Buck Shlegeris

September 9, 2015

Fan of elegant abstractions and functional programming   bshlegeris@gmail.com

bshlgrs.github.io

Employment

- **PayPal**   San Francisco
  January 2015-current
  - Writing distributed services in mostly functional Scala, with Akka, Spray, etc

- **App Academy**   San Francisco
  January 2014-July 2014
  - Taught Ruby on Rails, Ruby, Javascript, Backbone
  - Delivered lectures, wrote learning materials

Personal projects and presentations

- **Ruining the Coding Interview**   2015
  github.com/bshlgrs/ruining-the-coding-interview   ***Presented at Scala By The Bay 2015***
  - A project where I try to automate various types of code optimization which are typically done manually, such as data structure selection.
  - I presented a Scala program which takes (a subset of) Java source and outputs Ruby code with optimized data structures. This project also involved writing a computer algebra system.
  - Still largely a work in progress.

- **rPeANUt compiler**   mostly 2014
  github.com/bshlgrs/rpeanut-compiler
  - Compiler from a subset of C including pointer arithmetic to a RISC instruction set
  - Currently partially deployed to the web with ScalaJS.

Education

- **Australian National University**   Canberra, Australia
  BSc   *2012-2014*
  - Undergraduate coursework: Algorithms, operating systems, AI, algorithmic information theory and universal AI, theory of programming languages, computer architecture, linear algebra and ODEs, theory of computation
  - Director and presenter at CompCon, an inaugural Australian undergraduate CS conference; presented on algebraic behaviour of data structures
  - Completed two research projects and a variety of advanced undergraduate courses ahead of my year level.
  - One of those research projects was on using denoising autoencoders and other neural nets for q-learning in reinforcement learning, using Theano.

Skills

- I normally code in Scala, Ruby, Javascript, Python, Haskell.

- Web development, particularly in Rails and React

- Data structures, programming language theory/type theory, machine learning (especially neural nets)