

Bor-Yuh Evan Chang

Curriculum Vitae

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
End-User Program Analysis
Dissertation Committee:
Prof. George C. Necula (chair), Prof. Koushik Sen, and Prof. Jack Silver
Proposal Committee:
Prof. Rastislav Bodik (chair), Prof. Eric Brewer, Prof. George C. Necula, and Prof. Jack Silver
- MS **University of California, Berkeley**, Computer Science 2005
Advisor: Prof. George C. Necula
Type-Based Verification of Assembly Language
Committee: Prof. George C. Necula and Prof. Rastislav Bodik
- BS **Carnegie Mellon University**, Computer Science, 4.0 GPA 2002
University and College Honors
Minors: Biological Science and Mathematical Science
Iktara in ConCert: Realizing a Certified Grid Computing Framework from a Programmer's Perspective
Advisors: Prof. Robert Harper and Prof. Frank Pfenning

ACADEMIC APPOINTMENTS

- University of Colorado, Boulder (Boulder, CO) January 2009–present
Assistant Professor, Department of Computer Science
- University of Maryland, College Park (College Park, MD) September 2008–November 2008
Faculty Research Assistant (Postdoc), Department of Computer Science
Sponsor: Jeffrey S. Foster

CURRENT RESEARCH PROJECTS

Witness-Generating Program Analysis

In general, today's program analyzers provide poor explanations for their results and thus limiting their utility to everyday programmers. Such poor explanations are particularly problematic when analyzers report potential errors, as the user must decide whether the report is a real error or a false alarm due to an analyzer limitation. In this project, we devise algorithms for producing evidence for program analysis results. Importantly, this form of evidence will enable the analyzer user to interactively refine the results to identify real from false alarms.

(with Sam Blackshear, Sriram Sankaranarayanan, and Manu Sridharan)

Gradual Programming

There is no perfect programming language. Programmers must write code conforming to the idiosyncrasies of a programming language. Thus, there is often a disconnect between the intent of the developer and the meaning of the program. This semantic gap has a negative effect on programmer productivity, software reliability, and execution efficiency. In this project, we look to address this semantic gap through a drastic rethinking of how we develop software.

(with Devin Coughlin, Jeremy G. Siek, and Amer Diwan)

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)

Extensible Shape Analysis with Invariant Checkers

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)

PAST RESEARCH PROJECTS AND EXPERIENCE

University of California, Berkeley (Berkeley, CA)

2005–2006

Cooperating Decompilers for the Analysis of Low-Level Code

With Matthew Harren and George Necula, created a framework to extend source-level analyzers to operate on assembly code through the use of decompilation.

Microsoft Research (Redmond, WA)

2005

Inferring Object Invariants

With K. Rustan M. Leino, identified certain classes of invariants common in the verification of object-oriented programs and developed algorithms to try to infer them automatically.

- University of California, Berkeley (Berkeley, CA) 2003–2005
Extensible Verification of Low-Level Code
 With George Necula, Robert Schneck, and Adam Chlipala, explored developing a practical and extensible framework for foundational proof-carrying code. Developed two techniques for addressing this problem: a framework based on certified code verifiers (Proof-Carrying Verifiers) and a framework based on proof-generating code verifiers (the Open Verifier Framework).
- University of California, Berkeley (Berkeley, CA) 2003–2005
Coolaid: Type-Based Verification of Assembly Language
 With George Necula, developed techniques for type-checking assembly code given only source-level type information. Then, applied these techniques in a class on compiler design to help students debug their compiler projects, as well as to teach them about compilation and program analysis.
- Microsoft Research (Redmond, WA) 2004
Combining Abstract Interpreters
 With K. Rustan M. Leino, explored modularly extending abstract interpreters to deal with uninterpreted functions in order to cooperatively handle richer theories.
- University of California, Berkeley (Berkeley, CA) 2003
Toward a High-Level Formal Language for Biological Systems
 With Manu Sridharan, explored the use of concurrency theory for modeling biological systems, such as cellular pathways, as concurrent computational processes.
- Carnegie Mellon University (Pittsburgh, PA) 2001–2002
Realizing a Certified Grid Computing Framework from a Programmer’s Perspective
 With Robert Harper and Frank Pfenning, explored the development of a parallel theorem prover for linear logic to push the development of an architecture for certified grid computing.
- Carnegie Mellon University (Pittsburgh, PA) 2001
Human-Readable Machine-Verifiable Proofs for Teaching Constructive Logic
 With Frank Pfenning and Andreas Abel, extended the proof checker used in a constructive logic course to allow higher-level statements corresponding to steps in a rigorous mathematical proof one would typically carry out on paper.

