

Jean Yang

Statement of Goals

Research Mission

More and more people are programming: not only those with traditional Computer Science backgrounds, but also physical scientists, journalists, and more. We have an unprecedented opportunity to shape how people reason about software—and the systems that people do not yet call software. The goal of my research is to make *provable guarantees ubiquitous*. I am interested in designing (1) programming models, (2) language implementation strategies, (3) programming tools, and (4) software verification tools to make it easier for people to create the software they intended.

Outreach Mission

Public understanding and consumption determine the impact of research. In addition to educating my students to understand the importance of programming languages research, I am interested in *improving science communication* and *facilitating commercialization of technical ideas*. From 2013-2015 I co-directed NeuWrite Boston, a working group of scientists and science writers. To narrow the gap between academia and industry, I co-founded the Cybersecurity Factory, an accelerator for security startups.

Research Directions

I am investigating how the following two programming models facilitate the creation of programs and analyses that were previously difficult or impossible.

- **Policy-agnostic programming for security and privacy.** As an alternative to approaches for detecting information leaks, I propose a new programming model that factors out the specification of security and privacy concerns from the rest of the program and enforces the properties *by construction*. In my prior work I designed a language semantics for policy-agnostic programming with *informaton flow policies* and developed dynamic and static enforcement techniques. I am currently interested in (1) extending the approach for statistical privacy and (2) techniques for retrofitting legacy code with the policy-agnostic model, for purposes of fixing bugs and interacting with new policies and code.
- **Rule-based programming for biological modeling.** Traditionally, researchers model intracellular signalling using systems of ordinary differential equations (ODEs), but there are two problems with ODE models. First, a precise model requires a different ODE for each interaction between agents, causing ODE models to scale poorly with respect to number of agent types. Second, ODE models have little structure that we can exploit for scale-mitigating analyses. As an alternative, rule-based languages allow the representation of models as programs describing rewrites over graphs, where nodes correspond to proteins and edges describe protein complexes. Not only are these programs more concise than the corresponding ODE systems, but their structure also supports various analyses that are otherwise not possible. My current work focuses on analyzing *causal relationships* between rules, and combining causal information with language design and model-checking techniques to create biologically relevant model analyses.

Professional Employment

- 2016-present **Assistant Professor**, *Carnegie Mellon University*, Pittsburgh, PA.
Tenure-track position in the Computer Science Department with affiliate appointment in the Computational Biology Department.
- 2015-2016 **Assistant Professor (Adjunct)**, *Carnegie Mellon University*, Pittsburgh, PA.
Accepted and deferred tenure-track Assistant Professor position.

- 2015-2016 **Postdoctoral Researcher**, *Harvard Medical School*, Cambridge, MA.
Developing program verification and analysis techniques to aid in the construction of rule-based, graphical kinetic models for intracellular signalling.
- Summer 2015 **Co-founder**, *Cybersecurity Factory*, Cambridge, MA.
Started accelerator for early-stage cybersecurity companies with the goal of turning more research ideas into startups. Partnered with venture firm Highland Capital to run pilot. Program continues to run today.
- 2008-2015 **Research Assistant**, *Massachusetts Institute of Technology*, Cambridge, MA.
Graduate research in programming languages.
- Summer 2012 **Software Engineering Intern**, *Facebook, Inc.*, Menlo Park, CA.
Built verifier for backend privacy language. Continued contract work for four months post-internship and filed patent on resulting system.
- Summer 2010 **Research Intern, Programming Languages and Analysis Group**, *Microsoft Research*, Redmond, WA.
Worked with Nikhil Swamy and Juan Chen on extending a security-typed language Fine to support secure marshalling and cryptographic proofs. Resulted in publication of paper on the F* language in ICFP.
- Summer 2009 **Research Intern, Operating Systems Group**, *Microsoft Research*, Redmond, WA.
Worked with Chris Hawblitzel to build an operating system kernel verified for type-safety. Resulted in publication of paper on resulting Verve system in PLDI, Best Paper award, and the filing of a patent.
- Summer 2008 **Software Engineering Intern**, *Peerium, Inc.*, Cambridge, MA.
Worked at start-up creating a dependently typed functional language written in Haskell. Created parser for core language; wrote compiler optimizations; worked on GUI libraries.
- Summer 2007 **Software Engineering Intern**, *Google, Inc.*, Santa Monica, CA.
Completed standalone project on video search team using C++. Received full-time offer.
- Summer 2006 **Research Intern, Computational Biology Initiative**, *Harvard Medical School*, Boston, MA.
Worked with Dennis Wall and Leon Peshkin to develop and implement computational processes for tracing evolution and coevolution of presynaptic receptors.
- Summer 2005 **Software Development Intern**, *Mellon Financial*, Pittsburgh, PA.
Worked on data mapping and management project using SQL and ColdFusion.

