Eliza Brandt

jackbrandt11@gmail.com | 630-998-4473 | StackOverflow: Eliza Brandt | Github: tavrinky

Education

North Central College Naperville, IL

Bachelor's of Science in Mathematics and Shimer School of Great Books: Social Sciences; *Fall 2017 - Winter 2019*

Relevant Coursework: Abstract Algebra, Linear Algebra, Computer System Concepts, Fundamentals of Classical Logic and Mathematics, Fundamentals of Modern Logic and Mathematics, The Nature of Proof in Mathematics

Skills

- **Skills**: Haskell, Python, Javascript, Ruby, Java, Git, Postgres, LaTeX, Azure, Shell (Bash, Fish)

Experience

Conduent Inc. Morrisville, NC

Software Engineer (Blockchain Platform) *2019 - Present*

- **Katip Logging**: Introduced and developed utilities for a logging framework
  - Evaluated different logging libraries across the Haskell ecosystem for conformance to Haskell and industry best practices
  - Developed extensions to the Katip library for ease of use among beginner Haskell programmers
  - Integrated industry best practices (structured logging, varying severity and verbosity levels of logs, contextual logging) into the blockchain team
  - Developed a tutorial explaining the use and limitations of the logging framework
- **Vendor Data Management**: Yesod backend for a web app managing vendors and invoice payments, connected to Hyperledger Fabric
- Advised blockchain team on the best methods for implementing errors and exceptions into Haskell
- Developed and wrote a Haskell style guide for internal use
- Guided beginner Haskell programmers on different topics, from an introduction to functors to monad transformers and stacks to the three-layer cake model
- Developed resources on domain driven design in Haskell for internal use

Projects

- **Flashcards**: A Yesod website for studying languages
  - Quizzed the vocabulary words you were most likely to error on
  - Based off Anki, designed for better ergonomics

- **T and L**: Add-on to Haskell's type system
  - Added a type family T, which produced a type with as many inhabitants as its argument specified
  - Added a type family L, which computed the number of inhabitants of a type (assuming the type had only finitely many inhabitants)
  - Used to simplify isomorphisms, e.g. $a \to b =$ Vec (L a) b
  - Developed after Haskell's dependent typing facilities (used for indexing, particularly indexed comonads as an extension of McBride's indexed monads in "Monads of Outrageous Fortune") proved insufficient
  - Also used for working around issues of variable binding with Rank2Types
  - When implemented, would permit coercibility of nontrivial (i.e. not just newtype wrapped but otherwise equivalent types) types for reduction of boilerplate and increased optimization potential
  - Equivalence between types is decidable, and easy enough for a human to read (due to polynomial representation)

- **G**: Exploration of type equivalencies permitted by T and L, as a generalization of common data structures
  - Generalization developed after noticing similarity between indexed cofree comonads and k-ary trees
  - Cofree comonads are easily used for representing a variety of data structures depending on choice of functor, and are easily adapted to algebraic representation, and so were an excellent motivating example for the use of T and L

- **PDominion**: API for simulating the Dominion card games, with the ruleset forthe base game, all the base cards, and several basic player strategies already developed, written in Python
- **HDiff**: Symbolic differentiator written in Haskell, using an AST over mathematical expressions, looking to generalize over more broadly representative mathematical objects,ratherthan merely HaskellâĂŹs Num and Floating typeclasses

In progress papers on T and L and G available on request.