

## Tim Nelson

Brown University  
Computer Science Department  
115 Waterman St.  
Providence, RI, 02912

E-mail: [tim\\_nelson@brown.edu](mailto:tim_nelson@brown.edu)  
Web: <http://cs.brown.edu/~tbn/>

### EDUCATION

PhD in Computer Science (May 2013)  
Thesis: *First-Order Models for Configuration Analysis*  
Dissertation Supervisors: Prof. Kathi Fisler, Prof. Daniel J. Dougherty  
Worcester Polytechnic Institute, Worcester, MA

M.S. in Computer Science (May 2010)  
Worcester Polytechnic Institute, Worcester, MA

B.S. in Computer Science (May 2007) (summa cum laude)  
Worcester State University, Worcester, MA

A.A. in Computer Science (December 1999)  
Diablo Valley College, Pleasant Hill, CA

### REFEREED PUBLICATIONS

*Making Hay from Wheats: A Classsourcing Method to Identify Misconceptions*  
Siddhartha Prasad, Ben Greenman, Tim Nelson, John Wrenn, and Shriram Krishnamurthi  
Koli Calling International Conference on Computing Education Research 2022

*Little Tricky Logic: Misconceptions in the Understanding of LTL*  
Ben Greenman, Sam Saarinen, Tim Nelson, and Shriram Krishnamurthi  
The Art, Science, and Engineering of Programming 2022

*Applying Cognitive Principles to Model-Finding Output: The Positive Value of Negative Information*  
Tristan Dyer, Tim Nelson, Kathi Fisler, Shriram Krishnamurthi  
ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages & Applications (OOPSLA) 2022

*Automated, Targeted Testing of Property-Based Testing Predicates*  
Tim Nelson, Elijah Rivera, Sam Soucie, Thomas Del Vecchio, John Wrenn, and Shriram Krishnamurthi  
The Art, Science, and Engineering of Programming 2022

*Prototyping Formal Methods Tools: A Protocol Analysis Case Study*  
Abigail Siegel, Mia Santomauro, Tristan Dyer, Tim Nelson, and Shriram Krishnamurthi  
Protocols, Logic, and Strands: Essays Dedicated to Joshua Guttman on the Occasion of his 66.66 Birthday

*Solver-Aided Multi-Party Configuration*  
Kevin Dackow, Andrew Wagner, Tim Nelson, Shriram Krishnamurthi, and Theophilus A. Benson  
Hot Topics in Networking (HotNets) 2020

*Using Relational Problems to Teach Property-Based Testing*  
John Wrenn, Tim Nelson, and Shriram Krishnamurthi  
The Art, Science, and Engineering of Programming 2021 (**Editor's Choice award paper**)

*Synthesizing Mutable Configurations: Setting up Systems for Success*  
Tim Nelson, Natasha Danas, Theophilos Giannakopoulos, Shriram Krishnamurthi  
Workshop on Software Engineering for Infrastructure and Configuration Code (SEConfig) 2019

*CompoSAT: Specification-Guided Coverage for Model Finding*

Sorawee Porncharoenwase, Tim Nelson, and Shriram Krishnamurthi  
Symposium on Formal Methods (FM) 2018

*User Studies of Principled Model Finder Output*

Natasha Danas, Tim Nelson, Lane Harrison, Shriram Krishnamurthi, and Daniel J. Dougherty  
Software Engineering and Formal Methods (SEFM) 2017

*The Power of "Why" and "Why Not": Enriching Scenario Exploration with Provenance*

Tim Nelson, Natasha Danas, Daniel J. Dougherty, and Shriram Krishnamurthi  
Foundations of Software Engineering (FSE) 2017 (**award paper**)

*Switches are Monitors Too!: Stateful Property Monitoring as a Switch Design Criterion*

Tim Nelson, Nicholas DeMarinis, Timothy Adam Hoff, Rodrigo Fonseca, and  
Shriram Krishnamurthi Hot Topics in Networking (HotNets) 2016

*Simon: Scriptable Interactive Monitoring for SDNs*

Tim Nelson, Da Yu, Yiming Li, Rodrigo Fonseca, and Shriram Krishnamurthi  
Symposium on SDN Research (SOSR) 2015

*Exodus: Toward Automatic Migration of Enterprise Network Configurations to SDNs*

