2018 Compiler Project #3. Semantic Analysis

소프트웨어전공 2016024793 김유진

1. Compilation method and environment

- Compilation: "make" for cminus
- Environment: Ubuntu 18.04.1 LTS

2. Source code description

<main.c>

Semantic Analysis 를 위해 main.c 옵션을 변경.

<symtab.c & symtab.h>

```
+ typedef struct LineListRec
+    { int lineno;
+      struct LineListRec * next;
+    } * LineList;
+
+ /* The record in the bucket lists for
+  * each variable, including name,
+  * assigned memory location, and
+  * the list of line numbers in which
+  * it appears in the source code
+  */
+ typedef struct BucketListRec
+    { char * name;
+      LineList lines;
+      TreeNode *treeNode;
+      int memloc ; /* memory location for variable */
+      struct BucketListRec * next;
+    } * BucketList;
+
+
+ typedef struct ScopeRec
+    { char * funcName;
+      int nestedLevel;
+      struct ScopeRec * parent;
+      BucketList hashTable[SIZE]; /* the hash table */
+    } * Scope;
```

BucketList 를 Wrapping 하는 Scope 구조체 구현하여 symtab.h 에 정의. 다른 파일에서 BucketList 와 LineList 구조체를 사용 할 수 있도록 헤더파일로 이동함.

```
35  + static Scope scopes[MAX_SCOPE];
36  + static int nScope = 0;
37  + static Scope scopeStack[MAX_SCOPE];
38  + static int nScopeStack = 0;
```

```
63  + Scope sc_create(char *funcName);
64  + Scope sc_top( void );
65  + void sc_pop( void );
66  + void sc_push( Scope scope );
67  +
```

scope 별로 stack 형태로 관리하기 위해 symtab.c 에 scopes 와 scopeStack 을 정의하고, 위와 같은 함수를 추가함.

<analyze.c & analyze.h>

built-in function 을 넣어주기 위해 insertIOFunc()을 구현. (각 함수의 lineno 를 -1 로 설정함.)

Compound state 마다 새로운 scope 를 생성하고 scopestack 에 push 함. Compound state 가 끝나면 pop 함. traverse 하며 undeclared variables 를 찾는 등의 symbolError()를 찾아줌.

traverse()함수를 사용하여 한 번 더 syntax tree 를 순회하며 각 node 를 방문해 type check 를 진행.

3. Example and result screenshots

- result of test.cm : 명세서의 test.cm 과 동일한 코드를 사용하여 테스트한 결과

```
CMINUS COMPILATION: test.cm

Building Symbol Table...

Symbol table:

<global scope> (nested level: 0)
Symbol Name    Sym.Type  Data Type    Line Numbers
------------  --------  -----------  ------------
main           Function  Void           11
input          Function  Integer        -1   14   14
output         Function  Void           -1   15
gcd            Function  Integer         4    7   15

function name: gcd (nested level: 1)
Symbol Name    Sym.Type  Data Type    Line Numbers
------------  --------  -----------  ------------
u              Variable  Integer         4    6    7    7
v              Variable  Integer         4    6    7    7    7

function name: main (nested level: 1)
Symbol Name    Sym.Type  Data Type    Line Numbers
------------  --------  -----------  ------------
x              Variable  Integer        13   14   15
y              Variable  Integer        13   14   15


Checking Types...

Type Checking Finished
```

- 구현하지 못 한 부분

배열에 Assignment 연산을 하면 Segmentation fault 가 발생함을 확인하였습니다.

symtab.c | analyze.c | test.cm | test3.cm

```
1  int x[10];
2  int a;
3
4  void main(void)
5  {
6      x[1] = 1;
7      b = 3;
8  }
```

```
clare@ubuntu:~/Desktop/test$ ./cminus test3.cm

CMINUS COMPILATION: test3.cm
Segmentation fault (core dumped)
```

또한 명세의 Implementation 조건 중 'Check if conditional of "If" or "While" has a value'사항 체크 시 Segmentation fault 가 발생합니다.

- 구현한 부분

명세서의 Implementation Notes 중 구현한 사항들

- Variables follow scope of each compound statement.

```
1  int scope (int a) {
2    if (a==1){
3      int a;
4      output(a);
5    }
6  }
```

```
clare@ubuntu: ~/Desktop/test
File  Edit  View  Search  Terminal  Help
clare@ubuntu:~/Desktop/test$ ./cminus test3.cm

CMINUS COMPILATION: test3.cm

Building Symbol Table...
a

Symbol table:

<global scope> (nested level: 0)
Symbol Name    Sym.Type  Data Type    Line Numbers
-------------  --------  -----------  ------------
scope          Function  Integer         1
input          Function  Integer        -1
output         Function  Void           -1    4

function name: scope (nested level: 1)
Symbol Name    Sym.Type  Data Type    Line Numbers
-------------  --------  -----------  ------------
a              Variable  Integer         1    2

function name: scope (nested level: 2)
Symbol Name    Sym.Type  Data Type    Line Numbers
-------------  --------  -----------  ------------
a              Variable  Integer         3    4


Checking Types...

Type Checking Finished
```

