

Curriculum Vitae

Paul Gazzillo

paul.gazzillo 'at' yale.edu

Education

- 2016 PhD Computer Science, Courant Institute, New York University, New York, NY
Advisors: Robert Grimm, Thomas Wies
Thesis: Analyzing source code across static conditionals
- 2011 MS Computer Science, Courant Institute, New York University, New York, NY
Advisor: Robert Grimm
Thesis: Configuration-Preserving C Parsing
- 2003 BS Computer Science and Mathematics, Rutgers University, New Brunswick, NJ

Academic Positions

- 2015-Now Post-Doctoral Associate, Yale University, New Haven, CT
Advisor: Eric Koskinen
- 2011-2013 Research Assistant, New York University, New York, NY
Advisor: Robert Grimm
- 2010 Junior Research Scientist (Summer), New York University, New York, NY
Advisor: Robert Grimm

Industry Experience

- 2013 Software Engineering Intern (Summer), Google, Mountain View, CA
- 2009 Financial Software Development Intern (Summer), Bloomberg LP, New York, NY
- 2004-2008 Research Data Analyst, Educational Testing Service, Princeton, NJ
- 2000-2004 Programmer (Summers & Part-Time), Educational Testing Service, Princeton, NJ

Fellowships and Honors

- 2013-2015 MacCracken Fellowship, New York University
- Spr 2013 Global Research Initiative Fellowship, New York University Shanghai
- Oct 2012 SIGPLAN Research Highlight (for the PLDI 2012 paper on SuperC)
- May 2012 Matthew Smosna Prize, Courant Institute, New York University
- May 2005 Outstanding Contributor Award, Educational Testing Service

Drafts

- 2017 “Brief Announcement: A Transactional Universal Construction for Shared Objects” by Paul Gazzillo. Under Submission. May 2017.

Refereed Conference Proceedings

- ESEC/FSE 2017 “MAC: Makefile Analysis for Configurations” by Paul Gazzillo. To appear: Proceedings of the European Software Engineering Conference and ACM SIGSOFT

Symposium on the Foundations of Software Engineering.

Acceptance Rate: 24.4% (72 out of 295 submissions)

PODC 2017 “Adding Concurrency to Smart Contracts” by Thomas Dickerson, Paul Gazzillo, Maurice Herlihy, and Eric Koskinen. To appear: Proceedings of the 2017 ACM Symposium on Principles of Distributed Computing.

PODC 2017 “Brief Announcement: Proust: A Design Space for Highly-Concurrent Transactional Data Structures” by Thomas Dickerson, Paul Gazzillo, Maurice Herlihy, and Eric Koskinen. To appear: Proceedings of the 2017 ACM Symposium on Principles of Distributed Computing.

PLDI 2017 “Decomposition Instead of Self-Composition for Proving the Absence of Timing Channels” by Timos Antonopoulos, Paul Gazzillo, Michael Hicks, Eric Koskinen, Tachio Terauchi, and Shiyi Wei. Proceedings of the ACM SIGPLAN 2017 Conference on Programming Language Design and Implementation.

Acceptance Rate: 14.6% (47 out of 322 submissions)

PLDI 2012 “SuperC: Parsing all of C by taming the preprocessor” by Paul Gazzillo and Robert Grimm. Proceedings of the ACM SIGPLAN 2012 Conference on Programming Language Design and Implementation, pp. 323-334, June 2012.

Acceptance Rate: 18.8% (48 out of 255 submissions)

SIGPLAN Research Highlights Paper

Technical Reports

2016 “Vertical Composition of Reversible Atomic Objects” by Timos Antonopoulos, Paul Gazzillo, Eric Koskinen, and Zhong Shao. Technical Report YALEU/DCS/TR1529, Computer Science Department, Yale University, 2016

2015 “Kmax: Analyzing the Linux Build System” by Paul Gazzillo. Technical Report TR2015-976, Computer Science Department, New York University, 2015.

2011 “Parsing all of C by taming the preprocessor” Paul Gazzillo and Robert Grimm. Technical Report TR2011-939, Computer Science Department, New York University, 2011.

