Matthew A. Hammer

Assistant Professor

Computer Science Department
University of Colorado
1111 Engineering Dr, Boulder, CO 80309

Email: Matthew.Hammer@Colorado.Edu
Homepage: http://matthewhammer.org

Education

PhD (Computer Science)	University of Chicago	2012
MS (Computer Science)	Toyota Technological Institute at Chicago	2007
BS (Computer Science)	University of Wisconsin	2005

Publications

Toward a Semantics for Program Editors
Cyrus Omar, Ian Voysey, Michael Hilton, Joshua Sunshine, Claire Le Goues, Jonathan Aldrich, Matthew A. Hammer.
The 2nd Summit on Advances in Programming Languages (**SNAPL 2017**).
Monterey, California. May 2017.

Hazelnut: A Bidirectionally Typed Structure Editor Calculus
Cyrus Omar, Ian Voysey, Michael Hilton, Jonathan Aldrich, Matthew A. Hammer.
Principles of Programming Languages (**POPL 2017**).
Paris, France. January 2017.
(Acceptance Rate: 27%)

A Vision for Online Verification-Validation
Matthew A. Hammer, Bor-Yuh Evan Chang, David Van Horn
Generative Programming: Concepts & Experience (**GPCE 2016**).
Amsterdam, Netherlands. October 2016.
(Acceptance Rate: 32%)

The Random Access Zipper: Simple, Purely-Functional Sequences
Kyle Headley, Matthew A. Hammer.
Trends in Functional Programming (**TFP 2016**).
College Park, Maryland. June 2016.

Incremental Computation with Names
Matthew A. Hammer, Joshua Dunfield, Kyle Headley, Nicholas Labich, Jeffrey S. Foster and Michael Hicks.
Object-Oriented Programming, Systems, Languages, and Applications (**OOPSLA 2015**).
Pittsburgh, USA. October 2015.
(Acceptance Rate: 25%)

ADAPTON*: Composable, Demand-driven Incremental Computation*
Matthew A. Hammer, Yit Phang Khoo, Michael Hicks and Jeffrey S. Foster.
Programming Language Design and Implementation (**PLDI 2014**).
Edinburgh, Scotland. June 2014.
(Acceptance Rate: 20%)

WYSTERIA*: A Programming Language for Generic, Mixed-Mode Multiparty Computations*
Aseem Rastogi, Matthew A. Hammer and Michael Hicks.
35th IEEE Symposium on Security and Privacy (**IEEE S&P 2014**)
San Jose, California USA. May 2014.
(Acceptance Rate: 13.6%)

Implicit Self-Adjusting Computation for Purely Functional Programs
Yan Chen, Joshua Dunfield, Matthew A. Hammer and Umut A. Acar.
Journal of Functional Programming 2014 (**JFP 2014**).

Knowledge Inference for Optimizing Secure Multi-party Computation
Aseem Rastogi, Piotr Mardziel, Matthew A. Hammer and Michael Hicks.
Programming Languages and Analysis for Security (**PLAS 2013**).
Seattle, Washington USA. June 2013.

Self-Adjusting Stack Machines
Matthew A. Hammer, Georg Neis, Yan Chen and Umut A. Acar
Object-Oriented Programming, Systems, Languages, and Applications (**OOPSLA 2011**).
Portland, Oregon USA. October 2011.
(Acceptance Rate: 23%)

Implicit Self-Adjusting Computation for Purely Functional Programs
Yan Chen, Joshua Dunfield, Matthew A. Hammer and Umut A. Acar
International Conference on Functional Programming (**ICFP 2011**).
Tokyo, Japan. September 2011
(Acceptance Rate: 31%)

CEAL*: A C-Based Language for Self-Adjusting Computation*
Matthew A. Hammer, Umut A. Acar and Yan Chen.
Programming Language Design and Implementation (**PLDI 2009**).
Dublin, Ireland. June 2009.
(Acceptance Rate: 20%)

Memory Management for Self-Adjusting Computation
Matthew A. Hammer and Umut A. Acar.
International Symposium on Memory Management (**ISMM 2008**).
Tuscon, Arizona. June 2008.
(Acceptance Rate: 43%)

A Proposal for Parallel Self-Adjusting Computation
Matthew Hammer, Umut A. Acar, Mohan Rajagopalan, Anwar Ghuloum
Workshop on Declarative Aspects of Multicore Programming (**DAMP 2007**).
Nice, France. January 2007.

Running Quake II on a grid
G. Deen, M. Hammer, J. Bethencourt, I. Eiron, J. Thomas, and J. H. Kaufman.
IBM Systems Journal 2006.