REFEREED PUBLICATIONS

- Arlen Cox, Sriram Sankaranarayanan, and **Bor-Yuh Evan Chang**. A Bit Too Precise? Bounded Verification of Quantized Digital Filters. In *Proceedings of the Eighteenth International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS’12)*, March 2012.
- Sam Blackshear, **Bor-Yuh Evan Chang**, Sriram Sankaranarayanan, and Manu Sridharan. The Flow-Insensitive Precision of Andersen’s Analysis in Practice. In *Proceedings of the Eighteenth International Static Analysis Symposium (SAS’11)*, September 2011.
- Xavier Rival and **Bor-Yuh Evan Chang**. Calling Context Abstraction with Shapes. In *Proceedings of the Thirty-Eighth International Symposium on Principles of Programming Languages (POPL’11)*, January 2011.
- Robert Frohardt, **Bor-Yuh Evan Chang**, and Sriram Sankaranarayanan. Access Nets: Modeling Access to Physical Spaces. In *Proceedings of the Twelfth International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI’11)*, January 2011.
- Khoo Yit Phang, **Bor-Yuh Evan Chang**, and Jeffrey S. Foster. Mixing Type Checking and Symbolic Execution. In *Proceedings of the 2010 Conference on Programming Languages Design and Implementation (PLDI’10)*, June 2010.

- Vincent Laviron, **Bor-Yuh Evan Chang**, and Xavier Rival. Separating Shape Graphs. In *Proceedings of the Nineteenth European Symposium on Programming (ESOP'10)*, March 2010.
- Bor-Yuh Evan Chang** and Xavier Rival. Relational Inductive Shape Analysis. In *Proceedings of the Thirty-Fifth International Symposium on Principles of Programming Languages (POPL'08)*, January 2008.
- Bor-Yuh Evan Chang**, Xavier Rival, and George C. Necula. Shape Analysis with Structural Invariant Checkers. In *Proceedings of the Fourteenth International Static Analysis Symposium (SAS'07)*, August 2007.
- Bor-Yuh Evan Chang**, Matthew Harren, and George C. Necula. Analysis of Low-Level Code Using Cooperating Decompilers. In *Proceedings of the Thirteenth International Static Analysis Symposium (SAS'06)*, August 2006.
- Bor-Yuh Evan Chang**, Adam Chlipala, and George C. Necula. A Framework for Certified Program Analysis and Its Applications to Mobile-Code Safety. In *Proceedings of the Seventh International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'06)*, January 2006.
- Mike Barnett, **Bor-Yuh Evan Chang**, Robert DeLine, Bart Jacobs, and K. Rustan M. Leino. Boogie: A Modular Reusable Verifier for Object-Oriented Programs. In *Proceedings of the Fourth International Symposium on Formal Methods for Components and Objects (FMCO'05)*, November 2005.
- Bor-Yuh Evan Chang** and K. Rustan M. Leino. Abstract Interpretation with Alien Expressions and Heap Structures. In *Proceedings of the Sixth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'05)*, January 2005.
- Bor-Yuh Evan Chang** and K. Rustan M. Leino. Inferring Object Invariants. In *Proceedings of the First International Workshop on Abstract Interpretation of Object-Oriented Languages (AIOOL'05)*, January 2005.
- Bor-Yuh Evan Chang**, Adam Chlipala, George C. Necula, and Robert R. Schneck. The Open Verifier Framework for Foundational Verifiers. In *Proceedings of the Second International Workshop on Types in Language Design and Implementation (TLDI'05)*, January 2005.
- Bor-Yuh Evan Chang**, Adam Chlipala, George C. Necula, and Robert R. Schneck. Type-Based Verification of Assembly Language for Compiler Debugging. In *Proceedings of the Second International Workshop on Types in Language Design and Implementation (TLDI'05)*, January 2005.
- Bor-Yuh Evan Chang** and Manu Sridharan. PML: Toward a High-Level Formal Language for Biological Systems. In *Proceedings of the First Workshop on Concurrent Models in Molecular Biology (BioConcur'03)*, September 2003.
- Bor-Yuh Evan Chang**, Karl Crary, Margaret DeLap, Robert Harper, Jason Liska, Tom Murphy VII, and Frank Pfenning. Trustless Grid Computing in ConCert. In *Proceedings of the Third International Workshop on Grid Computing (GRID'02)*, November 2002.
- Andreas Abel, **Bor-Yuh Evan Chang**, and Frank Pfenning. Human-Readable Machine-Verifiable Proofs for Teaching Constructive Logic. In *Proceedings of the Workshop on Proof Transformations, Proof Presentations and Complexity of Proofs (PTP'01)*, June 2001.