Tim Nelson, Andrew D. Ferguson, Da Yu, Rodrigo Fonseca, and Shriram Krishnamurthi  
Symposium on SDN Research (SOSR) 2015

*Static Differential Program Analysis for Software-Defined Networks*

Tim Nelson, Andrew D. Ferguson, and Shriram Krishnamurthi  
Symposium on Formal Methods (FM) 2015

*Tierless Programming and Reasoning for Software-Defined Networks*

Tim Nelson, Andrew D. Ferguson, Michael J. G. Scheer and Shriram Krishnamurthi  
Networked Systems Design and Implementation (NSDI) 2014

*Geometric Logic for Policy Analysis*

Salman Saghaifi, Tim Nelson and Daniel J. Dougherty  
Workshop on Automated Reasoning in Security and Software Verification (ARSEC) 2013

*A Balance of Power: Expressive, Analyzable Controller Programming*

Tim Nelson, Arjun Guha, Daniel J. Dougherty, Kathi Fisler, and Shriram Krishnamurthi  
Workshop on Hot Topics in Software Defined Networking (HotSDN) 2013

*Aluminum: Principled Scenario Exploration Through Minimality*

Tim Nelson, Salman Saghaifi, Daniel J. Dougherty, Kathi Fisler, Shriram Krishnamurthi  
International Conference on Software Engineering (ICSE) 2013

*Toward a More Complete Alloy*

Tim Nelson, Daniel J. Dougherty, Kathi Fisler, Shriram Krishnamurthi  
International Conference on Abstract State Machines, Alloy, B, and Z (ABZ) 2012

*The Margrave Tool for Firewall Analysis*

Tim Nelson, Christopher Barratt, Daniel J. Dougherty, Kathi Fisler, Shriram Krishnamurthi  
USENIX Large Installation System Administration Conference (LISA) 2010

*On the Finite Model Property in Order-Sorted Logic*

Tim Nelson, Daniel J. Dougherty, Kathi Fisler and Shriram Krishnamurthi  
Synthesis, Verification, and Analysis of Rich Models (SVARM) 2010

Copies of these papers are available at: <http://cs.brown.edu/~tbn/publications/>

## RESEARCH SOFTWARE

A selection of the software artifacts from the above papers includes:

### *Amalgam* and *Aluminum*

These are variants of the popular *Alloy Analyzer* model-finding tool. *Aluminum* (ICSE 2013) presents only minimal models—those in which every truth is necessary—and supports disciplined exploration by allowing users to make additions to models and discover the consequences. *Amalgam* (FSE 2017) provides rigorous, proof-based explanations for why a specification forces elements of a model to be present (or not present). This can help users discover and correct over-constraint—a particularly insidious form of specification bug.

### *Flowlog*

*Flowlog* (NSDI 2014) is a *tierless* language for programming Software-Defined Network (SDN) controllers. It provides a unified abstraction for the three disparate layers of programming for SDNs: the controller program itself, forwarding rules on switches, and the controller's internal state. This not only simplifies development, but allows the runtime to prevent several common network bugs. *Flowlog* also has built-in support for automated program verification.

### *The Margrave Tool for Policy Analysis*

*Margrave* (LISA 2010) provides concrete scenarios that illustrate how security policies behave and interact. One might ask *Margrave* how packets are handled differently by different paths through a network, or use it to discover which policy rules contribute to that difference. *Margrave* supports several real-world policy languages, as well as its own intermediate policy language, and provides a flexible query language for users interested in verifying properties or in narrowing the scope of scenarios given.

### *Forge*

Brown's Logic for Systems course teaches modeling and reasoning about systems via constraint solving. It relentlessly focuses on tools and applications—covering the necessary formalisms only as needed. The choice of tool used is therefore vital. Our starting point was the *Alloy Analyzer*, which is used in several textbooks and courses outside Brown and works especially well for teaching formal methods to a broad audience. *Forge* is a tool in the spirit of *Alloy* that has been designed specifically for *pedagogy*.