Education

- 2010–2015 **Ph.D. computer science**, *Massachusetts Institute of Technology*, Cambridge, MA, USA.
Advisor: Armando Solar-Lezama. Thesis: *A Framework for Automatically Enforcing Information Flow Policies*.
- 2008–2010 **M.S. computer science**, *Massachusetts Institute of Technology*, Cambridge, MA, USA.
Advisor: Armando Solar-Lezama. Thesis: *Specification-Enhanced Programming*.
- 2004–2008 **A.B. computer science**, *Harvard University*, Cambridge, MA, USA.
Graduated *Magna Cum Laude*. Senior thesis advised by Greg Morrisett.

Awards and Honors

- 2016 *MIT Technology Review* Innovators Under 35.
Named one of thirteen alumni Centennial Laureates of The Ellis School.
- 2015 Keynote speaker, PrivacySecurityRisk.
Paul L. Penfield Student Service Award, MIT EECS.
- 2014-2015 Levine Fellowship.
- 2013 *GigaOm* 10 for 2013 Cloud Trailblazers.
- 2012-2013 Facebook Fellowship.
- 2010 Best Paper Award, Programming Language Design and Implementation (PLDI).
- 2008-2011 National Science Foundation Graduate Research Fellowship.
- 2008 Inducted into Phi Beta Kappa honor society.

Publications

Refereed Conference/Workshop Papers

- PLDI 2016 **Jean Yang**, Travis Hance, Thomas H. Austin, Armando Solar-Lezama, Cormac Flanagan, and Stephen Chong. Precise, Dynamic Information Flow for Database-Backed Applications. *Programming Language Design and Implementation*, 2016.
- JFP 2013 Nikhil Swamy, Juan Chen, Cédric Fournet, Pierre-Yves Strub, Karthikeyan Bhargavan, and **Jean Yang**. Secure Distributed Programming with Value-Dependent Types. *Journal of Functional Programming* 23(4), July 2013.
- PLAS 2013 Thomas H. Austin, **Jean Yang**, Cormac Flanagan, and Armando Solar-Lezama. Faceted Execution of Policy-Agnostic Programs. *Programming Languages and Security*, 2013.
- POPL 2012 **Jean Yang**, Kvat Yessenov, and Armando Solar-Lezama. A Language for Automatically Enforcing Privacy Policies. *Principles of Programming Languages*, 2012.
- CACM 2011 **Jean Yang** and Chris Hawblitzel. Safe to the Last Instruction: Automated Verification of a Type-Safe Operating System. *Communications of the Association for Computing Machinery*, December 2011.
- ICFP 2011 Nikhil Swamy, Juan Chen, Cédric Fournet, Pierre-Yves Strub, Karthikeyan Bhargavan, and **Jean Yang**. Secure Distributed Programming with Value-Dependent Types. *International Conference on Functional Programming*, 2011.
- PLDI 2010 **Jean Yang** and Chris Hawblitzel. Safe to the Last Instruction: Automated Verification of a Type-Safe Operating System. *Programming Language Design and Implementation*, 2010. **Best Paper Award**.

Technical Reports

- 2016 Nadia Polikarpova, Jean Yang, Shachar Itzhaky, and Jean Yang. Type-Driven Repair for Information Flow Security. <https://arxiv.org/abs/1607.03445>

Patents Issued

- 2013 Stephen C. Heise, **Jean Yang**, Dwayne Reeves, and Yiding Jia. Privacy verification tool. US20140282837 A1, filed March 15, 2013 and issued April 14, 2015.
- 2010 Chris Hawblitzel and **Jean Yang**. Automated verification of a type-safe operating system. US8341602 B2, filed February 27, 2010 and issued December 25, 2012.

