C-Minus Semantic Analysis

2017029807 성창호

본 프로그램은 C-Minus의 Symbol Table과 Type Checker를 구현한다.

Project Environment

- Ubuntu 16.04.6 (WSL)

Overview

Symbol Table과 Type Checker 제작을 위해 Syntax Tree를 Traversal하고, 이를 바탕으로 Semantic Analysis 한다.

Scope

- 각 Compound Statement마다 Scope 적용
- 선언되지 않는 변수가 있다면 에러
- input, output 함수는 기본적으로 포함

Type Checker

- 함수에서 return type 확인
- Assign 시, 두 피연산자(operand)의 Type 일치 확인
- Function Call 시 argument 수 및 Type 확인
- If나 While의 Expression이 값을 가지는지 확인
- 그 외 다른 여러 가지를 C-Minus 문법을 참조하여 확인

Implementation

`main.c`

```
#define NO_ANALYZE FALSE

int EchoSource = FALSE;
int TraceScan = FALSE;
int TraceParse = FALSE;
int TraceAnalyze = TRUE;
int TraceCode = FALSE;
```

본 프로그램에서는 Semantic Analysis를 수행하므로 `main.c`의 flag들을 조정한다.

```
#if !NO_ANALYZE
  if (! Error)
  { fprintf(listing,"\n");
    buildSymtab(syntaxTree);
    typeCheck(syntaxTree);
    if (TraceAnalyze && !Error)
    { fprintf(listing,"\nBuilding Symbol Table...\n");
      fprintf(listing,"\nSymbol table:\n\n");
      printSymTab(listing);
      fprintf(listing,"\nChecking Types...\n");
      fprintf(listing,"\nType Checking Finished\n\n");
    }
  }
```

그리고 Error 발생 시, 모든 Error를 출력하고 Symbol Table은 출력하지 않기 위해 기존의 코드를 수정하였다.

symtab.h

```
typedef struct LineListRec
   { int lineno;
     struct LineListRec * next;
   } * LineList;

typedef struct BucketListRec
   { char * name;
     TreeNode * treeNode;  /* tree node that having variable */
     LineList lines;
     int memloc ; /* memory location for variable */
     struct BucketListRec * next;
   } * BucketList;

typedef struct ScopeListRec
   { char * funcName;
     BucketList hashTable[SIZE]; /* the hash table */
     struct ScopeListRec * parent;
     int nestedLevel;
   } * ScopeList;
```

Syntax Tree를 traversal 하면서 node을 저장할 BucketList 구조체과 이를 Wrapping할 ScopeList 구조체를 만든다.

symtab.c

```
/* Procedure st_insert inserts line numbers and
 * memory locations into the symbol table
 * loc = memory location is inserted only the
 * first time, otherwise ignored
 */
void st_insert( char * name, int lineno, int loc, TreeNode * treeNode );

/* Function st_lookup returns the memory
```

```
 * location of a variable or -1 if not found
 */
int st_lookup (char * name );
void st_add_lineno( char * name, int lineno );
int st_lookup_top ( char * name );

BucketList get_bucket ( char * name );

/* Stack for static scope */
ScopeList sc_create ( char * funcName );
ScopeList sc_top ( void );
void sc_pop ( void );
void sc_push ( ScopeList scope );
int addLocation ( void );

/* Procedure printSymTab prints a formatted
 * listing of the symbol table contents
 * to the listing file
 */
void printSymTab(FILE * listing);
void print_SymTab(FILE * listing);
void print_FuncTab(FILE * listing);
void print_Func_globVar(FILE * listing);
void print_FuncP_N_LoclVar(FILE * listing);
```

Static Scope를 구현하기 위해 Scope을 Stack으로 관리하는 함수들을 추가한다. 그리고 Symbol Table 출력을 위한 함수들도 추가한다.

`analyze.c`

```
/* print Error */
static void typeError(TreeNode * t, char * message);
static void symbolError(TreeNode * t, char * message);
static void undeclaredError(TreeNode * t);
static void redefinedError(TreeNode * t);
static void funcDeclNotGlobal(TreeNode * t);
static void voidVarError(TreeNode * t, char * name);

/* initialize function */
static void insertIOFuncNode(void);

static void afterInsertNode(TreeNode * t);
static void beforeCheckNode(TreeNode * t);
```

`insertNode` 함수에서 Compound State를 추가할 때 마다 새로운 Scope를 생성하여 Stack에 Push한다. 그리고 `afterInsertNode` 함수를 통해 Compound State를 빠져나갈 때 Stack을 Pop한다.

