

Address:
anonymized for web

Phone: *anonymized for web*
Email: philip@pgbovine.net
Website: www.pgbovine.net

RESEARCH INTERESTS

Human-computer interaction [4,12,17,21,23], online learning technologies [1], program analysis [5,9,11,13,14,15,19,20], software engineering [2,7,8,10], scientific data provenance [3,6,16,18,22]

EDUCATION

- 09/2006 – 06/2012 **Stanford University**, Stanford, CA
Ph.D. in Computer Science
Dissertation: *Software Tools to Facilitate Research Programming*
Committee: Dawson Engler, Jeffrey Heer, Margo Seltzer (Harvard University)
- 06/2005 – 06/2006 **Massachusetts Institute of Technology**, Cambridge, MA
Master of Engineering in Electrical Engineering and Computer Science
Master's Thesis: *A Scalable Mixed-Level Approach to Dynamic Analysis of C and C++ Programs* (winner of the annual department master's thesis award)
Thesis Advisor: Michael Ernst
- 09/2001 – 06/2005 **Massachusetts Institute of Technology**, Cambridge, MA
Bachelor of Science in Electrical Engineering and Computer Science
5.0 / 5.0 GPA

EMPLOYMENT

- 07/2012 – 02/2013 **Google**, Mountain View, CA
Software Engineer – Online Education Group (within the research division)
Worked on my own Online Python Tutor project (www.pythontutor.com) [1], which attracted over 200,000 users; built other open-source tools for online education.
- 09/2006 – 06/2012 **Stanford University**, Stanford, CA
Research and Course Assistant
Ph.D. dissertation published as [3,4,5,6,16,18,19,20,22]; assisted in teaching 3 courses.
- 09/2011 – 01/2012 **Harvard University**, Cambridge, MA
Visiting Research Fellow
Developed the final portion of my dissertation [18], supervised by Prof. Margo Seltzer.
- 06/2011 – 09/2011 **Google**, Mountain View, CA
Software Engineering Intern
Continued development of CDE (www.pgbovine.net/cde.html), one of my dissertation projects that resulted in 4 publications [3,6,16,22] and over 10,000 users.
- 06/2009 – 09/2009 **Microsoft Research**, Redmond, WA
Research Intern
Empirical software engineering research on how professional programmers deal with software bugs, which led to 3 papers [2,7,8], including one *Best Paper Award*.
- 06/2007 – 09/2007 **Google**, Mountain View, CA
Software Engineering Intern – prototyped memory allocators for C and C++ programs

- 01/2004 – 06/2006 **Massachusetts Institute of Technology**, Cambridge, MA
Research Assistant – Program Analysis Group – Prof. Michael Ernst
 Undergraduate and master’s research on tools for analyzing C and C++ programs, which led to 2 papers [13,14] and an award-winning master’s thesis.
- 09/2003 – 01/2004 **Massachusetts Institute of Technology**, Cambridge, MA
Research Assistant – Computer Graphics Group – Prof. Fredo Durand
 Developed an HDR (high dynamic range) image editing tool for photographers.
- SUMMER 2003 & ’04 **Teradyne**, Agoura Hills, CA
Software Engineering Intern – wrote simulators for semiconductor test hardware
- 09/2002 – 06/2003 **Massachusetts Institute of Technology**, Cambridge, MA
Research Assistant – Teacher Education Program – Prof. Eric Klopfer
 Developed a suite of 5 educational games for Palm OS devices, which were deployed in high schools throughout the U.S. and in an MIT teacher training course.
- 06/2002 – 08/2002 **Codehost**, Culver City, CA
Software Engineering Intern – wrote embedded Linux tablet PC software