## WORK EXPERIENCE

Brown University Providence, RI

- July 2021 – Present: Lecturer, Department of Computer Science
- July 2018 – June 2021: Assistant Professor (Research), Department of Computer Science
- July 2017 – June 2018: Senior Research Associate, Department of Computer Science
- July 2013 – June 2017: Postdoctoral Research Associate, Department of Computer Science

Worcester Polytechnic Institute Worcester, MA

- Sept 2007 – June 2013: Graduate Research Assistant

Trust Tax Services of America Worcester, MA

- Feb 2000 – Mar 2002, Oct 2004 – Aug 2006: Programmer/Analyst

## TEACHING

Logic for Systems Brown University  
 CSCI 1710 (previously 1950-Y) Spring 2014–2023  
 Students are often introduced to formal logic only from a theoretical perspective, and never learn or develop an appreciation for how logic can be used to aid software design and implementation. *Logic for Systems* teaches logic in a different way: it is a project-based course where

students put concrete, logic-based tools—such as model finders, model checkers, and theorem provers—to work starting on the first day. The course also surveys some traditional logic results after students have seen their applicability. (The 2014 and 2015 offerings were titled “Logic for Hackers”; the 2014 offering was co-taught with Shriram Krishnamurthi.)

Since 2020, the course has used our pedagogic tool, “Forge”, which was developed in collaboration with Brown undergraduates.

Computing Foundations: Program Organization Brown University  
 CSCI 0112 Fall 2021–2022  
 This second-semester course follows on to the data-oriented introduction given in CSCI 0111. Many students in 0112 are non-CSCI concentrators or dual concentrators, yet the course is technically rigorous (covering topics such as asymptotic notation, machine learning, and random testing). (The Fall 2021 offering was co-taught with Rob Lewis.)

Software Engineering Brown University  
 CSCI 0320 Spring 2019–2023, Fall 2021–2022  
 CSCI 0320 is an intermediate-level software-engineering class focusing on designing, building, testing, and maintaining systems collaboratively. It covers programming techniques (using Java and TypeScript with various frameworks), object-oriented design, advanced testing (e.g., fuzz testing), debugging approaches, and tools such as source control systems. The course concludes with a major group project that students gather requirements for, then design and implement themselves.

Starting in Fall 2021, I have offered the course in both Spring and Fall semesters.

Executive Master’s in Cybersecurity Brown University  
 Human Factors (3 out of 10 modules) Spring 2019–Spring 2020  
 In Spring 2019, I taught three week-long online modules in Brown’s EMCS program. The material covered testing, configuration verification, and software-defined networking. These were part of a larger course on human factors in cybersecurity.

In Spring 2020, I expanded the material to four modules.

Integrated Intro to Computer Science (II) Brown University  
 CSCI 0180 Spring 2017–2018  
 CS 17/18 is one of two year-long introductory courses in computer science at Brown. CS 18 is the second semester of that course. (I co-taught the 2017 offering with Amy Greenwald, and the 2018 offering with Kathi Fisler.)

### Guest Lectures

- *Network Programming Languages (Cornell University)*
  - *Programming for Software-Defined Networks (Fall 2017)*
- *Programming Languages (Brown University)*
  - *Programming for Software-Defined Networks (Fall 2013)*
  - *Datalog (Fall 2014, 2015)*
  - *Network Programming and Synthesis (Fall 2019)*
- *Accelerated Intro to CS (Brown University)*
  - *Binary Decision Diagrams (Fall 2014)*
  - *Model Checking (Fall 2015)*
  - *Model Checking and Networks (Fall 2019)*
  - *Model Checking (Fall 2022)*
- *Multiprocessor Synchronization (Brown University)*

- *Model Checking (Fall 2014)*  
This one week lecture sequence included two homeworks.
- *Software Security Engineering (Worcester Polytechnic Institute)*
  - *Formal Security Models (Fall 2012)*

## Research with Undergraduates

Honors Theses Fall 2021 – Spring 2022  
Qianfan Chen’s thesis, “Language Levels in Teaching An Introductory Formal Methods Course” introduced a beginner-student language into the tooling for CSCI 1710 at Brown. This work enabled a range of benefits from more informative error messages to a reduction in prerequisites for the course.

Honors Theses Fall 2020 – Spring 2021  
Lucy Reyes’ thesis, “What You See Is Not Always What You Get: An Analysis of Informative Graphs in Formal Methods Languages”, studies the user benefit of local necessity, a concept introduced in my FSE 2018 Amalgam paper. Although the 2018 work received a best-paper award, it did no user studies to evaluate the impact of the concepts it introduced. Lucy embedded the core local-necessity algorithm in Forge, the model finder used in CSCI 1710, and proposed several new visualizations. She ran several user studies on Amazon’s Mechanical Turk platform, and found no statistically significant impact from local necessity in that setting.

