

I am currently an Associate Professor in Computer Science at Washington State University, and was previously Assistant Professor at the University of Alabama at Birmingham and a Victor Basili Fellow at the University of Maryland, at College Park.

Research Summary

My research interests center around the design and (scalable) implementation of high-level programming languages and systems for reasoning automatically about programs. I have contributed to the design of tunable whole-program control-flow analyses, information-flow analyses, contract verification systems, and have invented novel Datalog-based languages for specifying these analyses declaratively and implementing them efficiently. I develop high-performance-computing techniques for accelerating data-parallel relational algebra and sparse linear algebra on clusters and GPUs.

Education

University of Utah	Ph.D.	2017
---------------------------	-------	------

Developed a unified methodology for polyvariant (*e.g.*, flow/call/arg/obj sensitive) program analysis.

I applied this framework to produce a self-reflective form of polyvariance for continuations that avoids all return-flow conflation of values (a long-standing problem for control-flow analyses), guaranteeing ideal stack precision at no cost to average or worst-case model complexity, and at in terms of human labor to implement ($\sim 1\text{LOC}$ in proposed framework). Proved that the precision is equal to an incomputable analysis with an unbounded stack and mechanically verified the proof using the Coq proof assistant. My dissertation is titled “*Introspective Polyvariance for Control-Flow Analyses*”.

University of Utah	M.S.	2012
University of Oregon	B.S.	2010

Employment

Associate Professor	Washington State University	2024 - Present
----------------------------	-----------------------------	----------------

Working on research into scalable, tunable program analysis, program verification, logic solvers on high performance computers and clusters, and linguistic mechanisms for enforcing correctness, security/privacy, and termination properties of software; regularly teaching undergraduate and graduate-level classes in compilers and programming languages. Won an ARPA-H subaward to investigate private and secure querying for EHR systems, contracting with Galois, inc.

Assistant Professor	U. of Alabama, Birmingham	2018 - 2024
----------------------------	---------------------------	-------------

Developed my research program in high-performance reasoning and program analysis; regularly taught undergraduate and graduate-level classes in automata theory, programming languages, and automated reasoning. Won NSF PPOSS Large, NSF PPOSS Planning, and DARPA VSPILLS grants, bringing a total of \$3.12M in new research money to UAB.

Victor Basili Fellow	U. of Maryland, College Park	2016 - 2018
-----------------------------	------------------------------	-------------

Joined UMD’s PLUM lab with Michael W. Hicks, Jeffery Foster, and David Van Horn; worked on various collaborative projects including: soft contract verification, approximating permission-use provenance in Android, accelerating flow analyses in Datalog, and verification of faceted programs, among others. A departmental fellowship granted me great freedom to pursue long-term research.

Instructor	U. of Maryland, College Park	2017 - 2018
-------------------	------------------------------	-------------

While at UMD, I taught a section of the Intro to Programming Languages course and developed a new Compilers course in which my students built an R⁷RS Scheme Compiler from scratch.

Analysis Developer HP, inc. (was: Fortify, inc.) 2013 - 2015

Following a graduate internship in Summer 2013, I was hired to work remotely, concurrent with my work in the U-combinator lab. I implemented a new inter-procedural must-alias analysis based on the theory of *abstract counting* to supplement existing analyses in the Fortify source-code analyzer (SCA). I successfully addressed various bug reports and lead an architectural review of SCA's "phase 0" preprocessing code which performs SSA conversion and intra-procedural analysis.

Research Assistant U. of Utah 2011 - 2016

I studied polyvariant program analysis in Matt Might's U-combinator lab, implementing analyses as traditional operational semantics (abstract interpreters) and as linear algebra, to exploit inherent parallelism, and designing a new dynamic sparse-matrix format for SIMD hardware (such as GPUs). I worked on Android analysis and timed challenge events for DARPA's APAC program.

Teaching Assistant U. of Utah 2010 & 2012

I taught three sections of 200-level *Discrete Structures* as a teaching assistant and also helped to develop projects and tests for *Compilers* and *Advanced Compilers*, taught by my advisor.

