Bor-Yuh Evan Chang

Curriculum Vitæ

bec@cs.colorado.edu
http://www.cs.colorado.edu/~bec/
+1 (303) 492-8894 (phone)
+1 (303) 492-2844 (fax)

Department of Computer Science
University of Colorado Boulder
430 UCB
Boulder, CO 80309-0430 USA

EDUCATION

PhD **University of California, Berkeley**, Computer Science 2008
Advisor: Prof. George C. Necula

Thesis: ***End-User Program Analysis***
Dissertation Committee: Prof. George C. Necula (chair), Prof. Koushik Sen, and Prof. Jack Silver

MS **University of California, Berkeley**, Computer Science 2005
Advisor: Prof. George C. Necula

Thesis: ***Type-Based Verification of Assembly Language***

BS **Carnegie Mellon University**, Computer Science, 4.0 GPA 2002
University and College Honors
Minors: Biological Science and Mathematical Science
Advisors: Prof. Robert Harper and Prof. Frank Pfenning

Thesis: ***Iktara in ConCert: Realizing a Certified Grid Computing Framework from a Programmer's Perspective***

ACADEMIC APPOINTMENTS

University of Colorado Boulder January 2009–present
Assistant Professor, Department of Computer Science
Assistant Professor, Department of Electrical, Computer, and Energy Engineering (by courtesy).

University of Maryland, College Park September 2008–November 2008
Postdoctoral Researcher, Department of Computer Science
Advisor: Prof. Jeffrey S. Foster

RESEARCH INTERESTS

Algorithms, tools, and techniques for building, understanding, and ensuring reliable computational systems.

Keywords: software quality, programmer productivity, program analysis, automated reasoning, programming languages, semantics, logic, formal methods.

AWARDS AND HONORS

Professional

National Science Foundation CAREER Award December 2010

Graduate

College of Engineering Graduate Student Prize   December 2008
    University of California, Berkeley

Siebel Scholar, offered but declined because of graduation   August 2008
    University of California, Berkeley

National Science Foundation Graduate Research Fellowship   2004–2007

California Microelectronics Fellowship   2002–2003

Undergraduate

Phi Kappa Phi Honor Society, inducted   May 2002

Andrew Carnegie Society Presidential Scholar, selected   December 2001

Phi Beta Kappa Honor Society, inducted   October 2001

Lambda Sigma Honor Society, inducted   September 1999

Carnegie Mellon University Presidential and Institutional Scholarships   1998–2002

Award Papers

ECOOP 2013: Distinguished Artifact Award   July 2013
    QUIC Graphs: Relational Invariant Generation for Containers

TACAS 2012: Journal Special Issue Invitation   October 2012
    A Bit Too Precise? Bounded Verification of Quantized Digital Filters

GRANTS

I have been a PI or co-PI on sponsored research totaling in awards of approximately $9.4M of which $2.3M have been led by me as PI. My portion of these awards has totaled approximately $2.8M. A summary of these figures is given below:

Total Awarded:	$9,422,144
Awarded for CU Boulder:	$5,809,166
Awarded, my portion, approximate:	$2,763,879
Awarded as PI:	$2,308,927

NSF CCF-1619282, "SHF: Small: Collaborative Research: Online Verification-Validation," $449,999 with $309,990 for CU Boulder, 09/01/2016–08/31/2019, Matthew Hammer (PI); Bor-Yuh Evan Chang (co-PI); David Van Horn (PI at University of Maryland, College Park).

DARPA FA8750-15-2-0096, "STAC: Auditr: Securing Space/Time Defenses in Java Bytecode," $5,797,751 with $2,891,227 for CU Boulder, 04/22/2015–04/11/2019, Pavol Cerny (PI); John Black, Bor-Yuh Evan Chang, and Sriram Sankaranarayanan (co-PIs); Isil Dillig (PI at the University of Texas at Austin); Marijn Heule (co-PI); Henny Sipma (PI at Kestrel Technology).

DARPA FA8750-14-2-0263, "MUSE: Mining and Understanding Bug Fixes to Address Application-Framework Protocol Defects," $1,599,343, 09/24/2014–09/13/2018, Bor-Yuh Evan Chang (PI); Kenneth M. Anderson, Pavol Cerny, Sriram Sankaranarayanan, and Tom Yeh (co-PIs).

DARPA via Maryland Z8140001, "APAC: Specialized Binary Analysis for Vetting Android Apps Using GUI Logic," $733,734 with $202,774 for CU Boulder, 09/01/2013–10/31/2015 subcontract from University of Maryland, College Park, Atif Memon (PI at University of Maryland, College Park); Tom Yeh (PI at CU Boulder) and Bor-Yuh Evan Chang (co-PI).

NSF CCF-1218208, "SHF: Small: Modular Reflection," $250,000, 10/01/2012–09/30/2015, Bor-Yuh Evan Chang (PI); Jeremy G. Siek (co-PI).

NSF CCF-1055066, "CAREER: Cooperative Program Analysis: Bridging the Gap Between User and Tool Reasoning," $459,584, 06/01/2011–05/31/2017, Bor-Yuh Evan Chang (PI).

Joint Institute for Strategic Energy Analysis (JISEA) UGA-0-41026-04, "Verifiable Decision-Making Algorithms for Reconfiguration of Electric Microgrids," $49,985 with $14,500 for CU Boulder, 07/01/2010–06/30/2011, Siddharth Suryanarayanan (PI at Colorado State University); Sriram Sankaranarayanan (PI at CU Boulder); Bor-Yuh Evan Chang and Dirk Grunwald (co-PIs). JISEA is affiliated with the National Renewable Energy Laboratory (NREL).

NSF CCF-0939991, "EAGER: Exploratory Research on Gradual Programming," $81,748, 08/01/2009–07/31/2010, Jeremy G. Siek (PI); Bor-Yuh Evan Chang and Amer Diwan (co-PIs).

ADVISING

Graduated PhD Dissertation Advisees

1. Sam Blackshear, Fall 2010–Spring 2015, defended May 27, 2015. Computer Science. University of Colorado Boulder *Flexible Goal-Directed Abstraction*.

2. Devin Coughlin, Fall 2008–Spring 2015, defended January 7, 2015. *Type-Intertwined Separation Logic*. Computer Science. University of Colorado Boulder.

3. Arlen Cox, co-advised with Xavier Rival (ENS Paris) and Sriram Sankaranarayanan, Fall 2008–Fall 2014, defended November 17, 2014. *Parametric Heap Abstraction for Dynamic Language Libraries*. Electrical, Computer, and Energy Engineering. University of Colorado Boulder.

Graduated Senior Thesis Advisees

1. Evan Roncevich (BS 2016). Senior Thesis, Fall 2015–Spring 2016, defended May 6, 2016. Discovery Learning Apprentice, Fall 2014–Spring 2015. Computer Science. University of Colorado Boulder.

