

Project #2. Parser

2016025041 하태성

- Compilation Environment

- macOS Monterey 12.2.1(M1 Pro 16GB)
- clang : Apple clang version 13.1.6 (clang-1316.0.21.2.5)
- flex : flex 2.6.4 Apple(flex-34)

- C-minus Parser Implementation

- main.c
 - main.c 파일은 명세와 같이 이전 프로젝트와 마찬가지로 플러그만 바꿔주었습니다.
- globals.h
 - globals.h 파일에는 이제 token을 받아와서 syntax tree을 구성하는데 사용될 노드들을 프로젝트에 맞게 수정하였습니다. 주어진 명세에서 크게 독자적으로 벗어나기보다 'AST and Output Format'에 서술된 그래프를 기초로 필요한 state를 추가하였습니다.

```
typedef enum {StmtK, ExpK, DecK} NodeKind;
typedef enum {IfK, CompK, WhileK, RetK} StmtKind;
typedef enum {OpK, ConstK, IdK, AssignK, ArrK, CallK} ExpKind;
typedef enum {VarK, FunK, ParamK} DecKind;

/* ExpType is used for type checking */
typedef enum {Void, Integer} ExpType;
```

• util.c / util.h

- util.c 파일에서는 앞서 globals.h에 새로 추가된 Declaration Node를 위한 constructor, newDecNode()를 다른 constructor와 비슷하게 만들어주었습니다.
- printTree() 함수도 기존의 Stmt, Exp 노드 출력형식을 프로젝트 형식에 맞게 바꾸어주고 Dec 노드도 출력할 수 있도록 새로 추가해주었습니다. 크게 까다로운 점은 없었고 새로 추가된 Dec노드의 type를 출력하는 부분과 Parameter가 array일 경우 constant를 출력하지않도록 예외처리를 하는 부분이 조금 추가되었을뿐 대부분 기존의 함수 구조와 동일하게 구현하였습니다.

```
/* function newDecNode creates a new expression
 * node for syntax tree construction
 */
TreeNode *newDecNode(DecKind kind)
{
    TreeNode *t = (TreeNode *)malloc(sizeof(TreeNode));
    int i;
    if (t == NULL)
        fprintf(listing, "Out of memory error at line %d\n", lineno);
    else
    {
        for (i = 0; i < MAXCHILDREN; i++)
            t->child[i] = NULL;
        t->sibling = NULL;
        t->nodekind = DecK;
        t->kind.dec = kind;
        t->lineno = lineno;
    }
    return t;
}

/* if (tree->nodekind == DecK)
 *
 * switch (tree->kind.dec)
 * {
 *     case VarK:
 *         fprintf(listing, "Variable Declaration: name = %s, ", tree->attr.name);
 *         printDecType(tree);
 *         break;
 *     case FunK:
 *         fprintf(listing, "Function Declaration: name = %s, return type = ", tree->attr.name);
 *         switch (tree->type)
 *         {
 *             case Void:
 *                 fprintf(listing, "void\n");
 *                 break;
 *             case Integer:
 *                 fprintf(listing, "int\n");
 *                 break;
 *             default:
 *                 fprintf(listing, "error type\n");
 *                 break;
 *         }
 *         break;
 *     case Param:
 *         if (tree->attr.name == NULL) fprintf(listing, "void Parameter\n");
 *         else {
 *             fprintf(listing, "Parameter: name = %s, ", tree->attr.name);
 *             printDecType(tree);
 *             if (tree->child[0] == NULL && tree->child[0] == NULL) flag = 0;
 *         }
 *         break;
 *     default:
 *         fprintf(listing, "Unknown ExpNode kind\n");
 *         break;
 * }
 }
```