새로운 선언이 있을 경우, 현재의 Scope의 HashTable를 검사하여 중복이 있는지 확인한다. 또한 변수를 사용할 때는 현재 Scope Stack의 Top부터 탐색하여 해당 변수가 있는지 확인한다.

`globals.h`

```
typedef struct treeNode
   { struct treeNode * child[MAXCHILDREN];
     struct treeNode * sibling;
     int lineno;
     NodeKind nodekind;
     union { StmtKind stmt;
             ExpKind exp;
             DeclKind decl;
             ParamKind param;
             TypeKind type; } kind;
     union { TokenType op;
             TokenType type;
             int val;
             char * name;
             ArrayAttr arr;
             struct Scope * scope} attr;
     ExpType type; /* for type checking of exps */
   } TreeNode;
```

Tree Traversal 시, node를 통해 다른 Scope로 접근하는 경우가 발생하기 때문에 `attr` union에 Scope 구조체를 추가해주었다.

How to operate

```
$ make
$ ./cminus test.cm
```

Test Case

완벽한 Semantic Analysis를 위해 여러 가지 Error가 발생하는 Test Case를 생성하였다.

- 선언된 함수와 호출하는 함수의 인자 수가 맞지 않는 경우
- void type으로 변수가 선언된 경우
- type이 맞지 않는 두 변수가 연산하거나 assign되는 경우
- 변수가 선언되지 않은 경우
- void type의 함수에 integer 변수가 return 되는 경우
- main 함수 뒤에 함수가 선언되는 경우

