

컴파일러 설계

Project #3. Semantic Analysis

Compilation method and Environment

> Ubuntu 16.04.1 LTS

gcc 컴파일러

Explanation about How to implement and How to operate

> bison을 이용해서 *y.tab.h*와 *y.tab.c*를 생성한 뒤 *makefile*에서 오브젝트 파일을 생성하여 기타 오브젝트 파일과 묶어서 실행파일 *cminus*를 생성했습니다.

생성된 *cminus*파일이 있는 **project**폴더 내에서 *./cminus 샘플파일명*을 입력하면 Semantic Analysis 결과가 출력됩니다..

Explanation about modified code

1. main.c

PDF에 명시되어있는 것과 같이 "Semantic Analysis" 부분만을 출력하기 위해서 플래그들을 조정해 주었습니다.

2. symtab.h

PDF에 명시되어있는 것과 같이 몇몇 함수들의 인자와 return type을 조정해주었고, 기존에 symtab.c에 있던 LineList구조체와 BucketList구조체를 해당 헤더파일로 옮겨주었습니다. 또한 scope를 표현하는 ScopeList구조체를 새로 작성했습니다.

3. symtab.c

- 1) 기존에 BucketList로 선언되어있던 hashTable을 더 큰 범위인 ScopeList 구조체로 바꿔주었습니다.
- 2) getParentScope라는 함수를 추가해주었습니다. Scope는 계층적인 구조를 가집니다. 예를

들어 main함수 내에 for문이나 while문과 같은 구문이 존재하면 해당 for문이나 while문은 main의 scope를 부모로 가집니다. 따라서 자식 scope는 부모 scope를 식별할 수 있어야 scope의 계층적 구조를 표현할 수 있기 때문에 해당함수를 선언해주었습니다.

- 3) st_insert 함수와 흡사한 형태를 가진 st_insert_함수를 추가해주었습니다. st_insert 함수는 기존 symtab.c의 함수형태와 다르게 parameter로 scope를 추가로 가지며, st_insert함수에서는 심볼테이블에 scope를 추가해 주는 역할을 수행합니다. 그리고 st_insert_ 함수에서는 해당 scope에 속해있는 bucketlist를 추가해 주는 역할을 수행합니다.
- 4) st_lookup함수를 수정해주었고(리턴 타입, 함수 내용) st_lookup_함수와 st_lookup_excluding_parent 함수를 추가해주었습니다. 위와 같은 이유로 st_lookup은 scope를 탐색하며, st_lookup_은 bucketlist를 탐색합니다. st_lookup_excluding_parent함수에서는 해당 scope의 parent를 배제한 탐색을 수행합니다.
- 5) printSymtab을 수정해주었습니다. 기존의 printSymtab 함수에서는 bucketlist만을 출력하기 때문에 scope 또한 출력해줄 수 있도록 printSymtab함수를 두가지로 나누어주었습니다. printSymtab에서는 scope들을 차례로 출력하며, printSymtab내에서 printSymtab_를 호출해서 해당 scope가 가지고 있는 bucketlist를 출력해줍니다.

4. analyze.c

- 1) 새로운 scope를 찾아주는 getScope함수를 작성했습니다. 새로운 scope가 형성되는 조건은 새로운 function노드가 나타나거나 새로운 compound노드가 나타나는 두가지이기 때문에 두 경우에 한해서만 새로운 scope를 설정해주었습니다. 또한 compound의 경우에는 scope를 lineno를 기준으로 하기 때문에 lineno또한 구해주었습니다.
- 2) traverse함수를 수정해주어 노드의 형제와 자식을 모두 탐색할 수 있도록 해주었습니다.
- 3) Nodekind에 따라 노드를 해당 scope의 bucketlist에 추가해주는 함수를 작성해주었습니다.
- 4) PDF에 명세되어있는 input과 output 두 가지 built-in 함수를 노드의 형태로 추가해주는 함수를 작성해주었습니다.
- 5) buildSymtab함수에서 syntaxtree에 위에 작성된 두 가지 built-in 함수를 추가할 수 있도록 4)의 함수를 추가해주었습니다.
- 6) 타입체킹을 하는 checkNode를 제가 작성한 형식에 맞게 수정해주었습니다. 제가 작성한 AST는 Stmt, Exp 뿐만 아니라 declare, parameter 노드 종류 또한 추가되었기 때문에 각각의 종류에 맞게 형식을 추가해주었습니다. Parameter나 변수의 type이 void가 되면 오류 메시지를 출력하도록 해주었고 assign을 할 때 right side의 변수의 타입이 void가 되면

오류 메시지를 출력하도록 해주었습니다.

5. cminus.y

Project2에서 작성한 cminus.y파일을 큰 틀은 수정하지 않되 미흡했던 부분들을 수정해주었습니다. 또한 기존에는 lineno를 중요하게 생각하지 않아 lineno를 빼먹은 부분이 많았는데 그 부분들을 수정해주었습니다.