Co-founder, Developer Peculi, llc. 2006 - 2009

Helped to raise \$3,000 in start-up funds to build an art-sharing platform. Built out a high-performance backend and AJAST-style load-balancing to handle requests from a web-gallery front-end in Javascript. The site was home to more than 14,000 works of art contributed by roughly 800 active users. Wrote a message-board engine with social-media features. Wrote a secure CAPTCHA and various image processing scripts. Served as system administrator for the site's lifetime.

Intern, Freelance, etc. Various 2005 - 2010

Before my graduation from UO in 2010, I worked a variety of jobs: for HP, inc. developing tooling and a tutorial for using 3DSMAX to visualize Autodesk Inventor models; for Prof. Ostroverkhova (at OSU) developing a Latex-based CMS for PhD students in Physics; for fourteen companies or individuals (some repeatedly) as a freelance developer, mostly creating dynamic websites, implementing CSS designs, and customizing CMS deployments; for a startup VizMe, inc. co-founded by Prof. Eric Wills (from UO), setting up a Solr-based search engine in Java and writing Python scripts to manage MySQL databases, site-localization, and image analyses.

Awards & Recognition

- Directors Direction (D.D.) Grants totaling upward of 5M hours on ALCF's Theta (2019-2025).
- Grant Award: ARPA-H: Healthy Docs: Secure Interoperable Health Data Management (\$1.29M of \$10M total) (2025).
- Grant Award (as Lead PI): NSF: PPOSS: Large: A Full-stack Approach to Declarative Analytics at Scale (\$1.05M of \$5M total) (2023).
- Grant Award (as Lead PI): NSF: PPOSS: Planning: A Full-stack Approach to Declarative Analytics at Scale (\$83,116 of \$250,000 total) (2022).
- Grant Award: DARPA: VSPELLS: Promotion to Optimal Languages Yielding Modular Operator-driven Replacements and Programmatic Hooks (POLYMORPH) (\$400,000 of \$8M total) (2021).
- Won ISC Hans Meuer Best Paper award (2020).
- Featured by ALCF: my work advancing scalable MPI-based relational algebra was featured by ALCF's yearly *Science Report* magazine as a research highlight stemming my 2019-2020 D.D. grant of hours on ALCF's Theta supercomputer with Sidharth Kumar. https://www.alcf.anl.gov/sites/default/files/2021-04/ALCF_2020ScienceReport.pdf (page 37) (2020).

- Invited to the Journal of Functional Programming (2019).
- Won HiPC Best Paper award (2019).
- Invited to the Journal of Functional Programming (2018).
- Won PRACE ISC best paper award (2016).
- Victor Basili Fellowship at the University of Maryland, College Park (2016).
- Won TFP best student paper award (2013).

Teaching

(WSU) CptS 580: Advanced Programming Languages (Spring 2025)

I designed from scratch, and taught, an advanced programming-language theory course for graduate students at WSU, Spring 2025. The course focused on programming language theory, the lambda calculus, different styles of formal semantics, especially operational semantics, interpreters, type systems and let polymorphism, and several styles of program analysis.

(WSU) CptS 452: Compilers (Fall 2024)

I designed from scratch, and taught, a course on compiler implementation at WSU. The class was based around a central project, composed of manageable pieces, where students build their own Scheme-to-C compiler from scratch. The course covered standard intermediate representations, implementation of various constructs via first-class control, implementation of lambdas as closures, and a runtime that integrates the Boehm garbage collector.

(UAB) CS 660/760: Artificial Intelligence (Fall 2019–2021, 2023)

I designed and taught a course that focuses on symbolic AI, planning, logic, and automated reasoning (UAB had six other courses focusing on Machine Learning, which is outside my primary areas of expertise). We focused on problem solving by search, adversarial search and games, automated constraint solving, SAT solving, and logic.

(UAB) CS 401/501: Programming Languages (Spring 2019–2024)

I designed and taught a Programming Languages course for undergraduates at UAB. The course surveys a variety of languages features and paradigms, focusing especially on introducing functional and declarative programming, language design, type systems, semantics, and interpreters. The class is programming intensive, with a series of six projects and biweekly exercises, and includes a modest focus on implementation strategies. Students build a Church compiler, small-step and big-step interpreters, and a purely functional implementation of PageRank, among other unique projects.

