

Fundamentals of Compiler Construction Project III

(A Semantic Analyzer)

Description

In this programming assignment, you will implement a semantic analyzer for C- using a C- parser you built in the Project 2. The major requirement of the analyzer, aside from gathering information in the symbol table, is to perform type checking on the use of variables and functions. Since there are no pointers or structures, and the only basic type is integer, the types that need to be treated by the type checker are void, integer, array, and function.

For this, you first need to implement a symbol table utility suitable for the C- language. You are required to implement a symbol table structure that incorporates scope information as separate tables linked together as described in Section 6.3 in Chapter 6 of the textbook.

Based on **syntab.h** and **syntab.c** utility, you need to implement a semantic analyzer, **analyze.c**. When performing type checking, the type checker in your semantic analyzer must use "**name equivalence**" to determine when two type expressions represent same type.

Your documentation must include:

- o Explanation on how to implement the syntab.c and analyze.c conceptually:
- o **Only one** test file consisting of error-free C- program. (test.c) Test file must have 3-level nested scope at least. Also each nested scope needs local variable:
 - ※ Note that your test C- program should be non-trivial and should not be the examples of the textbook.
- o the outputs:
 - For each test program, you need to print out the content of the symbol table whenever a new scope is created:
 - Parsing error message when there are semantic parsing errors:

Your documentation will be graded (20 points) according to their readability and style. Their correctness will be graded in the program part (80 points).

Due Date for the project 3

- o Due Date for Submission : 23:00 on May 28
- o No extension period !!!

Additional guide

1. 전체 semantic analyzer의 수행과정을 2단계로 수행하도록 구현해야 합니다.
첫 번째 단계는 symbol table을 작성하고 scoping error에 대한 에러를 check하는 단계이고, 두 번째는 type check와 그 외 semantic error들에 대해서 check하는 단계입니다.
2. symbol table은 각 scope에 대해서 출력해야 합니다.
3. 오류 체크 사항
 - . 선언되지 않은 변수나 함수는 사용 불가
 - . 변수나 함수, 함수 파라미터 선언 시 중복된 이름 check
 - . 변수나 parameter 선언 시 void type으로 선언할 수 없음.
 - . 변수나 parameter 선언 시 array 변수인지 확인해야 함.
 - . 변수에 값을 assign 하는 경우 type에 대한 check
 - . 변수가 array인 경우 array index가 int가 아닌 경우
 - . array가 아닌 변수를 array처럼 사용하려 하는 경우
 - . 함수 호출 시 parameter의 개수와 type이 일치하는지 check
 - . 함수 호출 시 호출된 것이 함수가 맞는지 (즉 변수 등이 아니었는지) 확인해야
 - . 함수의 return 값과 return type이 일치하는지 check
 - . 함수의 return type이 void일 경우 return이 없어야 함
 - . main함수는 가장 마지막에 선언되어야 함
 - . main함수는 반드시 void로 선언되어야 함
 - . main함수는 반드시 parameter가 존재하지 않는다.
 - . loop문의 반복 조건을 표시하는 부분에 오는 statement는 int값을 가질 수 있어야 한다.

4. 테스트 파일

- . 에러 없는 파일로 1개 제출
- . 테스트 파일은 if, while같이 가능한 모든 keyword를 포함할 것.
- . 최하 3 level 이상의 nested scope를 포함해야 한다. 이 때 각 scope에는 local 변수가 선언되어야 한다.

. ex)

```
int function(int param){
    int var1;
    ...
    while(var1 != 1){
        int var2;
        ...
        if(var2 == 0){
            int var3;
            ...
        }
    }
}
```

5. 배점 (위의 3.오류 체크 사항 포함)

- . symbol table creation - 15 points
- . scoping error check - 15 points
- . type error check - 10 points
- . Function call checks - 10 points
- . return statement checks - 10 points
- . while/if statement checks - 10 points

6. symbol table 모양과 출력형식은 자유입니다. 그러나 아래의 사항들을 반드시 포함시켜주세요. 포함시키지 않을시 감점

(location, 이름, symbol이 나타난 line number들, 변수/파라미터/함수여부, array 여부, array크기, int/void type여부, scope)

