

Sarah Rovner-Frydman

✉ sarahzrf@protonmail.com
🌐 github.com/sarahzrf

Formal Education

2016–2020 **Marlboro College**, Marlboro, Vermont
Bachelor's degree in computer science & mathematics.

Experience

Tech-Related Employment

- 2021 **Research Intern**, *Awake Security*, (telecommuted)
Did research and prototyping work toward improving type safety in a product-embedded scripting language.
- 2016 **Google Summer of Code**, *Celluloid*, (telecommuted)
Restructured, cleaned up, and generally overhauled “ECell”, a Ruby library, going from nearly 5,000 lines of code to about 4,000. Also wrote some documentation for it.

Community

- 2020 **UCR ACT**
Gave a talk for the UC Riverside Applied Category Theory seminar (“[Separation logic through a new lens](#)”).
- 2014–2016 **Boston Haskell**
Worked on projects at Hac Boston (3-day Haskell hackathon) in 2014, 2015, and 2016. Participated in monthly meetups.

Projects

- 2019–2020 **Formal verification of an evaluator for a classical linear logic calculus**
For my bachelor's degree: Wrote an evaluator in C for Abramsky's linear calculus PE_2 , then attempted to use the Verified Software Toolchain to verify the correctness in Coq of a fragment of this evaluator with respect to the operational semantics of the language as defined by Abramsky.
- 2018–2019 **GUI for building sequent calculus derivations**
A GUI for bottom-up construction of cut-free sequent proofs, leveraging the properties of sequent calculus for an interaction model that requires [almost] no typing or selection of rules from menus. Supports the propositional fragments of systems LK, LJ, two-sided classical linear logic, and one-sided classical linear logic. On GitHub at <https://github.com/sarahzrf/sequents>.
- 2017–2018 **Compiler for a small lazy language**
A compiler for a small lazy language—close to being a tiny subset of Haskell—targeting LLVM IR, loosely inspired by the STG machine. Includes a basic copying garbage collector to link programs against. On GitHub at <https://github.com/sarahzrf/STG>.

2016 **Type theory-backed GUI for point-and-click algebraic reasoning**

A prototype of a GUI for performing algebraic manipulations with an interaction model based upon the ability to select subterms by clicking or using keyboard navigation. Under the hood, manipulations generate proof terms in a small type theory to ensure correctness. On GitHub at <https://github.com/sarahzrf/cas>.

Programming Languages

Fluent

- Haskell
- Python
- JavaScript
- Coq

Comfortable

- Ruby
- C
- PureScript

Usable

- Java
- bash
- Agda
- OCaml
- Rust