Parser Report

2019027001

문 석 원

**I. 컴파일 환경 및 방법**

**1. 컴파일 환경**

- WSL 2 (Windows Subsystem for Linux 2)

- Ubuntu 18.04.5 LTS

- bison (GNU Bison) 3.0.4

**2. 컴파일 방법**

- git clone <https://hconnect.hanyang.ac.kr/2021_ele4029_11216/2021_ele4029_2019027001.git>

- cd 2021\_ele4029\_2019027001/2\_Parser

- make

- ./cminus\_parser ./[테스트\_파일명]

**II. 구현 및 동작 방식**

**1. globals.h**

typedef enum {StmtK,ExpK} NodeKind;

typedef enum {IfK,IfElseK,WhileK,ReturnK,AssignK,VarDeclK,FunDeclK,VoidParamK,ParamK,CompoundK,CallK} StmtKind;

typedef enum {OpK,ConstK,IdK,TypeK} ExpKind;

typedef enum {Void,Integer,VoidPtr,IntegerPtr} ExpType;

기존 Tiny와 같이, Stmt Node와 Exp Node를 분류하되, C-Minus의 Context Free Grammar에 맞게 StmtKind와 ExpKind에 새로운 Node Type를 추가하였으며, void[] type과 int[] type 또한 지원해야 하므로, ExpType에 VoidPtr type과 IntegerPtr type을 추가했다.

**2. util.c**

util.c의 printTree 함수를 globals.h에 정의된 Node Kind에 맞게 수정했다.

**3. cminus.l**

char tokenString[2][MAXTOKENLEN+1];

char\* savedNameStack[MAXRECURSION];

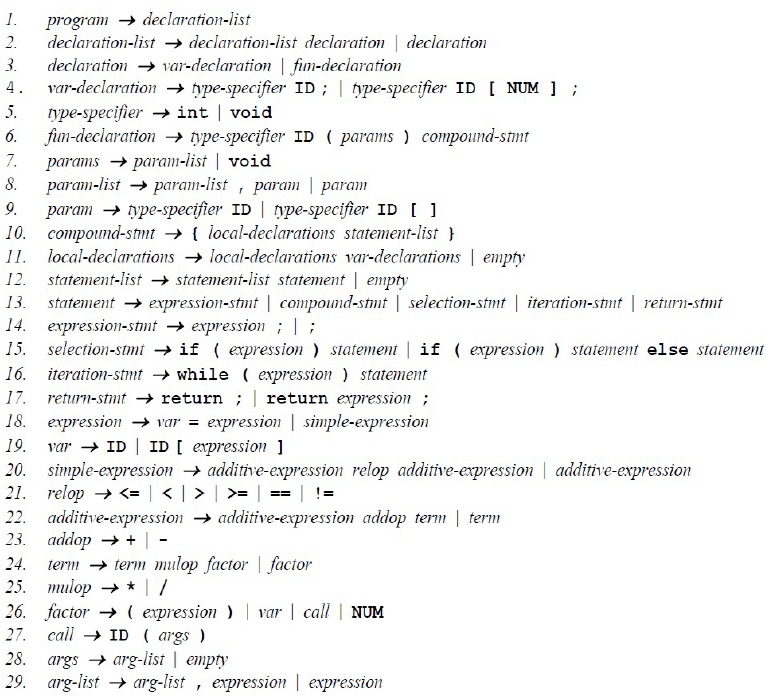
int curTokenPos = 0;

int curStackTop = 0;

변수명을 저장하기 위해 copyString을 하기 전에 이미 lookahead를 봐 버린 경우, tokenString에 ID 대신 세미콜론이 들어오는 이슈가 있어, tokenString을 2칸짜리 배열로 저장하여, 해당 사례에서도 ID를 제대로 찾을 수 있도록 했다.

또한, 자식 노드에서 copyString을 하는 경우, 기존 tiny의 savedName을 단독으로 사용하는 경우 자식 노드의 ID가 부모노드에도 그대로 덮어씌워지는 이슈가 있어, savedName을 stack구조로 바꾸었다. Stack의 최대 크기는 2000으로 설정되어 있다.

