Compiler Parser Report

|  |  |
| --- | --- |
| 학부 | 컴퓨터소프트웨어학부 |
| 학번 | 2018008977 |
| 이름 | 이종범 |

# Environment

Compiler: clang 14.0.3

Flex: 2.6.4 Apple(flex-34)

OS: Sonoma 14.1

Device: Apple MacBook Pro M1 Max 32GB

# How to Run?

* 경로 이동 후 Compile

$ make all

* 파일 실행

$ ./cminus\_parser [C-Minus File to Run]

# 주요 구현 사항

* cminus.y 파일을 수정하여 C-Minus Parser가 정상적으로 동작하도록 함
* 추가적인 테스트 코드를 작성하고, 자동화 스크립트를 통해 테스트 함

# cminus.y 구현 내용

* fun\_declaration
* 텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

  자동 생성된 설명해당 노드의 Child로 params와 compound\_stmt 지정
* type과 name은 $1과 $2의 값을 사용
* $1과 $2는 AST의 노드로 포함되지 않으므로 동적할당 해제
* params
* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명param\_list의 경우 해당 노드를 params로 사용
* VOID의 경우 Params 노드를 생성하고, VoidParameter로 출력될 수 있도록 flag를 True로 설정
* param\_list
* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명param\_list의 가장 끝 sibling을 param으로 지정
* declaration\_list 로직을 참고하여 적용
* param의 경우 노드 그대로 param\_list로 할당
* param

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* Params 노드를 생성하고, type과 name을 각각 $1과 $2에서 가져옴
* $1과 $2는 값을 AST에 포함되지 않으므로 동적 할당 해제
* LBRACE와 RBRACE가 포함된 경우에는 type\_specifier를 보고 대응되는 Type의 Array를 Type으로 지정
* compound\_stmt
* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명CompoundStmt 노드를 생성하고, child로 local\_declarations과 statement\_list를 지정
* local\_declarations
* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명local\_declaration의 가장 끝 sibling을 var\_declaration으로 지정
* Declaration\_list 로직을 참교하여 적용
* Empty의 경우 노드 그대로 statement\_list로 할당
* statement
* 텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

  자동 생성된 설명노드 그대로 statement로 할당
* selection\_stmt
* 텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

  자동 생성된 설명IF ELSE Statement의 경우 expression, statement, statement를 child로 지정
* Flag를 TRUE로 설정하여 IF Statement로 출력될 수 있도록 함
* 일반 IF Statement의 경우 expression과 statement를 child로 지정
* 폰트, 텍스트, 스크린샷, 라인이(가) 표시된 사진

  자동 생성된 설명expression\_stmt
* expression SEMI의 경우 expression 노드를 그대로 할당
* SEMI의 경우 노드에 NULL을 할당
* iteration\_stmt
* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명WhileStmt 노드를 만들고 expression과 statement를 child로 지정
* return\_stmt

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

* ReturnStmt 노드 생성
* return 하고자 하는 expression이 있는 경우 expression을 child 노드로 이를 지정
* non-value return의 경우 flag를 TRUE로 설정
* expression
* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명Assignment Statement의 경우 var과 expression을 child로 지정
* simple\_expression인 경우 노드를 그대로 사용
* var
* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명VarAccessExpr 노드를 생성
* 배열 변수의 Access일 경우, expression도을 child로 지정
* simple\_expression
* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명relop의 opcode를 opcode로 그대로 활용
* 양변을 child로 사용
* additive\_expression만 존재하는 경우 그대로 노드로 할당
* relop
  + 각 연산자에 대응하는 opcode를 지정
* additive\_expression
* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명$2의 opcode 그대로 사용하고 양변을 child로 지정
* 단일 terrm의 경우 Node로 그대로 사용
* addop
  + +, -에 각각 대응되는 opcode를 노드에 지정
* term
  + additive\_expression과 동일
* mulop
  + \*, /에 각각 대응되는 opcode를 노드에 지정
* factor
  + 각 단일 노드를 그대로 사용
* call
  + identifier에서 호출하고자 하는 함수 이름을 가져오고, args를 child로 사용
* args
  + 각 단일 노드를 그대로 사용
* args\_list
  + arg\_list의 가장 끝 sibling을 expression으로 지정
  + declaration\_list 로직을 참고하여 적용
  + expression의 경우 노드 그대로 arg\_list로 할당

# Test

* 제공된 Example 코드와 추가적인 C-Minus 코드를 작성하여 테스트
* 해당 명령어를 이용하여 실행
  + $ sh ./test.sh
* 결과 AST와 생성된 AST를 비교하여, diff 명령어를 활용하여 차이를 보여줌
* 제공된 TestSet 외에 Dangling Else와 단일 return;을 테스트하는 케이스를 추가함

# Result

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* 파일명 외에 차이가 없는 것을 확인할 수 있음
* Parser가 정상적으로 동작
* Dangling Else Problem과 return statement도 정상적으로 구현되었음을 확인