

CS 4310 Programming Exercise

Building a Type Checker and Translator to C

Due Date: Midnight, Wednesday Nov 29

The goal of this project is to build a program to detect and report semantic errors (typically type errors) in pseudo-Pascal (the language used for all projects in this class and hereby referred to as Pascal) programs and translate correct programs to C. The project basically has two components: type checking and translation to C. Your assignment is to develop a program *p2c* that takes a syntactically correct pseudo-Pascal program as input and generates 2 output files as follows:

1. A file containing the source Pascal program with semantic error messages interspersed in it.
2. A file containing the translated C program. The contents of this file are irrelevant if the program contains errors.

For this project you may assume the following:

- The input program does not contain any syntax errors.
- *The input program does not contain any procedures/function declarations or calls.* (Note that the parser given to you does contain syntax for procedures and functions, which you may ignore) You may also assume that the input program does not contain any *exclusive OR* operator, since there is no C equivalent for it.
- Note that a symbol table is essential for translation. It is possible to generate a correct declaration section for the output C program by simply scanning the symbol table and printing the appropriate declarations.
- You will need to replace pascal BEGIN...END sequences with curly braces in C. Don't worry if you have to put unnecessary curly braces in some scenarios since this does not change the C program semantics.
- You will need to translate Pascal `writeln` to the appropriate C `printf` calls, and add the necessary C include files.
- As a rule of thumb, it is generally easier to do the printing of the C program at a higher level in parsing (e.g at statement rules rather than at expression level) when there is a choice, since this gives more context.

Getting Started

All the material for this project is available via the Class webpage.
You can create the parser by typing `make`.

All questions regarding the project should be posted in our news group `uh.nsm.cosc4310`. Please keep checking the class webpage and newsgroup for any changes or clarifications. You may reuse your grammar from project 2 or use the grammar supplied. The files provided include a working parser and a

primitive symbol table. The symbol table code is provided just to help you get going and would have to be completely rewritten.

When you submit the project, please include the path to your *RCS* directory on Bayou in your README file. Please turn on read permission on this directory one week after submission deadline.

Submission Instructions

This project is due at midnight on Wednesday, Nov 29. Submission procedure will be posted on the web page.

Good luck!