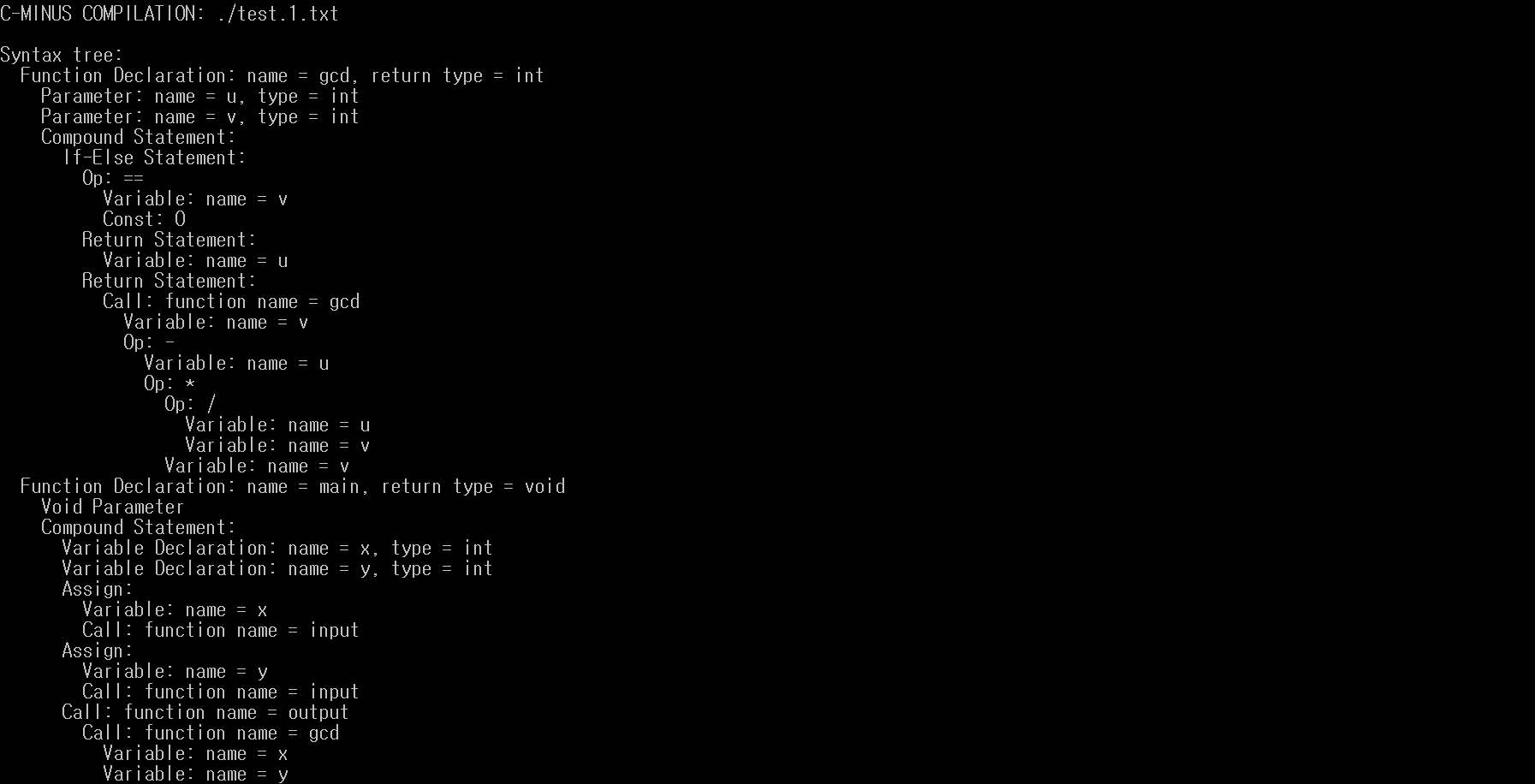
**4. cminus.y**

****

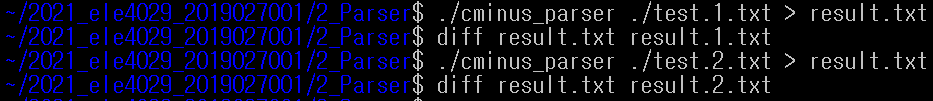
cminus.y에서는 위에 정의된 Context Free Grammar를 실제로 구현한다. 이 중에서 StmtNode를 생성하는 Non-terminal은 var\_declaration (VarDeclK), fun\_declaration (FunDeclK), params (Void-ParamK), param (ParamK), compound\_stmt (CompoundK), selection\_stmt (IfK, IfElseK), iteration\_stmt (WhileK), return\_stmt (ReturnK), expression (AssignK), call (CallK)가 있으며, ExpNode를 생성하는 Non-terminal은 type\_specifier (TypeK), var (IdK), relop, addop, mulop (OpK), 그리고 Terminal은 NUM (ConstK)가 있다.

**III. 구현 검증**

구현 검증을 위해, 제공된 테스트케이스(test.1.txt, test.2.txt)를 구현한 Parser로 read해보았다.



./cminus\_parser test.1.txt 실행 결과 result.1.txt와 동일한 결과가 나왔으며, test.2.txt 또한 마찬가지였다.



C-Minus Parser로 읽은 결과를 result.txt에 담고, 이를 result.1.txt, result.2.txt와 비교해본 결과, 결과가 완전히 동일함을 알 수 있다.