

Bor-Yuh Evan Chang

Curriculum Vitae

evan.chang@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

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.

Cooperating Decompilers for the Analysis of Low-Level Code

Analysis or verification of low-level code (e.g., assembly code) is useful for minimizing the disconnect between what is verified and what is actually executed and is necessary when source code is unavailable, is written in a hard to analyze language, or is, say, intermingled with inline assembly. Most program analyses, however, operate at the source code-level to avoid the complexity of dealing with the details of low-level code. To make it easier to build low-level analyses, our project proposes a modular framework for building pipelines of cooperating decompilers that rewrite low-level code into higher-level intermediate languages until it becomes appropriate for source-level tools. Each decompilation stage contains an abstract interpreter that encapsulates its findings about the program by translating the program into a particular intermediate language.

PAST RESEARCH PROJECTS AND EXPERIENCE

Microsoft Research (Redmond, WA) 2005

Inferring Object Invariants

Under Dr. 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

Under Prof. George Necula and with 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

Under Prof. 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

Under Dr. 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

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

Under Prof. Robert Harper and Prof. 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

Under Prof. Frank Pfenning and with 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

- 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 Liszka, 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

- 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

- 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 (Boulder, CO) Fundamentals of Programming Languages , <i>Instructor</i> Core graduate-level course on the fundamental ideas behind modern programming language design and analysis.	Spring 2009
University of California, Berkeley (Berkeley, CA) Programming Languages and Compilers , <i>Graduate Student Instructor</i> 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.	Spring 2004
Carnegie Mellon University (Pittsburgh, PA) Principles of Programming , <i>Teaching Assistant</i> Lower division course on abstraction and reasoning about programs and functional programming (taught in Standard ML), assisting Prof. Karl Crary and Prof. John Lafferty.	Fall 2000
Carnegie Mellon University (Pittsburgh, PA) Fundamentals of Computer Science I , <i>Teaching Assistant</i> Lower division course on data structures and algorithms in C++, assisting Prof. Klaus Sutner.	Spring 1999
Carnegie Mellon University (Pittsburgh, PA) Mathematical Foundations of Computer Science , <i>Teaching Assistant</i> Lower division course on fundamental concepts of discrete mathematics using Mathematica, assisting Prof. Edmund Clarke and Prof. Klaus Sutner.	Fall 1999

AWARDS AND HONORS

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

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

External Review Committees

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

External Reviews

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

Departmental Service

Department of Computer Science. University of Colorado, Boulder

Graduate Committee: 2009

Computer Science Division. University of California, Berkeley.

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

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