Thomas Gilray

gilray.org **gilray@uab.edu**

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

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 implementing these analyses efficiently—and declaratively. I have also contributed to high-performance-computing techniques for accelerating data-parallel relational algebra and sparse linear algebra on supercomputers.

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$1LOC 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

Assistant Professor U. of Alabama, Birmingham 2018 - *Present*

Working on research into scalable, tunable program analysis, 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 Automata Theory, Programming Languages, and Artificial Intelligence.

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 R^{7}RS 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.

Teaching

(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, 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, both small-step and big-step interpreters, and a purely functional implementation of PageRank, among other unique projects.

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

I designed and taught an AI course that focuses on symbolic AI, logic, and automated reasoning (UAB has six other courses focusing on Machine Learning, which is outside my primary areas of expertise). We focus on problem solving by search, adversarial search and games, automated constraint solving, SAT solving, and logic. Students complete group projects focusing on A*, two-player Reversi, and Sudoku solving and complete an individual term project. As my own research in Datalog falls within the scope of this course, I spend at least a week near the end of the term focusing on logic programming and HornSAT/Datalog after the students have implemented DPLL.

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

I improved UAB's existing course on automata, formal languages, and the theory of computation by adding two substantial coding projects to supplement the class' theoretical focus. For example, 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 toy programming 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. The course covered standard intermediate representations, implementation of effects, exception handling, garbage collection, and first-class control. Every student managed to complete a working Scheme compiler by the end of the term.

Service

- External reviewer, TOPLAS 2023
- PC member, PLDI Student Research Competition (SRC) 2022
- Curriculum committee member 2018-2022, UAB CS Department.
- PC member, Dynamic Languages Symposium (DLS) 2021 co-located with SPLASH.
- Co-organizer, UAB HSPC 2019-2021 (High School Programming Contest).
- PC member, Symposium On Applied Computing (SAC) 2021.
- NSF Panelist, PPoSS Program, 2021.
- External reviewer, POPL 2020.
- PC chair, Scheme and Functional Programming Workshop, 2019.
- PC member, IEEE TCBBSI 2019.
- PC member, MiniKanren Workshop, 2019.
- Faculty advisor to the UAB ACM & ACM-W student chapter, 2018-present.
- Lead a student team in developing an automated grading system for UAB in 2019.
- External reviewer, POPL 2019.
- External reviewer, POPL 2018.
- Artifact evaluation committee, ECOOP 2018.
- Assisted reviewing for CC 2013, ICFP 2014, SAS 2015, PLDI 2016, and CCS 2017.

Awards & Recognition

- NSF: PPoSS: Large program collaborative award for $5,000,000 over five years (2023).
- NSF: PPoSS: Planning program collaborative award for $250,000 over one year to prepare for an NSF Large proposal to follow (2022).
- DARPA: VSPELLS program award for $400,000 over four years via Galois, inc. (2021).
- Won ISC Hans Meuer Best Paper award (2020).

- Featured by ALCF: collaboration with Sidharth Kumar advancing scalable MPI-based relational algebra was featured by ALCF's yearly *Science Report* magazine as a research highlight growing from our 2019 D.D. grant of hours on ALCF's Theta supercomputer. `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).
- DOE Directors Direction (D.D.) Grant of 2M hours on ALCF's Theta (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).

Mentoring & Advising

MS/PhD Advisees (As Committee Chair)

- 2023–present Sowmith Kunapaneni (PhD)
- 2022–present Akshar Patel (MS)
- 2022–present Michael Gathara (MS)
- 2022–present Nick Netterville (MS)
- 2022–present Ashraful Islam (PhD)
- 2019–2020 Kyle Headley (PhD)—Left for personal reasons at the start of the pandemic.

MS/PhD Advisees (As Committee Member)

- 2021–present Akmedur Rahman Shovon (PhD)—At University of Illinois Chicago.
- 2020–present Yihao Sun (PhD)—At Syracuse University.
- 2019–present Arash Sahebolamri (PhD)—At Syracuse University.
- 2019–present Ke Fan (PhD)—At University of Illinois Chicago.

Other Mentees

- 2022–present Landon Dyken (PhD)
- 2021 Laura Thompson (BS)
- 2018–2021 Clark Ren (MS)—Now at Amazon, inc.
- 2016–2019 Phúc C. Nguyễn (PhD)—Now at Google, inc.
- 2016–2018 David Darais (PhD)—Now at Galois, inc.
- 2016–2018 Kristopher Micinski (PhD)—Now a faculty member at Syracuse University.
- 2016–2017 Javran Cheng (MS)—Now at Google, inc.
- 2014–2016 Guannan Wei (MS)—Now a PhD student at Purdue University.
- 2013–2014 Maria Jenkins (BS)—Now at Pixio, inc.
- 2011–2014 Leif Andersen (BS)—Now a PhD student at Northeastern University.