(UAB) CS 350/550: Automata and Formal Languages (Fall 2018–2023)

I improved UAB's course on automata, formal languages, and the theory of computation by adding two substantial coding projects to supplement the class' theoretical focus. I added a project where students build an automata library to convert a given regular expression to an NFA, convert this to a DFA, and then minimize the DFA. This library can match strings using any intermediate model of regular languages and can visualize any NFA/DFA using Graphviz so students can visually inspect the NFAs and DFAs they are building. I have also increased the class' focus on parsing and added a project where students write their own recursive-descent parser for a small language.

(UMD) CMSC 330: Intro to Programming Languages (Spring 2018)

I taught a spring section of CMSC 330, UMD’s core Programming Languages course which covers a series of languages and paradigms (using Ruby, OCaml, Rust, Prolog) and core PL theory on lambda calculus, semantics, and type systems.

(UMD) CMSC 430: Compilers (Fall 2017)

I designed and taught an elective course on compiler implementation at UMD. The class was based around a central project, composed of bite-sized chunks, where students build their own Scheme-to-LLVM-IR compiler, from scratch.

Doctoral Theses

1. *Introspective Polyvariance for Control-Flow Analyses*. **Thomas Gilray**. University of Utah. (U of U) 2016.

Journal Papers (Refereed)

† Indicates a student advisee and ‡ indicates a student mentee.

2. *Datalog with First-class Facts*. **Thomas Gilray**, Arash Sahebolamri†, Yihao Sun‡, Sowmith Kunapaneni†, Sidharth Kumar, and Kristopher Micinski. International Conference on Very Large Data Bases. (VLDB) Sep 2025.
3. *Abstract Allocation as a Unified Approach to Polyvariance in Control-flow Analyses*. **Thomas Gilray**, Michael D. Adams, and Matthew Might. Journal of Functional Programming. (JFP) 2018. <https://doi.org/10.1017/s0956796818000138>
Invited to the Journal of Functional Programming.

Conference Papers (Refereed)

4. *Multi-Node Mutli-GPU Datalog*. Ahmedur Rahman Shovon†, Yihao Sun‡, **Thomas Gilray**, Sidharth Kumar, and Kristopher Micinski. International Conference on Supercomputing. (ICS) Jun 2025.
5. *Optimizing Datalog for the GPU*. Yihao Sun‡, Ahmedur Rahman Shovon†, **Thomas Gilray**, Sidharth Kumar, and Kristopher Micinski. International Conference on Architectural Support for Programming Languages and Operating Systems. (ASPLOS—12.7% acceptance) Mar 2025. <https://doi.org/10.1145/3669940.3707274>
6. *Column-Oriented Datalog on the GPU*. Yihao Sun‡, Sidharth Kumar, **Thomas Gilray**, and Kristopher Micinski. AAAI Conference on Artificial Intelligence. (AAAI—23.4% acceptance) Feb 2025.
7. *Analysis of MPI Communication Time for Distribution of Repartitioned Data*. John-Paul Robinson‡, Ke Fan†, Sidharth Kumar, Steve Petruzza, and **Thomas Gilray**. Practice and Experience in Advanced Research Computing. (PEARC) Jul 2024. <https://doi.org/10.1109/HiPCW61695.2023.00020>
8. *Configurable Algorithms for All-to-all Collectives*. Ke Fan†, Steve Petruzza, **Thomas Gilray**, and Sidharth Kumar. ISC High Performance. (ISC—29% acceptance) May 2024. <https://doi.org/10.23919/ISC.2024.10528936>
9. *Communication-Avoiding Recursive Aggregation*. Yihao Sun‡, Sidharth Kumar, **Thomas Gilray**,

and Kristopher Micinski. IEEE International Conference on Cluster Computing. (CLUSTER) Nov 2023. <https://doi.org/10.1109/CLUSTER52292.2023.00024>

10. *Towards Iterative Relational Algebra on the GPU*. Ahmedur Rahman Shovon†, **Thomas Gilray**, Kristopher Micinski, and Sidharth Kumar. USENIX ATC. (USENIX ATC) Jul 2023.

