

Project 3

Due: 10/29/2025 10:30 AM

서론

- (1) 각 프로젝트의 파일은 Project<번호> -> Problem<번호> 폴더 구조로 구성한다. 예를 들어, Project3->Problem1 의 폴더에 Problem1 에서 요구하는 모든 파일을 저장한다. Project 폴더를 압축(zip)하고 압축 파일 이름은 학번으로 하여 제출한다. 명시된 폴더 구조를 따르지 않아 발생하는 불이익에 대한 책임은 본인에게 있음.
- (2) 보고서가 필요한 경우는 pdf 형식으로 제출하며, 언어는 영어/한글 모두 무방하다.
- (3) 프로젝트의 deadline 까지 LMS 의 <과제 및 평가>란에 제출한다.
- (4) 프로젝트의 마감기한을 지키지 못한 경우, 감점 없이 0 점 처리한다.
- (5) 프로젝트는 개별로 수행하며, 다른 사람 또는 자료에서 도움을 받거나 이를 참고로 하였을 경우 관련 소스코드와 보고서에 출처를 명확히 제시하여야 한다. 또한, 다른 학생에게 도움을 주었을 경우에도 소스코드와 보고서에 이에 관련하여 서술한다. 그렇지 않을 경우 모두 0 점 처리한다.

Project 3 은 Project2 에서 작성한 yacc file 과 node.h(node.c 를 header file 형태로 변경함) 를 바탕으로 하여, 각 scope 에 따른 symbol table 생성 및 scope & type analysis 를 수행합니다.

문제 1(50 점/200 점): Symbol Table Construction

아래의 제시문을 읽고, 조건을 만족하는 코드를 symtab.h 및 project3.y에 작성하여 제출합니다. C standard library 를 제외한 library 는 사용할 수 없습니다. (symtab.h 는 node.h 를 include 하도록 되어있습니다. Grammar rule 의 action 에서 정의하는 parse tree NODE 의 name 은 Project2 와 다르게 수정하여도 됩니다.)

- ① Symbol table 의 구조체 SYMTAB 과 entry 구조체 SYMBOL 을 symtab.h 에 정의해두었음. 이를 활용하여 SYMTAB 생성 및 추가, SYMBOL 생성 및 추가, SYMTAB 출력 등 SYMTAB 과 SYMBOL 에 관하여 필요한 함수를 구현할 것(20 점)
 - SYMTAB 생성 및 추가를 위한 함수는 예시용으로 제공하였음.
- ② Project3.sh 을 실행하여 executable file 을 생성하고, mat_mul.c 코드에 대한 symbol table 을 생성한 결과가 mat_mul.jpg 와 동일하게 나타나야 함(30 점)

문제 2(60점/200점): Scope Analysis

문제 2에서는 문제 1에서 생성한 symbol table을 사용하여, 코드 내의 모든 ID들에 대하여 scope analysis를 수행합니다. 제시문에 따라 진행하기 바랍니다.

- ① Scope analysis를 수행하는 함수(ScopeAnalysis)(30점)
 - 모든 ID에 대하여, 해당 ID가 정의되지 않았으면 에러 메시지를 출력.
 - 모든 ID에 대하여, 해당 ID가 정의된 scope이 사용되는 scope을 포함하지 않을 경우 에러 메시지를 출력.
 - 해당 함수는 symtab.h에 정의하고 problem3.y의 action 또는 main 함수에서 호출하도록 작성해도 되고, 또는 함수 형태로 작성하지 않고 grammar rule의 action에 곧바로 작성하여도 됨. 코드 구현은 자유롭게 하여도 무방함.
 - 코드가 구현된 project3.y와 symtab.h를 제출해야 함.
- ② mat_mul.c를 type analysis한 결과(10점)
 - mat_mul.jpg와 동일한 결과가 나타나야 함(에러 메시지 없음)
- ③ mat_mul_err1.c를 type analysis한 결과(10점)
 - mat_mul_err1.jpg와 동일한 결과가 나타나야 함
- ④ mat_mul_err2.c를 type analysis한 결과(10점)
 - mat_mul_err2.jpg와 동일한 결과가 나타나야 함

문제 3(60점/200점): Type Analysis

문제 3에서는 문제 1에서 생성한 symbol table을 사용하여, 코드 내의 모든 expression statements들에 대하여 type analysis를 수행합니다. 제시문에 따라 진행하기 바랍니다.

① Type analysis를 수행하는 함수(TypeAnalysis)(30점)

- 모든 expression/statement에 대하여, 각 연산의 operand의 type이 올바르게 매칭되지 않을 경우 에러 메시지를 출력.
- 해당 함수는 symtab.h에 정의하고 problem3.y의 action 또는 main 함수에서 호출하도록 작성해도 되고, 또는 함수 형태로 작성하지 않고 grammar rule의 action에 곧바로 작성하여도 됨. 코드 구현은 자유롭게 하여도 무방함.
 - 코드가 구현된 project3.y와 symtab.h를 제출해야 함.

② mat_mul.c를 type analysis한 결과(10점)

- mat_mul.jpg와 동일한 결과가 나타나야 함(에러 메시지 없음)

③ mat_mul_err3.c를 type analysis한 결과(10점)

- mat_mul_err3.jpg와 동일한 결과가 나타나야 함

④ mat_mul_err4.c를 type analysis한 결과(10점)

- mat_mul_err4.jpg와 동일한 결과가 나타나야 함

문제 4 (30점/200점)

문제 1~3에서 작성한 코드와 구현방법, 실행결과를 구체적으로 정리하여 report에 작성하기 바랍니다. (project3.pdf)

- ① 코드 구현에 대한 설명(15점)
 - 문제 1(Symbol table construction)(5점)
 - 문제 2(ScopeAnalysis)(5점)
 - 문제 3(TypeAnalysis)(5점)
- ② 실행 결과 이미지와 그에 대한 설명(15점)
 - 문제 1(5점)
 - 문제 2(5점)
 - 문제 3(5점)