2. Ross Holland (BS/MS 2016). Senior Thesis, Fall 2014–Spring 2015, defended April 30, 2015. Computer Science. University of Colorado Boulder.

3. Nick Vanderweit (BS 2014). Senior Thesis, Fall 2013–Spring 2014, defended May 7, 2014. Computer Science. University of Colorado Boulder.

4. Alexander Beal (BS 2013). Senior Thesis, Fall 2012–Spring 2013, defended May 9, 2013. Computer Science. University of Colorado Boulder.

Current Postdoctoral Advisees

1. Edmund S.L. Lam, Summer 2016–present. Computer Science. University of Colorado Boulder.

2. Sergio Mover, Fall 2015–present. Computer Science. University of Colorado Boulder.

Current PhD Dissertation Advisees

1. Benno Stein, Fall 2015–present. Computer Science. University of Colorado Boulder.

2. Tianhan Lu, Fall 2015–present. Computer Science. University of Colorado Boulder.

3. Shawn Meier, Fall 2014–present. Computer Science. University of Colorado Boulder.

Current Senior Thesis Advisees

1. Maxwell Russek (BS), Fall 2014–present. Senior Thesis. Computer Science. University of Colorado Boulder.

Past Research Advisees

I have advised or co-advised independent studies or semester research projects for the following students.

1. Aleksandar Chakarov (PhD), advised by Sriram Sankaranarayanan, Summer 2010, Summer 2016. Computer Science. University of Colorado Boulder.

2. Alexandra Gendreau (PhD), advised by Ben Shapiro, Fall 2014–Spring 2015. Computer Science. University of Colorado Boulder.

3. Christoph Reichenbach (PhD 2009), advised by Amer Diwan, Spring 2009–Fall 2009. Computer Science. University of Colorado Boulder.

4. Nicholas Dronen (PhD 2016). Independent Study, Fall 2009. Computer Science. University of Colorado Boulder.

5. Krishna Chaitanya Sripada (MS 2016), co-advised with Pavol Cerny, Summer 2015–Spring 2016. Electrical, Computer, and Energy Engineering. University of Colorado Boulder.

6. Yi-Fan Tsai (MS 2013), Fall 2011–Spring 2014. Computer Science. University of Colorado Boulder.

7. Robert Frohardt (MS), co-advised with Sriram Sankaranarayanan, Fall 2009–Fall 2010. Computer Science. University of Colorado Boulder.

8. Daniel Stutzman (MS 2010), Spring 2010–Fall 2010. Computer Science. University of Colorado Boulder.

9. Kyle Howell (BS/MS 2016), Fall 2014–Spring 2016. Discovery Learning Apprentice, Fall 2014–Spring 2015. Computer Science. University of Colorado Boulder.

10. Parker Evans (BS 2016), co-advised with Pavol Cerny. Discovery Learning Apprentice, Fall 2013–Spring 2014. Computer Science. University of Colorado Boulder.

11. Kira Quan (BS/MS 2014), Fall 2012–Spring 2013. Computer Science. University of Colorado Boulder.

12. Chris Bubernak (BS 2013). Discovery Learning Apprentice, Fall 2011–Spring 2012. Computer Science. University of Colorado Boulder.

13. James Holley (BS 2012). Discovery Learning Apprentice, Fall 2009–Spring 2010. Electrical, Computer, and Energy Engineering. University of Colorado Boulder.

Current Research Advisees

1. Athithyaa Panchapakesan Rajeswari (MS), Summer 2016–present. Computer Science. University of Colorado Boulder.

2. Rhys Braginton Pettee Olsen (BS), co-advised with Sriram Sankaranarayanan, Spring 2016–present. Computer Science. University of Colorado Boulder.

3. Peilun Zhang (BS). Discovery Learning Apprentice, Fall 2016–present. Computer Science. University of Colorado Boulder.

CURRENT RESEARCH PROJECTS

University of Colorado Boulder   2016–
Ovv: Online Verification-Validation
Today's programmers essentially face a choice between creating software that is extensible and software that is verifiable. Languages either emphasize open extensibility or closed enforcement of abstractions. In this project, we explore language and virtual machine design techniques to offer verification in an extensible world by permitting analyses that freely interpose abstract interpretation with concrete execution in an incremental run-time system. With Matthew A. Hammer and David Van Horn.

University of Colorado Boulder   2015–
Auditr: Securing Space/Time Defenses in Java Bytecode
Recent work has highlighted vulnerabilities in server applications using resourage usage attacks that can cause denial of service and/or leakage of secret information. In this project, we investigate analysis techniques that enable a security analyst to more effectively secure an application to such attacks. Our premise is that such a technique must carefully orchestrate components that the detect, refine, and witness possible resource usage-based exploits. With Tianhan Lu, Pavol Černý, Sriram Sankaranarayanan, Ashutosh Trivedi, Işıl Dillig, Marijn Heule, and Henny Sipma.

University of Colorado Boulder   2014–
Fixr: Mining and Understanding Bug Fixes to Address Application-Framework Protcol Defects
With modern software frameworks like Android, there are millions of client applications (apps) that must carefully follow the complex and implicit protocols prescribed by the framework. And thus the same bug patterns occur over and over in apps. In this project, we search for algorithmic techniques for finding and generalizing bug fixes latent in source code repositories of some apps that can be automatically transferred to other apps. With Shawn Meier, Max Russek, Aleksandar Chakarov, Sergio Mover, Edmund S.L. Lam, Pavol Černý, Sriram Sankaranarayanan, Kenneth M. Anderson, and Tom Yeh.

University of Colorado Boulder   2011–
Thresher: Refutations via Goal-Directed Abstract Interpretation
Sound static analyzers over-approximate the concretely possible behaviors of programs. This statements means that they are certain when a given program satisfies a property of interest, but the trade-off they make is that they may raise false bug alarms. From a user's perspective, the presence of false alarms undermines the trustworthiness of the analyzer. In this project, we approach the false alarm problem by empowering the user with tools for alarm triage. Rather than focusing on guessing up-front what precision is necessary in the analyzer, we investigate approaches by which a user can direct further analysis of suspious alarms to quickly filter out false ones. We have effectively applied this approach to prove heap reachability properties and event-order safety properties of Android applications, and we are currently exploring applying this approach to the static analysis of dynamic languages. With Sam Blackshear, Benno Stein, Manu Sridharan, and Anders Møller.

SELECTED PAST RESEARCH PROJECTS

University of Colorado Boulder   2012–2014
Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants
Reflection is a language feature that reduces boilerplate and affords flexible and powerful frameworks. In current languages, however, the safety of reflection is checked only at run-time. In this project, we examine a modular, dependent type analysis that ensures compile-time reflection safety by checking refinement predicates relating mutable storage locations. With Devin Coughlin.

