

Project #3 : Semantic Analysis

컴퓨터소프트웨어학부

2018008395

박정호

1. 실행 환경

우분투에서 진행했고, 버전은 16.04.12이다. GCC 버전은 다음과 같다.

컴파일은 명세에서 안내된 내용 그대로 Makefile을 사용했다.

```
pch@pch-VirtualBox:~$ gcc --version
gcc (Ubuntu 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

2. Implementation

A. Scope의 관리

모든 Symbol이 global scope에 속하는 기존의 Tiny와는 달리 C-minus는 local scope의 개념이 존재했고, 이로 인해 기존 scope의 형식을 고칠 필요가 있었다. 따라서 Scope라는 구조체를 새로 만들어서 해당 scope 내의 symbol을 관리하는 hash table, 상위 scope를 가리키는 포인터 등의 정보를 포함시켰다. 또한, TreeNode에 이 scope를 포함시켜서, 해당 노드가 포함된 scope가 어디인지를 저장하도록 했다.

또한 함수 내에서도 또다시 compound statement로 인해 nested scope가 생길 수 있었는데, 이를 위해서 스택을 이용했다. symbol table을 만드는 도중 compound statement를 만나게 되면, 새로운 Scope 구조체를 하나 만들고, 기존에 스택의 맨 위에 있던 scope를 parent로 갖게 했다. 또한 compound statement에 대해서 symbol table을 만드는 것이 끝나면, 스택에서 scope 하나를 제거했다. (traverse에서 postproc의 과정)

B. Semantic Error의 유형과 관리

Semantic Error의 유형은 크게 3가지가 있었다. 우선 가장 흔한 Type Error, 즉 expression의 type이 맞지 않는 것이 있고, Declaration Error, 즉 void 형 변수의 선언이나 변수의 재선언 등과 같은 잘못된 선언으로 인한 것이 있고, 마지막으로 Scope Error, 즉 해당 expression이 속한 scope에서는 볼 수 없는 symbol이 사용된 것이 있었다. 필자는 Declaration Error는 symbol table을 build하는 과정에서 찾아냈고, Scope Error와 Type Error는 symbol table을 다 만든 뒤에 진행되는 type checking에서 찾아냈다. error handling에 대한 내용은 다음 장에서 자세히 다루겠다.

C. Declaration Error의 관리

void형 변수나 재정의와 같은 error는 symbol table을 만들면서 진행했다.

우선 symbol의 재정의 여부를 판단하기 위해 변수 선언, 배열 선언, 함수 선언, 함수 내의 parameter 정의에서는 st_insert 이전에 st_lookup_excluding_parent 를 사용해서 같은 scope 내에 같은 이름의 symbol이 존재하는지에 대해서 체크했다. 만약 존재한다면 이는 재정의이므로 에러 메시지를 출력하고, 해당 symbol을 다시 symbol table에 추가하지는 않았다.

또한, 변수, 배열, parameter 선언에는 추가로 void 형인지 여부를 체크했다. 이 역시 st_insert 이전에 해당 노드에 저장한 type을 확인해서 VOID일 경우에는 에러 메시지를 출력하고 해당 symbol을 Invalid type으로 symbol table에 추가했다.

D. Scope Error의 관리

해당 symbol이 사용된 scope에서 st_lookup 을 사용해서 scope 내에 그 symbol이 존재하는지 확인했다. 만약 없다면 해당 symbol의 type을 Invalid로 두었다. 이는 과제 명세상의 예시 중 선언되지 않은 변수의 사용에 대해 "invalid~~~"라는 에러 메시지가 나온 것을 보고 이에 따라서 설계한 것이다. 즉, 잘못된 expression의 type을 Invalid로 설정하게 한 것이다.

E. Type Error의 관리

우선 모든 연산자의 피연산자는 integer type이어야 하기 때문에 이를 만족하지 못할 경우 에러 메시지를 출력했다. 또한, 배열의 접근에 있어서 그 인덱스가 integer type이 아닐 경우에도 에러 메시지를 출력했다. 그리고 if나 while에서 condition에 해당하는 부분이 integer type이 아닐 경우에도 에러 메시지를 출력했다.

F. 함수에서의 Type Error

함수에서는 return type과, 함수 호출 시에 parameter 개수, 형식 등에 있어서 type error가 발생할 수 있다. 따라서 이에 대한 type checking이 추가되어야 했다.

우선 return type을 check하기 위해 해당 scope의 이름을 이용해서 함수의 return type을 찾아냈다. 그리고 expression의 type과 함수의 return type을 비교해서 같지 않을 경우 에러 메시지를 출력했다.

parameter에 대한 오류에 대해서는 조금 더 복잡한 방식을 사용했는데, 우선 매개 변수 선언 시에, 그 매개변수가 속한 함수를 찾아서 해당 함수의 symbol table entry에 매개 변수의 형식을 저장했다. 그리고 함수 호출 시에 함수 이름을 이용해서 해당 함수의 symbol table entry의 params와 callK type의 노드의 자식인 args를 하나하나 type이 같은지 보며, 개수가 같은지도 비교했다. 만약 하나라도 같지 않다면 에러이다.

3. 실행 결과

실행에 사용한 input은 지난 과제에서 사용한 제공된 test1.txt, test2.txt, 이번 과제에서의 예시, 그리고 여기서 다루어지지 않은 Semantic Error를 포함한 몇가지 custom test case를 사용했음을 밝힌다. 각 테스트케이스의 내용물과 Semantic Analysis 결과를 같이 설명하도록 하겠다. Symbol Table은 공간 관계 상 출력하지 않겠다.

A. test1.txt, test2.txt

기준에 주어진 테스트이다.

C-minus 문법에 맞게 잘 짜여진 코드라 아무런 에러가 없는 것을 볼 수 있다.