TECHNICAL REPORTS

- Sam Blackshear, **Bor-Yuh Evan Chang**, Sriram Sankaranarayanan, and Manu Sridharan. The Flow-Insensitive Precision of Andersen’s Analysis in Practice (Extended Version). Technical Report CU-CS-1083-11, June 2011.
- Robert Frohardt, **Bor-Yuh Evan Chang**, and Sriram Sankaranarayanan. Access Nets: Modeling Access to Physical Spaces (Extended Version). Technical Report CU-CS-1076-10, November 2010.
- Bor-Yuh Evan Chang**, Xavier Rival, and George C. Necula. Shape Analysis with Structural Invariant Checkers. Technical Report UCB/EECS-2007-80, University of California, Berkeley, June 2007.
- Bor-Yuh Evan Chang**, Matthew Harren, and George C. Necula. Analysis of Low-Level Code Using Cooperating Decompilers. Technical Report UCB/EECS-2006-86, University of California, Berkeley, June 2006.
- Bor-Yuh Evan Chang**, Adam Chlipala, and George C. Necula. A Framework for Certified Program Analysis and Its Applications to Mobile-Code Safety. Technical Report UCB/ERL M05/32, University of California, Berkeley, November 2005.
- Bor-Yuh Evan Chang** and K. Rustan M. Leino. Abstract Interpretation with Alien Expressions and Heap Structures. Technical Report MSR-TR-2004-115, Microsoft Research, November 2004.
- Bor-Yuh Evan Chang** and Manu Sridharan. PML: Toward a High-Level Formal Language for Biological Systems. Technical Report UCB/CSD-03-1251, University of California, Berkeley, June 2003.
- Bor-Yuh Evan Chang**, Kaustuv Chaudhuri, and Frank Pfenning. A Judgmental Analysis of Linear Logic. Technical Report CMU-CS-03-131R, Carnegie Mellon University, December 2003.

PRESENTATIONS

- | | |
|--|-------------------|
| 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

CURRENT ADVISEES

PhD Sam Blackshear
PhD Devin Coughlin
PhD Arlen Cox
PhD Yi-Fan Tsai

GRANTS

NSF CCF-1055066, “CAREER: Cooperative Program Analysis: Bridging the Gap Between User and Tool Reasoning,” \$459,584, 06/01/2011–05/31/2016, Bor-Yuh Evan Chang (PI).

