

Project #2. Parser

컴퓨터소프트웨어학부 2020079689 신다혜

1. Environment

- Windows 11 Pro, Ubuntu 16.04 LTS
- Visual Studio Code 1.71.2
- gcc (Ubuntu 5.4.0-6ubuntu1~16.04.12) 5.4.0
- lex 2.6.0
- bison (GNU Bison) 3.0.4

2. Implementation

- *globals.h*

```
65 typedef enum {StmtK, ExpK, DeclK} NodeKind;
66 typedef enum {IfK, IfelseK, CompK, WhileK, ReturnK} StmtKind;
67 typedef enum {OpK, ConstK, IdK, AssignK, CallK, ArrK} ExpKind;
68 typedef enum {VarK, FuncK, ParamK, VoidParamK} DeclKind;
69
70 /* ExpType is used for type checking */
71 typedef enum {Void, Integer, Boolean, VoidArr, IntegerArr} ExpType;
```

DeclKind (declaration kind) – variable, function, parameter, void parameter 추가

StmtKind (statement kind) – cminus 문법에 맞게 repeat, read, write 삭제 및 if else, compound, while, return 추가

ExpKind (expression kind) – assign, call, array 추가

NodeKind (node kind) – DeclK 추가

ExpType (expression type) – void array, integer array 추가

```
75 typedef struct treeNode
76 { struct treeNode * child[MAXCHILDREN];
77   struct treeNode * sibling;
78   int lineno;
79   NodeKind nodekind;
80   union { StmtKind stmt; ExpKind exp; DeclKind decl; } kind;
81   union { TokenType op;
82           int val;
83           char * name; } attr;
84   ExpType type; /* for type checking of exps */
85 } TreeNode;
```

union kind – DeclKind decl 추가

- *util.c*

```
125  TreeNode * newDeclNode(DeclKind kind)
126  {  TreeNode * t = (TreeNode *) malloc(sizeof(TreeNode));
127      int i;
128      if (t==NULL)
129          fprintf(listing,"Out of memory error at line %d\n",lineno);
130      else {
131          for (i=0;i<MAXCHILDREN;i++) t->child[i] = NULL;
132          t->sibling = NULL;
133          t->nodekind = DeclK;
134          t->kind.decl = kind;
135          t->lineno = lineno;
136      }
137      return t;
138
139  }
```

globals.h에서 DeclKind를 추가했으므로 util.c에 newDeclNode method를 추가한다.

printTree(TreeNode*) pseudo code

```
while (tree != NULL) :
    if (tree->nodekind == StmtK) :
        tree->kind.stmt에 적합한 statement 출력
    elif (tree->nodekind == ExpK) :
        tree->kind.exp에 적합한 형태 출력
    elif (tree->nodekind == DeclK) :
        tree->kind.decl에 적합한 형태 출력
    else:
        "Unknown node kind" 출력
    for(i=0; i<MAXCHILDREN;i++) :
        printTree(tree->child[i])
    tree=tree->sibling
```

- *cminus.y*

C-Minus에 맞도록 tiny.y 파일의 token, grammar를 수정.

```
52  id      : ID
53      {
54          savedName = copyString(tokenString);
55          savedLineNo = lineno;
56      }
57      ;
58  num      : NUM
59      {
60          savedNum = atoi(tokenString);
61          savedLineNo = lineno;
62      }
```

하나의 line에 여러 개의 ID나 NUM이 있을 때 attr.name, attr.val이 제대로 출력되지 않아 ID와 NUM을 별개의 grammar rule로 분리.

```

207 if_stmt      : IF LPAREN exp RPAREN stmt %prec WOELSE
208               { $$ = newStmtNode(IfK);
209                 $$->child[0] = $3;
210                 $$->child[1] = $5;
211               }
212               | IF LPAREN exp RPAREN stmt ELSE stmt
213               { $$ = newStmtNode(IfelseK);
214                 $$->child[0] = $3;
215                 $$->child[1] = $5;
216                 $$->child[2] = $7;
217               }
218               ;