University of Colorado Boulder   2012–2014
Jsana: Abstract Domain Combinators for Dynamic Languages
A hallmark of dynamic languages is the presence of libraries that emulate other language features using underlying dynamic features (e.g., class systems using JavaScript's open objects). In this project, we investigate abstract domain combinators to enable modular reasoning of the dynamic language features, such as open,

dynamically-extendable objects, indirect property lookup, and property iteration. With Arlen Cox and Xavier Rival.

University of Colorado Boulder   2011–2012
Measuring Enforcements for Data-Driven Static Analysis Design
Static analysis design is incredibly expensive, as many long iterations are required to determine whether or not a design is sufficient. We say that a static analysis design is sufficient if it can prove the property of interest with an acceptable number of false alarms. Ultimately, the only way to confirm that an analysis design is sufficient is to implement it and run it on real-world programs. If the evaluation shows that the design is insufficient, the designer must return to the drawing board and repeat the process. In this project, we observe that developers embed information in the form of enforcements that are both important to static analysis design and measurable with dynamic instrumentation. With Devin Coughlin, Jeremy G. Siek, and Amer Diwan.

University of Maryland, College Park   2008–2011
Mix: Mixing Program Analyses
Program analysis design is an exercise in tradeoffs. A precise analysis verifies deeper properties but may become prohibitively expensive to use, while a coarse analysis is efficient but suffers from high false alarm rates. In this project, we examine how to mix radically different analysis algorithms of varying precision that enables the analysis user (rather than the analysis designer) to make such tradeoffs. As a case study, we investigate the mixing of type inference (efficient) and symbolic evaluation (precise). With Khoo Yit Phang and Jeffrey S. Foster.

University of California, Berkeley   2006–2010
Xisa: Extensible Inductive Shape Analysis
Shape analyses are unique in that they can capture detailed aliasing and structural information that is typically beyond the ability of other static program analyses. To do so, they rely on specialized data structure descriptions to build and decompose summaries of memory regions. Unfortunately, existing approaches suffer from usability and scalability issues that make them impractical to apply broadly. Typically, they either are insufficiently extensible or require low-level, expert interaction. Instead, our project focuses first on practicality by designing an extensible shape analysis based around high-level, program developer-oriented specifications. In particular, we observe that data structure checking code (e.g., used in testing or dynamic analysis) provides shape information that can also be used effectively in static analysis. With Xavier Rival and George Necula.

Refereed Publications in Proceedings

In many areas of Computer Science, including Programming Languages, the first and primary publication venues are full length papers in the proceedings of selective conferences. Many conferences have extensive two-phase review processes with author responses. The two main outlets are proceedings published by ACM (Association for Computing Machinery) and the Lecture Notes in Computer Science (LNCS) series published by Springer. The DBLP service (`http://dblp.uni-trier.de/db/`) maintains an up-to-date index of Computer Science literature. In the following, the † mark indicate students that I formally advise, and the ‡ mark indicate other student co-authors.

Matthew A. Hammer, **Bor-Yuh Evan Chang**, and David Van Horn. 2016. A Vision for Online Verification-Validation. In *ACM SIGPLAN International Conference on Generative Programming and Component Engineering (GPCE)*, 12 pages, October 2016. Acceptance: 37.7%.

Arlen Cox[†], **Bor-Yuh Evan Chang**, Huisong Li[‡], and Xavier Rival. 2015. Abstract Domains and Solvers for Sets Reasoning. In *International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR)*, volume 9450 of *Lect Notes Comput Sc*, pages 356-371, November 2015. Acceptance: 46.7%.

Sam Blackshear[†], **Bor-Yuh Evan Chang**, and Manu Sridharan. 2015. Selective Control-Flow Abstraction via Jumping. In *ACM SIGPLAN Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA)*, volume 50, number 10 of *ACM SIGPLAN Notices*, pages 163-182, October 2015. Acceptance: 25.2%.

Huisong Li[‡], **Bor-Yuh Evan Chang**, and Xavier Rival. 2015. Shape Analysis for Unstructured Sharing. In *International Static Analysis Symposium (SAS)*, volume 9291 of *Lect Notes Comput Sc*, pages 90-108, September 2015. Acceptance: 40.1%.

Arlen Cox[†], **Bor-Yuh Evan Chang**, and Xavier Rival. 2015. Desynchronized Multi-State Abstractions for Open Programs in Dynamic Languages. In *European Symposium on Programming (ESOP)*, volume 9032 of *Lect Notes Comput Sc*, pages 356-371, April 2015. Acceptance: 28.7%.

Arlen Cox[†], **Bor-Yuh Evan Chang**, and Xavier Rival. 2014. Automatic Analysis of Open Objects in Dynamic Language Programs. In *International Static Analysis Symposium (SAS)*, volume 8723 of *Lect Notes Comput Sc*, pages 134-150, September 2014. Acceptance: 37.7%.

Antoine Toubhans[‡], **Bor-Yuh Evan Chang**, and Xavier Rival. 2014. An Abstract Domain Combinator for Separately Conjoining Memory Abstractions. In *International Static Analysis Symposium (SAS)*, volume 8723 of *Lect Notes Comput Sc*, pages 285-301, September 2014. Acceptance: 37.7%.

Arlen Cox[†], **Bor-Yuh Evan Chang**, and Sriram Sankaranarayanan. 2014. QUICr: A Reusable Library for Parametric Abstraction of Sets and Numbers. In *International Conference on Computer Aided Verification (CAV)*, volume 8859 of *Lect Notes Comput Sc*, pages 866-873, July 2014. Acceptance: 24.9%.

Devin Coughlin[†] and **Bor-Yuh Evan Chang**. 2014. Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants. In *ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL)*, volume 49, number 1 of *ACM SIGPLAN Notices*, pages 73-86, January 2014. Acceptance: 23.2%.

Arlen Cox[†], **Bor-Yuh Evan Chang**, and Sriram Sankaranarayanan. 2013. QUIC Graphs: Relational Invariant Generation for Containers. In *European Conference on Object-Oriented Programming (ECOOP)*, volume 7920 of *Lect Notes Comput Sc*, pages 401-425, July 2013. Acceptance: 25.0%. **Distinguished Artifact Award**.

Sam Blackshear[†], **Bor-Yuh Evan Chang**, and Manu Sridharan. 2013. Thresher: Precise Refutations for Heap Reachability. In *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, volume 48, number 6 of *ACM SIGPLAN Notices*, pages 275-286, June 2013. Acceptance: 17.2%.