11. *Optimizing the Bruck Algorithm for Non-uniform All-to-all Communication*. Ke Fan†, **Thomas Gilray**, Valerio Pascucci, Xuan Huang, Kristopher Micinski, and Sidharth Kumar. International ACM Symposium on High-Performance Parallel and Distributed Computing. (HPDC—19% acceptance) Jun 2022. <https://doi.org/10.1145/3502181.35314>

12. *Seamless Deductive Inference Via Macros*. Arash Sahebolamri†, **Thomas Gilray**, and Kristopher Micinski. International Conference on Compiler Construction. (CC) Apr 2022. <https://doi.org/10.1145/3497776.35177>

13. *Load-balancing Parallel I/O of Compressed Hierarchical Layouts*. Ke Fan†, Duong Hoang, Steve Petruzza, **Thomas Gilray**, Valerio Pascucci, and Sidharth Kumar. International Conference on High Performance Computing, Data, and Analytics. (HiPC) Dec 2021. <https://doi.org/10.1109/HiPC53243.2021.00048>

14. *Compiling Data-parallel Datalog*. **Thomas Gilray**, Sidharth Kumar, and Kristopher Micinski. International Conference on Compiler Construction. (CC) Mar 2021. <https://doi.org/10.1145/3446804.3446855>

15. *Load-balancing Parallel Relational Algebra*. Sidharth Kumar and **Thomas Gilray**. ISC High Performance. (ISC—31% acceptance) Jun 2020. https://doi.org/10.1007/978-3-030-50743-5_15
Won ISC Hans Meuer Best Paper award.

16. *Abstracting Faceted Execution*. Kristopher Micinski, David Darais, and **Thomas Gilray**. IEEE Computer Security Foundations Symposium. (CSF) Jun 2020. <https://doi.org/10.1109/CSF49147.2020.00021>

17. *Distributed Relational Algebra at Scale*. Sidharth Kumar and **Thomas Gilray**. International Conference on High Performance Computing, Data, and Analytics. (HiPC—23% acceptance) Dec 2019. <https://doi.org/10.1109/HiPC.2019.00014>
Won HiPC Best Paper award.

18. *Size-Change Termination as a Contract*. Phúc C. Nguyễn†, **Thomas Gilray**, Sam Tobin-Hochstadt, and David Van Horn. Programming Language Design and Implementation. (PLDI—27% acceptance) Jun 2019. <https://doi.org/10.1145/3314221.3314643>
Invited to the Journal of Functional Programming.

19. *Soft Contract Verification for Higher-order Stateful Programs*. Phúc C. Nguyễn†, **Thomas Gilray**, Sam Tobin-Hochstadt, and David Van Horn. Symposium on Principles of Programming Languages. (POPL—23% acceptance) Jan 2018. <https://doi.org/10.1145/3158139>

20. *User Comfort with Android Background Resource Accesses in Different Contexts*. Daniel Votipka, Seth M. Rabin, Kristopher Micinski, **Thomas Gilray**, Michelle L. Mazurek, and Jeffrey S. Foster. Symposium on Usable Privacy and Security. (SOUPS—21% acceptance) 2018.

21. *Allocation Characterizes Polyvariance*. **Thomas Gilray**, Michael D. Adams, and Matthew Might. International Conference on Functional Programming. (ICFP—31% acceptance) Sep 2016. <https://doi.org/https://doi.org/10.1145/3022670.2951936>

- 22.** *Pushdown Control-Flow Analysis for Free.* **Thomas Gilray**, Steven Lyde, Michael D. Adams, Matthew Might, and David Van Horn. Symposium on Principles of Programming Languages. (POPL—23% acceptance) Jan 2016. <https://doi.org/10.1145/2837614.2837631>
- 23.** *Dynamic Sparse-Matrix Allocation on GPUs.* James King, **Thomas Gilray**, Robert M. Kirby, and Matthew Might. ISC High Performance. (ISC) 2016. https://doi.org/10.1007/978-3-319-41321-1_4
Won PRACE ISC best paper award.
- 24.** *A Survey of Polyvariance in Abstract Interpretations.* **Thomas Gilray** and Matthew Might. Symposium on Trends in Functional Programming. (TFP) May 2013.
Won TFP best student paper award.