제출방법

- o 보고서는 제안서와 결과 보고서 모두 Soft copy 1부, Hard copy 1부로 제출한다.
- o Hard copy는 AS 816호 앞의 지정된 box안에 제출하며 due date는 위와 같다.
- o Soft copy는 2019compiler@gmail.com로 보내며 메일의 보낸 시간 기준으로 due date를 넘기면 안 된다. 또한 결과보고서의 soft copy는 프로그램과 같이 제출한다.
또한 결과보고서의 soft copy 파일명은 메일의 제목과 똑같이 작성한다.
팀 2조의 경우, 결과보고서의 soft copy 파일명은 [Compiler]project_3_2.hwp(doc, docx) 이 된다.
- o 메일의 제목은 [Compiler]project_{project number}_{ team number}로 작성한다.
팀 4조가 결과 보고서를 제출한다면 [Compiler]project_3_4가 된다.
- o 결과 보고서와 프로그램은 모두 디렉토리 안에 포함되어야 하며 디렉토리 채로 tar로 압축해서 위 메일 주소로 제출한다. 디렉토리 안에 세부 디렉토리는 필요치 않는다.
팀 4조라면 압축파일명은 project3_4.tar가 된다.
- o 디렉토리 명은 project3_{team number}로 작성한다.
즉 팀 11조라면 디렉토리 명은 project3_11가 된다.
팀 4조라면 디렉토리 명은 project3_4가 된다.
- o 필요시 readme를 첨부할 것.
- o 메일 제목, 디렉토리 명 등, 위에 기술한 바를 어길시 감점 혹은 아예 채점 대상에서 제외 될 수 있으니 신중하게 할 것.
- o 문서에 code를 copy & paste하지 말 것. 감점의 요인이 될 수 있다.
- o 반드시 makefile을 통해 컴파일시키며, 실행파일명은 디렉토리명으로 한다.
즉 팀3조라면, ./project3_3 test1.c 이 된다.
- o 제안서 및 결과 보고서에 팀원의 역할 분담을 반드시 포함 시킬 것

Design and Development of Compiler for C- Language

Phase 3: Design and Implementation of Semantic Analyzer

과목명: [CSE4120] 기초 컴파일러 구성

담당교수: 서강대학교 컴퓨터공학과 정 성 원

개발자: 1조

홍길동(20141234)

김**(20141235)

개발기간: 2019. 05. xx - 2019. 05. xx

Project3 결과 보고서

/*

본 단계별 결과 보고서 작성의 목적은 자신들의 연구 개발 내용을 정확히 기술하고 그에 대한 자체 평가를 통하여 해결책을 제시하는 과정을 수행해보는 것임.

*/

프로젝트 제목: Design and Development of Compiler for C- Language:

Phase 3: Design and Implementation of Semantic Analyzer

제출일: 2019. . .

참여조원: 홍길동(20171234), 김**(20141235)

I. 개발 목표

- 내용이 변경되지 않았을 경우 “단계별 제안서 I. 개발 목표 참조”와 같이 기술할 것.
- 개발 목표가 중간보고서 작성 당시의 내용과 다를 경우 그 내용 및 변경 사유를 상세히 기술할 것. 가급적 개발 목표는 변경하지 말 것.

II. 개발 범위 및 내용

- 내용이 변경되지 않았을 경우 “단계별 제안서 II. 개발 범위 및 내용 참조”와 같이 기술할 것.
- 개발 범위 및 내용이 단계별 제안서 작성 당시의 내용과 다를 경우 그 내용 및 변경 사유를 상세히 기술할 것.

가. 개발 범위

나. 개발 내용

III. 추진 일정 및 개발 방법

- 내용이 변경되지 않았을 경우 “단계별 제안서 III. 추진 일정 및 개발 방법 참조”와 같이 기술할 것.
- 개발 추진 일정, 개발 방법, 역할 분담 등이 중간보고서 작성 당시의 내용과 다를 경우 그 내용 및 변경 사유를 상세히 기술할 것.