- cminus.y

- cminus.y에는 이번 프로젝트의 핵심이라고도 볼 수 있는 주어진 BNF 문법을 가지고 C-minus definition과 rules, actions을 정의해주었습니다.
- 먼저 오른쪽과 같이 사용할 token들을 정의해주었고 특이사항으로 NOELSE라는 토큰을 Dangling Else Problem을 해결하기 위해 추가하였습니다. 결론적으로 '%prec'를 사용해서 우선순위를 지정해줌으로써 해결이 가능하였습니다.

```
%token NOELSE
%token IF ELSE WHILE RETURN INT VOID
%token ID NUM
%token EQ NE LT LE GT GE
%token PLUS MINUS
%token TIMES OVER SEMI COMMA
%token LPAREN RPAREN LCURLY RCURLY LBRACE RBRACE ASSIGN
%token ERROR

%nonassoc NOELSE
%nonassoc ELSE

%left PLUS MINUS
%left TIMES OVER COMMA

select_stmt : IF LPAREN exp RPAREN stmt %prec NOELSE
{
    $$ = newStmtNode(IfK);
    $$->child[0] = $3;
    $$->child[1] = $5;
}
| IF LPAREN exp RPAREN stmt ELSE stmt
{
    $$ = newStmtNode(IfK);
    $$->child[0] = $3;
    $$->child[1] = $5;
    $$->child[2] = $7;
}
;
```

- 기본적인 Rule과 Action들은 주어진 BNF 문법과 Node 양식, 그리고 기존에 있던 tiny.y의 내용을 바탕으로 서술하였습니다.
- ID와 NUM이 포함된 경우에는 모든 non-terminal이 나온 후에 action을 취하게 되면 savedname이나 val에 들어가야될 값이 아닌 다른 값으로 바뀌게되기 때문에 ID나 NUM이 나오고 다음 non-terminal이 나오기전에 바로 action을 통해 id에 저장된 savedname을 노드의 attr.name에 넣어주었습니다.
- parameter의 경우 위에서 잠깐 서술한 것과 같이 출력시에 예외처리를 해주기 위해 attr.val에 평소엔 사용하지 않을 '\0'을 넣어주었습니다.

```
var_dec : type_spec id SEMI
{
    $$ = newDecNode(VarK);
    $$->attr.name = savedname;
    $$->type = $1->type;
}
| type_spec id
{
    $$ = newDecNode(VarK);
    $$->attr.name = savedname;
    $$->type = $1->type;
}
LBRACE NUM
{
    $$ = $3;
    $$->child[0] = newExpNode(ConstK);
    $$->child[0]->attr.val = atoi(tokenString);
}
RBRACE SEMI { $$ = $6; }
```

- Test & Result

- test.1.txt

```
./cminus_parser test.1.txt
C-MINUS COMPILATION: ./test.1.txt

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
```

- test.2.txt

```
./cminus_parser test.2.txt
C-MINUS COMPILATION: ./test.2.txt

Syntax tree:
Function Declaration: name = main, return type = void
Void Parameter
Compound Statement:
Variable Declaration: name = i, type = int
Variable Declaration: name = x, type = int[]
Const: 5
Assign:
Variable: name = i
Const: 0
While Statement:
Op: <
Variable: name = i
Const: 5
Compound Statement:
Assign:
Variable: name = x
Variable: name = i
Call: function name = input
Assign:
Variable: name = i
Op: +
Variable: name = i
Const: 1
Assign:
Variable: name = i
Const: 0
While Statement:
Op: <
Variable: name = i
Const: 4
Compound Statement:
If Statement:
Op: !=
Variable: name = x
Variable: name = i
Const: 0
Compound Statement:
Call: function name = output
Variable: name = x
Variable: name = i
```

사진과 같이 주어진 테스트는 정상적으로 모두 작동하는 것을 확인하였고, 실행결과를 모두 첨부하진 않았지만 그 외 dangling else problem, semantic error code 등의 문제들도 테스트를 해보았을 때 정상적으로 parsing이 되는 것을 모두 확인하였습니다.)