PROJECT #2 “Yacc” Parser

2017029970 우원진

#1 Compilation method

텍스트이(가) 표시된 사진

자동 생성된 설명

위 사진과 같이 CommandLine에 make를 입력하면 컴파일이 자동으로 된다. 텍스트이(가) 표시된 사진

자동 생성된 설명그러면 이와같이 실행에 필요한 파일들이 만들어진다.

#2 How to Operate

텍스트이(가) 표시된 사진

자동 생성된 설명

이게 Makefile인데 기존 pdf의 Makefile에서 OBJS 에 analyze.o와 cgen.o를 필요없다고 생각해서 뺐다. 처음에는 Lexer의 모든수행이 끝난후 Parser가 동작한다고 생각했는데 실제로 해보니 yyparse()함수에서 yylex()를 부르면서 번갈아가며 실행된다는것을 알았다.

#3 Modified Code

Cminus.l 파일은 이전 프로젝트의 파일이지만 감점이 되어서 잘못된 부분을 조금 고쳤다. 그리고 이번 프로젝트를위한 수정한 파일은 globals.h , util.c , util.h , cminus.y , main.c이다.

# globals.h

먼저 globals.h는 yacc폴더의 globals.h 를 가져와 수정한 것이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

기존에는 StmtKind와 ExpKind만 있었지만 pdf의 Ciminus Appendix를 충족하기위해 DeclKind(Declaration), ParamKind(Parameter), TypeKind(Type)를 추가했다.

StmtKind(Statement)에서는 RepeatK, ReadK, WriteK와 같은 사용하지않는것들은 지우고, 새로운 IfEK(if .. else ..), CompK(Compound), IterK(Iteration), RetK(Return)을 추가했다. 그리고 AssignK는 Statement 보다는 Expression에 더 맞다고 생각해서 ExpKind로 옮겼다.

ExpKind(Expression)는 AssignK, IdK, ArrIdK, CallK 를 추가했다.

DeclKind는 VarK(변수선언), FuncK(함수선언), ArrVarK(배열변수선언)으로 구성했다.

ParamKind는 ArrParamK(Array형태의 Parameter)와 NonArrParamK로 구성했다.

TypeKind는 TypeNameK로 여기에 해당되는것은 INT 와 VOID 이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Struct Node에 위에 추가한 kind들을 추가해줬다. 그리고 ArrayAttr 타입의 arr도 attr에 넣어줬는데 이는 변수나 이런것이 배열로 선언되면 그 크기와, 이름, TokenType를 한번에 저장하기 위한것이다.(union 특성상 하나씩 밖에 저장을 못해서 한번에 저장하도록 했다.)

텍스트이(가) 표시된 사진

자동 생성된 설명

# util.h

텍스트이(가) 표시된 사진

자동 생성된 설명

위는 TreeNode를 초기화하는 함수들인데 globals.h에 Stmt, Exp 말고도, TreeNode의 종류에 DeclarationNode, ParameterNode, TypeNode를 추가했기 때문에 이들을 초기화하는 함수들도 추가해줬다.

# util.c

Scanner Project에서 했던 util.c를 가져와 고쳤다.

텍스트이(가) 표시된 사진

자동 생성된 설명

기존에 StmtNode와, ExpNode를 초기화하는 함수 newStmtNode, newExpNode와 같이 DeclNode, ParamNode, TypeNode를 초기화하는 함수 newDeclNode, newParamNode, newTypeNode를 추가했다.

그리고 PrintTree함수에도 DeclNode, ParamNode, TypeNode일 경우를 추가했고, 각 Node에서의 토큰의 종류별로 Pdf의 예시와 비슷하게 출력하도록 했다.

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

#cminus.y

텍스트이(가) 표시된 사진

자동 생성된 설명

먼저 변수가 배열로 선언 되었을때, 그 크기를 저장해 놓기위해 savedNumber라는 변수를 하나 만들었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

먼저 나올수있는 Terminal들을 정의해줬다.

그리고 밑에서 IF .. 와 IF … ELSE 에대한 shift/reduce conflicts를 해결하기 위해 %nonassoc을 이용해 NO\_ELSE와 ELSE 사이에 precedence를 주었다.

그리고 %left PLUS MINUS, %left TIMES OVER, %right ASSIGN 을 통해 PLUS, MINUS 가 TIMES, OVER보다 우선순위가 낮다는걸 표현했고, PLUS, MINUS, TIMES, OVER은 왼쪽결합법칙을, ASSIGN은 오른쪽결합법칙을 따른다는 것을 표현했다.

그리고 나머지 코드들은 pdf의 Appendix를 Yacc에 맞도록 표현한것인데,

텍스트이(가) 표시된 사진

자동 생성된 설명

원래의 ID, NUM 에서 이 값들을 변수에 저장하기위해 두개의 non-terminal들을 추가로 정의했다.

텍스트이(가) 표시된 사진

자동 생성된 설명

여기 IF LPAREN exp RPAREN stmt 까지 들어왔을때 shift/reduce conflicts가 발생한다. 이를 해결하기 위해 아까 위에서 정의했던 %nonassoc NO\_ELSE , %nonassoc ELSE를 통해 이 Conflicts를 해결했다.

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

이 각각의 코드들은 원래의 Appendix에서 각각 (relop -> LE | LT | GT | GE | EQ | NE), (simple\_exp -> add\_exp relop add\_exp | add\_exp) 와 (add\_op -> PLUS | MINUS), (add\_exp -> add\_exp add\_op term | term) 과 (mulop -> TIMES | OVER), (term -> term mulop factor | factor) 이지만 이를 합쳐서 각각 하나의 simple\_exp, add\_exp, term 이라는 non-terminal로 표현한 것이다.

#Example and Result

Example로 사용한 코드는 Pdf의 예시에 있는 코드 입니다.

**Result**

텍스트이(가) 표시된 사진

자동 생성된 설명