

# Kenny Foner

Email: **kwf@very.science** Website: **very.science**  
Phone: **(508) 318-8593** GitHub: **github.com/kwf**  
Location: **Philadelphia, PA** Pronouns: **they/them/their**

## ★ WHO AM I?

I'm a **functional programmer** and a **programming languages researcher**. I love building and working with type systems, testing tools, static analyses, and domain-specific languages. I delight in applying meaningful theory to solve real problems, and I seek to use my craft as a computer scientist to make a positive difference for programmers and for the world.

## ★ EDUCATION

**University of Pennsylvania** (Philadelphia, PA) May 2018  
MASTER OF SCIENCE IN ENGINEERING in Computer and Information Science  
Advised by Dr. Stephanie C. Weirich  
Teaching: *Advanced Programming (Haskell)*, *Software Foundations (Coq)*

**Brandeis University** (Waltham, MA) May 2015  
BACHELOR OF SCIENCE in Computer Science *summa cum laude* with highest departmental honors  
Advised by Dr. Harry G. Mairson  
Thesis: *Getting a Quick Fix on Comonads* (later published in *Haskell Symposium '15*)  
Teaching: *Functional Programming (Haskell)*, *Structure & Interpretation of Computer Programs (Scheme)*

## ★ EXPERIENCE

**Galois** (Arlington, VA/Remote) 2018 – present  
SOFTWARE ENGINEER/RESEARCHER  
I work on multiple projects: developing a verification API for tamper-resistant elections technology, implementing a typed programming language for high assurance shell scripts, and building new front-end functionality for the SAW/Cryptol suite of open-source program analysis tools.

**Microsoft Research** (Redmond, WA) Summer 2016  
RESEARCH INTERN  
I formalized the metatheory of several small programming languages as a stress test for the experimental Dafny language/proof assistant, and designed an intermediate language to verify a multi-part compilation pipeline. I contributed to Dafny's development, implementing bug fixes and feature improvements.

**Galois** (Portland, OR) Summer 2015  
RESEARCH INTERN  
I implemented a user-friendly interactive code generation utility for the open-source SAW suite of program analysis tools, designed to help cryptography domain experts get started more easily with automated formal verification. In a separate project, I worked on the design of a prototype graph query language.

**Galois** (Portland, OR) Summer 2014  
RESEARCH INTERN  
I created an embedded domain-specific language for secure distributed computations, implemented using oblivious secret-sharing protocols. I developed an optimizing compiler, and an efficient bytecode interpreter which was several times faster than the previous best results on a series of established benchmarks.

**MIT Lincoln Laboratory** (Lexington, MA) Summer 2013  
RESEARCH INTERN  
I prototyped applications and protocols to evaluate experimental frameworks for dynamic information flow control (IFC). Within one of these, I implemented a secure distributed multi-player game of *Battleship*, and contributed to a comparative analysis of IFC frameworks published in PLAS '14.

## ★ PUBLICATIONS/TALKS

**SPLASH PLMW '19:** “How Can I Academia When My Brain Can’t Even” (talk)

**Compose '19:** “Functors of the World, Unite!” (talk)

**ICFP '18:** KEEP YOUR LAZINESS IN CHECK. [K. Foner](#), H. Zhang, and L. Lampropoulos. In *Proceedings of the 2018 ACM SIGPLAN International Conference on Functional Programming*.

**ICFP '18:** “Keep Your Laziness in Check” (talk)

**ICFP '18:** WHAT’S THE DIFFERENCE? A FUNCTIONAL PEARL ON SUBTRACTING BIJECTIONS. B. Yorgey and [K. Foner](#). In *Proceedings of the 2018 ACM SIGPLAN International Conference on Functional Programming*.

**Haskell '17:** ODE ON A RANDOM URN (FUNCTIONAL PEARL). L. Lampropoulos, A. Spector-Zabusky, and [K. Foner](#). In *Proceedings of the 2017 ACM SIGPLAN Symposium on Haskell*.

**Compose '17:** “Choose Your Own Derivative” (talk)

**TyDe '16:** CHOOSE YOUR OWN DERIVATIVE (EXTENDED ABSTRACT). J. Paykin, A. Spector-Zabusky, and [K. Foner](#). In *Proceedings of the 2016 ACM SIGPLAN Workshop on Type-Driven Development*.

**Compose '16:** “There and Back Again and What Happened After” (talk)

**Haskell '15:** FUNCTIONAL PEARL: GETTING A QUICK FIX ON COMONADS. [K. Foner](#). In *Proceedings of the 2015 ACM SIGPLAN Symposium on Haskell*.

**Haskell '15:** “Functional Pearl: Getting a Quick Fix on Comonads” (talk)

**PLAS '14:** YOU SANK MY BATTLESHIP!: A CASE STUDY IN SECURE PROGRAMMING. A. Stoughton, A. Johnson, S. Beller, K. Chadha, D. Chen, [K. Foner](#), and M. Zhivich. In *Proceedings of the 2014 ACM Workshop on Programming Languages and Analysis for Security*.

## ★ PROGRAMMING BACKGROUND

**Expert Knowledge: Haskell** – I’ve been programming in Haskell for more than six years. I’ve authored libraries using advanced type-system extensions, implemented compiler plugins, and published novel research about high-performance immutable data structures, strictness analysis, and generic programming.

**Professional Experience: Rust** – I currently use Rust in my day-to-day work. I’ve used it to implement interpreters, compilers, type-checkers, and a memory-efficient deserialization library.

**Academic Experience: OCaml, Coq, Scheme** – I’ve used these languages in my academic career to collaborate on research, create course materials, and teach university classes.

**Past Experience: Clojure, Python, C, Java** – I’ve used these languages in less-recent work, in university courses, or in hobby projects. I can read them, and I’m comfortable working in them with the aid of reference materials.

## ★ OPEN SOURCE

I’m the primary author and maintainer of the **StrictCheck** library for randomized dynamic demand analysis, and an author and maintainer of the **Urn** data structure for updateable discrete probabilistic sampling. I’m a current contributor to the **Cryptol** specification language and the **Software Analysis Workbench**. I’ve also contributed to the **Glasgow Haskell Compiler** and the **Dafny** language.