가. 추진 일정

나. 개발 방법

다. 연구원 역할 분담

IV. 연구 결과

- 각 단계별 연구 개발 결과를 자유롭게 기술할 것.
- 기초 컴파일러 구성: 다음과 같은 내용을 포함해야함. 그 외의 내용은 자유롭게 기술할 것.
 1. **합성 내용:** 설계 목표 달성에 필요한 요소 기술에 대하여 조사 분석하여 그들을 바탕으로 구성된 **전체 소프트웨어 구성도**, **각 부분의 역할 및 사용 기술 및 툴**(예: 사용한 이론적 방법, 공개 소프트웨어나 툴 또는 API 등), 구현 방법 등 관련 내용을 기술할 것. 또한 개발 소프트웨어의 사용 방법, 즉 사용자 인터페이스 규약을 명시하고 간단한 **사용자 매뉴얼**을 작성할 것.
 2. **분석 내용:** **알고리즘 구성 요소**에 관한 자세한 분석 내용, 어떠한 방식으로 개발하였는지 그 내용을 기술할 것.
 3. **제작 내용:** 개발 결과. (**개발한 함수, 변수에 대한 자세한 내용 작성할 것**)
 4. **시험 내용:** 자신이 개발한 소프트웨어를 자체 평가하기 위한 기준 및 수행한 평가 방법, 그리고 그 것에 기반을 둔 평가 결과를 상세히 기술하라. (반드시 평가 점수가 높은 것이 좋은 것이 아니라 자신이 설정한 방법이 문제점을 찾는데 도움이 되었다는 것을 보이는 것이 더 중요함.) **캡처 포함**
 5. **평가 내용:** 상기 내용을 바탕으로 자신이 개발한 소프트웨어의 장단점을 정확히 분석하고 그 내용을 기술할 것. 특히 개발 도중 예상치 않았던 문제가 발생하였다면 상세히 기술하고, 추후 어떠한 방법으로 해결할 수 있을지에 대하여 해결책을 기술할 것. (이 부분에서의 창의성이 중요하며, 반드시 자체 평가 내용이 좋을 필요는 없고, 스스로 문제점을 찾아 향후 해결책을 제시하는 것이 중요함.) 또한 공학 외적인 부분을 포함하여 경제 요건, 안정성, 신뢰성, 윤리성, 사회적 영향, 실행 가능성 등에 측면에 대해서도 연구 결과와 관련지어 논하라. 본 설계 프로젝트의 경우 **안정성과 신뢰성**에 대하여 반드시 기술할 것.

V. 기타

- 기타 관련 내용을 기술할 것.
- 다음과 같은 내용을 포함해야함. 그 외의 내용은 자유롭게 기술할 것.
 1. **연구 조원 기여도:** 토의 및 합의를 통하여 각 조원의 상대적인 기여도를 기술할 것. 이때

각 조원의 기여도 합이 100이 되어야 함.

2. **자체 평가**: 본 프로젝트의 수행에 있어 왜 자신들이 다른 팀보다 더 좋은 점수를 받아야 하는지를 기술하라. 위에서의 (5. **평가 내용**)은 개발한 소프트웨어에 대한 자체 기준을 통한 평가이고, **여기서의 자체 평가**는 프로젝트 수행 전반에 걸쳐 자신들이 얼마나 창의적으로 목표 및 방법을 설정하고, 수행 중 발생한 문제를 얼마나 현명하게 해결 했는지, 얼마나 독창적인 아이디어가 들어가 있는지, 자신들의 결과가 얼마나 우수한 지, **팀원 간의 협력이 얼마나 조직적으로 진행(팀 간의 회의를 바탕으로 진행이 되었다면, 이에 대한 문서를 작성해 결과보고서 제출 시에 별첨으로 첨부할 것, ex_ 회의록, 메신저 대화 내용)**이 되었는지 등의 프로젝트 수행 자체에 대한 평가임. 간략한 보고서 작성 및 발표를 통한 표현 능력도 중요한 평가 척도임.

3. 기타 본 설계 프로젝트를 수행하면서 **느낀 점**을 요약하여 기술하라. 그 내용은 어떤 것이든 상관없으며, 본 프로젝트에 대한 문제점 제시 및 제안을 포함하여 자유롭게 기술할 것.