Honors Theses Summer 2019 – Spring 2020  
Amy Huang’s thesis, “Mystery Functions” is an exploration of how students, developers, and others generalize their knowledge of functions from examples. With guidance from myself, Shriram Krishnamurthi, and Rob Goldstone (Psychology at Indiana University) Amy has developed a web application for giving mystery-functions quizzes. Participants are given a mystery function’s type and given the opportunity to send inputs and observe outputs, then guess the nature of the function.

Honors Theses Summer 2019 – Spring 2020  
Andrew Wagner’s work, which is related to my STR/DARPA collaboration, is on optimal configuration synthesis. Andrew investigated how to apply optimization techniques to our existing synthesis algorithms, and also worked on modeling service-mesh configurations. This latter work led to a publication at SEConfig (with Andrew is a co-author); Andrew went on to a PhD program at Northeastern University.

Honors Theses Fall 2015 – Spring 2016  
Adam Hoff’s thesis involved modifying a widely-used software switch to support rich debugging features. The resulting paper, with Adam as co-author, appeared at HotNets 2016.

Summer Research Fall 2020–Spring 2021  
Abigail Siegel and Mia Santomauro built a rapid-prototyping case study in the Forge tool (used in CSCI 1710), which resulted in an accepted paper (Guttmanfest 2022).

Summer Research Summer 2019–Summer 2021  
In 2019, I worked with two Brown students (Charlie Cutting and Luke West) on Forge, a new model-finding tool designed specifically for teaching. Thomas Del Vecchio continued this work in 2020. Ben Ryjikov and Qianfan Chen continued in 2021.

Forge is currently deployed in CSCI 1701. I expect to actively develop and refine Forge yearly with help from undergraduate researchers.

Summer Research Summer 2019 – Spring 2020  
Kevin Dackow worked with me on the same project that grew into Andrew Wagner’s honors thesis and our shared SEConfig publication (of which Kevin was also an author). Kevin went on to work for Google’s networking team.

Summer Research Summer 2017  
 I worked with a rising Senior (Sorawee Porncharoenwase) at Brown University on a followup to my Amalgam paper (FSE 2017). The resulting paper appeared in FM 2018. Sorawee went on to a PhD program at University of Washington. (This work was supported by the Randy Pausch undergraduate research award.)

Summer Research Summer 2013  
 I mentored a second-year undergraduate at Brown University in producing the initial version of the Flowlog programming language. The student was introduced to both software-defined networking and logic-programming languages. The resulting paper, with the student as co-author, appeared at NSDI in 2014.

Summer Research Summer 2010  
 I worked with a second-year undergraduate at Brown University to produce a new version of Margrave. I introduced the student to new concepts, including propositional and first-order logic and satisfiability checking. Later extensions to Margrave resulted in our LISA 2010 paper.

### **Tutoring**

Coding Practice Labs November 2012 – December 2012  
 Worcester Polytechnic Institute  
 I ran a weekly coding practice lab session for WPI's object-oriented design concepts class. I worked with groups of anywhere from one to twenty students, giving them fresh exercises to reinforce their regular lab sessions.

Computer Center Lab Assistant / Technician Sept 1997 – Dec 1999  
 Diablo Valley College  
 I tutored students in general computer use as well as computer-science course material: C, Java, QBASIC, data structures and algorithms. I also supervised other lab assistants.

## **PROFESSIONAL SERVICE**

### **Thesis and Exam Committees**

PhD Programming Comprehensive Exam (Brown University CS)  
 PhD Thesis Committee: Qi Xin (Brown University CS)  
 Comprehensive Exam Committee: Ahmed Agiza (Brown University CS)  
 Comprehensive Exam Committee: Natasha Danas (Brown University CS)  
 Comprehensive Exam Committee: Nicholas DeMarinis (Brown University CS)  
 Comprehensive Exam Committee: Da Yu (Brown University CS)

### **Program Committees and Reviewing**

Program Committee for ASE 2021 (Tools Track)  
 Program Committee for MoNeTech 2020  
 Program Committee for FM 2019  
 Program Committee for ESOP 2019  
 Program Committee for MoNeTech 2018  
 Program Committee for SOSR 2018  
 Program Committee for CoolSDN 2016  
 Program Committee for HOT-IoT 2016  
 External Review Committee for CAV 2016  
 Program Committee for SOSR 2016  
 Program Committee for CoolSDN 2015  
 Program co-chair of Spring 2015 ONUG Research Track