```

IF state와 IFELSE state의 구분을 위해 %nonassoc WOELSE, ELSE 사용

```

124 params       : param_list { $$ = $1; }
125               | VOID
126               {
127                 $$ = newDeclNode(VoidParamK);
128                 $$->lineno = lineno;
129               }

```

(cminus.y)

```

243             case VoidParamK:
244                 fprintf(listing,"Void Parameter\n");
245                 break;

```

(util.c)

void parameter 처리

```

186 stmt_list     : stmt_list stmt
187               {
188                 YYSTYPE t = $1;
189                 if (t != NULL)
190                 { while (t->sibling != NULL)
191                     t = t->sibling;
192                   t->sibling = $2;
193                   $$ = $1; }
194                 else $$ = $2;
195               }
207 if_stmt      : IF LPAREN exp RPAREN stmt %prec WOELSE
208               { $$ = newStmtNode(IfK);
209                 $$->child[0] = $3;
210                 $$->child[1] = $5;
211               }

```

TreeNode의 sibling과 child를 적절히 활용

3. Run

```

$ make
$ ./cminus_parser test.cm

```

- test2.cm, error_test.cm도 위 방법으로 동일하게 실행

4. Result

test.cm

```
test.cm
1  /* A program to perform Euclid's
2     Algorithm to computer gcd */
3
4  int gcd (int u, int v)
5  {
6      if (v == 0) return u;
7      else return gcd(v, u-u/v*v);
8      /* u-u/v*v == u mod v */
9  }
10
11 void main(void)
12 {
13     int x; int y;
14     x = input(); y = input();
15     output(gcd(x,y));
16 }
17
```

result (test.cm)

```
sdahye@ubuntu:~/Desktop/ele4029temp/parser$ ./cminus_parser test.cm
C-MINUS COMPILATION: test.cm

Syntax tree:
Function Declaration: name = gcd, return type = int
  Parameter: name = u, type = int
  Parameter: name = v, type = int
  Compound Statement:
    If-Else Statement:
      Op: ==
        Variable: name = v
        Const: 0
      Return Statement:
        Variable: name = u
      Return Statement:
        Call: function name = gcd
        Variable: name = v
      Op: -
        Variable: name = u
      Op: *
        Op: /
          Variable: name = u
          Variable: name = v
          Variable: name = v
Function Declaration: name = main, return type = void
  Void Parameter
  Compound Statement:
    Variable Declaration: name = x, type = int
    Variable Declaration: name = y, type = int
    Assign:
      Variable: name = x
      Call: function name = input
    Assign:
      Variable: name = y
      Call: function name = input
    Call: function name = output
    Call: function name = gcd
      Variable: name = x
      Variable: name = y
```

test2.cm

```
test2.cm
1  void main(void) {
2      /*
3          1
4          2 3    4
5          5 ***** 6
6          */
7
8      int i; int j;
9      i=0;
10     j=2;
11     while(i<j)
12     {
13         i = (i+1);
14     }
15 }
16
```

result (test2.cm)

```
sdahye@ubuntu:~/Desktop/ele4029temp/parser$ ./cminus_parser test2.cm
C-MINUS COMPILATION: test2.cm

Syntax tree:
Function Declaration: name = main, return type = void
  Void Parameter
  Compound Statement:
    Variable Declaration: name = i, type = int
    Variable Declaration: name = j, type = int
    Assign:
      Variable: name = i
      Const: 0
    Assign:
      Variable: name = j
      Const: 2
    While Statement:
      Op: <
        Variable: name = i
        Variable: name = j
      Compound Statement:
        Assign:
          Variable: name = i
          Op: +
            Variable: name = i
            Const: 1
```

error_test.cm

```
error_test.cm
1  /* Semantic Error Example */
2  /* (1) void-type variable a, b
3     * (2) uninitialized variable c (and b)
4     * (3) undefined variable d
5
6  int main ( void a[] )
7  {
8      void b;
9      int c;
10     d[1] = b + c;
11 }
12
```

result (error_test.cm)

```
sdahye@ubuntu:~/Desktop/ele4029temp/parser$ ./cminus_parser error_test.cm
C-MINUS COMPILATION: error_test.cm
Syntax error at line 11: syntax error
Current token: EOF

Syntax tree:
```