Jakob Hain

jakobeha.github.io | github.com/jakobeha jakobeha@fastmail.com | Winthrop, MA 02152

Area of Focus: Programming languages + software engineering

My goal is to create tools to help other people create content

Education

Northeastern University - *Khoury College of Computer Sciences* Sept 2017 - Dec 2019

Degree: Bachelor of Science in Computer Science
GPA: 3.887 / 4. Member of the Honors College
Notable Classes: CS4910 Verified Compilers, CS4620 Building Extensible Systems, CS4500 Software Development, CS4410 Compilers

Purdue University - *Computer Science Graduate Program* Sept 2020 - Present

Seeking: PhD in Computer Science
GPA: 2.98 / 4
Notable Classes: CS565 Programming Languages, CS503 Operating Systems, CS505 Machine Learning, CS592IML Interpretability of ML

Publications

R Melts Brains - ***DLS 2019*** Oct 2019

Olivier Flückiger, Guido Chari, Jan Ječmen, Ming-Ho Yee, Jakob Hain, Jan Vitek

Contextual Dispatch for Function Specialization - ***OOPSLA 20*** Oct 2020

Olivier Flückiger, Guido Chari, Ming-Ho Yee, Jan Ječmen, Jakob Hain, Jan Vitek

Research Experience

Ř - *Northeastern PRL* Sept 2017 - May 2020

- R̂ is a JIT compiler for R which uses static analysis and speculation to elide unused reflective data and capabilities such as string variable names, improving performance
- R̂ uses liveness analysis, taint analysis, scope analysis, SSA form, loop peeling, LICM, constant folding, type inference, deopt speculation, profiling, etc.
- Mainly worked on type inference fixes, serialization, and Software Transactional Memory to "safely" reduce lazy expressions when they don't produce side effects

UnderstandableBinary - *Purdue* Sept 2022 - Present

- ML model to disassemble and understand C object code
- Fetches and compiles C packages from APT, then uses them to train the model

Teaching Experience

CS2500 Fundamentals I - *Northeastern CCIS* Sept 2018 - Dec 2018

- Northeastern's mandatory introductory course, teaches foundations of programming (e.g. recursion) and good practices (documentation, testing) in a variant of Scheme

CS307 Software Engineering - *Purdue CS* Sept 2021 - Present

- Course which teaches Industry concepts and ethics, teams create their own software project (e.g. website), and submit design documents, and follow SCRUM
- As project coordinator I help teams specify their projects and review their documents
- As head TA I also manage logistics and answer questions from other coordinators

Work Experience

Freelance Developer - *RemoG, Remote* Jan 2018

- Built an iOS app to show sensor data (e.g. speed, temperature, pressure) for a car

Developer - *NextDroid (self-driving ground-truth analysis via LIDAR)* Dec 2019 - Jan 2020

- Fix website bugs and create camera view for analysis (frontend)
- Fix camera C++ driver and server (backend)

Community Service

Instructor - *Cervizzi's Martial Arts Academy, Winthrop MA* Apr 2015, Oct 2016

Counselor - *Parks and Recreation, Winthrop MA* June 2016 - Aug 2016

Counselor - *GER^{2}I, West Lafayette IN* July 2022

Tools

High performance:	Rust, C++, C, gdb/lldb/rr, Linux internals
Web development:	TypeScript, HTML, CSS, JavaScript, React, esbuild, Flask
Formal methods:	Coq (Isabelle), Dafny, Haskell
General purpose:	Kotlin, Swift/iOS, Java, Python, Lua, Docker, Bash, Git

Personal Projects

TreeScript - *DSL to transform syntax* Dec 2018

- Similar to "Parser Parser Combinators" (van Tonder, Le Goues)
- Works via transformers which match and replace syntax, e.g. "foo(\x) -> bar(\x)"
- Finalist at Northeastern's RISE 2019

Descript - *Simple language which transforms its own code (2 versions)* Feb 2018

- To perform refactors like renaming symbols & adding fields to structures, run one Descript file on another file, and it will transform it in place (both versions)
- IDE (VS Code) extension which highlights errors and renames symbols (old version)

cge-ai - *ML AI for turn-based games* Jan 2022 - May 2022

- Based on AlphaZero, modified to support more flexible games (e.g. more players)

devolve (WIP) - *Node-based DSL and experimental low-code UI* June 2021 - Present

- node-based allows for high-level view and rapid iteration
- seamless integration with traditional languages (define devolve nodes from language functions and vice versa). devolve compiler/IR-interpreter written in Rust
- devolve-UI = high-performance, low-code reactive UI by composing devolve scripts
- experimental "prompt-based" scripts in Python for asynchronous (e.g. sequential) UI

Hobbies: Running, Weightlifting, Graphic Design, Electronic Music Production