Result

```
C-MINUS COMPILATION: sort.cm


Building Symbol Table...

Symbol table:
```

```
< Symbol Table >
Variable Name  Variable Type  Scope Name  Location  Line Numbers
-------------  -------------  ----------  --------  ------------
main           Function       global      5           35
sort           Function       global      4           21  42
input          Function       global      0            0  39
minloc         Function       global      3            4  27
output         Function       global      1            0  45
x              IntegerArray   global      2            3  39  42  45
low            Integer        minloc      1            4   8   9  10
a              IntegerArray   minloc      0            4   9  12  13
i              Integer        minloc      3            5  10  11  12  13  14  16  16
k              Integer        minloc      5            7   8  14  18
x              Integer        minloc      4            6   9  12  13
high           Integer        minloc      2            4  11
low            Integer        sort        1           21  24
a              IntegerArray   sort        0           21  27  28  29  29  30
i              Integer        sort        3           22  24  25  27  29  30  31  31
k              Integer        sort        4           23  27  28  29
high           Integer        sort        2           21  25  27
t              Integer        sort        0           26  28  30
i              Integer        main        0           36  37  38  39  40  40  43  44
45  46  46

< Function Table >
Function Name  Scope Name  Return Type  Parameter Name  Parameter Type
-------------  ----------  -----------  --------------  --------------
main           global      Void                         Void
sort           global      Void
                                        low             Integer
                                        a               IntegerArray
                                        high            Integer
input          global      Integer                      Void
minloc         global      Integer
                                        low             Integer
                                        a               IntegerArray
                                        high            Integer
output         global      Void
                                                        Integer

< Function and Global Variables >
   ID Name      ID Type    Data Type
-------------  ---------  -----------
main           Function   Void
sort           Function   Void
input          Function   Integer
minloc         Function   Integer
output         Function   Void
x              Variable   IntegerArray

< Function Parameter and Local Variables >
  Scope Name    Nested Level     ID Name      Data Type
```

```
-------------  ------------  -------------  -----------
minloc         1             low            Integer
minloc         1             a              IntegerArray
minloc         1             i              Integer
minloc         1             k              Integer
minloc         1             x              Integer
minloc         1             high           Integer

sort           1             low            Integer
sort           1             a              IntegerArray
sort           1             i              Integer
sort           1             k              Integer
sort           1             high           Integer

sort           2             t              Integer

main           1             i              Integer


Checking Types...

Type Checking Finished

================================================================================

C-MINUS COMPILATION: 1.cm


Building Symbol Table...

Symbol table:

< Symbol Table >
Variable Name  Variable Type  Scope Name  Location  Line Numbers
-------------  -------------  ----------  --------  ------------
main           Function       global      2            1
input          Function       global      0            0   8
output         Function       global      1            0  18
i              Integer        main        0            3   5   6   8  10  10  13  14
16  18
x              IntegerArray   main        1            3   8  16  18

< Function Table >
Function Name  Scope Name  Return Type  Parameter Name  Parameter Type
-------------  ----------  -----------  --------------  --------------
main           global      Void                         Void
input          global      Integer                      Void
output         global      Void
                                                        Integer

< Function and Global Variables >
   ID Name      ID Type    Data Type
-------------  ---------  -----------
```

```
main           Function   Void
input          Function   Integer
output         Function   Void

< Function Parameter and Local Variables >
  Scope Name    Nested Level     ID Name      Data Type
--------------  ------------  -------------  -----------
main            1             i              Integer
main            1             x              IntegerArray


Checking Types...

Type Checking Finished

============================================================================================

C-MINUS COMPILATION: 2.cm


Building Symbol Table...

Symbol table:

< Symbol Table >
Variable Name  Variable Type  Scope Name  Location  Line Numbers
-------------  -------------  ----------  --------  ------------
main           Function       global      3           11
input          Function       global      0            0  14  14
output         Function       global      1            0  15
gcd            Function       global      2            4   7  15
u              Integer        gcd         0            4   6   7   7
v              Integer        gcd         1            4   6   7   7   7
x              Integer        main        0           13  14  15
y              Integer        main        1           13  14  15

< Function Table >
Function Name  Scope Name  Return Type  Parameter Name  Parameter Type
-------------  ----------  -----------  --------------  --------------
main           global      Void                         Void
input          global      Integer                      Void
output         global      Void
                                                        Integer
gcd            global      Integer
                                        u               Integer
                                        v               Integer

< Function and Global Variables >
   ID Name      ID Type    Data Type
-------------  ---------  -----------
main           Function   Void
input          Function   Integer
output         Function   Void
```

```
gcd             Function    Integer

< Function Parameter and Local Variables >
  Scope Name     Nested Level      ID Name       Data Type
--------------  ------------  -------------  -----------
gcd              1             u              Integer
gcd              1             v              Integer

main             1             x              Integer
main             1             y              Integer


Checking Types...

Type Checking Finished