Workshop Papers (Refereed)

- 25.** *A Visual Guide to MPI All-to-all.* Nick Netterville†, Ke Fan†, Sidharth Kumar, and **Thomas Gilray**. Workshop on Education for High Performance Computing. (EduHiPC) Dec 2022. <https://doi.org/10.1109/HiPCW57629.2022.00008>
- 26.** *Accelerating Datalog Applications with cuDF.* Ahmedur Rahman Shovon†, Landon Richard Dyken, Oded Green, **Thomas Gilray**, and Sidharth Kumar. Workshop on Irregular Applications: Architectures and Algorithms. (IA3) Nov 2022. <https://doi.org/10.1109/IA356718.2022.00012>
- 27.** *Exploring MPI Collective I/O and File-per-process I/O for Checkpointing a Logical Inference Task.* Ke Fan†, Kristopher Micinski, **Thomas Gilray**, and Sidharth Kumar. Workshop on High Performance Storage. (HPS) May 2021. <https://doi.org/10.1109/IPDPSW52791.2021.00153>
- 28.** *Symbolic Path Tracing to Find Android Permission-Use Triggers.* Kristopher Micinski, **Thomas Gilray**, Daniel Votipka, Jeffrey S. Foster, and Michelle L. Mazurek. Workshop on Binary Analysis Research. (BAR) Jan 2019. <https://doi.org/10.14722/bar.2019.23083>
- 29.** *Racets: Faceted Execution in Racket.* Kristopher Micinski, Zhanpeng Wang, and **Thomas Gilray**. Scheme Workshop. (SW) Sep 2018.
- 30.** *Toward Parallel CFA with Datalog, MPI, and CUDA.* **Thomas Gilray** and Sidharth Kumar. Scheme Workshop. (SW) Sep 2017.
- 31.** *A Linear Encoding of Pushdown Control-Flow Analysis.* Steven Lyde, **Thomas Gilray**, and Matthew Might. Scheme Workshop. (SW) Nov 2014.
- 32.** *Concrete and Abstract Interpretation: Better Together.* Maria Jenkins‡, Leif Andersen‡, **Thomas Gilray**, and Matthew Might. Scheme Workshop. (SW) Nov 2014.
- 33.** *Partitioning 0-CFA for the GPU.* **Thomas Gilray** and Matthew Might. Workshop on Functional and Constraint Logic Programming. (WFLP) Aug 2014.

34. *A Unified Approach to Polyvariance in Abstract Interpretations.* **Thomas Gilray** and Matthew Might. Scheme Workshop. (SW) Nov 2013.

35. *Sound and Precise Malware Analysis for Android via Pushdown Reachability and Entry-Point Saturation.* Shuying Liang, Andrew W. Keep, Matthew Might, David Van Horn, Steven Lyde, **Thomas Gilray**, and Petey Aldous. ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices. (SPSM) Nov 2013. <https://doi.org/10.1145/2516760.2516769>

Mentoring & Advising

PhD Advisees (As Committee Chair)

- 2025–present Henry Olson (PhD)
- 2024–present Akash Rao (PhD)
- 2023–present Sowmith Kunapaneni (PhD)
- 2021–present Ahmedur Rahman Shovon (PhD)—At UIC, co-advised with Sidharth Kumar.
- 2019–2025 Ke Fan (PhD)—Co-advised with Sidharth Kumar, now Asst. Prof. at Temple U.