TEACHING

- COURSE ASSISTANT My responsibilities as a *course assistant* at Stanford included co-creating and grading homework and exams, handling course logistics, office hours, and one-on-one tutoring.
- Stanford CS343 – Advanced Topics in Compilers (Spring quarter 2012)
 - Stanford CS242 – Programming Languages (Autumn quarter 2009)
 - Stanford CS243 – Advanced Compiling Techniques (Winter quarter 2008)
 - I also volunteered to teach a weekly lecture focused on problem-solving.
- TEACHING ASSISTANT My responsibilities as a *teaching assistant* included teaching a weekly recitation section of 25 students, holding office hours, grading, and supervising team-based final projects.
- MIT 6.170 – Laboratory in Software Engineering (Spring semester 2006)
- LABORATORY ASSISTANT My responsibilities as a *laboratory assistant* included working one-on-one with students in the computer/electronics laboratory to explain concepts and to debug lab projects.
- MIT 6.170 – Laboratory in Software Engineering (Fall semester 2004)
 - MIT 6.111 – Introductory Digital Systems Laboratory (Fall semester 2004)
 - MIT 6.001 – Structure and Interpretation of Computer Programs (Spring 2002)
- INDEPENDENT TEACHING For these independent teaching jobs, my responsibilities included customized curriculum development and in-depth personalized tutoring of nontraditional students.
- Taught Python, JavaScript, and web application programming to Brian Goler, a San Francisco Bay Area entrepreneur with a background in sales and marketing, throughout ~50 one-on-one project-based tutoring sessions (07/2011 – 03/2012)
 - Taught introductory Python programming to 10 non-CS majors (e.g., technical writers, biologists, business analysts) at 23andMe, a biotech startup in the San Francisco Bay Area (04/2008 – 07/2008)

STUDENT ADVISING

- FALL 2012 Hung Doan, undergraduate at National University of Singapore
 Project: *A JavaScript backend for Online Python Tutor*
- FALL 2012 Ankur Ankan, undergraduate at IT-BHU (Banaras Hindu University, India)
 Project: *A pure client-side port of Online Python Tutor*

SPRING 2009	Roger Liao, undergraduate at Stanford Project: <i>Cross-checking network router code using the Klee test-generation tool</i>
SPRING 2007	Nazia Zaman, master's student at Stanford Project: <i>Finding bugs in Linux device drivers using the Klee test-generation tool</i>

REFEREED PUBLICATIONS

- CONFERENCE PAPERS
- [1] **Philip J. Guo**. Online Python Tutor: Embeddable Web-Based Program Visualization for CS Education. In Proceedings of SIGCSE 2013: *ACM Technical Symposium on Computer Science Education*, March 2013 (to appear).
 - [2] Thomas Zimmermann, Nachiappan Nagappan, **Philip J. Guo**, Brendan Murphy. Characterizing and Predicting Which Bugs Get Reopened. In Proceedings of ICSE 2012: *ACM/IEEE International Conference on Software Engineering*, Software Engineering In Practice track, June 2012. (**Best Paper Award**)
 - [3] **Philip J. Guo**. CDE: Run Any Linux Application On-Demand Without Installation. In Proceedings of LISA 2011: *USENIX Large Installation System Administration Conference*, December 2011.
 - [4] **Philip J. Guo**, Sean Kandel, Joseph M. Hellerstein, Jeffrey Heer. Proactive Wrangling: Mixed-Initiative End-User Programming of Data Transformation Scripts. In Proceedings of UIST 2011: *ACM Symposium on User Interface Software and Technology*, October 2011.
 - [5] **Philip J. Guo** and Dawson Engler. Using Automatic Persistent Memoization to Facilitate Data Analysis Scripting. In Proceedings of ISSTA 2011: *ACM International Symposium on Software Testing and Analysis*, July 2011.
 - [6] **Philip J. Guo** and Dawson Engler. CDE: Using System Call Interposition to Automatically Create Portable Software Packages. Short paper in Proceedings of USENIX 2011: *USENIX Annual Technical Conference*, June 2011.
 - [7] **Philip J. Guo**, Thomas Zimmermann, Nachiappan Nagappan, Brendan Murphy. “Not My Bug!” and Other Reasons for Software Bug Report Reassignments. In Proceedings of CSCW 2011: *ACM Conference on Computer Supported Cooperative Work*, March 2011.
 - [8] **Philip J. Guo**, Thomas Zimmermann, Nachiappan Nagappan, Brendan Murphy. Characterizing and Predicting Which Bugs Get Fixed: An Empirical Study of Microsoft Windows. In Proceedings of ICSE 2010: *ACM/IEEE International Conference on Software Engineering*, May 2010.
 - [9] Adam Kiezun, Vijay Ganesh, **Philip J. Guo**, Pieter Hooimeijer, Michael D. Ernst. HAMPI: A Solver for String Constraints. In Proceedings of ISSTA 2009: *ACM International Symposium on Software Testing and Analysis*, July 2009. (ACM SIGSOFT **Distinguished Paper Award**)
 - [10] **Philip J. Guo** and Dawson Engler. Linux Kernel Developer Responses to Static Analysis Bug Reports. Short paper in Proceedings of USENIX 2009: *USENIX Annual Technical Conference*, June 2009.
 - [11] Adam Kiezun, **Philip J. Guo**, Karthick Jayaraman, Michael D. Ernst. Automatic Creation of SQL Injection and Cross-site Scripting Attacks. In Proceedings of ICSE 2009: *ACM/IEEE International Conference on Software Engineering*, May 2009.