Antoine Toubhans[‡], **Bor-Yuh Evan Chang**, and Xavier Rival. 2013. Reduced Product Combination of Abstract Domains for Shapes. In *International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 7737 of *Lect Notes Comput Sc*, pages 375-395, January 2013. Acceptance: 37.5%.

Alejandro Sánchez[‡], Sriram Sankaranarayanan, César Sánchez, and **Bor-Yuh Evan Chang**. 2012. Invariant Generation for Parametrized Systems using Self-Reflection. In *International Static Analysis Symposium (SAS)*, volume 7460 of *Lect Notes Comput Sc*, pages 146-163, September 2012. Acceptance: 40.3%.

Devin Coughlin[†], **Bor-Yuh Evan Chang**, Amer Diwan, and Jeremy G. Siek. 2012. Measuring Enforcement Windows with Symbolic Trace Interpretation: What Well-Behaved Programs Say. In *International Symposium on Software Testing and Analysis (ISSTA)*, pages 276-286, July 2012. Acceptance: 28.7%.

Arlen Cox[†], Sriram Sankaranarayanan, and **Bor-Yuh Evan Chang**. 2012. A Bit Too Precise? Bounded Verification of Quantized Digital Filters. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 7214 of *Lect Notes Comput Sc*, pages 33-47, March 2012. Acceptance: 24.5%.

Sam Blackshear[†], **Bor-Yuh Evan Chang**, Sriram Sankaranarayanan, and Manu Sridharan. 2011. The Flow-Insensitive Precision of Andersen's Analysis in Practice. In *International Static Analysis Symposium (SAS)*, volume 6887 of *Lect Notes Comput Sc*, pages 60-76, September 2011. Acceptance: 32.8%.

Xavier Rival and **Bor-Yuh Evan Chang**. 2011. Calling Context Abstraction with Shapes. In *ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL)*, volume 46, number 1 of *ACM SIGPLAN Notices*, pages 173-186, January 2011. Acceptance: 23.4%.

Robert Frohardt[†], **Bor-Yuh Evan Chang**, and Sriram Sankaranarayanan. 2011. Access Nets: Modeling Access to Physical Spaces. In *International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 6538 of *Lect Notes Comput Sc*, pages 184-198, January 2011. Acceptance: 33.8%.

Khoo Yit Phang[‡], **Bor-Yuh Evan Chang**, and Jeffrey S. Foster. 2010. Mixing Type Checking and Symbolic Execution. In *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, volume 45, number 6 of *ACM SIGPLAN Notices*, pages 436-447, June 2010. Acceptance: 20.1%.

Vincent Laviron[‡], **Bor-Yuh Evan Chang**, and Xavier Rival. 2010. Separating Shape Graphs. In *European Symposium on Programming (ESOP)*, volume 6012 of *Lect Notes Comput Sc*, pages 387-406, March 2010. Acceptance: 24.8%.

Bor-Yuh Evan Chang and Xavier Rival. 2008. Relational Inductive Shape Analysis. In *ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL)*, volume 43, number 1 of *ACM SIGPLAN Notices*, pages 247-260, January 2008. Acceptance: 16.5%.

Bor-Yuh Evan Chang, Xavier Rival, and George C. Necula. 2007. Shape Analysis with Structural Invariant Checkers. In *International Static Analysis Symposium (SAS)*, volume 4634 of *Lect Notes Comput Sc*, pages 384-401, August 2007. Acceptance: 30.6%.

Bor-Yuh Evan Chang, Matthew Harren, and George C. Necula. 2006. Analysis of Low-Level Code Using Cooperating Decompilers. In *International Static Analysis Symposium (SAS)*, volume 4134 of *Lect Notes Comput Sc*, pages 318-335, August 2006. Acceptance: 28.8%.

Bor-Yuh Evan Chang, Adam Chlipala, and George C. Necula. 2006. A Framework for Certified Program Analysis and Its Applications to Mobile-Code Safety. In *International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 3855 of *Lect Notes Comput Sc*, pages 174-189, January 2006. Acceptance: 48.3%.

Bor-Yuh Evan Chang and K. Rustan M. Leino. 2005. Abstract Interpretation with Alien Expressions and Heap Structures. In *International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 3385 of *Lect Notes Comput Sc*, pages 147-163, January 2005. Acceptance: 29.3%.

Refereed Journal Publications

Benjamin Livshits, Manu Sridharan, Yannis Smaragdakis, Ondřej Lhoták, J. Nelson Amaral, **Bor-Yuh Evan Chang**, Samuel Z. Guyer, Uday P. Khedker, Anders Møller, and Dimitrios Vardoulakis. 2015. In Defense of Soundiness: A Manifesto. Commun. ACM, 58(2), pages 44-46, February 2015.

Arlen Cox[†], Sriram Sankaranarayanan, and **Bor-Yuh Evan Chang**. 2014. A Bit Too Precise? Verification of Quantized Digital Filters. Int J Softw Tools Technol Transfer, 16(2), pages 175-190, 2014.

Peer-Reviewed Invited Publications

Xavier Rival, Antoine Toubhans[‡], and **Bor-Yuh Evan Chang**. 2014. Construction of Abstract Domains for Heterogeneous Properties (Position Paper). In *International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISOLA)*, volume 8803 of *Lect Notes Comput Sc*, pages 489-492, October 2014.

Bor-Yuh Evan Chang and Xavier Rival. 2013. Modular Construction of Shape-Numeric Analyzers. In *Semantics, Abstract Interpretation, and Reasoning about Programs: Essays Dedicated to David A. Schmidt on the Occasion of his Sixtieth Birthday (SAIRP)*, volume 129 of *Electr P in Theor Comput Sc*, September 2013.

Mike Barnett, **Bor-Yuh Evan Chang**, Robert DeLine, Bart Jacobs, and K. Rustan M. Leino. 2005. Boogie: A Modular Reusable Verifier for Object-Oriented Programs. In *International Symposium on Formal Methods for Components and Objects (FMCO)*, volume 4111 of *Lect Notes Comput Sc*, pages 364-387, 2005.

Invited Publications

Bor-Yuh Evan Chang. 2014. Refuting Heap Rechability (Extended Abstract). In *International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 8318 of *Lect Notes Comput Sc*, pages 137-141, January 2014. **Invited contribution**.

Refereed Workshop Publications and Peer-Reviewed Short Papers

Workshops in Computer Science vary in whether they have formal proceedings (i.e., are considered publications) or only informal ones. Formal proceedings are published by ACM, in LNCS by Springer, or in Electronic Notes in Theoretical Computer Science (ENTCS) by Elsevier.

Sam Blackshear[†], Alexandra Gendreau[†], and **Bor-Yuh Evan Chang**. 2015. Droidel: A General Approach to Android Framework Modeling. In *ACM SIGPLAN Workshop on State of the Art in Program Analysis (SOAP)*, pages 19-25, June 2015.

