Tikhon Jelvis

http://jelv.is
http://github.com/TikhonJelvis
tikhon@jelv.is
(650) 283-4713

Education

University of California, Berkeley: BS EECS — in progress

Skills

Languages Haskell, OCaml, Scheme (Racket), JavaScript, Python, Gosu, Java, C, Prolog

Topics Programming languages, program synthesis, functional programming, FRP

Tech Linux, Emacs, Git, SMT solvers, LaTeX

Achievements

StackOverflow I'm in the top 10 for the Haskell tag and in the 99^{th} percentile overall — 2010–Now

Math contests AMC12: 100.5 (97.5^{th} percentile), AIME: 3 — 2010

Hackathon Got 2^{nd} place at the Berkeley CSUA Hackathon — 2011

Publications

PLDI 2014 — Phitchaya Mangpo Phothilimthana, **Tikhon Jelvis**, Rohin Shah, Nishant Totla, Sarah Chasins, Rastislav Bodík, "Chlorophyll: Synthesis-Aided Compiler for Low-Power Spatial Architectures", *Proceedings of the 35th Annual Conference on Programming Language Design and Implementation (PLDI), ACM Press, 2014*

Experience

Jane Street Capital Jun 2013–Aug 2013

Worked on a few different projects as well as learning about market making, arbitrage, ETFs and finance in general:

- A time-series graphing tool by compiling OCaml to JavaScript with `js_of_ocaml`
  - Wrote some reusable OCaml web libraries including a typesafe CSS system and a Functional Reactive Programming (FRP) framework
- A replacement for the automount daemon in OCaml
- A mock trading bot, working with two trading interns. The bot used OCaml, but I also used Haskell for exploratory code, data analysis and testing.

Berkeley ParLab Aug 2012–Current

Worked on program synthesis—generating programs to some specification

- Targeted GreenArrays, a very constrained, energy-efficient, stack-based architecture
- Implemented two synthesizers, one in Racket and one in Haskell
- Used the Microsoft Z3 SMT solver, Berkeley's Sketch solver and some machine learning algorithms

MixRank
Dec 2012–Feb 2013

Worked part-time at MixRank, a YC startup in San Francisco

– Developed an aggregation framework for processing streaming web crawler data
– Used Python and PostgreSQL

Ashima Arts
May 2012–Aug 2012

Worked on `gloc`, a utility for creating and linking modules for GLSL written in OCaml

– Embedded `gloc` in a browser extension, using `js_of_ocaml`
– Wrote a small open-source library for semantic version numbers: `ocaml-semver`

Copilot Labs
Sep 2011–May 2012

Worked part-time at Copilot, a San Francisco startup targeting restaurants

Did a bit of everything:

– Backend work with Python and Django, including some simple machine learning
– Frontend work with JavaScript, jQuery and Raphael.js
– Ops, setting up automated testing and Jenkins CI

Guidewire
May 2011–Jul 2011

Guidewire is an enterprise software company serving P&C insurance companies

Worked on the internal tools team developing a tool to monitor the company's servers

– Wrote the backend and monitoring daemon in Gosu, Guidewire's own JVM language
– Built the frontend with JavaScript, jQuery and jQuery UI

Personal Projects

Semantic Diff
`jelv.is/cow`

Working on a semantic diff and merge tool in Haskell codenamed `cow` that

– Compares programs by diffing and analyzing the parse trees
– Finds matching subtrees to robustly detect moved blocks of code
  ◦ can detect blocks of code that are both modified and moved *simultaneously*
– Does scope analysis to find renamed variables
– Can do three-way merges with more extensive, semantically aware conflict resolution

TPL
`jelv.is/tpl`

Designed and implemented (in Haskell) a dynamically typed scripting language, TPL

– Influenced by Self, JavaScript, Lua, Scheme and Haskell
– Has some interesting features:
  ◦ "Laziness on demand"—fine control over what gets evaluated when
  ◦ A unified system for variable scope and object-oriented programming
– Some silly features as well, like being able to change operator precedence at runtime

More

I have more projects at my website <`http://jelv.is/projects.html`> and on my GitHub account <`http://github.com/TikhonJelvis`>. I also have a few packages up on Hackage: `modular-arithmetic`, `array-forth` and `mcmc-synthesis`.