- [12] Joel Brandt, **Philip J. Guo**, Joel Lewenstein, Mira Dontcheva, Scott R. Klemmer. Two Studies of Opportunistic Programming: Interleaving Web Foraging, Learning, and Writing Code. In *Proceedings of CHI 2009: ACM Conference on Human Factors in Computing Systems*, April 2009. (**Best Paper Honorable Mention**)
- [13] **Philip J. Guo**, Jeff H. Perkins, Stephen McCamant, Michael D. Ernst. Dynamic Inference of Abstract Types. In *Proceedings of ISSTA 2006: ACM International Symposium on Software Testing and Analysis*, July 2006.
- [14] Brian Demsky, Michael D. Ernst, **Philip J. Guo**, Stephen McCamant, Jeff H. Perkins, Martin Rinard. Automatic Inference and Enforcement of Data Structure Consistency Specifications. In *Proceedings of ISSTA 2006: ACM International Symposium on Software Testing and Analysis*, July 2006.
- JOURNAL ARTICLES [15] Michael D. Ernst, Jeff H. Perkins, **Philip J. Guo**, Stephen McCamant, Carlos Pacheco, Matthew S. Tschantz, Chen Xiao. The Daikon system for dynamic detection of likely invariants. In *Science of Computer Programming*, 2007.
- MAGAZINE ARTICLES [16] **Philip J. Guo**. CDE: A Tool For Creating Portable Experimental Software Packages. In *Computing in Science and Engineering: Special Issue on Software for Reproducible Computational Science*, Jul/Aug 2012.
- [17] Joel Brandt, **Philip J. Guo**, Joel Lewenstein, Mira Dontcheva, Scott R. Klemmer. Opportunistic Programming: Writing Code to Prototype, Ideate, and Discover. In *IEEE Software: Special Issue on End-User Software Engineering*, Sep/Oct 2009.
- WORKSHOP PAPERS [18] **Philip J. Guo** and Margo Seltzer. Burrrito: Wrapping Your Lab Notebook in Computational Infrastructure. In *Proceedings of TaPP 2012: USENIX Workshop on the Theory and Practice of Provenance*, June 2012.
- [19] **Philip J. Guo**. Sloppy Python: Using Dynamic Analysis to Automatically Add Error Tolerance to Ad-Hoc Data Processing Scripts. In *Proceedings of WODA 2011: ACM International Workshop on Dynamic Analysis*, July 2011.
- [20] **Philip J. Guo** and Dawson Engler. Towards Practical Incremental Recomputation for Scientists: An Implementation for the Python Language. In *Proceedings of TaPP 2010: USENIX Workshop on the Theory and Practice of Provenance*, February 2010.
- [21] Joel Brandt, **Philip J. Guo**, Joel Lewenstein, Scott R. Klemmer. Opportunistic Programming: How Rapid Ideation and Prototyping Occur in Practice. In *WEUSE 2008: ACM Workshop on End-User Software Engineering*, May 2008.

INVITED BOOK CHAPTERS

- [22] **Philip J. Guo**. CDE: Automatically Package and Reproduce Computational Experiments. Book chapter in *Implementing Reproducible Computational Research*, edited by V. Stodden, F. Leisch, R. Peng. Taylor & Francis Group, 2013 (to appear).
- [23] Joel Brandt, **Philip J. Guo**, Joel Lewenstein, Mira Dontcheva, Scott R. Klemmer. How the Web Helps People Turn Ideas Into Code. Book chapter in *No Code Required: Giving Users Tools to Transform the Web*, edited by A. Cypher, M. Dontcheva, T. Lau, J. Nichols. Morgan Kaufmann, 2010.