Software Artifacts

Jeeves, a policy-agnostic language for automatically enforcing information flow policies, implemented in Python. Also released with Jeeves is an implementation of Jacqueline, a policy-agnostic web framework implemented on top of the Python Django web framework. (<https://github.com/jeanqasaur/jeeves>)

Ask Reeves, a backend privacy verifier. Built while interning at Facebook.

Verve, the first operating system automatically verified end-to-end for type safety. Built while interning at Microsoft.

Service

Technical Program Chairs

- 2018 Co-Chair, Principles of Programming Languages Artifact Evaluation Committee (POPL AEC), with Catalin Hritcu
- 2017 Static Analysis for Systems Biology (SASB), with John A. Bachman
Co-Chair, Principles of Programming Languages Artifact Evaluation Committee (POPL AEC), with Stephen Chong

Program Committees

- 2017 Principles of Programming Languages (POPL)
Computational Methods for Systems Biology (CMSB)
- 2016 Principles of Programming Languages Student Research Competition (POPL SRC)
IEEE Symposium on Security and Privacy (Oakland)
Programming Language Design and Implementation Program Committee (PLDI)
- 2015 Principles of Programming Languages Artifact Evaluation Committee (POPL AEC)

- 2014 ML Workshop
Other Committees
- 2018 Publicity Chair, Principles of Programming Languages Artifact Evaluation Committee (POPL)
University Service
- 2017-2018 Undergraduate Review Committee, SCS, CMU

Academic Advising

Current PhD Students

- Alison Kao, Ph.D. student (CMU, 2016-present)
- Jonathan Laurent, Ph.D., student (CMU, 2016-present)

Postdoctoral researchers

- Qinsi Wang, postdoctoral researcher (CMU, 2016-present)

Research Experience for Undergraduates

- Scott Krulcik, CMU, spring 2016-present, working on policy synthesis for information flow security
- Jacob Imola, CMU, spring 2016-present, working on policy-agnostic programming for statistical privacy
- Serena Wang, CMU, spring 2016-present, working on implementing the Jeeves language in Haskell
- Jordan Brown, CMU, fall 2016-present, working on type-driven repair for policy-agnostic programming in database-backed applications
- Ariel Jacobs, MIT, spring 2013-summer 2013, worked on applying policy-agnostic programming to HIPAA
- Benjamin Shaibu, MIT, spring 2012-spring 2014, worked on case studies for policy-agnostic programming

Doctoral Dissertation Committees

- Ferdinanda Camporesi, École Normale Supérieure, *Formal and exact reduction for differential models of signalling pathways in rule-based languages*. Defended January 2017.

Masters' Theses Supervised

- Chelsea Voss, MIT EECS, worked on *A Tool for Automated Inference in Rule-Based Biological Models*, completed May 2016.
- Travis Hance, MIT EECS, worked on *Jelf: A Web Framework for Automatic Enforcement of Privacy Policies*, completed May 2014.

Contract and Grant Support

- Funder: NSF
- 2017-2021 Title: SaTC: CORE: Towards a Usable, Practical, and Provably Secure Browser Infrastructure
People: Limin Jia (PI), Ljudevit Bauer, Matthew Fredrikson, Jean Yang
Awarded: \$1,200,000
- Funder: Center for Machine Learning and Health
- 2017-2018 Title: A Framework for Managing Patient Privacy
People: Jean Yang (PI), Matthew Fredrikson, Jian Ma
Awarded: \$300,000
- Funder: NSF
- 2017-2019 Title: CRII: SaTC: Repairing Code from Inferred Specifications of Information Flow Security
People: Jean Yang (PI)
Awarded: \$174,794
- Funder: DARPA
- 2017-2019 Title: KaZam: An integrated inference engine for assembly
People: Jean Yang (PI)
Awarded: \$400,000