Theses

Self-Adjusting Machines
University of Chicago, December 2012.
Committee:

- John Reppy (*Chair*)
- Umut A. Acar (*PhD Advisor*)
- David MacQueen
- Rupak Majumdar

Patents

Distributing and geographically load balancing location aware communication device client-proxy applications
Viktors Berstis, John Bethencourt, Kevin Damm, Glenn Deen, Matthew A. Hammer, James H Kaufman, Toby Lehman
US Patent 7,702,784

Handling of players and objects in massive multi-player on-line games
Viktors Berstis, John Bethencourt, Kevin Damm, Glenn Deen, Matthew A. Hammer, James H Kaufman, Toby Lehman
US Patent 8,057,307

Concurrent Management of Adaptive Programs
Matthew Hammer, Mohan Rajagopalan, Anwar Ghuloum
US Patent App. 11/750,441

Funding

NSF Small: Online Verification-Validation ($310k to CU Boulder)

Mozilla Research Funding (unrestricted gift, $90k)

Students

Kyle Headley (PhD program, CU Boulder)

Jared Wright (PhD program, CU Boulder)

Monal Narasimhamurthy (PhD program, CU Boulder)

Service

Thesis Committees: Max Russek (CU Undergrad; Spring 2016)

SRC Judge: PLDI 2016 Student Research Competition.

Program Committee (PC) member: GPCE 2017, ESOP 2017, PLAS 2015

External Review Committee (ERC) member: PLDI 2015

External reviewer: ESOP 2017, ESOP 2016, IEEE S&P 2015, POPL 2015, OOPSLA 2014, PLAS 2014, SOFSEM 2014, PLDI 2013, POPL 2012, ICFP 2010, ML Workshop 2009, PLDI 2008.

Graduate Student Representative. May 2010–October 2011.
Max Planck Institute for Software Systems.

Teaching

CSCI 7000: Programming languages for incremental computing
University of Colorado, Boulder. Spring 2017.

CSCI 5535: Foundations of Programming Languages
University of Colorado, Boulder. Fall 2016.

CSCI 7000: Programming language design for interfaction
University of Colorado, Boulder. Spring 2016.

CSCI 5535: Foundations of Programming Languages
University of Colorado, Boulder. Fall 2015.

CMSC 631: *Program Analysis and Understanding*.
University of Maryland, College Park. Spring 2013.
Co-instructed with Michael Hicks, Jeffrey S. Foster and Stevie Strickland.

Teaching assistant for CMCS 336: *Type Systems for Programming Languages*.
Toyota Technological Institute / University of Chicago. Winter 2008.
Instructors: Umut Acar and Amal Ahmed.

Talks

A Vision for Online Verification-Validation
Matthew A. Hammer, Bor-Yuh Evan Chang, David Van Horn
Generative Programming: Concepts & Experience (**GPCE 2016**).
Amsterdam, Netherlands. October 2016.

Incremental Computation with Names
Matthew A. Hammer, Joshua Dunfield, Kyle Headley, Nicholas Labich, Jeffrey S. Foster and Michael Hicks.
Object-Oriented Programming, Systems, Languages, and Applications (**OOPSLA 2015**).
Pittsburgh, USA. October 2015.

WYSTERIA*: A Programming Language for Generic, Mixed-Mode Multiparty Computations*
Dagstuhl seminar 14492: The synergy between programming languages and cryptography.
Schloss Dagstuhl. Wadern, Germany. December 2014.

ADAPTON*: Composable, Demand-driven Incremental Computation*
Programming Language Design and Implementation (**PLDI 2014**).
Edinbugh, Scotland. June 2014.

Self-Adjusting Stack Machines
Object-Oriented Programming, Systems, Languages, and Applications (**OOPSLA 2011**).
Portland, Oregon USA. October 2011.

Self-Adjusting Stack Machines and the CEAL Compiler
Invited talk. Max Planck Institute for Software Systems advisory board visit day.
Frankenstein, Rhineland-Palatinate Germany. May 2011.

A Compilation Framework for Self-Adjusting Computation
Dissertation proposal.
Chicago, Illinois USA. December 2010.

CEAL*: A C-Based Language for Self-Adjusting Computation*
Programming Language Design and Implementation (**PLDI 2009**).
Dublin, Ireland. June 2009.

Memory Management for Self-Adjusting Computation,
International Symposium on Memory Management (**ISMM 2008**).
Tuscon, Arizona. June 2008.

A Proposal for Parallel Self-Adjusting Computation,
Workshop on Declarative Aspects of Multicore Programming (**DAMP 2007**).
Nice, France. January 2007.

Software

ADAPTON: Composable, Demand-Driven Incremental Computation. ADAPTON provides library primitives

(currently in OCaml and Rust, and previously, in Python) for creating incremental computation (IC). Unlike prior approaches, ADAPTON supports demand-driven IC (e.g., computations that use laziness).

WYSTERIA: A Programming Language for Generic, Mixed-mode Multiparty Computation. WYSTERIA is a high-level functional programming language for writing mixed-mode secure computations. Such computations interleave local, private computations with secure multiparty computations.

CEAL: A C-based language (compiler and run-time system) for self-adjusting computation. CEAL extends C with a small set of primitives that allow programmers to write self-adjusting computations in a manner similar to conventional C programming.

Student Internships

Intel, Programming Systems Lab at Santa Clara (June 2007–September 2007)
Graduate Research Intern

Intel, Programming Systems Lab at Santa Clara (June 2006–September 2006)
Graduate Research Intern

IBM, Almaden Research Center (May 2005–September 2005)
Research Intern

IBM, Almaden Research Center (May 2004–August 2004)
Research Intern

IBM, Extreme Blue Program (June 2003–August 2003)
Computer Science Intern

Last updated: April 5, 2017