TALKS

CONFERENCE & WORKSHOP PAPER PRESENTATIONS

- Burrito: Wrapping Your Lab Notebook in Computational Infrastructure. *USENIX Workshop on the Theory and Practice of Provenance*, Boston, MA, June 2012.
- CDE: Run Any Linux Application On-Demand Without Installation. *USENIX Large Installation System Administration Conference*, Boston, MA, Dec 2011.
- Proactive Wrangling: Mixed-Initiative End-User Programming of Data Transformation Scripts. *ACM Symposium on User Interface Software and Technology*, Santa Barbara, CA, Oct 2011.
- Using Automatic Persistent Memoization to Facilitate Data Analysis Scripting. *ACM International Symposium on Software Testing and Analysis*, Toronto, Canada, July 2011.
- Sloppy Python: Using Dynamic Analysis to Automatically Add Error Tolerance to Ad-Hoc Data Processing Scripts. *ACM International Workshop on Dynamic Analysis*, Toronto, Canada, July 2011.
- CDE: Using System Call Interposition to Automatically Create Portable Software Packages. *USENIX Annual Technical Conference*, Portland, OR, June 2011.
- Towards Practical Incremental Recomputation for Scientists. *USENIX Workshop on the Theory and Practice of Provenance*, San Jose, CA, Feb 2010.
- Linux kernel developer responses to static analysis bug reports. *USENIX Annual Technical Conference*, San Diego, CA, June 2009.
- Automatic Creation of SQL Injection and Cross-site Scripting Attacks. *ACM/IEEE International Conference on Software Engineering*, Vancouver, Canada, May 2009.
- Dynamic Inference of Abstract Types. *ACM International Symposium on Software Testing and Analysis*, Portland, ME, July 2006.

INVITED TALKS

- Online Python Tutor: Web-Based Program Visualization for CS Education. *Sonoma State University – Computer Science Colloquium*, Rohnert Park, CA, Nov 2012.
- The Ph.D. Grind: Candid Discussions About Ph.D. Life. *UC Riverside – Computer Science Colloquium*, Riverside, CA, Oct 2012.
- Online Python Tutor. *Hacker School*, New York, NY, Oct 2012.
- The Ph.D. Grind: Candid Discussions About Ph.D. Life. *Google Tech Talk*, Mountain View, CA, August 2012.
- CDE: A tool for automatically creating reproducible experimental software packages. *Reproducible Research: Tools and Strategies for Scientific Computing* interdisciplinary meeting, Vancouver, Canada, July 2011.
- CDE: Using System Call Interposition to Automatically Create Portable Software Packages. *Google Tech Talk*, Mountain View, CA, Feb 2011.
- The potentials and challenges of implementing automatic test generation using combined concrete and symbolic execution. *Fujitsu*, Sunnyvale, CA, Oct 2009.
- Happiness and Academic High-Achievers. *Sun Yat-Sen University Alumni Association – Northern California*, Cupertino, CA, Oct 2009.
- Automatic Creation of SQL Injection and Cross-site Scripting Attacks. *Samsung R&D Center*, San Jose, CA, May 2009.

AWARDS AND HONORS

06/2012	ICSE Software Engineering In Practice Best Paper Award for <i>Characterizing and Predicting Which Bugs Get Reopened</i>
07/2009	ACM SIGSOFT Distinguished Paper Award for <i>HAMPI: A Solver for String Constraints</i>
04/2009	CHI Best Paper Honorable Mention for <i>Two Studies of Opportunistic Programming: Interleaving Web Foraging, Learning, and Writing Code</i>
09/2009 – 06/2011	National Science Foundation (NSF) Graduate Fellowship
09/2006 – 09/2009	National Defense Science and Engineering (NDSEG) Graduate Fellowship
05/2006	MIT Charles and Jennifer Johnson Thesis Award for Outstanding Computer Science Master of Engineering Thesis

PROFESSIONAL SERVICE

PROGRAM COMMITTEE	TaPP 2012 (Workshop on the Theory and Practice of Provenance), PLATEAU 2012 (Workshop on Evaluation and Usability of Programming Languages and Tools)
PAPER REVIEWER	PLDI 2013 (Conference on Programming Language Design and Implementation), POPL 2011 (Symposium on Principles of Programming Languages), ECOOP 2006 and 2009 (European Conference on Object-Oriented Programming)
ARTIFACT EVAL. COMMITTEE	ESEC/FSE 2011 (Symposium on the Foundations of Software Engineering)

PERSONAL

Since 1997, I have maintained a personal website (www.pgbovine.net) with over 100 articles on topics ranging from computer programming to academic advice for undergraduate and graduate students. Over 250,000 people visit my website each year.

In June 2012, I wrote *The Ph.D. Grind* (phdgrind.com), a free e-book that is the first known detailed account of an entire Ph.D. experience. So far, over 100,000 people have downloaded it, and hundreds of readers have emailed me with heartfelt responses.

REFERENCES

anonymized for web; please email me for their contact information.