Teaching

- Spring 2017 **Co-Instructor, Software Foundations of Security and Privacy (15-316)**, *Carnegie Mellon University*.
Co-designed and co-taught new undergraduate course with Matthew Fredrikson, with goal of teaching students to formally specify and prove security and privacy properties, with topics including safety, information flow, and statistical privacy. Reviews averaged 4.7/5.0 (teaching) and 4.6/5.0 (course) with course size 12.
- “Awesome class, good presentation and interesting topics.”
 - “. . . I made \$1000 from this course! After learning about implicit flows, I was able to identify security bugs in Instagram, caused by their lack of information flow control. Since it was a security bug, I received a reward from their bug bounty program.”
 - “. . . I really liked this class! It was great to look at these security concepts in a functional programming context.”
 - “Jean also gave some really great lectures, especially the more technical ones about information flow type systems.”
 - “I liked the course very much and the homework load was very fair.”
 - “Lectures were really fun and *[sic]* felt like I learned a lot over the course of the semester.”
- Fall 2016 **Instructor, Domain-Specific Languages (15-819)**, *Carnegie Mellon University*.
Designed and taught experimental discussion-based graduate special topics seminar on programming language design and evaluation across a variety of domains, with focused units on systems programming, security and privacy, and biological modelling. Reviews averaged 4.6/5.0 (teaching) and 4.0/5.0 (course) with course size of 8.
- "Definitely offer this course again and at the undergraduate level. Not many discussion-based courses happen in CS and they should."
 - "I feel like this course gave me a great opportunity to read a bunch of DSL papers and really understand the current state of the field. Thanks for a great semester!"
- Fall 2010 **Teaching Assistant, Foundations of Program Analysis**, *Massachusetts Institute of Technology*.
Designed and graded assignments and held recitations for graduate-level program analysis course.
- January 2010 **Instructor, C Memory Management and C++ Object-Oriented Programming**, *Massachusetts Institute of Technology*.
Designed and co-taught a for-credit Independent Activities Period (IAP) course for over 100 undergraduates. Prepared lectures and assignments; managed multiple graders; published materials on MIT's Open Courseware.
- January 2010 **Instructor, So You've Always Wanted to Learn Haskell?**, *Massachusetts Institute of Technology*.
Designed and co-taught an Independent Activities Period (IAP) course introducing the Haskell language and its applications.
- Spring 2008 **Teaching Fellow, Principles of Programming Languages**, *Harvard University*.
Helped with new course introducing programming languages concepts using the Coq proof assistant. Effectiveness rating 4.6/5.0. Received Certificate of Distinction in Teaching.
- Spring 2007 **Teaching Fellow, Introduction to Computer Science II**, *Harvard University*.
Responsible for problem sets, exams, section, and office hours for course using Scheme and C++. Effectiveness rating 4.6/5.0; nominated for Undergraduate Council's Levenson Teaching Prize.
- Fall 2006 **Teaching Fellow, Introduction to Formal Systems**, *Harvard University*.
Responsible for problem sets, exams, section, and office hours for course on computational models and complexity. Effectiveness rating 4.2/5.0. Nominated for departmental teaching award.
- Fall 2005 **Course Assistant, Introduction to Calculus**, *Harvard University*.
Graded problem sets and ran weekly problem session. Effectiveness rating 4.4/5.0.

Speaking

Invited Research Talks

- 2017 “Preventing Information Leaks by Construction”
- Curry On Barcelona! (June 2017)
 - Programming Languages Mentory Workshop, colocated with PLDI (June 2017)
 - École Normale Supérieure (January 2017)
- “Analyses for Rule-Based Models of Cellular Signalling”, Cambridge University (May 2017)
- “Policy-Agnostic Programming for Preventing Information Leaks”