MS Advisees (As Committee Chair)

- 2025–present Nathan Waltz (MS)
- 2024–present Hunter Smith (MS)
- 2022–2024 Ashraful Islam (MS)
- 2021–2023 Nick Netterville (MS)

MS/PhD Mentees (As Committee Member)

- 2024–present Tashi Stirewalt (PhD)
- 2020–present Yihao Sun (PhD)—At Syracuse University.
- 2019–2023 Arash Sahebolamri (PhD)—At Syracuse University.
- 2019–2023 Guimu Guo (PhD)—Now Assistant Professor at Rowan University.

Other Mentees

- 2025–present Sri Varsha Adavath (MS)
- 2024–present Ralph Lewis (BS)
- 2022–2024 Akshar Patel (MS)—Now a PhD student at U. Illinois Chicago (UIC).
- 2022–2024 Michael Gathara (MS)
- 2022–2024 John-Paul Robinson (PhD)
- 2018–2021 Clark Ren (MS)—Now at Amazon, inc.
- 2021 Laura Thompson (BS)
- 2019–2020 Adrian Gose (BS)
- 2016–2019 Phúc C. Nguyễn (PhD)—Now at Google, inc.
- 2016–2017 Javran Cheng (MS)—Now at Google, inc.
- 2014–2016 Guannan Wei (MS)—Now an Assistant Professor at Tufts University.
- 2011–2014 Leif Andersen (BS)—Now a PhD student at Northeastern University.

- 2013–2014 Maria Jenkins (BS)—Now at Pixio, inc.

Awarded Grant Proposals

- ARPA-H: HealthyDocs: Secure Interoperable Health Data Management. 2025-2029. (co-PI, in collaboration with the University of Colorado Barbara Davis Center, the University of Alabama at Birmingham Precision Medicine Institute, and Galois, inc.) (**Total: \$10M; My lab's share: \$1.29M**)
- NSF: PPOSS: Large: A Full-stack Approach to Declarative Analytics at Scale. 2023-2028. (Lead PI, in collaboration with Ohio State University, the University of Texas at Austin, the University of Alabama at Birmingham, the University of Illinois Chicago, and Syracuse University) (**Total: \$5M; My lab's share: \$1.05M**)
- NSF: PPOSS: Planning: A Full-stack Approach to Declarative Analytics at Scale. 2022-2023. (co-PI, with Syracuse University) (**Total: \$250,000; My lab's share: \$83,116**)
- DARPA: VSPELLS: Promotion to Optimal Languages Yielding Modular Operator-driven Replacements and Programmatic Hooks (POLYMORPH). 2021-2025. (PI, via subcontract with Galois, inc) (**Total: \$8M; My lab's share: \$400,000**)

Pending Grant Proposals

- DOE: SC ASCR: AI-Driven Autonomous Visualization and Data Management for Scientific Discovery on Supercomputers. 2025-2030. (Co-PI, in collaboration with Aric Hagberg, Pascal Grosset, George Stelle, and Terry Turton at LANL) (**Total: \$19.26M; My lab's share: \$749,999**)
- NSF: CICI: Architecting Resilient and Robust Cyberinfrastructure for Secure and Scalable Data Analytics in the Cloud. 2025-2028. (Co-PI, in collaboration with Xu Lin at WSU and Sidharth Kumar and Anrin Chakraborti at UIC) (**Total: \$1.2M; My lab's share: \$300,000**)

Grant Reviewing

- NSF Panelist, SHF CS2 Program, 2024
- NSF Panelist, PPOSS Program, 2024
- NSF Panelist, PPOSS Program, 2021

Journal Reviewing

- Transactions on Programming Languages and Systems (TOPLAS) 2023.
- Transactions on Computational Biology and Bioinformatics (TCBB) 2019.

Conference Program Committees

- International Conference on Software Engineering (ICSE) 2026.
- International Conference on Compiler Construction (CC) 2025.
- High-Performance Computing (HiPC) 2024.
- High-Performance Computing Student Research Symposium (HiPC SRS) 2023.
- Programming Language Design and Implementation Student Research Competition (PLDI SRC) 2022.
- Dynamic Languages Symposium (DLS) 2021.
- Symposium on Applied Computing (SAC) 2021.
- Principles of Programming Languages (POPL) 2020 (ERC).