Khalid Alharbi[‡], Sam Blackshear[†], Emily Kowalczyk[‡], Atif Memon, **Bor-Yuh Evan Chang**, and Tom Yeh. 2014. Android Apps Consistency Scrutinized. In *Extended Abstracts at ACM SIGCHI Conference on Human Factors in Computing Systems (CHI-EA)*, pages 2347-2352, April 2014.

Bor-Yuh Evan Chang, Amer Diwan, and Jeremy G. Siek. 2009. Gradual Programming: Bridging the Semantic Gap (Position Paper). In *Fun Ideas and Thoughts at ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI-FIT)*, 2 pages, June 2009.

Bor-Yuh Evan Chang and K. Rustan M. Leino. 2005. Inferring Object Invariants. In *International Workshop on Abstract Interpretation of Object-Oriented Languages (AIOOL)*, volume 131 of *Electr Notes Theor Comput Sc*, pages 63-74, January 2005. Acceptance: 84.6%.

Bor-Yuh Evan Chang, Adam Chlipala, George C. Necula, and Robert R. Schneck. 2005. Type-Based Verification of Assembly Language for Compiler Debugging. In *ACM SIGPLAN International Workshop on Types in Language Design and Implementation (TLDI)*, pages 91-102, January 2005. Acceptance: 43.5%.

Bor-Yuh Evan Chang, Adam Chlipala, George C. Necula, and Robert R. Schneck. 2005. The Open Verifier Framework for Foundational Verifiers. In *ACM SIGPLAN International Workshop on Types in Language Design and Implementation (TLDI)*, pages 1-12, January 2005. Acceptance: 43.5%.

Bor-Yuh Evan Chang and Manu Sridharan. 2003. PML: Toward a High-Level Formal Language for Biological Systems. In *Workshop on Concurrent Models in Molecular Biology (BioConcur)*, volume 180, number 3 of *Electr Notes Theor Comput Sc*, pages 15-30, September 2003.

Bor-Yuh Evan Chang, Karl Crary, Margaret DeLap, Robert Harper, Jason Liszka, Tom Murphy VII, and Frank Pfenning. 2002. Trustless Grid Computing in ConCert. In *International Workshop on Grid Computing (GRID)*, volume 2536 of *Lect Notes Comput Sc*, pages 112-125, November 2002.

Andreas Abel, **Bor-Yuh Evan Chang**, and Frank Pfenning. 2001. Human-Readable Machine-Verifiable Proofs for Teaching Constructive Logic. In *Workshop on Proof Transformations, Proof Presentations, and Complexity of Proofs (PTP)*, 14 pages, June 2001.

Editing

Bor-Yuh Evan Chang. 2013. *Workshop on Tools for Automatic Program Analysis (TAPAS)*, volume 311 of *Electr Notes Theor Comput Sc*.

Technical Reports

Yi-Fan Tsai[†], Devin Coughlin[†], **Bor-Yuh Evan Chang**, and Xavier Rival. 2015. Synthesizing Short-Circuiting Validation of Data Structure Invariants. Technical Report arXiv:1511.04846, University of Colorado Boulder, 18 pages.

Alejandro Sánchez[‡], Sriram Sankaranarayanan, César Sánchez, and **Bor-Yuh Evan Chang**. 2012. Invariant Generation for Parametrized Systems using Self-Reflection. Technical Report CU-CS-1094-12, University of Colorado Boulder, 31 pages.

Devin Coughlin[†], **Bor-Yuh Evan Chang**, Amer Diwan, and Jeremy G. Siek. 2012. Measuring Enforcement Windows with Symbolic Trace Interpretation: What Well-Behaved Programs Say. Technical Report CU-CS-1093-12, University of Colorado Boulder, 18 pages.

Sam Blackshear[†], **Bor-Yuh Evan Chang**, Sriram Sankaranarayanan, and Manu Sridharan. 2011. The Flow-Insensitive Precision of Andersen's Analysis in Practice. Technical Report CU-CS-1083-11, University of Colorado Boulder.

Robert Frohardt[†], **Bor-Yuh Evan Chang**, and Sriram Sankaranarayanan. 2011. Access Nets: Modeling Access to Physical Spaces. Technical Report CU-CS-1076-10, University of Colorado Boulder, 23 pages.

Khoo Yit Phang[‡], **Bor-Yuh Evan Chang**, and Jeffrey S. Foster. 2010. Mixing Type Checking and Symbolic Execution. Technical Report CS-TR-4954, University of Maryland, College Park, 19 pages.

Bor-Yuh Evan Chang. 2008. End-User Program Analysis. Technical Report UCB/EECS-2008-161, University of California, Berkeley.

Bor-Yuh Evan Chang, Xavier Rival, and George C. Necula. 2007. Shape Analysis with Structural Invariant Checkers. Technical Report UCB/EECS-2007-80, University of California, Berkeley.

Bor-Yuh Evan Chang, Matthew Harren, and George C. Necula. 2006. Analysis of Low-Level Code Using Cooperating Decompilers. Technical Report UCB/EECS-2006-86, University of California, Berkeley.

Bor-Yuh Evan Chang, Adam Chlipala, and George C. Necula. 2006. A Framework for Certified Program Analysis and Its Applications to Mobile-Code Safety. Technical Report UCB/ERL M05/32, University of California, Berkeley.

Bor-Yuh Evan Chang. 2005. Type-Based Verification of Assembly Language. Technical Report UCB/EECS-2008-186, University of California, Berkeley.

Bor-Yuh Evan Chang and K. Rustan M. Leino. 2005. Abstract Interpretation with Alien Expressions and Heap Structures. Technical Report MSR-TR-2004-115, Microsoft Research.

Bor-Yuh Evan Chang and Manu Sridharan. 2003. PML: Toward a High-Level Formal Language for Biological Systems. Technical Report UCB/CSD-03-1251, University of California, Berkeley.

Bor-Yuh Evan Chang, Kaustuv Chaudhuri, and Frank Pfenning. 2003. A Judgmental Analysis of Linear Logic. Technical Report CMU-CS-03-131R, Carnegie Mellon University.

Bor-Yuh Evan Chang. 2002. Iktara in ConCert: Realizing a Certified Grid Computing Framework from a Programmer's Perspective. Technical Report CMU-CS-02-150, Carnegie Mellon University.

