

# Larry Diehl

FORMAL METHODS RESEARCHER & ENGINEER

144 Hawkeye Ct. Apt. 215 – Iowa City, IA – 52246

☎ 407-718-7665 | ✉ [larrytheliquid@gmail.com](mailto:larrytheliquid@gmail.com) | 🏠 [www.larrytheliquid.com](http://www.larrytheliquid.com) | 💻 [larrytheliquid](#) | 🐦 [@larrytheliquid](#) | 🌐 [larrytheliquid](#)

## Education

### Portland State University

PH.D. IN COMPUTER SCIENCE

- Advised by Tim Sheard.
- Defended on May 8, 2017.

Portland, OR

2012 - 2017

### University of Central Florida

B.S. IN INFORMATION SYSTEMS TECHNOLOGY

- Minor in Computer Science.
- University Honors.

Orlando, FL

2005 - 2009

## Languages

**Programming** Agda, Cedille, Coq, Haskell, Javascript, Ruby  
**Spoken** English, German

## Experience

### University of Iowa

POSTDOCTORAL RESEARCHER

- Research on generic programming and zero-cost reuse in Curry-style dependent type theory.
- Contributed to the development of the dependently typed Cedille language.

Iowa City, IA

Jun 2017 - Current

### Portland State University

GRADUATE RESEARCH ASSISTANT

- Generic dependently typed programming over type theoretic models using Agda.
- Formal correctness proofs of programming languages (especially semantic termination) using Agda.
- Implementation of dependently typed languages (Ditto and Spire) using Haskell.
- Co-authored and awarded NSF/CISE/CCF grant #1320934.

Portland, OR

Aug 2012 - May 2017

### Engine Yard

SOFTWARE ENGINEER

- Worked on a cloud hosting platform on top of Amazon Web Services (AWS).
- Ruby web application and API programming using Ruby on Rails and Sinatra.
- Ruby system automation using Chef.
- Unit testing using RSpec.
- Integration testing using Cucumber and Selenium.

San Francisco, CA

May 2009 - Aug 2012

### IZEA

SOFTWARE ENGINEER

- Worked on a social media advertising platform.
- Ruby web application and API programming using Ruby on Rails.
- Unit testing using RSpec.

Orlando, FL

Jan 2007 - Aug 2008

### Bear Den Designs

SOFTWARE ENGINEER

- Worked on medical resident management software.
- Ruby web application programming using Ruby on Rails.
- Unit testing using Test::Unit.

Jacksonville, FL

May 2006 - Jan 2007

## Publications

### Generic Zero-Cost Reuse for Dependent Types

INTERNATIONAL CONFERENCE ON FUNCTIONAL PROGRAMMING (ICFP)

L. Diehl, D. Firsov, & A. Stump

2018

### Zero-Cost Coercions for Program and Proof Reuse

ARXIV DRAFT

L. Diehl & A. Stump

2018

## Fully Generic Programming over Closed Universes of Inductive-Recursive Types [↗](#)

PH.D. THESIS

*L. Diehl*

2017

## Generic Lookup and Update for Infinitary Inductive-Recursive Types [↗](#)

PROCEEDINGS OF THE 1ST INTERNATIONAL WORKSHOP ON TYPE-DRIVEN DEVELOPMENT

*L. Diehl & T. Sheard*

2016

## Hereditary Substitution by Canonical Evaluation (SbE) [↗](#)

TECHNICAL REPORT

*L. Diehl & T. Sheard*

2014

## Generic Constructors and Eliminators from Descriptions: Type Theory as a Dependently Typed Internal DSL [↗](#)

PROCEEDINGS OF THE 10TH ACM SIGPLAN WORKSHOP ON GENERIC PROGRAMMING

*L. Diehl & T. Sheard*

2014

## Leveling Up Dependent Types: Generic Programming over a Predicative Hierarchy of Universes [↗](#)

PROCEEDINGS OF THE 2013 ACM SIGPLAN WORKSHOP ON DEPENDENTLY-TYPED PROGRAMMING

*L. Diehl & T. Sheard*

2013

## Verified Stack-Based Genetic Programming via Dependent Types [↗](#)

PROCEEDINGS OF AAIP 2011 4TH INTERNATIONAL WORKSHOP ON APPROACHES AND APPLICATIONS OF INDUCTIVE PROGRAMMING

*L. Diehl*

2011

# Software

---

## Cedille [↗](#)

DEPENDENTLY TYPED PROGRAMMING LANGUAGE

*Agda*

2018

- Curry-style type theory.
- Closed universe of types.
- Impredicative quantification, intersection types, and heterogeneous equality.
- Interactive editing and navigation.

## Ditto [↗](#)

DEPENDENTLY TYPED PROGRAMMING LANGUAGE

*Haskell*

2015

- Open universe of types.
- Dependent pattern matching.
- Implicit arguments via dynamic pattern unification and constraint postponement.
- Mutual functions, induction-recursion, and induction-induction.
- Eta-equality for functions.
- Interactive holes and case splitting.
- Novel enhanced form of coverage checking.

## Spire [↗](#)

DEPENDENTLY TYPED PROGRAMMING LANGUAGE

*Haskell*

2013

- Proof of concept.
- Closed universe of types.
- Generic constructors and eliminators.

## Lemmachine [↗](#)

FORMAL WEB FRAMEWORK

*Agda*

2010

- Proof of concept.
- Request headers correct w.r.t. previous headers.
- Response headers and code correct w.r.t. previous request and headers.
- Verified HTTP parser.

## Dataflow [↗](#)

DATAFLOW CONCURRENCY LIBRARY

*Ruby*

2009

- Dataflow concurrency for Ruby inspired by the Oz programming language.