- Scheme and Functional Programming Workshop (SW) 2019 (PC Chair).
- MiniKanren Workshop (MW) 2019.
- Principles of Programming Languages (POPL) 2019 (ERC).
- European Conference on Object-Oriented Programming (ECOOP) 2018 (AEC).
- Principles of Programming Languages (POPL) 2018 (ERC).

Departmental (Non-thesis) Committees

- Qualifying Exam Committee (Chair) for Anais Barja, WSU, 2025.
- Faculty Search Committee, WSU, 2024.
- Graduate Admissions Committee, UAB, 2023-2024.
- Qualifying Exam Committee for Guimu Guo, UAB, 2019.
- Curriculum Committee, UAB, 2018-2023.

Community & Outreach Activities

- The Datalog Modulo Theories Working Group, Organizer, 2025.
- WSU Senior Design Poster Competition Judge, 2025.
- UAB High-School Programming Contest (HSPC) Organizer, 2024.
- High-Performance Computing Student Research Symposium (HiPC SRS) Mentor and Judge, 2023.
- UAB High-School Programming Contest (HSPC) Organizer, 2023.
- ABC's Talk of Alabama (TV Panel Discussion): *Artificial Intelligence & Your World: What is it, and where is it headed?*, 2023.
- ABC's Talk of Alabama (TV Interview): *What is Artificial Intelligence?*, 2023.
- Taught three sessions of introductory programming to grades 6,7, and 9 at the *Diksha School* in India, while on travel to HiPC 2023.
- Programming Languages Mentoring Workshop (PLMW) Mentor and Panelist, "*Formal Methods: Theory and Practice*", 2022.
- UAB High-School Programming Contest (HSPC) Organizer, 2021.
- Organized and ran the Scheme and Functional Programming Workshop, 2019.
- Lead a student team in developing an automated grading system (autograde.org), 2019.
- UAB High-School Programming Contest (HSPC) Organizer, 2019.
- Faculty advisor to the UAB student ACM & ACM-W, 2018-2024.
- ACM and SIGPLAN member since before 2013.

Invited Talks

- From Theory to Throughput: Evolving Datalog’s Semantics and Performance. Minnowbrook Logic-programming Seminar. Syracuse University. 2025.
- Correctness and Verification using Software Contracts. Cyser Summer Workshop. Washington State University, 2025.
- Challenges in High-performance Deductive Programming. AP2S: Automated Program and Proof Synthesis. Vancouver, BC. AAAI Bridge. 2024.
- Formal Methods: Theory and Practice (*invited panel discussion*). Ljubljana, Slovenia. PLMW at ICFP 2022.
- Challenges Scaling Declarative Program Analysis. University of Illinois at Chicago. 2022.
- Declarative Program Analysis at Scale. Syracuse University. 2022.
- Tunable Abstract Abstract Machines. Workshop on Declarative Program Analysis (co-located with PLDI). 2019.
- Contracts for Correctness (today and tomorrow). Jet, inc. 2019.
- The Best of Both Worlds: Tunable, Correct-by-design Static Analysis. University of Alabama at Birmingham. 2018.
- Static Analysis with Introspective Polyvariance. Indiana University. 2016.
- Static Analysis with Introspective Polyvariance. University of Maryland. 2016.

Contributed (Refereed Paper) Talks

- Datalog with First-class Facts. London, UK. VLDB 2025.
- Load-balancing Parallel Relational Algebra. Frankfurt, Germany (remote). ISC 2020.
- Toward Parallel CFA with Datalog, MPI, and CUDA. Oxford, UK. SW 2017.
- Allocation Characterizes Polyvariance. Nara, Japan. ICFP 2016.
- Pushdown Control-Flow Analysis for Free. St. Petersburg, FL. POPL 2016.
- Partitioning 0-CFA for the GPU. Wittenberg, Germany. WFLP 2014.
- A Unified Approach to Polyvariance in Abstract Interpretations. Alexandria, VA, USA. SW 2013.
- A Survey of Polyvariance in Abstract Interpretations. Provo, UT, USA. TFP 2013.

An updated publication list, personal references, and other materials are available upon request.

Generated on June 4, 2025