PRESENTATIONS

Secure and Reliable Software Systems with Cooperative Program Analysis Computer Science Colloquium. University of Colorado Boulder. Boulder, Colorado, USA.	September 29, 2016
Fixr: Mining and Understanding Bug Fixes for App-Framework Protocol Defects. DARPA MUSE PI Meeting. Santa Cruz, California, USA.	September 8, 2016
Fixr: Mining and Understanding Bug Fixes for App-Framework Protocol Defects. DARPA MUSE Site Visit. University of Colorado Boulder. Boulder, Colorado, USA.	May 3, 2016
Fixr: Mining and Understanding Bug Fixes for App-Framework Protocol Defects. DARPA MUSE Demonstration Workshop. Arlington, Virginia, USA.	February 1, 2016
Cooperative Program Analysis for Reliable Mobile Applications. Huawei Vision Forum. Santa Clara, California, USA.	October 21, 2015

Goal-Directed Program Analysis with Jumping. Max Planck Institute for Software Systems. Kaiserslautern, Germany. — October 3, 2015

Goal-Directed Program Analysis with Jumping. École Normale Supérieure. Paris, France. — October 1, 2015

Goal-Directed Program Analysis with Jumping. Google. Mountain View, California, USA. — July 24, 2015

Fixr: Mining and Understanding Bug Fixes for App-Framework Protocol Defects. DARPA MUSE PI Meeting. SRI International. Menlo Park, California, USA. — July 22, 2015

Type-Intertwined Heap Analysis. Aarhus University. Aarhus, Denmark. — May 18, 2015

Fixr: Mining and Understanding Bug Fixes for App-Framework Protocol Defects. DARPA MUSE Site Visit. University of Colorado Boulder. Boulder, Colorado, USA. — February 25, 2015

Cooperative Program Analysis. Computer Science Colloquium. Colorado State University. Fort Collins, Colorado, USA. — September 22, 2014

Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants. Japan Advanced Institute of Science and Technology. Nomi, Japan. — August 6, 2014

Cooperative Program Analysis. National Taiwan University. Taipei, Taiwan. — August 5, 2014

Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants. National Taiwan University. Taipei, Taiwan. — August 1, 2014

Refuting Heap Reachability. National Chiao Tung University. Hsinchu, Taiwan. — July 31, 2014

Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants. Academia Sinica. Taipei, Taiwan. — July 30, 2014

Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants. Carnegie Mellon University. Pittsburgh, Pennsylvania, USA. — April 23, 2014

Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants. University of Maryland, College Park. College Park, Maryland, USA. — February 26, 2014

Refuting Heap Reachability. Invited Keynote: Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI). San Diego, California, USA. — January 20, 2014

Cooperative Program Analysis. Computer Science Colloquium. University of Colorado Boulder. Boulder, Colorado, USA. — October 3, 2013

Precise Heap Reachability by Refutation Analysis. Université Paris Diderot. Paris, France. — July 22, 2013

Precise Heap Reachability by Refutation Analysis. École Normale Supérieure. Paris, France. — July 19, 2013

Precise Heap Reachability by Refutation Analysis. Dagstuhl Seminar 13162: Pointer Analysis. Wadern, Germany. — April 16, 2013

Witnessing Heap Reachability for Resource Leaks in Android. Google. Mountain View, California, USA.   August 29, 2012

Measuring Enforcement Windows with Symbolic Trace Interpretation: What Well-Behaved Programs Say. Danish Static Analysis Symposium (DANSAS). Odense, Denmark.   August 24, 2012

Xisa: Extensible Inductive Shape Analysis. Aarhus University. Aarhus, Denmark.   August 23, 2012

Modular Reflection Checking using Relationship Refinements. Aarhus University. Aarhus, Denmark.   August 20, 2012

The Flow-Insensitive Precision of Andersen's Analysis in Practice. University of California, Berkeley. Berkeley, California, USA.   June 10, 2011

Xisa: Extensible Inductive Shape Analysis. Carnegie Mellon University. Pittsburgh, Pennsylvania, USA.   March 16, 2011

Calling Context Abstraction with Shapes National Taiwan University. Taipei, Taiwan.   December 17, 2010

Mixing Type Checking and Symbolic Execution. Front Range Architecture, Compilers, Tools, and Languages Workshop (FRACTAL). Boulder, Colorado, USA.   December 5, 2009

End-User Program Analysis for Data Structures. National Taiwan University. Taipei, Taiwan.   August 12, 2009

Using Checkers for End-User Shape Analysis. National Taiwan University. Taipei, Taiwan.   August 11, 2009

End-User Shape Analysis. National Taiwan University. Taipei, Taiwan.   August 11, 2009

Reduction in End-User Shape Analysis. Dagstuhl Seminar 09301: Typing, Analysis, and Verification of Heap-Manipulating Programs. Wadern, Germany.   July 24, 2009

Gradual Programming: Bridging the Semantic Gap. Fun Ideas and Thoughts Session (FIT) at the 2009 Conference on Programming Language Design and Implementation (PLDI'09). Dublin, Ireland.   June 16, 2009

End-User Program Analysis for Data Structures. Computer Science Department Colloquium. University of Virginia. Charlottesville, Virginia, USA.   November 24, 2008

End-User Program Analysis. Dissertation Talk. University of California, Berkeley. Berkeley, California, USA.   August 28, 2008

Extensible Shape Analysis by Designing with the User in Mind. Open Source Quality Project Retreat. Santa Cruz, California, USA.   May 16, 2008

Precise Program Analysis with Data Structures. Job Talk.   February–April 2008

Relational Inductive Shape Analysis. Thirty-Fifth International Symposium on Principles of Programming Languages (POPL'08). San Francisco, California, USA.   January 11, 2008

Materialization in Shape Analysis with Structural Invariant Checkers. Copenhagen Programming Language Seminar. IT University of Copenhagen. Copenhagen, Denmark. — August 27, 2007

Shape Analysis with Structural Invariant Checkers. Fourteenth International Static Analysis Symposium (SAS'07). Kongens Lyngby, Denmark. — August 24, 2007

Shape Analysis with Structural Invariant Checkers. Open Source Quality Project Retreat. Santa Cruz, California, USA. — May 10, 2007

Analysis of Low-Level Code Using Cooperating Decompilers. Thirteenth International Static Analysis Symposium (SAS'06). Seoul, Korea. — August 31, 2006

Inferring Object Invariants. First International Workshop on Abstract Interpretation of Object-Oriented Languages (AIOOL'05). Paris, France. — January 21, 2005

Abstract Interpretation with Alien Expressions and Heap Structures. Sixth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'05). Paris, France. — January 18, 2005

Type-Based Verification of Assembly Language for Compiler Debugging. Second International Workshop on Types in Language Design and Implementation (TLDI'05). Long Beach, California, USA. — January 10, 2005

Extensible Verification of Untrusted Code. Open Source Quality Project Retreat. Santa Cruz, California, USA. — May 13, 2004

PML: Toward a High-Level Formal Language for Biological Systems. First Workshop on Concurrent Models in Molecular Biology (BioConcur'03). Marseille, France. — September 6, 2003

Human-Readable Machine-Verifiable Proofs for Teaching Constructive Logic. Workshop on Proof Transformations, Proof Presentations and Complexity of Proofs (PTP'01). Siena, Italy. — June 19, 2001

TEACHING EXPERIENCE

University of Colorado Boulder — Fall 2016, Fall 2015, Spring 2014, Fall 2012, Spring 2012, and Fall 2009
Principles of Programming Languages, CSCI 3155, *Instructor*
Undergraduate-level course on the concepts of programming languages. In Spring 2012, I revised course using project-based principles.

