Samuel Breese, breess@rpi.edu

Education

Rensselaer Polytechnic Institute:
Bachelor of Science, Computer Science, June 2018
Autodidacticism

Knowledge: Programming Languages

Mastery of:

Haskell 98, C, Forth, R5RS Scheme

Extensive experience with:

Idris, Python, C++, Perl 5, Common Lisp, Clojure, x86 Intel-syntax assembly, Zilog Z80 machine code, POSIX-compatible shell, LLVM IR

Significant experience with:

Java, JavaScript, Go, Scala, Erlang, PHP, Standard ML/OCaml, Emacs Lisp

Some experience with:

Coq, Agda, Elm, R, Factor, K, Elixir, Shen, Frege, Julia, Rust, Prolog, Octave, Ruby, Perl 6, non-POSIX shells (e.g. Plan 9's `rc` shell)

Knowledge: Technologies

Very familiar with Unix system administration, from unconventional kernel builds and init systems to window managers and common applications. Particularly familiar with optimizing kernel and system space and time usage on Gentoo GNU/Linux. General familiarity with the software ecosystem of Linux-based (and to a lesser extent BSD-based and Plan 9-based) systems overall.

Working experience with Linux kernel development, particularly within TCP/IP.

Very familiar with common parsing tools from parser generators like Lex and Yacc to parser combinator systems as seen in functional languages.

Generally familiar with the library and tooling ecosystem for many programming languages, especially Haskell, Clojure, Common Lisp, and Scheme.

Knowledgeable about low-level graphics (i.e. framebuffer/DRM) on GNU/Linux systems.

Familiar with digital audio encoding and synthesis, particularly through procedural generation.

Versed in a variety of version control systems, such as Git, Mercurial, Subversion, and Bazaar, as well as other common software development tools (e.g. continuous integration, testing frameworks).

Knowledge: Other

Familiar with category theory, comfortable with common category-theoretic abstractions as applied to statically typed functional programming.

Familiar with concepts in the formal specification of programming languages.

Knowledgeable of the type-theoretic foundations of statically typed functional programming in Martin-Löf's intuitionistic type theory.

Experience

Since 2015 I have worked in a research and development context on Submitty, Rensselaer Polytechnic Institute's in-house open source homework submission and automatic grading system. In this position I developed a framework of tools for performing static analysis upon student source code using Haskell, including a sophisticated plagiarism detection tool. I also developed significant development and deployment infrastructure around this tool and others, including a unit testing framework, continuous integration, and automatic deployment. This work was done on a part-time basis during the school year and full-time for two summers.

In Spring 2016 I began working as an undergraduate TA for Rensselaer Polytechnic Institute's Data Structures course (CSCI 1200), and continued this work every subsequent semester. As an undergraduate TA I worked in a grading role within weekly lab sections in addition to a teaching role during weekly office hours.

Projects

Over the past decade, I have contributed to the open source community through existing projects as well as novel projects. These projects range in scope from the Python and Elixir standard libraries to small collections of utilities for personal use. A subset of more recent contributions can be found on GitHub at `https://github.com/chameco`. Some examples of these projects are given here:

In 2012 I developed a Markdown parser for Clojure as an example of best-practice usage of a popular parser-generator library. This project developed considerable community momentum, inspiring several forks for different markup languges in addition to being featured in Read-Eval-Print-λove, a bi-monthly newsletter for the Lisp family of languages.

In 2013 I wrote and released a bytecode-interpreted scripting language under the MIT license. This language supports modern functionality like lexical closure, higher-order programming, a prototype object system similar to JavaScript, and compilation to a novel bytecode format for a custom virtual machine in order to facilitate some basic optimizations. This project received immensely positive reception, remaining the top trending C language project on GitHub for weeks, and was used by a third party as a scripting language on embedded platforms.

Conference Presentations

Using Static Analysis for Automated Assignment Grading in Introductory Programming Classes
Samuel Breese, Ana Milanova, and Barbara Cutler
Poster at ACM Technical Symposium on Computing Science Education (SIGCSE), March 2017

Submitty: An Open Source, Highly-Configurable Platform for Grading of Programming Assignments
Matthew Peveler, Jeremey Tyler, Samuel Breese, Barbara Cutler, and Ana Milanova
Demo Presentation at ACM Technical Symposium on Computing Science Education (SIGCSE), March 2017

User Experience and Feedback on the RPI Homework Submission Server
Andrea Wong, Eric Tran, Joe Jung, Ben Shaw, Marina Espinoza, Beverly Sihsobhon, Melissa Lindquist, Samuel Breese, Matthew Peveler, and Barbara Cutler
Poster at ACM Technical Symposium on Computing Science Education (SIGCSE), March 2016