**컴파일러설계 Project2 Report**



컴퓨터소프트웨어학부

2018007974 김서연

1. 컴파일 방법과 환경

- 주어진 Makefile을 이용해 make함

![텍스트이(가) 표시된 사진

자동 생성된 설명]()

- 환경: Ubuntu 16.04.7 / gcc 5.4.0

2. 구현 방법과 동작 방식

main.c

- NO\_ANALYZE와 TraceParse를 TRUE로 설정함

globals.h

- NodeKind에 DeclK를 추가함: 변수나 함수를 선언할 때 사용할 노드 종류로, VarK, FunK, ParamK를 둠

- StmtKind에 compound statement를 위한 CompK, if와 if-else를 위한 SelectK, while을 위한 IterK, return을 위한 ReturnK를 둠

- ExpKind에 assignment expression을 위한 AssignK, call expression을 위한 CallK, operator를 위한 OpK, NUM을 위한 ConstK, ID를 위한 IdK, variable 생성을 위한 VarEK, type을 위한 TypeK를 둠

- DeclKind에 variable declaration을 위한 VarK, function declaration을 위한 FuncK, parameter를 위한 ParamK를 둠

- ExpType에 int 배열과 void 배열을 의미하는 IntArrK와 VoidArrK를 추가함

utils.c

- 새로운 Declaration 노드를 만드는 newDeclNode()를 만듦

- ExpType에 맞게 type의 출력 형태(type 배열의 인덱스)를 정하는 get\_type\_index()를 만듦

- printTree()를 수정해서 원하는 결과대로 출력하도록 함

: 노드의 종류에 따라 출력 형태를 결정하고, if와 if-Else의 차이는 SelectK에서 child의 개수로, non-value return과 return의 차이는 ReturnK에서 child의 개수로 판단하도록 함, 또한 ParamK의 경우 attr.name이 NULL인지를 통해 void parameter인지의 여부를 판단하도록 함

cminus.y

- tiny.y를 적절히 수정함

- savedName과 비슷한 역할을 하는 변수 saveNum을 만듦

- token들을 우선순위와 associativity에 맞게 정의함, 이 때 dangling else problem이 생기지 않도록 하기 위해 %nonassoc NOTELSE와 ELSE를 만들어 selection\_stmt에서 사용함

- declaration\_list나 statement\_list 등의 list는 기존의 tiny.y와 같은 방법으로 구현함

- 노드의 이름은 copyString()을 이용해 tokenString을 savedName에 복사해둔 뒤 적절한 때에 attr.name에 다시 savedName을 복사하는 식으로 구현했으나 conflict가 생기지 않는 선에서 intra-rule action을 사용하기도 함

- 상수 값 역시 atoi()를 이용해 tokenString을 savedNum에 복사해둔 뒤 필요할 때 사용함

- empty의 경우 $$에 NULL을 저장하는 방식으로 구현함

- 전체적인 문법은 주어진 AST format에 맞는 노드 구조가 되도록 작성함

- TypeK 노드: type\_specifier

- VarK 노드: var\_declaration

- FuncK 노드: fun\_declaration

- ParamK 노드: params, param

- CompK 노드: compound\_stmt

- SelectK 노드: selection\_stmt

- IterK 노드: iteration\_stmtn

- ReturnK 노드: return\_stmt

- AssignK 노드: expression

- VarEK 노드: var

- OpK 노드: relop, addop, mulop

- ConstK 노드: factor (factor -> num일 때)

- Call 노드: call

scan.c / cminus.l

- Project1에서 잘못된 부분을 수정함

① scan.c

: 주석 안에 ‘\* /’가 있을 때 무시되지 않는 것과 주석 안에서 EOF를 만나면 무한루프가 생기는 것을 수정함

② cminul.l

: 주석 안에 ‘\* /’가 있을 때 무시되지 않는 것을 수정함

3. 예제 적용 결과

- 주어진 dangling else problem 예제와 test.2.txt를 조금 변형하여 테스트함

- test. txt는 아래와 같음

![텍스트이(가) 표시된 사진

자동 생성된 설명]()

- 실행 결과는 아래와 같음

![텍스트이(가) 표시된 사진

자동 생성된 설명]()