- INRIA Paris (June 2017)
- Oxford University (May 2017)
- The 2nd Summit on Advances in Programming Languages (May 2017)
- “Policy-Agnostic Programming for Database-Backed Applications,” Cornell University (April 2017)
- “Preventing Information Leaks by Construction,” École Normale Supérieure (January 2017)
- 2016 “A Logical Deduction Tool for Assembly,” 6th Workshop on Logic and Systems Biology (July 2016)
- 2015 “Preventing Information Flow Leaks with Jeeves,” Singapore Data Privacy Workshop (July 2015)
- “Preventing Information Leaks with Jeeves”
 - Columbia University (Special Seminar, February 2015)
 - University of California, Berkeley (Special Seminar, March 2015)
 - University of Illinois, Urbana-Champaign (Special Seminar, April 2015)
 - Carnegie Mellon University (Special Seminar, April 2015)
 - Microsoft Research Redmond (April 2015)
 - Samsung Research (April 2015)
- 2011-2014 “Jeeves: A Language for Automatically Enforcing Privacy Policies”
 - Cornell University (August 2014)
 - Columbia University (May 2014)
 - Microsoft Research - Cambridge (October 2013)
 - Gigaom Structure Conference (June 2013)
 - Tufts University (Colloquium, December 2012)
 - Brown University (June 2012)
 - University of California, Berkeley (April 2012)
 - Google - Mountain View (April 2012)
 - Facebook - Menlo Park (March 2012)
 - Northeastern University (December 2011)
 - Harvard University (December 2011)
 - Google - New York (July 2011)
 - New York University (April 2011)

Invited Panels

- 2016 “What I wish I knew when I started grad school,” Programming Languages Mentoring Workshop (PLMW) at the Programming Languages Design and Implementation (PLDI) conference, Santa Barbara, CA, 2016.

Selected Public Speaking

- 2017 “Programming Languages for Biological Modeling,” WECODE Harvard, Cambridge, MA (February 2017).
- 2016 “Meet the Innovators Under 35,” *MIT Technology Review* EmTech Conference, Cambridge, MA (September 2016).
- “Securing Software by Construction,” Philly Emerging Tech Conference, Philadelphia, PA (April 2016).
- 2015 “On the Front Lines: New Risks and Knowledge,” panel at *AtlanticLIVE*’s “Cybersecurity Today” summit, Washington, DC (October 2015).
- “Cybersecurity: How to Use What We Already Know,” keynote at PrivacySecurityRisk, Las Vegas, NV (October 2015).
- 2014 “A Brief History of Programming”
 - Geek Girl Dinner Boston, Cambridge, MA (December 2014).
 - Women’s Coding Collective Boston, Cambridge, MA (December 2014).
- “An Axiomatic Basis for Computer Programming,” Papers We Love NYC, New York, NY (November 2014).

- “Challenging Technical Privilege: How Race and Gender Matter,” MIT, Cambridge, MA (October 2014).
- 2013 “Graduate School 101,” panel at Scientista Symposium, MIT, Cambridge, MA (April 2013).
- “How I Got There,” panel at Women in Advanced Computing (WiAC) Summit, San Jose, CA (June 2013).

Selected Podcast Appearances

- 2017 “Secure Programming with Jean Yang,” *The Women in Tech Show*, February 13, 2017.
- 2015 “Three Female Computer Scientists Walk into an AMA,” *Upvoted*, February 18, 2015.

Writing for the Public

Articles About Technology

- 2017 “James Comey’s Twitter Security Problem Is Your Problem, Too,” **Jean Yang**, *MIT Technology Review*, April 27, 2017.
- “Building privacy right into software code,” **Jean Yang**, *The Conversation*, February 20, 2017.
- 2016 “Research for Practice: Web Security and Mobile Web Computing,” **Jean Yang**, *ACM Queue*, October 4, 2016.
- “The robot apocalypse is here: and it’s not what we expected,” **Jean Yang**, *ACM SIGCAS*, March 2016.
- “Making It Easy to Make Apps.” **Jean Yang**, *MIT Technology Review*, June 21, 2016.
- 2015 “The Real Software Security Problem is Us.” **Jean Yang**, *MIT Technology Review*, June 22, 2015.
- “C is Manly, Python is for ‘n00bs’: How False Stereotypes Turn Into Technical ‘Truths.’” **Jean Yang** and Ariel Rabkin, *Model View Culture*, January 20, 2015.