Joint Institute for Strategic Energy Analysis UGA-0-41026-04, “Verifiable Decision-Making Algorithms for Reconfiguration of Electric Microgrids,” \$49,985, 07/01/2010–06/30/2011, Siddharth Suryanarayanan, Colorado State University and Sriram Sankaranarayanan (PIs); Bor-Yuh Evan Chang and Dirk Grunwald (co-PIs).

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).

TEACHING EXPERIENCE

University of Colorado, Boulder (Boulder, CO) Spring 2012 and Fall 2009
Principles of Programming Languages, *Instructor*
Undergraduate-level course on the concepts of programming languages.

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

University of Colorado, Boulder (Boulder, CO) Fall 2011
Compiler Construction, *Instructor*
Combined undergraduate and graduate-level course on an introduction to compiler construction concepts and techniques.

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

University of Colorado, Boulder (Boulder, CO) Fall 2010
Program Analysis: Theory and Practice, *Instructor*
Graduate seminar on the theoretical foundations and practical implementations of program analysis.

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

University of California, Berkeley (Berkeley, CA) Spring 2004
Programming Languages and Compilers, *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 (Pittsburgh, PA) Fall 2000

Principles of Programming, *Teaching Assistant*

Lower division course on abstraction and reasoning about programs and functional programming (taught in Standard ML), assisting Prof. Karl Cray and Prof. John Lafferty.

Carnegie Mellon University (Pittsburgh, PA) Spring 1999

Fundamentals of Computer Science I, *Teaching Assistant*

Lower division course on data structures and algorithms in C++, assisting Prof. Klaus Sutner.

Carnegie Mellon University (Pittsburgh, PA) Fall 1999

Mathematical Foundations of Computer Science, *Teaching Assistant*

Lower division course on fundamental concepts of discrete mathematics using Mathematica, assisting Prof. Edmund Clarke and Prof. Klaus Sutner.

AWARDS AND HONORS

National Science Foundation CAREER Award December 2010

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

National Science Foundation Graduate Research Fellowship 2004–2007

California Microelectronics Fellowship 2002–2003

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

PROFESSIONAL ACTIVITIES

Program Committees

Thirteenth International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'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)

External Review Committees

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

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

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.

Departmental Service

Department of Computer Science. University of Colorado, Boulder

Graduate Committee: 1/2009–8/2011

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

Faculty Search Committee: 8/2011–present

Computer Science Division. University of California, Berkeley.

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

University Service

Innovative Seed Grant Proposal Reviews: 2011

Professional Affiliations

Association for Computing Machinery (ACM)
Special Interest Group on Programming Languages (SIGPLAN)

INDUSTRY EXPERIENCE

Inktomi (Foster City, CA) Summer 2001

Web Search Content, Intern

Investigated and developed a tool for generating summaries for arbitrary web pages, explored automated identification of affiliate networks/spam, developed a tool for gathering near-duplicate information between hosts for automated mirror site identification.

Hewlett-Packard (Fort Collins, CO) Summer 2000

Static Timing Analysis, Intern

Designed an architecture for performing static timing analysis to enable the development of a suite of static timing tools with varying levels of accuracy/performance. Initiated the development of both the underlying architecture and a logic depth analysis tool using this architecture.

Hewlett-Packard (Fort Collins, CO) Summer 1999

VLSI Design Database Infrastructure, Intern

Investigated and performed customer interviews concerning an unfolded/occurrence model for a design database and a memory efficient model for representing electrical information. Designed and developed a prototype for the electrical information model and a reader to utilize this model.

Hewlett-Packard (Fort Collins, CO) Summer 1998

VLSI Design Manipulation, Intern

Developed CAD tools to smash design hierarchy and to splice electric circuits using an existing electric-circuit connectivity model for optimal speed and memory usage.

CITIZENSHIP

United States of America