Pranav Garg

Department of Computer Science, 201 North Goodwin Avenue, Urbana, IL 61801
1.217.377.5608 • garg11@illinois.edu • web.engr.illinois.edu/∼garg11

Research Interests

Ensuring correctness, reliability, and security of software systems. My research focuses on building automatic techniques based on machine learning that significantly lessen the burden on a programmer trying to prove her program secure or correct.

Education

University of Illinois at Urbana-Champaign 2015 (Expected)
Ph.D., Department of Computer Science
Thesis: Learning-based Automatic Program Verification
Advisor: Madhusudan Parthasarathy

Indian Institute of Technology Kanpur, India 2009
B.Tech., Department of Computer Science and Engineering

Research Experience

University of Illinois at Urbana-Champaign
Research Assistant, Department of Computer Science 2009 - present
Verification technology that enables building verified and secure software systems with very little manual effort. Developed learning based automatic software verification including machine learning algorithms for learning inductive program invariants that scale to real software [1, 4, 5, 12] and fully automatable natural proofs [3, 7, 11], thereby significantly reducing the annotation burden on the programmer verifying such a system (partly supported by NSF grant on ExCAPE: Expeditions in Computer Augmented Program Engineering).

Microsoft Research India Bangalore, India
Research Intern Spring 2013
Designed and developed new deductive verifiers for concurrent programs. Further, developed static program analysis techniques for proving absence of security bugs, like buffer overflows, in concurrent Linux device drivers by automated learning of precise, adequate inductive invariants.
Mentor: Akash Lal

NEC Laboratories America Princeton, NJ
Research Intern Summer 2011
Designed and developed a hybrid unit test generation technique that combines random test generation with symbolic execution to significantly improve the code coverage provided by a feedback-directed random unit testing framework [8, 15].
Mentor: Franjo Ivančić

École Polytechnique Fédérale de Lausanne (EPFL) Lausanne, Switzerland
Undergraduate Research Intern Summer 2008
Full-blown functional verification of a deadlock immunity system nucleus using automated theorem provers. The verified nucleus helps guarantee deadlock immunity in large Java systems with modest runtime and verification overheads.
Advisor: George Candea

Awards

Travel grant for the following conferences: MVD 2010, POPL 2011, ISCA 2011, PLDI 2013 and CAV 2014.

Indian National Physics Olympiad 2005.

Gold Medal for best individual and team performance at International Young Mathematicians' Convention 2004.

National Talent Search Scholarship awarded by the Government of India 2003-2008.

Publications

Journal Articles

[1] Pranav Garg, Christof Löding, P. Madhusudan, and Daniel Neider. Quantified Data Automata for Linear Data Structures. A Register Automaton Model with Applications to Learning Invariants of Programs Manipulating Arrays and Lists. (Invited Paper). *Formal Methods in System Design (FMSD)*, Special Issue on Computer Aided Verification, 2014. Under Review.

[2] Rajeev Alur, Rastislav Bodik, Eric Dallal, Dana Fisman, Pranav Garg, Garvit Juniwal, Hadas Kress-Gazit, P. Madhusudan, Milo M. K. Martin, Mukund Raghothaman, Shamwaditya Saha, Sanjit A. Seshia, Rishabh Singh, Armando Solar-Lezama, Emina Torlak, and Abhishek Udupa. Syntax-Guided Synthesis. (Invited Paper). *NATO Proceedings of the Maktoberdorf Summer School*, 2014. Under Review.

Conference Papers

[3] Ankush Desai, Pranav Garg, and P. Madhusudan. Natural Proofs for Asynchronous Programs using Almost-Synchronous Invariants. In *Proceedings of the 28th ACM International Conference on Object Oriented Programming Systems Languages & Applications* **(OOPSLA)**, 2014.

[4] Pranav Garg, Christof Löding, P. Madhusudan, and Daniel Neider. ICE: A Robust Framework for Learning Invariants. In *Proceedings of the 26th International Conference on Computer Aided Verification* **(CAV)**, 2014. **Invited for submission to Journal of the ACM (JACM)**.

