Thomas Gilray

gilray.org **thomas.gilray@gmail.com** **+1 (541) 740-7503**

Education

University of Utah Ph.D. 2016 (expected Summer 2016)

Developed a unified methodology for polyvariant (*e.g.*, context sensitive) static analysis and applied the approach to produce a specific adaptive style of polyvariance, obtaining a form of perfect precision at no asymptotic cost to analysis complexity.

The technique avoids all return-flow conflation of values (a long-standing problem for control-flow analyses), guaranteeing perfect stack precision and at no cost in terms of human labor to implement, and both average and worst-case model complexity. Proved that the precision is equal to an incomputable analysis with an unbounded stack and mechanically verified the proof using the Coq system. My dissertation is titled *"Introspective Polyvariance for Control-Flow Analyses"*.

University of Utah M.S. 2012

Completed coursework toward an M.S. in Spring, 2012 and applied to the Ph.D. program to do research in Prof. Matt Might's U-Combinator lab. My GPA at Utah was 3.90/4.

University of Oregon B.S. 2010

Received a B.S. in Computer Science, Spring 2010, with a minor in Business Administration. Focused undergraduate studies on programming languages, interpreters, compilers, and data structures.

Professional Experience

Intern (static analysis) HP, inc. (Fortify) Summers 2013 & 2014

Implemented an inter-procedural must-alias analysis to supplement existing analyses in the Fortify source-code analyzer. This was based on the theory of abstract counting, a cardinality analysis which attempts to place an upper bound on the number of concrete addresses an abstract address could be representing in a given program analysis. Successfully addressed various bug reports and performed an architectural review for a body of preprocessing code.

Research Assistant University of Utah 2011 - 2016

Studied static analysis (particularly for functional languages) using abstract interpretation. Conducted research, with a variety of collaborators, studying polyvariant program analysis, implementing analyses as operational semantics (interpreters) and as linear algebra, and designing a new dynamic sparse-matrix format for SIMD hardware (*e.g.*, GPUs). Participated in a lab-wide effort on the timed challenge events for DARPA's APAC program. Each of these involved using our lab's static analysis tools to assist a collaborative effort identifing hidden malware deliberately planted in a series of android applications.

Teaching Assistant University of Utah 2010 & 2012

Assisted teaching for sophomore *"Discrete Structures"*. Responsibilities included grading, holding office hours, and teaching three back-to-back review sections on fridays. These covered topics being taught in the main lectures including basic first introductions to relations, functions, graph theory, counting problems, transfinite sets, propositional logic, and first-order logic. Also assisted teaching for graduate *"Advanced Compilers"*, helping to design assignments (including open and closed test suites), grading, and holding office hours to help students write their compiler passes.

Backend Developer VizMe, inc. 2009 - Summer 2010

Worked for a startup co-founded by Prof. Eric Wills (from the University of Oregon). Setup a Solr-based search engine in Java and wrote Python scripts to manage MySQL databases, translation spreadsheets for site-localization, and simple image analyses, among other projects.

Web Developer (*etc.*) Various 2005 - 2009

Worked for more than a dozen clients (some repeatedly) to produce website designs, web-based tools, content management systems (CMS), and website backends. Worked with (X)HTML, JavaScript, CSS, PHP, Python, MySQL, Apache, Debian Servers, and used/customized frameworks such as MediaWiki, Wordpress, DOMAssistant, and JQuery to fit clients' needs. Spent two summers working for Prof. Ostroverkhova's OPE research group at Oregon State University and developed browser-based educational tools including an online CMS (supporting LaTeX) particularly suited for graduate students in physics. Before college, I did a summer internship for HP producing a flash and html tutorial showing how to import and render Autodesk Inventor's models within 3ds Max.

Extracurricular Projects

Taught myself C++, implementing a variety of small applications and visualizations, such as an equation plotter (non-function grapher). I learned fundamentals of programming and computer science by implementing a wide variety of basic libraries and data structures. These included AVL trees, red-black trees (including deletion), hash array mapped tries, splay trees, dynamically reallocating arrays, and templated strings supporting ASCII, unicode, and fast UTF-8 encoding/decoding. Implemented recursive descent parsers (and played around with parser generators), producing multiple toy interpreters. These culminated in a long term project (as a near-terminal undergraduate) to produce a language incorporating correct implementations of various language features learned in school such as objects, inheritance, and closures. Though this project was far from a perfect attempt, the challenges it presented (and the fun I had) spurred me to apply to graduate school.

Collaborated with two friends to produce a social networking website for artists which allowed them to upload and share their work. The website made heavy use of `XMLHttpRequest` and included features like groups, comments, favorites, and user profiles, with few page refreshes required. The website was run (exclusively not for profit) for several years and at one point was home to over 1000 active users and nearly 10,000 works of art.

Selected Publications

Thomas Gilray, Steven Lyde, Michael D. Adams, Matthew Might, David Van Horn.
Pushdown Control-Flow Analysis for Free. SIGPLAN, Principles of Programming Languages. 2016.

Thomas Gilray, Michael D. Adams, Matthew Might.
Allocation Characterizes Polyvariance. (In submission to) SIGPLAN, International Conference on Functional Programming. 2016.

James King, **Thomas Gilray**, Robert M. Kirby, Matthew Might.
Dynamic Sparse-Matrix Allocation on GPUs. International Supercomputing Conference. 2016.
(Winner of the **PRACE-ISC best paper** award.)

Thomas Gilray, James King, Matthew Might.
Partitioning 0-CFA for the GPU. International Workshop on Functional and Constraint Logic Programming. 2014.

Maria Jenkins, Leif Andersen, **Thomas Gilray**, Matthew Might.
Concrete and Abstract Interpretation: Better Together. The Scheme Workshop. 2014.

Thomas Gilray, Matthew Might.
A Unified Approach to Polyvariance in Abstract Interpretations. The Scheme Workshop. 2013.

Thomas Gilray, Matthew Might.
A Survey of Polyvariance in Abstract Interpretations. Trends in Functional Programming. 2013.
(Winner of the **best student paper** award.)

A full publication list, personal references, and other materials are available upon request.