============================================================================================

C-MINUS COMPILATION: 3.cm


Building Symbol Table...

Symbol table:

< Symbol Table >
Variable Name  Variable Type  Scope Name  Location  Line Numbers
-------------  -------------  ----------  --------  ------------
input          Function       global      0            0
function       Function       global      4            3
i              Integer        global      3            2   4
aaa            IntegerArray   global      2            1   4
output         Function       global      1            0
a              Integer        function    0            3
b              Integer        function    1            3
c              IntegerArray   function    2            3   4
d              Integer        function    3            3

< Function Table >
Function Name  Scope Name  Return Type  Parameter Name  Parameter Type
-------------  ----------  -----------  --------------  --------------
input          global      Integer                      Void
function       global      Integer
                                        a               Integer
                                        b               Integer
                                        c               IntegerArray
                                        d               Integer
output         global      Void
                                                        Integer

< Function and Global Variables >
   ID Name       ID Type    Data Type
-------------  ---------  -----------
```

```
input          Function   Integer
function       Function   Integer
i              Variable   Integer
aaa            Variable   IntegerArray
output         Function   Void

< Function Parameter and Local Variables >
  Scope Name    Nested Level     ID Name      Data Type
--------------  ------------  -------------  -----------
function        1             a              Integer
function        1             b              Integer
function        1             c              IntegerArray
function        1             d              Integer


Checking Types...

Type Checking Finished

================================================================================

C-MINUS COMPILATION: 4.cm


Building Symbol Table...

Symbol table:

< Symbol Table >
Variable Name  Variable Type  Scope Name  Location  Line Numbers
-------------  -------------  ----------  --------  ------------
main           Function       global      6           37
input          Function       global      0            0
k              Integer        global      4            3
output         Function       global      1            0
x              Integer        global      2            1
y              Integer        global      3            2
abc            Function       global      5            5
dd             Integer        abc         5           10  29  29  32  33
ee             IntegerArray   abc         7           12  26  27  28  29  30  31  32
33
qre            Integer        abc         8           13  16
qwe            Integer        abc         0            5
aa             Integer        abc         2            7  18  20  21  26  26  30  34
zzz            IntegerArray   abc         6           11
bb             Integer        abc         3            8  18  27  27  30  31
lol            Integer        abc         1            5
cc             Integer        abc         4            9  15  20  28  28  31  32  33

< Function Table >
Function Name  Scope Name  Return Type  Parameter Name  Parameter Type
-------------  ----------  -----------  --------------  --------------
main           global      Integer                      Void
```

```
input          global      Integer                     Void
output         global      Void
                                                       Integer
abc            global      Integer
                                       qwe             Integer
                                       lol             Integer

< Function and Global Variables >
   ID Name      ID Type    Data Type
-------------  ---------  -----------
main           Function   Integer
input          Function   Integer
k              Variable   Integer
output         Function   Void
x              Variable   Integer
y              Variable   Integer
abc            Function   Integer

< Function Parameter and Local Variables >
  Scope Name    Nested Level     ID Name      Data Type
--------------  ------------  -------------  -----------
abc             1             dd             Integer
abc             1             ee             IntegerArray
abc             1             qre            Integer
abc             1             qwe            Integer
abc             1             aa             Integer
abc             1             zzz            IntegerArray
abc             1             bb             Integer
abc             1             lol            Integer
abc             1             cc             Integer


Checking Types...

Type Checking Finished

============================================================================================

C-MINUS COMPILATION: 5(mulit_func_error).cm

Error: Type error at line 12: invalid function call

============================================================================================

C-MINUS COMPILATION: 6(var_void_error).cm

Error: Variable Type cannot be Void at line 3 (name : x)

============================================================================================

C-MINUS COMPILATION: 7.cm
```

```
Building Symbol Table...

Symbol table:

< Symbol Table >
Variable Name  Variable Type  Scope Name  Location  Line Numbers
-------------  -------------  ----------  --------  ------------
main           Function       global      2            1
input          Function       global      0            0
output         Function       global      1            0
a              IntegerArray   main        0            3   4   5   6   6

< Function Table >
Function Name  Scope Name  Return Type  Parameter Name  Parameter Type
-------------  ----------  -----------  --------------  --------------
main           global      Integer                      Void
input          global      Integer                      Void
output         global      Void
                                                        Integer

< Function and Global Variables >
   ID Name      ID Type    Data Type
-------------  ---------  -----------
main           Function   Integer
input          Function   Integer
output         Function   Void

< Function Parameter and Local Variables >
  Scope Name    Nested Level     ID Name      Data Type
--------------  ------------  -------------  -----------
main            1             a              IntegerArray


Checking Types...

Type Checking Finished

================================================================================

C-MINUS COMPILATION: 8.cm


Building Symbol Table...

Symbol table:

< Symbol Table >
Variable Name  Variable Type  Scope Name  Location  Line Numbers
-------------  -------------  ----------  --------  ------------