[5] Pranav Garg, Christof Löding, P. Madhusudan, and Daniel Neider. Learning Universally Quantified Invariants of Linear Data Structures. In *Proceedings of the 25th International Conference on Computer Aided Verification* **(CAV)**, 2013. **Invited for submission to Formal Methods in System Design (FMSD)**.

[6] Pranav Garg, P. Madhusudan and Gennaro Parlato. Quantified Data Automata on Skinny Trees: An Abstract Domain for Lists. In *Proceedings of the 20th Static Analysis Symposium* **(SAS)**, 2013.

[7] Xiaokang Qiu, Pranav Garg, Andrei Stefanescu, and P. Madhusudan. Natural Proofs for Structure, Data, and Separation. In *Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation* **(PLDI)**, 2013.

[8] Pranav Garg, Franjo Ivancic, Gogul Balakrishnan, Naoto Maeda, and Aarti Gupta. Feedback-Directed Unit Test Generation for C/C++ using Concolic Execution. In *Proceedings of the 35th International Conference on Software Engineering* **(ICSE)**, 2013.

[9] Rishi Agarwal, Pranav Garg, and Josep Torrellas. Rebound: Scalable Checkpointing for Coherent Shared Memory. In *Proceedings of the 38th International Symposium on Computer Architecture* **(ISCA)**, 2011.

[10] Pranav Garg and P. Madhusudan. Compositionality Entails Sequentializability. In *Proceedings of the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems* **(TACAS)**, 2011.

Workshop Papers

[11] Ankush Desai, Pranav Garg, and P. Madhusudan. A New Reduction for Event-Driven Distributed Programs. In *7th International Workshop on Exploiting Concurrency Efficiently and Correctly* **(EC2)**, 2014.

Articles under Submission

[12] Pranav Garg, P. Madhusudan, Daniel Neider, and Dan Roth. Learning Invariants Using Decision Trees and Implication Counterexamples. 2015. Under Submission.

[13] Alex Gyori, Pranav Garg, Edgar Pek, and P. Madhusudan. Abstraction-guided Runtime Checking of Assertions on Lists. 2015. Under Submission.

[14] Shambwaditya Saha, Pranav Garg, and P. Madhusudan. Alchemist: Learning Guarded Affine Functions. 2015. Under Submission.

Patents

[15] Pranav Garg, Franjo Ivancic, Gogul Balakrishnan, Naoto Maeda, and Aarti Gupta. Feedback-Directed Random Class Unit Test Generation Using Symbolic Execution. US Patent Number 20130091495, Issued April 11, 2013.

Software

ICE-DTREE — A static verification tool for Boogie (an intermediate verification language from Microsoft with translators from several high-level languages like C and C# into Boogie) programs. ICE-DTREE automatically verifies programs by learning inductive program invariants using custom decision tree learning algorithms [12].

ICE-CS — A static verification tool, built on the Boogie platform from Microsoft, that uses algorithms based on constraint-solvers for learning inductive program invariants [4].

ALCHEMIST — A SyGuS (Syntax-Guided Synthesis) solver that learns linear integer arithmetic programs. ALCHEMIST uses computational geometry and decision tree learning algorithms to synthesize non-convex piece-wise linear functions from logical specifications [2].

P-ASI — An automatic tool for verifying P systems using a combination of natural proof tactics and invariant generation using model-checking [3, 11]. P is a domain-specific language from Microsoft for writing event-driven device-drivers. P-ASI verified the responsiveness of the USB device-driver stack in Microsoft Windows phone, and is being used at Microsoft.

RANDOOP — Automatic unit test generation tool for C/C++ programs that combines feedback-directed random unit testing with symbolic execution to obtain improved code coverage [8, 15]. The tool was developed as part of an internship at NEC Laboratories America.

QDA-LEARN — An automatic tool that uses Angluin's L* algorithm to generate *likely* quantified invariants for programs over arrays and list structures from dynamic test runs [1, 5].

QDA-ABSINT — An abstract interpretation based prototype tool that automatically synthesizes quantified invariants for programs over arrays and lists structures, using an automata based abstract domain [6].