Result Screenshot

```
C- COMPILATION: test/gcd.cm
Building Symbol Table...
Symbol table:
Scope name : ~:gcd
-----
Variable Name  Variable Type  Location  Line Numbers
-----
u              Integer      0         4    5    6    6
v              Integer      1         4    5    6    6    6
-----

Scope name : ~:output
-----
Variable Name  Variable Type  Location  Line Numbers
-----
arg            Integer      0         0
-----

Scope name : ~
-----
Variable Name  Variable Type  Location  Line Numbers
-----
main           Void          3         10
input          Integer      1         0    12    12
output         Void          0         0    13
gcd            Integer      2         4    6    13
-----

Scope name : ~:main:14
-----
Variable Name  Variable Type  Location  Line Numbers
-----
x              Integer      0        11    12    13
y              Integer      1        11    12    13
-----

Checking Types...
Type Checking Finished
hyomin@hyomin:~/2018_ELE4029_2013012041/project$
```

gcd.cm(예제파일)

```
hyomin@hyomin:~/2018_ELE4029_2013012041/project$ ./cminus test/sort.cm
```

```
C- COMPILATION: test/sort.cm
```

```
Building Symbol Table...
```

```
Symbol table:
```

```
Scope name : ~:minloc
```

Variable Name	Variable Type	Location	Line Numbers
low	Integer	1	3 5 6 7
a	Integer	0	3 6 9 10
high	Integer	2	3 8

```
Scope name : ~:output
```

Variable Name	Variable Type	Location	Line Numbers
arg	Integer	0	0

```
Scope name : ~:main:46
```

Variable Name	Variable Type	Location	Line Numbers
i	Integer	0	33 34 35 36 37 37 41 42 43 44 44

```
Scope name : ~:minloc:17
```

Variable Name	Variable Type	Location	Line Numbers
i	Integer	0	4 7 8 9 10 11 13 13
k	Integer	2	4 5 11 16
x	Integer	1	4 6 9 10

```
Scope name : ~:sort:30:29
```

Variable Name	Variable Type	Location	Line Numbers
t	Integer	0	23 25 27

```
Scope name : ~
```

Variable Name	Variable Type	Location	Line Numbers
main	Void	14	32
sort	Void	13	19 40
input	Integer	1	0 36
minloc	Integer	12	3 24
output	Void	0	0 43
x	Integer	2	2 36 40 43

```

Scope name : ~:sort
-----
Variable Name  Variable Type  Location  Line Numbers
-----
low            Integer       1         19   21
a              Integer       0         19   24   25   26   26   27
high           Integer       2         19   22   24
-----

Scope name : ~:sort:30
-----
Variable Name  Variable Type  Location  Line Numbers
-----
i              Integer       0         20   21   22   24   26   27   28   28
k              Integer       1         20   24   25   26
-----

Checking Types...

Type Checking Finished
hyomin@hyomin:~/2018_ELE4029_2013012041/project$ cd ..

```

Sort.cm(예제 파일)

```

Symbol table:

Scope name : ~:gcd
-----
Variable Name  Variable Type  Location  Line Numbers
-----
u              Integer       0         4    6    7    7
v              Integer       1         4    6    7    7    7
-----

Scope name : ~:output
-----
Variable Name  Variable Type  Location  Line Numbers
-----
arg            Integer       0         0
-----

Scope name : ~
-----
Variable Name  Variable Type  Location  Line Numbers
-----
main           Void          3         11
input          Integer       1         0    14   14
output         Void          0         0    15
gcd            Integer       2         4    7    15
-----

Scope name : ~:main:16
-----
Variable Name  Variable Type  Location  Line Numbers
-----
x              Integer       0         13   14   15
y              Integer       1         13   14   15
-----

Checking Types...

Type Checking Finished
hyomin@hyomin:~/2018_ELE4029_2013012041/project$

```

Test.cm

Symbol table:

Scope name : ~:gcd

Variable Name	Variable Type	Location	Line Numbers
u	Integer	0	4 6 7 7
v	Integer	1	4 6 7 7 7

Scope name : ~:output

Variable Name	Variable Type	Location	Line Numbers
arg	Integer	0	0

Scope name : ~

Variable Name	Variable Type	Location	Line Numbers
main	Void	3	11
input	Integer	1	0 14 14
output	Void	0	0 15
gcd	Integer	2	4 7 15

Scope name : ~:main:16

Variable Name	Variable Type	Location	Line Numbers
x	Integer	0	13 14 15
y	Integer	1	13 14 15

Checking Types...

Type Checking Finished

hyomin@hyomin:~/2018_ELE4029_2013012041/project\$

```
void main(void)
{
    int i; int x[5];

    i = 0;
    while( i+1 < 5-1 )
    {
        x[i] = input();

        i = i + 1;
    }

    i = 0;
    while( i <= 4 )
    {
        if( x[i] != 0 )
        {
            output(x[i]);
        }
    }
}
```

test.2.txt

```

C- COMPILATION: test/test2.cm
Building Symbol Table...
Symbol table:
Scope name : ~:output
-----
Variable Name  Variable Type  Location  Line Numbers
-----
arg            Integer       0         0
-----

Scope name : ~:main:8
-----
Variable Name  Variable Type  Location  Line Numbers
-----
i              Integer      2         6   7
x              Integer      0         5   7
y              Integer      1         5   7
-----

Scope name : ~
-----
Variable Name  Variable Type  Location  Line Numbers
-----
main           Integer       2         3
input          Integer       1         0
output         Void          0         0
-----

Checking Types...
Type Checking Finished
hyomin@hyomin:~/2018_ELE4029_2013012041/project$

```

```

1 /*hi*/
2
3 int main(void)
4 {
5     int x; int y;
6     int i;
7     i = x+y;
8 }

```

test2.cm