University of Colorado Boulder — Spring 2016, Fall 2014, Spring 2013, and Fall 2010
Program Analysis: Theory and Practice, CSCI 7135, *Instructor*
Graduate seminar on the theoretical foundations and practical implementations of program analysis.

University of Colorado Boulder — Spring 2015 and Fall 2011
Compiler Construction, CSCI 4555/5525, ECEN 4553/5523, *Instructor*
Combined undergraduate and graduate-level course on an introduction to compiler construction concepts and techniques.

University of Colorado Boulder — Spring 2012
Readings in Programming Languages, CSCI 7900, *Instructor*
Advanced graduate reading seminar on current research topics. Discussions are primarily led by advanced graduate students.

University of Colorado Boulder   Spring 2011
Program Analysis Practicum, CSCI 7135, *Instructor*
Graduate practicum on program analysis design and implementation.

University of Colorado Boulder   Fall 2013, Spring 2010, and Spring 2009
Fundamentals of Programming Languages, CSCI 5535, *Instructor*
Core graduate-level course on the fundamental ideas behind modern programming language design and analysis.

University of California, Berkeley   Spring 2004
Programming Languages and Compilers, CS164, *Graduate Student Instructor*
Upper division course on programming language principles and compiler design, assisting Prof. George Necula. Also, applied research ideas to develop Coolaid, an assembly-level type-checking tool, to help students with compiler development and understanding.

Carnegie Mellon University   Fall 2000
Principles of Programming, 15-212, *Teaching Assistant*
Lower division course on abstraction and reasoning about programs and functional programming (taught in Standard ML), assisting Prof. Karl Crary and Prof. John Lafferty.

Carnegie Mellon University   Spring 1999
Fundamentals of Computer Science I, 15-211, *Teaching Assistant*
Lower division course on data structures and algorithms in C++, assisting Prof. Klaus Sutner.

Carnegie Mellon University   Fall 1999
Mathematical Foundations of Computer Science, 15-151, *Teaching Assistant*
Lower division course on fundamental concepts of discrete mathematics using Mathematica, assisting Prof. Edmund Clarke and Prof. Klaus Sutner.

PROFESSIONAL ACTIVITIES

Program Chair

Fifteenth Asian Symposium on Programming Languages and Systems (APLAS'17)

Fourth International Workshop on Tools for Automatic Program Analysis (TAPAS'13)

Program Committees

Sixth International Workshop on State of the Art in Program Analysis (SOAP'17)

Twenty-Ninth International Conference on Computer-Aided Verification (CAV'17)

Thirty-First European Conference on Object-Oriented Programming (ECOOP'17)

The 2017 Conference on Programming Language Design and Implementation (PLDI'17)

Fourteenth Asian Symposium on Programming Languages and Systems (APLAS'16)

Twenty-Third International Static Analysis Symposium (SAS'16)

Seventeenth International Conference on Verification, Model Checking, and Abstract Interpretation (VM-CAI'16)

Forty-Third Symposium on Principles of Programming Languages (POPL'16)

Seventh IFIP Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE'15)

Third International Workshop on Tools for Automatic Program Analysis (TAPAS'12)

Thirteenth International Conference on Verification, Model Checking, and Abstract Interpretation (VM-CAI'12)

Tenth Workshop on Program Analysis for Software Tools and Engineering (PASTE'11)

Twenty-Fourth International Workshop on Languages and Compilers for Parallel Computing (LCPC'11)

Third International Workshop on Numerical and Symbolic Abstract Domains (NSAD'11)

First Workshop on Abstract Interpretation of Object-Oriented Languages (AIOOL'05)

Steering Committees

International Workshop on Tools for Automatic Program Analysis (TAPAS)

External Review Committees

The 2015 Conference on Programming Language Design and Implementation (PLDI'15)

Forty-Second Symposium on Principles of Programming Languages (POPL'15)

The 2012 Conference on Programming Language Design and Implementation (PLDI'12)

Thirty-Ninth Symposium on Principles of Programming Languages (POPL'12)

The 2009 Conference on Programming Language Design and Implementation (PLDI'09)

External Reviews

Science of Computer Programming (2016)

Transactions on Programming Languages and Systems (2016)

Theoretical Computer Science (2015)

Computing Surveys (2015)

Transactions on Programming Languages and Systems (2014)

Computing Surveys (2014)

Acta Informatica (2014)

Journal of Automated Reasoning (2014)

The 2014 Conference on Programming Language Design and Implementation (PLDI'14)

Twenty-Fourth European Symposium on Programming (ESOP'14)

Forty-First Symposium on Principles of Programming Languages (POPL'14)

Eleventh Asian Symposium on Programming Languages and Systems (APLAS'13)

Semantics, Abstract Interpretation, and Reasoning about Programs: Essays Dedicated to David A. Schmidt on the Occasion of his Sixtieth Birthday (SAIRP'13)

Transactions on Programming Languages and Systems (2013)

Twenty-Third European Symposium on Programming (ESOP'13)

Fortieth Symposium on Principles of Programming Languages (POPL'13)

Second International Conference on Certified Programs and Proofs (CPP'12)

Transactions on Programming Languages and Systems (2012)

Transactions on Programming Languages and Systems (2011)

Twenty-Second European Symposium on Programming (ESOP'12)

Ninth Asian Symposium on Programming Languages and Systems (APLAS'11)

Eighteenth International Static Analysis Symposium (SAS'11)

Twelfth Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'11)

Thirty-Eighth Symposium on Principles of Programming Languages (POPL'11)

Nineteenth European Symposium on Programming (ESOP'10)

Thirteenth Conference on Foundations of Software Science and Computation Structures (FoSSaCS'10)

Thirty-Seventh Symposium on Principles of Programming Languages (POPL'10)

Eighteenth European Symposium on Programming (ESOP'09)

Fourteenth Static Analysis Symposium (SAS'07)

Twenty-First Symposium on Logic in Computer Science (LICS'06)

The 2006 Conference on Programming Language Design and Implementation (PLDI'06)

The 2006 Symposium on Security and Privacy (Oakland'06)

Thirty-Third Symposium on Principles of Programming Languages (POPL'06)

Thirty-Second Symposium on Principles of Programming Languages (POPL'05)

Ninth Conference on Functional Programming (ICFP'04)

Professional Service

Selection Committee, Student Reserach Competition, the 2015 Conference on Object Systems, Programming, Languages and Applications: Software for Humanity (SPLASH'15)

Selection Committee, Student Reserach Competition, the 2015 Conference on Programming Language Design and Implementation (PLDI'15)

Co-Treasurer, Forty-First Symposium on Principles of Programming Languages (POPL'14)

Treasurer, Fortieth Symposium on Principles of Programming Languages (POPL'13)

Treasurer, Thirty-Ninth Symposium on Principles of Programming Languages (POPL'12)

Treasurer, Thirty-Eighth Symposium on Principles of Programming Languages (POPL'11)

Co-Treasurer, Thirty-Seventh Symposium on Principles of Programming Languages (POPL'10)

Organizer, Front Range Architecture, Compilers, Tools, and Languages Workshop (FRACTAL), Fall 2009.