main           Function       global      3            6
input          Function       global      0            0
f              Function       global      2            1   9
output         Function       global      1            0
```

```
a              Integer        main        0            8   9

< Function Table >
Function Name  Scope Name  Return Type  Parameter Name  Parameter Type
-------------  ----------  -----------  --------------  --------------
main           global      Void                         Void
input          global      Integer                      Void
f              global      Integer                      Void
output         global      Void
                                                        Integer

< Function and Global Variables >
   ID Name      ID Type    Data Type
-------------  ---------  -----------
main           Function   Void
input          Function   Integer
f              Function   Integer
output         Function   Void

< Function Parameter and Local Variables >
  Scope Name    Nested Level     ID Name      Data Type
--------------  ------------  -------------  -----------
main            1             a              Integer


Checking Types...

Type Checking Finished

=============================================================================================

C-MINUS COMPILATION: 9.cm

Error: Type error at line 3: operands have different type
Error: Type error at line 3: invalid variable type
Error: Type error at line 4: invalid return type
Error: Type error at line 16: invalid function call
Error: Type error at line 20: invalid function call

=============================================================================================

C-MINUS COMPILATION: 10(invalid_expression).cm

Error: Type error at line 6: operands have different type
Error: Type error at line 7: invalid variable type

=============================================================================================

C-MINUS COMPILATION: 11.cm


Building Symbol Table...
```

```
Symbol table:

< Symbol Table >
Variable Name  Variable Type  Scope Name  Location  Line Numbers
-------------  -------------  ----------  --------  ------------
main           Function       global      2            1
input          Function       global      0            0
output         Function       global      1            0
a              Integer        main        0            3   6
b              Integer        main        1            4
r              Integer        main        0            7   8

< Function Table >
Function Name  Scope Name  Return Type  Parameter Name  Parameter Type
-------------  ----------  -----------  --------------  --------------
main           global      Integer                      Void
input          global      Integer                      Void
output         global      Void
                                                        Integer

< Function and Global Variables >
   ID Name      ID Type    Data Type
-------------  ---------  -----------
main           Function   Integer
input          Function   Integer
output         Function   Void

< Function Parameter and Local Variables >
  Scope Name    Nested Level     ID Name      Data Type
--------------  ------------  -------------  -----------
main            1             a              Integer
main            1             b              Integer

main            2             r              Integer


Checking Types...

Type Checking Finished

============================================================================================

C-MINUS COMPILATION: 12(conflict).cm


Building Symbol Table...

Symbol table:

< Symbol Table >
Variable Name  Variable Type  Scope Name  Location  Line Numbers
-------------  -------------  ----------  --------  ------------
main           Function       global      2            1
```

```
input           Function       global      0            0
output          Function       global      1            0
a               Integer        main        0            2   3   4   5   7

< Function Table >
Function Name  Scope Name  Return Type  Parameter Name  Parameter Type
-------------  ----------  -----------  --------------  --------------
main           global      Void                         Void
input          global      Integer                      Void
output         global      Void
                                                        Integer

< Function and Global Variables >
   ID Name      ID Type    Data Type
-------------  ---------  -----------
main           Function   Void
input          Function   Integer
output         Function   Void

< Function Parameter and Local Variables >
  Scope Name    Nested Level     ID Name      Data Type
--------------  ------------  -------------  -----------
main            1             a              Integer


Checking Types...

Type Checking Finished

================================================================================================

C-MINUS COMPILATION: 13(undecl_ret_error).cm

Error: Undeclared Variable "x" at line 3
Error: Type error at line 3: invalid return type

================================================================================================

C-MINUS COMPILATION: 14.cm


Building Symbol Table...

Symbol table:

< Symbol Table >
Variable Name  Variable Type  Scope Name  Location  Line Numbers
-------------  -------------  ----------  --------  ------------
main           Function       global      2            1
input          Function       global      0            0  10
output         Function       global      1            0
a              IntegerArray   main        0            3  12
b              IntegerArray   main        1            4  12
```

```
c              Integer        main        2            6   7   8  11  11  12
d              Integer        main        0            9  10  12

< Function Table >
Function Name  Scope Name  Return Type  Parameter Name  Parameter Type
-------------  ----------  -----------  --------------  --------------
main           global      Integer                      Void
input          global      Integer                      Void
output         global      Void
                                                        Integer

< Function and Global Variables >
   ID Name      ID Type    Data Type
-------------  ---------  -----------
main           Function   Integer
input          Function   Integer
output         Function   Void

< Function Parameter and Local Variables >
  Scope Name    Nested Level     ID Name      Data Type
--------------  ------------  -------------  -----------
main            1             a              IntegerArray
main            1             b              IntegerArray
main            1             c              Integer

main            2             d              Integer


Checking Types...

Type Checking Finished

================================================================================

C-MINUS COMPILATION: 15(backward_func_decl_error).cm

Error: Undeclared Function "f" at line 3
Error: Type error at line 4: invalid return type
Error: Type error at line 8: invalid return type
```