Functional Design Project Proposal

James Coleman Gibson

October 4, 2017

Background

Being a teaching assistant for CSSE304, Programming Language Concepts, is a very large time commitment. The assistants for the class are asked to spend time in the CS labs to assist students with concepts, debugging, are asked to give feedback on assignments, and to grade exams. Examining students' coding submissions is particularly time consuming, as the assistants must interpret intentions and attempt to distinguish between small mistakes and fundamental understanding. For assignments with efficiency requirements the assistants must also determine control flow to determine whether things like `append` will break linearity requirements in their specific position. In the beginning of the course, we must give feedback to simplify expressions with things like η conversions and for shortening code with existing built in procedures. In short, a huge amount of the assistants' time is spent working as a human code analysis tool.

Proposal

To simplify the lives of both the teaching assistants and the students in the course, I propose the construction of a static analysis tool to be used by graders and students alike. The feature list below is an approximate list of features for such a tool.

Features

Code Suggestions

A huge number of students in CSSE304 struggle with code simplification suggestions, including reductions from `(append (list x) ...)` to `(cons x ...)` and basic η-reductions. Implementing an analysis tool to allow both these suggestions and others to be implemented would prevent students from making these mistakes in the first place, leaving them more time to learn more important topics and leaving the assistants more time to give more meaningful feedback.

Procedure Arity Checking

One of the most frustrating errors to get in the interpreter project and the individual assignments is the infamous `Incorrect number of arguments to #<procedure>` error. This is difficult to debug for a variety of reasons. The first and most basic is that the error does not include a line number of where the procedure was called. This on its own is a rather simple issue which would be easy to fix by using an interpreter which tracks source metadata. This could be improved even further by including the source metadata in the procedure itself, allowing you to also see where the offending procedure was defined. Even better would be if you didn't have to run the code at all to detect these errors. Using a $k - CFA$ implementation to find arity violations would save students from needing to sort through pages of anonymous procedure traces and would be a stepping stone into other useful analyses.

HTTP Interface

For this project I will be using Haskell, which is not a commonly used language by either students or professors at Rose-Hulman. As such, installation becomes a tricky issue. Allowing for expression based or program based analysis through an HTTP interface would make this much more simple, as only one server in the school would actually need to be running Haskell code.

An Interpreter

Although Chez Scheme does feature a number of debugging tools, many of them can be difficult to use for beginner Scheme programmers. This would not such a significant issue, except that Chez Scheme's error messages are uninformative as to the locations of errors. An interpreter built specifically to be used for debugging would allow students to spend more time learning the language, and less time learning to use `trace`.

Shape Analysis

Several of the more difficult assignments to give feedback on in CSSE304 are the programs with specific requirements on time and space complexity. Detecting this automatically would save many hours of time for assistants in the course and would help students to learn why their code does not meet these requirements before they actually lose points. A potential way to solve this is through program shape analysis, to detect properties like whether there are multiple visits to a node in a tree.

Advanced Testing

A last possibility is to implement a more advanced testing system for the course. This could be done with something like test generation or through property based testing. Although this is dependent on the interpreter feature for the project, this has a great potential to help prevent incorrect solutions from slipping by the grading server.