Awarded Grant Proposals

- NSF: PPoSS: Large: A Full-stack Approach to Declarative Analytics at Scale. 2023-2028. (Lead PI, in collaboration with Sidharth Kumar, Kristopher Micinski, Swarat Chaudhuri, Suren Byna, Ananth Kalyanaraman, and Matthew Might) **(total: $5M; UAB's share: $2.63M)**
- NSF: PPoSS: Planning: A Full-stack Approach to Declarative Analytics at Scale. 2022-2023. (co-PI, with Sidharth Kumar and Kristopher Micinski) **(total: $250,000; UAB'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; UAB's share: $400,000)**
- DOE: Director's Discretion Program: Distributed Relational Algebra. 2019-*present*. (co-PI, with Sidharth Kumar) **(total: 2M compute hours for ALCF's Theta)**

Declined Grant Proposals

- ARAPA-H: HealthyDocs (abstract). 2023. (PI, via subcontract with Galois, inc).
- CAREER: Automating Declarative Program Analysis. 2022. (PI, declined with scores very good, very good/good, good, and good/fair, with quite encouraging feedback.)
- SHF: Small: Parallel Relational Algebra for Logical Inference at Scale. 2020. (co-PI, with Sidharth Kumar)
- SHF: Small: Parallel Relational Algebra for Logical Inference at Scale. 2019. (co-PI, with Sidharth Kumar)
- DOE INCITE 2019. Request for 14,000,000 compute hours. (co-PI, with Sidharth Kumar)

Doctoral Theses

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

Journal Papers

2. *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.
Invited to the Journal of Functional Programming.

Conference Papers

3. *Communication-Avoiding Recursive Aggregation.* Yihao Sun, Sidharth Kumar, **Thomas Gilray**, and Kristopher Micinski. IEEE International Conference on Cluster Computing.
(CLUSTER) Nov 2023.

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

5. *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.

6. *Seamless Deductive Inference Via Macros.* Arash Sahebolamri, **Thomas Gilray**, and Kristopher Micinski. International Conference on Compiler Construction.
(CC) Apr 2022.

7. *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.

8. *Compiling Data-parallel Datalog.* **Thomas Gilray**, Sidharth Kumar, and Kristopher Micinski. International Conference on Compiler Construction.
(CC) Mar 2021.

9. *Load-balancing Parallel Relational Algebra.* Sidharth Kumar and **Thomas Gilray**. ISC High Performance.
(ISC—31% acceptance) Jun 2020.
Won ISC Hans Meuer Best Paper award.

10. *Abstracting Faceted Execution.* Kristopher Micinski, David Darais, and **Thomas Gilray**. IEEE Computer Security Foundations Symposium.
(CSF) Jun 2020.

11. *Distributed Relational Algebra at Scale.* Sidharth Kumar and **Thomas Gilray**. International Conference on High Performance Computing, Data, and Analytics.
(HiPC—23% acceptance) Dec 2019.
Won HiPC Best Paper award.

12. *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.
Invited to the Journal of Functional Programming.

13. *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.

14. *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.

15. *Allocation Characterizes Polyvariance.* **Thomas Gilray**, Michael D. Adams, and Matthew Might. International Conference on Functional Programming.
(ICFP—31% acceptance) Sep 2016.

16. *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.

17. *Dynamic Sparse-Matrix Allocation on GPUs.* James King, **Thomas Gilray**, Robert M. Kirby, and Matthew Might. ISC High Performance.
(ISC) 2016.
Won PRACE ISC best paper award.

18. *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

19. *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.

20. *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.

21. *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.

22. *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.

23. *Racets: Faceted Execution in Racket.* Kristopher Micinski, Zhanpeng Wang, and **Thomas Gilray**. Scheme Workshop.
(SW) Sep 2018.

24. *Toward Parallel CFA with Datalog, MPI, and CUDA.* **Thomas Gilray** and Sidharth Kumar. Scheme Workshop.
(SW) Sep 2017.

25. *A Linear Encoding of Pushdown Control-Flow Analysis.* Steven Lyde, **Thomas Gilray**, and Matthew Might. Scheme Workshop.
(SW) Nov 2014.

26. *Concrete and Abstract Interpretation: Better Together.* Maria Jenkins, Leif Andersen, **Thomas Gilray**, and Matthew Might. Scheme Workshop.
(SW) Nov 2014.

27. *Partitioning 0-CFA for the GPU.* **Thomas Gilray** and Matthew Might. Workshop on Functional and Constraint Logic Programming.
(WFLP) Aug 2014.

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

29. *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.

Invited Talks

- 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.
- 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 Talks

- 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.