

# Ambrose Bonnaire-Sergeant Last update on May 22, 2018

[abonnaresergeant\[at\]gmail.com](mailto:abonnaresergeant[at]gmail.com) • [ambrosebs.com](http://ambrosebs.com) • [github.com/frenchy64](https://github.com/frenchy64)

Luddy Hall, Room 3025 • 700 N Woodlawn Ave • Bloomington • IN 47408 • USA

---

## Summary

*Programming language tooling developer, bringing academia to the masses.*

I love delivering tools and lessons from academia to working programmers. My work in building optional type systems has been incredibly rewarding, and fulfills a need of mine to make a real impact on other people's lives.

My work on Typed Clojure is indicative of my approach. I created the optional type system for Clojure in 2011. Each subsequent year has taken a new direction based on the feedback of the community, using my expertise to match solutions to problems.

I love the communicating and marketing ideas. It is both rewarding and challenging. I have orchestrated crowdfunding campaigns that have collectively raised over \$50,000 to support my goals in open source development—supported by companies like Cognitect, Walmart Labs, Prismatic, and CircleCI.

I have given many talks in industry and academic conferences, including Strangeloop, Clojure Conj, Philly ETE, Lambda Jam Brisbane, Code Mesh, ESOP, and the International Lisp Conference. I also enjoy talking at local Clojure user groups.

---

## Experience

### Research Assistant

INDIANA UNIVERSITY BLOOMINGTON

**Advisor:** Sam Tobin-Hochstadt

Fall 2014 — Summer 2015, Spring 2016 — Fall 2016

We have worked to formalise and publish a paper on the fundamentals of Typed Clojure. We are also working towards extending Typed Clojure with gradual typing. More recently, we are developing a framework to infer type annotations for optional type systems based on [runtime instrumentation](#).

### Assistant Instructor

INDIANA UNIVERSITY BLOOMINGTON

**Instructors:** Sam Tobin-Hochstadt, Suzanne Menzel, Chung-chieh Shan

Fall 2015, Fall 2016 —

Taught undergraduates C211, an introductory programming language course based on How To Design Programs. Also C343, introductory data structures. Conducted labs and office hours.

### Software Intern

SPARKFUND

May 2017 — February 2018

Studied verification techniques used in practice (Clojure).

### Google Summer of Code

CLOJURE ORGANISATION

**Student**

2012, 2013

**Mentor**

2014 (3 students), 2015 (2 students)

**Administrator**

2014, 2015

### Crowdfunded Open Source work

PERTH, WESTERN AUSTRALIA

**Typed Clojure**

2013

**Gradual Typing for Clojure**

2015

**Automatic Annotations for Typed Clojure and [clojure.spec](#)**

2016

### Analyst Programmer

UNIVERSITY OF WESTERN AUSTRALIA 2010 — 2011

I worked with the UWA library staff to gather information on the format and location of various University research data sources, and set up VIVO semantic web software. I used Java, XSLT, and bash, and administered several Linux machines.

## Education

[Indiana University Bloomington](#)

BLOOMINGTON, INDIANA

**Computer Science PhD student (In progress)**

Fall 2014 —

[Indiana University Bloomington](#)

BLOOMINGTON, INDIANA

**Master of Science in Computer Science**

2014 — 2017

[University of Western Australia](#)

WESTERN AUSTRALIA, AUSTRALIA

**BSc in Computer Science with Honours**

2008 — 2013

## Publications

Practical Optional Types for Clojure (ESOP16)

WITH ROWAN DAVIES, SAM TOBIN-HOCHSTADT

A Practical Optional Type System for Clojure (Honours dissertation)

SUPERVISED BY ROWAN DAVIES

## Selected Open Source Contributions

[core.typed](#)

LEAD DEVELOPER

[core.match](#)

CONTRIBUTOR

[Clojure](#)

CONTRIBUTOR

## Selected Talks

Typed Clojure: From Optional to Gradual Typing ([Video](#))

STRANGELOOP 2015

Typed Clojure in Practice ([Video](#))

STRANGELOOP 2014

Practical Optional Types for Clojure ([Slides](#))

ESOP 2016

Introduction to Logic Programming ([Video](#))

CLOJURE CONJ 2011

## Skills

**Technical expertise:** Experienced with open source development and maintenance, including Git, GitHub, and continuous integration. Enjoys building static and dynamic program verification tools for dynamically typed languages. Experienced Clojure programmer. Some experience in Java, C, Racket, and Scheme.

## Personal

Interests

Singing, guitar, clarinet, reading, teaching.