

# Talia Lily Ringer

<http://tlringer.github.io/>

## EDUCATION

### University of Washington

2015 – Present

Ph.D. in Computer Science

M.S. in Computer Science

2017

Advisor: Dan Grossman

Programming Languages & Software Engineering

### University of Maryland, College Park

2008 – 2012

B.S. in Mathematics and Computer Science

Advisor: Lawrence Washington

Honors Thesis: [An Elliptic Curve Threshold Key Establishment Scheme](#)

## PUBLICATIONS

Talia Ringer, Nathaniel Yazdani, John Leo, and Dan Grossman.

Ornaments for Proof Reuse in Coq.

To Appear at ITP 2019. DEVOID [tool repository](#).

Talia Ringer, Karl Palmskog, Ilya Sergey, Milos Gligoric, and Zachary Tatlock.

QED at Large: A Survey of Engineering of Formally Verified Software.

Accepted with Minor Revisions to FTPL. [Project website](#).

Talia Ringer, Nathaniel Yazdani, John Leo, and Dan Grossman.

[Adapting Proof Automation to Adapt Proofs](#).

CPP 2018. [Talk video](#), PUMPKIN PATCH [tool repository](#).

Talia Ringer, Dan Grossman, Daniel Schwartz-Narbonne, and Serdar Tasiran.

[A Solver-Aided Language for Test Input Generation](#).

OOPSLA 2017. [Talk Video](#).

Talia Ringer, Dan Grossman, and Franziska Roesner.

[AUDACIOUS: User-Driven Access Control with Unmodified Operating Systems](#).

CCS 2016. [Talk Video](#).

## RESEARCH VISION

My vision is a future of **verification** and **proof engineering** that makes interactive theorem provers accessible to any programmer. My research applies techniques from **programming by example**, **program evolution**, and **proof reuse** to a view of **proof automation** that considers how verification projects change over time.

## CURRENT RESEARCH

### **Coq Change Analytics**

*with Alex-Sanchez Stern, Sorin Lerner, and Dan Grossman*

In recent years, verification efforts using interactive theorem provers like Coq have reached large, critical software projects. With this scale comes an interest in new proof engineering principles and technologies. In spite of this, there is little data on the development processes of proof engineers in the wild. We are developing and deploying a Coq plugin to a group of proof engineers that collects data on the changes that proof engineers make in development, then analyzing the data and using it to inform the next generation of proof automation tooling.

## STUDENTS ADVISED

Jasper Hugunin.

Constructing Inductive-Inductive Types in Cubical Type Theory.

To appear at FOSSACS 2019.

## HONORS & AWARDS

**NSF GRFP Fellow**

**Graduated with Honors in Computer Science**

**Graduation Speech Finalist**

**Corporate Scholar**

**Scholar Athlete**

*University of Washington*

*University of Maryland*

*University of Maryland*

*University of Maryland*

*University of Maryland*

## MENTORSHIP, DIVERSITY, & OUTREACH

**UW CSE Care Committee**

Lead organizer of a support network for graduate students in times of need.

*2019 – Present*

**Jewish Family Services**

ESL tutor and friendly visitor for an elderly refugee.

*2017 – Present*

**UW CSE**

Mentor for undergraduate women and graduate students in computer science.

*2015 – Present*

**UW QMP**

Mentor for LGBT students from any major.

*2016 – Present*

**The Identity Function**

Author of a [blog interview series](#) about LGBT computer science researchers.

*2016 – Present*

**TUNE House**

Mentor for undergraduate women in computer science.

*2015 – 2016*

**Amazon**

Technical and career mentor for software engineers.

*2012 – 2015*

## SERVICE

<b>CAV Artifact Evaluation Committee</b>	2019
<b>CoqPL Program Committee</b>	2019
<b>POPL Artifact Evaluation Committee</b>	2018, 2019
<b>ITP Sub-Reviewer</b>	2018
<b>University of Washington Graduate Admissions Committee</b>	2018

## TEACHING

<b>University of Washington</b> <i>Teaching Assistant for Concepts of Programming Languages</i>	<i>Fall 2018</i>
<b>University of Washington</b> <i>Teaching Assistant for Compilers</i>	<i>Winter 2016</i>
<b>University of Maryland, College Park</b> <i>Teaching Assistant for Computer and Network Security</i>	<i>Spring 2012</i>
<b>University of Maryland, College Park</b> <i>Mathematics and Computer Science Tutor for Student-Athletes</i>	<i>2010</i>

## INDUSTRY

<b>Amazon</b> <i>Research Scientist Intern</i> Worked with the Automated Reasoning Group on automatic test generation. Developed a solver-aided domain-specific language to generate test inputs.	<i>Summer 2016</i>
<b>Amazon</b> <i>Software Development Engineer</i> Worked with a team to develop the AmazonSupply website. Wrote and deployed code used company-wide and loaded hundreds of thousands of times per day. Developed a data flow analysis tool. Launched Amazon Business.	<i>2012 – 2015</i>
<b>Amazon</b> <i>Software Development Engineer Intern</i> Developed an internal web application to generate metadata for the AmazonSupply website in a safe and user-friendly manner. Enabled version control and staging for the metadata.	<i>Summer 2011</i>
<b>Carr Astronautics</b> <i>Corporate Scholars Program – Software Intern</i> Assisted in the development of a parallel image mosaicing application. Wrote code in C, MATLAB, and Java to read, alter, and write TIFF images with associated geographic data. Awarded a scholarship through the University of Maryland's Corporate Scholars Program.	<i>2010 – 2011</i>

## INTERESTS

Other academic interests of mine include **domain-specific languages, program analysis, type systems, category theory, algebra, computer security, and cryptology.**

I enjoy writing **Coq plugins** and have implemented several tutorial plugins to help other plugin developers. I have also [extended](#) Rosette to handle strings.

My favorite programming languages are **Coq, OCaml, and Rosette.**

I compete for **Club Northwest**, a top distance running club. I served on the board of Club Northwest from 2015 to 2016. My role was to promote our top runners through social media and writing. I ran **NCAA Division I Cross-Country** in 2009.

I also enjoy **logic and number puzzles, chocolatiering, writing poetry, singing, studying Russian, making bagels, and composing music for the piano.**