```
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ cat test/test1.cm
/* A program to perform Euclid's
   Algorithm to computer gcd*/

int gcd (int u, int v)
{
    if (v == 0) return u;
    else return gcd(v, u-u/v*v);
    /*u-u/v*v == u mod v*/
}

void main(void)
{
    int x; int y;
    x = input(); y = input();
    output(gcd(x,y));
}
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ ./cminus test/test1.cm
C-MINUS COMPILATION: test/test1.cm

pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ cat test/test2.cm
void main(void)
{
    int i; int x[5];

    i = 0;

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

    i = 0;
    while( i <= 4 )
    {
        if( x[i] != 0 )
        {
            output(x[i]);
        }
    }
}
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ ./cminus test/test2.cm
C-MINUS COMPILATION: test/test2.cm
```

B. 과제 명세에서 주어진 예시

과제 명세의 example 부분에 있던 테스트 예제이다. 메시지가 동일하지는 않을 수 있으나, 결과적으로 에러를 잘 잡아내고 있음을 알 수 있다.

순서대로 type error(non-integer에 대한 수식), declaration error(void 형식의 변수 선언), type error(잘못된 개수의 arguments), scope error(선언되지 않은 변수)를 잘 잡아내는 것을 알 수 있다.

```
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ cat test/test7.cm
int main(void)
{
    int x;
    int y[3];

    x + y;

    return 0;
}
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ ./cminus test/test7.cm
C-MINUS COMPILATION: test/test7.cm
Type error at line 6: invalid expression

pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ cat test/test8.cm
int main(void)
{
    void x;
    return 0;
}
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ ./cminus test/test8.cm
C-MINUS COMPILATION: test/test8.cm
Declaration error at line 3: variable type cannot be void (name : x)

pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ cat test/test9.cm
int x(int y)
{
    return y + 1;
}

int main(void)
{
    int a;
    int b;
    int c;

    return x(a, b, c);
}
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ ./cminus test/test9.cm
C-MINUS COMPILATION: test/test9.cm
Type error at line 12: invalid function call

pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ cat test/test10.cm
int main(void)
{
    return x;
}
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ ./cminus test/test10.cm
C-MINUS COMPILATION: test/test10.cm
Undeclared symbol "x" at line 3
Type error at line 3: invalid return type
```

C. 배열 관련 type error

배열에 관한 에러를 잡는 테스트이다. 우선 []안에 integer가 아닌 값을 넣은 것을 잘 잡아내고 있고, 이로 인해 arr[arr]이 invalid해져서 assign도 invalid해진 것을 볼 수 있다.

```
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ cat test/test11.cm
int arr[10];

void main(void){
    int main;

    arr[arr] = 1;
    arr[1] = arr;
    main = 1;
    arr = 1;
}
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ ./cminus test/test11.cm
C-MINUS COMPILATION: test/test11.cm
Type error at line 6: invalid array index
Type error at line 6: invalid expression
Type error at line 7: invalid expression
Type error at line 9: invalid expression
```

D. Scope 관련 에러

main 함수 아래에 a가 global scope로 선언된 것을 볼 수 있다. 다만, C-minus는 forward reference를 허용하지 않으므로 a를 사용한 main 함수 내의 문장들이 모두 에러가 발생하는 것을 볼 수 있다. 그리고 이로 인해 a가 invalid expression이 되어서 if나 while의 condition으로 사용하는 것 또한 에러로 취급되는 것을 확인할 수 있다.

```
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ cat test/test6.cm
void main(void)
{
    if (a)
        while (a)
            if (a)
                ;
        else
            ;
}

int a;
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ ./cminus test/test6.cm

C-MINUS COMPILATION: test/test6.cm
Undeclared symbol "a" at line 3
Undeclared symbol "a" at line 4
Undeclared symbol "a" at line 5
Type error at line 5: if test is not integer
Type error at line 4: while test is not integer
Type error at line 3: if test is not integer
```

E. 함수 관련 에러

return type과 parameter type 관련 테스트이다.

우선 return type에 관해서는 int 형 함수에서 void를 return할 경우와 void형 함수에서 int를 return하는 경우 모두 에러가 발생하는 것을 볼 수 있다.

parameter type에 관해선, 잘못된 arguments를 사용한 fa 호출은 에러가 나지만, 올바른 arguments를 보낸 fb는 오류가 나지 않은 것을 볼 수 있다.

```
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ cat test/test14.cm
int fa(int a[], int b){
    return;
}

void fb(int c, int d[]){
    return c;
}

void main(void){
    int arr[10];
    int num;

    fa(num, num);
    fb(num, arr);
}
pch@pch-VirtualBox:~/2020_ELE4029_2018008395/3_Semantic$ ./cminus test/test14.cm

C-MINUS COMPILATION: test/test14.cm
Type error at line 2: invalid return type
Type error at line 6: invalid return type
Type error at line 13: wrong parameter type
```