2011, 2012, 2013 Program Committees of Large Installation System Administration (LISA)  
External Reviewer for ASIACCS, CCS, SOSR, IEEE S&P, IEEE TNSM, ACM TISSEC,  
IJIS, SEFM

### Invited Talks

*Forge: Building a Pedagogic Solver Tool in Racket*  
RacketCon (October 2022)

*Getting Students to Properties*  
Newton Institute event on Formal Education (July 2022)

*Building a Formal Methods Class for Everybody*  
Tutorial Series of the Formal Methods Europe Teaching Committee (June 2022)

*The Human in Formal Methods*  
International Symposium on Formal Methods (FM) 2019  
(Note: this talk was joint with and delivered by Shriram Krishnamurthi.)

*PL in the Service of Configuration*  
Programming Languages Group Talk  
University of California San Diego, November 2019

*Disciplined Model Finding*  
Programming Languages Discussion Group  
Cornell University, September 2017

*When Good Programs Go Bad*  
A Day on College Hill (STEM Lecture)  
Brown University, April 2016

*Tierless Compilation and Reasoning for SDNs*  
Bellairs Seminar: Formal Methods for Software-Defined Networks  
Bellairs Research Institute, Barbados March 2016

*SAT Applications Tutorial (plus a pinch of Margrave)*  
Dagstuhl Seminar: Formal Foundations for Networking  
Schloss Dagstuhl – Leibniz Center for Informatics, February 2015

*Tierless Programming and Reasoning for SDNs*  
(with Shriram Krishnamurthi and Andrew D. Ferguson)  
Princeton University, February 2014

*The Margrave Project for Configuration Analysis*  
(with Shriram Krishnamurthi)  
Cornell University  
Formal Methods in Networking Summer School, June 2013

*Applications and Foundations of Scenario-Finding Tools*  
Boston University  
iBench Initiative Group, February 2013

*Applications and Foundations of Scenario-Finding Tools*  
Massachusetts Institute of Technology  
Software Design Group, December 2012

*The Margrave Tool for Firewall Analysis*  
ACM Symposium on Computer Human Interaction for Management of  
Information Technology, December 2011

### Other Contributions

Advisory Board for the Alloy Analyzer Model Finder

**EXTERNAL FUNDING**

\$1,004,152 from July 1, 2014: National Science Foundation CCF-1408745, "*A Balance of Power: Programming and Reasoning for Software-Defined Networks*" (lead author; could not formally be PI or co-PI due to position)

\$450,000 from July 1, 2017: National Science Foundation CCF-1714431, "*The Power of "Why?": Using Provenance for Disciplined Exploration in Model Finding*" (co-PI with PI Shriram Krishnamurthi)

\$815,111 from August, 2018 through January 2022 (extended to February 2022): DARPA (subcontract to STR), "*Minimizing Attack Surface Through Configuration Optimization*" (with Shriram Krishnamurthi)

\$ 297,881 from 09/01/2020: National Science Foundation 2039354, "*EAGER: SaTC-EDU: Instilling a Mindset of Adversarial Thinking into Computer Science Courses Early and Often*" (co-PI with Malte Schwarzkopf (PI) and Stephen H Bach, Kathi Fisler, and Shriram Krishnamurthi (co-PIs))

\$ 499,999 from 06/01/2022 through 05/31/2025: National Science Foundation 2208731, "*Pedagogical Tools for Formal Methods*" (co-PI with PI Shriram Krishnamurthi)

\$ 599,566 from 01/01/2023 through 12/31/2025: National Science Foundation 2227863, "*Little Tricky Logics: Misconceptions in Understanding Logics and Formal Properties*" (co-PI with PI Shriram Krishnamurthi, co-PIs Robert Y. Lewis and Milda Zizyte)

**OTHER**

First Place, Science Category  
WPI Innovation Exchange Poster Competition  
"*An Analyzable Language for Network Programming*"  
March 2013

Graduate Assistance in Areas of National Need (GAANN) Fellowship  
September 2012 – May 2013

Carl and Inez Weidenmiller Fellowship  
September 2008 – May 2009

WPI Institute Fellowship  
September 2007 – May 2008

Dean's List  
Worcester State University  
2004 – 2007