Talks

Black Box Invariant Synthesis

Expeditions in Computer Augmented Program Engineering (ExCAPE) Annual Research Meeting, Berkeley, CA. — Mar 2014

ICE: A Robust Learning Framework for Synthesizing Invariants
The Midwest Verification Day (MVD), Chicago, IL. Sep 2013

Learning Universally Quantified Invariants of Linear Data Structures
25th International Conference on Computer Aided Verification (CAV), Saint Petersburg, Russia. Jul 2013

Quantified Data Automata on Skinny Trees: An Abstract Domain for Lists
20th Static Analysis Symposium (SAS), Seattle, WA. Jun 2013

Synthesizing Universally Quantified Invariants of Linear Data Structures
Indian Institute of Science, Bangalore, India. Invited Talk. May 2013

Rebound: Scalable Checkpointing for Coherent Shared Memory
Illinois-Intel Parallelism Center (I2PC) Distinguished Speaker Series, University of Illinois at Urbana-Champaign, IL. Invited Talk. Sep 2011

38th International Symposium on Computer Architecture (ISCA), San Jose, CA. Jun 2011

NEC Laboratories America, Princeton, NJ. Jun 2011

Compositionality Entails Sequentializability
17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Saarbrücken, Germany. Mar 2011

The Midwest Verification Day (MVD), Iowa City, IA. Sep 2010

Teaching Experience

Teaching Assistant
Software Engineering II CS428 (Undergraduate course, UIUC) Spring 2014
Closely mentored multiple student teams on their course projects, prepared iteration deliverables for them and monitored their progress throughout the semester. Created and graded a final exam, and gave a guest lecture on Software Quality Assurance.

Instructor
ACA Summer School on Data Structures and Algorithms (Workshop, IITK) Summer 2009
Designed the syllabus and gave lectures on data structures and algorithms and on C++ programming. The target audience was engineering undergraduate students, including those from non computer-science backgrounds.

Service

Organization
LEARN: Tutorial on Learning based methods in Design and Analysis of Systems, co-located with Computer Aided Verification (CAV) 2015

Program Committee Membership
International Symposium on Software Testing and Analysis (ISSTA), Artifact Evaluation 2015

Refereeing and Reviewing
Programming Language Design and Implementation (PLDI) 2014

Computer Aided Verification (CAV) 2011, 2012, 2013, 2015

Tools and Algorithms for the Construction and Analysis of Systems (TACAS) 2012
Concurrency Theory (CONCUR) 2011

Departmental and University Service

Member of the CS Graduate Academic Council, UIUC. 2014-15
Proposed and reviewed new initiatives to enhance graduate education, organize academic seminars, and enhance communication networks among CS graduate students.

Student representative of the CS Graduate Admissions Committee, UIUC. 2011-12
Evaluated graduate school admission applications and made recommendations to the Admissions Committee.

Industry Experience

Software Developer

Loop Optimizations in the Gnu C Compiler (GCC), Google Summer of Code Summer 2009
Implemented the loop blocking/strip mining loop optimizations in the GCC compiler. The developed code is part of GCC version 4.6 and onwards.
Mentor: Sebastian Pop, Free Software Foundation

Embedded Software for Industrial Control Systems Summer 2006
Micro-controller programming for designing a PID controller which controlled the fueling system of an IC engine, through a close loop processing of the engine speed error data.

References

Franjo Ivančić
Senior Software Engineer
Google Inc.,
111 8th Avenue, New York, NY 10011, and
Adjunct Assistant Professor
Department of Computer Science
Columbia University
New York, NY 10027
ivancic@google.com
609-640-1473

Sumit Gulwani
Principal Researcher
Microsoft Research (Redmond Lab)
One Microsoft Way, Redmond, WA 98052
sumitg@microsoft.com
425-706-7709

Madhusudan Parthasarathy
Associate Professor
Department of Computer Science
University of Illinois at Urbana-Champaign
201 N. Goodwin Avenue, Urbana, IL 61801
madhu@illinois.edu
217-244-1323

Dan Roth
Professor
Department of Computer Science and The Beckman Institute
University of Illinois at Urbana-Champaign
201 N. Goodwin Avenue, Urbana, IL 61801
danr@illinois.edu
217-244-7068