Other Articles

- 2016 “Sex, Crime, and Justice,” **Jean Yang**, *Jacobin Magazine*, September 30, 2016.
- 2014 “MIT Computer Scientists Demonstrate the Hard Way that Gender Still Matters,” Elena Glassman, Neha Narula, and **Jean Yang**, *Wired*, December 29, 2014.
- “I Worked Without Email for 10 Days, Survived, and Even Learned Something,” **Jean Yang**, *Newsweek* via *The Daily Muse*, September 16, 2014.
- “First person / The power of girls,” **Jean Yang**, *Pittsburgh Post-Gazette*, May 31, 2014.
- “Only Woman on the Team? 4 Communication Tips You Need to Know,” **Jean Yang**, *Forbes* via *The Daily Muse*, March 26, 2014.

Popular Blog Posts

- 2016 “What to Wear for Academic Interviews, or How to Dress Like a Man Without Looking Like a Man,” *Blogger*, December 25, 2015.
- “Why It’s Not Academia’s Job to Produce Code That Ships,” *Blogger*, May 7, 2016.
- “What My PhD Was Like,” *Blogger*, February 27, 2016.
- 2015 “What do you think of Scott Aaronson’s comment #171 and the subsequent posts?,” *Quora*, January 12, 2015.
- 2014 “Speaking Fake-Polish is No Different from Speaking ‘Male,’” *Medium*, February 4, 2014.

In the Media

Selected Press

- MIT Tech Review “Why don’t computers keep our personal data secure by default?” Patrick Doyle, *MIT Technology Review*, August 23, 2016.
- TechCrunch “Coding In The Cloud Era Demands A Structural Rethink To Bake In Security And Privacy.” Natasha Lomas, *TechCrunch*, Sept. 27, 2015.
- Wired “The Quest to Rescue Security Research from the Ivory Tower.” Klint Finley, *Wired*, July 2, 2015.
- Fortune “Cybersecurity Factory Nurtures Early-Stage Startups in a Tough Field.” Barb Darrow, *Fortune*, June 26, 2015.

- Boston Globe “MIT Students, Highland Capital, Partner to Launch Cybersecurity Factory.” Janelle Nanos, *The Boston Globe*, March 31, 2015.
- Fast CoExist “A Better Way To Protect Privacy? Take The Programmer Out Of The Equation.” Jessica Leber, *Fast CoExist*, March 7, 2014.
- Wired “Out in the Open: A New Programming Language With Built-In Privacy Protocols.” Klint Finley, *Wired*, March 3, 2014.
- Gigaom “Want to build privacy into your apps? Check out Jeeves, now available in Python.” Barb Darrow, *Gigaom*, Feb. 11, 2014.
- MIT Tech Review “New Programming Language Removes Human Error from Privacy Equation.” MIT CSAIL, *MIT Technology Review*, Feb. 10, 2014.
- Gigaom “Cloud Trailblazers: 10 for 2013. Mission Possible? Jean Yang.” Barb Darrow, *Gigaom*, May 28, 2013.
- New Scientist “What your online friends reveal about where you are.” Jacob Aron, *New Scientist*, January 25, 2012.

Book Appearances

Andi Deihn. *Technology: Cool Women Who Code*. Nomad Press, 2015.

Other Interests and Activities

My interests in programming language design and software verification come from a desire to empower. For similar reasons I am interested in promoting equal opportunity in science, technology, engineering, and mathematics (STEM) fields, as well as science communication and outreach. My main activities have been as follows:

- In 2009 I co-founded **Graduate Women at MIT**, an organization that, as of spring 2014, had 1,800 mailing list members, over 80 planning committee members, and an annual budget of over \$20K involved in its three flagship activities: the fall Leadership Conference, the Spring Empowerment conference, and a year-round mentoring program. We won the William L. Steward, Jr. Award for “outstanding contributions” to extracurricular activities.
- In 2010 I started the **MIT Programming Languages and Software Engineering Offsite**, now an annual mini-conference of the 6+ research groups working in related areas at MIT.
- From 2013-2015 I ran **NeuWrite Boston**, a collaborative working group of scientists and writers, with Amanda Geffer, author of *Trespassing on Einstein's Lawn*.

As an undergraduate at Harvard I co-founded the Harvard College Engineering Society, an activity that involved starting the RoboCup team for competing in autonomous soccer competitions. I am happy to see that the team remains active today.