- Throws an error when an undeclared variable is used.

```
1   int x[10];
2   int a;
3
4   void main(void)
5   {
6       a = 1;
7       b = 3;
8   }
```

```
clare@ubuntu: ~/Desktop/test
File  Edit  View  Search  Terminal  Help
clare@ubuntu:~/Desktop/test$ ./cminus test2.cm

CMINUS COMPILATION: test2.cm
Segmentation fault (core dumped)
clare@ubuntu:~/Desktop/test$ ./cminus test3.cm

CMINUS COMPILATION: test3.cm

Building Symbol Table...
Symbol error at line 7: undelcared symbol
```

- Built-in functions should be always accessible.
- Type checking for functions and variables.
  - The type "void" is only available for functions. (syntax error 로 처리)

```
clare@ubuntu:~/Desktop/test$ ./cminus test3.cm

CMINUS COMPILATION: test3.cm
Syntax error at line 1: syntax error
Current token: =
clare@ubuntu:~/Desktop/test$ cat test3.cm
void a = 3;clare@ubuntu:~/Desktop/test$
```

  - Check return type.

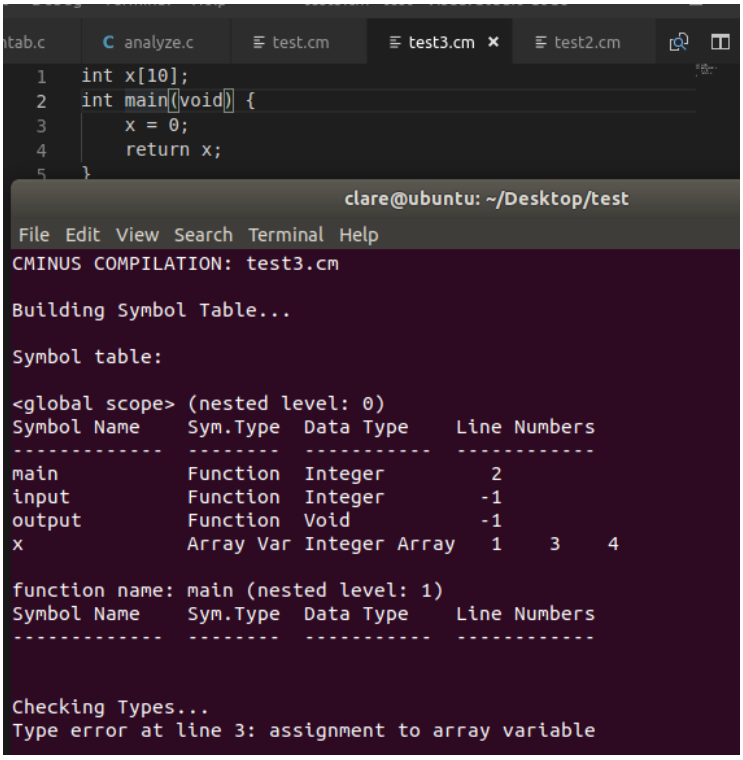
```
symtab.c  analyze.c  test.cm  test3.cm ×  test2.cm
1   int main(void) {
2       return;
3   }
```

```
clare@ubuntu: ~/Desktop/test
File  Edit  View  Search  Terminal  Help

CMINUS COMPILATION: test3.cm

Building Symbol Table...

Symbol table:

<global scope> (nested level: 0)
Symbol Name     Sym.Type   Data Type     Line Numbers
-------------   --------   -----------   ------------
main            Function   Integer          1
input           Function   Integer         -1
output          Function   Void            -1

function name: main (nested level: 1)
Symbol Name     Sym.Type   Data Type     Line Numbers
-------------   --------   -----------   ------------


Checking Types...
Type error at line 2: expected return value

Type Checking Finished
```

```
symtab.c ×  analyze.c  test.cm  test3.cm ×  test
1   void main(void) {
2       return 0;
3   }
```

```
clare@ubuntu: ~/Desktop/tes
File  Edit  View  Search  Terminal  Help

CMINUS COMPILATION: test3.cm

Building Symbol Table...

Symbol table:

<global scope> (nested level: 0)
Symbol Name     Sym.Type   Data Type     Line Numbers
-------------   --------   -----------   ------------
main            Function   Void             1
input           Function   Integer         -1
output          Function   Void            -1

function name: main (nested level: 1)
Symbol Name     Sym.Type   Data Type     Line Numbers
-------------   --------   -----------   ------------


Checking Types...
Type error at line 2: expected no return value

Type Checking Finished
```

- Verify the type match of two operands when assigning.



- Check the argument number when calling function.