Patents

2017 Adding Concurrency to Smart Contracts (patent pending)

2014 Configuration-Preserving Preprocessor, 9,389,842, issued 2016

2013 Configuration-Preserving Preprocessor and Configuration-Preserving Parser, 8,806,456, issued 2014

2008 Data Structure for Defining a Chart, application 2008/0086679

2008 Method and System for XML Multi-Transform, 9,189,464, issued 2015

Dissertation

2016 “Analyzing source code across static conditionals” by Paul Gazzillo. PhD Thesis, Department of Computer Science, New York University.

Talks

MAC: Makefile Analysis for Configurations

Sep 2017 European Software Engineering Conference and Foundations of Software Engineering (ESEC/FSE).

Decomposition Instead of Self-Composition for Proving the Absence of Timing Channels

Jun 2017 Programming Language Design and Implementation (PLDI), Universitat Politècnica de Catalunya, Barcelona, Spain

Jun 2017 University of Maryland, College Park, MD

Jun 2017 New England Programming Languages and Systems Symposium (NEPLS), University of Massachusetts, Lowell, MA

Adding Concurrency to Smart Contracts

Apr 2017 Shanghai Jiao Tong University, Shanghai, China

Apr 2017 Yale University, New Haven, CT

Topic: Enabling Variability-Aware Software Tools (SuperC and Kmax)

Mar 2017 Feature-Oriented Software Development Conference (FOSD), Technische Universität Darmstadt, Darmstadt, Germany

Sep 2016 NJ Programming Languages and Systems Seminar (NJPLS), Rutgers University, NJ

Jan 2016 Carnegie Mellon University Institute for Software Research, Pittsburgh, PA

Jan 2016 Yale University, New Haven, CT

Nov 2015 IBM Programming Languages Day, Yorktown Heights, NY

Parsing All of C by Taming the Preprocessor

Jan 2017 Eurosys Shadow Program Committee Workshop, Zurich, Switzerland

Jun 2012 Programming Language Design and Implementation (PLDI), Beijing, China

GraphicML: A Markup Language for Describing Charts

May 2006 John W. Tukey Seminar on Data Preparation and Presentation, ETS, Princeton, NJ

Poster Sessions

2017 A High-Performance Engine for Smart Contracts, Yale Innovation Summit, New Haven, CT

2016 Partial Makefile Evaluation for Analyzing Build Systems, Yale University, New Haven, CT

2016 Automatic Bound Computation, Yale University, New Haven, CT

2012 SuperC: Parsing all of C by taming the preprocessor, PLDI 2012, Beijing, China

Software

Concurrent smart contracts We created a prototype implementation with benchmarks (PODC 2017).

Blazer We created and implemented static analyses for finding complexity and side-channel attacks (PLDI 2017).

RAO I built a prototype implementation of the transactional universal construction for Reversible Atomic Objects (Yale TR 2016).

Kmax I designed and built a Makefile abstract interpreter for analyzing configurations of the Linux build system (NYU TR 2015, ESEC/FSE 2017).

Courgette I contributed to Google Chrome's unique compression algorithm for enabling smaller software updates (Google Internship 2013).

SuperC I designed and built a framework for configuration-preserving preprocessing and parsing with an implementation for C (PLDI 2012).

GraphicML I designed an intermediate language for data graphics and built a translation tool to generate charts as vector graphics (ETS 2000-2008).

NAEP Questions Tool I was the lead developer and version 2 architect (ETS 2000-2008).

Professional Activities

- 2017 Principles of Programming Languages Artifact Evaluation Committee (POPL 2018 AEC)
- 2017 Proposal Reviewer, Netherlands Organisation for Scientific Research (NWO)
- 2017 Shadow Program Committee Member (Heavy), ACM European Conference on Computer Systems (EuroSys)
- 2016 Journal Reviewer, ACM Transactions on Parallel Computing (TOPC)
- 2013 PhD Student Representative, Computer Science Department, New York University
- 2010 President, NYU Master's Association of Computer Science
- 2009 Treasurer, NYU Master's Association of Computer Science

Teaching Experience

- Spr 2015 Recitation Leader, Data Structures, New York University
- Fall 2014 Recitation Leader, Data Structures, New York University
- 2009-2012 Teacher, cSplash one-day festival of Math and CS, New York University, NY
- Spr 2010 Teaching Assistant, Compilers, New York University
- Spr 2010 Teaching Assistant, Operating Systems, New York University
- Fall 2009 Teaching Assistant, Computer Organization, New York University