 的報告，謝謝助教批改。