Student Committees

PhD Dissertation Committee

Aditya Zutshi (PhD), advised by Sriram Sankaranarayanan. Electrical, Computer, and Energy Engineering. University of Colorado Boulder.

Magnus Madsen (PhD 2015), advised by Anders Møller. Department of Computer Science. Aarhus University, Denmark.

Weiyu Miao (PhD 2013), advised by Jeremy G. Siek. Electrical, Computer, and Energy Engineering. University of Colorado Boulder.

Graham Price (PhD 2011), advised by Manish Vachharajani. Electrical, Computer, and Energy Engineering. University of Colorado Boulder.

Christoph Reichenbach (PhD 2009), advised by Amer Diwan. Computer Science. University of Colorado Boulder.

PhD Proposal/PhD Comprehensive Examination Committee

Hadi Ravanbakhsh (passed May 12, 2016), advised by Sriram Sankaranarayanan. Computer Science. University of Colorado Boulder.

Aditya Zutshi (passed December 16, 2015), advised by Sriram Sankaranarayanan. Electrical, Computer, and Energy Engineering. University of Colorado Boulder.

Aleksandar Chakarov (passed February 16, 2015), advised by Sriram Sankaranarayanan. Computer Science. University of Colorado Boulder.

Graham Price (passed 2010), advised by Manish Vachharajani. Electrical, Computer, and Energy Engineering. University of Colorado Boulder.

Joseph Blomstedt (passed December 9, 2009), advised by Dirk Grunwald and Jeremy Siek. Electrical, Computer, and Energy Engineering. University of Colorado Boulder.

PhD Preliminary Examination Committee

Hadi Ravanbakhsh, advised by Sriram Sankaranarayanan. Computer Science. University of Colorado Boulder.

Aleksandar Chakarov, advised by Sriram Sankaranarayanan. Computer Science. University of Colorado Boulder.

Chris Wailes, advised by Dirk Grunwald and Jeremy Siek. Computer Science. University of Colorado Boulder.

Weiyu Miao, advised by Jeremy Siek. Electrical, Computer, and Energy Engineering. University of Colorado Boulder.

MS Thesis Committee

Erin Duggan (MS 2015), advised by Tom Yeh. Computer Science. University of Colorado Boulder.

Shashank Bharadwaj (MS 2012), advised by Jeremy Siek. Electrical, Computer, and Energy Engineering. University of Colorado Boulder.

Erik Silkensen (MS 2012), advised by Jeremy Siek. Computer Science. University of Colorado Boulder.

Huxley Bennett (MS 2012), advised by Sriram Sankaranarayanan. Computer Science. University of Colorado Boulder.

Moss Prescott (MS 2010), advised by Jeremy Siek. Computer Science. University of Colorado Boulder.

Scott Mackey (MS 2009), advised by Ken Anderson. Computer Science. University of Colorado Boulder.

ME Report Committee

Jesse Bowes (ME 2016), advised by Ken Anderson. Computer Science. University of Colorado Boulder.

Mark Grebe (ME 2013), advised by Ken Anderson. Computer Science. University of Colorado Boulder.

Russell Winkler (ME 2012), advised by Ken Anderson. Computer Science. University of Colorado Boulder.

BS Thesis Committee

Trystan Binkley-Jones (BS 2015), advised by Ken Anderson. Computer Science. University of Colorado Boulder.

Emily Bertelson (BS 2015), advised by Tom Yeh. Computer Science. University of Colorado Boulder.

Eric Horacek (BS 2013), advised by Ken Anderson. Computer Science. University of Colorado Boulder.

Pavol Zelinsky (BS 2012), advised by Elizabeth R. Jessup. Computer Science. University of Colorado Boulder.

Erik Silkensen (BS 2011), advised by Jeremy Siek. Computer Science. University of Colorado Boulder.

Robert Stimpfling (BS 2010), advised by Ken Anderson. Computer Science. University of Colorado Boulder.

Departmental Service

Department of Computer Science. University of Colorado Boulder

Executive Committee: 8/2015–present

Faculty Search Committee: 8/2014–5/2016

Colloquium Chair: 6/2012–present

Graduate Committee: 1/2009–8/2011

PhD Recruiting Weekend: co-organized 2009, 2011

Organizes PhD Preliminary Exam in Programming Languages: 2010–present

Department of Electrical, Computer, and Energy Engineering. University of Colorado Boulder

Faculty Search Committee: 8/2011–5/2012

Computer Science Division. University of California, Berkeley.

Computer Science Graduate Student Association (CSGSA) Faculty Candidate Committee: 2007 (chair), 2006, and 2005

College Service

Transfer Credit Evaluator for Computer Science courses on behalf of the College of Engineering and Applied Sciences: 04/2015–present.

University Service

Innovative Seed Grant Proposal Reviews: 2011

Professional Affiliations

Association for Computing Machinery (ACM)
  Special Interest Group on Programming Languages (SIGPLAN)
  Special Interest Group on Software Engineering (SIGSOFT)
  Special Interest Group on Logic and Computation (SIGLOG)

CURRENT COLLABORATORS

CU: Profs. Kenneth M. Anderson, Matthew Hammer, Sriram Sankaranarayanan, and Tom Yeh (Computer Science); Prof. Pavol Černý (Electrical, Computer, and Energy Engineering).

National: Prof. Isil Dillig, Dr. Marijn J. H. Heule (UT Austin); Dr. Henny Sipma (Kestrel Technology); Prof. Samuel Z. Guyer (Tufts); Prof. Amer Diwan (Google); Prof. Jeremy Siek (Indiana); Profs. Jeffrey S. Foster, Atif Memon, David Van Horn (Maryland); Dr. Manu Sridharan (Samsung Research).

International: Dr. Xavier Rival (INRIA/ENS/CNRS, France); Prof. Anders Møller (Aarhus University, Denmark).

CITIZENSHIP

United States of America