

Fundamentals of Compiler Construction Project 2

(Design and Implementation of LALR Parser)

Description

In this programming assignment, you will implement a C- parser (i.e., `cm.y`, similar to `tiny.y`) using Bison and the lexical analyzer (`lex.yy.c`) in your Project 1. For this, you are also required to extend the following 4 supporting files being used in Tiny Compiler.

- o `global.h`: Global types and vars for C- compiler
- o `scan.h`: The scanner interface for the C-compiler
- o `util.h`: Utility function interface for the C- compiler
- o `util.c`: Utility function implementation for the C- compiler

Your documentation should list each shift-reduce conflict (if any) and explain how it is resolved. The output of your program is the abstract syntax tree, which you need to print out using `printTree` given in "util.c" to the output file. When there are syntactic parsing errors in C- programs, you must print out relevant syntactic parsing error messages with the corresponding line numbers. Note that you may need to extend `printTree` procedure for this project before using it. The `printTree` procedure use indentation to indicate subtrees. Your documentation must include:

- o Explanation on how to run your program:
- o Precedence Directives with explanations:
- o List of all conflicts with the explanations of how each conflict are resolved.
 - `y.output` file should be included here when you explain the conflicts
- o Explanation on the implementation including semantic action part:
- o Source Codes and the output
- o two test C- programs: one parsing error-free C- program and one C- program with parsing errors. (`errorfree.c`, `error.c`)
 - ※ Note that your 2 test C- programs should be non-trivial and should not be those in the textbook.

Due Date for the project 2

- o Due Date for Result Report: 5/3(by 23:00)
- o No extension period !!!

제출방법

- o 보고서는 제안서와 결과 보고서 모두 Soft copy 1부, Hard copy 1부로 제출한다.
- o Hard copy는 수업 시간 전이나, AS816호 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. (디렉토리 안에 결과보고서를 넣어 함께 tar로 압축)
또한 결과보고서의 soft copy 파일명은 메일의 제목과 똑같이 작성한다.
결과보고서의 soft copy 파일명은 [Compiler]project_2_result_20101621.hwp(doc, docx, pdf) 이 된다.
- o 메일의 제목은 [Compiler]project_2_{student number}로 작성한다.
학번이 20101621 결과 보고서를 제출한다면 메일 제목은 [Compiler]project_2_20101621이 된다.
- o 결과 보고서와 프로그램은 모두 디렉토리 안에 포함되어야 하며 디렉토리 채로 tar로 압축해서 위 메일 주소로 제출한다. 디렉토리 안에 세부 디렉토리는 필요치 않는다.
- o 반드시 makefile을 통해 컴파일시키며, 실행파일명은 학번으로 한다.
리눅스(cspro.sogang.ac.kr)상에서 실행한다면, ./20101621 test1.c 이 된다.
- o 디렉토리 명은 project2_{student number}로 작성한다.
즉 20101621이라면 디렉토리명은 project2_20101621이 된다.
- o 파일 압축명은 project2_{student number}.tar로 압축한다.
즉 20101621이라면 파일 압축은 project2_20101621.tar가 된다.
- o 필요시 readme를 첨부할 것.
즉 readme.txt로 디렉토리 안에 존재하면 된다. 필요하지 않다면 생략 가능.
- 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.

Design and Development of Compiler for C- Language

(Compiler 개발은 한 학기에 대한 것이므로, 현 문서의 범위에
대해 기술하도록 한다.)

1. 각 단계별 결과 보고서 (현 문서의 명확한 목적을 기입한다.)

과목명: [CSE4120] 기초 컴파일러 구성

담당교수: 서강대학교 컴퓨터공학과 정 성 원

개발자: 20101621 김용담

개발기간: 2019. 04. xx - 2019. 05. xx

각 단계별(현 개발 단계) 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 2: Design and Implementation of LALR Parser

제출일: 2019. . . .

개발자: 김용담

I. 개발 목표

- 지난 강의 내용(문제 제시)을 바탕으로 팀원 간의 토의를 통하여 본 설계 프로젝트에서 추구하는 개발 목표를 설정하고 그 내용을 기술할 것.

II. 개발 범위 및 내용

- 자신들이 설계한 개발 목표를 달성하기 위하여 어떠한 내용의 개발을 수행할 지 그 범위와 예상되는 개발 내용을 기술할 것.

가. 개발 범위

나. 개발 내용

III. 추진 일정 및 개발 방법

- 자신이 설정한 연구 개발 목표를 달성하기 위한 개발 일정을 설정하고, 각 요소 문제를 해결하기 위하여 어떤 방법을 사용할 지 기술할 것.

가. 추진 일정

나. 개발 방법

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 매뉴얼을 작성할 것.

2. **분석 내용:** 각 구성 요소를 상세히 분석하여 어떠한 방식으로 개발하였는지 그 내용을 기술할 것.

3. **제작 내용:** 개발 결과.

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.)

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.

- 기초 컴파일러 구성: 다음과 같은 내용을 포함해야함. 그 외의 내용은 자유롭게 기술할 것.

1. **자체 평가:** 본 프로젝트의 수행에 있어 왜 자신이 다른 사람보다 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 자신이 얼마나 창의적으로 목표 및 방법을 설정하고, 수행 중 발생한 문제를 얼마나 현명하게 해결 했는지, 얼마나 독창적인 아이디어가 들어가 있는지, 자신의 결과가 얼마나 우수한 지에 대한 평가임.

2. 기타 본 설계 프로젝트를 수행하면서 느낀 점을 요약하여 기술하라. 그 내용은 어떤 것이든 상관이 없으며, 본 프로젝트에 대한 문제점 제시 및 제안을 포함하여 자유롭게 기술할 것.